FreeBSD ハンドブック

FreeBSD ハンドブック

改訂: 45861
2014-10-22 : ryusuke.
製作著作 © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014 The FreeBSD Documentation Project

概要

FreeBSD へようこそ! このハンドブックは FreeBSD 8.4-RELEASE, FreeBSD 9.3-RELEASE と FreeBSD 10.0-RELEASE のインストールおよび、日常での使い方について記述したものです。 本ハンドブックは改編作業中であり、 さまざまな人々が編集に携わっています。いま存在するセクションの中には情報が古くなってしまったため、更新作業の必要があるものも含まれています。もし、このハンドブックを編集するプロジェクトに協力したいとお考えなら、 FreeBSD documentation project メーリングリスト まで電子メールを(英語で)送ってください。 この文書の最新バージョンは、いつでも 日本国内版の FreeBSD ウェブサイト および FreeBSD ウェブサイト から入手できます (この文書の以前のバージョンは `http://docs.FreeBSD.org/doc/` から入手できます)。また、他のさまざまな文書形式、圧縮形式のものが FreeBSD FTP サーバ や数多くの ミラーサイト からダウンロードできます。このハンドブックの書籍版 (英語版) は、FreeBSD Mall から購入できます。また、ハンドブックの検索 を行なうこともできます。

FreeBSD ハンドブック日本語版の作成は FreeBSD 日本語ドキュメンテーションプロジェクト (FreeBSD doc-jp) がおこなっています。 ハンドブックの日本語訳に関することは FreeBSD 日本語ドキュメンテーションプロジェクト <doc-jp@jp.FreeBSD.org > において日本語で議論されています。

文書の日本語訳に関するお問い合わせや、文書の原文に関する問い合わせをしたいが英語が得意でないという方は FreeBSD 日本語ドキュメンテーションプロジェクト <doc-jp@jp.FreeBSD.org > まで、日本語でコメントをお寄せください。

Copyright

Redistribution and use in source (XML DocBook) and 'compiled' forms (XML, HTML, PDF, PostScript, RTF and so forth) with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code (XML DocBook) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.

2. Redistributions in compiled form (transformed to other DTDs, converted to PDF, PostScript, RTF and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

重要

THIS DOCUMENTATION IS PROVIDED BY THE FREEBSD DOCUMENTATION PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD DOCUMENTATION PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

FreeBSD は The FreeBSD Foundation の登録商標です。

3Com および HomeConnect は 3Com Corporation の登録商標です。

3ware は 3ware Inc. の登録商標です。

ARM は ARM Limited の登録商標です。

Adaptec は Adaptec, Inc. の登録商標です。

Adobe, Acrobat, Acrobat Reader, Flash および PostScript は アメリカ合衆国および/またはその他の国の Adobe Systems Incorporated の登録商標または商標です。

Apple, AirPort, FireWire, iMac, iPhone, iPad, Mac, Macintosh, Mac OS, Quicktime および TrueType は Apple Inc. の商標で、アメリカ合衆国およびその他の国で登録されています。

Sound Blaster は アメリカ合衆国および/またはその他の国における Creative Technology Ltd. の商標です。

Android は Google Inc. の商標です。

Heidelberg, Helvetica, Palatino, および Times Roman はアメリカ合衆国およびその他の国における Heidelberger Druckmaschinen AG の登録商標または商標です。

IBM, AIX, OS/2, PowerPC, PS/2, S/390 および ThinkPad は アメリカ合衆国、その他の国、または両方における International Business Machines Corporation の商標です。

IEEE, POSIX および 802 は アメリカ合衆国における Institute of Electrical and Electronics Engineers, Inc. の登録商標です。

Intel, Celeron, Centrino, Core, EtherExpress, i386, i486, Itanium, Pentium および Xeon はアメリカ合衆国およびその他の国における Intel Corporation またはその関連会社の商標または登録商標です。

Intuit および Quicken は アメリカ合衆国およびその他の国における Intuit Inc. またはその関連会社の 商標および/または登録商標です。

Linux は Linus Torvalds の登録商標です。

LSI Logic, AcceleRAID, eXtremeRAID, MegaRAID および Mylex は LSI Logic Corp. の商標または登録商標です。

Microsoft, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media および Windows NT は アメリカ合衆国および/またはその他の国における Microsoft Corporation の登録商標または商標です。

Motif, OSF/1 および UNIX は アメリカ合衆国およびその他の国における The Open Group の登録商標で、IT DialTone および The Open Group は同じく商標です。

Oracle は Oracle Corporation の登録商標です。

RealNetworks, RealPlayer および RealAudio は RealNetworks, Inc. の登録商標です。

Red Hat, RPM は アメリカ合衆国およびその他の国における Red Hat, Inc. の商標または登録商標です。

Sun, Sun Microsystems, Java, Java Virtual Machine, JDK, JRE, JSP, JVM, Netra, OpenJDK, Solaris, StarOffice, SunOS および VirtualBox は アメリカ合衆国およびその他の国における Sun Microsystems, Inc. の 商標または登録商標です。

MATLAB は The MathWorks, Inc. の登録商標です。

SpeedTouch は Thomson の商標です。

U.S. Robotics および Sportster は U.S. Robotics Corporation の登録商標です。

VMware は VMware, Inc. の商標です。

Waterloo Maple および Maple は Waterloo Maple Inc. の商標または登録商標です。

Mathematica は Wolfram Research, Inc. の登録商標です。

XFree86 は The XFree86 Project, Inc. の商標です。

Ogg Vorbis および Xiph.Org は Xiph.Org の商標です。

製造者および販売者が製品を区別するのに 用いている表示の多くは、商標とされています。この文書に登場する表示のうち FreeBSD Project がその商標を確認しているものには、その表示に続いて「#」または「®」記号がおかれています。

目次

前書き xix
I. 導入 1
1. はじめに 5
1.1. この章では 5
1.2. FreeBSD へようこそ! 5
1.3. FreeBSD プロジェクトについて 9
2. FreeBSD のインストール (9.X 以降の場合) 15
2.1. この章では 15
2.2. ハードウェア要件 16
2.3. インストール前に行う作業 17
2.4. インストールの開始 24
2.5. bsdinstall の紹介 29
2.6. ネットワークからのインストール 33
2.7. ディスク領域の割り当て 34
2.8. インストール操作の確定 41
2.9. インストール後の作業 44
2.10. トラブルシューティング 64
2.11. Live CD を使う 65
3. FreeBSD 8.X のインストール 67
3.1. この章では 67
3.2. ハードウェア要件 68
3.3. インストール前に行う作業 69
3.4. インストールの開始 78
3.5. sysinstall(8) の紹介 85
3.6. ディスク領域の割り当て 90
3.7. 何をインストールするか選択する 106
3.8. インストールメディアの選択 108
3.9. インストール操作の確定 110
3.10. インストール後の作業 111
3.11. トラブルシューティング 143
3.12. 高度なインストールガイド 148
3.13. カスタムインストールメディアの準備 150
4. UNIX の基礎知識 157
4.1. この章では 157
4.2. 仮想コンソールと端末 157
4.3. 許可属性 162
4.4. ディレクトリ構造 165
4.5. ディスク構成 168
4.6. ファイルシステムのマウントとアンマウント 177
4.7. プロセス 180
4.8. デーモン、シグナルとプロセス終了 182
4.9. シェル 185
4.10. テキストエディタ 188
4.11. デバイスとデバイスノード 189

4.12. さらに詳しい情報を得るには... 189
5. アプリケーションのインストール - packages と ports 193
5.1. この章では 193
5.2. ソフトウェアのインストール 193
5.3. ソフトウェアの探し方 195
5.4. pkgng によるバイナリ package の管理 197
5.5. Ports Collection の利用 203
5.6. インストール後の作業 213
5.7. うまく動作しない ports に遭遇した場合には 214
6. X Window System 217
6.1. この章では 217
6.2. 用語の説明 217
6.3. X11 のインストール 219
6.4. X11 の設定 220
6.5. X11 でのフォントの使用 227
6.6. X ディスプレイマネージャ 232
6.7. デスクトップ環境 235
II. 日々の生活 241
7. デスクトップアプリケーション 245
7.1. この章では 245
7.2. ブラウザ 246
7.3. 生産的なアプリケーション 251
7.4. ドキュメントビューア 255
7.5. 財務管理ソフトウェア 257
8. マルチメディア 261
8.1. この章では 261
8.2. サウンドカードの設定 262
8.3. MP3 オーディオ 267
8.4. ビデオ再生 270
8.5. TV カードの設定 276
8.6. MythTV 278
8.7. 画像スキャナ 279
9. FreeBSD カーネルのコンフィグレーション 285
9.1. この章では 285
9.2. なぜカスタムカーネルを作るか? 285
9.3. システムのハードウェアについて知る 287
9.4. コンフィグレーションファイル 288
9.5. カスタムカーネルの構築とインストール 290
9.6. 問題が起きた場合には 291
10. プリンタの利用 293
10.1. この章では 293
10.2. はじめに 293
10.3. 基本的な設定 295
10.4. プリンタ設定上級編 311
10.5. プリンタを使う 348
10.6. 標準スプーラの代替品 357

10.7. トラブルシューティング 358
11. Linux バイナリ互換機能 365
11.1. この章では 365
11.2. インストール 366
11.3. Mathematica のインストール 370
11.4. Oracle のインストール 372
11.5. SAP R/3 (4.6B - IDES) のインストール 376
11.6. 高度なトピックス 395
III. システム管理 399
12. 設定とチューニング 403
12.1. この章では 403
12.2. 中核となる設定 403
12.3. アプリケーションの設定 404
12.4. サービスの起動 405
12.5. バーチャルホスト 406
12.6. 設定ファイル 407
12.7. sysctl によるチューニング 412
12.8. ディスクのチューニング 412
12.9. Kernel 制限のチューニング 414
13. FreeBSD の起動のプロセス 417
13.1. この章では 417
13.2. 起動時の問題 417
13.3. ブートマネージャと起動ステージ 419
13.4. カーネル起動時の応答 424
13.5. Device Hints 425
13.6. init: プロセス制御の初期化 426
13.7. シャットダウン動作 427
14. ユーザと基本的なアカウントの管理 429
14.1. この章では 429
14.2. アカウントの種類 429
14.3. アカウント情報の管理 432
14.4. ユーザへの制限 438
14.5. グループの管理 441
15. セキュリティ 445
15.1. この章では 445
15.2. はじめに 445
15.3. FreeBSDの安全性を高める 448
15.4. DES, MD5 と Crypt 458
15.5. S/Key 459
15.6. Kerberos 466
15.7. ファイアウォール 474
15.8. OpenSSL 486
15.9. IPsec 486
15.10. OpenSSH 493
15.11. ファイルシステムアクセス制御リスト 498
16. ストレージ 501

16.1. この章では 501
16.2. デバイス名 501
16.3. ディスクの追加 502
16.4. RAID 505
16.5. 光メディア (CD & DVD) の作成と使用 511
16.6. フロッピーディスクの作成と使用 518
16.7. データテープの作成と使用 521
16.8. フロッピーディスクへのバックアップ 524
16.9. バックアップの基本 526
16.10. ネットワーク、メモリ、そしてファイルベースのファイルシステム 535
16.11. ファイルシステムのスナップショット 540
16.12. ファイルシステムクォータ 542
16.13. ディスクパーティションの暗号化 546
17. 地域化 (localization) - I18N/L10N の利用と設定 553
17.1. この章では 553
17.2. 基礎知識 553
17.3. 地域化の利用 554
17.4. 国際化に対応したプログラムのコンパイル 561
17.5. FreeBSD を特定の言語に地域化する 561
18. FreeBSD のアップデートとアップグレード 565
18.1. この章では 565
18.2. FreeBSD Update 566
18.3. ドキュメントのアップデート 575
18.4. 開発ブランチを追いかける 579
18.5. ソースの同期 582
18.6. world の再構築 583
18.7. 複数のマシンで追いかける 595
IV. ネットワーク通信 597
19. シリアル通信 601
19.1. この章では 601
19.2. はじめに 601
19.3. シリアル端末 607
19.4. ダイアルインサービス 612
19.5. ダイアルアウトサービス 622
19.6. シリアルコンソールの設定 626
20. PPP と SLIP 637
20.1. この章では 637
20.2. ユーザ ppp の利用 637
20.3. カーネル PPP の利用 653
20.4. PPP オーバイーサネット (PPPoE) の利用 662
20.5. SLIP の利用 663
21. 電子メール 677
21.1. この章では 677
21.2. 電子メールを使う 678
21.3. sendmail の設定 679

21.4. MTA の変更 682
21.5. トラブルシュート 686
21.6. 先進的なトピックス 689
21.7. UUCP とともに SMTP を使う 692
21.8. ダイアルアップ接続でメールを使う 694
21.9. SMTP 認証 696
22. 高度なネットワーク 699
22.1. この章では 699
22.2. ゲートウェイと経路 699
22.3. 無線ネットワーク 707
22.4. Bluetooth 714
22.5. ブリッジ 724
22.6. NFS 727
22.7. ディスクレス稼働 734
22.8. ISDN 742
22.9. NIS/YP 747
22.10. DHCP 767
22.11. DNS 773
22.12. NTP 788
22.13. ネットワークアドレス変換 (NAT) 791
22.14. inetd「スーパサーバ」 795
22.15. パラレルライン IP (PLIP) 800
22.16. IPv6 803
V. 付録 809
A. FreeBSD の入手方法 813
A.1. CD および DVD セット 813
A.2. FTP サイト 814
A.3. CTM を使う 823
A.4. Subversion を使う 826
A.5. rsync を使用する 830
B. 参考図書 833
B.1. FreeBSD 専門の書籍 833
B.2. 利用者向けのガイド 834
B.3. 管理者向けのガイド 835
B.4. プログラマ向けのガイド 835
B.5. オペレーティングシステム内部 836
B.6. セキュリティの参考資料 837
B.7. ハードウェアの参考資料 837
B.8. UNIX® の歴史 838
B.9. 定期刊行物、雑誌およびジャーナル 839
C. インターネット上のリソース 841
C.1. メーリングリスト 841
C.2. Usenet ニュースグループ 857
C.3. World Wide Web サイト 858
C.4. Email アドレス 861
C.5. シェルアカウント 861

D. PGP 公開鍵 863
D.1. オフィサ 863
FreeBSD Glossary 881
索引 901

図の一覧

2.1. FreeBSD ブートローダメニュー 25
2.2. 典型的なデバイス検出結果の例 28
2.3. インストールメディアの選択モード 29
2.4. キーマップの選択 30
2.5. キーボードメニューの選択 30
2.6. ホスト名の設定 31
2.7. インストールするコンポーネントの設定 32
2.8. ネットワークからのインストール 33
2.9. ミラーサイトの選択 34
2.10. パーティションの分割に Guided と Manual のどちらを用いるかの選択 34
2.11. 複数のディスクから選択する 36
2.12. Entire Disk または Partition の選択 37
2.13. 作成されたパーティションの確認 37
2.14. Manual によるパーティションの分割 38
2.15. 手動でパーティションを作成する 38
2.16. 手動でパーティションを作成する 39
2.17. 最後の確認 42
2.18. 配布ファイルのダウンロード 42
2.19. 配布ファイルの検証 43
2.20. 配布ファイルの取り出し 43
2.21. `root` パスワードの設定 44
2.22. イーサネットインタフェースの選択 45
2.23. ワイヤレスアクセスポイントのスキャン 45
2.24. ワイヤレスネットワークの選択 46
2.25. WPA2 のセットアップ 46
2.26. IPv4 ネットワークの選択 47
2.27. IPv4 DHCP 設定の選択 48
2.28. IPv4 Static 設定 48
2.29. IPv6 ネットワークの設定 49
2.30. IPv6 SLAAC 設定の選択 50
2.31. IPv6 static 設定 50
2.32. DNS の設定 51
2.33. ローカルまたは UTC クロックの選択 52
2.34. 地域の選択 52
2.35. 国名の選択 53
2.36. タイムゾーンの選択 53
2.37. タイムゾーンの確定 54
2.38. 追加で有効にするサービスの選択 54
2.39. クラッシュダンプの設定 55
2.40. 新しいユーザのアカウントの作成 56
2.41. ユーザ情報の入力 56
2.42. ユーザおよびグループの管理を終了する 57
2.43. 最後の設定 58
2.44. Manual Configuration 59

2.45. インストールの終了 59
3.1. FreeBSD ブートローダメニュー 81
3.2. 古典的なデバイス検出結果の例 83
3.3. 国の選択メニュー 84
3.4. キーボードメニューの選択 85
3.5. Sysinstall のメインメニューから Usage を選択 86
3.6. ドキュメントメニューの選択 86
3.7. Sysinstall ドキュメントメニュー 87
3.8. Sysinstall メインメニュー 88
3.9. Sysinstall キー配列メニュー 88
3.10. Sysinstall メインメニュー 89
3.11. Sysinstall オプション 89
3.12. 標準インストールの開始 90
3.13. fdisk を実行するディスクの選択 93
3.14. 典型的な FDisk パーティション 95
3.15. ディスク全体を使う FDisk 構成 96
3.16. sysinstall ブートマネージャメニュー 97
3.17. ドライブ選択の終了 98
3.18. sysinstall ディスクラベルエディタ 102
3.19. デフォルトの自動割り当てによる sysinstall ディスクラベルエディタ 103
3.20. ルートパーティションに割りあてる容量 104
3.21. ルートパーティションサイズの編集 104
3.22. ルートパーティションタイプの選択 105
3.23. ルートのマウントポイント選択 105
3.24. sysinstall ディスクラベルエディタ 106
3.25. ディストリビューションの選択 107
3.26. ディストリビューションの確認 108
3.27. インストールメディアの選択 109
3.28. イーサネットデバイスを選ぶ 112
3.29. *ed0* ネットワークの設定 113
3.30. `inetd.conf` の編集 116
3.31. Anonymous FTP のデフォルトの設定 118
3.32. FTP Welcome メッセージの編集 119
3.33. `exports` ファイルの編集 120
3.34. システムコンソール設定オプション 121
3.35. スクリーンセーバオプション 121
3.36. スクリーンセーバのタイムアウト 122
3.37. システムコンソールの設定の終了 122
3.38. 地域の選択 123
3.39. 国名の選択 124
3.40. タイムゾーンの選択 124
3.41. マウスプロトコルのタイプの選択 125
3.42. マウスプロトコルの選択 126
3.43. マウスポートの設定 126
3.44. マウスポート の設定 127
3.45. マウスデーモンを有効にする。 127

3.46. マウスデーモンのテスト 128
3.47. Package カテゴリの選択 129
3.48. Packages の選択 129
3.49. Packages のインストール 130
3.50. Package のインストールの確認 131
3.51. User の選択 132
3.52. ユーザ情報の入力 132
3.53. ユーザおよびグループの管理を終了する 134
3.54. インストールの終了 135
3.55. ネットワーク設定 (メニューの前半) 136
3.56. デフォルトの MTA の選択 137
3.57. Ntpdate の設定 138
3.58. ネットワーク設定 (メニューの後半) 139

表の一覧

2.1. パーティションスキーム 39
3.1. サンプルデバイス一覧 69
3.2. 一つ目のディスクのパーティションレイアウト 99
3.3. 残りのディスクのパーティション構成 100
3.4. FreeBSD ISO イメージの名前と内容 151
4.1. ディスクデバイス記号 176
8.1. 良くあるエラーメッセージ 265
16.1. 物理ディスクへの名前付け 501
22.1. ネットワーク向けのパラレル (クロス) ケーブル結線 800
22.2. 予約された IPv6 アドレス 805

例の一覧

2.1. 既存のパーティションを使用 19
2.2. 既存のパーティションを縮小する 19
2.3. 伝統的なファイルシステムのパーティションを作成する。..... 41
3.1. 既存のパーティションを変更せずに使用 71
3.2. 既存のパーティションを縮小する 72
4.1. ディスク名、スライス名、パーティション名のサンプル 176
4.2. ディスクの概念的構成 176
13.1. boot0 のスクリーンショット 419
13.2. boot2 のスクリーンショット 421
13.3. /etc/ttys の insecure コンソール 426
14.1. スーパーユーザ権限でプログラムをインストールする 432
14.2. FreeBSD におけるユーザの追加 433
14.3. rmuser による対話的なアカウントの削除 435
14.4. スーパユーザによる対話的な chpass 436
14.5. 通常のユーザによる対話的な chpass 436
14.6. 自分のパスワードの変更 437
14.7. スーパーユーザ権限での他のユーザのパスワード変更 437
14.8. pw(8) によるグループの追加 442
14.9. pw(8) により新しいグループにメンバを追加する 442
14.10. pw(8) によるグループへのユーザ追加 442
14.11. グループに所属しているユーザを調べるための id(1) の使い方 443
15.1. SSH を用いた SMTP 用の安全なトンネルの作成 497
16.1. ssh 越しの dump の利用 527
16.2. RSH を設定した ssh 越しの dump を利用 527
16.3. ブートフロッピーを作成するスクリプト 531
16.4. FreeBSD 4.X での vnconfig を用いた既存のファイルシステムイメージのマウント 536
16.5. vnconfig を用いたファイルベースディスクの新規作成 536
16.6. FreeBSD 5.X での mdconfig を用いた既存のファイルシステムイメージのマウント 537
16.7. mdconfig を用いたファイルシステムイメージの新規作成 537
16.8. mdmfs を用いたファイルベースディスクの設定とマウント 538
16.9. FreeBSD 4.X での md メモリディスク 539
16.10. mdconfig を用いたメモリベースディスクの新規作成 539
16.11. mdmfs を用いたメモリベースディスクの新規作成 540
19.1. 端末の項目を /etc/ttys に追加する 610
21.1. sendmail アクセスデータベースの設定 680
21.2. メールエイリアス 680
21.3. 仮想ドメインメール対応表の例 682
22.1. amd によるエクスポートされたファイルシステムのマウント 732
22.2. 支店または家庭のネットワーク 745
22.3. 本社 LAN や他の LAN 746

22.4. inetd への HangUP シグナル送付 797

前書き

想定している読者

最初の部分は FreeBSD を使い始めた人向けで、FreeBSD のインストールの過程を手引きし、UNIX® の基礎となっている概念や慣習を丁寧に紹介します。この部分に取り組むために必要なのは、探究心と、紹介された新たな概念を理解する能力だけです。

その次の、ハンドブックのはるかに大きな部分では、FreeBSD システム管理者が興味を抱くあらゆる種類の話題が分かりやすく言及されています。 一部の章は、その章の前に読んでおくべきことが推奨されており、各章の始めの概要で述べられています。

さらなる情報源の一覧は、付録B 参考図書 をご覧ください。

第 3 版からの変更

オンライン版のハンドブックは、FreeBSD ドキュメンテーションプロジェクトの献身的なメンバーによる 10 年以上に渡る作業の頂点に立つものです。2004 年に出版された 2 巻組の第 3 版からの主な変更は、次のようなものです。

- 強力なパフォーマンス解析ツール DTrace に関する情報が追加されました。
- ファイルシステム対応の章が追加されました。Sun™ の ZFS のような FreeBSD ネイティブではないファイルシステムへの対応について説明しています。
- セキュリティ監査の章が追加されました。FreeBSD における新しい監査のケイパビリティおよびその使用方法について説明しています。
- 仮想化の章が追加されました。os; を仮想化ソフトへインストールする方法などを取り上げています。
- 新しいインストールユーティリティの bsdinstall を用いた FreeBSD のインストール方法を説明する 2章FreeBSD のインストール (9.X 以降の場合) という章が追加されました。

第 2 版 (2004) からの変更

第 3 版は、FreeBSD ドキュメンテーションプロジェクトの献身的なメンバーによる 2 年以上に渡る作業の頂点に立つものです。 サイズが大きくなったため、印刷版は、2 巻での出版となりました。この新たな版における主な変更は、次のようなものです。

- 12章設定とチューニング に、ACPI 電源管理、`cron` システムユーティリティ、およびカーネルチューニングオプションに関するより多くの情報が追加されました。

- 15章セキュリティ に、Virtual Private Network (VPN)、ファイルシステムアクセスコントロールリスト (ACL)、およびセキュリティ勧告に関する情報が追加されました。

- Mandatory Access Control (MAC) の章がこの版で追加されました。MAC がどのようなもので、このメカニズムがどのように FreeBSD システムを安全にするかについて説明しています。

- 16章ストレージ に、USB ストレージデバイス、ファイルシステムスナップショット、ファイルシステムクォータ、ファイルおよびネットワークベースのファイルシステム、暗号化されたディスクパーティションに関する情報が追加されました。

- 20章PPP と SLIP に、トラブルシューティングの節が追加されました。

- 21章電子メール, に、他のメール転送エージェント、SMTP 認証、UUCP, fetchmail, procmail や他の高度な話題についての情報が追加されました。

- ネットワークサービスの章が、この版で新しく追加されました。　この章では、Apache HTTP サーバ、fptd および Samba を用いて Microsoft® Windows® クライアント用にサーバを設定する方法などを取り上げています。再構成によりいくつかの節が、22章高度なネットワーク から移動してきました。

- 22章高度なネットワーク に、FreeBSD での Bluetooth® デバイスの使用、ワイヤレスネットワークの設定、Asynchronous Transfer Mode (ATM) ネットワークに関する情報が追加されました。

- 本書で使われている専門用語の定義をまとめた用語集が追加されました。

- 本書を通じて表および図の表現において数多くの改良がおこなわれました。

第 1 版 (2001) からの変更

第　2　版は、FreeBSD　ドキュメンテーションプロジェクトの献身的なメンバーによる　2 年以上に渡る作業の頂点に立つものでした。この新たな版における主な変更は、次のようなものでした。

- 完備した索引が追加されました。

- ASCII キャラクタによる図はすべて画像に置き換えられました (訳注: 日本語版は作業中です)。

- 各章に、章に記載されている内容と、 読者に期待される予備知識がすぐに分かるように、一定の内容の概要が付け加えられました。

- 内容は、「始めに」、「システム管理」、「付録」の 3 つの論理的な部分に再構成されました。

- 3章FreeBSD 8.X のインストール は新規ユーザーが文章の内容を理解しやすいように多くのスクリーンショットを入れて完全に書き直されました。

- 4章UNIX の基礎知識 には、プロセス、デーモン、シグナルに関する情報が追加されました。

- 5章アプリケーションのインストール - packages と ports には、バイナリパッケージの管理に関する情報が追加されました。

- 6章X Window System は、XFree86™ 4.X 上で KDE や GNOME のような近代的なデスクトップテクノロジーを利用することに力点をおいて、完全に書き直されました。

- 13章FreeBSD の起動のプロセス が拡張されました。

- 16章ストレージ は、「ディスク」と「バックアップ」の 2 つの章に分かれていたものをまとめて書き直されました。私たちは、この話題は 1 つの章にまとめて示した方が分かりやすいと感じています。RAID (ハードウェアとソフトウェアの両方) に関する節も追加されました。

- 19章シリアル通信 は FreeBSD 4.X/5.X 向けに一から再構成されました。

- 20章PPP と SLIP は大幅に更新されました。

- 22章高度なネットワーク に、多くの新しい節が追加されました。

- 21章電子メール に、sendmail の設定についてより多くの情報が加えられました。

- 11章Linux バイナリ互換機能 には、Oracle® や Mathematica® のインストール情報が加えられました (訳注: 日本語版は作業中です)。

- この第 2 版では、以下の新たな話題が扱われています。

  - 12章設定とチューニング

  - 8章マルチメディア

この文書の構成

この文書は 5 部構成になっています。第 1 部導入では、FreeBSD のインストールと基本的な使い方を扱います。各章は順に読むことを想定していますが、馴染み深い話題を扱った章は飛ばしてもよいでしょう。第 2 部日々の生活では、FreeBSD で良く

使われる機能について説明します。この章とそれに続く章は、順不同に読むことができます。各章の始めにはその章が何を扱っていて、読者にどんな予備知識が期待されるかを簡潔に述べた概要がおかれています。第 3 部システム管理は、システム管理に関する話題を扱っています。第 4 部ネットワーク通信では、ネットワークおよびサーバに関する話題を扱っています。第 5 部は参考情報からなる付録です。

1章はじめに
: 新規ユーザに FreeBSD を紹介します。ここでは、FreeBSD プロジェクトの歴史、目標と開発モデルについて述べています。

2章FreeBSD のインストール (9.X 以降の場合)
: bsdinstall を用いた FreeBSD 9.*x* 以降のシステムのインストール過程を一通りユーザに案内しています。

3章FreeBSD 8.X のインストール
: sysinstall を用いた FreeBSD 8.*x* 以降のシステムの インストール過程を一通りユーザに案内しています。また、シリアルコンソール経由でのインストールのような高度な話題もいくらか扱っています。

4章UNIX の基礎知識
: FreeBSD オペレーティングシステムの基本的なコマンドや機能を扱っています。Linux® やその他の UNIX® 風のものに馴染んでいたら、この章を飛ばしても構わないでしょう。

5章アプリケーションのインストール - packages と ports
: FreeBSD の革新的な「Ports Collection」および標準的なバイナリパッケージによるサードパーティアプリケーションのインストールについて説明しています。

6章X Window System
: X Window System 全般と、特に FreeBSD 上での X11 の利用について述べています。また、KDE や GNOME のような一般的なデスクトップ環境にも触れています。

7章デスクトップアプリケーション
: Web ブラウザや生産性向上ツールのような一般的なデスクトップアプリケーションをいくつか挙げ、FreeBSD におけるインストール方法を説明しています。

8章マルチメディア
: システムを音声やビデオ再生に対応させるためにどう設定するかを説明します。 また、音声やビデオアプリケーションも例示しています。

9章FreeBSD カーネルのコンフィグレーション
: どのような場合に新たにカーネルを構成する必要があるかを説明し、カスタムカーネルのコンフィグレーション、構築、インストールについて詳しく説明しています。

10章プリンタの利用
: FreeBSD におけるプリンタの取り扱いを説明しています。たとえば、バナーページ、プリンターの課金、初期設定といったことです。

11章Linux バイナリ互換機能
FreeBSD の Linux® バイナリ互換機能を説明しています。また、 Oracle®, Mathematica® といった人気の高い Linux® アプリケーションのインストールを詳しく説明しています。

12章設定とチューニング
システム管理者が FreeBSD システムを調整して最適な性能を引き出すのに利用できるパラメータについて述べています。また、FreeBSD で利用されている様な設定ファイルとそのありかも解説しています。

13章FreeBSD の起動のプロセス
FreeBSD の起動プロセスを解説し、 このプロセスを設定オプションで制御する方法を説明しています。

15章セキュリティ
FreeBSD システムを安全に保つために役立つ Kerberos, IPsec および OpenSSH といった利用可能なさまざまなツールについて説明しています。

16章ストレージ
FreeBSD でストレージメディアやファイルシステムをどう扱うかを説明しています。対象は、物理ディスク、RAID アレイ、 光学およびテープメディア、メモリベースのディスク、ネットワークファイルシステムなどです。

17章地域化 (localization) - I18N/L10N の利用と設定
FreeBSD を英語以外の言語で使う方法を説明しています。 システムとアプリケーション両方のレベルの地域化を扱っています。

18章FreeBSD のアップデートとアップグレード
FreeBSD-STABLE, FreeBSD-CURRENT と FreeBSD のリリースの違いを説明します。 どんなユーザにとって開発システムを追随するのが有用かを述べ、 その方法の概要をまとめています。システムを最新のセキュリティリリースへアップデートする方法についても説明しています。

19章シリアル通信
FreeBSD システムに端末やモデムを、ダイヤルインまたはダイヤルアウト用に接続する方法を説明しています。

20章PPP と SLIP
FreeBSD で、PPP を使ってリモートシステムに接続する方法を説明しています。

21章電子メール
電子メールサーバの構成要素をそれぞれ説明し、 最もよく使われているメールサーバソフトウェアである sendmail について、単純な設定をとりあげています。

22章高度なネットワーク
LAN 上の他のコンピュータとインターネット接続の共有、 高度なルーティングに関するトピックス、ワイヤレスネットワーク、Bluetooth®, ATM, IPv6 等々、ネットワークに関するさまざまな話題を取り扱っています。

付録A FreeBSD の入手方法
FreeBSD を収録した CDROM や DVD の様々な入手先や、FreeBSD をダウンロードしてインストールできるインターネット上のサイトを挙げています。

付録B 参考図書
この文書は、もっと詳しい説明が欲しくなるかもしれないさまざまな題目について触れています。参考図書には、このハンドブックで参照している、多くの素晴らしい本が挙げられています。

付録C インターネット上のリソース
FreeBSD ユーザが FreeBSD について質問したり、技術的な議論に参加できる、多くの公開された場について説明しています。

付録D PGP 公開鍵
多くの FreeBSD 開発者の PGP fingerprint を載せています。

この文書で用いられている表記法

一貫して読みやすい文章を提供するために、この文書全体では以下の表記法が用いられています。

書体による表記

イタリック体
イタリック体 のフォントは、ファイル名、URL, 強調表現、技術用語の最初の使用を表すのに使われています。

##
##フォントは、エラーメッセージ、コマンド、環境変数、ports の名称、ホスト名、ユーザ名、 グループ名、デバイスの名称、変数、 コードの断片を表すのに使われています。

太字
太字のフォントは、アプリケーション、コマンド、キーを表すのに使われています。

ユーザー入力

文章の他の部分と区別するため、キーは太字で示されています。同時に押すことを意図したキーの組み合わせは、キーの間に `+' を入れて表されます。たとえば

Ctrl+Alt+Del

は、ユーザーが Ctrl, Alt それから Del キーを同時に押すことを意図しています。

順に押すことを意図したキーは、カンマで区切って表されます。たとえば

Ctrl+X, Ctrl+S

は、ユーザーが Ctrl キーと X キーを同時に押してから、Ctrl キーと S キーを同時に押すことを意図しています。

例

`C:\>` で始まる例は、MS-DOS® コマンドを表しています。特に注釈がなければ、それらのコマンドは最近の Microsoft® Windows® の「コマンドプロンプト」環境でも実行できます。

```
E:\> tools\fdimage floppies\kern.flp A:
```

`#` で始まる例は、FreeBSD 上でスーパーユーザ権限で実行しなければならないコマンドを示しています。そのコマンドを入力するには、`root` としてログインするか、通常のアカウントでログインして、スーパーユーザ権限を取得するために su(1) を使います。

```
# dd if=kern.flp of=/dev/fd0
```

`%` で始まる例は、通常のユーザアカウントで実行するべきコマンドを示しています。特に断りのない限り、環境変数の設定やその他のシェルコマンドには C シェルの文法が使われています。

```
% top
```

謝辞

あなたが手にしている文書は、 世界中の何百人もの人々の努力の賜物です。 誤字脱字の修正を送ったのか、文章を丸々投稿したのかによらず、 すべての貢献が役に立ちました。

多くの会社が、 著者らを雇用してフルタイムでこの文書に取り掛かれるようにしたり、 出版費用を出したりして、この文書を作り上げるのを援助してくれました。特に、BSDi (その後 Wind River Systems に買収されました) は、フルタイムでこの文書の改善作業をするように FreeBSD ドキュメンテーションプロジェクトのメンバーを雇用し、それが 2000 年 3 月の最初の出版 (ISBN 1-57176-241-8) につながりました。 その後、Wind River Systems は、印刷出力の仕組みを整備し、 章を追加するために著者を何名か追加で雇用してくれました。この作業は、 2001 年 11 月の第 2 版の出版 (ISBN 1-57176-303-1) に結実しました。2003-2004 年には、ハンドブック第 3 版の出版準備のために FreeBSD Mall, Inc が貢献者を雇用してくれました。

パート I. 導入

FreeBSD ハンドブックの第 1 部はユーザと FreeBSD が初めての管理者向けです。各章の内容は以下のとおりです。

- FreeBSD の紹介
- インストールの手順の解説
- UNIX® の基礎
- FreeBSD で利用できる豊富なサードパーティ製のアプリケーションの インストール方法
- UNIX® におけるウィンドウシステム X、およびクリエイティブなデスクトップ環境の設定の詳細の紹介

このハンドブックでは頻繁にページを飛すことなく前から後へと スムーズに読み進めるように、後方への参照を極力抑えるようにしています。

目次

1. はじめに 5
    1.1. この章では 5
    1.2. FreeBSD へようこそ! 5
    1.3. FreeBSD プロジェクトについて 9
2. FreeBSD のインストール (9.*X* 以降の場合) 15
    2.1. この章では 15
    2.2. ハードウェア要件 16
    2.3. インストール前に行う作業 17
    2.4. インストールの開始 24
    2.5. bsdinstall の紹介 29
    2.6. ネットワークからのインストール 33
    2.7. ディスク領域の割り当て 34
    2.8. インストール操作の確定 41
    2.9. インストール後の作業 44
    2.10. トラブルシューティング 64
    2.11. Live CD を使う 65
3. FreeBSD 8.*X* のインストール 67
    3.1. この章では 67
    3.2. ハードウェア要件 68
    3.3. インストール前に行う作業 69
    3.4. インストールの開始 78
    3.5. sysinstall(8) の紹介 85
    3.6. ディスク領域の割り当て 90
    3.7. 何をインストールするか選択する 106
    3.8. インストールメディアの選択 108
    3.9. インストール操作の確定 110
    3.10. インストール後の作業 111
    3.11. トラブルシューティング 143
    3.12. 高度なインストールガイド 148
    3.13. カスタムインストールメディアの準備 150
4. UNIX の基礎知識 157
    4.1. この章では 157
    4.2. 仮想コンソールと端末 157
    4.3. 許可属性 162
    4.4. ディレクトリ構造 165
    4.5. ディスク構成 168
    4.6. ファイルシステムのマウントとアンマウント 177
    4.7. プロセス 180
    4.8. デーモン、シグナルとプロセス終了 182
    4.9. シェル 185
    4.10. テキストエディタ 188
    4.11. デバイスとデバイスノード 189
    4.12. さらに詳しい情報を得るには... 189
5. アプリケーションのインストール - packages と ports 193

5.1. この章では 193
5.2. ソフトウェアのインストール 193
5.3. ソフトウェアの探し方 195
5.4. pkgng によるバイナリ package の管理 197
5.5. Ports Collection の利用 203
5.6. インストール後の作業 213
5.7. うまく動作しない ports に遭遇した場合には 214
6. X Window System 217
6.1. この章では 217
6.2. 用語の説明 217
6.3. X11 のインストール 219
6.4. X11 の設定 220
6.5. X11 でのフォントの使用 227
6.6. X ディスプレイマネージャ 232
6.7. デスクトップ環境 235

第1章 はじめに

再構成、部分的な改訂: Jim Mock.

1.1. この章では

FreeBSD に興味を持っていただきありがとうございます! この章では FreeBSD の歴史、目標、開発モデルなど、FreeBSD プロジェクトに関するさまざまな事柄を扱います。

この章に書かれている話題は、次のようなものです。

- FreeBSD とその他のオペレーティングシステムとの違い
- FreeBSD プロジェクトの歴史
- FreeBSD プロジェクトの目標
- FreeBSD オープンソース開発モデルの基本的な考え方
- そして、「FreeBSD」という名前の由来について

1.2. FreeBSD へようこそ!

FreeBSD は、4.4BSD-Lite から派生したオペレーティングシステムで、Intel (x86 および Itanium®), AMD64, Sun UltraSPARC® コンピュータに対応しています。他のアーキテクチャに対する移植も進行中です。FreeBSD の歴史や、現在のリリースについても読むことができます。プロジェクトへの何らかの貢献 (ソースコード、ハードウェア、資金の提供など) について興味があれば、FreeBSD への貢献をご覧ください。

1.2.1. FreeBSD で何ができるの?

FreeBSD には多くの注目すべき機能があります。例を挙げれば以下のようになります:

- 優先度を動的に調節する機能を備えることで アプリケーションとユーザとの間で円滑かつ公平な コンピュータ資源共有を実現し、特に高い負荷にも耐えることができる堅牢さを備えた プリエンプティブマルチタスキング。
- 多くの人々が 1 つの FreeBSD システムをさまざまな目的で同時に使うことを可能にする マルチユーザ機能。 これは例えば、プリンタやテープデバイスといったシステムの周辺機器が、 そのシステムを利用する全てのユーザだけでなく ネットワーク経由においても自然な形で共有され、さらに重要なシステム資源の使い過ぎを防ぐために 個々の資源に対する制限がユーザ単位、グループ単位で設定できる、というようなことを意味しています。

- SCTP や DHCP, NFS, NIS, PPP, SLIP, IPsec, IPv6 といった業界標準規格のサポートを含んだ堅固な TCP/IP ネットワーキング。これによって、FreeBSD マシンが商用サーバと同じように相互に運用でき、NFS (リモートファイルアクセス) や、電子メールサービスのような極めて重要な機能を提供します。また、WWW や FTP, ルーティング、ファイアウォール (セキュリティ) サービスを用いてインターネットと接続できます。

- アプリケーション (あるいはユーザ) がお互いに干渉できない ようにするメモリ保護機能。 アプリケーションがクラッシュしても、 どのような場合でも他のアプリケーションには影響を与えません。

- 業界標準である X Window システム (X11R7) は、普通の VGA カードやモニタでグラフィカルユーザインタフェース (GUI) を提供し、すべてのソースコードも一緒に提供されます。

- Linux や SCO, SVR4, BSDI, NetBSD 用に作られた多くのプログラムとの バイナリ互換性。

- 何千ものすぐに実行可能な アプリケーションが FreeBSD の ports や packages コレクションで利用可能です。 ここに用意されているものはネットを探し回る必要がありません

- インターネット上で入手可能な、移植が容易な 何千ものアプリケーションを追加できます。FreeBSD は最も評判のよい商用の UNIX® システムとソースコードレベルで互換性があります。 このため、ほとんどのアプリケーションは、 もしあったとしてもほんの少しの変更でコンパイルすることができます。

- デマンドページング仮想メモリ とそれに「付随の VM/buffer キャッシュ」の設計は、多くのメモリを要求するアプリケーションに対して 効率よくメモリを与えるようにする一方で、他のユーザに対しても対話的な応答を維持します。

- 複数の CPU を搭載したマシンにおける SMPのサポート。

- 完全な C や C++ の開発ツール。進んだ研究や開発のための多くの他の言語も ports や packages コレクションで提供されています。

- システム全体のソースコードが提供されているので、要求に合わせて環境を最大限に適合させることができます。 真のオープンシステムが利用できるのですから、 所有権のある解決方法に締めつけられ、ベンダのなすがままになる必要はありません。

- 膨大な量の オンラインドキュメント。

- もう書ききれません!

FreeBSD はカリフォルニア大学バークレイ校の Computer Systems Research Group (CSRG) による 4.4BSD-Lite リリースを基にしており、BSD システムの開発の優れた伝統を守り続けています。CSRG による素晴らしい活動に加えて、FreeBSD

プロジェクトは何千時間もの時間を注ぎ込んで、 実際の使用の場において最大の性能と信頼性を 発揮するためにシステムのチューニングをおこなっています。 多くの大企業が PC オペレーティングシステムの分野で 実現しようと奮闘しているそのような機能や性能、信頼性を FreeBSD は今すぐ提供できます!

あなたの思いつく限りのアプリケーションは、何でも FreeBSD で実行できます。ソフトウェア開発からファクトリオートメーション、 在庫制御から遠く離れた人工衛星のアンテナの方向調整まで; 商用 UNIX® 製品でできることは、FreeBSD でも十分にできるのです! また、FreeBSD は世界中の研究センターや大学によって開発される 文字通り何千もの高品質で、たいていはほとんど無料で利用できる アプリケーションによる恩恵を得ることができます。 商用のアプリケーションも提供されており、 日々増え続けています。

FreeBSD のソースコードは広く提供されているので、 システムも特別なアプリケーションやプロジェクトに合わせて、 いくらでもカスタマイズすることができます。 これは有名な商業ベンダから出ているほとんどのオペレーティング システムでは不可能なことです。以下に現在 FreeBSD を 使っている人々のアプリケーションの例をいくつか上げます:

- インターネットサービス: FreeBSD に組み込まれている 頑強な TCP/IP ネットワーキング機能は次のようなさまざまな インターネットサービスの理想的なプラットフォームになります:

  - FTP サーバ
  - World Wide Web サーバ (標準、もしくは安全な [SSL])
  - IPv4 および IPv6 ルーティング
  - ファイアウォールと NAT (「IP マスカレード」) ゲートウェイ
  - 電子メールサーバ
  - USENET ニュースおよび電子掲示板システム
  - さらにいろいろ…

  FreeBSD を利用すれば、小規模で安価な 386 クラスの PC でも気軽に導入することができますし、事業の成長に合わせてアップグレードした 4 つの Xeon プロセッサと RAID ストレージデバイスを備えたシステムでも、 全くそのまま使うことができるのです。

- 教育: あなたは、計算機科学または関連分野の工学を専攻する学生さんですか? オペレーティングシステムやコンピュータアーキテクチャ、 ネットワークについて学習するなら、実際に FreeBSD のソースコードを読んで、それがどのように動作するのかを学ぶのが一番よい方法です。また、無料で利用できる CAD や数学、グラフィックデザインのパッケージがいくつもあるので、コンピュータに関わる主要な目的が、他のことをすることにある方にも、大いに役立ちます。

- 研究: システム全体のソースコードが利用できるため、FreeBSD はオペレーティングシステムの研究だけでなく、 計算機科学の他の部門においても優れたプラットフォームです。自由に利用できる FreeBSD の特長は、オープンフォーラムで 議論される特別なライセンスの同意や制限について心配することなく、離れたグループでもアイディアや開発の共有による共同研究を可能にします。

- ネットワーキング: 新しいルータが必要? ネームサーバ (DNS) は? 内部のネットワークを人々から守るファイアウォールは? FreeBSD はすみに眠っている使われていない 386 や 486 の PC を簡単に 洗練されたパケットフィルタリング機能を持つ高級なルータに 変えることができます。

- X Window ワークステーション: 自由に利用できる X11 サーバを使うことによって、安価な X 端末として FreeBSD を使うこともできます。X 端末とは違って FreeBSD は多くのアプリケーションをローカルに走らせることもでき、 中心のサーバの負荷を軽減することも可能です。 FreeBSD は「ディスクレス」でもブート可能であり、 個々のワークステーションを安価で、容易に管理することさえ可能にします。

- ソフトウェア開発: 基本的な FreeBSD システムには、有名な GNU の C/C++ コンパイラやデバッガを含んだ完全な開発ツールがついてきます。

FreeBSD は、ソースとバイナリの両方とも、CD-ROM または anonymous FTP で入手可能です。詳しくは 付録A FreeBSD の入手方法 をご覧ください。

1.2.2. FreeBSD はどこに使われていますか?

FreeBSD は、以下に代表されるような世界最大クラスの IT 会社のデバイスおよび製品のプラットフォームとして利用されています。

- Apple
- Cisco
- Juniper
- NetApp

FreeBSD は、以下のサイトに代表されるような、インターネット上で最大クラスのサイトでも利用されています。

- Yahoo!
- Yandex
- Apache
- Rambler

- Sina
- Pair Networks
- Sony Japan
- Netcraft
- Netflix
- NetEase
- Weathernews
- TELEHOUSE America
- Experts Exchange

また、この他にもあります。

1.3. FreeBSD プロジェクトについて

以下の節では簡単な歴史やプロジェクトの目標、開発モデルなど、普段は表にでない話題を提供しています。

1.3.1. FreeBSD 小史

FreeBSD プロジェクトは 1993 年の始めに Unofficial 386BSD Patchkit の最後の 3 人のまとめ役によって、部分的に patchkit から派生する形で開始されました。ここでの 3 人のまとめ役というのは、Nate Williams, Rod Grimes と、Jordan Hubbard です。

このプロジェクトのもともとの目標は、patchkit という仕組みではもう十分に解決できなくなってしまった 386BSD の数多くの問題を修正するための、386BSD の暫定的なスナップショットを作成することでした。 こういった経緯を経ているので、 このプロジェクトの初期の頃の名前は 386BSD 0.5 や 386BSD 暫定版 (Interim) でした。

386BSD は、Bill Jolitz が (訳注: バークレイ Net/2 テープを基に) 作成したオペレーティングシステムです。当時の 386BSD は、ほぼ一年にわたって放っておかれていた (訳注: 作者がバグの報告を受けても何もしなかった) というひどい状況に苦しんでいました。作者の代わりに問題を修正し続けていた patchkit は日を追うごとに不快なまでに膨張してしまっていました。このような状況に対して、彼らは暫定的な「クリーンアップ」スナップショットを作成することで Bill を手助けしようと決めました。しかし、この計画は唐突に終了してしまいました。Bill Jolitz が、このプロジェクトに対する受け入れ支持を取り下げることを突然決意し、 なおかつこのプロジェクトの代わりに何をするのかを一切言明しなかったのです。

たとえ Bill が支持してくれないとしても、彼ら 3 人の目標には依然としてやる価値があると考えていたため、David Greenman が考案した名称「FreeBSD」をプロジェクトの名前に採用し、新たなスタートを切りました。この時点でのプロジェクトの初期目標は、すでにこのシステム (訳注: 386BSD + Patchkit) を使っていた利用者たちと相談して決められました。プロジェクトが実現に向けて軌道に乗ってきたことが明確になった時点で、Jordan は Walnut Creek CDROM 社に連絡してみました。CD-ROM を使って FreeBSD を配布することによって、インターネットに容易に接続できない多くの人々が FreeBSD を簡単に入手できるようになると考えたからです。Walnut Creek CDROM 社は FreeBSD を CD で配布するというアイデアを採用してくれたばかりか、作業するためのマシンと高速なインターネット回線をプロジェクトに提供してくれました。当時は海のものとも山のものともわからなかったこのプロジェクトに対して、Walnut Creek CDROM 社が信じられないほどの信頼を寄せてくれたおかげで、FreeBSD は短期間のうちにここまで大きく成長したのです。

CD-ROM による最初の配布 (そしてネットでの、ベータ版ではない最初の一般向け配布) は FreeBSD 1.0 で、1993 年 12 月に公開されました。これはカリフォルニア大学バークレイ校の 4.3BSD-Lite (「Net/2」) を基とし、386BSD や Free Software Foundation からも多くの部分を取り入れたものです。これは初めて公開したものとしては十分に成功しました。続けて 1994 年 5 月に FreeBSD 1.1 を公開し、非常に大きな成功を収めました。

この時期、 あまり予想していなかった嵐が遠くから接近してきていました。 バークレイ Net/2 テープの法的な位置づけについて、Novell 社とカリフォルニア大学バークレイ校との間の長期にわたる 法廷論争において和解が成立したのです。和解の内容は、Net/2 のかなりの部分が「権利つき (encumbered)」コードであり、それは Novell 社の所有物である、というバークレイ校側が譲歩したものでした。なお、Novell 社はこれらの権利を裁判が始まる少し前に AT&T 社から買収していました。和解における譲歩の見返りにバークレイ校が得たのは、 4.4BSD-Lite が最終的に発表された時点で、4.4BSD-Lite は権利つきではないと公式に宣言されること、そしてすべての既存の Net/2 の利用者が 4.4BSD-Lite の利用へと移行することが強く奨励されること、という Novell 社からの「ありがたき天からの恵み」でした (訳注: 4.4BSD-Lite はその後 Novell 社のチェックを受けてから公開された)。FreeBSD も Net/2 を利用していましたから、1994 年の 7 月の終わりまでに Net/2 ベースの FreeBSD の出荷を停止するように言われました。ただし、このときの合意によって、私たちは締め切りまでに一回だけ最後の公開をすることを許されました。そしてそれは FreeBSD 1.1.5.1 となりました。

それから FreeBSD プロジェクトは、まっさらでかなり不完全な 4.4BSD-Lite を基に、文字どおり一から再度作り直すという、 難しくて大変な作業の準備を始めました。「Lite」 バージョンは、部分的には本当に軽くて、中身がなかったのです。 起動し、 動作できるシステムを実際に作り上げるために必要となるプログラムコードのかなりの部分がバークレイ校の CSRG (訳注: BSDを作っているグループ) によって (いろいろな法的要求のせいで) 削除されてしまっていたということと、4.4BSD の Intel アーキテクチャ対応が元々かなり不完全であったということがその理由です。 この移行作業は結局 1994 年の 11 月までかかりました。そしてその時点で FreeBSD 2.0 をネットと

CD-ROM (12 月末ごろ) を通じて公開しました。これは、かなり粗削りなところが残っていたにもかかわらず、かなりの成功を収めました。そしてその後に、より信頼性が高く、そしてインストールが簡単になった FreeBSD 2.0.5 が 1995 年の 6 月に公開されました。

これ以降、FreeBSD の安定性、速さや機能は改善され、リリースが行われてきました。

長期的な開発プロジェクトは 10.X-CURRENT 開発ブランチ (トランク) で続けられ、10.X のスナップショットリリースは、開発の進行状況に応じて スナップショットサーバ より継続して入手できます。

1.3.2. FreeBSD プロジェクトの目標

##: Jordan Hubbard.

FreeBSD プロジェクトの目的は、いかなる用途にも使用でき、 何ら制限のないソフトウェアを供給することです。 私たちの多くは、 コード (そしてプロジェクト) に対してかなりの投資をしてきており、 これからも多少の無駄はあっても投資を続けて行くつもりです。ただ、 他の人達にも同じような負担をするように主張しているわけではありません。FreeBSD に興味を持っている一人の残らず全ての人々に、 目的を限定しないでコードを提供すること。これが、私たちの最初のそして最大の「任務」であると信じています。そうすれば、コードは可能な限り広く使われ、 最大の恩恵をもたらすことができるでしょう。これが、 私たちが熱烈に支持しているフリーソフトウェアの最も基本的な目的であると、私は信じています。

私たちのソースツリーに含まれるソースのうち、 GNU 一般公有使用許諾 (GPL) または GNU ライブラリ一般公有使用許諾 (LGPL) に従っているものについては、多少制限が課せられています。ただし、 ソースコードへのアクセスの保証という、 一般の制限とはいわば逆の制限 (訳注1) です。GPL ソフトウェアの商利用には、そのライセンスにある 複雑な側面が影響してくることがあります。 ですから私たちは、そうすることが合理的であると判断されたときには、より制限の少ない、BSD 著作権表示を採用しているソフトウェアを選択するようにしています。

(訳注1) GPL では、「ソースコードを実際に受け取るか、 あるいは、希望しさえすればそれを入手することが可能であること」を求めています。

1.3.3. FreeBSD の開発モデル

##: ## #.

FreeBSD の開発は非常に開かれた、柔軟性のあるプロセスです。 貢献者リスト を見ていただければわかるとおり、FreeBSD は文字通り世界中の何百という人々の努力によって開発されています。FreeBSD の開発環境は、この何百という開発者がインターネット経由で共同作業できるようになっているのです。 新しい開発者はいつでも大歓迎ですので、FreeBSD technical discussions メーリングリスト にメールを送ってください。FreeBSD announcements メーリングリスト もありますので、他の FreeBSD ユーザに自分のやっていることを宣伝したい時にはどうぞ使ってください。

あと、FreeBSD プロジェクトとその開発プロセスについて、どなたにも知っていていただきたいのは以下のようなことです。

SVN リポジトリ
: 長年にわたり FreeBSD のソースツリーは CVS (Concurrent Versions System) によってメンテナンスされてきました。CVS はソースコード管理用のフリーソフトウェアで、FreeBSD のリリースにも含まれています。2008 年 6 月、プロジェクトはソースコード管理のシステムを SVN (Subversion) に移行しました。ソースツリーの急速な増加や、これまでに蓄積された膨大な量の履歴によって、CVS の持つ技術的な限界が明かになってきたためです。ドキュメンテーションプロジェクトと Ports Collection リポジトリも、それぞれ 2012 年 5 月と 7 月に CVS から SVN へと移行しました。FreeBSD `src/` リポジトリを取得するための情報は ソースツリーの同期 の章を、FreeBSD Ports Collection を取得するための詳細については Ports Collection の利用 の章をご覧ください。

ソースツリー管理者
: コミッター (committers) は Subversion ツリーへの書き込み権限を持っている人、FreeBSD のソースに変更を加えることができる人です (リポジトリに変更を加えるには、ソースをコントロールする `commit` というコマンドを使うので、これらの人々は英語では「committers」と呼ばれます)。開発者にコードを送って見てもらうのに一番いい方法は send-pr(1) コマンドを使うことです。もし、何か問題があって `send-pr` が使えないなら FreeBSD committer's メーリングリスト にメールを送っていただいても構いません。

FreeBSD コアチーム
: FreeBSD コアチームは FreeBSD プロジェクトが会社だとすると取締役会にあたるものです。コアチームとして一番重要な役割は FreeBSD プロジェクトが全体としてよい方向に向かっていることを確認することです。責任感あふれる開発者を上記のソースツリー管理者として招くこと、また仕事上の都合などでコアチームをやめた人たちの後任を見つけることもコアチームの役割です。現在のコアチームは FreeBSD 開発者 (committer) の中から 2014 年 7 月に選挙によって選出されました。コアチームを選出するための選挙は、2 年ごとに行なわれています。

  コアチームのうち何人かは特定の担当分野を持っており、システムのうち一部に特に重点をおいて面倒を見ています。FreeBSD 開発者と担当分野の完全なリストはコントリビュータのリストをご覧ください。

注記

忘れてほしくないのは、コアチームのほとんどは FreeBSD に対してボランティアの立場であり、FreeBSD プロジェクトからは何ら金銭的な支援を受けていない、ということです。ですから、ここでの「責任」は「保証されたサポート」ではありません。そういう意味で、上記の「取締役会」という例えはあまりよ

> くないかもしれません。むしろ、FreeBSD のために人生を棒に振ってしまった人の集まりといった方が正しいかも…。

その他のコントリビュータ
: 最後になりますが、 もっとも重要で多数をしめる開発者はフィードバックやバグフィクスをどんどん送ってくれるユーザ自身です。 FreeBSD の開発に関わっていきたいという人は、議論の場である FreeBSD technical discussions メーリングリストに参加するとよいでしょう。FreeBSD 関連メーリングリストに関する詳細は、付録C インターネット上のリソース をご覧ください。

  FreeBSD への貢献者リスト は日に日に長くなっています。あなたも今日、何か送ることからはじめてみませんか?

  もちろん FreeBSD に貢献するには、コードを書くほかにもいろいろな方法があります。助けが求められている分野については、 FreeBSD プロジェクトのウェブサイトをご覧ください。

ひとことで言うと、FreeBSD の開発組織はゆるやかな同心円状になっています。ともすると中央集権的に見えがちなこの組織は、 FreeBSD のユーザがきちんと管理されたコードベースを 容易に追いかけられるようにデザインされているもので、貢献したいという人を締め出す意図は全くありません! 私たちの目標は安定したオペレーティングシステムと 簡単にインストールして使うことのできる アプリケーションを提供することです。この方法は、それを達成するために非常にうまくはたらきます。

これから FreeBSD の開発にたずさわろうという人に、 私たちが望むことはただ一つです。FreeBSD の成功を継続的なものにするために、現在の開発者と同じような情熱を持って接してください!

1.3.4. サードパーティ製プログラム

FreeBSD では基本配布セットに加え、移植されたソフトウェア集として数千の人気の高いプログラムを提供しています。 この文書を書いている時点で 24,000 以上の ports (移植ソフトウェア) が存在します。ports には http サーバから、ゲーム、言語、エディタまでありとあらゆるものが含まれています。ports はオリジナルソースに対する「差分」という形で表現されており、 Ports Collection 全体でも 500 MB 程度にしかなりません。ports をコンパイルするには、インストールしたいと思っているプログラムのディレクトリに移動し、`make install` とすると、あとはすべてシステムがやってくれます。どの ports もオリジナルの配布セットを動的に取ってくるので、ディスクは構築したいと思っている ports の分だけを準備しておけば十分です。ほとんどの ports は、すでにコンパイルされた状態で「package」として提供されており、ソースコードからコンパイルしたくない場合、これを使うと (`pkg install` というコマンドで) 簡単にインストールできます。package と ports に関する詳細は、5章アプリケーションのインストール - packages と ports をご覧ください。

1.3.5. ドキュメント

最近の FreeBSD では、システムの最初のセットアップ時に、インストーラ (sysinstall(8) または bsdinstall(8) のどちらでも) 上で、ドキュメントを `/usr/local/share/doc/freebsd` 以下にインストールすることを選択できます。システムのインストール後でも、「ports を用いたドキュメンテーションのアップデート」に記述されている package を使うことで、いつでもドキュメントをインストールできます。これらのローカルにインストールされたドキュメントは、HTML ブラウザを使って以下の URL から参照できます。

FreeBSD ハンドブック (英文オリジナル)
`/usr/local/share/doc/freebsd/handbook/index.html`

FreeBSD に関する FAQ (英文オリジナル)
`/usr/local/share/doc/freebsd/faq/index.html`

また、`http://www.FreeBSD.org/` にはマスタ (かなり頻繁に更新されます) がありますので、こちらも参照してください。

第2章 FreeBSD のインストール (9.X 以降の場合)

再構成、部分的に書き直し: Jim Mock.
sysinstall のウォークスルー、スクリーンショット、 およびその他原稿: Randy Pratt 、 Randy Pratt.
bsdinstall に向けた改訂: Gavin Atkinson 、Warren Block.

2.1. この章では

FreeBSD では、テキストベースの使いやすいインストールプログラムが用意されています。 FreeBSD 9.0-RELEASE 以降では bsdinstall、 FreeBSD 9.0-RELEASE より前のリリースでは sysinstall と呼ばれるインストールプログラムを使います。 この章では、bsdinstall の使い方について説明します。 sysinstall の利用については、 3章FreeBSD 8.X のインストール で説明しています。

この章を読むと、以下のことがわかります。

- FreeBSD インストールメディアの作り方。
- FreeBSD がハードディスクをどのように分割し、参照するか。
- bsdinstall をどのように起動するか。
- bsdinstall が聞いてくる質問がどのような意味であり、 またどのように答えれば良いか。

この章を読む前に、以下のことを確認して下さい。

- インストールしようとしているバージョンに付属しているサポートハードウェア一覧を読み、あなたの使っているハードウェアがサポートされているかどうかを確認して下さい。

注記

このインストールの説明書は i386™ (「PC 互換」) アーキテクチャのコンピュータを対象にしています。他のプラットフォームに特有の説明についてはそのつど明記しています。 インストーラとこの文書で記述している内容の間には、 いくらかズレがあることがあります。この章を正確で忠実な手順書としてではなく、一般的なガイドとしてご利用ください。

2.2. ハードウェア要件

2.2.1. 最小構成

FreeBSD をインストールする最小構成は、 FreeBSD のバージョンやハードウェアのアーキテクチャによって異なります。

以下の節では、最小構成についての情報をまとめています。 また、FreeBSD のインストール方法によって、対応している CDROM ドライブや、場合によってはネットワークアダプタが必要となります。これに関しては「インストールメディアの準備」で説明します。

2.2.1.1. FreeBSD/i386

FreeBSD/i386 アーキテクチャは、486 以上のプロセッサと、少なくとも 64 MB の RAM が必要です。最小のインストールでは、少なくとも 1.1 MB のハードディスクの空き容量が必要です。

注記

古いコンピュータでは、 速いプロセッサを入手するより、より多くの RAM やより多くの容量のハードドライブを用意するほうが重要です。

2.2.1.2. FreeBSD/amd64

FreeBSD/amd64 を実行可能なプロセッサには 2 つのクラスがあります。1 つ目のクラスは、AMD Athlon™64, AMD Athlon™64-FX, AMD Opteron™ やそれより新しいプロセッサです。

もう 1 つのクラスは、Intel® EM64T アーキテクチャを採用しているプロセッサです。これらのプロセッサの例は、Intel® Core™ 2 Duo, Quad, Extreme プロセッサファミリ、Intel® Xeon™ 3000, 5000, 7000 系のプロセッサ、および Intel® Core™ i3, i5, i7 プロセッサです。

nVidia nForce3 Pro-150 ベースのコンピュータを使用するには、BIOS のセットアップを使って、IO APIC を無効にする 必要があります。もし、このようなオプションがなければ、代わりに ACPI を無効にする必要があるでしょう。Pro-150 チップセットには回避策が見つかっていないバグがあります。

2.2.1.3. FreeBSD/powerpc Apple® Macintosh®

すべての USB 内蔵の New World Apple® Macintosh® システムに対応しています。複数の CPU を持つコンピュータは SMP に対応しています。

32-bit カーネルは、RAM の最初の 2 GB だけを利用できます。 Blue & White PowerMac G3 では、FireWire® に対応していません。

2.2.1.4. FreeBSD/sparc64

FreeBSD/sparc64 が対応しているハードウェアの一覧については、 FreeBSD/sparc64 プロジェクトをご覧ください。

現時点では、 他のオペレーティングシステムとディスクの共有ができないので、FreeBSD/sparc64 専用のディスクが必要です。

2.2.2. サポートされているハードウェア

FreeBSD がサポートしているハードウェアアーキテクチャやデバイスの一覧は、 各リリースのハードウェアノートとして提供されます。 このドキュメントは通常リリースメディアのルートディレクトリにある `HARDWARE.TXT` という名前のファイルで見ることができます。ハードウェアリストは、FreeBSD のウェブサイトの リリース情報 のページにもあります。

2.3. インストール前に行う作業

2.3.1. データのバックアップ

FreeBSD をインストールするコンピュータに残されている価値のあるデータをすべてバックアップしてください。そして、インストール作業を進める前にバックアップが正しく取れていることを確認してください。 FreeBSD のインストーラは、ハードディスクに変更を加える前に確認を求めますが、一度実際に書き込む作業が始まってしまうと、もう元に戻すことはできません。

2.3.2. FreeBSD をインストールする場所の決定

インストールするオペレーティングシステムが FreeBSD のみで、ハードディスクすべてを使ってインストールする場合には、この節の後半を飛ばすことができます。しかし、ハードディスクに FreeBSD と他のオペレーティングシステムを共存させる必要がある場合には、ディスクレイアウトに関する基本的な部分を理解しておく必要があります。

2.3.2.1. FreeBSD/i386 および FreeBSD/amd64 アーキテクチャでのディスクレイアウト

ハードディスクを複数の塊に分割することができます。これらの塊は パーティション と呼ばれます。

ディスクをパーティションに分割する方法は 2 通りあります。 伝統的な Master Boot Record (MBR) では、ディスク 1 台あたり最大 4 つの プライマリパーティション をパーティションテーブルに持つことができます (歴史的な理由により、FreeBSD は、これらのパーティションのことを スライス と呼びます)。 大きなディスクにとって、作成できるパー

ティションの数が 4 つという制限は限定的なので、プライマリパーティションの 1 つに 拡張パーティション を作成できます。論理パーティション と呼ばれる特別のパーティションは、この拡張パーティションの内部に作成できます。これは、多少扱いにくいものです。

GUID Partition Table (GPT) は、ディスクをパーティションに分ける簡単で新しい方法です。伝統的な MBR パーティションテーブルと比べると、GPT ははるかに万能です。一般的な GPT の実装では、1 つのディスクに 128 個までのパーティションの作成が可能です。扱いにくい論理パーティションのような回避策は必要ありません。

警告

Windows® XP のような古いオペレーティングシステムは、GPT パーティションと互換性がありません。FreeBSD をこのようなオペレーティングシステムとディスク上で共存させる場合には、 MBR パーティションテーブルを使う必要があります。

FreeBSD の標準のブートローダは、プライマリまたは GPT パーティションに対応しています (FreeBSD の起動プロセスのより詳しい情報については、13章FreeBSD の起動のプロセス をご覧ください)。ディスク上のすべてのプライマリ、もしくは GPT パーティションが使われているのであれば、そのひとつを FreeBSD のために開放してください。

FreeBSD の最小のインストールでは、1 GB 程度のディスク容量が必要です。しかし、これは 非常に 小さなインストールであり、自分のファイルを作成するスペースはほとんど残らないでしょう。現実的には、グラフィカルな環境が必要なければ 3 GB、グラフィカルユーザインタフェースを使用するのであれば 5 GB 以上は必要でしょう。サードパーティ製ソフトウェアをインストールするならば、さらに多くのスペースが必要になります。

さまざまな パーティションサイズを変更するフリーや商用のツール を利用できます。GParted Live は、GParted パーティションエディタを含む完全なライブ CD です。多くの Linux Live CD ディストリビューションでも GParted を利用できます。

警告

ディスクパーティションのアプリケーションは、 ディスク上のデータを壊す可能性があります。 ディスクのパーティションを変更する前に、必ず全体のバックアップをとり、完全性を検証してください。

Microsoft® Vista のパーティションサイズの変更は、難しい可能性があります。このような作業を行う際は、Vista のインストール CDROM を手元に置いておくことをお勧めします。

例2.1 既存のパーティションを使用

既に Windows® がインストールされている 40 GB のハードディスクが 1 台接続されており、そのハードディスクは、20 GB の 2 つのパーティションに分割されていると仮定します。Windows® では、それぞれ `C:` および `D:` と呼びます。`C:` パーティションには 10 GB のデータ、`D:` パーティションには 5 GB のデータがあるとします。

`D:` にあるデータをすべて `C:` にコピーすれば、2 つ目のパーティションを解放し、FreeBSD のために使うことができるようになります。

例2.2 既存のパーティションを縮小する

Windows® がインストールされている 40 GB のハードディスクが 1 台接続された PC を使用していると仮定します。ディスクのすべてを 1 つの大きなパーティションとして使用しています。Windows® では、この 40 GB のパーティションを 1 つの `C:` ドライブとして表示します。15 GB のデータがあるとします。Windows® が 20 GB パーティション、残りの 20 GB パーティションを FreeBSD で使いたいとします。

インストールを行うには、2 つの方法があります。

1. Windows® のデータをバックアップし、インストール時に 20 GB のパーティションを作成して再インストールする。
2. 先に述べた GParted をはじめとするパーティションを縮小するツールを使って、Windows® のパーティションを縮小し、フリーの空間上に、FreeBSD のためのパーティションを作成する。

異なるオペレーティングシステムを含むディスクパーティションでは、どのオペレーティングシステムも同時に実行できるのは 1 つです。複数のオペレーティングシステムを同時に実行する方法については、 Virtualization の章で説明します。

2.3.3. ネットワークの詳細をまとめる

FreeBSD のインストール方法によっては、ネットワークに接続し、ファイルのダウンロードが必要です。イーサネットに接続 (または、ケーブル / DSL モデム経由でイーサネットイ

ンタフェースを利用して接続) するためには、インストール中にこれらの情報を入力する必要があります。

DHCP は、 自動的にネットワークの設定情報を得るのによく使われます。 DHCP を利用できない場合には、 システム管理者かプロバイダにネットワーク情報を問い合わせる必要があります。

1. IP アドレス

2. サブネットマスク

3. デフォルトルータの IP アドレス

4. ローカルネットワークのドメイン名

5. DNS サーバの IP アドレス

2.3.4. FreeBSD Errata の確認

FreeBSD プロジェクトでは FreeBSD の各リリースができる限り安定するよう努力していますが、時々バグが発生してしまうことがあります。極まれに、発生したバグがインストールプロセスに影響を与えることがあります。 これらの問題は発見され解決されるとともに、FreeBSD のウェブサイトの FreeBSD Errata に掲示されます。インストールに関して注意すべき既知の問題が無いことを確かめるために、インストールする前に Errata を確認してください。

すべてのリリースに関する情報や Errata は、FreeBSD のウェブサイト の リリース情報 の項で確認することができます。

2.3.5. インストールメディアの準備

FreeBSD のインストールは、インストール用の CD, DVD または USB メモリスティックとともに、コンピュータを起動するところから始まります。インストーラは、オペレーティングシステムで実行できるようなプログラムではありません。

FreeBSD のインストールファイルをすべて含んでいる通常のインストールメディアに加え、 bootonly を利用できます。 bootonly インストールメディアは、インストールファイルを含んでいません。 そのかわり、インストールの途中ですべてのファイルをネットワークからダウンロードします。 そのため、bootonly インストール CD は小さく、また、インストール中に必要なファイルだけをダウンロードするので、使用するバンド幅を減らすことが出来ます。

FreeBSD のインストールメディアは FreeBSD ウェブサイト から入手できます。またイメージファイルと同じディレクトリから `CHECKSUM.SHA256` をダウンロードし、チェックサム を計算してイメージファイルの完全性を確認してください。 FreeBSD では sha256(1) を用いてチェックサムを計算できますが、 他のオペレーティングシステムも同様のプログラムを持っています。 計算したチェックサムと `CHECKSUM.SHA256`

に示されている値を比較してください。チェックサムは完全に一致している必要があります。もしチェックサムが一致していなければ、ファイルは壊れているので削除してください。

ヒント

FreeBSD の入っている CDROM や DVD, USB メモリスティックを持っているのであれば、この章を飛ばしてもかまいません。

FreeBSD の CD および DVD イメージは、起動可能な ISO ファイルです。インストールには、一枚の CD または DVD があれば十分です。現在使用しているオペレーティングシステムにインストールされている、ISO イメージを CD に書き込むアプリケーションを用いて、起動可能な CD または DVD を作成してください。FreeBSD では、Ports Collection の `sysutils/cdrtools` の cdrecord(1) を使って書き込むことができます。

起動可能なメモリスティックを作成する場合には、以下の手順にしたがってください。

1. メモリスティックのイメージの取得

   FreeBSD 9.0-RELEASE 以降のメモリスティックのイメージは、`ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/`*`arch`*`/`*`arch`*`/ISO-IMAGES/`*`version`*`/FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-memstick.img` の `ISO-IMAGES/` ディレクトリからダウンロードできます。ここで、*`arch`* と *`version`* の部分を、それぞれインストールするアーキテクチャとバージョン番号に置き換えてください。たとえば、FreeBSD/i386 9.0-RELEASE のメモリスティックのイメージは、`ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/i386/ISO-IMAGES/9.0/FreeBSD-9.0-RELEASE-i386-memstick.img` から入手できます。

   #### ヒント

   FreeBSD 8.*X* 以前では、これとは異なるディレクトリパスが使われています。FreeBSD 8.*X* 以前のバージョンのダウンロードとインストールの詳細については、3章FreeBSD 8.*X* のインストール で説明しています。

   メモリスティックイメージには、`.img` という拡張子がついています。`ISO-IMAGES/` ディレクトリには複数の異なるイメージがあり、インストールする

FreeBSD のバージョンや、インストールするハードウェアによって、必要なファイルが変わります。

重要

以下の作業によってデータが 消去 されるので、先に進む前に、使用する USB スティックにあるデータをバックアップしてください。

2. イメージファイルをメモリスティックに書き込む

手順2.1 FreeBSD を使ってイメージを書き込む

警告

以下の例では、イメージを書き込むデバイスを `/dev/da0` としています。正しいデバイスを出力先に設定していることを十分確認してくだい。 さもなければ、現在あるデータを破壊してしまうことになります。

- dd(1) を使ってイメージを書き込む

  `.img` ファイルは、通常のファイルではありません。メモリスティックの完全な内容の イメージ です。通常のファイルのようなコピーは できません。dd(1) を使用して直接ターゲットディスクに書き込む必要があります。

```
# dd if=FreeBSD-9.0-RELEASE-i386-memstick.img of=/
dev/da0 bs=64k
```

手順2.2 Windows® を使ってイメージを書き込む

警告

適切なドライブレターを出力先に設定していることを十分確認してくだい。さもなければ、現在あるデータを破壊してしまうでしょう。

1. Image Writer for Windows® を入手する

   Image Writer for Windows® は、イメージファイルをメモリスティックに正しく書き込むことのできるフリーのアプリケーションです。 `https://launchpad.net/win32-image-writer/` からダウンロードして、フォルダに展開してください。

2. イメージライタを使ってイメージを書き込む

   Win32DiskImager アイコンをダブルクリックして、プログラムを起動します。`Device` の下に表示されるデバイスレターが、メモリスティックのドライブであることを確認してください。フォルダのアイコンをクリックして、メモリスティックに書き込むイメージファイルを選択します。[Save] をクリックして、イメージファイルの名前をアクセプトしてください。すべてが正しく行われたかどうか、また、他のウィンドウでメモリスティックのフォルダが開かれていないことを確認してください。準備ができたら、[Write] を押して、メモリスティックにイメージファイルを書き込みます。

注記

フロッピーディスクからのインストールには、もはや対応していません。

これで FreeBSD をインストールする用意ができました。

2.4. インストールの開始

重要

デフォルトでは、次のメッセージが表示されるまで　インストーラはディスクに何の変更も加えません。

```
Your changes will now be written to disk.  If you
have chosen to overwrite existing data, it will
be PERMANENTLY ERASED. Are you sure you want to
commit your changes?
```

この警告の前であれば、 いつでもハードディスクの内容を変更することなくインストールを中断できます。 もし、何かを間違って設定してしまったことが心配ならば、 最後の警告の前に単にコンピュータをオフにしてください。被害を残さずに済みます。

2.4.1. 起動

2.4.1.1. i386™ および amd64 アーキテクチャでの起動

1. 「インストールメディアの準備」で説明されている「起動」USB スティックを使用する場合には、コンピュータを立ち上げる前に、USB スティックを挿入してください。

   CDROM から起動する場合には、コンピュータを立ち上げ、すぐに CDROM を挿入してください。

2. CDROM または USB から起動するようにコンピュータを設定してください。この作業は、インストールに用いるメディアによって異なります。起動するデバイスを BIOS で設定できます。ほとんどのシステムでは、通常、起動時に F10, F11, F12 または Escape キーを押すことで、起動するデバイスを選択できます。

3. もし、コンピュータがすでに存在している OS を読み込み、通常通り起動してしまったのであれば、以下の原因が考えられます。

   1. 起動ディスクが起動プロセスにおいて十分早いタイミングで挿入されていません。ディスクをそのままにしてコンピュータを再起動してください。

   2. 先程の BIOS の変更が適切に行われていません。正しいオプションを設定してやり直してください。

   3. 使用している　BIOS　は、 希望しているメディアからの起動に対応していません。Plop Boot Manager を使うと、古いコンピュータを CD や USB メディアから起動できます。

4. FreeBSD は起動を開始します。CDROM から起動している場合、次のような画面が表示されるでしょう (バージョン情報は省略しています)。

```
Booting from CD-ROM...
645MB medium detected
CD Loader 1.2

Building the boot loader arguments
Looking up /BOOT/LOADER... Found
Relocating the loader and the BTX
Starting the BTX loader

BTX loader 1.00 BTX version is 1.02
Consoles: internal video/keyboard
BIOS CD is cd0
BIOS drive C: is disk0
BIOS drive D: is disk1
BIOS 636kB/261056kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/boot/kernel/kernel text=0x64daa0 data=0xa4e80+0xa9e40 ↺
syms=[0x4+0x6cac0+0x4+0x88e9d]
\
```

5. FreeBSD ブートローダにより、以下の画面が表示されます。

図2.1 FreeBSD ブートローダメニュー

10 秒間待つか、Enter を押してください。

2.4.1.2. Macintosh® PowerPC® での起動

ほとんどのコンピュータでは、起動中にキーボードの C を押しておくと、CD から起動します。別の方法では Command+Option+O+F、または non-Apple® キーボードでは Windows+Alt+O+F を押してください。`0 >` プロンプトで

```
boot cd:,\ppc\loader cd:0
```

と入力してください。

キーボードのない Xserves では、Open Firmware に起動する方法について Apple®'s support web site を参照してください。

2.4.1.3. SPARC64® アーキテクチャでの起動

ほとんどの SPARC64® システムは、ディスクから自動的に起動するように設定されています。FreeBSD をインストールするには、ネットワークまたは CDROM から起動する必要があり、そのためには PROM (OpenFirmware) に入る必要があります。

PROM に入るにはシステムを再起動し、ブートメッセージが表示されるまで待ってください。モデルによりますが、以下のような表示です。

```
Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
Copyright 1998-2001 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.2, 128 MB memory installed, Serial #51090132.
Ethernet address 0:3:ba:b:92:d4, Host ID: 830b92d4.
```

もしシステムがこの時点でディスクから起動するようでしたら、キーボードから L1+A または Stop+A を押すか、シリアルコンソールから **BREAK** (たとえば、tip(1) または cu(1) では **~#**) を送信して、PROM プロンプトを表示してください。次のような表示です。

```
ok    ❶
ok {0} ❷
```

❶ 単一の CPU を持つシステムでのプロンプト
❷ SMP システムのプロンプト。数字はアクティブな CPU の数。

ここで、CDROM をドライブに挿入し、PROM プロンプトで `boot cdrom` と入力してください。

2.4.2. デバイス検出結果を再表示する

画面に表示される数百行の文字列は (バッファに) 記憶されており、再表示することが出来ます。

バッファを再表示するには、Scroll Lock キーを押します。これで画面をスクロールできます。結果を見るためには、矢印キーもしくは PageUp か PageDown を使います。Scroll Lock をもう一度押すと、スクロールを停止します。

直ちにこの作業を行って、カーネルがデバイス検出を実行している時に 画面の外に流れた文字列を再表示してください。 図2.2「典型的なデバイス検出結果の例」 によく似た画面が現われるでしょう。ただし、あなたのコンピュータに搭載されているデバイスによって、表示される文字列は異なります。

```
Copyright (c) 1992-2011 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, ↺
1994
        The Regents of the University of California. All rights ↺
reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 9.0-RELEASE #0 r225473M: Sun Sep 11 16:07:30 BST 2011
    root@psi:/usr/obj/usr/src/sys/GENERIC amd64
CPU: Intel(R) Core(TM)2 Duo CPU     T9400  @ 2.53GHz (2527.05-MHz K8-
class CPU)
  Origin = "GenuineIntel"  Id = 0x10676  Family = 6  Model = 17  ↺
Stepping = 6
  ↺
Features=0xbfebfbff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE
  ↺
Features2=0x8e3fd<SSE3,DTES64,MON,DS_CPL,VMX,SMX,EST,TM2,SSSE3,CX16,xTPR,PDCM,SSE4.1>
  AMD Features=0x20100800<SYSCALL,NX,LM>
  AMD Features2=0x1<LAHF>
  TSC: P-state invariant, performance statistics
real memory  = 3221225472 (3072 MB)
avail memory = 2926649344 (2791 MB)
Event timer "LAPIC" quality 400
ACPI APIC Table: <TOSHIB A0064   >
FreeBSD/SMP: Multiprocessor System Detected: 2 CPUs
FreeBSD/SMP: 1 package(s) x 2 core(s)
 cpu0 (BSP): APIC ID:  0
 cpu1 (AP): APIC ID:  1
ioapic0: Changing APIC ID to 1
ioapic0 <Version 2.0> irqs 0-23 on motherboard
kbd1 at kbdmux0
acpi0: <TOSHIB A0064> on motherboard
acpi0: Power Button (fixed)
acpi0: reservation of 0, a0000 (3) failed
acpi0: reservation of 100000, b6690000 (3) failed
Timecounter "ACPI-safe" frequency 3579545 Hz quality 850
acpi_timer0: <24-bit timer at 3.579545MHz> port 0xd808-0xd80b on ↺
acpi0
cpu0: <ACPI CPU> on acpi0
ACPI Warning: Incorrect checksum in table [ASF!] - 0xFE, should be ↺
0x9A (20110527/tbutils-282)
cpu1: <ACPI CPU> on acpi0
pcib0: <ACPI Host-PCI bridge> port 0xcf8-0xcff on acpi0
pci0: <ACPI PCI bus> on pcib0
vgapci0: <VGA-compatible display> port 0xcff8-0xcfff mem ↺
0xff400000-0xff7fffff,0xe0000000-0xefffffff irq 16 at device 2.0 on ↺
pci0
agp0: <Intel GM45 SVGA controller> on vgapci0
agp0: aperture size is 256M, detected 131068k stolen memory
vgapci1: <VGA-compatible display> mem 0xffc00000-0xffcfffff at ↺
device 2.1 on pci0
pci0: <simple comms> at device 3.0 (no driver attached)
em0: <Intel(R) PRO/1000 Network Connection 7.2.3> port ↺
0xcf80-0xcf9f,mem 0xff9c0000-0xff9dffff,0xff9fe000-0xff9fefff irq ↺
20 at device 25.0 on pci0
em0: Using an MSI interrupt
em0: Ethernet address: 00:1c:7e:6a:ca:b0
uhci0: <Intel 82801I (ICH9) USB controller> port 0xcf60-0xcf7f irq ↺
16 at device 26.0 on pci0
usbus0: <Intel 82801I (ICH9) USB controller> on uhci0
uhci1: <Intel 82801I (ICH9) USB controller> port 0xcf40-0xcf5f irq ↺
21 at device 26.1 on pci0
usbus1: <Intel 82801I (ICH9) USB controller> on uhci1
```

図2.2 典型的なブート検出結果の例

丹念に検出結果をチェックして、あなたが予期したすべてのデバイスを FreeBSD が検出できた事を確認してください。デバイスが見つからなかった場合、 検出結果へ表示されません。カーネルモジュール を構築することで、GENERIC カーネルに含まれていないデバイスのサポートを追加できます。

デバイス検出後、 図2.3「インストールメディアの選択モード」 が表示されます。 インストールメディアは、FreeBSD のインストール、live CD、または FreeBSD シェルへのアクセスの 3 つの目的に利用出来ます。矢印キーを使ってオプションを選択し、Enter を押してください。

図2.3 インストールメディアの選択モード

ここで [Install] を選択すると、インストール作業が始まります。

2.5. bsdinstall の紹介

bsdinstall は、Nathan Whitehorn <nwhitehorn@FreeBSD.org > により書かれた、テキストベースのインストールプログラムです。2011 年の FreeBSD 9.0 で導入されました。

注記

Kris Moore <kmoore@FreeBSD.org > の pc-sysinstall は、PC-BSD に含まれており、FreeBSD のインストール にも使用できます。bsdinstall と混同されることもありますが、この 2 つのインストーラには関係はありません。

bsdinstall のメニューは、カーソルキー、Enter, Tab, Space 等で操作します。

2.5.1. キー配列メニューの選択

使用しているシステムのコンソールにもよりますが、bsdinstall は、最初にデフォルト以外のキーボードレイアウト使うかどうかを尋ねてきます。

図2.4 キーマップの選択

[YES] を選択すると、以下のキーボード選択画面が表示されます。そうでなければ、この選択メニューは表示されず、デフォルトのキーボードマップが使われます。

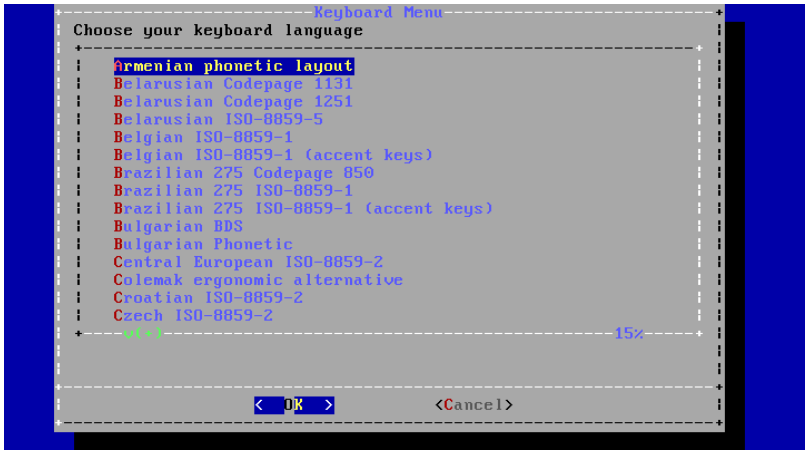


図2.5 キーボードメニューの選択

システムのキーボードに最も近いキーマップを上下のカーソルキーで選び、 Enter キーを押してください。

注記

Esc を押すと、 デフォルトのキーボードマップを使うようになります。 どのキーボードマップを選べばよいかわからない場合には、United States of America ISO-8859-1 キーマップを選ぶとよいでしょう。

2.5.2. ホスト名の設定

次に bsdinstall は、新しくインストールするシステムに与えるホスト名の入力に移ります。

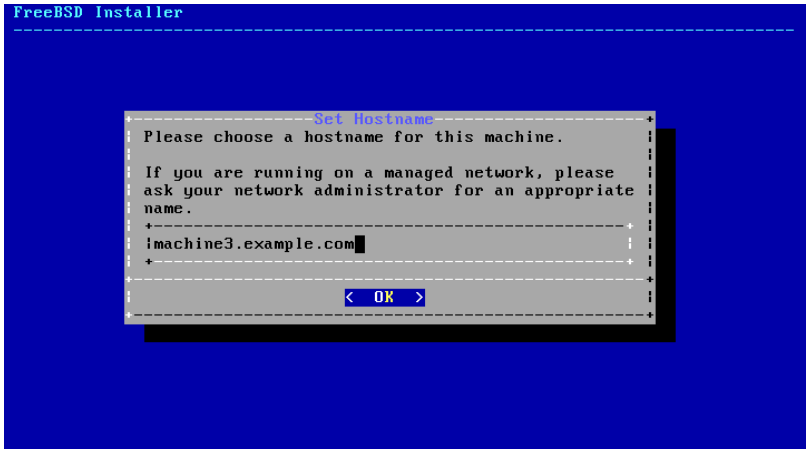


図2.6 ホスト名の設定

入力するホスト名は、`machine3.example.com` のように完全修飾のホスト名で入力してください。

2.5.3. インストールするコンポーネントの設定

次に、bsdinstall は、インストールするオプションのコンポーネントの選択に移ります。

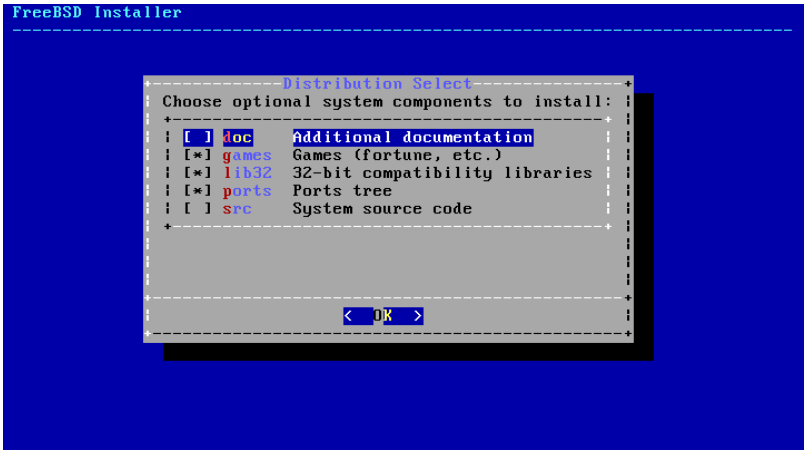


図2.7 インストールするコンポーネントの設定

どのコンポーネントをインストールするかは、 システムの用途と用意されているディスク容量に依存します。FreeBSD カーネルとユーザランド (総称 は「base system」) は、常にインストールされます。

インストールのタイプによっては表示されないコンポーネントもあります。

- `doc` - 追加の文書。多くは歴史的な興味のものです。FreeBSD ドキュメンテーションプロジェクトが提供している文書は、あとでインストールされます。
- `games` - fortune, rot13 などの伝統的な BSD ゲームをインストールします。
- `lib32` - 32-bit のアプリケーションを 64-bit 版の FreeBSD で実行する際に必要となる互換ライブラリ。
- `ports` - FreeBSD Ports Collection

  Ports Collection は、 ソフトウェアのインストールを簡単に行う方法を提供します。Ports Collection は、ソフトウェアをコンパイルするのに必要なソースコードを含んでいません。サードパーティ製ソフトウェアパッケージのダウンロード、コンパイル、インストールを自動化するように設計されたファイルの集まりです。Ports Collection の使い方については、5章アプリケーションのインストール - packages と ports で説明します。

警告

インストールプログラムは、 システムのディスクに十分な空き容量があるかどうかを確認しないので、このオプションを選択する際には、 ハードディスクの容量が十分あることを確認してくださ

> い。FreeBSD 9.0 では、Ports Collection が必要とする容量は、約 500 MB です。より最新の FreeBSD のバージョンでは、安全のためにもう少し大きなサイズを想定してください。

- `src` - システムのソースコード

  FreeBSD には、カーネルとユーザランド両方のソースコードが付属しています。ほとんどのアプリケーションは必要としませんが、ソースで提供されているソフトウェア (デバイスドライバやカーネルモジュールなど) によってはコンパイル時、または FreeBSD そのものを開発する場合に必要となります。

  すべてのソースツリーをインストールするには 1 GB のディスク容量を必要とします。また、FreeBSD システム全体のコンパイルには、さらに 5 GB の容量が必要です。

2.6. ネットワークからのインストール

bootonly インストールメディアは、インストールファイルを含んでいません。bootonly インストールメディアを使う場合には、 ネットワーク経由でファイルをダウンロードする必要があります。

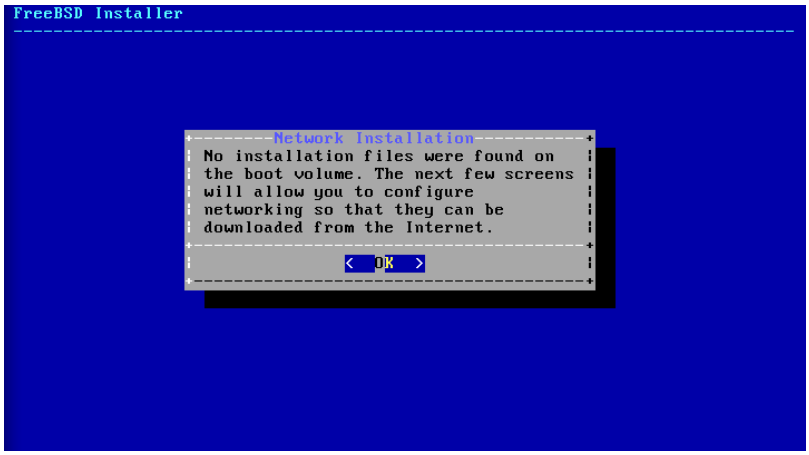


図2.8 ネットワークからのインストール

「ネットワークインタフェースの設定」 に示されているように、 ネットワーク接続の設定が終わっていれば、ミラーサイトが選択されます。 ミラーサイトは、FreeBSD ファイルのコピーをキャッシュしています。 FreeBSD をインストールするコンピュータと同じ地域のミラーサイトを選んでください。ターゲットコンピュータの近くにミラーがあると、ファイルのダウンロードは早く終わるので、インストールの時間が短くなります。

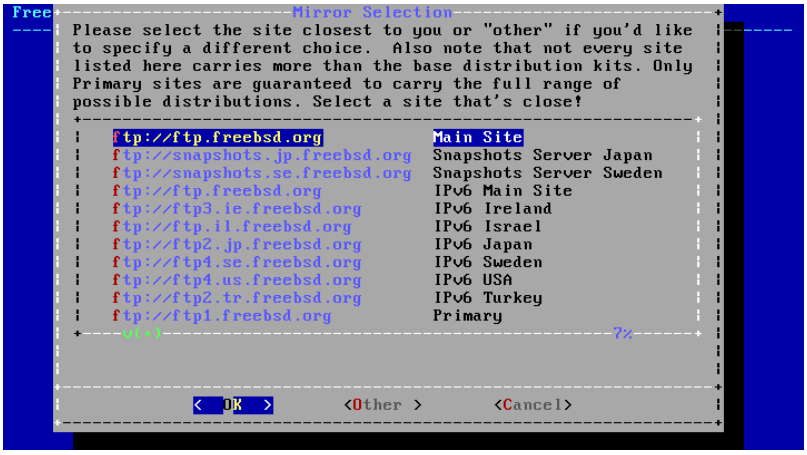


図2.9 ミラーサイトの選択

インストールファイルをローカルメディアに用意できたら、インストールは先に進みます。

2.7. ディスク領域の割り当て

FreeBSD へディスク領域を割り当てるには 3 つの方法があります。 Guided によるパーティションの分割方法では、ディスクパーティションを自動的に設定します。一方 Manual によるパーティションの分割方法は、 高度な知識を持つユーザ向けで、カスタマイズしたパーティションを作成できます。 3 つ目の方法として、シェルを利用し、gpart(8), fdisk(8), bsdlabel(8) のようなコマンドラインのプログラムを実行する方法があります。

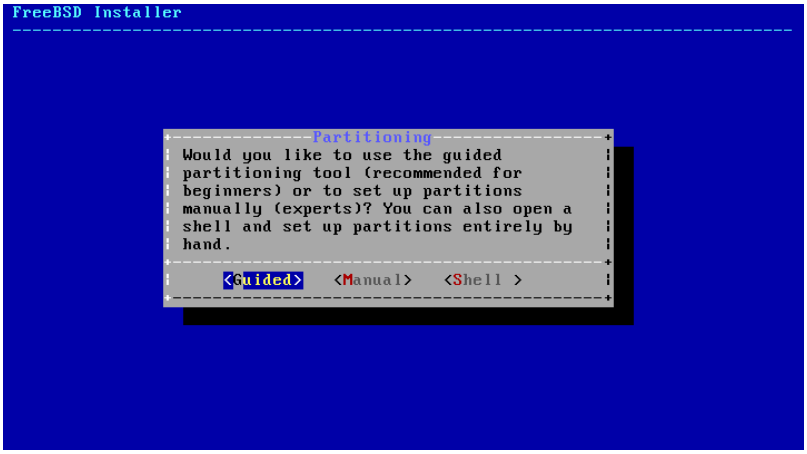


図2.10 パーティションの分割に Guided と Manual のどちらを用いるかの選択

この章では、 ディスクパーティションをレイアウトする際の検討事項を説明します。 その後、Guided Partitioning および Manual Partitioning screens の使用方法について説明します。

2.7.1. パーティションレイアウトのデザイン

ファイルシステムのレイアウトを行う際には、ハードディスクの外周部は内周部よりもデータ転送が速いということを思い出してください。これに従えば、小さくて激しくアクセスされるファイルシステムを外周付近に、 /usr のようなより大きなパーティションはディスクの内側に配置すべきでしょう。 そのため、パーティションを作成する際には、/、 スワップ、/var, /usr のような順で作ってゆくのがよいでしょう。

/var パーティションのサイズは、 あなたが計算機をどのように使おうとしているかを反映します。 このパーティションには主としてメールボックスやログファイル、 プリンタスプールが置かれます。 メールボックスとログファイルは、 システムのユーザ数やログの保持期間に依存して予期し得ぬサイズにまで成長する可能性があります。 概して、ほとんどのユーザは、/var にギガバイト以上の空き容量を必要とはしないでしょう。

注記

時には、たくさんのディスク容量が /var/tmp に必要になるときがあります。新しいソフトウェアをインストールする際、package のツールは、package の一時的なコピーを /var/tmp 以下に展開します。/var/tmp 以下に十分なディスク容量が用意されていないと、Firefox, OpenOffice や LibreOffice のような、大きなソフトウェア package のインストールが、困難になることがあります。

/usr パーティションには、FreeBSD Ports Collection およびシステムのソースコードを含む、システムをサポートするのに必要な多くのファイル群が置かれます。このパーティションには、少なくとも 2 ギガバイトを用意することをおすすめします。

パーティションのサイズを考える時、 必要量を念頭に置いてください。 別のパーティションには潤沢にスペースが余っているのに、 あるパーティションでスペースが足らないままというのは、フラストレーションがたまるものです。

経験からスワップパーティションのサイズは物理メモリ (RAM) の 2 倍というのが一般的です。RAM の少ないシステムでは、もっとスワップを増した方が性能がよくなります。スワップが少なすぎる設定は、あなたが後にメモリを増設したときに問題を起すばかりではなく、VM ページスキャニングコードの能率を落します。

複数の SCSI ディスクや異なるコントローラで操作される複数の IDE ディスクを持つ大規模なシステムでは、それぞれのドライブ (4 台まで) にスワップを設定することを推

奨します。 各ドライブのスワップパーティションはほぼ同一サイズであるべきです。 カーネルは任意のサイズを扱うことができますが、 内部のデータ構造は最大のスワップパーティションの 4 倍に調節されます。スワップパーティションをほぼ同一のサイズにしておくことで、カーネルはスワップスペースを最適なかたちでディスクをまたいでストライプさせることができます。あなたが通常スワップをたくさん使わないとしても、多くのスワップサイズを用意しておくと良いでしょう。 プログラムが暴走しても再起動させられる前に回復することが容易になります。

システムを適切にパーティション化することで、 小さいが書き込みの激しいパーティションによって引き起こされるフラグメント化を、読み出し専門のパーティションにまで波及させずにすみます。 また、書き込みの激しいパーティションをディスクの周辺部に配置することで、I/O パフォーマンスを増大させることができます。大きなパーティション内の I/O パフォーマンスもまた必要とされているでしょうが、 ディスク周辺部へ移動させたとしても、/var を周辺部に移動させることによって大きな効果が得られたのとは対照的に、意味のあるパフォーマンスの増加は見込めないでしょう。

2.7.2. Guided によるパーティションの分割

複数のディスクが接続されている場合には、FreeBSD をインストールするディスクを選択してください。

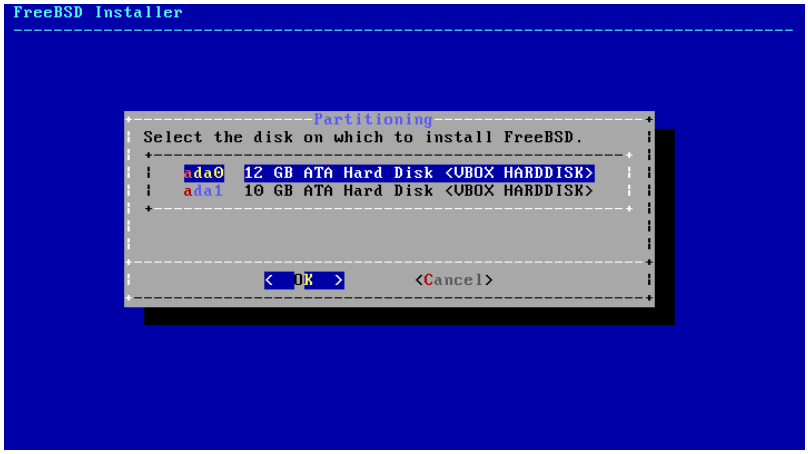


図2.11 複数のディスクから選択する

ディスクのすべて、または一部を FreeBSD に割り当てます。[Entire Disk] を選択すると、一般的なパーティションレイアウトが作成されます。[Partition] を選択すると、ディスクの使用していない領域にパーティションレイアウトを作成します。

図2.12 Entire Disk または Partition の選択

パーティションのレイアウトを作成したら、 正しく作成できているかどうか注意深く確認してください。間違いを発見したら、[Revert] を選択して、直前に作成したパーティションをリセットしてください。また、[Auto] を選択すると、自動的にもう一度 FreeBSD パーティションを作成します。 パーティションを手動で作成、変更、削除できます。 正しくパーティションを作成出来たら、[Finish] を選択し、インストールを進めてください。

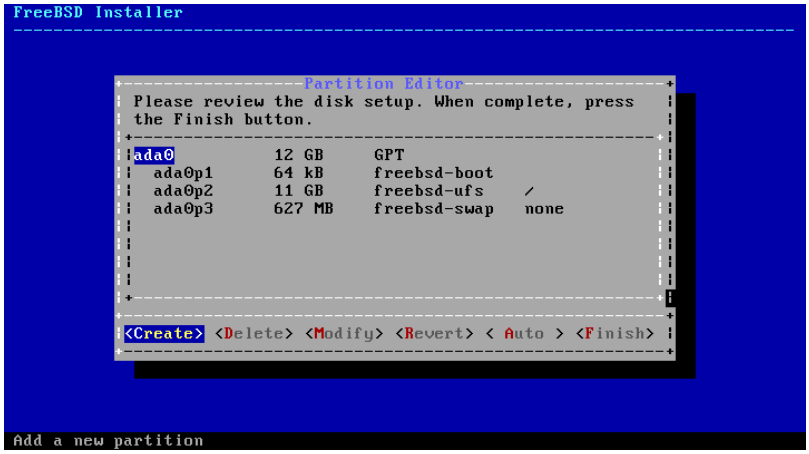


図2.13 作成されたパーティションの確認

2.7.3. Manual によるパーティションの分割

Manual によるパーティションの分割では、直接パーティションエディタが起動します。

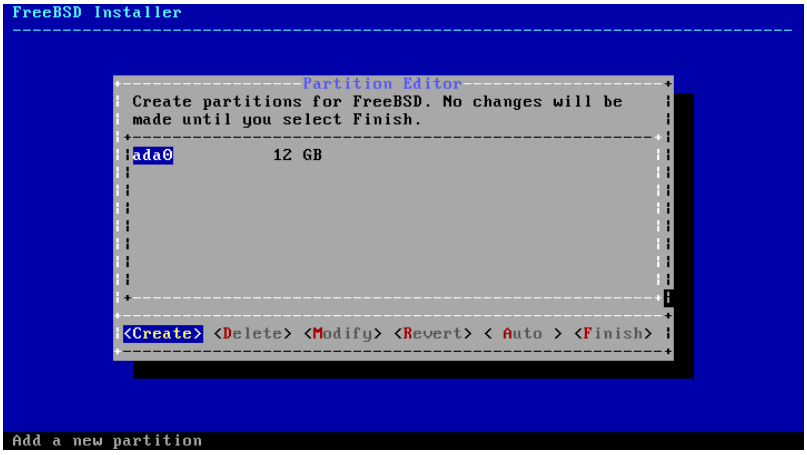


図2.14 Manual によるパーティションの分割

ドライブ (この例では ada0) を選び、[Create] を選択すると、partitioning scheme を選択するメニューが表示されます。

図2.15 手動でパーティションを作成する

PC 互換のコンピュータでは、通常 GPT によるパーティション分割が最も適切な選択となります。GPT に対応していないような古い PC オペレーティングシステムでは、MBR パーティションを使う必要があります。 他のパーティションスキームは、使うことがまれであったり、古いコンピュータで用いられます。

表2.1 パーティションスキーム

省略形	説明
APM	Apple パーティションマップ。PowerPC® Macintosh® で使われます。
BSD	MBR を用いない BSD ラベル。しばしば「dangerously dedicated mode」と呼ばれます。bsdlabel(8) をご覧ください。
GPT	GUID パーティションテーブル
MBR	Master Boot Record
PC98	MBR の亜種。NEC PC-98 コンピュータで使われます。
VTOC8	Volume Table Of Contents。Sun SPARC64 および UltraSPARC コンピュータで使われます。

パーティションスキームを選択して作成した後で、もう一度 [Create] を選択すると、新しいパーティションが作成されます。

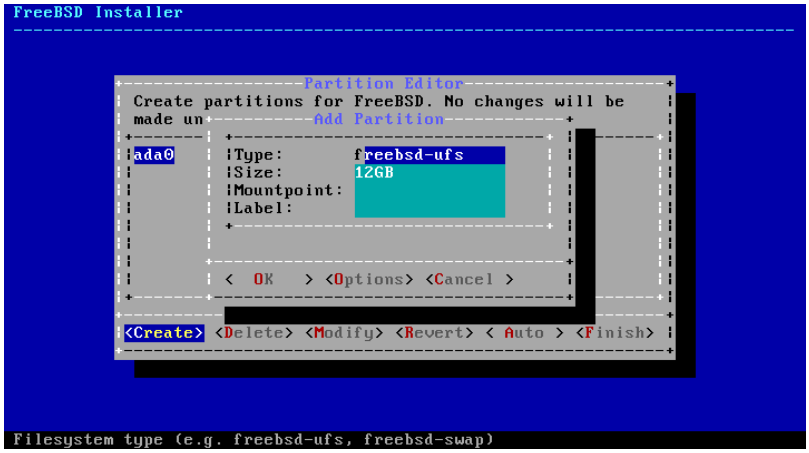


図2.16 手動でパーティションを作成する

標準の FreeBSD GPT のインストールでは、少なくとも 3 つのパーティションが使われます。

- `freebsd-boot` - FreeBSD ブートコード。
- `freebsd-ufs` - FreeBSD UFS ファイルシステム。
- `freebsd-swap` - FreeBSD スワップ空間。

他のパーティション形式に `freebsd-zfs` があります。これは FreeBSD ZFS ファイルシステムを含めるためのものです。The Z File System (ZFS) をご覧ください。利用可能な GPT パーティションタイプについては、gpart(8) をご覧ください。

複数のファイルシステムのパーティションを使うことができます。`/`, `/var`, `/tmp` そして `/usr` といった伝統的なパーティション分割のレイアウトを好む人もいます。レイアウトの例が 例2.3「伝統的なファイルシステムのパーティションを作成する。」にあります。

サイズを入力する際に、K (キロバイト)、M (メガバイト)、G (ギガバイト) といった通常の省略形を使用出来ます。

ヒント

セクタを適切に配置することで、 最良のパフォーマンスを得ることができます。また、パーティションサイズを 4K バイトの偶数倍にすると、512 バイトまたは 4K バイトのセクタでドライブが配置しやすくなります。一般的に、4K の偶数倍の場所からパーティションが開始するように設定する簡単な方法は、1M または 1G の偶数倍のパーティションサイズを用いることです。 ただし、例外があり、現在のところ freebsd-boot パーティションは、ブートコードの制限により 512K 以下である必要があります。

ファイルシステムを持つパーティションでは、マウントポイントが必要となります。1 つの UFS パーティションだけを作成するのであれば、マウントポイントは `/` となります。

同様に label の作成も必要です。ラベルは作成したパーティションを認識するための名前です。ドライブ名や番号は、ドライブが別のコントローラやポートに接続されると変わることがありますが、パーティションラベルは変わりません。`/etc/fstab` のようなファイルの中で、ドライブ名やパーティション番号ではなく、ラベルを参照することにより、システムがハードウェアの変更に対して、より寛容になります。GPT のラベルは、ディスクが接続されると `/dev/gpt/` に現れます。他のパーティションスキームでは別のラベルとなり、`/dev/` 以下の異なるディレクトリにラベルが現れます。

ヒント

同じ名前による衝突を避けるため、 すべてのファイルシステムに対し、一意的な名前使ってください。 コンピュータ名、使用、位置情報を表す単語をラベルに追加できます。 たとえば、研究室のコン

ピュータの UFS のルートパーティションを labroot または rootfs-lab とします。

例2.3 伝統的なファイルシステムのパーティションを作成する。

伝統的なパーティションレイアウト (/, /var, /tmp および /usr ディレクトリが各パーティションの別のファイルシステム) を作成するには、GPT パーティションスキームを作成し、その後、示されているようにパーティションを作成してください。示されているパーティションサイズは 20G のディスク用です。ディスクにより多くの容量があれば、swap または /var パーティションを大きく取ると良いでしょう。ここで示されているラベルには、example を意味する ex が付けられていますが、実際には上で説明したように、これとは別のユニークなラベルをつけてください。

FreeBSD の gptboot は、デフォルトでは最初に見つかった UFS パーティションが、/ パーティションであることを前頭としています。

パーティションタイプ	サイズ	マウントポイント	ラベル
freebsd-boot	512K		
freebsd-ufs	2G	/	exrootfs
freebsd-swap	4G		exswap
freebsd-ufs	2G	/var	exvarfs
freebsd-ufs	1G	/tmp	extmpfs
freebsd-ufs	デフォルト (ディスクの残りのすべての容量)	/usr	exusrfs

カスタムパーティション を作成したら [Finish] を選択して、インストールを先に進んでください。

2.8. インストール操作の確定

インストールによるハードディスクへの変更を中止するなら、ここが最後のチャンスです。

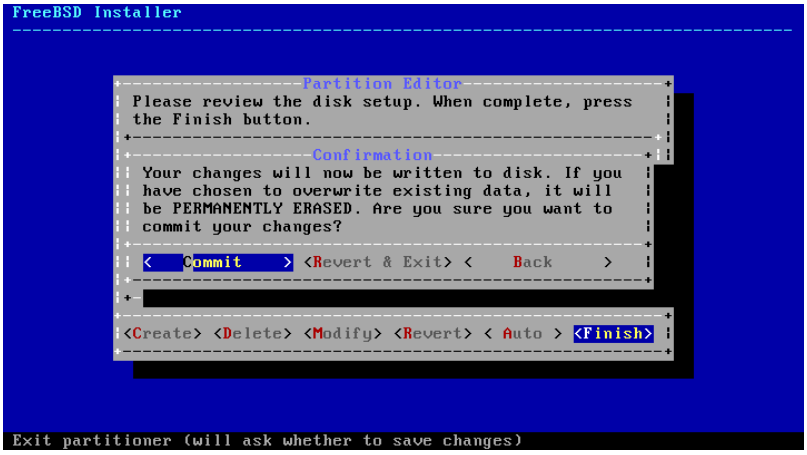


図2.17 最後の確認

[Commit] を選択して、Enter キーを押すとインストールが開始します。もし変更が必要であれば、[Back] を選択してパーティションエディタまで戻ってください。[Revert & Exit] を選択すると、ハードドライブへの変更なしにインストールを終了できます。

インストールにかかる時間は、どのディストリビューションを選んだか、どのインストールメディアを使ったか、 そしてコンピュータの速度にも依存します。 進行状況を表すメッセージが逐次表示されます。

まず最初に、インストーラは、ディスクにパーティション情報を書き込み、`newfs` を実行してパーティションを初期化します。

ネットワーク経由でのインストールでは、 bsdinstall は、インストールに必要な配布ファイルをダウンロードします。

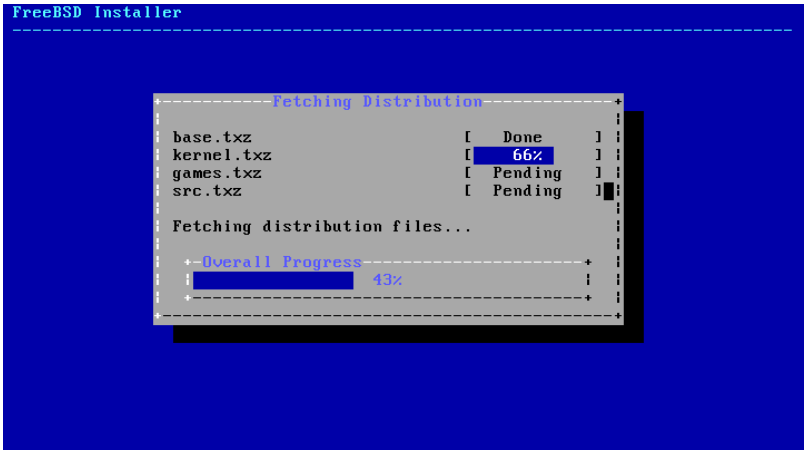


図2.18 配布ファイルのダウンロード

次に、ダウンロードの際にエラーが含まれなかったか、インストールメディアからの読み取り中に読み間違いが起きなかったかどうか等、配布ファイルの完全性の検証が行われます。

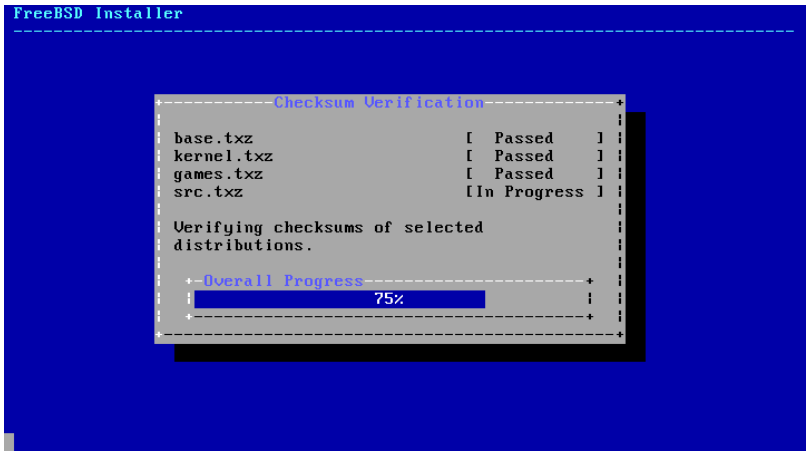


図2.19 配布ファイルの検証

最後に、検証された配布ファイルがディスクへ取り出されます。

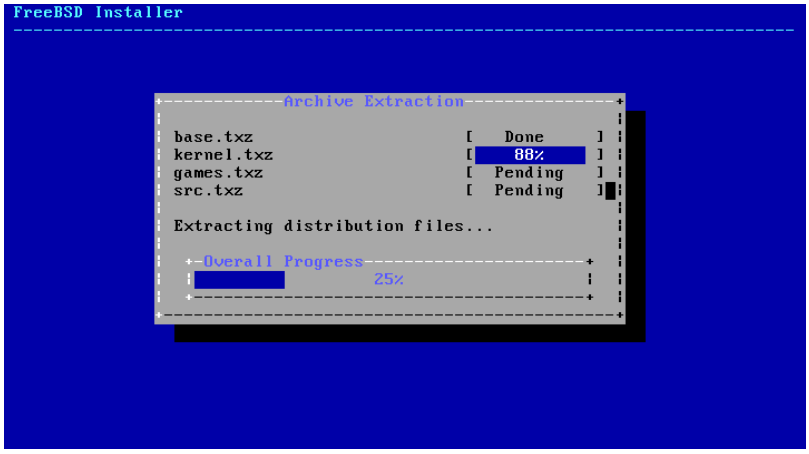


図2.20 配布ファイルの取り出し

必要な配布ファイルがすべて取り出されたら、 bsdinstall は、 インストール後の作業 (「インストール後の作業」) に進みます。

2.9. インストール後の作業

FreeBSD のインストールが完了したら、さまざまなオプションの設定に移ります。新しくインストールした FreeBSD を起動する前に、最後のメニューの configuration オプションの項目に入りなおすことでも、オプションの設定を変更できます。

2.9.1. root パスワードの設定

root のパスワードを設定する必要があります。パスワードを入力している際に、入力している文字は画面に表示されません。パスワードの入力後、もう一度入力する必要があります。これは入力ミスを防ぐためです。

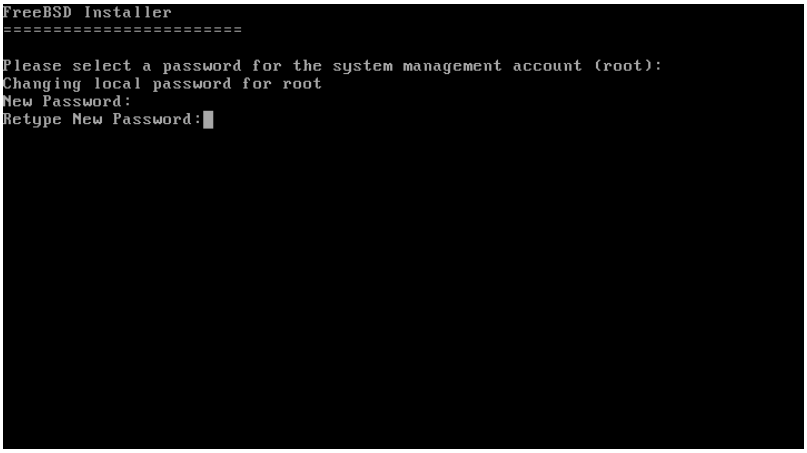


図2.21 root パスワードの設定

パスワードの入力に成功したら、インストールは次に進みます。

2.9.2. ネットワークインタフェースの設定

注記

bootonly によるインストール作業の一部として、すでにネットワークの設定を終えているのであれば、この作業を飛ばすことができます。

次に、コンピュータが認識したすべてのネットワークインタフェースが表示されます。設定するネットワークインタフェースを選んでください。

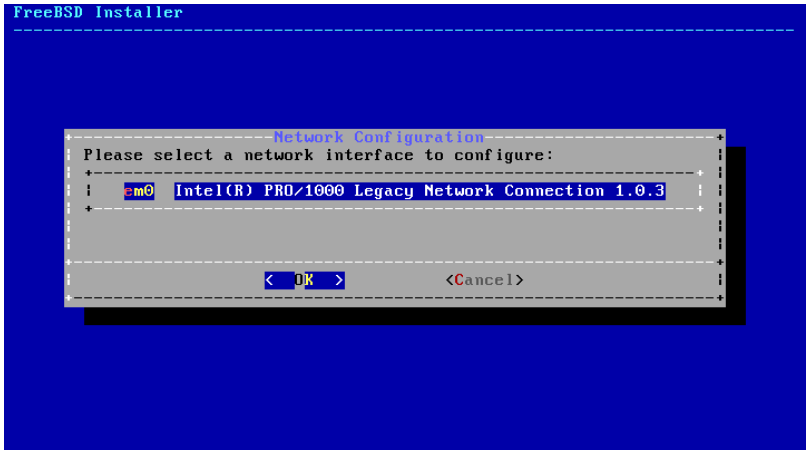


図2.22 イーサネットインタフェースの選択

2.9.2.1. ワイヤレスネットワークインタフェースの設定

ワイヤレスネットワークインタフェースを選択したのであれば、ネットワークに接続するようにワイヤレス認証およびセキュリティパラメータの入力が必要となります。

ワイヤレスネットワークは Service Set Identifier (SSID) によって識別されます。SSID は、それぞれのネットワークに与えられる、短く、一意的な名前です。

ほとんどのワイヤレスネットワークは、送信データを暗号化して、情報を盗聴から保護します。WEP のような古い暗号の安全性は低いので、WPA2 暗号を強く薦めます。

ワイヤレスネットワークへ接続する最初のステップは、　ワイヤレスアクセスポイントのスキャンです。

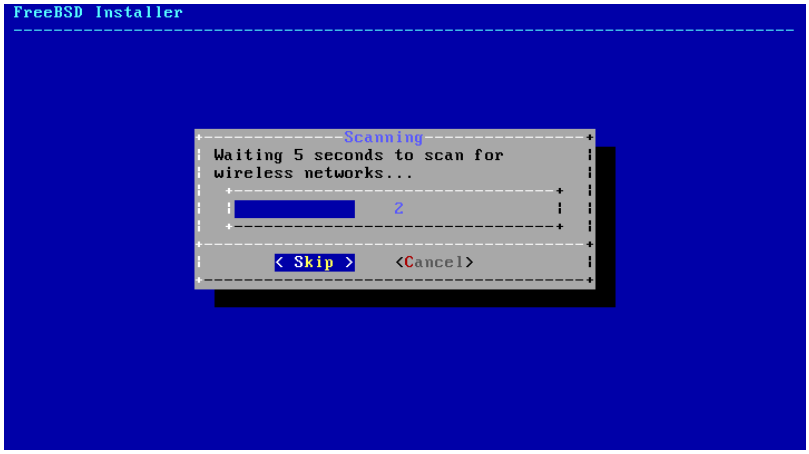


図2.23 ワイヤレスアクセスポイントのスキャン

スキャンで見つかった SSID の一覧が、そのネットワークで利用できる暗号化のタイプの説明とともに表示されます。もし、期待した SSID が一覧に表示されないのであれば、[Rescan] を選択してもう一度スキャンしてください。もし、期待したネットワークが表示されないのであれば、接続のためのアンテナやコンピュータをアクセスポイントの近くに移動させてみてください。その後もう一度スキャンしてください。

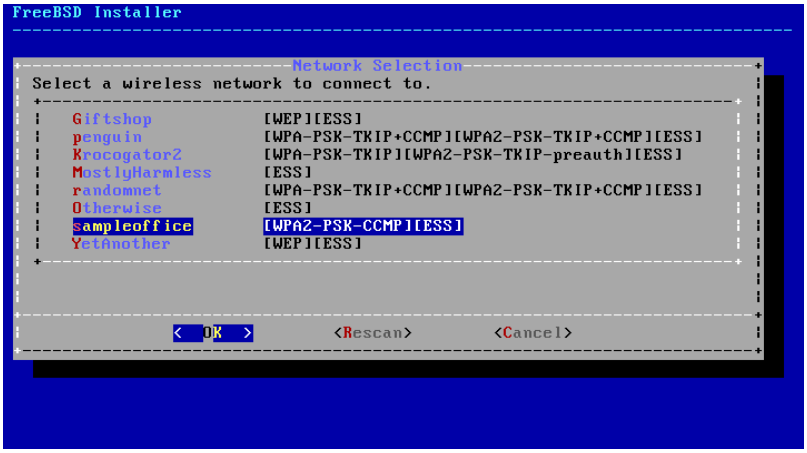


図2.24 ワイヤレスネットワークの選択

ネットワークの選択後、ワイヤレスネットワークに接続するための暗号情報を入力します。WPA2 では、パスワード (Pre-Shared Key または PSK とも呼ばれます) のみが必要です。セキュリティ上の観点から、入力ボックスに入力した文字はアスタリスクで表示されます。

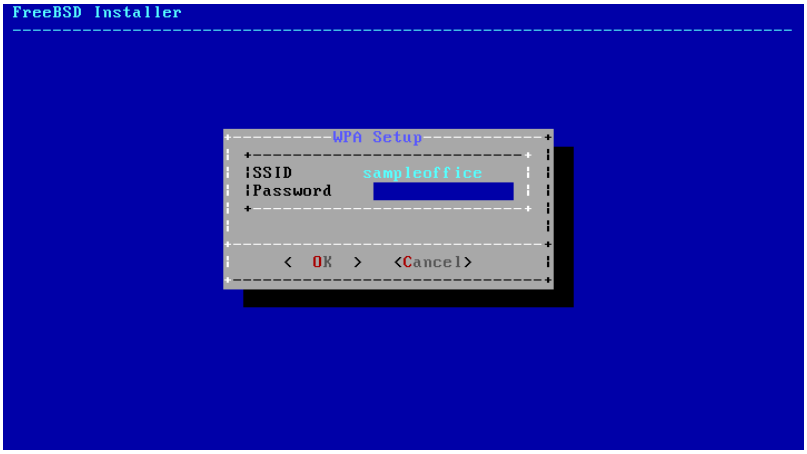


図2.25 WPA2 のセットアップ

ワイヤレスネットワークを選択後、ネットワークの設定は接続情報の入力に進みます。

2.9.2.2. IPv4 ネットワークの設定

IPv4 ネットワークを使うかどうかを選択してください。これは、ネットワーク接続の最も一般的なタイプです。

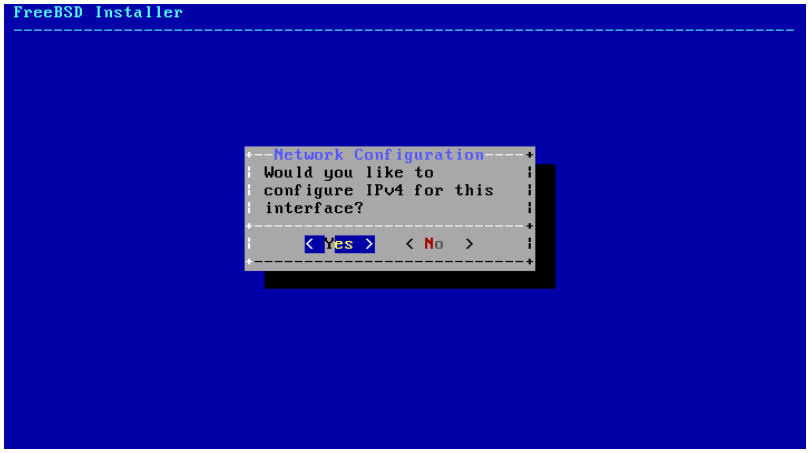


図2.26 IPv4 ネットワークの選択

IPv4 の設定方法は 2 通りあります。DHCP はネットワークインタフェースを自動的に適切に設定する方法で、推奨されています。Static 設定では、ネットワーク情報を手動で入力する必要があります。

注記

適当なネットワーク情報を入力しても動かないので、ネットワーク管理者またはサービスプロバイダから 「ネットワークの詳細をまとめる」に示されている情報を入手してください。

2.9.2.2.1. IPv4 DHCP ネットワークの設定

DHCP サーバを利用できるのであれば、[Yes] を選択して、ネットワークインタフェースの設定を自動的に行ってください。

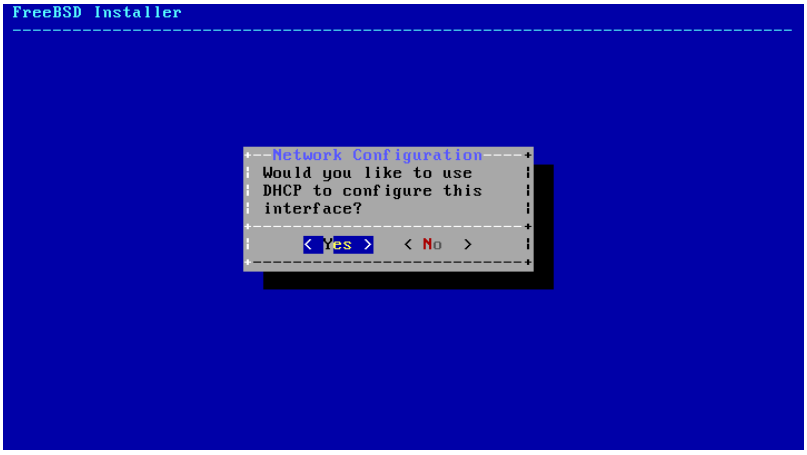


図2.27 IPv4 DHCP 設定の選択

2.9.2.2.2. 静的な IPv4 ネットワークの設定

静的なネットワークインタフェースの設定では、いくつかの IPv4 情報を入力する必要があります。

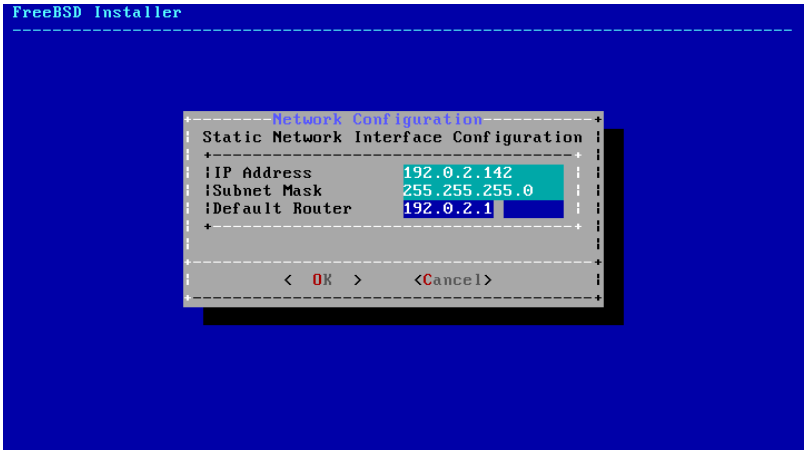


図2.28 IPv4 Static 設定

- `IP Address` - コンピュータに手動で与える IP アドレスです。このアドレスは一意的なものである必要があり、ローカルネットワーク上の他のネットワーク機器で使われていてはいけません。

- `Subnet Mask` - ローカルネットワークで用いられるサブネットマスクです。一般的には `255.255.255.0` です。

- Default Router - このネットワークのデフォルトルータの IP アドレスです。これは、通常ルータか、ローカルネットワークをインターネットへ接続するネットワーク機器のことです。デフォルトゲートウェイ とも呼ばれます。

2.9.2.3. IPv6 ネットワークの設定

IPv6 はネットワークの設定の新しい方法です。IPv6 が利用でき、希望するのであれば、[Yes] を選択してください。

図2.29 IPv6 ネットワークの設定

IPv6 の設定に関しても 2 つの方法があります。 SLAAC (StateLess Address AutoConfiguration) は、ネットワークインタフェースを自動的に適切に設定します。Static 設定では、ネットワーク情報を手動で入力する必要があります。

2.9.2.3.1. IPv6 Stateless Address Autoconfiguration

SLAAC では、IPv6 ネットワークコンポーネントがローカルルータから自動設定に関する情報をリクエストできるようにします。詳細については、RFC4862 をご覧ください。

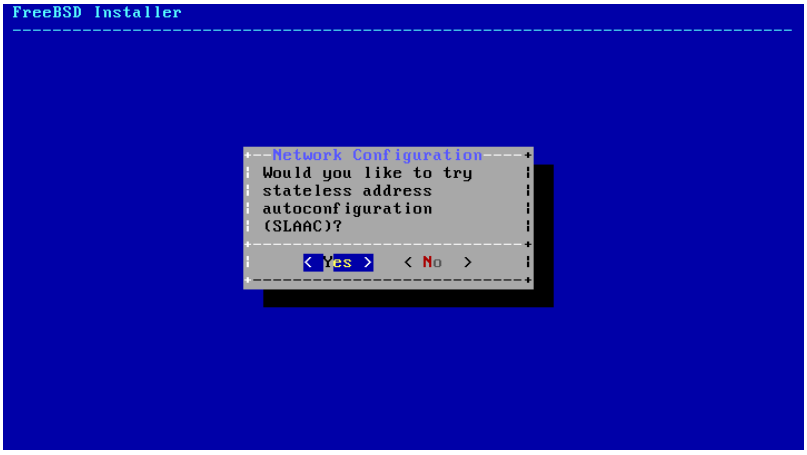


図2.30 IPv6 SLAAC 設定の選択

2.9.2.3.2. 静的な IPv6 ネットワークの設定

静的なネットワークインタフェースの設定では、IPv6 の設定に関する情報を入力する必要があります。

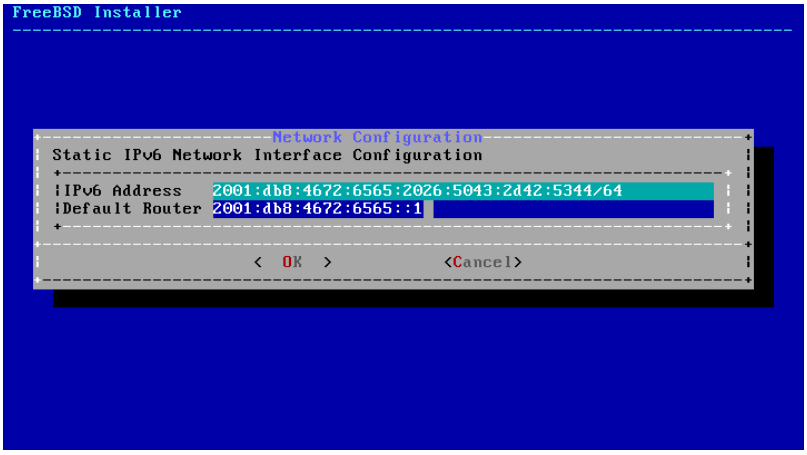


図2.31 IPv6 static 設定

- IPv6 Address - このコンピュータに割り当てられた IP アドレスです。手動で設定します。このアドレスは一意的なものである必要があり、ローカルネットワーク上の他のネットワーク機器で使われていてはいけません。
- Default Router - このネットワークのデフォルトルータの IPv6 アドレスです。これは、通常ルータか、ローカルネットワークをインターネットへ接続するネットワーク機器のことです。デフォルトゲートウェイ とも呼ばれます。

2.9.2.4. DNS の設定

Domain Name System (または DNS) リゾルバは、ホスト名とネットワークアドレスを変換します。すでに DHCP または SLAAC を使って自動的にネットワークインタフェースを設定したのであれば、`Resolver Configuration` には値がすでに入っているでしょう。そうでなければ、`Search` フィールドにローカルネットワークのドメイン名を入力してください。`DNS` #1 および `DNS` #2 は、ローカル DNS サーバの IP アドレスです。少なくとも、1 つの DNS サーバは必要です。

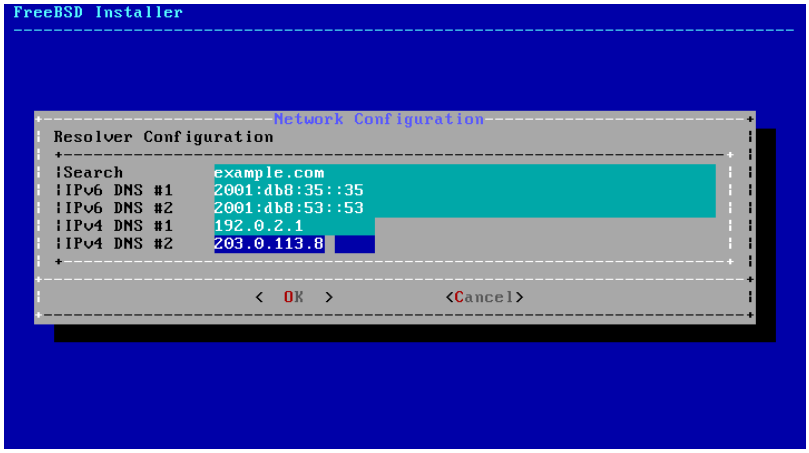


図2.32 DNS の設定

2.9.3. タイムゾーンの設定

使用しているコンピュータのタイムゾーンを設定することで、地域による時刻の違いが自動的に調整され、タイムゾーンに関連した機能が適切に取り扱われます。

ここでの例では、コンピュータが United States の Eastern タイムゾーンにあるものとします。実際の地理的位置を選択してください。

図2.33 ローカルまたは UTC クロックの選択

コンピュータのクロックの設定にしたがい、[Yes] または、[No] を選択し、Enter を押しください。UTC とローカルタイムのどちらを選ぶべきかわからない場合には、[No] を選択して、より一般的なローカルタイムに設定してください。

図2.34 地域の選択

矢印キーを使って、適切な地域を選択し、Enter を押してください。

図2.35 国名の選択

矢印キーを使って、適切に国名を選び、Enter を押してください。

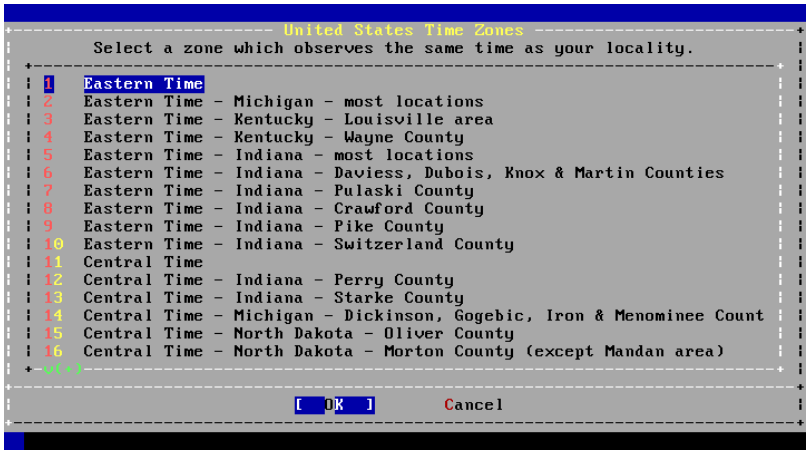


図2.36 タイムゾーンの選択

矢印キーを使って適切なタイムゾーンを選択し、Enter を押してください。

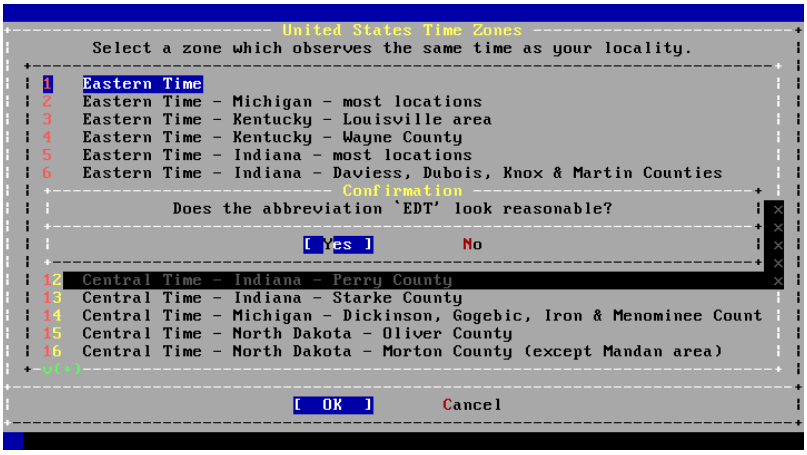


図2.37 タイムゾーンの確定

タイムゾーンの省略形が正しいかどうかを確認してください。 問題がないようであれば Enter を押して、インストール後の設定を続けてください。

2.9.4. 有効にするサービスの選択

ブート時に起動するシステムサービスを追加で有効に出来ます。 これらのサービスはすべてオプションです。

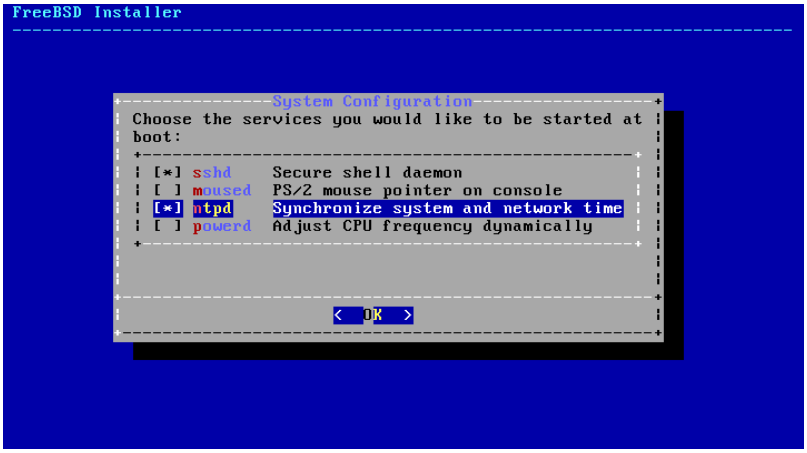


図2.38 追加で有効にするサービスの選択

- `sshd` - 安全なリモートアクセスのためのセキュアシェル (SSH) デーモン
- `moused` - システムのコンソールで、マウスを利用できるようにします。
- `ntpd` - 自動時刻同期のネットワークプロトコル (NTP) デーモン

- powerd - 電源の管理およびエネルギーを節約するための電源コントロールユーティリティ

2.9.5. クラッシュダンプの設定

次に bsdinstall は、クラッシュダンプの設定に移ります。システムのデバッグを行う上で、クラッシュダンプにより得られる情報は非常に有用です。可能であればクラッシュダンプを有効にすると良いでしょう。[Yes] を選択してクラッシュダンプを有効にするか、または [No] を選択してクラッシュダンプを無効化し、先に進んでください。

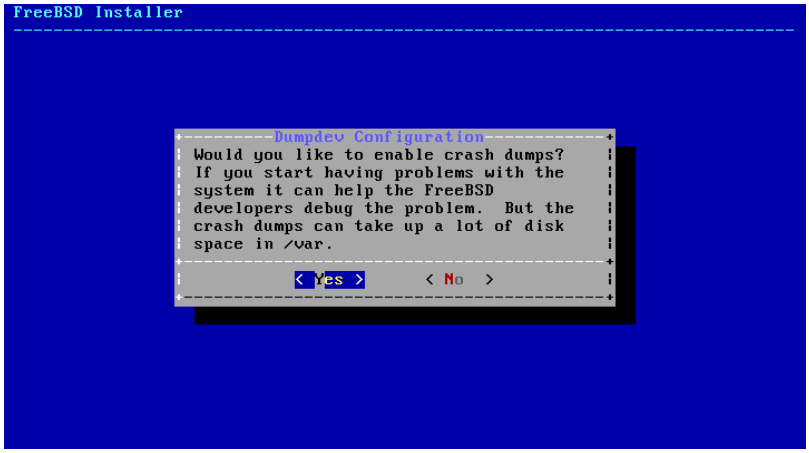


図2.39 クラッシュダンプの設定

2.9.6. ユーザの追加

システムに root でログインすることを避けるため、インストール時には、少なくとも一人のユーザを追加してください。root 権限では、実行に対して制限がなく、また、保護されません。 通常のユーザでログインすることにより、 安全でセキュリティ的に危険が少なくなります。

[Yes] を選択し、新しいユーザを追加してください。

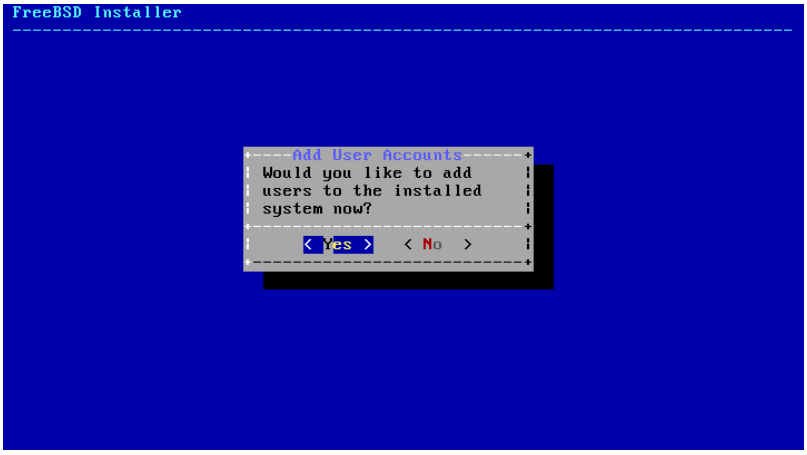


図2.40 新しいユーザのアカウントの作成

追加するユーザの情報を入力してください。

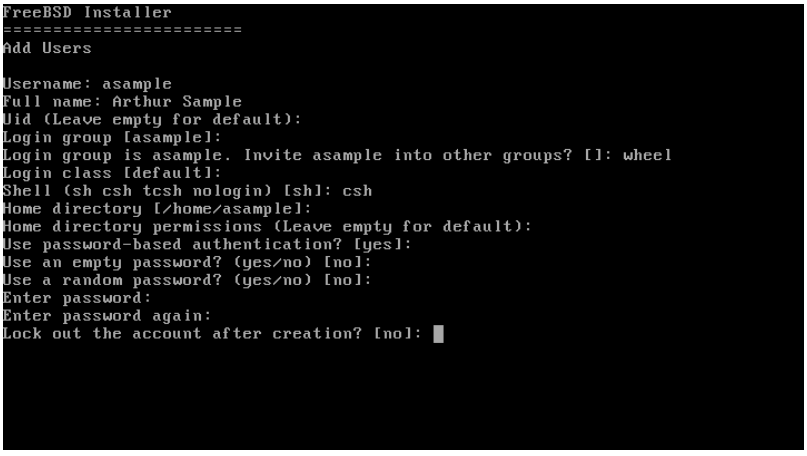


図2.41 ユーザ情報の入力

- `Username` - ログイン時のユーザ名を入力します。典型的なものでは、ファーストネームの最初の文字とラストネームを組み合わせます。
- `Full name` - ユーザのフルネーム
- `Uid` - ユーザ ID 番号。通常は、システムが自動的に割り当てるように、空欄のままにします。
- `Login group` - 新しいユーザのログイングループ。空欄のままにすると、自動的に割り当てられます

- `Invite user into other groups?` - ユーザを別のグループのメンバーとして追加するかどうか。
- `Login class` - 空欄にするとデフォルトの設定になります。
- `Shell` - ユーザのログインシェル。この例では csh(1) に設定されています。
- `Home directory` - ユーザのホームディレクトリ。通常は、デフォルトの場所が適切です。
- `Home directory permissions` - ユーザのホームディレクトリの権限。通常は、デフォルトが適切です。
- `Use password-based authentication?` - 通常は `yes` です。
- `Use an empty password?` - 通常は `no` です。
- `Use a random password?` - 通常は `no` です。
- `Enter password` - ユーザのパスワードです。入力している文字は画面に表示されません。
- `Enter password again` - 確認のため、パスワードをもう一度入力します。
- `Lock out the account after creation?` - 通常は `no` です。

すべてを入力したら、サマリが表示され、 正しいかどうかの確認を求められます。 入力した情報に間違いがあれば、`no` を入力してもう一度作業を行なってください。すべてが正しく入力されていれば、`yes` を入力して、新しいユーザを作成してください。

```
Login group [asample]:
Login group is asample. Invite asample into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh nologin) [sh]: csh
Home directory [/home/asample]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : asample
Password   : *****
Full Name  : Arthur Sample
Uid        : 1001
Class      :
Groups     : asample wheel
Home       : /home/asample
Home Mode  :
Shell      : /bin/csh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (asample) to the user database.
Add another user? (yes/no):
```

図2.42 ユーザおよびグループの管理を終了する

さらにユーザを追加するのであれば、Add another user? の質問に対し、yes を入力してください。no を入力すると、ユーザの追加が終わり、次に進みます。

ユーザの追加や、ユーザ管理の詳細については、「この章では」を参照してください。

2.9.7. 最後の設定

すべてをインストールし、設定が終わった後に、最後に設定を修正する機会が与えられます。

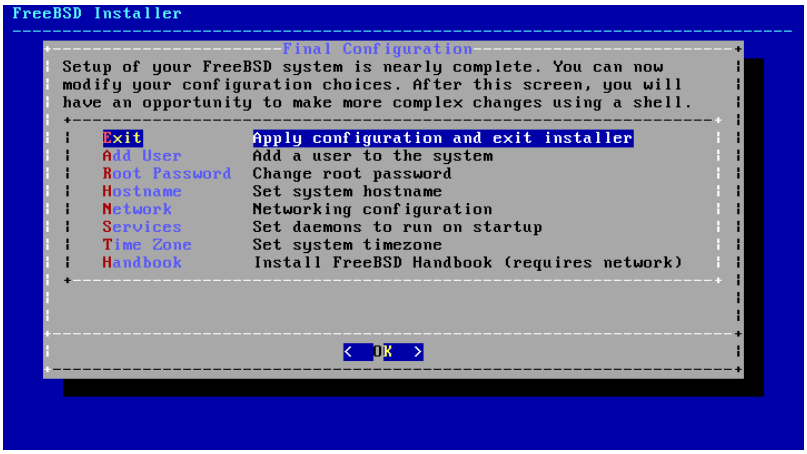


図2.43 最後の設定

インストールを完了する前に、このメニューを使って変更、または、追加の設定を行なってください。

- Add User - 「ユーザの追加」で説明しています。
- Root Password - 「root パスワードの設定」で説明しています。
- Hostname - 「ホスト名の設定」で説明しています。
- Network - 「ネットワークインタフェースの設定」で説明しています。
- Services - 「有効にするサービスの選択」で説明しています。
- Time Zone - 「タイムゾーンの設定」で説明しています。
- Handbook - FreeBSD ハンドブック (現在あなたが読んでいるこの文書のことです) のダウンロードとインストール。

最後の設定が完了したら、Exit を選んでインストールを終了してください。

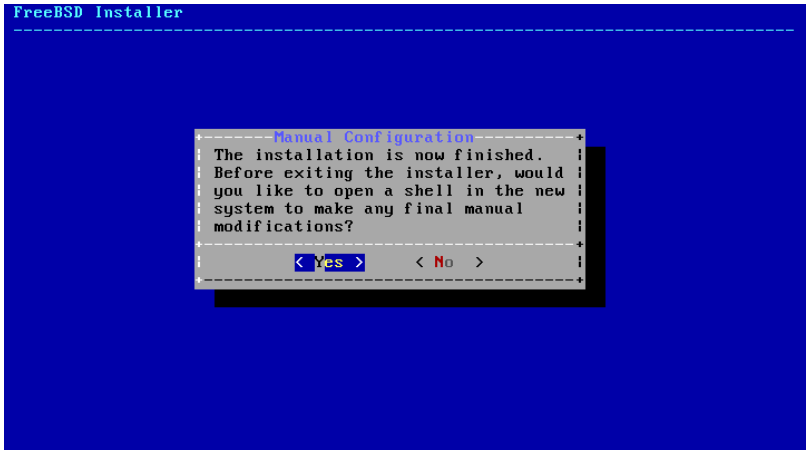


図2.44 Manual Configuration

新しいシステムを再起動する前に、bsdinstall は追加の設定が必要かどうかを尋ねてきます。[Yes] を選択して新しいシステムのシェルに入るか、または [No] を選択して、インストールの最後のステップに進んでください。

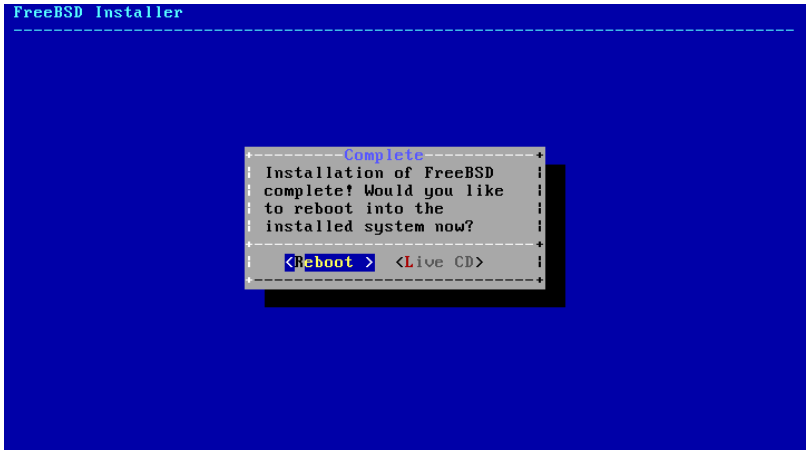


図2.45 インストールの終了

追加の設定や、特別なセットアップが必要であれば、[Live CD] を選んでインストールメディアを Live CD モードで起動してください。

インストールが終わったら、[Reboot] を選んで、コンピュータを再起動し、新しい FreeBSD システムを動かしてください。再起動する前には、忘れずに FreeBSD インストール CD, DVD または USB メモリスティックを外してください。さもないと、もう一度インストールメディアから起動してしまいます。

2.9.8. FreeBSD の起動とシャットダウン

2.9.8.1. FreeBSD/i386 の起動

FreeBSD の起動時には、多くのメッセージが画面に表示されます。ほとんどは、画面上からスクロールして見えなくなってしまいますが、 これは正常です。システムの起動が終わった後で、ログインプロンプトが表示されます。画面上からスクロールして消えてしまったメッセージは Scroll-Lock を押し、 scroll-back buffer で、見ることができます。PgUp, PgDn そして矢印キーでメッセージをスクロールバックできます。Scroll-Lock をもう一度押すと、画面のロックを外し、通常の画面に戻ることができます。

`login:` プロンプトで、インストール時に追加したユーザ名を入力してください。この例では、`asample` です。必要のない限り、`root` でのログインを避けてください。

上で示した scroll-back buffer では、バッファの容量制限により、すべてが表示されないかもしれません。ログインの後にプロンプトからコマンドラインで `dmesg | less` と入力することによって、すべてを見ることができます。確認後に q を押すと、コマンドラインに戻ります。

典型的なブートメッセージ (バージョン情報は省略しています)

```
Copyright (c) 1992-2011 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, ↺
1994
        The Regents of the University of California. All rights ↺
reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.

    root@farrell.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC amd64
CPU: Intel(R) Core(TM)2 Duo CPU     E8400  @ 3.00GHz (3007.77-MHz K8-
class CPU)
  Origin = "GenuineIntel"  Id = 0x10676  Family = 6  Model = 17  ↺
Stepping = 6
  ↺
Features=0x783fbff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE
  Features2=0x209<SSE3,MON,SSSE3>
  AMD Features=0x20100800<SYSCALL,NX,LM>
  AMD Features2=0x1<LAHF>
real memory  = 536805376 (511 MB)
avail memory = 491819008 (469 MB)
Event timer "LAPIC" quality 400
ACPI APIC Table: <VBOX   VBOXAPIC>
ioapic0: Changing APIC ID to 1
ioapic0 <Version 1.1> irqs 0-23 on motherboard
kbd1 at kbdmux0
acpi0: <VBOX VBOXXSDT> on motherboard
acpi0: Power Button (fixed)
acpi0: Sleep Button (fixed)
Timecounter "ACPI-fast" frequency 3579545 Hz quality 900
acpi_timer0: <32-bit timer at 3.579545MHz> port 0x4008-0x400b on ↺
acpi0
```

```
cpu0: <ACPI CPU> on acpi0
pcib0: <ACPI Host-PCI bridge> port 0xcf8-0xcff on acpi0
pci0: <ACPI PCI bus> on pcib0
isab0: <PCI-ISA bridge> at device 1.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <Intel PIIX4 UDMA33 controller> port ↺
0x1f0-0x1f7,0x3f6,0x170-0x177,0x376,0xd000-0xd00f at device 1.1 on ↺
pci0
ata0: <ATA channel 0> on atapci0
ata1: <ATA channel 1> on atapci0
vgapci0: <VGA-compatible display> mem 0xe0000000-0xe0ffffff irq 18 ↺
at device 2.0 on pci0
em0: <Intel(R) PRO/1000 Legacy Network Connection 1.0.3> port ↺
0xd010-0xd017 mem 0xf0000000-0xf001ffff irq 19 at device 3.0 on pci0
em0: Ethernet address: 08:00:27:9f:e0:92
pci0: <base peripheral> at device 4.0 (no driver attached)
pcm0: <Intel ICH (82801AA)> port 0xd100-0xd1ff,0xd200-0xd23f irq 21 ↺
at device 5.0 on pci0
pcm0: <SigmaTel STAC9700/83/84 AC97 Codec>
ohci0: <OHCI (generic) USB controller> mem 0xf0804000-0xf0804fff ↺
irq 22 at device 6.0 on pci0
usbus0: <OHCI (generic) USB controller> on ohci0
pci0: <bridge> at device 7.0 (no driver attached)
acpi_acad0: <AC Adapter> on acpi0
atkbdc0: <Keyboard controller (i8042)> port 0x60,0x64 irq 1 on acpi0
atkbd0: <AT Keyboard> irq 1 on atkbdc0
kbd0 at atkbd0
atkbd0: [GIANT-LOCKED]
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: model IntelliMouse Explorer, device ID 4
attimer0: <AT timer> port 0x40-0x43,0x50-0x53 on acpi0
Timecounter "i8254" frequency 1193182 Hz quality 0
Event timer "i8254" frequency 1193182 Hz quality 100
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff ↺
on isa0
atrtc0: <AT realtime clock> at port 0x70 irq 8 on isa0
Event timer "RTC" frequency 32768 Hz quality 0
ppc0: cannot reserve I/O port range
Timecounters tick every 10.000 msec
pcm0: measured ac97 link rate at 485193 Hz
em0: link state changed to UP
usbus0: 12Mbps Full Speed USB v1.0
ugen0.1: <Apple> at usbus0
uhub0: <Apple OHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on ↺
usbus0
cd0 at ata1 bus 0 scbus1 target 0 lun 0
cd0: <VBOX CD-ROM 1.0> Removable CD-ROM SCSI-0 device
cd0: 33.300MB/s transfers (UDMA2, ATAPI 12bytes, PIO 65534bytes)
cd0: Attempt to query device size failed: NOT READY, Medium not ↺
present
```

```
ada0 at ata0 bus 0 scbus0 target 0 lun 0
ada0: <VBOX HARDDISK 1.0> ATA-6 device
ada0: 33.300MB/s transfers (UDMA2, PIO 65536bytes)
ada0: 12546MB (25694208 512 byte sectors: 16H 63S/T 16383C)
ada0: Previously was known as ad0
Timecounter "TSC" frequency 3007772192 Hz quality 800
Root mount waiting for: usbus0
uhub0: 8 ports with 8 removable, self powered
Trying to mount root from ufs:/dev/ada0p2 [rw]...
Setting hostuuid: 1848d7bf-e6a4-4ed4-b782-bd3f1685d551.
Setting hostid: 0xa03479b2.
Entropy harvesting: interrupts ethernet point_to_point kickstart.
Starting file system checks:
/dev/ada0p2: FILE SYSTEM CLEAN; SKIPPING CHECKS
/dev/ada0p2: clean, 2620402 free (714 frags, 327461 blocks, 0.0% ↺
fragmentation)
Mounting local file systems:.
vboxguest0 port 0xd020-0xd03f mem ↺
0xf0400000-0xf07fffff,0xf0800000-0xf0803fff irq 20 at device 4.0 on ↺
pci0
vboxguest: loaded successfully
Setting hostname: machine3.example.com.
Starting Network: lo0 em0.
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 ↺
mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:9f:e0:92
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
Starting devd.
Starting Network: usbus0.
DHCPREQUEST on em0 to 255.255.255.255 port 67
DHCPACK from 10.0.2.2
bound to 192.168.1.142 -- renewal in 43200 seconds.
add net ::ffff:0.0.0.0: gateway ::1
add net ::0.0.0.0: gateway ::1
add net fe80::: gateway ::1
add net ff02::: gateway ::1
ELF ldconfig path: /lib /usr/lib /usr/lib/compat /usr/local/lib
32-bit compatibility ldconfig path: /usr/lib32
Creating and/or trimming log files.
Starting syslogd.
No core dumps found.
Clearing /tmp (X related).
Updating motd:.
Configuring syscons: blanktime.
```

```
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
10:a0:f5:af:93:ae:a3:1a:b2:bb:3c:35:d9:5a:b3:f3 root@machine3.↺
example.com
The key's randomart image is:
+--[RSA1 1024]----+
|    o..          |
|   o . .         |
|  .   o          |
|       o         |
|    o   S        |
|   + + o         |
|o . + *          |
|o+ ..+ .         |
|==o..o+E         |
+-----------------+
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
7e:1c:ce:dc:8a:3a:18:13:5b:34:b5:cf:d9:d1:47:b2 root@machine3.↺
example.com
The key's randomart image is:
+--[ DSA 1024]----+
|       ..     . .|
|      o  .   . + |
|     . ..   . E .|
|    . .  o o . . |
|     +  S = .    |
|    +  . = o     |
|     +  . * .    |
|    . .  o .     |
|      .o. .      |
+-----------------+
Starting sshd.
Starting cron.
Starting background file system checks in 60 seconds.

Thu Oct  6 19:15:31 MDT 2011

FreeBSD/amd64 (machine3.example.com) (ttyv0)

login:
```

低速なマシンでは、RSA および DSA キーを作成するのに少々時間がかかるかもしれません。これはインストール後の最初の起動時に、sshd のセットアップが自動的に開始する時に実行されます。その後の起動はより速くなるでしょう。

FreeBSD はデフォルトでは、グラフィカルな環境をインストールしませんが、多くのものを利用できます。6章X Window System をご覧ください。

2.9.9. FreeBSD のシャットダウン

適切に FreeBSD をシャットダウンすることは、ハードウェアをダメージから守ったり、データの保護につながります。ただ電源を落すということはしないでください。`wheel` グループのメンバとなっているユーザは、コマンドラインから `su` と入力し、`root` パスワードを入力して `root` となってください。または `root` としてログインし、`shutdown -p now` を使用します。システムは正しくシャットダウンし、電源が落ちます。

Ctrl+Alt+Del というキーの組合せを使ってシステムをリブートすることもできますが、通常の運用においてこれは推奨されません。

2.10. トラブルシューティング

この節では、インストールの際の、 これまで報告された共通の問題に対する解決のための情報が書いてあります。

2.10.1. なにかおかしいときには何をすればよいでしょうか

PC アーキテクチャのさまざまな制限により、100% 確実に原因を突き止めることは不可能ですが、失敗した時にいくつかできることがあります。

インストールする FreeBSD のバージョンの ハードウェアノート を調べて、使っているハードウェアに対応しているかどうかを確認してください。

もしハードウェアがサポートされているにもかかわらず、 動作しなかったり他の問題点がある時は、カスタムカーネル を構築する必要があります。`GENERIC` カーネルに含まれていないデバイスのサポートを追加することができます。 起動ディスクのカーネルでは、ほとんどのハードウェアデバイスの IRQ, IO アドレス、 DMA チャネルが工場出荷時の状態であると設定されています。もしハードウェアの設定が変更されていると、カーネルコンフィグレーションファイルを編集し、再コンパイルを行なって、これらの値を FreeBSD に設定しなければなりません。

存在しないデバイスを認識してしまうことにより、その後実際に存在するデバイスの認識を失敗してしまうことがあります。このような場合は衝突しているドライバを無効にします。

注記

いくつかのインストール上の問題はさまざまなハードウェア装置の、 特にマザーボードのファームウェアのアップデートで回避または緩和することができます。 マザーボードのファームウェアは BIOS と呼ばれることもあり、多くのマザーボードまたはコンピュータ製造メーカーは、 アップデートやアップグレード情報を載せているウェブサイトを用意しています。

通常、製造メーカーは、重要な更新のようなそれなりの理由がない限り、マザーボードの BIOS のアップグレードは行わないよう推奨しています。アップデートの過程で失敗する可能性があり、その場合 BIOS が不完全な状態になり、コンピュータが動作しない原因となり得るからです。

2.10.2. トラブルシューティングに関する Q & A

問: 起動時のハードウェア検出で、システムがハングアップします。 または、インストール中にシステムがおかしくなります。

答: i386, amd64 および ia64 プラットフォームにおいて、FreeBSD はシステムの設定を手助けするシステム ACPI サービスを、起動時に検出された場合に広く使います。残念ながら、まだいくつかの不具合が、ACPI ドライバとシステムのマザーボードおよび BIOS ファームウェア両方に存在しています。起動ステージ 3 において、ヒント情報 `hint.acpi.0.disabled` を以下のように設定すると ACPI を無効にできます。

```
set hint.acpi.0.disabled="1"
```

この設定はシステムが起動するたびにリセットされるので、 `/boot/loader.conf` ファイルに `hint.acpi.0.disabled="1"` を追加してください。ブートローダのより詳しい情報については「この章では」で説明します。

2.11. Live CD を使う

FreeBSD の live CD は、メインのインストールプログラムと同じ CD に含まれています。オペレーティングシステムに FreeBSD を使おうかどうか考えている方や、インストールする前に機能を試して見たいと思っている方に有用です。

注記

live CD を使う際は、以下のことに気をつけてください。

- システムにアクセスする際には、認証を求められます。 ユーザ名は `root`、パスワードは空欄としてください。
- システムは CD から起動するため、 ハードディスクにインストールされたシステムに比べ、パフォーマンスはかなり遅い可能性があります。

- live CD のユーザインタフェースはコマンドプロンプトです。グラフィカルなユーザインタフェースではありません。

第3章 FreeBSD 8.X のインストール

再構成、部分的に書き直し: Jim Mock.
Sysinstall について、ウォークスルー、スクリーンショット、及びその他原稿: Randy Pratt.

3.1. この章では

FreeBSD は、テキストベースの使いやすいインストールプログラムを用意しています。FreeBSD 9.0-RELEASE 以降では bsdinstall(8) と呼ばれるインストールプログラムを使い、FreeBSD 8.*X* では sysinstall(8) を使います。この章では、sysinstall(8) の使い方について説明します。bsdinstall(8) の利用については 2章FreeBSD のインストール (9.*X* 以降の場合) で説明します。

この章を読めば、次のことがわかるでしょう:

- FreeBSD インストールメディアの作り方。
- FreeBSD がハードディスクをどのように参照するか、またどのように分割するか。
- sysinstall(8) をどのように起動するか。
- sysinstall(8) が聞いてくる質問がどのような意味であり、またどのように答えれば良いか。

この章を読む前に、以下のことを確認して下さい:

- インストールしようとするバージョンの FreeBSD に付属しているサポートハードウェア一覧を読んで、システムのハードウェアがサポートされているかどうか確認して下さい。

注記

この章で説明しているインストール手順は、i386™ および FreeBSD/amd64 アーキテクチャを対象にしています。他のプラットフォームに特有の説明についても、該当する場所に明記しています。インストーラとこの文書で記述している内容との間にはいくらかズレがあります。この章を正確なインストールマニュアルとしてではなく、一般的なガイドとしてご利用ください。

3.2. ハードウェア要件

3.2.1. 最小構成

FreeBSD をインストールする最小構成は、 FreeBSD のバージョンやハードウェアのアーキテクチャによって異なります。

以下の節では、最小構成についての情報をまとめています。 また、FreeBSD のインストール方法によって、フロッピードライブ、 CDROM ドライブ、もしくはネットワークアダプタが必要となります。インストールメディアの準備方法については、「ブートメディアの準備」で説明されています。

3.2.1.1. FreeBSD/i386 および FreeBSD/pc98

FreeBSD/i386 と FreeBSD/pc98 の両方のアーキテクチャは、486 以上のプロセッサと、少なくとも 24 MB の RAM が必要です。最小のインストールでは、少なくとも 150 MB のハードディスクの空き容量が必要です。

注記

通常、古いハードウェアでは、 速いプロセッサを入手するより、より多くの RAM やより多くの容量のハードドライブを用意するほうが重要となります。

3.2.1.2. FreeBSD/amd64

FreeBSD/amd64 を実行可能なプロセッサには 2 つのクラスがあります。1 つ目のクラスは、AMD Athlon™64, AMD Athlon™64-FX, AMD Opteron™ やそれより新しいプロセッサです。

もう 1 つのクラスは、Intel® EM64T アーキテクチャを採用しているプロセッサです。これらのプロセッサの例は、Intel® Core™ 2 Duo, Quad, Extreme プロセッサファミリおよび Intel® Xeon™ 3000, 5000, および 7000 系のプロセッサです。

nVidia nForce3 Pro-150 ベースのコンピュータを使用するには、BIOS のセットアップを使って、IO APIC を無効にする 必要があります。もし、このようなオプションがなければ、かわりに ACPI を無効にしてください。Pro-150 チップセットには回避策が見つかっていないバグがあるためです。

3.2.1.3. FreeBSD/sparc64

FreeBSD/sparc64 をインストールするには、サポートされているプラットフォームが必要です (「サポートされているハードウェア」をご覧ください)。

現時点では、他のオペレーティングシステムとディスクの共有ができないので、FreeBSD/sparc64 専用のディスクが必要です。

3.2.2. サポートされているハードウェア

サポートされるハードウェアのリストは FreeBSD のリリースと一緒に FreeBSD ハードウェアノートとして提供されます。このドキュメントは通常 CDROM や FTP 配布の一番上のディレクトリや sysinstall(8) のドキュメントメニューにある `HARDWARE.TXT` という名のファイルで見ることができます。 アーキテクチャごとに用意されるそのリストを見ることで、FreeBSD の各リリースでどのようなハードウェア装置がサポートされるかがわかります。リリースごと、およびアーキテクチャごとのハードウェアリストは、FreeBSD のウェブサイトの リリース情報 のページにあります。

3.3. インストール前に行う作業

3.3.1. コンピュータの部品一覧を作る

FreeBSD をインストールする前に、コンピュータで使用している部品の一覧を作っておくといいでしょう。 FreeBSD のインストールルーチンでは、ハードディスク、ネットワークカード、CDROM ドライブなどの部品を、モデル番号、メーカー名とともに表示します。さらにこれらのデバイスについて、使用する IRQ, I/O ポートまで含めて正しい設定を認識しようと試みます。しかし、コンピュータのハードウェアによっては、 この過程が完全に成功するとは限らず、手動で修正しなければならないこともあります。

既に別のオペレーティングシステムがインストールされているのであれば、それらのオペレーティングシステムでのハードウェアの設定を参考にすると良いでしょう。 拡張カードなどの設定がよく分からない場合は、 カードそのものに印刷されていることもあります。よく使われる IRQ は 3, 5, 7 で、I/O ポートアドレスは通常 `0x330` のような 16 進数で書かれています。

FreeBSD をインストールする前に、これらの情報を印刷するか書き留めておくことを勧めます。例えば次のような表です:

表3.1 サンプルデバイス一覧

デバイスの名前	IRQ	I/O ポート	メモ
1 台目のハードディスク	N/A	N/A	40 GB, Seagate 製、プライマリ IDE マスタ
CDROM	N/A	N/A	プライマリ IDE スレーブ
2 台目のハードディスク	N/A	N/A	20 GB, IBM 製、セカンダリ IDE マスタ

デバイスの名前	IRQ	I/O ポート	メモ
1 つ目の IDE コントローラ	14	0x1f0	
ネットワークカード	N/A	N/A	Intel® 10/100
モデム	N/A	N/A	3Com® 56K ファックスモデム、COM1 に接続
…			

コンピュータで使用している部品の一覧を作成したら、インストールする FreeBSD のリリースのハードウェア要件を満たしているかどうかを確認してください。

3.3.2. データのバックアップ

コンピュータに価値のあるデータが入っている場合には、 FreeBSD をインストールする前に、データのバックアップをとり、 さらに正しくバックアップがとれていることを確認して下さい。FreeBSD のインストーラは、ハードディスクに実際に書き込む前に確認を求めますが、一度実際に書き込む作業が始まってしまうと、もう元に戻すことはできません。

3.3.3. FreeBSD をインストールする場所の決定

ハードディスク全体に FreeBSD をインストールする場合には、この節は読み飛ばしてください。

しかし FreeBSD を他のオペレーティングシステムと共存させる必要がある場合は、データがディスクにどのように配置されるかを大まかに理解する必要があります。

3.3.3.1. FreeBSD/i386 アーキテクチャでのディスクレイアウト

PC のディスクは、 パーティション と呼ばれる、 個別の塊に分割することができます。FreeBSD は内部にもパーティションを持つので、この名称は混乱をおこします。 そのため、FreeBSD ではこの塊をスライスと呼びます。 たとえば、fdisk(8) の FreeBSD 版は、 パーティションではなくスライスと呼びます。 PC では、ディスク 1 台あたり 4 つまでパーティションを作成できます。これらのパーティションはプライマリパーティションと呼ばれます。この制限を回避し、 4 つ以上のパーティションを作成するために拡張パーティションと呼ばれる新しいパーティション形式が作られました。 ディスク 1 台につき 1 つだけ拡張パーティションを作成できます。論理パーティションと呼ばれる特別のパーティションは、この拡張パーティションの内部に作成できます。

それぞれのパーティションは、そのパーティションに含まれるデータの形式を示す数字である、パーティション ID を持ちます。FreeBSD パーティションのパーティション ID は 165です。

通常、各オペレーティングシステムは、それぞれ独自の方法で個々のパーティションを識別します。たとえば Windows® は、プライマリパーティションおよび論理パーティションそれぞれに `C:` から始まる ドライブレターを割り当てます。

FreeBSD はプライマリパーティションにインストールしなければなりません。複数のディスクがある場合は、そのうちのいくつか、もしくはすべてに、FreeBSD パーティションを作成することもできます。FreeBSD をインストールする時は 少なくとも 1 つのパーティションを使用可能にしておかなければなりません。このパーティションは、空白のパーティション、もしくは無くなっても構わないデータしか入っていない、既存のパーティションでも構いません。

すべてのディスクのすべてのパーティションが使用されている場合は、他のオペレーティングシステムで提供されているツール (たとえば Windows® の `fdisk`) を使用して、そのうちの 1 つを FreeBSD のために解放してください。

予備のパーティションがある場合には、それを使ってください。もし、利用できる容量が少なければ、1 つまたは複数の既存のパーティションを縮小して、利用できる容量を確保してください。

FreeBSD の最小のインストールには、100 MB 程度のディスク容量が必要です。しかしこれは非常に小さなインストールであり、 自分のファイルを作成するスペースはほとんど残らないでしょう。現実的には、グラフィカルな環境が必要ないのであれば 250 MB、グラフィカルユーザインタフェースを使用するのであれば 350 MB 以上は必要でしょう。その上、サードパーティ製ソフトウェアをインストールするならば、さらに多くのスペースが必要になるでしょう。

FreeBSD のためのスペースを用意する際、 パーティションサイズを変更するために、GParted などのツールを使用できます。GParted は、NTFS を操作することができ、SystemRescueCD といった多くの Live CD Linux ディストリビューションで利用できます。

警告

パーティションサイズを縮小するツールの使用方法を誤ると、 ディスク上のデータが消えてしまう可能性があります。 この種のツールを使う際には、 使用前に必ず最新のデータのバックアップをとって下さい。

例3.1 既存のパーティションを変更せずに使用

既に Windows® がインストールされている 4 GB のハードディスクが 1 台接続されており、そのハードディスクは 2 つのドライブレター `C:` および `D:` に分割さ

れており、それぞれのサイズが 2 GB であるとします。`C:` には 1 GB, `D:` には 0.5 GB のデータがあるとします。

ディスクには、ドライブレター 1 つあたり 1 つのパーティション、合計で 2 つのパーティションがあることを意味します。この場合、`D:` にあるデータをすべて `C:` にコピーすれば、2 つ目のパーティションを解放し、FreeBSD のために使うことができるでしょう。

例3.2 既存のパーティションを縮小する

既に Windows® がインストールされている 4 GB のハードディスクが 1 台接続された PC を使用しているとします。さらに Windows® をインストールする際、1 つの大きなパーティションを作成し、`C:` ドライブとして 4 GB を割り当てたとします。現在 1.5 GB 使用しており、FreeBSD で 2 GB 使いたいとします。

FreeBSD をインストールするためには、以下のどちらかを行わなければなりません。

1. Windows® のデータをバックアップし、インストール時に 2 GB のパーティションを作成して Windows® を再インストールする。

2. 先に述べた Windows® パーティションを縮小するツールを使用する。

3.3.4. ネットワーク設定の詳細をまとめる

FTP サイトまたは NFS サーバからインストールする場合は、ネットワークの設定を知る必要があります。インストールを完了するためにネットワークに接続できるよう、これらの情報をインストーラのプロンプトに入力する必要があります。

3.3.4.1. イーサネットもしくはケーブル / DSL モデムでの接続

イーサネットネットワークに接続する場合や、ケーブル / DSL モデム経由でイーサネットアダプタを利用してインターネットに接続する場合は、次の情報が必要になります:

1. IP アドレス

2. デフォルトゲートウェイの IP アドレス

3. ホスト名

4. DNS サーバの IP アドレス

5. サブネットマスク

これらの情報がわからなければ、システム管理者かサービスプロバイダに問い合わせて下さい。DHCP を使用して、この情報を自動的に割り当てることもあります。

3.3.4.2. モデムを使用した接続

非常に長い時間がかかりますが、ダイアルアップモデムを使って、インターネット経由で FreeBSD をインストールできます。

この場合、以下の内容をあらかじめ確認しておく必要があります:

1. インターネットサービスプロバイダ (ISP) にダイアルする際の電話番号

2. 接続に使用する COM: ポート

3. ISP のアカウントのユーザ名及びパスワード

3.3.5. FreeBSD Errata の確認

FreeBSD プロジェクトでは FreeBSD の各リリースができる限り安定するよう努力していますが、時々バグが発生してしまうことがあります。極まれに、発生したバグによりインストールプロセスに影響を与えることがあります。これらの問題は発見され解決されるとともに、FreeBSD のウェブサイトの FreeBSD Errata に掲示されます。注意すべき既知の問題が無いことを確かめるために、インストールする前に Errata を確認してください。

Errata を含む、すべてのリリースに関する情報は、FreeBSD のウェブサイト の リリース情報 の項で確認することができます。

3.3.6. FreeBSD インストールファイルの入手

FreeBSD のインストーラは、以下のいずれかの場所に置いてあるファイルから FreeBSD をインストールします。

- CDROM または DVD
- USB メモリスティック
- 同じコンピュータ上の MS-DOS® パーティション
- フロッピーディスク (FreeBSD/pc98 のみ)
- ファイアウォール経由または HTTP プロキシを利用した FTP サイト

- NFS サーバ

- パラレルまたはシリアル専用接続

購入した CD または DVD から FreeBSD をインストールするのであれば、「ブートメディアの準備」を読み飛ばしてください。

FreeBSD のインストールファイルを入手するには、「カスタムインストールメディアの準備」まで読み飛ばしてください。 インストールメディアをどのように準備すれば良いか説明しています。このセクションを読み終わった後、ここに戻ってきて、「ブートメディアの準備」を読んでください。

3.3.7. ブートメディアの準備

FreeBSD のインストールプロセスは、FreeBSD インストーラでコンピュータを起動することから始まります。 インストーラは、 別のオペレーティングシステムで実行するプログラムではありません。 通常、コンピュータはハードディスクにインストールされたオペレーティングシステムから起動しますが、CDROM や USB ディスクから起動するように設定することもできます。

ヒント

CDROM および DVD からの起動が可能な BIOS を持つコンピュータに CDROM または DVD をインストールするのであれば、この節を飛ばしてください。FreeBSD の CDROM と DVD イメージは起動可能であり、 他の特別な準備をすることなく FreeBSD のインストールで利用できます。

起動可能なメモリスティックを作成する場合には、以下の手順にしたがってください。

1. メモリスティックのイメージの取得

   FreeBSD 8.*X* のメモリスティックのイメージは、`ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/`*`arch`*`/ISO-IMAGES/`*`version`*`/FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-memstick.img` の `ISO-IMAGES/` ディレクトリからダウンロードできます。 ここで、*`arch`* と *`version`* の部分を、 それぞれインストールするアーキテクチャとバージョン番号に置き換えてください。 たとえば、FreeBSD/i386 9.3-RELEASE のメモリスティックのイメージは、`ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/ISO-IMAGES/9.3/FreeBSD-9.3-RELEASE-i386-memstick.img` から入手できます。

ヒント

FreeBSD 9.0-RELEASE からは、異なるディレクトリパスが使われています。FreeBSD 9.X のダウンロードとインストールの詳細については、2章FreeBSD のインストール (9.X 以降の場合) で説明します。

メモリスティックイメージには、`.img` という拡張子がついています。`ISO-IMAGES/` ディレクトリには複数の異なるイメージがあり、 用いるべきイメージは、FreeBSD のバージョンやインストールするハードウェアが対応しているメディアに依存します。

重要

以下の作業によってデータが消去されるので、 先に進む前に、使用する USB スティックにあるデータをバックアップしてください。

2. イメージファイルをメモリスティックに書き込む

手順3.1 FreeBSD を使ってイメージを書き込む

警告

以下の例では、起動するデバイスを `/dev/da0` としています。 適切なデバイスを出力先に設定していることを十分確認してくだい。 さもなければ、現在あるデータを破壊してしまうでしょう。

- イメージを dd(1) を使って書き込む

  `.img` ファイルは、単純にメモリスティックにコピーされるような、通常のファイルではありません。ディスクの内容の完全なイメージです。したがって、dd(1) を使用して、直接ディスクにイメージを書き込む必要があります。

```
# dd if=FreeBSD-9.3-RELEASE-i386-memstick.img of=/
dev/da0 bs=64k
```

`Operation not permitted` エラーが表示されたら、ターゲットデバイスが他で利用されていないかどうか、マウントされていないかどうか、他のプログラムが自動的にマウントしていないかどうかを確認してください。その後、もう一度実行してください。

手順3.2 Windows® を使ってイメージを書き込む

警告

以下のコマンドは、 指定したデバイスに現在あるデータを上書きして破壊してしまうので、 出力先に適切なドライブレターを設定していることを十分確認してくだい。

1. Image Writer for Windows を入手する

   Image Writer for Windows は、 イメージファイルをメモリスティックに正しく書き込むことのできるフリーのアプリケーションです。 `https://launchpad.net/win32-image-writer/` からダウンロードして、フォルダに展開してください。

2. イメージライタを使ってイメージを書き込む

   Win32DiskImager アイコンをダブルクリックして、プログラムを起動します。`Device` の下に表示されるデバイスレターが、 メモリスティックのドライブであることを確認してください。フォルダのアイコンをクリックして、メモリスティックに書き込むイメージファイルを選択します。 Save をクリックして、 イメージファイルの名前をアクセプトしてください。すべてが正しく行われたかどうか、また、他のウィンドウでメモリスティックのフォルダが開かれていないことを確認してください。 最後に、Write を押して、ドライブにイメージファイルを書き込みます。

FreeBSD/pc98 のインストール用に起動フロッピーディスクを作成するためには、次のステップに従って下さい:

1. 起動フロッピーイメージの取得

   FreeBSD/pc98 のブートディスクは、 floppies ディレクトリ、 `ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/pc98/` *version* -

RELEASE/floppies/ . から入手できます。ここで *version* の部分を、それぞれインストールするバージョン番号に置き換えてください。

フロッピーイメージは .flp という拡張子がついています。floppies/ には複数の異なるイメージがあります。 boot.flp とともに、 kern.small* または kern* といったインストールの種類に関連づいた複数のファイルをダウンロードしてください。

重要

FTP プログラムを使用してこれらのディスクイメージをダウンロードする時は、必ずバイナリモードにして下さい。Web ブラウザによっては テキスト または アスキー モードでダウンロードするものがあり、ディスクから起動できないときは大抵これが原因です。

2. フロッピーディスクの用意

ダウンロードしたイメージファイル 1 つにつき 1 枚のフロッピーディスクを用意してください。これらのディスクに欠陥があってはいけません。これを確認する最も簡単な方法は、自分自身でフォーマットしてみることです。フォーマットする前のフロッピーを信用してはいけません。Windows® のフォーマットユーティリティは、不良ディスクがあっても教えてはくれないでしょう。それらを「bad」とマークして、無視するだけです。新品のフロッピーを使うことをお薦めします。

重要

インストーラがクラッシュしたりフリーズしたり、 おかしな動作をした時、 まずはじめに疑うべきもののうちの 1 つはフロッピーです。フロッピーイメージを新しいディスクに書き込んで、もう一度試してみて下さい。

3. フロッピーディスクへイメージファイルを書き込む。

.flp ファイルは、 ディスクにコピーできるような、 通常のファイルではありません。完全なディスクの内容のイメージです。イメージを直接ディスクに書き込む特別なツールを使用する必要があります。

FreeBSD は、Windows® が動作しているコンピュータでフロッピーを作成する `rawrite` と呼ばれるツールを提供しています。 このツールは FreeBSD の FTP サイト上の `ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/pc98/` *`version`*`-RELEASE/tools/` からダウンロードできます。このツールをダウンロードしたら、フロッピーを挿入してください。 その後、ファイル名を指定してフロッピードライブに書き込んでください。

```
C:\> rawrite boot.flp A:
```

毎回フロッピーディスクを入れ換え、ファイルの名前を示すラベルを付けながら、それぞれの `.flp` ファイルに対してこのコマンドを繰り返します。`.flp` ファイルを置いた場所に応じて、コマンドラインを変更して下さい。

FreeBSD システムのような、UNIX®-like システムでフロッピーへの書き込みを行う場合は、イメージファイルを直接ディスクに書き込むために dd(1) を使ってください。FreeBSD 上では、次のように実行してください。

```
# dd if=boot.flp of=/dev/fd0
```

FreeBSD においては、`/dev/fd0` が 1 台目のフロッピーディスクドライブを表します。他の UNIX® システムでは、フロッピーディスクデバイスには別の名前がついているかもしれないので、 必要に応じてそのシステムのドキュメントを確認して下さい。

これで FreeBSD をインストールする用意ができました。

3.4. インストールの開始

重要

デフォルトでは、 次のメッセージが表示されるまでインストーラはディスクに何の変更も加えません。

```
Last Chance: Are you SURE you want continue the
 installation?

If you're running this on a disk with data you wish
 to save then WE
STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before
 proceeding!

We can take no responsibility for lost disk
 contents!
```

この最後の警告の前であれば、 いつでもハードディスクの内容を変更することなくインストールを中断できます。 もし、何かを間違って設定してしまったことが心配ならば、最後の警告の前に単にコンピュータをオフにしてください。被害を残さずに済みます。

3.4.1. 起動

3.4.1.1. i386™ アーキテクチャでの起動

1. コンピュータの電源をいれます。 起動が始まると、すぐにシステムまたは BIOS のセットアップメニューに入るためのオプションが表示されるはずです。 一般的には F2, F10, Del のようなキーまたは Alt+S のようなキーの組合せです。画面上に示されるキーを使用してください。コンピュータによっては、起動する間に画像を表示するものもあります。大抵の場合、Esc キーを押すと、画像が消えてブートメッセージが表示されます。

2. どの装置からシステムが起動するかを制御する設定を探してください。 通常は「Boot Order」と呼ばれ、一般的には、`Floppy`, `CDROM`, `First Hard Disk` などの装置の一覧として表示されています。

   CD または DVD から起動する場合には CDROM が選択されていることを、USB ディスクから起動するなら USB ディスクが選択されていることを確認してください。自信が無い場合は、 コンピュータやマザーボード付属のマニュアルを参照してください。

   変更を保存して終了してください。コンピュータが再起動するはずです。

3. 「ブートメディアの準備」で説明されている「起動」USB スティックを使用する場合には、コンピュータを立ち上げる前に、USB スティックを挿入してください。

   CD または DVD から起動する場合、コンピュータを立ち上げ、すぐに CD または DVD を挿入する必要があります。

注記

FreeBSD/pc98 では、 インストール用の起動フロッピーが用意されており、「ブートメディアの準備」 で説明されている方法で作成できます。 `boot.flp` が最初に挿入するフロッピーディスクです。 インストーラを起動できるように、 このフロッピーをフロッピードライブに挿入してください。

もし、コンピュータがすでに存在している OS を読み込み、通常通り起動してしまったのであれば、以下の原因が考えられます。

1. 起動ディスクが起動プロセスにおいて十分早いタイミングで挿入されていません。ディスクをそのままにしてコンピュータを再起動してください。

2. BIOS の変更が適切に行われていません。正しいオプションを設定してやり直してください。

3. 使用している BIOS は、希望しているメディアからの起動に対応していません。

4. FreeBSD は起動を開始します。CD または DVD から起動している場合、次のような画面が表示されるでしょう。

```
Booting from CD-Rom...
645MB medium detected
CD Loader 1.2

Building the boot loader arguments
Looking up /BOOT/LOADER... Found
Relocating the loader and the BTX
Starting the BTX loader

BTX loader 1.00 BTX version is 1.02
Consoles: internal video/keyboard
BIOS CD is cd0
BIOS drive C: is disk0
BIOS drive D: is disk1
BIOS 636kB/261056kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/boot/kernel/kernel text=0x64daa0 data=0xa4e80+0xa9e40 ↺
syms=[0x4+0x6cac0+0x4+0x88e9d]
\
```

フロッピーディスクから起動している場合、次のような画面が現れるでしょう。

```
Booting from Floppy...
Uncompressing ... done

BTX loader 1.00  BTX version is 1.01
Console: internal video/keyboard
BIOS drive A: is disk0
BIOS drive C: is disk1
BIOS 639kB/261120kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/kernel text=0x277391 data=0x3268c+0x332a8 |
```

```
Insert disk labelled "Kernel floppy 1" and press any key...
```

`boot.flp` フロッピーを抜き、次のフロッピーを挿入し、Enter を押してください。プロンプトが表示されたら、要求される通りに他のディスクを挿入してください。

5. 起動プロセスは FreeBSD ブートローダメニューを表示します。

図3.1 FreeBSD ブートローダメニュー

10 秒間待つか、Enter を押してください。

3.4.1.2. SPARC64® アーキテクチャでの起動

ほとんどの SPARC64® システムは、ディスクから自動的に起動するように設定されています。FreeBSD をインストールするには、ネットワークまたは CD/DVD から起動し、ブートメッセージが表示されるまで待ってください。モデルによりますが、以下のようなメッセージが表示されます。

```
Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
Copyright 1998-2001 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.2, 128 MB memory installed, Serial #51090132.
Ethernet address 0:3:ba:b:92:d4, Host ID: 830b92d4.
```

もしシステムがディスクから起動するようでしたら、キーボードから L1+A または Stop+A を押すか、シリアルコンソールから tip(1) または cu(1) で **~#** を使って **BREAK** を送信して、PROM プロンプトを表示してください。次のような表示です。

```
ok     ❶
ok {0} ❷
```

❶ 単一の CPU を持つシステムでのプロンプト
❷ SMP システムのプロンプト。数字はアクティブな CPU の数。

ここで、CD または DVD をドライブに挿入し、PROM プロンプトで `boot cdrom` と入力してください。

3.4.2. デバイス検出結果を再表示する

画面に表示される数百行の文字列は (バッファに) 記憶されており、再表示することが出来ます。

バッファを再表示するには、Scroll Lock を押してください。これで、画面をスクロールできます。結果を見るためには、矢印キーもしくは PageUp か PageDown を使います。Scroll Lock をもう一度押すと、スクロールを停止します。

この作業を行って、カーネルがデバイス検出を実行している時に画面の外に流れた文字列を再表示してください。図3.2「古典的なデバイス検出結果の例」によく似た画面が現われるでしょうが、 コンピュータに搭載されているデバイスによって、表示される文字列は異なります。

```
avail memory = 253050880 (247120K bytes)
Preloaded elf kernel "kernel" at 0xc0817000.
Preloaded mfs_root "/mfsroot" at 0xc0817084.
md0: Preloaded image </mfsroot> 4423680 bytes at 0xc03ddcd4

md1: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1:<VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device ↺
1.0 on pci0
pci1: <PCI bus> on pcib1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <iSA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device ↺
7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
uhci0 <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at ↺
device 7.2 on pci
0
usb0: <VIA 83572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr1
uhub0: 2 ports with 2 removable, self powered
pci0: <unknown card> (vendor=0x1106, dev=0x3040) at 7.3
dc0: <ADMtek AN985 10/100BaseTX> port 0xe800-0xe8ff mem ↺
0xdb000000-0xeb0003ff ir
q 11 at device 8.0 on pci0
dc0: Ethernet address: 00:04:5a:74:6b:b5
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xec00-0xec1f irq 9 ↺
at device 10.
0 on pci0
ed0 address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
orm0: <Option ROM> at iomem 0xc0000-0xc7fff on isa0
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq2 on ↺
isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5” drive> on fdc0 drive 0
atkbdc0: <Keyboard controller (i8042)> at port 0x60,0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq1 on atkbdc0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: model Generic PS/@ mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff ↺
on isa0
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
sio1: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
pppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
```

図3.2 古典的なデバイス検出結果の例

丹念に検出結果をチェックして、すべてのデバイスを FreeBSD が検出できた事を確認してください。デバイスが見つからなかった場合、検出結果へ表示されません。カスタムカーネル を構築することで、GENERIC カーネルに含まれていないデバイスに対応できるようになります。

デバイス検出後、図3.3「国の選択メニュー」に示されるメニューが表示されます。矢印キーを使って、国、地域またはグループを選択し、その後、Enter を押して決定してください。

図3.3 国の選択メニュー

国に United States を選択した場合には、標準のアメリカのキーボードマップが使われるようになります。他の国を選択すると以下のメニューが表示されます。矢印キーを使い、適切なキーボードマップを選択して Enter を押してください。

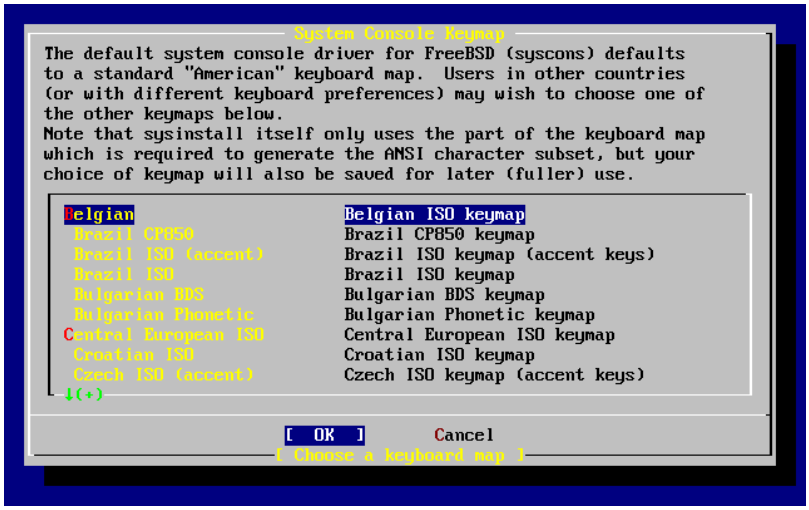


図3.4 キーボードメニューの選択

国の設定が終わったら、sysinstall(8) のメインメニューが表示されます。

3.5. sysinstall(8) の紹介

FreeBSD 8.X のインストーラである sysinstall(8) は、コンソール上で動作し、いくつものメニューや画面から構成され、設定やインストールプロセスをコントロールできます。

メニューはカーソルキー、Enter, Tab, Space 等で操作します。これらのキーの詳しい解説とその意味については、図3.5「Sysinstall のメインメニューから Usage を選択」に示されているように、Usage の項目が強調表示され、[Select] ボタンが選択されていることを確認してから、Enter を押してください。

メニューシステムの使い方が表示されます。読み終ったら、 Enter を押してメインメニューに戻りましょう。

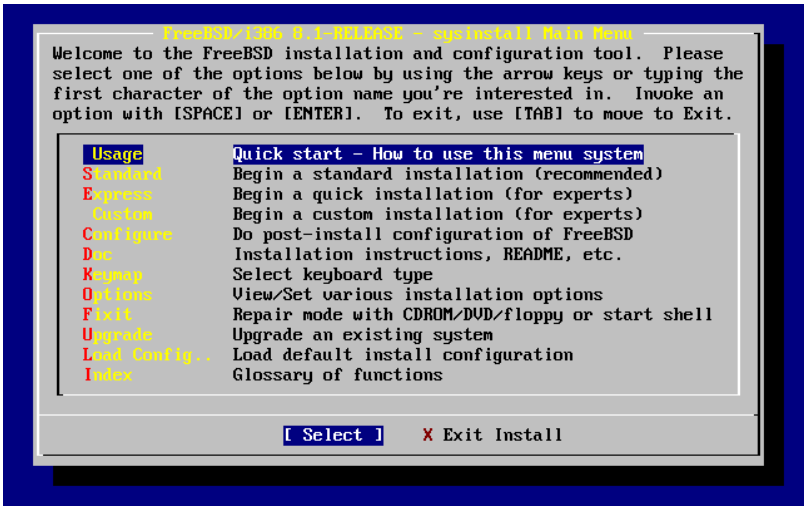


図3.5 Sysinstall のメインメニューから Usage を選択

3.5.1. ドキュメントメニューの選択

メインメニューから、カーソルキーで Doc を選択して Enter を押してください。

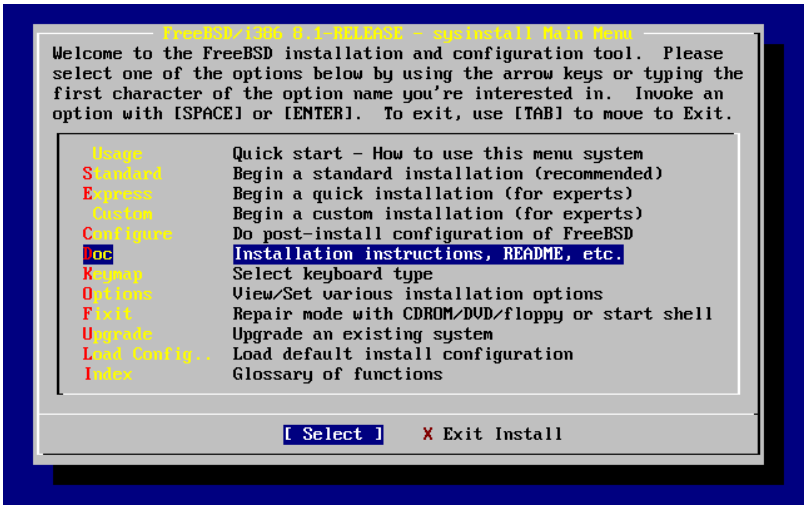


図3.6 ドキュメントメニューの選択

これでドキュメントメニューが表示されます。

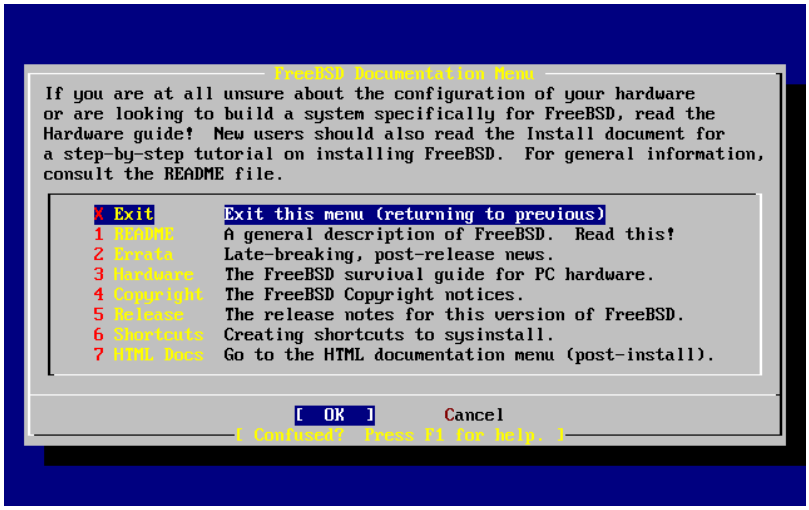


図3.7 Sysinstall ドキュメントメニュー

付属のドキュメントを読むことは重要です。 ドキュメントを読むには、カーソルキーで選択して Enter を押してください。読み終えたら、Enter を押すとドキュメントのメニューに戻ります。

メインのインストールメニューに戻るには、カーソルキーで Exit を選択して、Enter を押してください。

3.5.2. キー配列メニューの選択

キーボード配列を変更するには、カーソルキーで Keymap を選択して、Enter を押してください。使用しているキーボードが、標準ではなかったり、または US ではない場合に変更作業が必要となります。

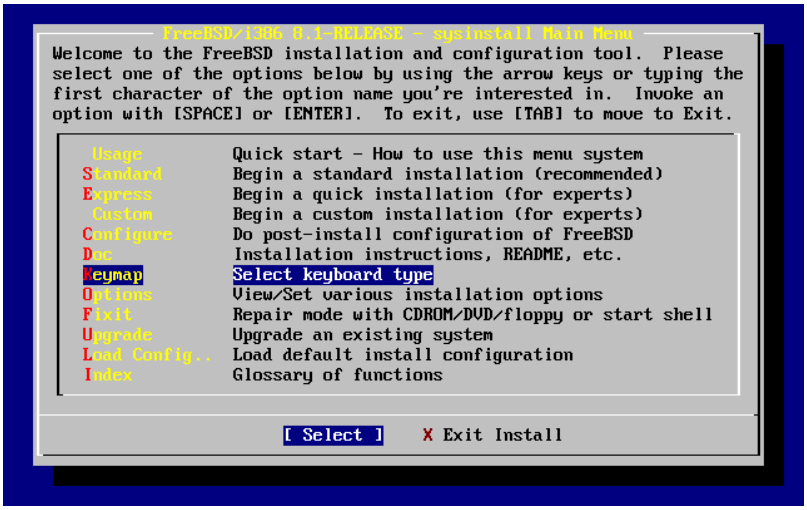


図3.8 Sysinstall メインメニュー

上下のカーソルキーでメニュー項目を選択して Space を押すと、異なるキーボード配列を選択できます。もう一度 Space を押すと、選択を解除します。完了したら、カーソルキーで [OK] を選択して Enter を押してください。

このスクリーン表示では、リストの一部しか表示されていません。 Tab を押して [Cancel] を選択すると、デフォルトのキーマップを使用して、メインインストールメニューに戻ります。

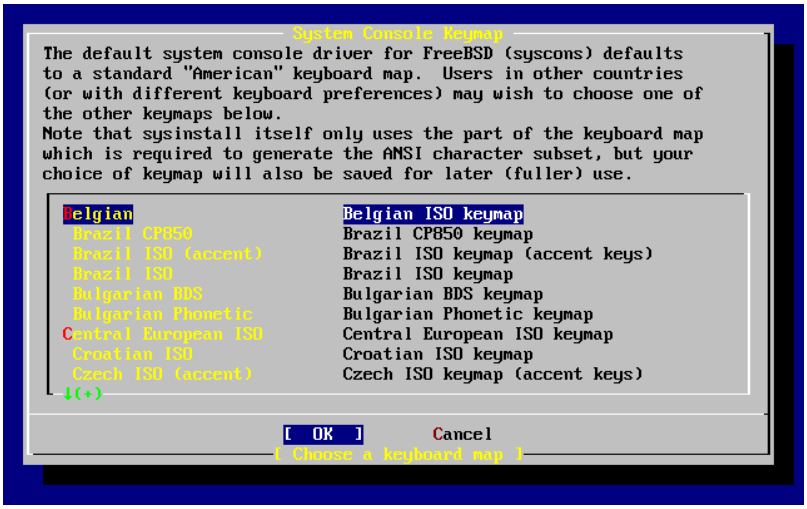


図3.9 Sysinstall キー配列メニュー

3.5.3. インストールオプション画面

Options を選択して、Enter を押してください。

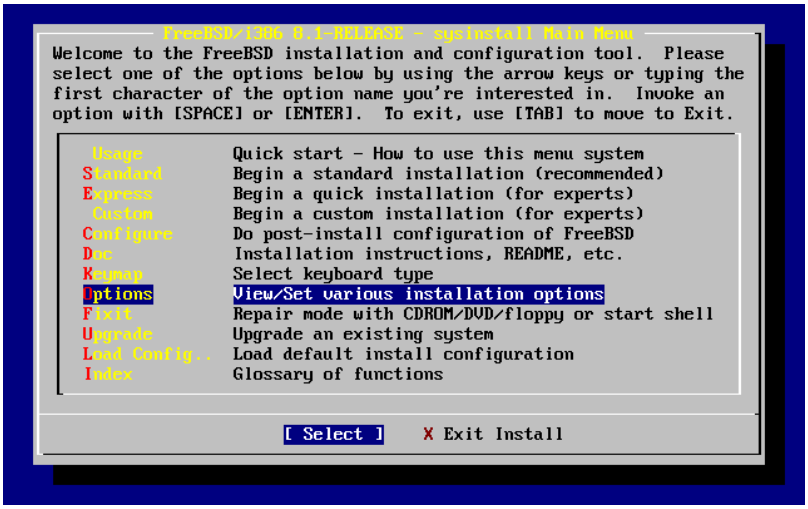


図3.10 Sysinstall メインメニュー

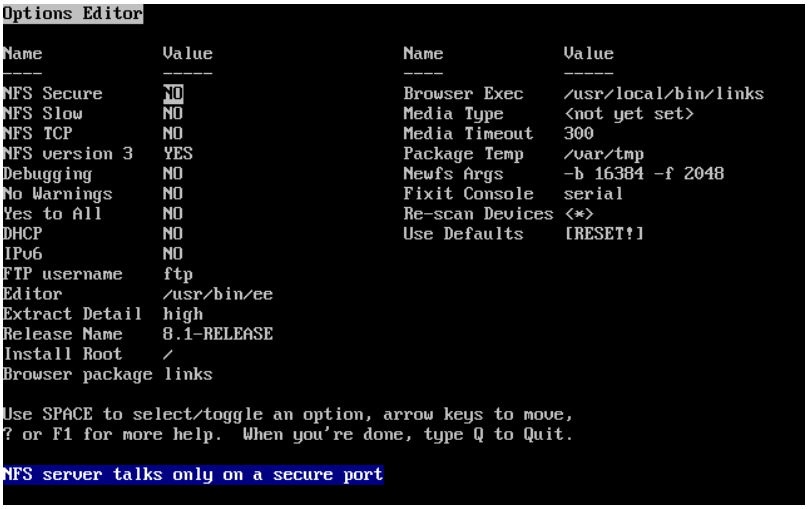


図3.11 Sysinstall オプション

ほとんどのユーザーにはデフォルトの値でよく、変更する必要はありません。 リリース名は、インストールするバージョンによって変わります。

選択された項目の説明が、画面の下部に青く強調表示されます。オプションの 1 つは Use Defaults という、すべての値を初期値に戻すものであることに気をつけてください。

F1 を押すと、さまざまなオプションについての説明を読むことができます。

Q を押すと、メインインストールメニューに戻ります。

3.5.4. 標準インストールの開始

UNIX® や FreeBSD が初めての人には Standard インストールがお奨めです。カーソルキーで Standard を選択し、Enter を押してインストールを始めてください。

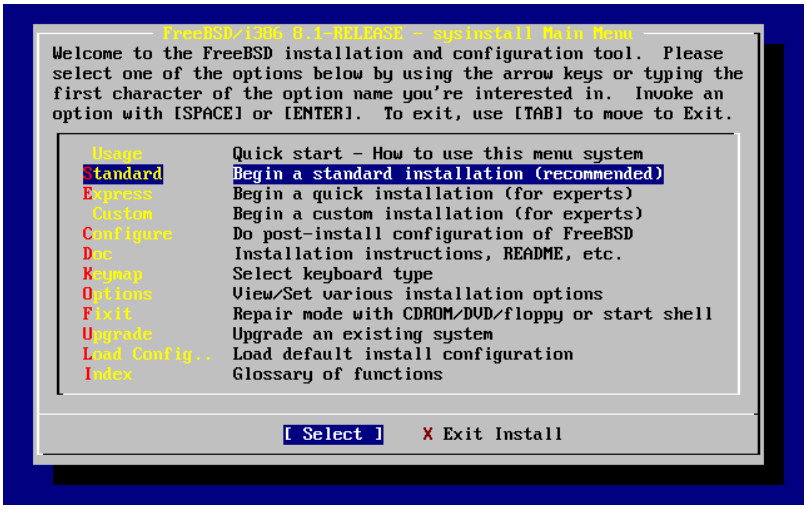


図3.12 標準インストールの開始

3.6. ディスク領域の割り当て

最初の課題は、FreeBSD 用にディスク領域を割り当て、sysinstall(8) が使えるようにラベルを作成することです。そうするためには、FreeBSD がディスク上にどのような情報があることを前提としているかを知らなければなりません。

3.6.1. BIOS ドライブの番号付け

FreeBSD をインストールして設定する前に、FreeBSD がどのように BIOS ドライブマッピングを取り扱うかを理解しておくことが重要です。

Microsoft® Windows® のような に依存したオペレーティングシステムが動いている PC では、 BIOS がディスクドライブの順序を構成でき、 OS はその変化に追従します。これにより、"プライマリマスタ" 以外のディスクから起動することができます。 この仕組みを用いればシステムのバックアップを取る簡単な方法を構築できます。もう一つ同じディスクを買い、一つ目のディスクから二つめのディスクへのコピーを定期的に取ればいいのです。 そして、一つ目のディスクに障害が起きたり、ウィルスに感染したり、 オペレー

ティングシステムの不具合でめちゃくちゃにされてしまった時には、BIOS に対してドライブを論理的に交換するように指示することで簡単に復旧できるのです。 この方法はドライブのケーブルを交換するのと同じようなことなのですが、 ケースを開ける必要がありません。

SCSI コントローラを備えたシステムでは、しばしば BIOS に拡張が施されており、同じように 7 台までのドライブの順番を組み換えることができるようになっています。

以上のような機能を便利に使っているユーザは、FreeBSD では同じような結果にならないことに驚くかもしれません。FreeBSD は BIOS を利用しないため、「論理 BIOS ドライブマッピング」については知らないのです。このため、特に同じ物理ジオメトリを持っているいくつかのドライブをデータのクローンとして使っている時に、ややこしい状況になり得ます。

FreeBSD を使う時は、インストール前にドライブの番号付けが自然なものになるように、必ず BIOS の設定を戻しておきましょう。もしドライブの番号付けを変更する必要がある場合には、ケースを開けジャンパーやケーブルを移動してください。

Bill と Fred のイケイケ冒険記より

Bill は Fred のためにもう一つ FreeBSD 箱を作ろうと古い Wintel 箱を潰しました。Bill は ユニット番号 0 の SCSI ドライ ブを一つ追加し、そこに FreeBSD を入れました。

Fred はこのシステムを使い始めましたが、数日後その古い SCSI ドライブがたくさんのエラーを吐いていることに気付きました。

その問題に対処するため、 Bill は同じ SCSI ドライブを取ってきました。 このドライブを SCSI ユニット番号 4 と して取り付け、ドライブ 0 からドライブ 4 へのイメージコピーを行ないました。新しいドライブがインストールされ、しかもうまく動いているため、Bill はそれを使い始めてもいいだろうと思いました。そこで彼は SCSI BIOS の機能を使ってシステムがユニット 4 から起動するようにディスクドライブの順序を入れ換えました。FreeBSD が起動し、調子良く動き始めました。

Fred は作業を続けましたが、すぐに 新しいバージョンの FreeBSD にアップグレードすることにしました。Bill は SCSI ユニット 0 のディスクは当てにならないので取りはずし、別の新しい同じドライブと交換しました。そして、Bill は新しいバージョンの FreeBSD を、新しい SCSI ユニット 0 にインストールしました。インストールはうまくいきました。

Fred は新しいバージョンの FreeBSD を数日使ってみて、技術部門でも使えるくらい十分に良いものだと確認しました。古いバージョンからすべての作業をコピーする時が来たのです。そこで Fred は、古い FreeBSD で行なっていた作業の最新のものが置いてあるはずの SCSI ユニット 4 をマウントしました。ところが、Fred は SCSI ユニット 4 には自分の作業がなにも残っていないことを発見して慌てふためきました。

Bill がオリジナルの SCSI ユニット 0 のイメージをユニット 4 にコピーした時に、ユニット 4 が「新クローン」になっていたのです。Bill がユニット 4 から起動するように SCSI BIOS で順序の入れ換えを行なった後でも、FreeBSD は依然として SCSI ユニット 0 上で動いていたのです。BIOS にこのような変更を行なっても Boot と Loader のコードの一部もしくは全部は選択された BIOS ドライブから取得されます。しかしながら、処理が FreeBSD のカーネルドライバに引き渡された時から BIOS ドライブの順序は無視され、FreeBSD は、通常のドライブ番号順に移行するのです。さきほどの例では、システムはオリジナルの SCSI ユニット 0 で動き続けており、Fred のデータはすべて SCSI ユニット 4 ではなくそのディスクに残っていたのです。システムが SCSI ユニット 4 で動いているように見えたのは単に人の期待からくる妄想だったのです。

幸運にも、古い SCSI ユニット 0 は見つけ出され、Fred が行なった作業はすべて彼のもとへ返ってきたのです。

この例では SCSI ドライブが用いられましたが、その概念は IDE ドライブにも同じように当てはまります。

3.6.2. fdisk でスライス作成

sysinstall(8) で Standard インストールを選択すると、次のメッセージが表示されるでしょう。

```
                                  Message
 In the next menu, you will need to set up a DOS-style ("fdisk")
 partitioning scheme for your hard disk. If you simply wish to devote
 all disk space to FreeBSD (overwriting anything else that might be ↺
on
 the disk(s) selected) then use the (A)ll command to select the ↺
default
 partitioning scheme followed by a (Q)uit. If you wish to allocate ↺
only
 free space to FreeBSD, move to a partition marked "unused" and use ↺
the
 (C)reate command.
                                 [  OK  -]

                       [ Press enter or space -]
```

Enter を押すと、カーネルがデバイス検出時に見付けた、すべてのハードドライブのリストが表示されます。図3.13「fdisk を実行するディスクの選択」は ad0 および ad2 と名付けられた 2 つの IDE ディスクをもつシステムの例です。

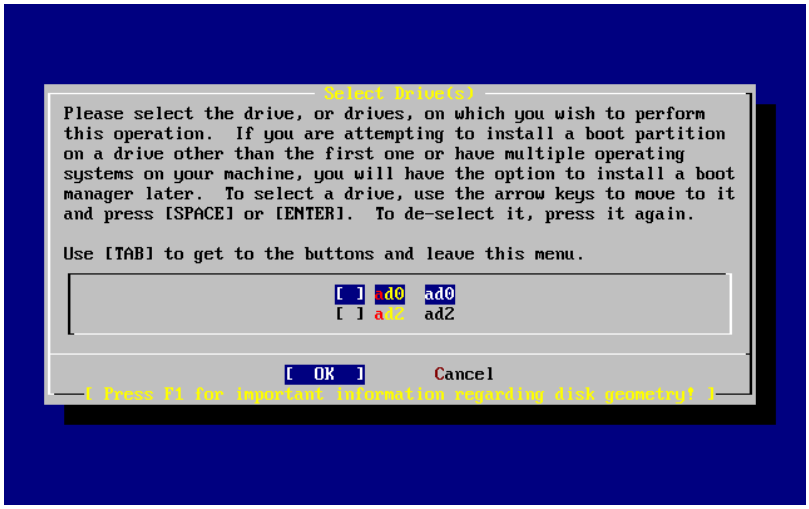


図3.13 fdisk を実行するディスクの選択

この画面では、一覧に ad1 が表示されていないことに注目してください。

一つ目はプライマリ IDE コントローラのマスタ、二つ目はセカンダリ IDE コントローラのマスタとして接続している 2 つの IDE ハードディスクを考えてください。もし FreeBSD がこれらを ad0 と ad1 のように番号つけても、すべては動作するでしょう。

しかし、その後プライマリ IDE コントローラのスレーブとして 3 つ目のディスクを追加したとしたら、それはたった今 ad1 となり、以前の ad1 は ad2 となるでしょう。デバイス名はファイルシステムを見つけるのに使われるため、ファイルシステムのいくつかは突然正しく現れなくなるかもしれず、FreeBSD の設定を変更する必要があるでしょう。

これを解決するために、ディスクが見つかった順番ではなく、 どこに接続されているかということに基づいて IDE ディスクを名前づけするようにカーネルを設定できます。この機構によって、セカンダリ IDE コントローラのマスタディスクは、たとえ ad0 または ad1 デバイスがないとしてもいつでも ad2 になるでしょう。

この設定は FreeBSD カーネルの標準設定です。これがこの例で画面に ad0 および ad2 が表示される理由です。このスクリーンショットが得られたマシンには、IDE コントローラの両方のマスタチャネルにディスクがあり、スレーブチャネルにはありません。

FreeBSD をインストールしたいディスクを選択して、 [OK] を押してください。 図 3.14「典型的な FDisk パーティション」のような表示とともに Fdisk が起動するでしょう。

Fdisk の画面は三つのセクションに分かれます。

一つ目のセクションは、これは表示の先頭二行にわたっているのですが、 現在選択されているディスクの詳細を表示します。 ディスクの詳細には FreeBSD でのデバイス名、ディスクのジオメトリ、そしてディスクの全容量が含まれます。

二つ目のセクションは現在ディスク上にあるスライスを表示します。 スライスの開始セクタと終了セクタ、大きさ、FreeBSD 上での名前、種類とサブタイプが表示されます。この例では、PC 上のディスクレイアウト機能が生み出した、 未使用の小さなスライスを二つ表示しています。また大きな FAT スライスも一つ表示しています。これは、Windows® において C: ドライブ、および拡張スライスとして現れます (この場合、他のドライブレターとなりえます)。

三つ目のセクションは FDisk において利用可能なコマンドを表示します。

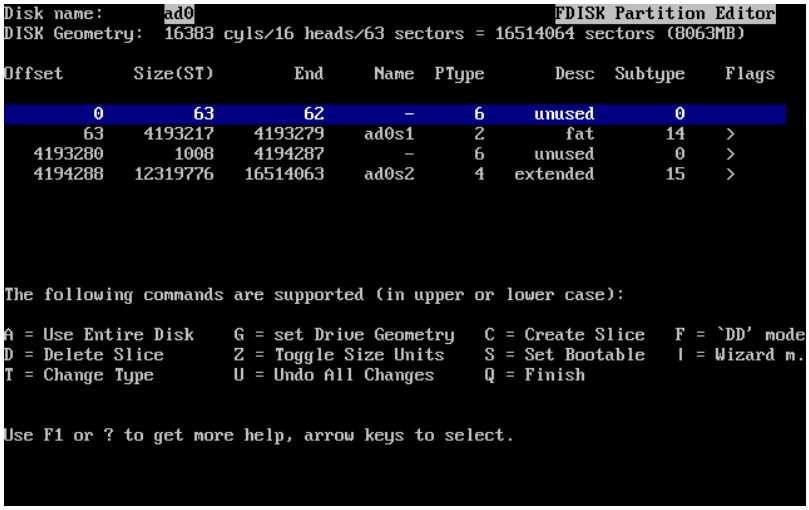


図3.14 典型的な FDisk パーティション

今から行うことは、どのようにディスクを分割するかによります。

FreeBSD をディスク全体にインストールするのであれば、Use Entire Disk オプションを表す A キーを押してください。 この場合、このディスクのすべてのデータは削除されます。 存在しているスライスは取り除かれ、 unused の小さな領域と、 FreeBSD のための大きなスライスへ置きかわるでしょう。 次に方向キーを使って新しく作成された FreeBSD スライスを選択し、スライスに起動可能の印をつけるために S キーを押してください。そのとき、画面の見た目は 図3.15「ディスク全体を使う FDisk 構成」とよく似たものとなるでしょう。Flags 列の A に注意してください。これはこのスライスが アクティブ で、ここから起動することを示します。

FreeBSD のための空き領域を作成するために、存在しているスライスを削除しようとしているのなら、方向キーをつかってスライスを選択して D キーを押してください。それから C キーを押すと、作成したいスライスの大きさの入力を促されます。適切な数字を入力して Enter キーを押してください。この欄に表示されているデフォルト値は、スライスに対して割り当てることのできる最大の値です。この値は、割り当てられていない領域の連続したブロック、または、ハードディスクの全サイズです。

FreeBSD のための空き領域を既に作成しているなら、 そのときは C キーを押して新しいスライスを作成できます。 再び、作成したいスライスの大きさの入力を促されるでしょう。

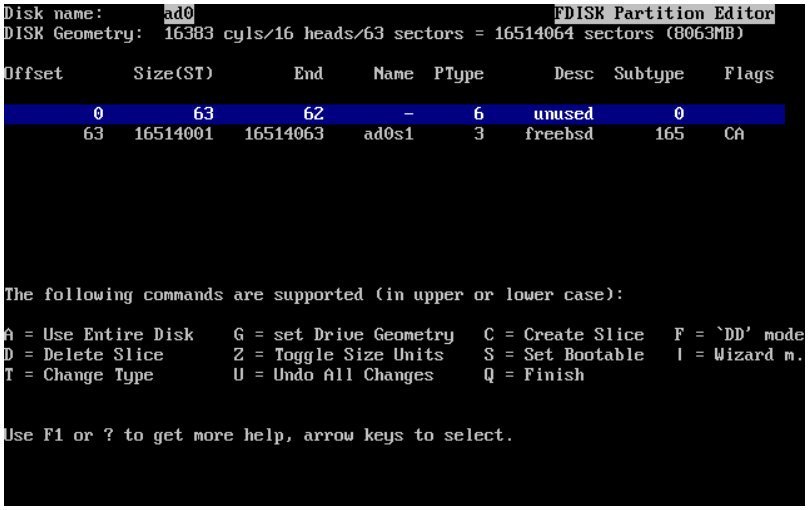


図3.15 ディスク全体を使う FDisk 構成

終了したら Q キーを押しください。あなたの変更は sysinstall(8) 内に保存されるでしょう。しかし、まだディスクには書きこまれません。

3.6.3. ブートマネージャのインストール

次のメニューでは、 ブートマネージャをインストールするかどうかを選択します。 一般的に、次の場合は FreeBSD ブートマネージャをインストールするべきです。

- 二つ以上のドライブがあり、一番目のドライブ以外に FreeBSD をインストールする場合

- FreeBSD を同じディスク内に他の OS と共存させてインストールしており、コンピュータを起動する際に FreeBSD か他の OS かを選択したい場合

もし、コンピュータに FreeBSD のみをインストールするのであれば、最初のハードディスクにインストールし、Standard ブートマネージャを選択してください。もし、FreeBSD をブート可能なサードパーティ製のブートマネージャを使うのであれば、 None を選択してください。

選択をして Enter キーを押してください。

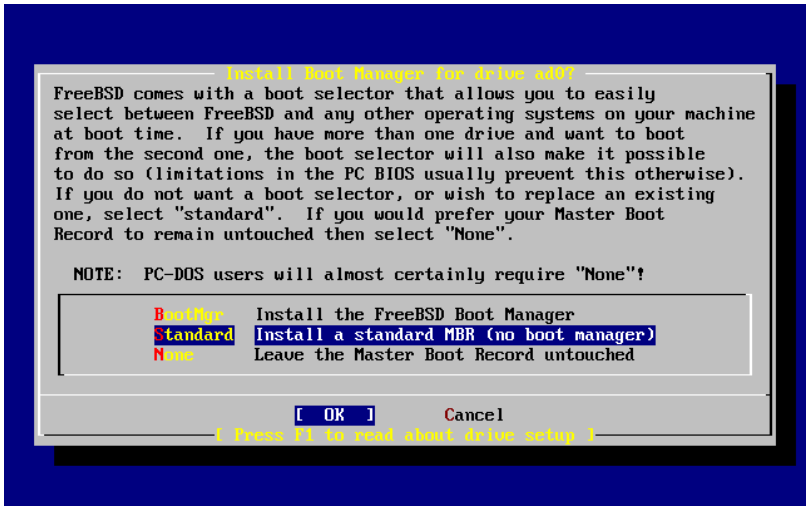


図3.16 sysinstall ブートマネージャメニュー

F1 キーを押すと表示されるヘルプ画面では、ハードディスクを OS 間で共有する場合に起こり得る問題について議論しています。

3.6.4. 他のドライブのスライスの作成

二つ以上ドライブがある場合、 ブートマネージャを選択した後、ドライブ選択画面に戻ります。FreeBSD を二つ以上のディスクにインストールしたいのなら、他のディスクを選択し、FDisk を用いてスライス作成の作業を繰りかえしてください。

重要

一番目以外のドライブに FreeBSD をインストールするのであれば、両方のドライブに FreeBSD のブートマネージャをインストールする必要があります。

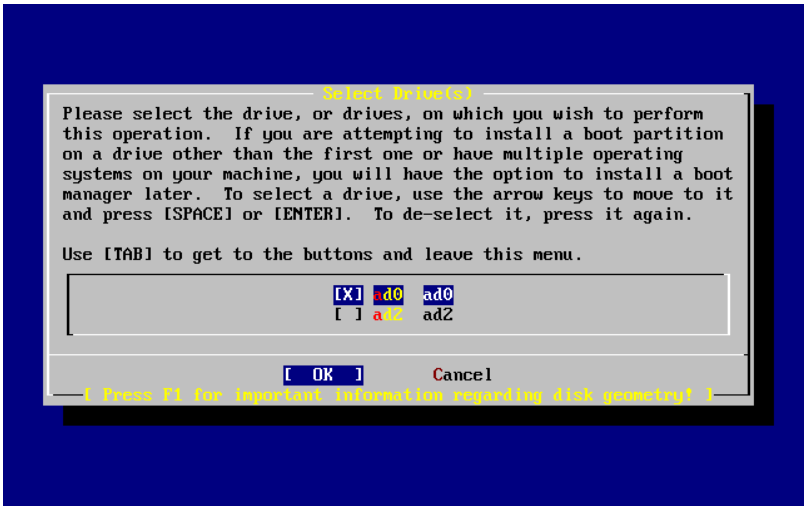


図3.17 ドライブ選択の終了

Tab キーを押して、最後に選択したドライブと [OK] および [Cancel] の間を切りかえてください。

[OK] に移るために Tab キーを一度押し、それからインストールを続けるために Enter キーを押してください。

3.6.5. disklabel でパーティション作成

次に、作成したばかりのスライス内にパーティションをいくつか作成してください。それぞれのパーティションには a から h までの文字がつけられ、b, c そして d パーティションは守るべき慣習的な意味を持っています。

特定のアプリケーションは、特別のパーティション構成 (特に二つ以上のディスクにわたってパーティションを構成している場合) から利益を得ることができます。 しかしながら、はじめて FreeBSD をインストールする場合、ディスクをどのようにパーティションに区切るか、ということをあまり大げさに考えることはありません。FreeBSD をインストールして使い方を学びはじめることの方がより重要です。 オペレーティングシステムにより詳しくなった後、いつでも FreeBSD を再インストールすることで、パーティション構成を変更できます。

以下は四つのパーティション：一つはスワップ領域、三つのファイルシステム、の構成です。

表3.2 一つ目のディスクのパーティションレイアウト

パーティション	ファイルシステム	サイズ	説明
a	/	1 GB	これはルートファイルシステムです。他のファイルシステムはすべてこの下のどこかにマウントされるでしょう。 ユーザのファイルはここに格納されず、また、通常の FreeBSD インストールでは、ここに配置するデータは約 128 MB なので、1 GB はこのファイルシステムに手ごろなサイズです。
b	なし	メモリの 2-3 倍	b パーティションには、 システムのスワップ領域が保持されています。スワップ領域の正しい容量を決めることは、ちょっとした芸術かもしれません。 確実な経験則は、物理的なメモリ (RAM) の 2-3 倍のサイズのスワップ領域とするように、としています。 また、最低 64 MB のスワップを確保するといいでしょう。 したがって、コンピュータが 32 MB 未満のメモリを搭載している時は、 スワップ領域を 64 MB にしてください。 二つ以上のディスクがあるときには、それぞれのディスクにスワップを置くことができます。FreeBSD はそれぞれのディスクをスワップに用い、スワップ動作を効果的に高速化させるでしょう。この場合、必要とするスワップサイズの合計を計算し、 それぞれのディスクに置くスワップサイズを求めるために、ディスク数で割ってください。
e	/var	512 MB ～ 4096 MB	/var には、ログファイルやその他の管理ファイルといった、 絶えず変化するファイルが保存されています。 これらのファイルの多くは FreeBSD の毎日の動作の間に広範囲にわたって読みこまれ、 書きこまれます。 (訳注: /とは) 別のファイルシステムにこれらのファイルを置くことは、 異なったアクセス頻度のディレクトリ内にある他のファイルに影響を与えずに、 FreeBSD がこれらのファイルへのアクセスを最適化することを可能とします。
f	/usr	ディスクの残り (少なくとも 8 GB)	他のすべてのファイルは、主に /usr およびそのサブディレクトリ内に保存されます。

警告

上記のサイズの値は、例として載せたものです。 経験のあるユーザのみが使ってください。 FreeBSD パーティションエディタの `Auto Defaults` で選択できる自動パーティションレイアウトが推奨されています。

二つ以上のディスクに FreeBSD をインストールしようとしているのなら、設定した他のスライスにもパーティションを作成しないといけません。最も簡単な方法は、それぞれのディスクに二つのパーティションを作成することです。 一つはスワップ領域、 そして一つはファイルシステムのためのパーティションです。

表3.3 残りのディスクのパーティション構成

パーティション	ファイルシステム	サイズ	説明
b	なし	説明を参照	スワップ領域をそれぞれのディスクにわたって分割できます。 たとえ a パーティションが (訳注: そのスライスに) ないとしても、 慣習により、スワップ領域は b パーティションとなります。
e	/disk*n*	ディスクの残り	ディスクの残りは一つの大きなパーティションとなります。これは e パーティションの代わりに、 a パーティション上へ簡単に置くことができるかもしれません。しかしながら慣習により、スライス上の a パーティションはルート (/) ファイルシステムとなるファイルシステムのために予約されています。 この慣習にしたがう必要はありませんが sysinstall(8) はこのように設定します。したがってこの慣習に従うことで，インストールが少しだけわかりやすくなります。 このファイルシステムをマウントする場所はどこにでも選択できます。 この例では、これらを /disk*n* としてマウントするよう提案しています。 ここで

パーティション	ファイルシステム	サイズ	説明
			n はそれぞれのディスクごとに変更する数字です。

パーティション構成を決めたら、sysinstall(8) を用いて作成してください。

```
                                Message
Now, you need to create BSD partitions inside of the fdisk
partition(s) just created. If you have a reasonable amount of disk
space (1GB or more) and don't have any special requirements, simply
use the (A)uto command to allocate space automatically. If you have
more specific needs or just don't care for the layout chosen by
(A)uto, press F1 for more information on manual layout.

                                [  OK  -]
                          [ Press enter or space -]
```

Disklabel と呼ばれる FreeBSD パーティションエディタを使うために、Enter キーを押してください。

図3.18「sysinstall ディスクラベルエディタ」は Disklabel を起動したときの画面表示です。この画面は三つのセクションに分かれています。

はじめの数行は現在作業しているディスクの名前、そして作成しようとしているパーティションを含むスライスを表示します この時点で Disklabel はスライスのことをスライス名ではなく**########**と呼びます 。この画面はスライス内の空き領域の合計も表示しています。この空き領域はスライス内に存在していて、しかしまだパーティションに割り当てられていない領域です。

画面の中段は作成したパーティションを表示しています。 それぞれのパーティションが含むファイルシステムの名前、サイズ、 ファイルシステム作成にまつわるいくつかのオプションが表示されます。

画面の下段にある三つ目のセクションは disklabel で有効なキー操作を表示します。

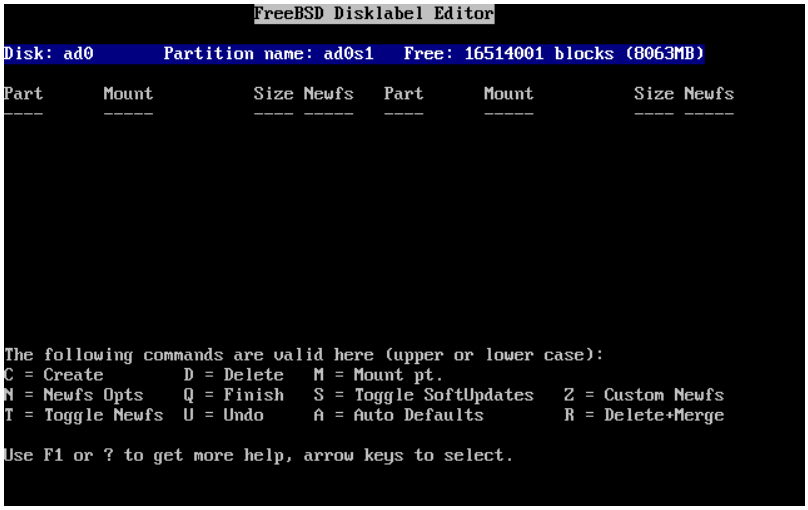


図3.18 sysinstall ディスクラベルエディタ

Disklabel は自動的にパーティションを作成し、 デフォルトのサイズを割りあてます。デフォルトのサイズは、ディスクのサイズからパーティションサイズを決めるアルゴリズムによって計算されます。 A キーを押すと、 図3.19「デフォルトの自動割り当てによる sysinstall ディスクラベルエディタ」とよく似た画面が表示されます。デフォルトのサイズは使用しているディスクのサイズに依存するので、希望するサイズになることもあるし、ならないこともあるでしょう。

注記

デフォルトのパーティション構成では、/tmp に対して、/ パーティションの一部ではなく、それ自身のパーティションを割り当てます。このことは、一時ファイルによって / パーティションがあふれてしまうことを防ぐのに役立ちます。

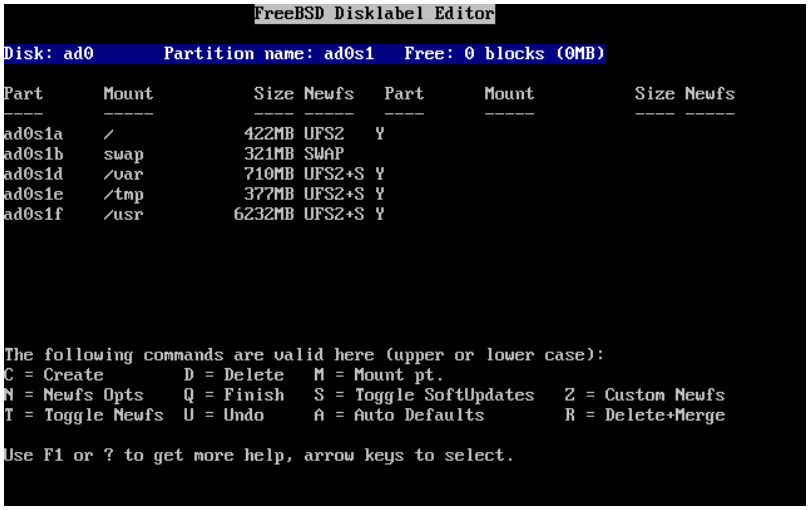


図3.19 デフォルトの自動割り当てによる sysinstall ディスクラベルエディタ

デフォルトのパーティションを変更するには、方向キーを用いて一番目のパーティションを選択し、D キーを押して削除してください。この作業を繰り返して、提案されたすべてのパーティションを削除してください。

一番目のパーティション（/ としてマウントされる a）を作成するには、画面の先頭の適切なディスクスライスが選択されていることを確認して、C キーを押してください。図3.20「ルートパーティションに割りあてる容量」に示されるような、新しいパーティションのサイズの入力をうながすダイアログが現れます。使用したいサイズをブロック数で入力するか、数字の後に M をつけてメガバイト単位、G をつけてギガバイト単位、C をつけてシリンダ単位で入力してください。

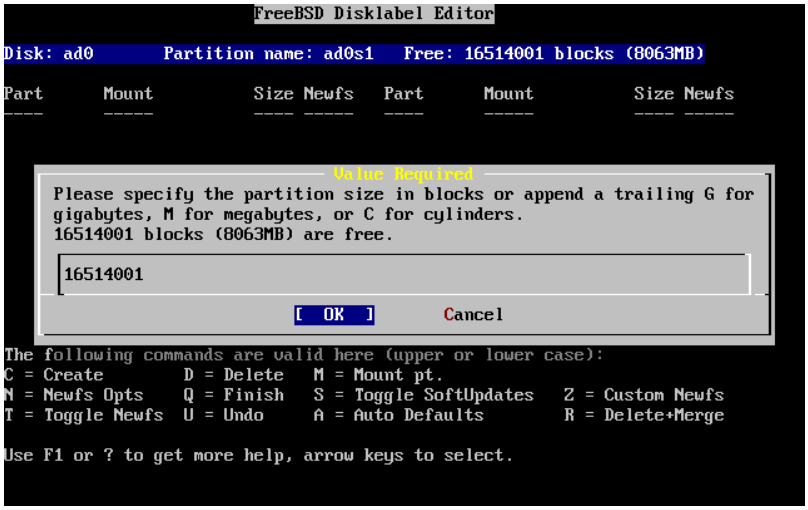


図3.20 ルートパーティションに割りあてる容量

(訳注: ダイアログに) 表示されるデフォルトのサイズは、 スライスに残っているサイズのパーティションを作成するでしょう。 上述したパーティションサイズを用いる場合は Backspace キーを用いて表示されている数字を削除し 図3.21「ルートパーティションサイズの編集」のように **512M** と入力してください。そして [OK] を押してください。

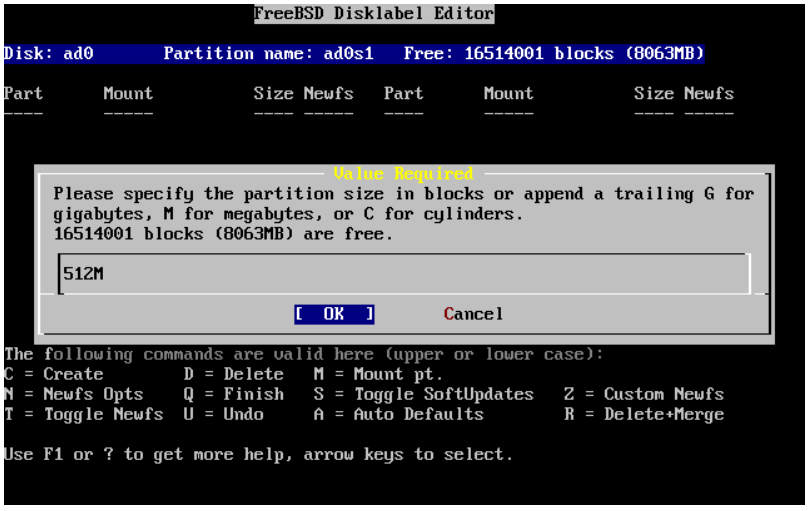


図3.21 ルートパーティションサイズの編集

パーティションのサイズを入力すると、 次にインストーラは、このパーティションがファイルシステムなのか、 それともスワップ領域なのかを聞いてきます。 図3.22「ルートパーティションタイプの選択」 のようにダイアログが表示されます。 この一番目のパーティションにはファイルシステムが含まれるので、FS を選択し、Enter キーを押します。

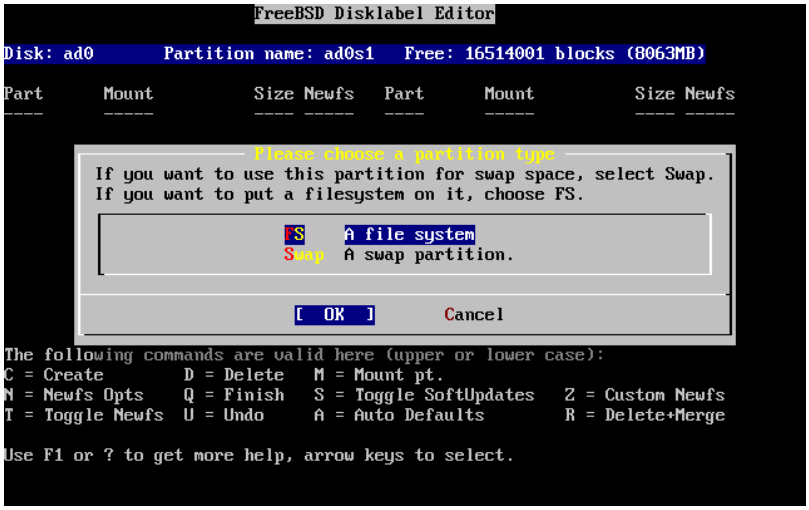


図3.22 ルートパーティションタイプの選択

最後に Disklabel に対して、このファイルシステムがどこにマウントされるか教えなければなりません。図3.23「ルートのマウントポイント選択」のようなダイアログが表示されます。**/** と入力し Enter キーを押してください。

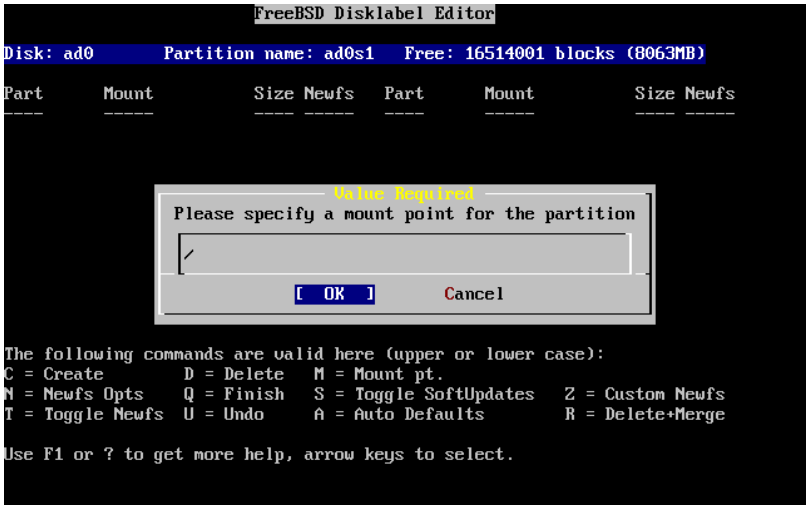


図3.23 ルートのマウントポイント選択

画面は更新されて、新しく作成したパーティションが表示されます。この作業を繰りかえして、 他のパーティションを作成してください。 スワップパーティションを作成したときには、ファイルシステムのマウントポイントの入力は促されません。最後のパーティションの

/usr を作成するときは、表示されたサイズをそのままにしておいて、スライスに残っているサイズを使用してください。

最終的な FreeBSD ディスクラベルエディタの画面は、選択された数値は異なっているかもしれませんが、図3.24「sysinstall ディスクラベルエディタ」のようになるでしょう。Q キーを押して終了します。

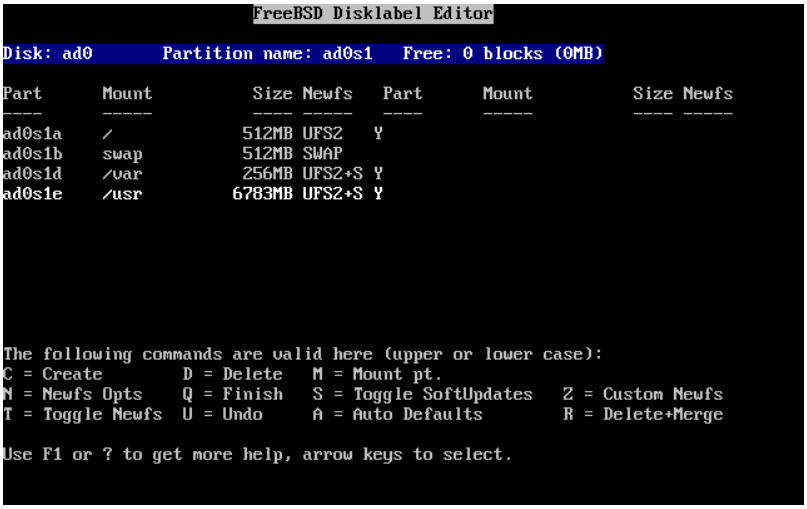


図3.24 sysinstall ディスクラベルエディタ

3.7. 何をインストールするか選択する

3.7.1. ディストリビューションセットの選択

インストールするディストリビューションセットを選択します。これは使おうとするシステムや利用可能なディスクの容量に依存します。 最小構成のインストールから全てのインストールまでのオプションがあらかじめ用意されています。 UNIX® や FreeBSD に慣れてない人はこれらのどちらかを選ぶべきです。 ディストリビューションセットのカスタマイズは、主としてより経験を積んだユーザーのためにあります。

F1 キーを押すとそのディストリビューションセットのオプションやそれらが何を含んでいるかというより詳しい情報が表示されます。ヘルプを見終わった後には、Enter キーを押してディストリビューションの選択メニューに戻ってください。

グラフィカルユーザインタフェースを望む場合には、FreeBSD のインストール後に Xorg の設定とデフォルトデスクトップの選択を行ってください。Xorg のインストールや設定に関する詳細な情報は 6章X Window System にあります。

カスタムカーネルをコンパイルする予定ならばソースコードを含むオプションを選択します。なぜカスタムカーネルを構築するのか、またカスタムカーネルの構築方法に関しては 9章FreeBSD カーネルのコンフィグレーション を参照してください。

すべてを含むシステムが最も多くのことを行えます。 十分なディスクスペースがあるならば、図3.25「ディストリビューションの選択」で示されるように All を矢印キーで選択し、Enter を押します。 ディスクスペースに懸念があるならば、状況に合ったよりふさわしいオプションを選択します。 インストール後にそのほかのディストリビューションを加えることができるので、完璧な選択をしようとして悩まないでください。

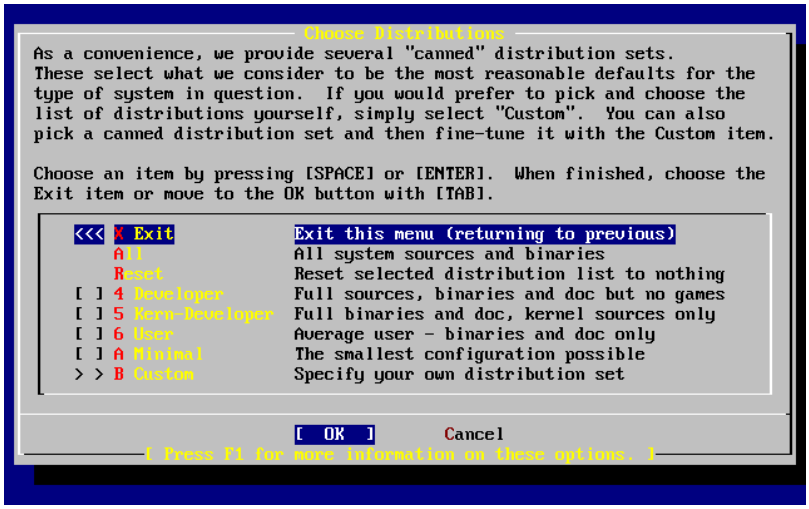


図3.25 ディストリビューションの選択

3.7.2. Ports Collection のインストール

お望みのディストリビューションを選ぶと、FreeBSD Ports Collection をインストールするかどうかを尋ねてきます。Ports Collection とは、簡単にソフトウェアをインストールする方法です。 サードパーティ製のソフトウェアパッケージのダウンロード、 コンパイル、インストールを自動的に行うためのファイルの集まりです。 5章アプリケーションのインストール - packages と ports で Ports Collection の使用方法が解説されています。

インストールプログラムはディスクに十分な空き容量があるかどうかをチェックしませんので、 十分な空き容量がある場合のみこのオプションを選択してください。 FreeBSD 10.0 では、FreeBSD Ports Collection は、約 500 MB のディスク容量を必要とします。より最新の FreeBSD のバージョンでは、安全のためもう少し大きなサイズを想定してください。

```
                        User Confirmation Requested
FreeBSD Ports Collection ############?

##################24,000 ###############
```

```
################################ 500 MB
#################### tarball ###############
################ (####FreeBSD # CD/DVD ########
extra disk ########## /cdrom ################
###############) -#

Ports Collection #############/usr #########
######################## Yes #########
####

Ports Collection ################ Ports ######
###############
    http://www.FreeBSD.org/ports

                              [ Yes -]     No
```

Ports Collection をインストールするなら [Yes] を、このオプションを選択しないのであれば [No] を矢印キーで選んでください。Enter キーを押して先に進みます。ディストリビューションの選択メニューが再度表示されます。

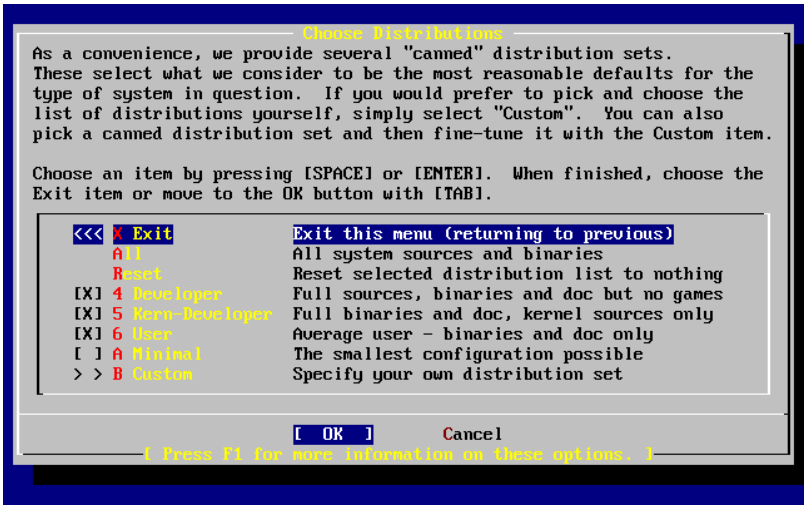


図3.26 ディストリビューションの確認

オプションを選択したら矢印キーで Exit を選び、[OK] がハイライトされていることを確認し、Enter を押して先に進んでください。

3.8. インストールメディアの選択

CD または DVD からインストールする場合、矢印キーを使って Install from a FreeBSD CD/DVD を選んで下さい。[OK] が選択されていることを確認して、Enter キーを押しインストールを開始してください。

そのほかの方法でインストールする場合には、 適切なオプションを選択しその後の指示に従ってください。

F1 キーを押すとインストールメディアに関するオンラインヘルプが表示されます。メディア選択メニューに戻るには Enter キーを押してください。

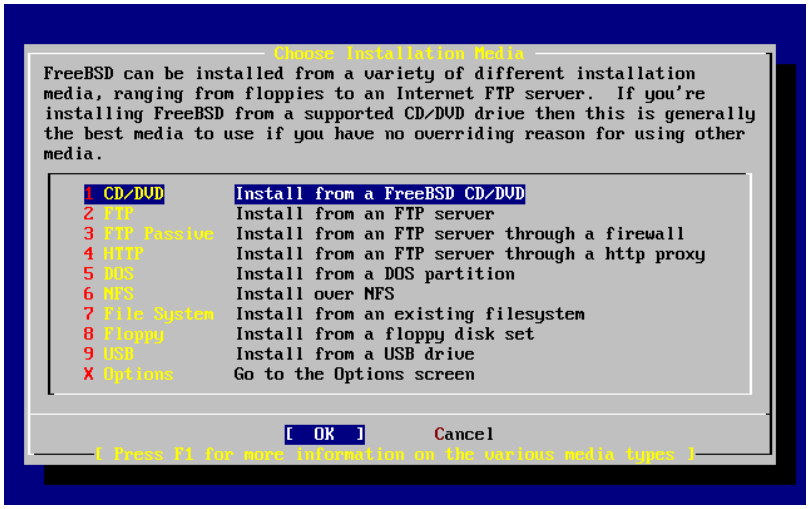


図3.27 インストールメディアの選択

FTP インストールモード

FTP 経由のインストールを行う場合、active FTP, passive FTP, HTTP proxy 経由の三種類の FTP インストールモードから選択することになります。

FTP Active: Install from an FTP server
: すべての FTP 転送の際に「Active」モードを使用します。ファイアウォール内部のマシンではうまく動きませんが、 多くの場合 passive モードに対応していないような古い FTP サーバでも動作します。passive モードでの FTP 転送 (こちらがデフォルトです) に失敗した場合には、active モードで確かめましょう!

FTP Passive: Install from an FTP server through a firewall
: このオプションでは、sysinstall(8) は、すべての FTP 転送の際に passive モードを使用します。このモードを使用すると、ランダム TCP ポートによる外部からの接続を許可していないファイアウォールを越えることができるようになります。

HTTP proxy 経由の FTP: Install from an FTP server through a http proxy

この方法では、sysinstall(8) は、HTTP プロトコルを使って proxy サーバに接続し、FTP の操作を実現します。 proxy サーバは FTP 要求を (訳注: HTTP から FTP に) 変換して FTP サーバに送るため、ファイアウォールが FTP 接続を禁止していても、HTTP proxy サーバが提供されていれば ファイアウォールを超えた FTP 接続を行なうことが可能です。この方法を用いる場合には、FTP サーバの他に proxy サーバを指定する必要があります。

proxy FTP サーバに対しては、ユーザ名の一部として 実際に接続したいサーバの名前を 「@」 記号に続けて与える必要があります。そうすると proxy サーバは本当のサーバの「ふり」 をするようになります。 たとえば、1234 ポートで要求を待つ proxy FTP サーバ `foo.example.com` を使って、`ftp.FreeBSD.org` をインストールするには、 オプションメニューで FTP ユーザ名に `ftp@ftp.FreeBSD.org` を、 パスワードとしてあなたのメールアドレスを設定します。そして、インストールメディアに FTP (proxy サーバが対応しているならば passive FTP)、URL として `ftp://foo.example.com:1234/pub/FreeBSD` を指定します。

`ftp.FreeBSD.org` の `/pub/FreeBSD` は `foo.example.com` が代理で処理をおこなうことになるので、インストール時の要求に応じて `ftp.FreeBSD.org` からファイルを取り寄せます。

3.9. インストール操作の確定

これで良いと思ったら、ここでインストールを進めることができます。 インストールするためにおこなうハードディスクに対する変更をやめるなら、 ここが最後のチャンスでもあります。

```
                    User Confirmation Requested
Last Chance! Are you SURE you want to continue the installation?

If you're running this on a disk with data you wish to save then WE
STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!

We can take no responsibility for lost disk contents!

                           [ Yes -]    No
```

[Yes] を選択して Enter キーを押すことでインストールは進みます。

インストールにかかる時間は、どのディストリビューションを選んだか、どのインストールメディアを使ったか、 そしてコンピュータの速度にも依存します。 インストール状況に従って、メッセージが逐次表示されます。

次のメッセージが表示されればインストールは終わりです。

```
                                Message

Congratulations! You now have FreeBSD installed on your system.

We will now move on to the final configuration questions.
For any option you do not wish to configure, simply select No.

If you wish to re-enter this utility after the system is up, you may
do so by typing: /usr/sbin/sysinstall.

                                  [ OK -]

                        [  Press enter or space  -]
```

Enter キーを押せばインストール後の設定画面に 移ることができます。

[No] を選択して Enter キーを押すと、インストールを中断して システムには何の変更もおこないません。そのときには次のメッセージが表示されます。

```
                                Message
Installation complete with some errors.  You may wish to scroll
through the debugging messages on VTY1 with the scroll-lock feature.
You can also choose "No" at the next prompt and go back into the
installation menus to retry whichever operations have failed.

                                  [ OK -]
```

何もインストールされなかったため、このメッセージが表示されています。 Enter キーを押しますと、インストーラを終了するためにメインメニューに戻ります。

3.10. インストール後の作業

インストール完了後に数々のオプション設定を行います。 オプションは、インストールした FreeBSD を起動する前に configuration の項目に入りなおすか、再起動後に sysinstall(8) の中にある Configure の項目で設定することができます。

3.10.1. ネットワークデバイスの設定

もし、FTP インストールのために PPP を設定したのならば、この画面は表示されず、前項で述べたように後から設定できます。

LAN の詳細情報と、FreeBSD のゲートウェイ・ルータの設定については 高度なネットワーク の章を参照してください。

```
                    User Confirmation Requested
  Would you like to configure any Ethernet or PPP network devices?

     (######### SLIP/PPP ####################)

                                [ Yes -]   No
```

ネットワークデバイスを設定するには [Yes] を選び、Enter キーを押します。設定しないのであれば、[No] を選びます。

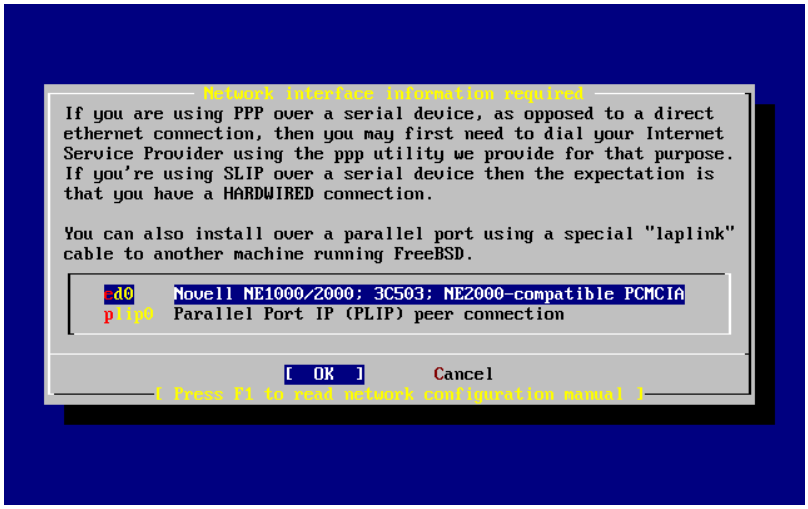


図3.28 イーサネットデバイスを選ぶ

設定したいインタフェースを矢印キーで選び、Enter キーを押します。

```
                    User Confirmation Requested
       Do you want to try IPv6 configuration of the interface?

      (############# IPv6 #############)

                                 Yes   [ No -]
```

プライベートな LAN において、現行のインターネットプロトコル (IPv4) で十分なのであれば、矢印キーで [No] を選び Enter を押してください。

RA サーバと IPv6 ネットワークで接続しているのであれば、[Yes] を選んで Enter を押してください。すると、数秒間 RA サーバを検索します。

```
                                User Confirmation Requested
```

```
        Do you want to try DHCP configuration of the interface?

      (############# DHCP ############)

                                Yes    [ No -]
```

もし、Dynamic Host Configuration Protocol (DHCP) が必要なければ、矢印で [No] を選び Enter を押してください。

[Yes] を選んだ場合、dhclient(8) を実行します。設定が成功すると、ネットワーク設定情報が自動補完されます。詳細については「DHCP」を参照してください。

次のネットワーク設定画面では、LAN 用ゲートウェイシステムのイーサネットデバイス構成が表示されます。

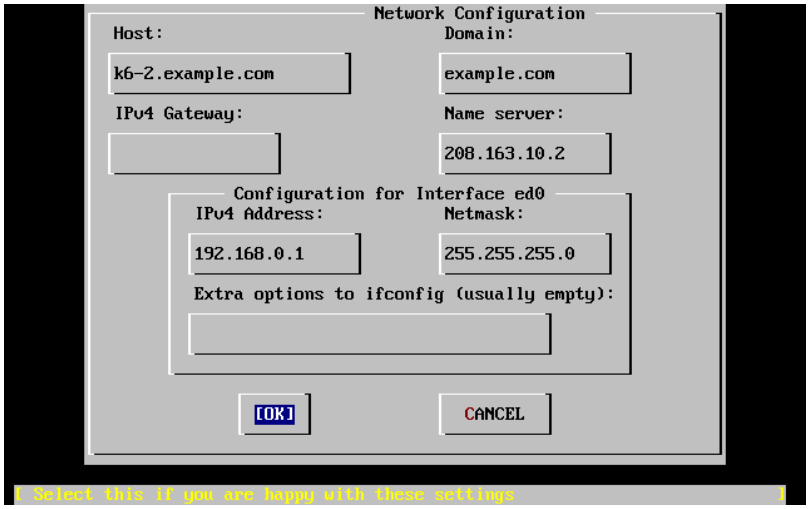


図3.29 *ed0* ネットワークの設定

Tab をつかって項目を選び、適切な情報を入力してください。

Host
: 完全修飾されたホスト名。この例では、`k6-2.example.com` 。

Domain
: コンピュータのドメイン名。この例では `example.com` 。

IPv4 Gateway
: ローカル以外のホストへパケットを送る際に使う IP アドレスです。 もしコンピュータがネットワーク上にいる場合は入力する必要があります。 このコンピュータがネットワークのインターネットゲートウェイである場合には この項目は空白のままにしま

す。IPv4 ゲートウェイはデフォルトゲートウェイまたはデフォルトルートとも呼ばれます。

Name server
: ローカル DNS サーバの IP アドレスを指定します。ローカル DNS サーバがプライベート LAN 上にないので、プロバイダの DNS サーバ (`208.163.10.2`) を指定しました。

IPv4 address
: このインタフェースで使う IP アドレスは `192.168.0.1` 。

Netmask
: この例では、LAN で使うアドレスブロックは、`192.168.0.0` - `192.168.0.255` です。ネットマスクは (`255.255.255.0`) です。

Extra options to ifconfig(8)
: ifconfig(8) への追加のオプションを入力してください。今回はありませんでした。

終了するときは、Tab キーを使って [OK] を選び、Enter を押します。

```
                    User Confirmation Requested
       Would you like to bring the ed0 interface up right now?

     (############################)

                              [ Yes -]   No
```

[Yes] を選んで Enter を押すと、ネットワークへの設定を行い、利用可能な状態にします。しかしながら、コンピュータをまだ再起動する必要があるので、インストールの間には、多くのことは出来ません。

3.10.2. ゲートウェイの設定

```
                     User Confirmation Requested
      Do you want this machine to function as a network gateway?

     (##############################)

                               [ Yes -]    No
```

もし、マシンが LAN のゲートウエイであり、他マシン間の パケットを転送している場合は、[Yes] を選んで Enter を押します。もし、マシンがネットワーク上のノードである場合、[No] を選び Enter を押して先に進んでください。

3.10.3. インターネットサービスの設定

```
                     User Confirmation Requested
Do you want to configure inetd and the network services that it ↺
provides?
```

```
      (##inetd ##########################)

                                Yes   [ No -]
```

もし [No] を選んだ場合には、さまざまなサービスは有効になりません。これらのサービスは、インストール後に /etc/inetd.conf をテキストエディタで編集することで有効になります。詳しくは inetd の章 を見てください。

これらのサービスを利用したいと考えているならば，[Yes] を選びインストールしてください。追加の確認事項が表示されるでしょう。

```
                       User Confirmation Requested
The Internet Super Server (inetd) allows a number of simple Internet
services to be enabled, including finger, ftp and telnetd.  Enabling
these services may increase risk of security problems by increasing
the exposure of your system.

      (##inetd # finger#ftp#telnet ###################
      ################################
      ########################)

With this in mind, do you wish to enable inetd?

                               [ Yes -]   No
```

[Yes] を選んで続けます。

```
                       User Confirmation Requested
inetd(8) relies on its configuration file, /etc/inetd.conf, to ↺
determine
which of its Internet services will be available.  The default ↺
FreeBSD
inetd.conf(5) leaves all services disabled by default, so they must ↺
be
specifically enabled in the configuration file before they will
function, even once inetd(8) is enabled.  Note that services for
IPv6 must be separately enabled from IPv4 services.

Select [Yes] now to invoke an editor on /etc/inetd.conf, or [No] to
use the current settings.

      (##inetd(8) ############################
    ###### (/etc/inetd.conf) #######FreeBSD #####
    inetd.conf(5) ############################
    ####inetd(8) ###################
    ########################IPv4 #######
    IPv6 ##############################

    [Yes] #### /etc/inetd.conf #############
    [No] #################)

                               [ Yes -]   No
```

[Yes] を選ぶと、対応するサービスの先頭行にある # を削除することで、サービスを有効にできます。

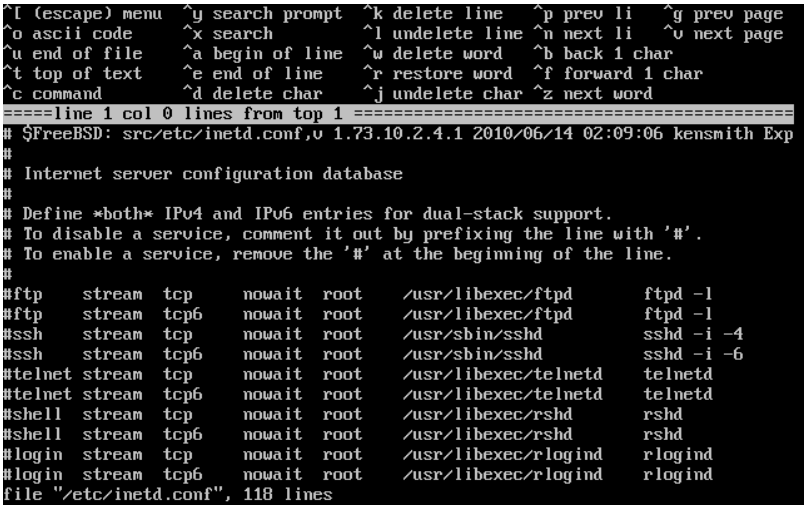


図3.30 inetd.conf の編集

変更が終わったら、Esc を押し、 表示されるメニューでエディタを終了し、変更点を保存してください。

3.10.4. SSH ログインを有効にする

```
                    User Confirmation Requested
                 Would you like to enable SSH login?
                          Yes        [  No  -]
```

[Yes] を選択すると、OpenSSH のデーモンである sshd(8) が有効になります。これは、コンピュータへの安全なリモートアクセスを許可します。OpenSSH のより詳しい情報については、「OpenSSH」をご覧ください。

3.10.5. Anonymous FTP

```
                    User Confirmation Requested
 Do you want to have anonymous FTP access to this machine?

                              Yes    [ No -]
```

3.10.5.1. Anonymous FTP を許可しない

デフォルトの [No] を選択して Enter を押してください。アカウントを持ったユーザは、パスワードを使い FTP でコンピュータにアクセスできます。

3.10.5.2. Anonymous FTP を許可する

anonymous FTP の接続を許可すると、誰でもコンピュータにアクセスできるようになります。このオプションを有効にする前に、セキュリティ上の問題点を検討してください。セキュリティに関する詳細については、15章セキュリティ をご覧ください。

anonymous FTP を有効にするには、矢印キーを使って [Yes] を選択し、Enter を押してください。以下のような確認事項が表示されるでしょう。

```
                        User Confirmation Requested
 Anonymous FTP permits un-authenticated users to connect to the ↺
system
 FTP server, if FTP service is enabled.  Anonymous users are
 restricted to a specific subset of the file system, and the default
 configuration provides a drop-box incoming directory to which ↺
uploads
 are permitted.  You must separately enable both inetd(8), and enable
 ftpd(8) in inetd.conf(5) for FTP services to be available.  If you
 did not do so earlier, you will have the opportunity to enable ↺
inetd(8)
 again later.

 If you want the server to be read-only you should leave the upload
 directory option empty and add the -r command-line option to ftpd(8)
 in inetd.conf(5)

 Do you wish to continue configuring anonymous FTP?

                            [ Yes -]         No
```

このメッセージは、anonymous FTP 接続を許可するのであれば、`/etc/inetd.conf` においても FTP サービスを有効にする必要があることを示しています。[Yes] を選び Enter を押して先に進んでください。以下の画面が表示されます。

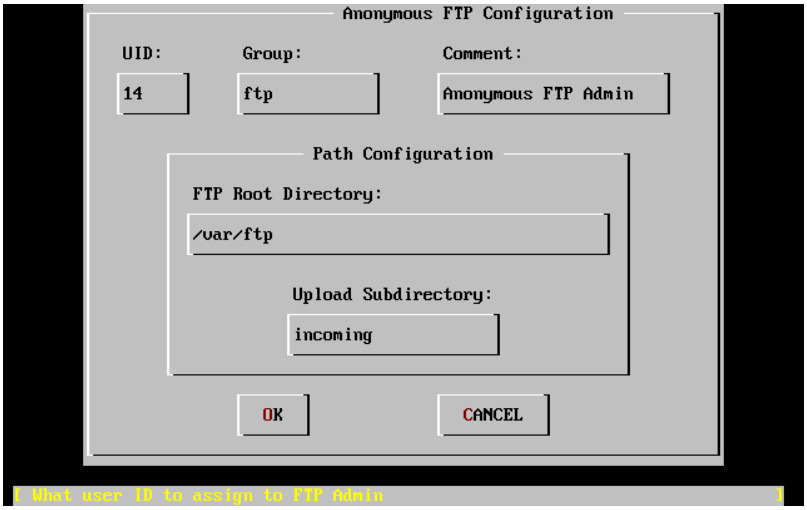


図3.31 Anonymous FTP のデフォルトの設定

Tab を使って項目を選び、適切な情報で埋めてください。

UID
: anonymous FTP ユーザに対して割り当てられるユーザ ID。アップロードされるすべてのファイルの所有者はこの ID となります。

Group
: anonymous FTP ユーザが所属するグループ

Comment
: `/etc/passwd` 内のユーザ情報を表す文字列

FTP Root Directory
: anonymous FTP で利用可能なファイルの置き場

Upload Subdirectory
: anonymous FTP ユーザによりアップロードされるファイルが置かれる場所

デフォルトでは、FTP のルートディレクトリは `/var` に設定されています。FTP が必要とする十分な容量が用意されていない場合には、かわりに FTP のルートディレクトリを `/usr/ftp` と設定し、`/usr` を使ってください。

十分な設定ができたら、Enter を押して先に進んでください。

```
                    User Confirmation Requested
       Create a welcome message file for anonymous FTP users?

                              [ Yes -]    No
```

[Yes] を選択した場合には、Enter を押すと、メッセージ編集用のエディタが自動的に起動します。

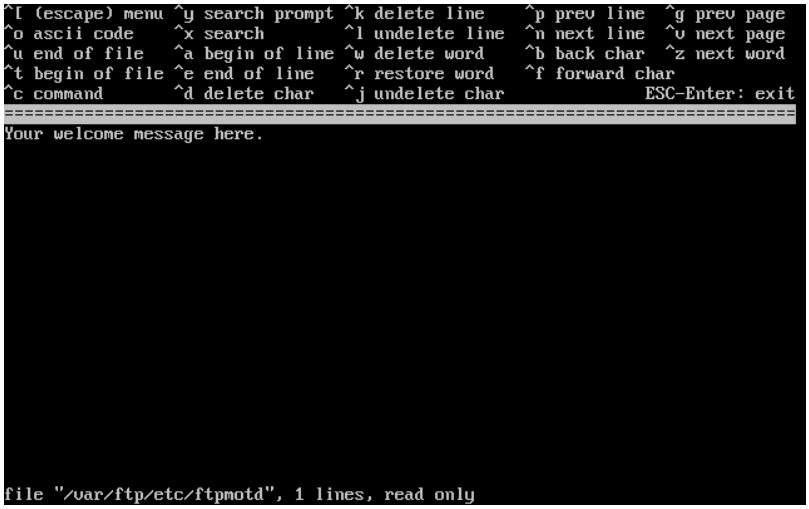


図3.32 FTP Welcome メッセージの編集

操作の説明に従ってメッセージを変更してください。 エディタの画面の下に表示されるファイルの名前、場所に注意してください。

Esc を押すと、ポップアップメニューが表示されます。 デフォルトは a) leave editor です。 Enter を押して終了し、先に進んでください。 変更を行った場合には、もう一度 Enter を押して、変更点を保存してください。

3.10.6. ネットワークファイルシステムの設定

ネットワークファイルシステム (NFS) を用いると、ネットワークを介してファイルの共有を行うことができます。コンピュータをサーバ、クライアントまたは両方に設定できます。より詳しい情報については「NFS」を参照してください。

3.10.6.1. NFS サーバ

```
                    User Confirmation Requested
 Do you want to configure this machine as an NFS server?

                                Yes    [ No -]
```

NFS の機能を必要としないのであれば、[No] を選択し Enter を押してください。

[Yes] を選んだ場合には、`/etc/exports` を作成する必要があることを示すメッセージが表示されます。

```
                                Message
 Operating as an NFS server means that you must first configure an
```

```
/etc/exports file to indicate which hosts are allowed certain kinds ↺
of
access to your local filesystems.
Press [Enter] now to invoke an editor on /etc/exports
                                [ OK -]
```

Enter を押して先に進んでください。 `/etc/exports` を編集するためにテキストエディタが立ち上がります。

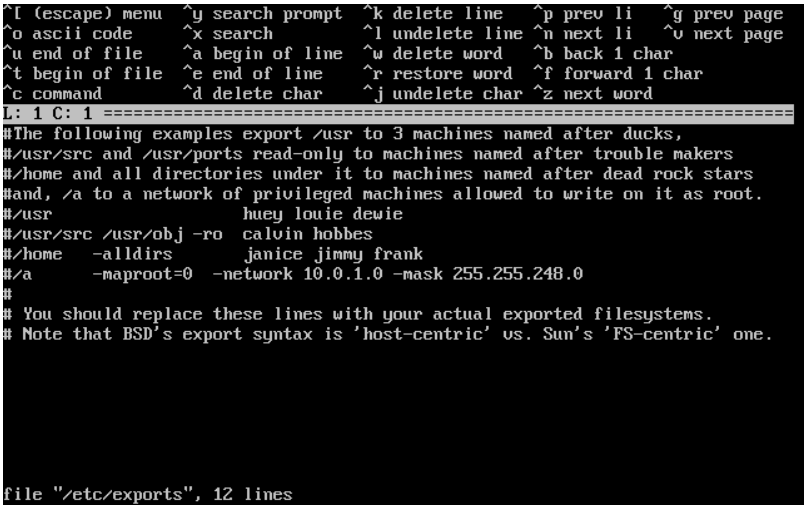


図3.33 `exports` ファイルの編集

説明に従い、共有されるファイルシステムを追加してください。エディタのスクリーンの下に表示されているファイル名の場所の情報に注意してください。

Esc を押すとポップアップメニューが表示されます。デフォルトは a) leave editor です。Enter を押して終了し、先に進んでください。

3.10.6.2. NFS クライアント

利用しているコンピュータが NFS クライアントによって NFS サーバにアクセスできるようになります。

```
                    User Confirmation Requested
 Do you want to configure this machine as an NFS client?

                               Yes   [ No -]
```

矢印キーを使って、適切に [Yes] または [No] を選択してください。

3.10.7. システムコンソールの設定

システムコンソールをカスタマイズするオプションがあります。

```
                    User Confirmation Requested
        Would you like to customize your system console settings?

                                  [ Yes -]  No
```

[Yes] を選択し、Enter を押してオプションの確認および設定をしてください。

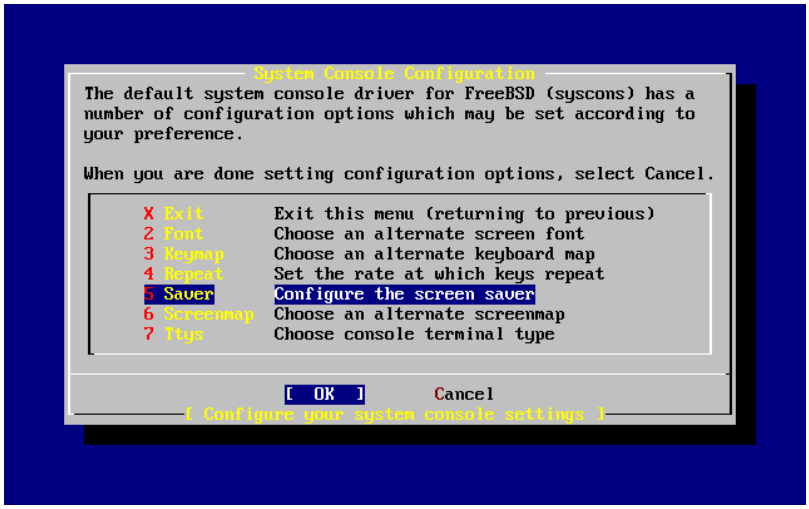


図3.34 システムコンソール設定オプション

良く使われるオプションはスクリーンセーバです。 矢印キーを使って Saver を選択し、Enter を押してください。

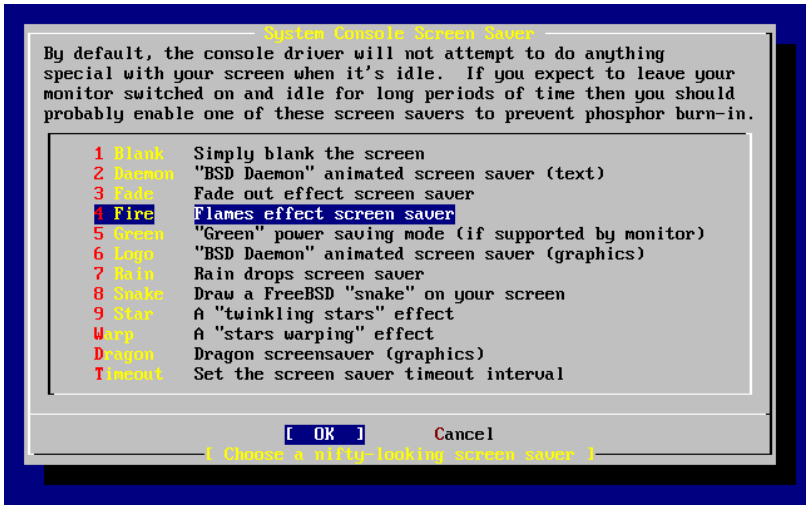


図3.35 スクリーンセーバオプション

矢印キーを使って、お望みのスクリーンセーバを選択し、Enter を押してください。システムコンソールの設定メニューが再び表示されます。

デフォルトの時間の間隔は 300 秒です。時間の間隔を変更する場合には、もう一度 Saver を選択してください。スクリーンセーバのオプションメニューにて、矢印キーで Timeout を選択して Enter を押してください。ポップアップメニューが表示されます。

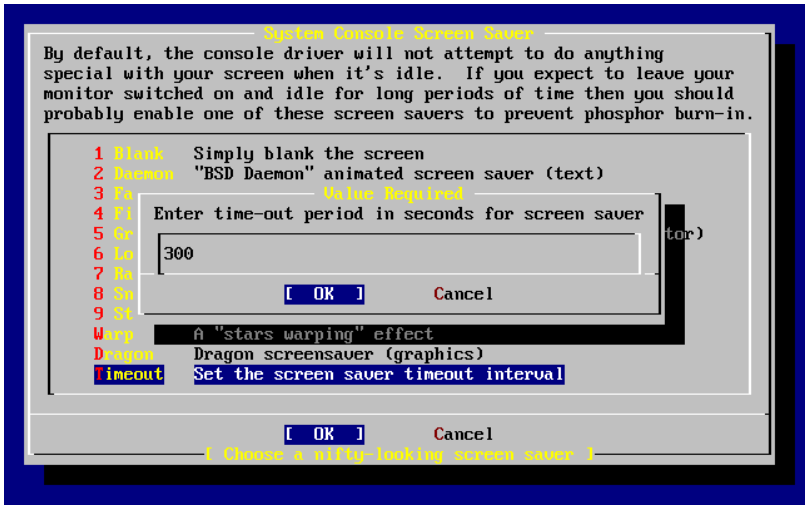


図3.36 スクリーンセーバのタイムアウト

値を変更したら [OK] を選択し、Enter を押して、システムコンソールの設定メニューに戻ってください。

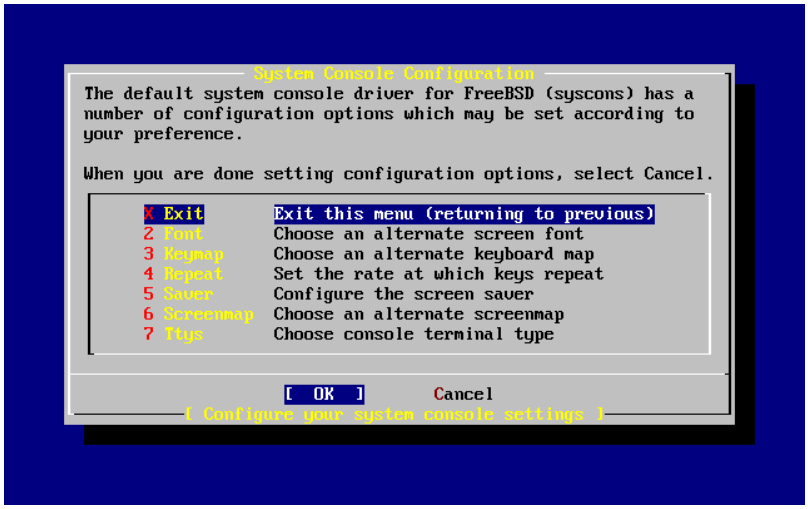


図3.37 システムコンソールの設定の終了

Exit を選択し、Enter を押して、インストール後の設定を続けてください。

3.10.8. タイムゾーンの設定

使用しているコンピュータのタイムゾーンを設定することで、地域による時刻の違いが自動的に調整され、タイムゾーンに関連した機能が適切に取り扱われます。

ここでの例では、コンピュータが United States の Eastern タイムゾーンにあるとしています。実際の地理的位置を選択してください。

```
                    User Confirmation Requested
         Would you like to set this machine's time zone now?

                                [ Yes -]   No
```

[Yes] を選択し Enter を押して、タイムゾーンを設定してください。

```
                    User Confirmation Requested
Is this machine's CMOS clock set to UTC? If it is set to local time
or you don't know, please choose NO here!

                                Yes   [ No -]
```

コンピュータのクロックがどのように設定されているかに従って、[Yes] または [No] を選択し、Enter を押してください。

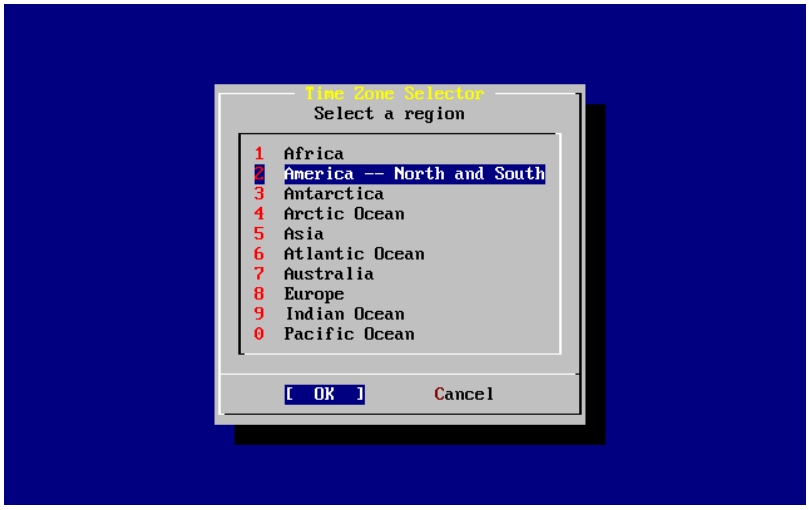


図3.38 地域の選択

矢印キーを使って、適切な地域を選択し、Enter を押してください。

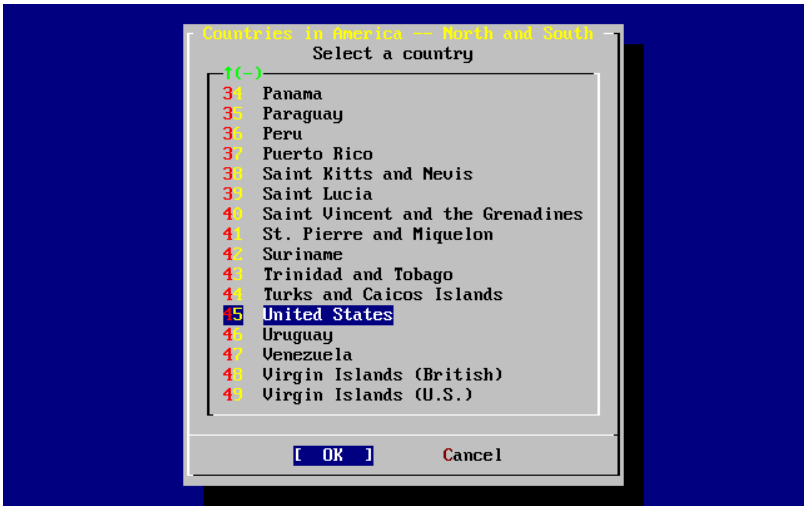


図3.39 国名の選択

矢印キーを使って、適切に国名を選び、Enter を押してください。

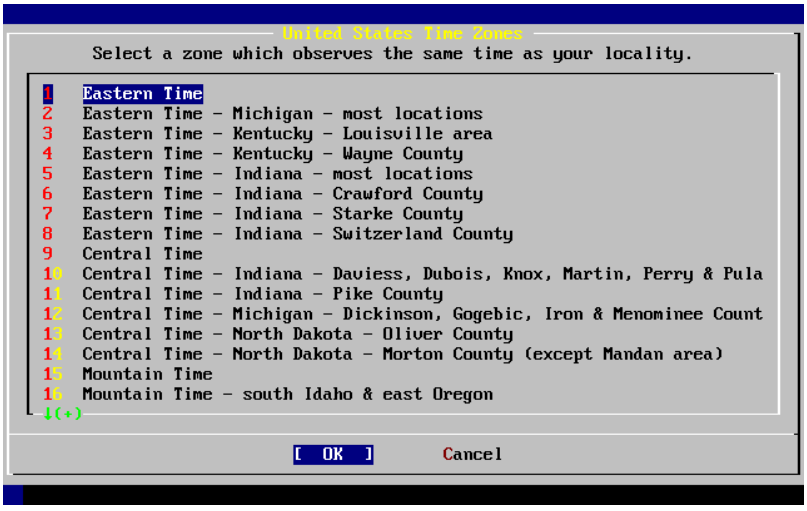


図3.40 タイムゾーンの選択

矢印キーを使って適切なタイムゾーンを選択し、Enter を押してください。

```
                    Confirmation
        Does the abbreviation 'EDT' look reasonable?

                    [ Yes -]   No
```

タイムゾーンの短縮形が適切かどうかを確認をしてください。問題なければ Enter を押して、インストール後の設定を続けてください。

3.10.9. マウスの設定

このオプションを設定することで、 コンソールやユーザプログラムにおいて 3-ボタンマウスによるテキストのカットアンドペーストが可能になります。2-ボタンマウスを使っている場合には、3-ボタンのエミュレートについて moused(8) を参照してください。ここでは、非 USB マウスの設定を例に説明します。

```
                    User Confirmation Requested
          Does this system have a PS/2, serial, or bus mouse?

                                [ Yes -]    No
```

PS/2、シリアルまたはバスマウスを使っている場合には [Yes] を選択してください。USB マウスを利用する場合には、[No] を選択し Enter を押してください。

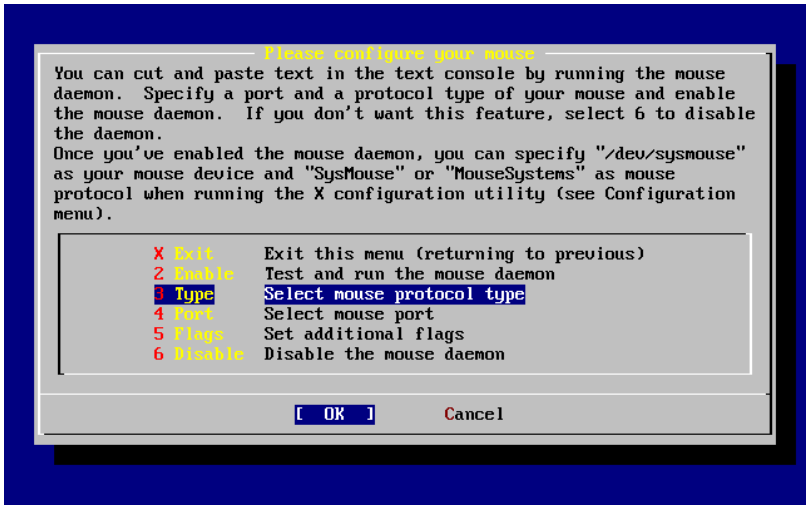


図3.41 マウスプロトコルのタイプの選択

矢印キーで Type を選択し、Enter を押してください。

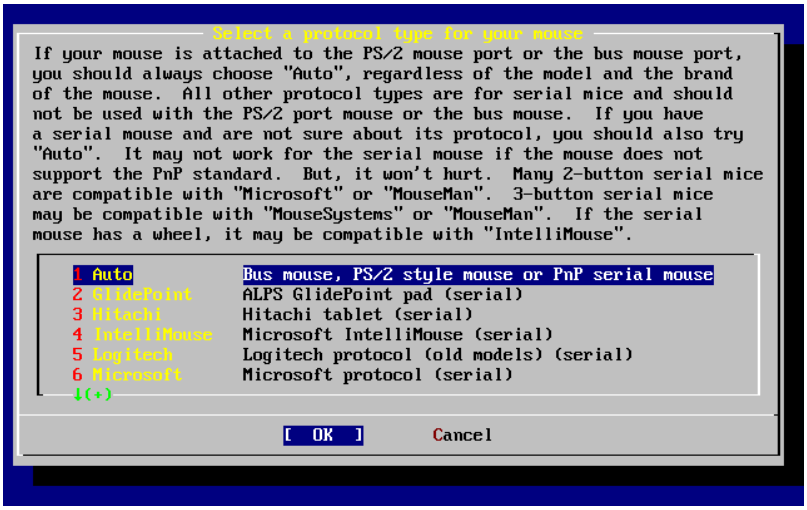


図3.42 マウスプロトコルの選択

この例では PS/2 マウスを使うので、デフォルトの Auto が適切です。マウスのプロトコルを変更するには、矢印キーを使って他のプロトコルを選択してください。[OK] がハイライトされていることを確認し、Enter を押してこのメニューを終了してください。

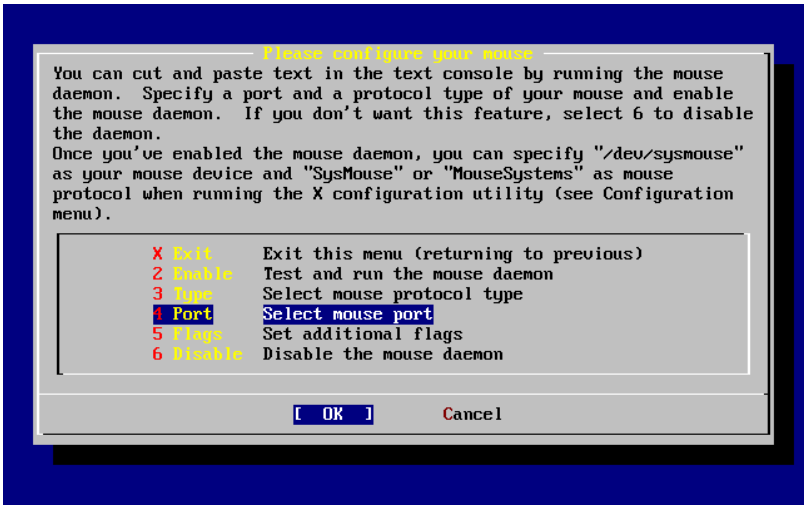


図3.43 マウスポートの設定

矢印キーで Port を選択して Enter を押してください。

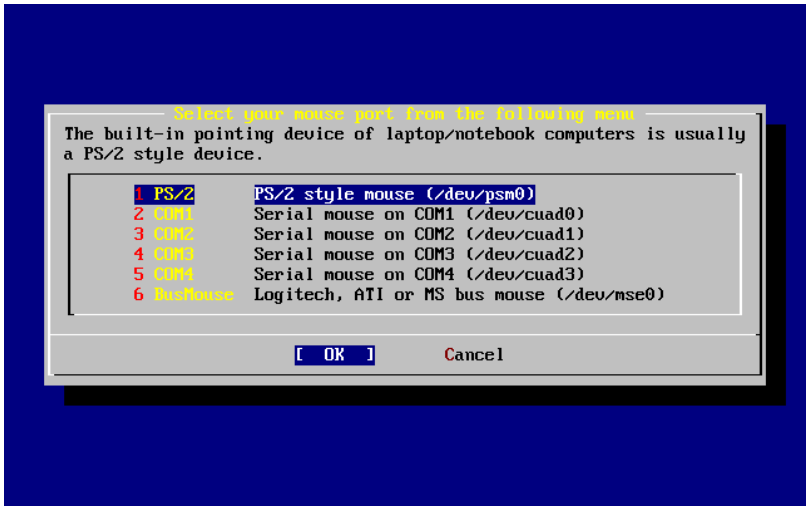


図3.44 マウスポート の設定

このシステムでは PS/2 マウスを使っているので、デフォルトの PS/2 が適切です。ポートを変更する場合には、矢印キーで選択して Enter を押してください。

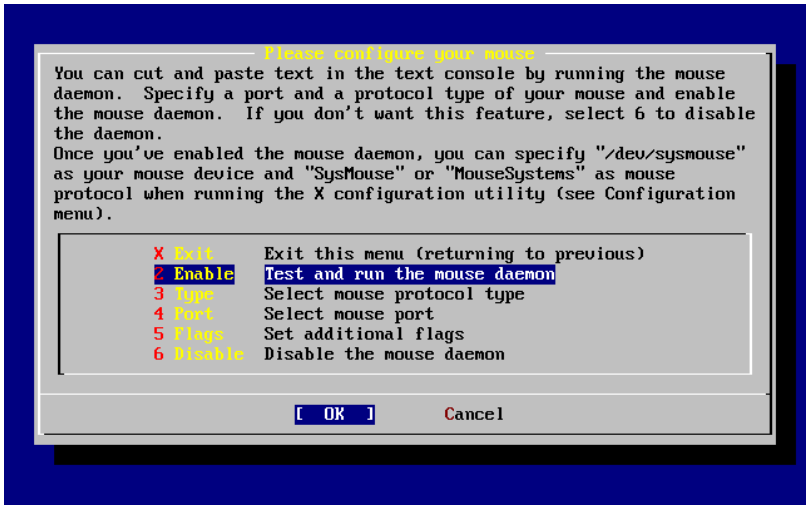


図3.45 マウスデーモンを有効にする。

最後に、矢印キーで Enable を選択して Enter を押し、マウスデーモンを有効にして、テストしてください。

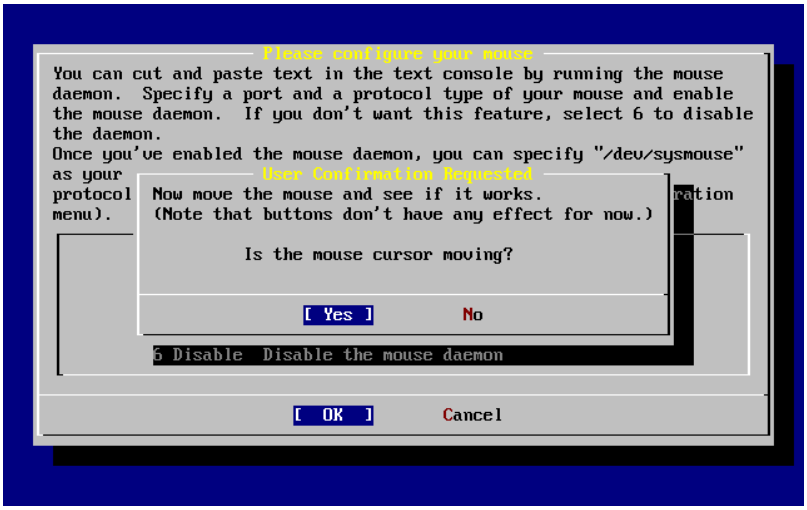


図3.46 マウスデーモンのテスト

スクリーン上でマウスを動かし、カーソルが正しく反応することを確認してください。確認ができたら、[Yes] を選択して Enter を押してください。もし、うまく動かないのであれば、マウスの設定が適切に行われていません。— [No] を選択して、他の設定オプションを使って試してください。

矢印キーで Exit を選択し、Enter を押してインストール後の設定を続けてください。

3.10.10. Packages のインストール

packages はコンパイル済みのバイナリで、 ソフトウェアをインストールするのに便利な方法です。

説明では、例としてひとつの package をインストールしますが、お望みであれば、ここで複数の package を追加することもできます。インストール後は、sysinstall(8) を使って package を追加できます。

```
                    User Confirmation Requested
The FreeBSD package collection is a collection of hundreds of
ready-to-run applications, from text editors to games to WEB servers
and more. Would you like to browse the collection now?

                              [ Yes -]   No
```

[Yes] を選択して、Enter を押してください。Package を選択する画面が表示されます。

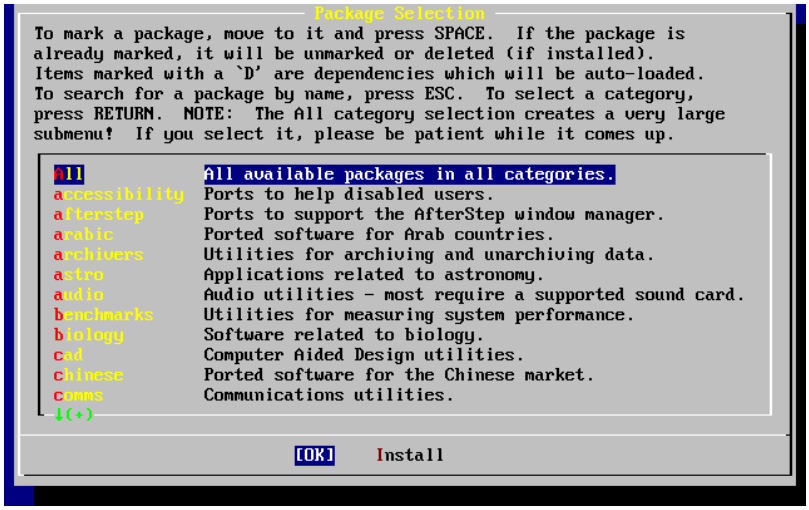


図3.47 Package カテゴリの選択

現在のインストールメディアに存在する packages のみをインストールできます。

All を選ぶと、利用可能なすべての package が表示されます。もしくは、特定のカテゴリを選択してください。矢印キーで選択したいカテゴリをハイライトし、Enter を押してください

選択したカテゴリで利用可能なすべての package のメニューが表示されます。

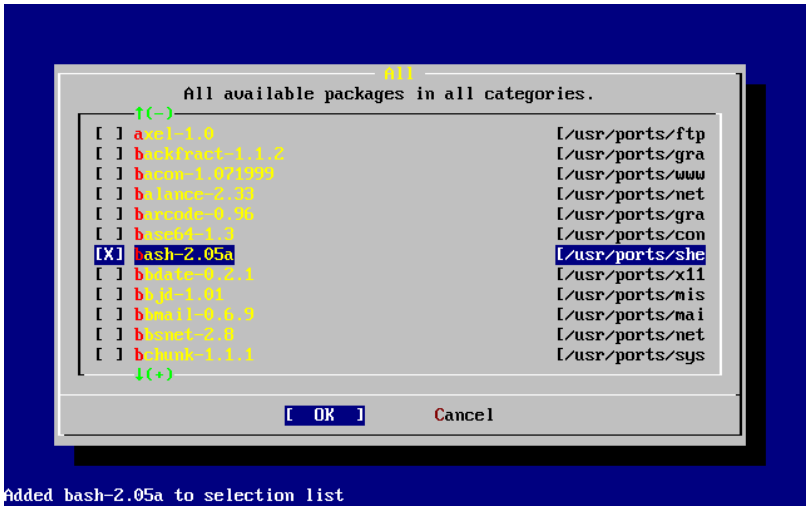


図3.48 Packages の選択

ここでは、bash シェルが選択されています。希望する数だけ package を選択してハイライト表示にし、Space キーを押してください。それぞれの package の簡単な説明がスクリーン下の左端に表示されます。

Tab キーを押すと、最後に選択した package と [OK], [Cancel] で選択が切り替わります。

インストールする package の選択が終わったら、Tab を一度押して [OK] に切り替え、その後 Enter を押して Package 選択メニューに戻ってください。

左右の矢印キーでも、[OK] と [Cancel] 切り替えることができます。この方法で [OK] を選択することもできますので、その後、Enter を押して Package 選択メニューに戻ってください。

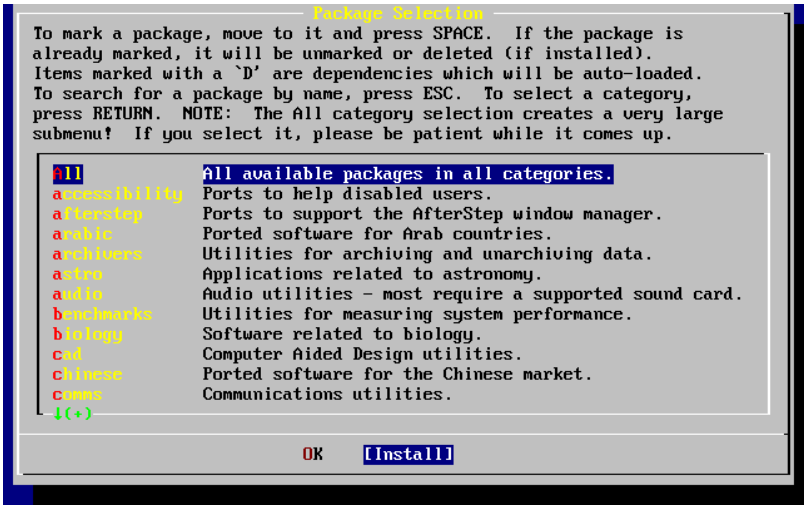


図3.49 Packages のインストール

Tab と矢印キーで [Install] を選択し、Enter を押すと、インストールを確認するメニューが表示されます。

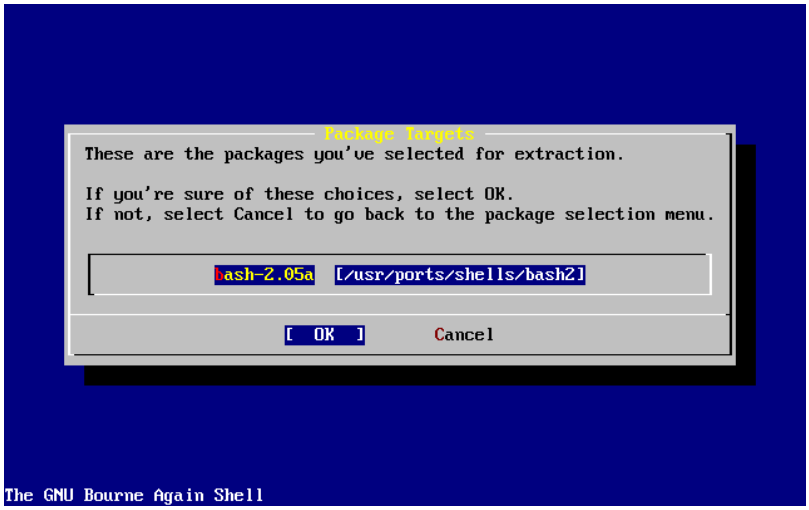


図3.50 Package のインストールの確認

[OK] を選択し、Enter を押すと package のインストールが始まります。インストールが終わるまで、インストールのメッセージが表示されます。エラーメッセージが表示されないかどうか注意してください。

packages をインストールしたら、最後の設定に進みます。package を選択しなかった場合も、Install を選択して、最後の設定に戻ってください。

3.10.11. ユーザ/グループの追加

システムに root 以外でログインできるよう、インストール時には、少なくとも一人のユーザを追加してください。通常 root パーティッションの容量は小さいので、root 権限でアプリケーションを実行すると、すぐに消費してしまいます。

```
                    User Confirmation Requested
 Would you like to add any initial user accounts to the system? ↺
Adding
 at least one account for yourself at this stage is suggested since
 working as the "root" user is dangerous (it is easy to do things ↺
which
 adversely affect the entire system).

                              [ Yes -]   No
```

[Yes] を選択し、Enter を押して、ユーザを追加してください。

図3.51 User の選択

矢印キーで User を選択して、Enter を押してください。

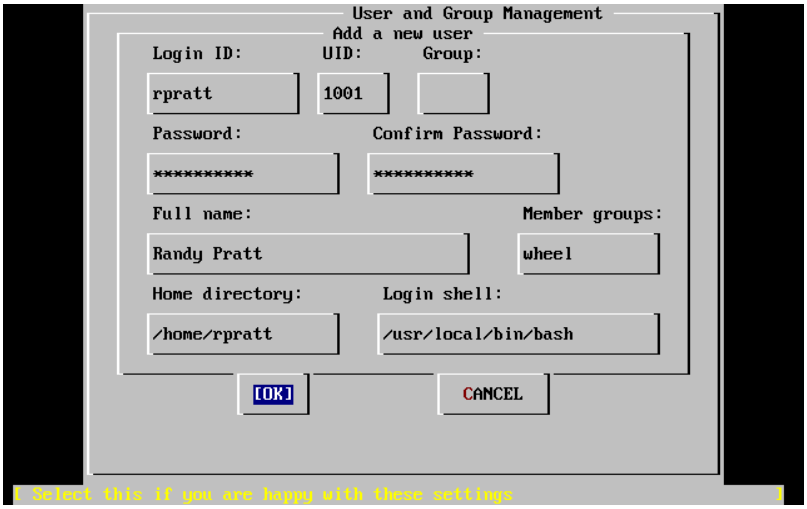


図3.52 ユーザ情報の入力

Tab で項目を選択すると、必要とされる情報の入力を助けるため、スクリーンの下に以下の説明が表示されます。

Login ID
　新しいユーザのログインネーム (必須)

UID
　新しいユーザの ID 番号 (空欄のままにすると、自動的に割り当てられます)

Group
　新しいユーザのログイングループ

Password
　このユーザのパスワード (このフィールドは注意して入力してください!)

Full name
　ユーザのフルネーム (コメント)

Member groups
　このユーザの所属するグループについて (すなわち、どのようなアクセス権を持つか)

Home directory
　ユーザのホームディレクトリ (空欄にするとデフォルトで設定されます)

Login shell
　ユーザのログインシェル (空欄にするとデフォルトの `/bin/sh` に設定されます)。

この例では、ログインシェルは `/bin/sh` から、先ほど package でインストールした bash を使うよう、`/usr/local/bin/bash` に変更されています。インストールされていないシェルを選ぶと、ユーザはログインできなくなりますので、インストールされていないシェルは選ばないでください。FreeBSD で最も標準的なシェルは C シェルです。`/bin/tcsh` で指定できます。

ユーザは、`root` 権限を持つスーパユーザになることができる `wheel` グループにも追加されています。

十分に記述したら、[OK] を押して、User および Group 管理メニューを再び表示してください。

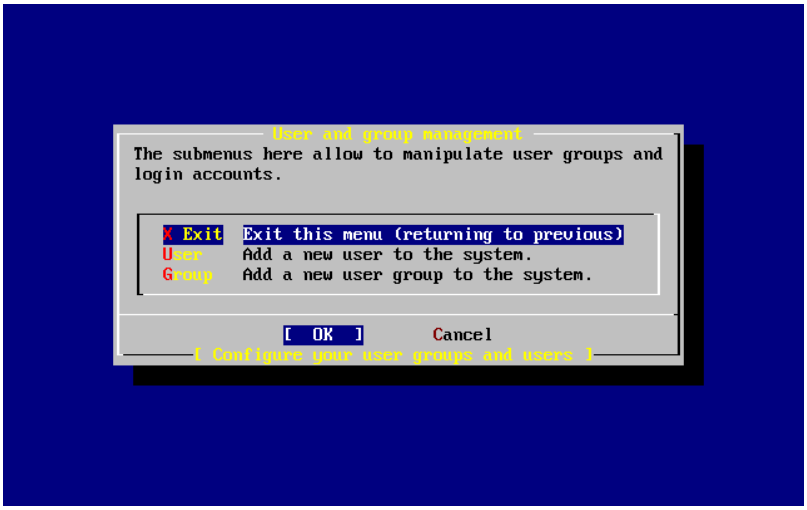


図3.53 ユーザおよびグループの管理を終了する

ここでグループを追加することもできます。または、後で sysinstall(8) を利用してアクセスできます。

ユーザを追加したら、矢印キーで、Exit を選択して、Enter を押して、次のインストール項目に進んでください。

3.10.12. root パスワードの設定

```
                        Message
Now you must set the system manager's password.
This is the password you'll use to log in as "root".

                          [ OK -]

               [ Press enter or space -]
```

Enter を押して、root パスワードを設定してください。

パスワードは、正しく 2 回入力してください。パスワードは忘れないでください。パスワードとして入力した文字は表示されませんし、アスタリスクが表示されることもありません。

```
New password:
Retype new password :
```

パスワードの入力に成功したら、インストールは次に進みます。

3.10.13. インストールの終了

以下のようなメッセージが表示され、設定が終わったかどうかを尋ねられます。

```
                    User Confirmation Requested
Visit the general configuration menu for a chance to set any last
options?

                                 Yes   [ No -]
```

[No] を選択し、Enter を押すことでメインインストールメニューに戻ります。

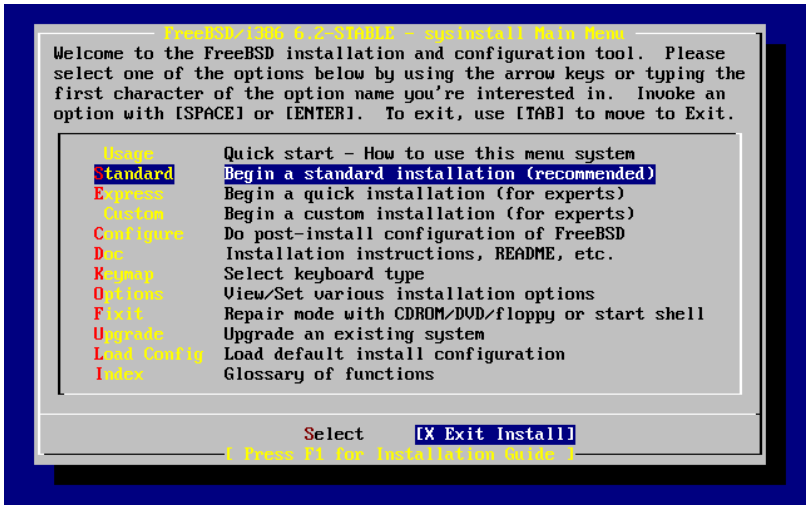


図3.54 インストールの終了

方向キーで [X Exit Install] を選択し、Enterを押します。インストーラがインストールを終了してもよいか確認を求めます。

```
                    User Confirmation Requested
Are you sure you wish to exit? The system will reboot.

                               [ Yes -]   No
```

[Yes] を選択してください。CDROM ドライブから起動しているのなら、ディスクを取り出すこと示す以下のメッセージが表示されます。

```
                        Message
Be sure to remove the media from the drive.

                        [ OK -]
              [ Press enter or space -]
```

マシンが再起動し始めると CDROM ドライブはロックされるので、ドライブから素早くディスクを取り出してください。[OK] を押して再起動してください。

システムは再起動するので、 表示されるかもしれないすべてのエラーメッセージに注意してください。詳細については、「FreeBSD のブート」をご覧ください。

3.10.14. ネットワークサービスの設定

##: Tom Rhodes.

ネットワークサービスの設定は、新しいユーザにとって、 この領域について基本的な知識を持たないと気が遠くなるような作業となるでしょう。インターネットを含むネットワークは、現代のすべてのオペレーティングシステムにとって重要なので、FreeBSD の広範囲にわたるネットワークの能力を理解することは有益でしょう。

ネットワークサービスは、 ネットワーク上のどこからでもアクセスを許可するプログラムです。 時間が経過するとネットワークサービスのバグが攻撃者により発見されるので、 必要なネットワークサービスのみを有効にすることは重要です。 もし、あるネットワークサービスが必要かどうか疑わしい場合には、それが必要となるまでは、そのネットワークサービスを有効にしないでください。後で sysinstall(8) を実行するか、`/etc/rc.conf` ファイルを編集することで機能を有効にできます。

Networking オプションを選択すると、以下のようなメニューが表示されます。

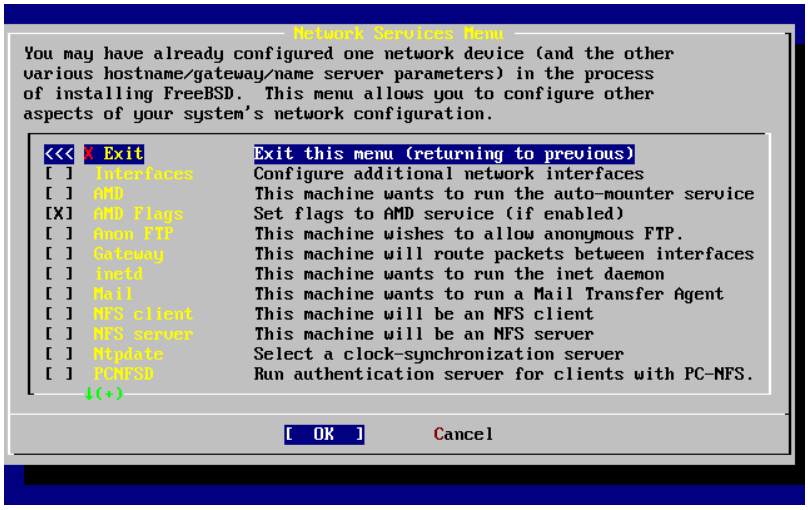


図3.55 ネットワーク設定 (メニューの前半)

最初のオプションである Interfaces については、「ネットワークデバイスの設定」で説明されています。

AMD オプションを選択すると amd(8) に対応します。このオプションは、通常 NFS プロトコルともに、リモートファイルシステムの自動マウントのために使われます。

次は AMD Flags オプションです。 選択すると AMD フラグを入力するためのメニューが表示されます。 メニューには以下のようなデフォルトのオプションのセットが含まれています。

```
-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map
```

`-a` は、デフォルトのマウント位置を指定します。ここでは `/.amd_mnt` を指定しています。`-l` は、デフォルトの `log` を指定します。ただし、syslogd(8) が指定されていると、すべてのログはシステムログのデーモンに送られます。`/host` は、リモートホストからエクスポートされたシステムのマウントに使われます。一方 `/net` は、IP アドレスからエクスポートされたファイルシステムのマウントに使われます。`/etc/amd.map` は、AMD エクスポートのデフォルトオプションを定義します。

Anon FTP は、anonymous FTP 接続を許可します。 使用しているコンピュータを FTP サーバとして利用する場合には、このオプションを選択してください。このオプションを選択した際のセキュリティ上のリスクについて把握しておいてください。 セキュリティリスクの説明、および、詳細の設定のためのメニューが表示されます。

Gateway メニューでは、コンピュータをゲートウェイに設定します。もし、インストールの過程で Gateway オプションを間違って選択したのであれば、ここで無効にできます。

Inetd オプションを使うと、inetd(8) を設定したり、完全に無効にできます。

Mail オプションでは、システムのデフォルトの Mail Transfer Agent (MTA) を設定できます。このオプションを選択すると、以下のメニューが起動します。

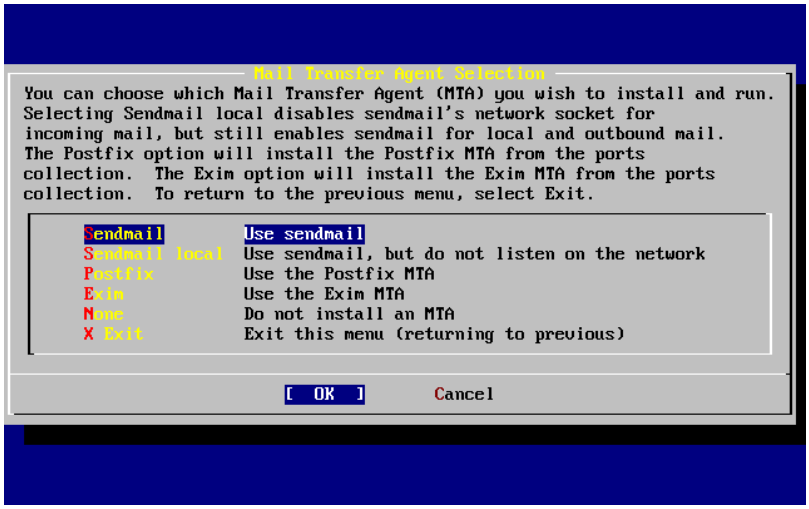


図3.56 デフォルトの MTA の選択

このメニューでは、インストールしてデフォルトに設定する MTA を選択できます。MTA は、システム上またはインターネット上のユーザにメールを配送するメールサーバです。

Sendmail を選択すると、デフォルトの MTA として sendmail がインストールされます。Sendmail local を選択すると、sendmail をデフォルトの MTA に設定しますが、インターネットからの電子メールの受け取りを無効にします。他のオプションである Postfix や Exim も、Sendmail と同じようにメールを配送します。

MTA メニューの次のメニューは NFS client です。 このメニューは、システムが NFS サーバと通信するように設定します。NFS では、NFS プロトコルを通じて、他のコンピュータのファイルシステムを利用できるようになります。 クライアントやサーバの設定についての詳細は「NFS」をご覧ください

その下のオプションが NFS server オプションです。システムを NFS サーバに設定します。このオプションは、Remote Procedure Call RPC サービスに必要なオプションを追加します。RPC は、ホストとプログラム間の接続の調整に利用できます

次の行は Ntpdate オプションです。これは、時刻の同期を取り扱います。選択すると下のようなメニューが表示されます。

図3.57 Ntpdate の設定

このメニューでは、 地理的に最も近いサーバを選択してください。 遠い場所のサーバを選ぶと、 ネットワークの応対時間が長くなる可能性があるので、 最も近いサーバを選択することにより、より正確な時刻の同期ができるようになります。

次のオプションは、PCNFSD の選択です。このオプションは、Ports Collection から net/pcnfsd package をインストールします。Microsoft の MS-DOS® オペレーティングシステムのような、自分自身では NFS 認証サービスを提供できないシステムに対し、NFS 認証サービスを提供する有効なユーティリティです。

ここで、他のオプションを見るために、少しスクロールダウンしてください。

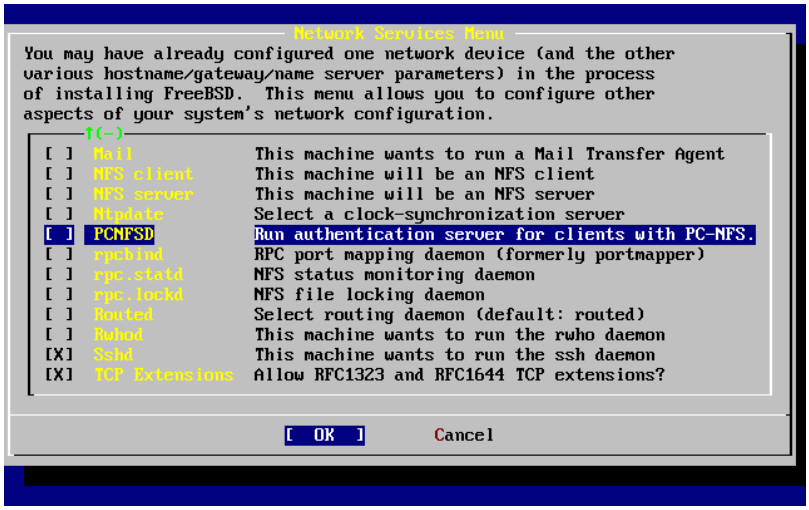


図3.58 ネットワーク設定 (メニューの後半)

RPC。 rpcbind(8) は、NFS サーバとクライアントの通信を取り扱い、 NFS サーバが正しく機能するために必要です。 rpc.statd(8) デーモンは、ステータスを監視し、 報告されたステータスは、通常 /var/db/statd.status に記録されます。 次のオプションの rpc.lockd(8) は、 ファイルロックサービスを提供します。 この機能は通常 rpc.statd(8) と共に、 どのホストがロックを必要としているか、 どのくらいの頻度でリクエストするかをモニタするために利用されます。 最後の二つのオプションは、デバッグ時に有用ですが、NFS サーバおよびクライアントの動作時には必要ありません。

次のメニューの Routed は、 ルーティングデーモンを設定します。 routed(8) は、ネットワークルーティングテーブルを管理し、 マルチキャストルーターを理解し、 そして、リクエストがあると、物理的に接続したホストに対し、 ネットワークを通してルーティングテーブルを提供します。このユーティリティは、主にローカルネットワークのゲートウェイマシンで利用されます。 選択すると、 ユーティリティのデフォルトの場所を要求するメニューが表示されます。デフォルトの場所で良ければ、Enter キーを押してください。その後、別のメニューが表示され、 routed(8) に渡すフラグを要求します。画面にには、デフォルトのフラグの -q が表示されます。

次のメニューは Rwhod オプションです。 このオプションは、システムの初期化時に rwhod(8) を起動します。 rwhod ユーティリティは、定期的にシステムのメッセージをネットワークにブロードキャストし、「consumer」モードでは、システムのメッセージを収集します。詳細な情報は、ruptime(1) および rwho(1) ページにあります。

リストの最後から 2 番目のオプションは、sshd(8) です。これは OpenSSH のためのセキュアシェルサーバです。標準の telnetd(8) や ftpd(8) サーバではなく、このサーバが強く推奨されています。このサーバは、暗号化された通信を行い、あるホストと他のホストを安全に接続します。

最後のオプションが TCP Extensions オプションです。このオプションは、RFC 1323 および RFC 1644 で定義されます。多くのホストにおいて、このオプションはコネクションを高速にする一方で、いくつかのコネクションを破棄します。サーバに対して推奨はされませんが、スタンドアローンなコンピュータには有益でしょう。

ネットワークサービスの設定が終わったら、一番上の X Exit までスクロールアップして次の設定項目に進むか、X Exit を 2 回選択し、その後 [X Exit Install] を選び、sysinstall(8) を終了してください。

3.10.15. FreeBSD のブート

3.10.15.1. FreeBSD/i386 の起動

すべてがうまくいっていれば、画面上にメッセージがスクロールし、 ログインプロンプトが表示されます。 このメッセージを見るには、Scroll-Lock を押し、PgUp および PgDn キーを使ってください。Scroll-Lock をもう一度押すとプロンプトに戻ります。

バッファの容量制限により、すべてのメッセージは表示されないかもしれませんが、ログインの後に dmesg(8) で見ることができます。

インストールの際に設定したユーザ名とパスワードを用いてログインしてください。 必要な時を除き `root` としてログインすることは避けてください。

典型的なブートメッセージ (バージョン情報は省略しています)

```
Copyright (c) 1992-2002 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, ↺
1994
        The Regents of the University of California. All rights ↺
reserved.

Timecounter "i8254"  frequency 1193182 Hz
CPU: AMD-K6(tm) 3D processor (300.68-MHz 586-class CPU)
  Origin = "AuthenticAMD"  Id = 0x580  Stepping = 0
  Features=0x8001bf<FPU,VME,DE,PSE,TSC,MSR,MCE,CX8,MMX>
  AMD Features=0x80000800<SYSCALL,3DNow!>
real memory  = 268435456 (262144K bytes)
config> di sn0
config> di lnc0
config> di le0
config> di ie0
config> di fe0
config> di cs0
config> di bt0
config> di aic0
config> di aha0
config> di adv0
config> q
avail memory = 256311296 (250304K bytes)
Preloaded elf kernel "kernel" at 0xc0491000.
Preloaded userconfig_script "/boot/kernel.conf" at 0xc049109c.
```

```
md0: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1: <VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device ↺
1.0 on pci0
pci1: <PCI bus> on pcib1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device ↺
7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
uhci0: <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at ↺
device 7.2 on pci0
usb0: <VIA 83C572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr 1
uhub0: 2 ports with 2 removable, self powered
chip1: <VIA 82C586B ACPI interface> at device 7.3 on pci0
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xe800-0xe81f irq 9 at
device 10.0 on pci0
ed0: address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq 2 ↺
on isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
atkbdc0: <keyboard controller (i8042)> at port 0x60-0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq 1 on atkbdc0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: model Generic PS/2 mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff ↺
on isa0
sc0: <System console> at flags 0x1 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
sio1: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
ppbus0: IEEE1284 device found /NIBBLE
Probing for PnP devices on ppbus0:
plip0: <PLIP network interface> on ppbus0
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
```

```
ppi0: <Parallel I/O> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master using ↺
UDMA33
ad2: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata1-master using ↺
UDMA33
acd0: CDROM <DELTA OTC-H101/ST3 F/W by OIPD> at ata0-slave using PIO4
Mounting root from ufs:/dev/ad0s1a
swapon: adding /dev/ad0s1b as swap device
Automatic boot in progress...
/dev/ad0s1a: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1a: clean, 48752 free (552 frags, 6025 blocks, 0.9% ↺
fragmentation)
/dev/ad0s1f: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1f: clean, 128997 free (21 frags, 16122 blocks, 0.0% ↺
fragmentation)
/dev/ad0s1g: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1g: clean, 3036299 free (43175 frags, 374073 blocks, 1.3% ↺
fragmentation)
/dev/ad0s1e: filesystem CLEAN; SKIPPING CHECKS
/dev/ad0s1e: clean, 128193 free (17 frags, 16022 blocks, 0.0% ↺
fragmentation)
Doing initial network setup: hostname.
ed0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        inet6 fe80::5054::5ff::fede:731b%ed0 prefixlen 64 tentative ↺
scopeid 0x1
        ether 52:54:05:de:73:1b
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x8
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
Additional routing options: IP gateway=YES TCP keepalive=YES
routing daemons:.
additional daemons: syslogd.
Doing additional network setup:.
Starting final network daemons: creating ssh RSA host key
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
cd:76:89:16:69:0e:d0:6e:f8:66:d0:07:26:3c:7e:2d root@k6-2.example.com
 creating ssh DSA host key
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
f9:a1:a9:47:c4:ad:f9:8d:52:b8:b8:ff:8c:ad:2d:e6 root@k6-2.example.↺
com.
setting ELF ldconfig path: /usr/lib /usr/lib/compat /usr/X11R6/lib
/usr/local/lib
a.out ldconfig path: /usr/lib/aout /usr/lib/compat/aout /usr/X11R6/
lib/aout
starting standard daemons: inetd cron sshd usbd sendmail.
```

```
Initial rc.i386 initialization:.
rc.i386 configuring syscons: blank_time screensaver moused.
Additional ABI support: linux.
Local package initialization:.
Additional TCP options:.

FreeBSD/i386 (k6-2.example.com) (ttyv0)

login: rpratt
Password:
```

低速なマシンでは、RSA および DSA キーを作成するのに少々時間がかかるかもしれません。これは初めてインストールした後に初めてブートする時だけ起こります。その後のブートはより速くなるでしょう。

Xorg が設定され、デフォルトデスクトップが選択されていた場合、コマンドラインで `startx` と入力することで Xorg を起動することが可能です。

3.10.16. FreeBSD のシャットダウン

オペレーティングシステムを適切にシャットダウンすることは重要です。ただ電源を落すということはしないでください。まず、コマンドラインから su(1) と入力し、`root` パスワードを入力してスーパユーザとなります。これは `wheel` のメンバとなっているユーザだけが行うことができます。そうでなければ `root` としてログインしてください。システムをシャットダウンするには、`shutdown -h now` と入力してください。

```
The operating system has halted.
Please press any key to reboot.
```

シャットダウンコマンドが実行され、「Please press any key to reboot」というメッセージが現れた後に電源を落すことは安全です。電源を落す代わりに何らかのキーが押された場合、システムはリブートするでしょう。

Ctrl+Alt+Del というキーの組合せを使ってシステムをリブートすることもできます。しかしながら、推奨はされません。

3.11. トラブルシューティング

この節では、インストールの際に起こりえる共通の問題を解決するための情報が書いてあります。また、FreeBSD と Windows® のデュアルブートを行う際のいくつかの質問と回答も書いてあります。

3.11.1. なにかおかしいときには何をすればよいでしょうか

PC アーキテクチャのさまざまな制限により、100% 確実にデバイスを認識することは不可能です。しかしながら、失敗した時にいくつかできることがあります。

インストールする FreeBSD のバージョンの ハードウェアノート を調べて、使っているハードウェアに対応しているかどうかを確認してください。

もしハードウェアがサポートされているにもかかわらず、 動作しなかったり他の問題点がある時は、カスタムカーネル を構築して `GENERIC` カーネルに含まれていないデバイスのサポートを追加してください。 デフォルトのカーネルでは、ほとんどのハードウェアデバイスの IRQ, IO アドレス、DMA チャネルが工場出荷時の状態であると仮定しています。もしハードウェアの設定が変更されているのであれば、 カスタムカーネルのコンフィグレーションファイルを作成し、再コンパイルを行なって、FreeBSD が認識するよう設定してください。

存在しないデバイスを認識してしまうことにより、その後実際に存在するデバイスの認識を失敗してしまうことがあります。このような場合は衝突しているドライバを無効にします。

注記

いくつかのインストール上の問題は、各ハードウェア装置、 特にマザーボード BIOS のファームウェアのアップデートで回避または緩和することができます。 ほとんどのマザーボードまたはコンピュータ製造メーカーは、アップグレード情報を載せているウェブサイトを用意しています。

多くの製造メーカーは、 重要な更新のようなそれなりの理由がない限りマザーボードの BIOS のアップグレードは行わないよう強く推奨しています。 アップデートの過程で失敗するかもしれず、BIOS チップに深刻なダメージを与える原因となり得るからです。

3.11.2. Windows® ファイルシステムの利用

現時点では、FreeBSD は、Double Space™ アプリケーションで圧縮されたファイルシステムに対応していません。そのため、FreeBSD がデータにアクセスする前に、ファイルシステムを展開しておく必要があります。 展開するには、Start> Programs > System Tools メニューの Compression Agent を実行してください。

FreeBSD は、MS-DOS® ファイルシステム (FAT ファイルシステムとも呼ばれます) に対応しています。 mount_msdosfs(8) コマンドは、 ファイルシステムの内容にアクセスできるように、ファイルシステムを現存のディレクトリ構成にマウントします。通常 mount_msdosfs(8) プログラムは、直接用いられることはありません。システムから `/etc/fstab` 経由で利用されたり、mount(8) から適切なパラメータとともに用いられます。

`/etc/fstab` の典型的な記述例は以下のようになります。

```
/dev/ad0sN  /dos  msdosfs rw  0 0
```

注記

この記述がうまくいくためには、/dos が存在している必要があります。/etc/fstab のフォーマットの詳細については、fstab(5) をご覧ください。

FAT ファイルシステムをマウントする mount(8) の典型例は以下となります。

```
# mount -t msdosfs /dev/ad0s1 /mnt
```

この例では、FAT ファイルシステムは、プライマリのハードディスクの最初のパーティションに配置されています。dmesg(8) や mount(8) の出力から、パーティションレイアウトを把握するのに十分な情報を得ることが出来ます。

注記

FreeBSD が割り当てる FAT パーティションの番号は、他のオペレーティングシステムのものとは異なることがあります。特に拡張パーティションには、プライマリパーティションより大きなスライス番号が通常割り当てられます。どのスライスが FreeBSD に属し、どのスライスが他のオペレーティングシステムに属するかを決めるのに fdisk(8) は役立ちます。

NTFS パーティションについても、同様に mount_ntfs(8) コマンドでマウントできます。

3.11.3. トラブルシューティングに関する Q & A

問: 起動時のハードウェア検出で、システムがハングアップします。または、インストール中にシステムがおかしくなります。

答: i386, amd64 および ia64 プラットフォームにおいて、FreeBSD はシステムの設定を手助けするシステム ACPI サービスを、起動時に検出された場合に広く使います。残念ながら、まだいくつかの不具合が、ACPI ドライバとシステムのマザーボードに存在しています。起動ステージ 3 において、ヒント情報 hint.acpi.0.disabled を以下のように設定すると ACPI を無効にできます。

```
set hint.acpi.0.disabled="1"
```

この設定はシステムが起動するたびにリセットされるので、 `/boot/loader.conf` に `hint.acpi.0.disabled="1"` を追加して、永続的に変更するようにしてください。ブートローダのより詳しい情報については「この章では」で説明します。

問: FreeBSD をインストールした後の最初のハードウェアからの起動で、カーネルが読み込まれ、ハードウェアを検出しましたが、 次のようなメッセージを表示して停止しました。

```
changing root device to ad1s1a panic: cannot mount root
```

何が問題ですか?

答: 起動ディスクがシステムの最初のディスクではない場合には、 この問題が起こりえます。BIOS は、FreeBSD とは異なる番号付けを用い、どの番号がどの番号に対応するかを正しく見つけ出すことは簡単ではありません。

この問題が起きた場合には、 BIOS ディスク番号、ディスクのタイプおよびそのディスクタイプに対する FreeBSD ディスク番号を設定することで、ルートファイルシステムがどこにあるかを FreeBSD に指定する必要があります。

システムのそれぞれ IDE バスのマスタに接続された 2 つの IDE ディスクがあり、2 番目のディスクから FreeBSD を起動する場合を考えます。BIOS は、これらのデバイスをディスク 0 および ディスク 1 と認識し、一方 FreeBSD は、`ad0` および `ad2` と認識します。

BIOS ディスク 1 に FreeBSD があり、ディスクタイプが `ad` で、FreeBSD ディスク番号が 2 の場合には、次のようになるでしょう。

```
1:ad(2,a)kernel
```

プライマリバスにスレーブが接続されている場合には、 上記は必要なく、適切ではありません。

2 番目の状況は、システムに 1 つまたはそれ以上の IDE ディスクがあるときに、SCSI ディスクから起動するような場合です。このようなケースでは、FreeBSD ディスク番号は、BIOS ディスク番号より小さな番号になります。2 つの IDE ディスクと SCSI ディスクがある場合には、SCSI ディスクは BIOS ディスク 2、タイプ `da`、FreeBSD ディスク番号が 0 となるため、FreeBSD に BIOS ディスク 2 (システムの 1 番目の SCSI ディスク) から起動するように設定するには以下のようになります。

```
2:da(0,a)kernel
```

IDE ディスクが 1 つの場合には、かわりに `1:` を指定してください。

使用する適切な値が決まったら、 テキストエディタを使って、コマンドを `/boot.config` に記述してください。 他に指定がなければ、FreeBSD はこのファイルの内容を、 デフォルトの `boot:` プロンプトへのレスポンスとして使います。

問: FreeBSD をインストールした後、ハードウェアから起動しましたが、 ブートマネージャは、ブートメニューで `F?` を表示し、それ以上起動が進みません。

答: FreeBSD のインストール時に、 パーティションエディタで指定したハードウェアのディスクジオメトリが正しくありません。パーティションエディタまで戻り、 ハードディスクの正しいジオメトリを指定してください。ジオメトリを正しく設定する所から FreeBSD を再インストールする必要があります。

FreeBSD 専用のシステムで、 他のオペレーティングシステムとの互換性が必要なければ、インストーラのパーティションエディタで A) を選択して、ディスクのすべてを使ってください。

問: システムが ed(4) ネットワークカードを検出しましたが、デバイスのタイムアウトエラーが出ます。

答: 使用しているカードの IRQ が、`/boot/device.hints` で指定された IRQ と異なる可能性があります。ed(4) ドライバは、デフォルトでは「ソフトウェア」コンフィグレーションを用いませんが、インターフェースのヒント情報に `-1` を指定した場合には、ソフトウェアコンフィグレーションが用いられます。

カードのジャンパを変更してコンフィグレーションの設定を行うか、
`hint.ed.0.irq="-1"` と指定して IRQ に `-1` を設定してください このように設定すると、カーネルがソフトウェアコンフィグレーションを利用します。

その他として、利用しているカードが IRQ 9 を使っている可能性があります。この IRQ は IRQ 2 と共有され、特に、IRQ 2 を使っている VGA カードを持っている場合には、しばしば問題を引き起こします。できるだけ IRQ 2 や 9 を避けるべきでしょう。

問: Xorg 端末から sysinstall(8) を使うと、ライトグレーの背景にフォントが黄色で表示され、読みずらいです。このアプリケーションのコントラスト を高くすることはできますか?

答: x11/xterm や x11/rxvt 使用時における sysinstall(8) のデフォルトのカラーが読みにくい場合には、`XTerm*color7: #c0c0c0` を `~/.Xdefaults` に追加して、より暗いグレーの背景に変更してください。

3.12. 高度なインストールガイド

##: Valentino Vaschetto.
##: Marc Fonvieille.

この節では特殊な状況での FreeBSD のインストール方法が記述してあります。

3.12.1. FreeBSD をモニタやキーボードのないシステムにインストールする

この形式のインストールは 「表示のないインストール」 と呼ばれます。 なぜなら、FreeBSD をインストールしようとするマシンにモニタが接続されていないか、 VGA 出力をもっていないからです。このようなシステムへのインストールはシリアルコンソールを使い、 別のマシンをシステムの主ディスプレイやキーボードとして使うと可能になります。これを行うには、「ブートメディアの準備」の説明に従い、インストール用の USB メモリスティックを作成するか、「インストール ISO の作成」で説明されているインストール用の ISO イメージをダウンロードしてください。

以下の説明に従って、 これらのメディアをシリアルコンソールを使って起動するように変更してください CD/DVD メディアを使うのであれば、最初の手順を飛ばしてください。

1. インストール用 USB スティックのシリアルコンソールへの起動を有効にする

   デフォルトでは、USB スティックで起動すると、インストーラを起動します シリアルコンソールを使って起動するには、 os; のシステムに mount(8) コマンドを使って USB ディスクをマウントしてください。

   ```
   # mount /dev/da0a /mnt
   ```

   注記

   デバイスノードおよびマウントポイントの部分は、利用している環境に合わせてください。

   スティックをマウントしたら、USB スティックをシリアルコンソールを使って起動するように設定してください。以下の行を USB スティックの `/boot/loader.conf` に追加してください。

   ```
   # echo 'console="comconsole"' >> /mnt/boot/loader.conf
   ```

   USB スティックを正しく設定することができたら、umount(8) を使ってディスクをアンマウントしてください。

```
# umount /mnt
```

その後 USB スティックを抜き、3 番目の手順に進んでください。

2. インストール用 CD/DVD のシリアルコンソールへの起動を有効にする

デフォルトでは、インストール用の CD/DVD で起動すると、FreeBSD は通常のインストールモードで起動します。シリアルコンソールで起動するには、ISO イメージを CD/DVD メディアに焼く前に、展開して、変更し、再び ISO イメージを作成する必要があります。

インストール用の ISO イメージが保存されている FreeBSD システムで、以下のようにすべてのファイルを tar(1) を使って展開してください。

```
# mkdir /path/to/headless-iso
# tar -C /path/to/headless-iso  -pxvf FreeBSD-10.0-
RELEASE-i386-disc1.iso
```

次に、 インストールメディアをシリアルコンソールを使って起動するように設定します。以下の行を展開した ISO イメージの `/boot/loader.conf` に追加します。

```
# echo 'console="comconsole"' >>  /path/to/headless-
iso/boot/loader.conf
```

その後、修正されたツリーを使って新しい ISO イメージを作成します。 この例では、sysutils/cdrtools package または port の mkisofs(8) を使っています。

```
# mkisofs -v -b boot/cdboot -no-emul-boot -r -J -V
 "Headless_install " \
     -o Headless-FreeBSD-9.3-RELEASE-i386-disc1.iso /
path/to/headless-iso
```

適切な ISO イメージを作成できたので、お気に入りの CD ライタ用のアプリケーションを使って CD/DVD に焼いてください。

3. ヌルモデムケーブルの接続

2 つのマシン間のシリアルポートを、ヌルモデムケーブルで接続してください。普通のシリアルケーブルはここでは使えません。ヌルモデムケーブルが必要です。

4. インストールのために起動

いよいよインストールを開始します。USB メモリスティックまたは CD/DVD メディアを、画面のない、インストールを行おうとしているマシンに挿入し、電源を投入してください。

5. 画面がないマシンへの接続

cu(1) でインストールを行いたいマシンに接続してください:

```
# cu -l /dev/cuau0
```

cu(1)を通して画面がないマシンを操作することができます。カーネルを読み込み、その後どの種類の端末を使うか選択する画面が表示されます。FreeBSD カラーコンソールを選んでインストールを続行してください!

3.13. カスタムインストールメディアの準備

ある状況では、カスタマイズした FreeBSD インストールメディアやソースを用意する必要があります。 これは sysinstall(8) がインストールファイルを持ってくることのできる、 物理的なメディアであったり、ソースの場合もあります。 たとえば、以下のような状況です。

- たくさんのマシンが繋がったローカルネットワークに FreeBSD インストールファイルを持つローカル FTP サーバをホストし、各マシンはインストールにこのサーバのファイルを使うようにします。
- FreeBSD は CD/DVD ドライブを認識できないが、Windows® はそれができるとします。この場合、同じコンピュータ上の Windows® パーティションに FreeBSD のインストールファイルをコピーして、FreeBSD のインストールにこれらのファイルを使います。
- インストールを行いたいコンピュータが CD/DVD ドライブやネットワークカードを搭載してなくても、ヌルプリンタケーブルを用いてそのコンピュータに接続できます。
- 磁気テープを FreeBSD のインストールに使えます。

3.13.1. インストール ISO の作成

FreeBSD プロジェクトは、それぞれのリリースに対して 「ISO イメージ」 をサポートされているアーキテクチャごとに用意します。 書き込み用のアプリケーションを用いてイメージを CD または DVD メディアに書き込む (「焼く」) ことができ、 それを使って FreeBSD をインストールできます。CD/DVD ライタを利用できるのであれば、これが FreeBSD をインストールする最も簡単な方法です。

1. 適切な ISO イメージのダウンロード

   それぞれのリリースの ISO イメージは、 `ftp://ftp.FreeBSD.org/pub/FreeBSD/ISO-IMAGES-` *`arch/version`* もしくは近くのミラーからダウンロードできます。ここで、*`arch`* と *`version`* の部分は適切なものに置き換えてください。

   通常、このディレクトリには以下のイメージが置いてあります。

表3.4 FreeBSD ISO イメージの名前と内容

ファイル名	内容
FreeBSD-*version*-RELEASE-*arch*-bootonly.iso	CD-ROM ドライブから起動することでインストールを行うことができる CD イメージです。この CD のみを使って FreeBSD をインストールすることはできません。この CD から起動した後は、FTP サーバなどからネットワーク経由でインストール作業をします。
FreeBSD-*version*-RELEASE-*arch*-dvd1.iso.gz	この DVD イメージには、FreeBSD オペレーティングシステムのベースシステム、コンパイル済みの package 一式、ドキュメントをインストールするために必要なものが入っています。「livefs」ベースのレスキューモードで起動することも可能です。
FreeBSD-*version*-RELEASE-*arch*-memstick.img	USB メモリスティックに書き込めるイメージです。USB ドライブから起動可能なコンピュータで、インストールに使うことができます。「livefs」ベースのレスキューモードで起動することも可能です。package は、ドキュメントに関連するもののみが入っています。
FreeBSD-*version*-RELEASE-*arch*-disc1.iso	USB メモリスティックに書き込めるイメージです。USB ドライブから起動可能なコンピュータで、インストールに使うことができます。bootonly.iso イメージと同様に、配布物は含まれていませんが、(ftp などの) ネットワーク経由でのインストールに対応しています。
FreeBSD-version-RELEASE-arch-disc1.iso	この CD イメージには、FreeBSD オペレーティングシステムのベースシステムとドキュメントインストールするために必要なものが入っています。その他の package は含まれていません。
FreeBSD-*version*-RELEASE-*arch*-disc2.iso	この CD イメージには、ディスクに収まる容量のサードパーティ製 package が含まれています。このイメージは、FreeBSD *9.X* 以降では利用できません。

ファイル名	内容
FreeBSD-*version*-RELEASE-*arch*-disc3.iso	ディスクに収まる容量のサードパーティ製 package を含むもう 1 つの CD イメージです。このイメージは、 FreeBSD 9.*X* 以降では利用できません。
FreeBSD-*version*-RELEASE-*arch*-livefs.iso	「livefs」 ベースのレスキューモードで起動できるイメージです。 この CD のみを使ってシステムをインストールすることはできません。

CD からインストールを行うには、**bootonly** ISO イメージまたは **disc1** のどちらかをダウンロードしてください。**disc1** には、**bootonly** ISO イメージに含まれているものはすべて含まれているので、両方をダウンロードする必要はありません。

インターネット経由でネットワークからインストールを行うには、**bootonly** ISO を使ってください。 追加のソフトエアが必要であれば、 5章アプリケーションのインストール - packages と ports に説明されているように Ports Collection を用いてインストールできます。

FreeBSD をインストールし、サードパーティ製の package をディスク上から利用したいのであれば、**dvd1** を使ってください。

2. メディアへイメージを書き込む

次に、ダウンロードしたイメージをディスクに書き込んでください。他の FreeBSD システムで書き込みを行う場合には、「burncd」および「cdrecord」の手順を参考にしてください。

他のプラットフォーム上で書き込みを行う場合には、 そのシステムにインストールされている書き込み用のユーティリティを利用してください。 提供されているイメージは標準の ISO フォーマットです。多くの CD ライタ用のアプリケーションがこのフォーマットに対応しています。

注記

カスタマイズした FreeBSD のリリースを構築するには、Release Engineering Article をご覧ください。

3.13.2. FreeBSD ディスクを使ったローカル FTP サイトの作成

FreeBSD のディスクは FTP サイトと同じ配置がなされています。そのため、ローカルな FTP サイトを構築して、ネットワーク上の他のマシンが FreeBSD をインストールするのに使うようにできます。

1. FTP サイトのホストとなる FreeBSD のコンピュータに、CD/DVD をドライブに入れ、マウントしてください。

```
# mount /cdrom
```

2. anonymous FTP のアカウントを作成してください。vipw(8) を使用して次の行を加えてください。

```
ftp:*:99:99::0:0:FTP:/cdrom:/nonexistent
```

3. FTP サービスを請け負うために `/etc/inetd.conf` で有効にしてください。

このマシンにネットワークで繋がっているマシンは、 インストール時にメディアタイプで FTP を選択し、FTP サイトメニューで「Other」を選んだ後 **ftp://*your machine*** と打ち込むことができます。

注記

もし FTP クライアントの起動メディアの FreeBSD のバージョンが FTP サイトで提供されているバージョンと正確に同じでなければ、sysinstall(8) でのインストールはうまくいきません。バージョンを書き換えるには、Options メニューで distribution name を any に変更してください。

警告

この方法はローカルネットワーク上の、 ファイアウォールで保護されたマシンでは容認できますが、 インターネットを介して他のマシンに FTP サービスを提供することは、セキュリティリスクが高まります。インターネット経由でサービスを提供する場合には、セキュリティの状態に注意することを強く提言します。

3.13.3. Windows® パーティションからのインストール

ハードディスクの Windows® パーティションからインストールするときには、配布ファイルを例えば `c:\freebsd` などそのパーティションのルートディレクトリにコピーします。ディレクトリ構造を反映してコピーしなければならないので、CD/DVD からコピーする場

合には robocopy の使用をおすすめします。例えば、最低限の FreeBSD のインストールを行うには、次のように準備します:

```
C:\> md c:\freebsd
C:\> robocopy e:\bin c:\freebsd\bin\ /s
C:\> robocopy e:\manpages c:\freebsd\manpages\ /s
```

ここで C: ドライブには十分な空き容量があり、CD/DVD は E: ドライブにマウントされているものとします。

あるいは、ftp.FreeBSD.org から配布ファイルをダウンロードしてください。それぞれの配布ファイルは独自のディレクトリに入っています。例えば、base 配布ファイルは 9.3/base/ ディレクトリにあります。

Windows® パーティションの c:\freebsd にインストールの配布ファイルをコピーしてください。base および kernel 配布ファイルは、最低限必要です。

3.13.4. ネットワーク経由のインストールの前に

3 種類のネットワークインストールを行うことができます。 イーサネット、PPP、そして、PLIP です。

最も高速なネットワークインストールを行うには、 イーサネットアダプタを使ってください。FreeBSD はきわめて多くの PC イーサネットカードをサポートしています。サポートされているカードの一覧は、 FreeBSD それぞれのリリースのハードウェアノートで提供されます。 サポートされている PCMCIA イーサネットカードを使う場合、 システムの電源を入れる前に差し込んでおくことに注意してください。FreeBSD はインストール時の PCMCIA カードの活線挿抜には対応していません。

静的に設定する場合には、システムの IP アドレス、サブネットマスク、ホスト名、デフォルトゲートウェイアドレス、DNS サーバアドレスを知っておくことが必要です。HTTP プロキシ経由で FTP インストールを行いたい場合には、プロキシサーバのアドレスも必要となります。 これらの情報を知らない場合には、 イーサネット経由でのインストールを始める前に、ネットワーク管理者か ISP に相談するべきでしょう。

もしダイアルアップモデムを使用しているなら、 インストールプロセスの初期にサービスプロバイダの PPP の情報が必要となるので、この情報を手元に用意してください。

もし ISP に接続するのに PAP や CHAP を用いていて、スクリプトを使用していないのであれば、 dial と FreeBSD の ppp のプロンプトに対して入力してください。それ以外の場合、モデム固有の「AT コマンド」を使って ISP にダイヤルする方法を知っておく必要があります。 PPP ダイヤラはとてもシンプルな端末エミュレータしか提供していません。 これ以上の情報については「ユーザ ppp の利用」 や http://www.FreeBSD.org/doc/ja_JP.eucJP/books/faq/ppp.html を参照してください。set log local ... コマンドを用いてログを画面に吐くこともできます。

FreeBSD の動いている別のマシンと直接接続が可能でしたら、 ヌルモデムパラレルポートケーブルを介してインストールすることもできます。パラレルポート経由のデータ転送スピードは、シリアルラインでの一般的なスピードよりも高速です。

3.13.4.1. NFS 経由のインストールの前に

NFS インストールはまったく単純明快です。FreeBSD の配布ファイルを NFS サーバの好きな場所にコピーしておいて、インストーラのメディア選択で NFS を選択してください。

もしサーバが「特権ポート」へのアクセスのみをサポートしている場合、 インストールを進める前に Options メニューで `NFS Secure` を選択してください。

イーサネットカードの性能が悪く、転送速度が遅くて困っている場合には、`NFS Slow` フラグを選択するといいでしょう。

NFS 経由でインストールするためには、サブディレクトリも含んだマウントにサーバが対応している必要があります。例えば FreeBSD 10.0 の配布ファイルが `ziggy:/usr/archive/stuff/FreeBSD` , にあるとすると、マシン `ziggy` では `/usr` や `/usr/archive/stuff` だけでなく `/usr/archive/stuff/FreeBSD` の直接マウントが可能になっていなければなりません。

FreeBSD の `/etc/exports` ファイルでは、このことは `-alldirs` によって制御されています。他の NFS サーバの場合だとまた話が違ってくるかもしれません。もしサーバから permission denied が返ってくるようでしたら、サブディレクトリマウントをちゃんと有効にできていないことが考えられます。

第4章 UNIX の基礎知識

改訂Chris Shumway.

訳: 中井 幸博 <nakai@mlab.t.u-tokyo.ac.jp >, 1996 年 10 月 12 日.

4.1. この章では

この章では FreeBSD オペレーティングシステムの基本的なコマンドと機能について記述しています。ここに書かれてあることのほとんどは、どんな UNIX® オペレーティングシステムにもあてはまります。この章に書いてあることに馴染みがあるなら、この章は気軽に流し読みしてください。あなたが FreeBSD の初心者なら、何か質問する前にこの章を読んでおいた方がきっといいはずです。

この章を読んで分かることは、次のようなことです。

- FreeBSD の「仮想コンソール」の使い方
- UNIX® のファイルの許可属性の仕組みと FreeBSD のファイルフラグについて
- FreeBSD のファイルシステムの構成
- FreeBSD のディスク構成
- ファイルシステムをマウント、アンマウントする方法
- プロセス、デーモンとシグナルとはなにか
- シェルとはなにか。また、デフォルトのログイン環境を変える方法
- テキストエディタの基本的な使い方
- デバイスおよびデバイスノードとはなにか
- さらに詳しい情報を得るためのマニュアルページの読み方

4.2. 仮想コンソールと端末

FreeBSD は様々な使い方ができます。その中の一つが、テキスト端末でコマンドを入力することです。この方法で FreeBSD を使えば、UNIX® オペレーティングシステムの能力と柔軟性を手にすることができます。この節では、「コンソール」と「端末」はどのようなもので、FreeBSD でどう使うかを 説明します。

4.2.1. コンソール

起動時に自動的にグラフィカルな環境が起動するように FreeBSD を設定していなければ、システムが起動してスタートアップ スクリプトが実行されると、すぐにログインプロンプトが出てくるでしょう。次のようものが表示されるはずです。

```
Additional ABI support:.
Local package initialization:.
Additional TCP options:.

Fri Sep 20 13:01:06 EEST 2002

FreeBSD/i386 (pc3.example.org) (ttyv0)

login:
```

あなたのシステムではメッセージが多少異なるかもしれませんが、似たようなものが見られるはずです。最後の 2 行が、今関心を向けているものです。最後から 2 行目は、以下のようになっています。

```
FreeBSD/i386 (pc3.example.org) (ttyv0)
```

この行には、起動したばかりのシステムについていくばくかの情報があります。あなたは、x86 アーキテクチャ上の Intel または その互換プロセッサ上で動作している「FreeBSD」の コンソールを目にしているのです[1]。このマシンの名称 (どの UNIX® 機にも名前がついて います) は `pc3.example.org` で、あなたはそのシステムコンソール、`ttyv0` 端末に向かっています。

最後の行は、常に以下のものになります。

```
login:
```

ここは、FreeBSD にログインするために「ユーザ名」を入力するところです。次の節でどうするか説明します。

4.2.2. FreeBSD にログインする

FreeBSD は、マルチユーザ、マルチプロセスなシステムです。これは、1 台のマシンで何人もの人が交互に多くのプログラムを 動かせるシステムに与えられる正式な説明です。

あらゆるマルチユーザシステムには、ある「ユーザ」を他のユーザと区別する何がしかの手段が必要です。FreeBSD (とすべての UNIX® like なオペレーティングシステム) では、すべてのユーザに対してプログラムの実行を可能にするのに、システムに「ログイン」することを義務付けてこれを実現しています。どのユーザにも、一意な名前 (「ユーザ名」) と個人的な秘密の鍵 (「パスワード」) があります。FreeBSD はユーザにプログラムの実行を許可する前に、この 2 つの入力を要求します。

[1] `i386` が意味しているのはそういうことです。FreeBSD を Intel の 386 CPU 上で動かしていなくても、ここは `i386` になります。ここで表示されるのはプロセッサの種類ではなく、プロセッサの「アーキテクチャ」です。

FreeBSD が起動してスタートアップスクリプトを実行し終わった 直後に[2]、プロンプトを表示して有効なユーザ名の入力を促します。

```
login:
```

この例では john というユーザ名を使う ことにしましょう。このプロンプトに対して john と入力して、Enter を 押してください。そうすると、次のような「パスワード」の入力を要求するプロンプトが 表示されます。

```
login: john
Password:
```

それでは john のパスワードを入力して Enter を押してください。パスワードは 表示されません。これについては、当面は 気にする必要はありません。セキュリティのためといえば十分でしょう。

パスワードを正確に入力したら、FreeBSD にログインして 利用可能なすべてのコマンドを試せるようになっているはずです。

MOTD、もしくはコマンドプロンプト (#, $ または %) に表示されるメッセージを読むようにしましょう。これは FreeBSD へのログインに成功したときに表示されます。

4.2.3. 多数のコンソール

一つのコンソールで UNIX® コマンドを動かすのは結構なことですが、FreeBSD は多くのプログラムを一度に動かせます。コマンドを入力できるコンソールが一つというのは、FreeBSD のようにいくつものプログラムを同時に動かせる オペレーティングシステムの場合は少しもったいないことです。ここで、「仮想コンソール」が非常に役に立ちます。

FreeBSD は、異なる仮想コンソールを複数 表示するように設定できます。キーボード上である組合せのキーを押せば、 その中の一つから他の仮想コンソールのどれかに切り替えられます。 それぞれのコンソールは、個別の出力チャンネルを持っており、 また FreeBSD はある仮想コンソールから次に切り替えるのに応じて、キーボード入力とモニター出力を適切につなぎ直します。

FreeBSD は、コンソールを切り替えるために、 特別なキーの組合せを予約しています[3]。FreeBSD では Alt+F1, Alt+F2 から Alt+F8 までを、別の仮想コンソールに切り替えるのに使えます。

---

[2] スタートアップスクリプトは、起動時に FreeBSD が自動的に実行するプログラムです。主な機能は、全プログラムが動作するように設定を行なうことと、 バックグラウンドで動作するように設定した 有用なサービスを開始することです。

[3] FreeBSD のコンソールとキーボードドライバの詳細全体に ついて、それなりに技術的かつ正確な説明は syscons(4), atkbd(4), vidcontrol(1) および kbdcontrol(1) のマニュアルにあります。ここではその詳細には立ち入りませんが、興味をもった方は、いつでもマニュアルを参照して、動作に関する詳細な説明を読むことができます。

あるコンソールから他に切り替えるのに応じて、FreeBSD は画面 への出力を保存して戻します。結果として、FreeBSD で動かすコマン ドを入力するのに使える複数の画面とキーボードを「仮想的に」実現できるのです。ある仮想コンソールで実行したプログラムは、そのコンソールが見えなくなっている時も実行を停止しません。別の仮想コンソールに切り替えても動き続けます。

4.2.4. /etc/ttys ファイル

初期設定では、FreeBSD は 8 つの仮想コンソールを立ち上げます。 この設定はもともと埋め込まれているわけではなく、インストールしたものが、もっと多いまたは少ない数の仮想コンソールで 起動するように、容易にカスタマイズできます。仮想コンソールの数と 設定は /etc/ttys ファイルに書かれています。

FreeBSD の仮想コンソールを設定するには /etc/ttys ファイルを利用します。このファイルのコメントアウトされていない (# 文字で始まっていない) 行は、一つの端末または仮想コンソールの 設定があります。FreeBSD の初期設定では、仮想コンソールを 9 つ設定し、そのうち 8 つを有効にしています。ttyv で始まる行がそれです。

```
# name  getty                           type    status          ↺
comments
#
ttyv0   "/usr/libexec/getty Pc"         cons25  on  secure
# Virtual terminals
ttyv1   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv2   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv3   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv4   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv5   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv6   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv7   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv8   "/usr/X11R6/bin/xdm -nodaemon"  xterm   off secure
```

このファイルのそれぞれのカラムと仮想コンソールに設定可能な 全オプションの詳しい説明は、ttys(5) のマニュアルを 参照してください。

4.2.5. シングルユーザモードのコンソール

「シングルユーザモード」とは何かという詳しい説明は、「シングルユーザモード」にあります。FreeBSD を シングルユーザモードで動かしている場合は一つしかコンソールが ないということは注意しておくに値するでしょう。仮想コンソールは 利用できません。シングルユーザモードのコンソールの設定は、同じく /etc/ttys ファイルにあります。console で始まる行を探してください。

```
# name  getty                           type    status          ↺
comments
#
# If console is marked "insecure", then init will ask for the root ↺
password
```

```
# when going to single-user mode.
console none                            unknown off secure
```

注記

`console` 行の上のコメントが示すように、この行を編集して `secure` を `insecure` に変更できます。そうすると、FreeBSD がシングルユーザモードで起動した場合にも `root` のパスワードを要求します。

これを **`insecure`** に変更する場合は十分注意してください。`root` のパスワードを忘れてしまったら、シングルユーザモードで起動するのは少しややこしくなります。できることはできますが、FreeBSD の起動の過程とそれに関わる プログラムにあまり親しんでいない人には少し難しいかも知れません。

4.2.6. コンソールのビデオモードの変更

FreeBSD のデフォルトのビデオモードは 1024x768 や 1280x1024 など、グラフィックカードとディスプレイが対応しているサイズに調整されます。別のビデオモードを使うには、以下の 2 つのオプションを有効にしてカーネルを再コンパイルする必要があります。

```
options VESA
options SC_PIXEL_MODE
```

1 度このオプションを有効にしてカーネルを再コンパイルしたら、あなたのハードウェアがどのビデオモードに対応しているか、vidcontrol(1) を用いて知ることができます。以下を実行すると、どのビデオモードに対応しているかを知ることができます。

```
# vidcontrol -i mode
```

このコマンドの出力結果があなたのハードウェアが対応しているビデオモードです。その後 `root` ユーザで vidcontrol(1) を実行することで、新しくどのビデオモードを使うかを選択できます。

```
# vidcontrol MODE_279
```

このビデオモードで良いと思ったら、起動時に自動的に設定されるように `/etc/rc.conf` ファイルに以下のように設定してください。

```
allscreens_flags="MODE_279"
```

4.3. 許可属性

FreeBSD は BSD UNIX® の直系の子孫であり、いくつかの鍵となる UNIX® 思想にもとづいています。まず最も際だった特徴として最初に言えるのは、FreeBSD がマルチユーザのオペレーティングシステムだということです。FreeBSD は同時に働いている複数のユーザすべてを、 完全に分離したタスク上で処理する能力を持っています。また FreeBSD は、ハードウェアデバイス、周辺装置、メモリ、CPU 時間等への要求を、各ユーザが平等に利用できるように適切に共有し、管理する役割を担っています。

システムがマルチユーザをサポートしているため、 システムが管理する資源はすべて、誰がその資源を読み・書き・実行できるかを支配する、一組の許可属性を持っています。これらの許可属性は 3 つの部分からなる 3 桁の 8 進数の形で格納されています。それはそのファイルの所有者 (owner) に対するもの、そのファイルが所属するグループ (group) に対するもの、その他 (others) に対するものの 3 つです。これを数字を使って表現すると、次のようになります。

値	許可属性	ディレクトリの表示
0	読み込み不可、書き込み不可、実行不可	`---`
1	読み込み不可、書き込み不可、実行可能	`--x`
2	読み込み不可、書き込み可能、実行不可	`-w-`
3	読み込み不可、書き込み可能、実行可能	`-wx`
4	読み込み可能、書き込み不可、実行不可	`r--`
5	読み込み可能、書き込み不可、実行可能	`r-x`
6	読み込み可能、書き込み可能、実行不可	`rw-`
7	読み込み可能、書き込み可能、実行可能	`rwx`

ls(1) に対してコマンドライン引数 `-l` を使うと、詳細なディレクトリリストを見ることができ、 ファイルの所有者、グループ、その他への許可属性を示す欄があるのがわかります。例えば、`ls -l` を実行して、適当なディレクトリを表示させると以下のようになります。

```
% ls -l
total 530
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 myfile
```

```
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 otherfile
-rw-r--r--  1 root  wheel    7680 Sep  5 12:31 email.txt
...
```

以下に示すのは、`ls -l` の最初の行を抜き出したものです。

```
-rw-r--r--
```

最初の (一番左の) 文字は、それが 普通のファイルなのか、ディレクトリなのか、キャラクタ型のデバイス特殊ファイルなのか、ソケットなのか、その他の特殊な疑似ファイルデバイスなのかといった種類を示す特別な文字です。この場合、`-` という文字は、普通のファイルであることを示します。この例でその次に来る `rw-` と書かれた 3 文字は、そのファイルの所有者に許可を与えるものです。その次の `r--` の 3 文字は、そのファイルが所属しているグループに許可を与えます。最後の `r--` の 3 文字は、システムに存在するその他のユーザに許可を与えます。「-」は許可が与えられていないことを示します。このファイルの例では、ファイルの所有者はこのファイルを読み書きでき、ファイルの所属しているグループに属するユーザはファイルを読むことだけでき、そのどちらでもないユーザは、このファイルを読むだけできるように許可属性が与えられています。上の表によれば、このファイルに与えられた許可属性は 644 となります。ここで各数字は、このファイルの許可属性の 3 つの部分を表しています。

ファイルについてはここまでの説明で十分です。しかし、デバイスの場合の許可属性はどのようにコントロールされているのでしょうか? FreeBSD は、大部分のハードウェアをファイルとして取り扱います。 そのため、プログラムからは普通のファイルとまったく同じようにオープンし、 データの読み書きができるようになっています。 これらのデバイス特殊ファイルは `/dev` ディレクトリに収められています。

ディレクトリもまた、ファイルと同様に扱われます。 それは読み込み/書き込み/実行の許可属性を持ちます。ディレクトリの実行ビットはファイルのそれとは少し違った意味を持ちます。ディレクトリが実行可能になっているとき、そのディレクトリに移動することができます。つまり、そのディレクトリに「cd」(change directory) することが可能です。また、実行可能属性がついているディレクトリでは、 名前が分かっているファイルにアクセスすることもできます (もちろんそのファイル自体の許可属性によります)。

特に、ディレクトリの中の一覧を表示するには、そのディレクトリに読み込み属性が設定されていなければなりません。 一方、名前が分かっているファイルを削除するには、 そのファイルが含まれているディレクトリに 書き込み属性と実行属性 の両方が必要です。

この他にも許可属性ビットはありますが、いずれも setuid バイナリや sticky ディレクトリなどといった特殊な状況で使われます。ファイルの許可属性そのものについて、また、それらの設定のしかたに関する詳しい情報は、chmod(1) マニュアルページを参照してください。

4.3.1. シンボリック表記

##: Tom Rhodes.

シンボリック表記と呼ばれる許可属性を表す方法では、ファイルやディレクトリの許可属性を、8 進数ではなく記号を用いて設定します。シンボリック表記では、(who), (action), (permissions) という書式が用いられます。利用できる値は以下の通りです。

オプション	文字	意味
(who)	u	ユーザ
(who)	g	ファイルを所持しているグループ
(who)	o	その他
(who)	a	すべて (「world」)
(action)	+	許可属性を与える
(action)	-	許可属性を取り除く
(action)	=	許可属性を指定したものにする
(permissions)	r	読み込み
(permissions)	w	書き込み
(permissions)	x	実行
(permissions)	t	Sticky ビット
(permissions)	s	UID または GID を設定する

これらの値は、これまでと同様に chmod(1) コマンドで用いますが、文字で指定します。たとえば、*FILE* に対して自分以外のユーザからアクセスを一切受け付けたくない、というときには以下のコマンドを実行してください。

```
% chmod go= FILE
```

カンマ区切りで設定することで、ファイルの属性を一度に 2 つ以上変更できます。以下の例では、*FILE* に対して自分以外のユーザから書き込みの権限を取り上げ、かわりにすべてのユーザが *FILE* を実行できるようにします。

```
% chmod go-w,a+x FILE
```

4.3.2. FreeBSD のファイルフラグ

##: Tom Rhodes.

先ほど説明したファイルの許可属性に加え、FreeBSD では「ファイルフラグ」を使えます。これはファイルにセキュリティや管理上の属性を追加するものですが、ディレクトリには追加しません。

これらのファイルフラグはファイルに管理上の属性を追加し、root ユーザでさえ誤ってファイルを消去、変更してしまうことを防ぎます。

ファイルフラグは、chflags(1) を使って、簡単なインタフェースで設定できます。 例えば、file1 というファイルにシステムレベルで消去不可のフラグを設定するには、以下のコマンドを実行してください。

```
# chflags sunlink  file1
```

また、消去不可のフラグを削除するには、以下のように先ほどのコマンドの sunlink の前に「no」をつけるだけです。

```
# chflags nosunlink  file1
```

ファイルにどのフラグが設定されているのかを見るには、ls(1) コマンドを -lo オプションと一緒に使ってください。

```
# ls -lo file1
```

出力は以下のようになります。

```
-rw-r--r--  1 trhodes  trhodes  sunlnk 0 Mar  1 05:54 file1
```

いくつかのフラグの追加、削除は root ユーザしかできません。他のフラグは、ファイルの所有者が変更できます。システム管理者は chflags(1) と chflags(2) から、より詳細な情報を得ることをおすすめします。

4.4. ディレクトリ構造

FreeBSD のディレクトリ構造は、システム全体を理解するに当たって重要です。把握しておくべき最も重要なものは、「/」 ディレクトリです。このディレクトリは起動時に一番最初にマウントされ、オペレーティングシステムをマルチユーザで動作させるために 必要な基本システムが含まれています。また、ルートディレクトリには、マルチユーザへの移行中に他のファイルシステムをマウントするためのマウントポイントも含まれます。

マウントポイントとは、 追加のファイルシステムと接続するためのディレクトリのことです (普通はルートファイルシステムにあります) 。より詳細な説明は「ディスク構成」の節にあります。標準的なマウントポイントには /usr, /var, /tmp, /mnt, /cdrom があります。通常これらのディレクトリについては、/etc/fstab というファイル中のエントリが参照されます。/etc/fstab はさまざまなファイルシステムとマウントポイントの表であり、システムが参照します。/etc/fstab に書かれたファイルシステムは noauto オプションが指定されていなければ、起動時に rc(8) スクリプトによって自動的にマウントされます。詳細は「fstab ファイル」をご覧ください。

ファイルシステム構造を網羅した説明は hier(7) に書かれています。ここでは、もっともよく使われるディレクトリについて簡単に 見るだけで十分でしょう。

ディレクトリ	説明
/	ファイルシステムのルートディレクトリ
/bin/	シングルユーザ環境とマルチユーザ環境の両方で重要な ユーザユーティリティ
/boot/	オペレーティングシステムの起動時に使われるプログラムと設定ファイル
/boot/defaults/	デフォルトの起動設定ファイル; loader.conf(5) 参照
/dev/	デバイスノード; intro(4) 参照
/etc/	システム設定ファイルとスクリプト
/etc/defaults/	デフォルトのシステム設定ファイル; rc(8) 参照
/etc/mail/	sendmail(8) のようなメール転送エージェントの設定ファイル
/etc/namedb/	named 設定ファイル; named(8) 参照
/etc/periodic/	cron(8) 経由で毎日・毎週・毎月実行されるスクリプト; periodic(8) 参照
/etc/ppp/	ppp 設定ファイル; ppp(8) 参照
/mnt/	システム管理者が一時的なマウントポイントとしてよく使う 空のディレクトリ
/proc/	プロセスファイルシステム; procfs(5) と mount_procfs(8) 参照
/rescue/	緊急時のために静的にリンクされているプログラム; 詳しくは rescue(8) 参照
/root/	root アカウントのホームディレクトリ
/sbin/	シングルユーザ環境とマルチユーザ環境の両方で重要な システムプログラムと管理ユーティリティ
/tmp/	一時的なファイル。 通常、/tmp の内容はシステムの再起動で失われます。 メモリファイルシステムはよく /tmp にマウントされます。これは rc.conf(5) の tmpmfs 関係の変数を使うか、/etc/fstab に設定

ディレクトリ	説明
	項目を記入することで自動化できます。詳しくは mdmfs(8) を参照して下さい。
/usr/	大部分のユーザユーティリティとアプリケーション
/usr/bin/	よく使うユーティリティとプログラミングツールとアプリケーション
/usr/include/	C の標準ヘッダファイル
/usr/lib/	ライブラリ
/usr/libdata/	いろいろなユーティリティのデータファイル
/usr/libexec/	システムデーモンとシステムユーティリティ (他のプログラムから実行される)
/usr/local/	ローカルのプログラムやライブラリなど。FreeBSD ports 構成のデフォルトインストール先としても使われます。/usr/local 内では、hier(7) に書かれている /usr のための一般構造が使われます。例外は man ディレクトリで、/usr/local/share の下ではなく /usr/local の下に直接置かれ、ports 関係文書は share/doc/port にあります。
/usr/obj/	/usr/src ツリーのビルドで作られるアーキテクチャ依存のターゲットツリー
/usr/ports	FreeBSD Ports Collection (インストールしなくてもよい)。
/usr/sbin/	(ユーザが実行する) システムデーモンとシステムユーティリティ
/usr/share/	アーキテクチャに依存しないファイル
/usr/src/	BSD のソースファイルまたはローカルのソースファイル、あるいは両方
/usr/X11R6/	X11R6 のプログラム、ライブラリなど (インストールしなくてもよい)
/var/	ログ・一時的なファイル・スプールファイルなどいろいろな用途。メモリファイルシステムは時々 /var にマウントされます。これは rc.conf(5) の varmfs 関係の変数を使うか、/etc/fstab に設定項目を

ディレクトリ	説明
	記入することで自動化できます。 詳しくは mdmfs(8) を参照して下さい。
/var/log/	いろいろなシステムログファイル
/var/mail/	ユーザのメールボックスファイル
/var/spool/	プリンタとメールシステムのスプールディレクトリなどなど
/var/tmp/	一時的なファイル。/var がメモリファイルシステムでなければ、ここにあるファイルはシステムが再起動しても失われません。
/var/yp	NIS のマップ

4.5. ディスク構成

ファイルを見つけるために FreeBSD が使用する構成の一番小さな単位はファイル名です。 ファイル名は、大文字と小文字を区別します。 このことは readme.txt および README.TXT が異なる二つのファイルであることを意味します。FreeBSD はそのファイルがプログラム、または文書、あるいはその他の形式かどうかを決定するために拡張子を使用しません。

ファイルはディレクトリ内に格納されます。 ディレクトリはファイルを一つも含んでいないかもしれせんし、または数百のファイルを含んでいるかもしれません。ディレクトリはまた別のディレクトリを含むことができます。 つまり、ディレクトリの階層構造を構築することができます。このことにより、データ構造がはるかに簡単になります。

ファイルおよびディレクトリは、 必要な他のディレクトリ名とスラッシュ (/) を後に続けてファイル名またはディレクトリ名を与えることによって参照されます。 foo ディレクトリがあって、その中に bar ディレクトリがあるとします。そして、その中に readme.txt があるとすると、ファイルへのフルネーム、またはパスは foo/bar/readme.txt となります。

ディレクトリおよびファイルはファイルシステム内に格納されます。 どのファイルシステムは、そのファイルシステムのための ルートディレクトリ とよばれる、 まさに頂点の位置にちょうど一つのディレクトリを含んでいます。このルートディレクトリは他のディレクトリを含むことができます。

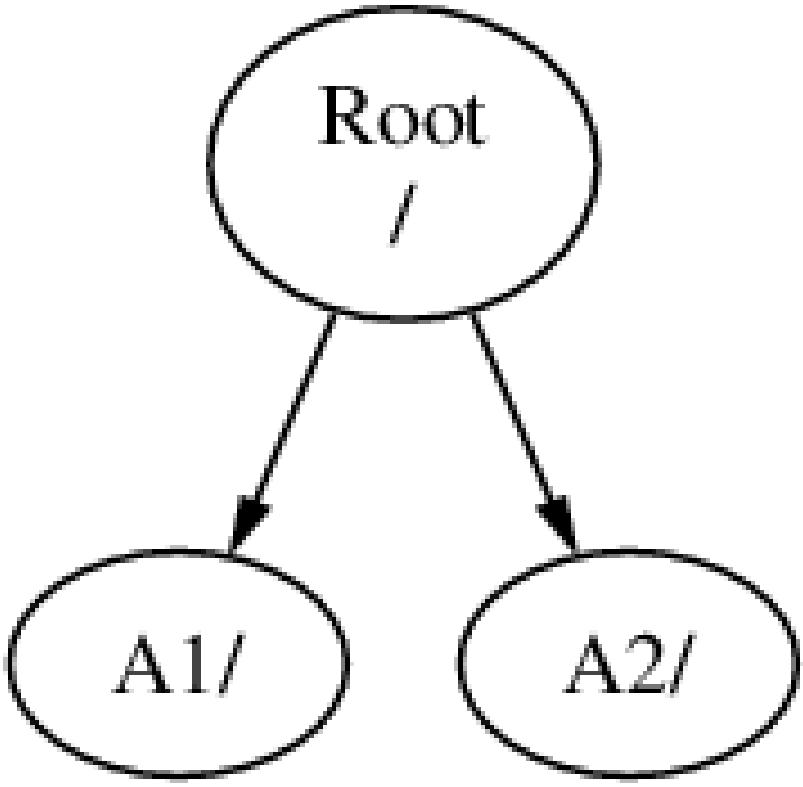
これまでのところ、これはあなたの使ったことのある他の OS とおそらく似ているかもしれません。少し違いがあります。 たとえば、MS-DOS® ではファイル名とディレクトリ名を分けるのに \ を使います。一方、Mac OS® では : を使います。

FreeBSD はパス内にドライブレターまたは他のドライブ名を使いません。 あなたは FreeBSD で `c:/foo/bar/readme.txt` とは書かないでしょう。

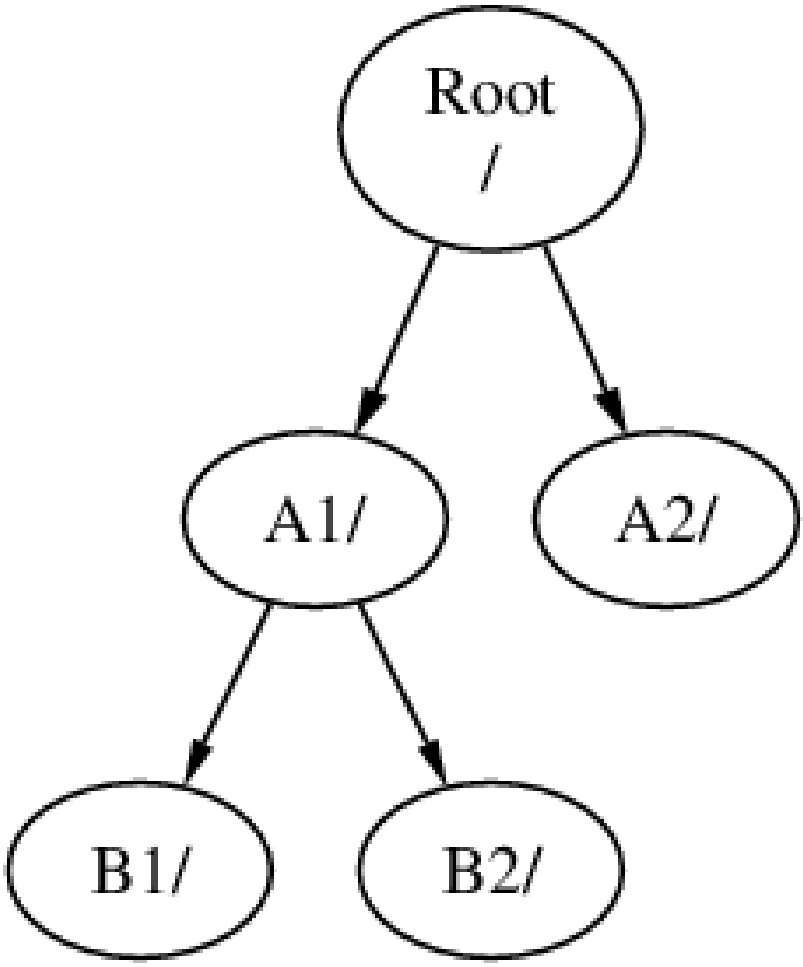
その代わり、一つのファイルシステムは ルートファイルシステムとして設計されています。ルートファイルシステムのルートディレクトリは `/` として参照されます。それから、他のすべてのファイルシステムは、 ルートファイルシステム以下に マウント されます。 あなたが FreeBSD システムでどんなに多くのディスクを使用しても、すべてのディレクトリは、同じディスクの一部であるように見えるので問題ありません。

`A`,`B` および `C` と呼ばれる三つのファイルシステムがあると仮定しましょう。それぞれのファイルファイルシステムには一つのルートディレクトリがあり、`A1`, `A2` と呼ばれている二つの他のディレクトリを含んでいます (同様に `B1`, `B2` および `C1`, `C2` があります)。

`A` をルートファイルシステムとします。 このディレクトリになにが含まれているか見るために `ls` コマンドを使うと、`A1` および `A2` の二つのサブディレクトリが現れるでしょう。ディレクトリツリーは以下のようになります。

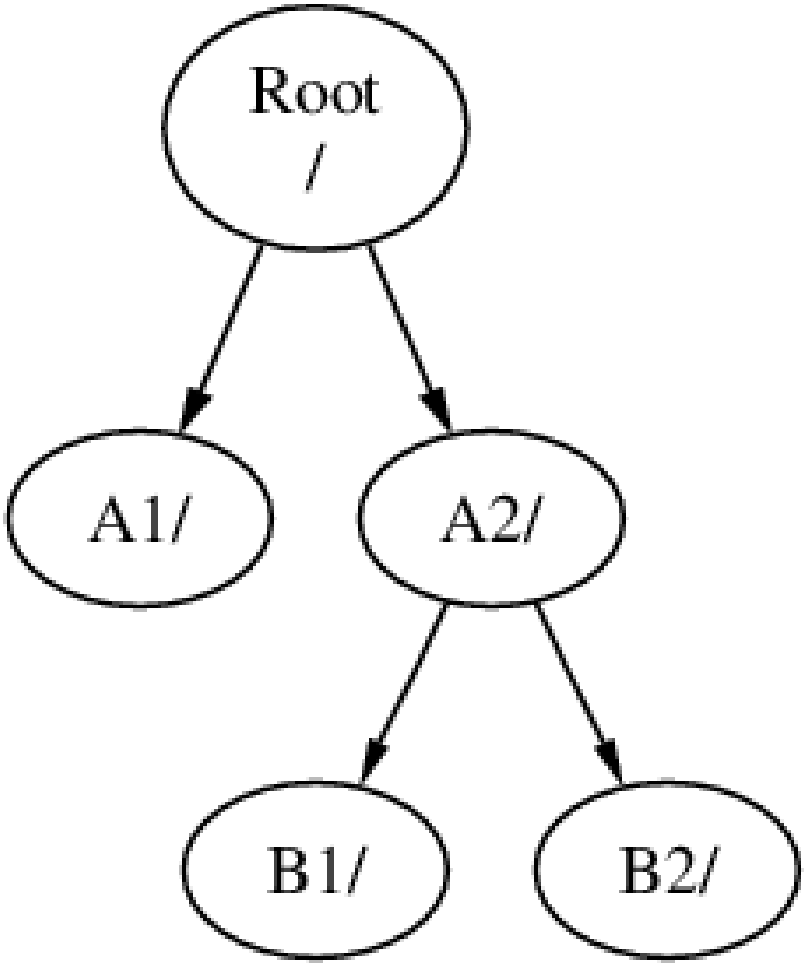


ファイルシステムはファイルシステム内のディレクトリにマウントしなければいけません。それでは、A1 ディレクトリに B ファイルシステムをマウントすると仮定します。B のルートディレクトリは A1 に置き換えられ、そして B 内のディレクトリがそれに応じて現れます。



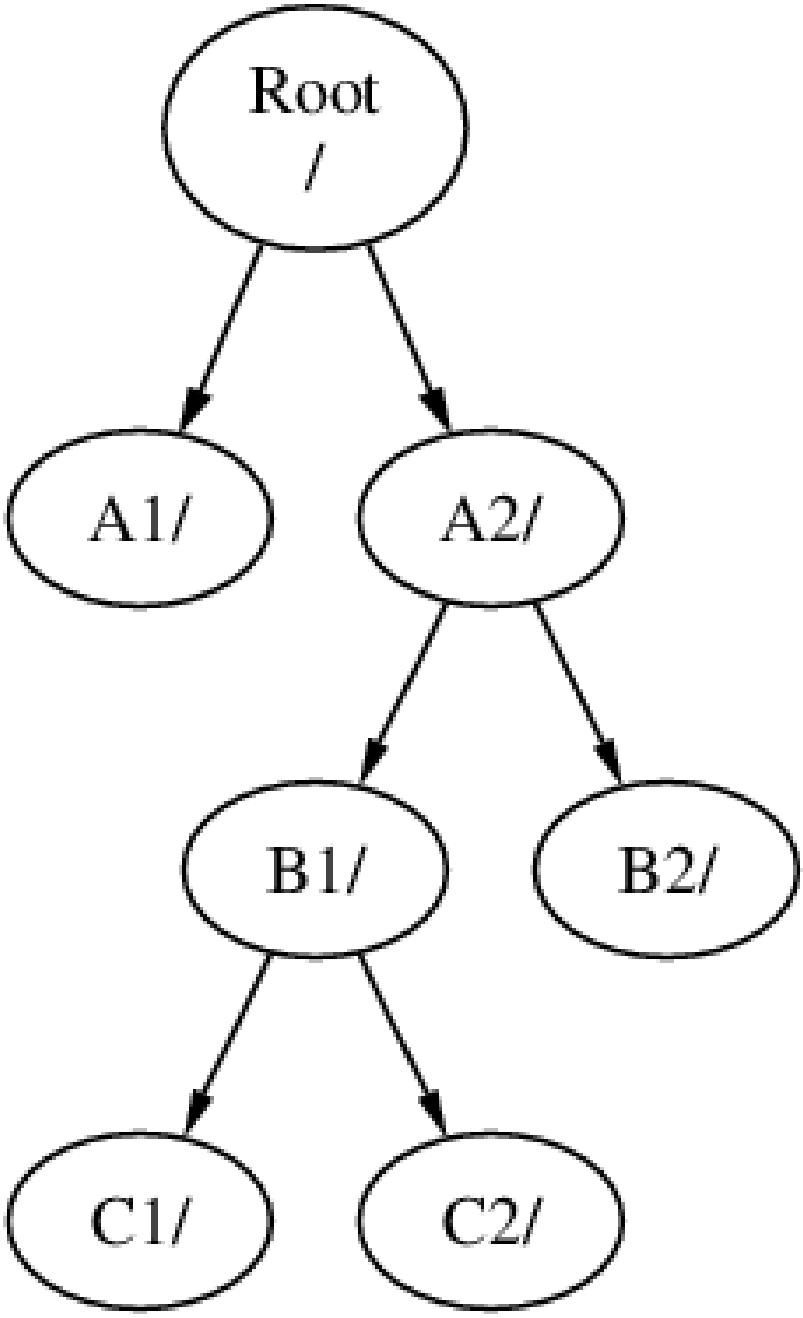
B1 または B2 内にあるどんなファイルも、必要なときに /A1/B1 または /A1/B2 で到達できます。/A1 にあったすべてのファイルは一時的に隠されました。それらは B が A からアンマウントされたら再び現れるでしょう。

もし B が A2 にマウントされていたら、この図のようになります。

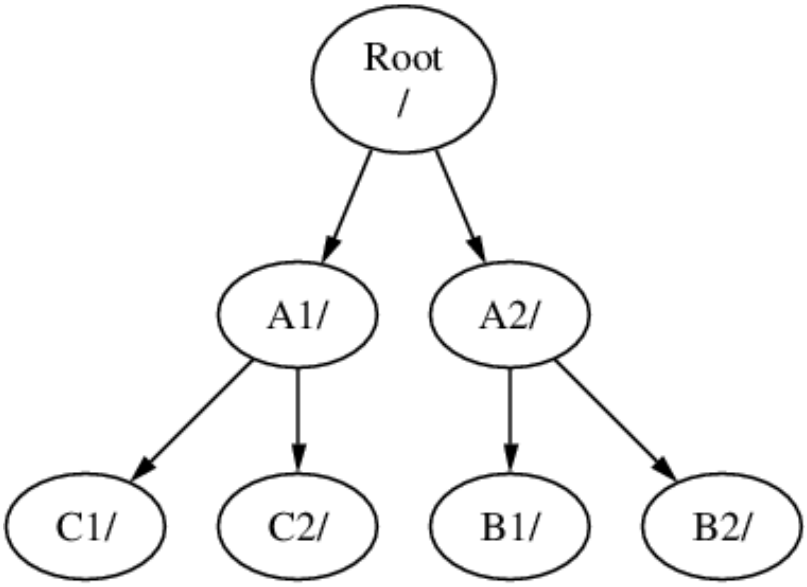


そして、パスはそれぞれ /A2/B1 および /A2/B2 となるでしょう。

ファイルシステムは互いのファイルシステム上にもマウントできます。 上記の最後の例に続けて、C ファイルシステム は B ファイルシステム内の B1 ディレクトリ上にマウントできます。次の図のようになります。


Root
/
A1/
A2/
B1/
B2/
C1/
C2/

または C は A1 の下の A ファイルシステムに直接マウントできます。



もしあなたが MS-DOS® を使いなれているなら、まったく同じではありませんが、これは join コマンドと 似ています。

これは、通常あなた自身が心配する必要のあるものではありません。 一般的に、FreeBSD をインストールするときにファイルシステムを作成し、 どこにマウントするか決定します。そして、 新しいディスクを追加しなければそれらを変更することはありません。

一つの大きなファイルシステムを用意し、 他のファイルシステムを作成する必要としないことはまったくもって可能です。この方法にはいくつかの短所と一つの利点があります。

- 異なったファイルシステムは異なった マウントオプション を使用できます。 たとえば、注意深い考えなのですが、ルートファイルシステムを読みだし専用でマウントして、不注意によって重大なファイルを削除、 または編集できないようにすることができます。また、/home のようなユーザが書き込み可能なファイルシステムを他のファイルシステムと分けることによって、 nosuid でマウントすることも可能になります。このオプションは、ファイルシステムに記録されている suid/guid の実行可能ビットを有効にしないので、安全性を高めることができるでしょう。

- FreeBSD はファイルシステムがどのように使われているかによって、 自動的にファイルシステム上のファイルの配置を最適化します。したがって、連続的に書き込まれた多くの小さなファイルが含まれているファイルシステムは、より大きく少ないファイルが含まれているファイルシステムと異なる最適化をするでしょう。 一つの大きなファイルシステムを作成すると、この最適化は成り立たなくなります。

- FreeBSD のファイルシステムはトラブルが起きてもとても強固です。しかしながら臨界点でのトラブルは、ファイルシステムの構造にまだ損害を与えるかもしれません。マルチファイルシステムへデータを分割しておくことで、 必要なときにバックアップからレストアすることをより容易にして、まだシステムが回復するかもしれません。

- ファイルシステムは固定サイズです。 FreeBSD をインストールするときにファイルシステムを作成して、固定サイズを割りあてたなら、後になってそのパーティションをより大きくする必要があると気づくかもしれません。 パーティションのサイズを変更するには、バックアップ、新しいサイズを指定したファイルシステムの再作成、バックアップしたデータをリストアする作業が必要となるでしょう。

重要

FreeBSD には、growfs(8) コマンドがあります。このコマンドは、この制限を取り除いて、ファイルシステムのファイルを直ちに増加させることを可能にします。

ファイルシステムはパーティション内に含まれています。FreeBSD の UNIX® 遺産のために、 これは普段使われるパーティション (例えば MS-DOS® パーティション) という用語の意味とは違う意味を持っています。 それぞれのパーティションは a から h までの文字で区別されます。 それぞれのパーティションは、 一つのファイルシステムだけを含むことができます。このことは、ファイルシステムがファイルシステムの階層上の典型的なマウントポイント、または含まれているパーティションの文字によって記述されることを意味します。

FreeBSD は スワップ領域 にもまたディスク領域を使用します。 スワップ領域は FreeBSD に 仮想メモリ を提供します。これはあなたのコンピュータが、実際に搭載している以上のメモリがあるかのように振舞います。 FreeBSD がメモリを使い果たしたときに、現在使用されていないデータのいくつかをスワップ領域に移動し、そのデータが必要となったときに (その他のデータをスワップ領域に移動させてから) メモリ内に移動しなおします。

いくつかのパーティションはある慣習と関係づけられています。

パーティション	慣習
a	通常、ルートパーティションを含みます。
b	通常、スワップ領域を含みます。
c	通常、スライス全体と同じサイズです。 これは、スライス全体にアクセス必要のあるユーティリティ (たとえば、ひどいブロックスキャナ) が、c パー

パーティション	慣習
	ティションにアクセスすることを可能にします。通常、 このパーティション内にファイルシステムを作成しないでしょう。
d	d パーティションは、それに関連づけられた特別な意味を持っていましたが、今は無いので、普通のパーティションとして動作するでしょう。

ファイルシステムを含んだそれぞれのパーティションは、FreeBSD が スライス と呼ぶものの中に格納されます。スライスは FreeBSD の用語で、普通はパーティションと呼ばれるものです。もう一度言及しますが、これは FreeBSD の UNIX® 背景によるものです。スライスは 1 から 4 までの番号がつけられます。

スライス番号は 1 から始まり s を前につけられて、デバイス名の後に続きます。したがって、「da0s1」は一番目の SCSI ドライブ上の 一番目のスライスです。ディスク上に四つの物理スライスだけが存在できます。しかし、適切な種類の物理スライス内に論理スライスをもつことができます。これらの拡張されたスライス番号は 5 から始まります。したがって、「ad0s5」は、一番目の IDE ディスク上の一番目の拡張スライスです。これらのデバイスは、スライスを占有することを予期するファイルシステムによって使用されます。

スライスや「危険な専用」の物理ドライブ、そして他のドライブは a から h までの文字として表される パーティション を含んでいます。この文字はデバイス名に追加されます。したがって、「da0a」は一番目の「危険な専用」da ドライブ上の a パーティションです。「ad1s3e」は、二番目の IDE ディスク上の 三番目のスライス内にある五番目のパーティションです。

最後に、システム上のそれぞれのディスクは識別されます。ディスク名はどの種類のディスクであるかを示す記号ではじまり、どのディスクかを示す数字が続きます。 スライスとは違いディスクの番号づけは 0 から始まります。共通の記号は 表4.1「ディスクデバイス記号」 に示されます。

パーティションを参照するときには、 FreeBSD はパーティションを含むスライスおよびパーティションも指定することを必要とします。 そしてスライスを参照するときはディスク名も参照しないといけません。したがって、ディスク名、s、スライス番号、 そしてパーティション文字を並べることによってパーティションを参照します。 例4.1「ディスク名、スライス名、パーティション名のサンプル」に例があります。

例4.2「ディスクの概念的構成」 は理解をより明らかにすることを助けるための、ディスク構成の概念のモデルを示します。

FreeBSD をインストールするために、 まずはじめにディスクスライスの設定をし、 次に FreeBSD に用いるスライス内のパーティションを作成し、それからそれぞれのパーティション内にファイルシステム (またはスワップ領域) を作成し、 ファイルシステムがどこにマウントされるか決定しなければいけません。

表4.1 ディスクデバイス記号

記号	意味
ad	ATAPI (IDE) ディスク
da	SCSI ダイレクトアクセスディスク
acd	ATAPI (IDE) CDROM
cd	SCSI CDROM
fd	フロッピーディスク

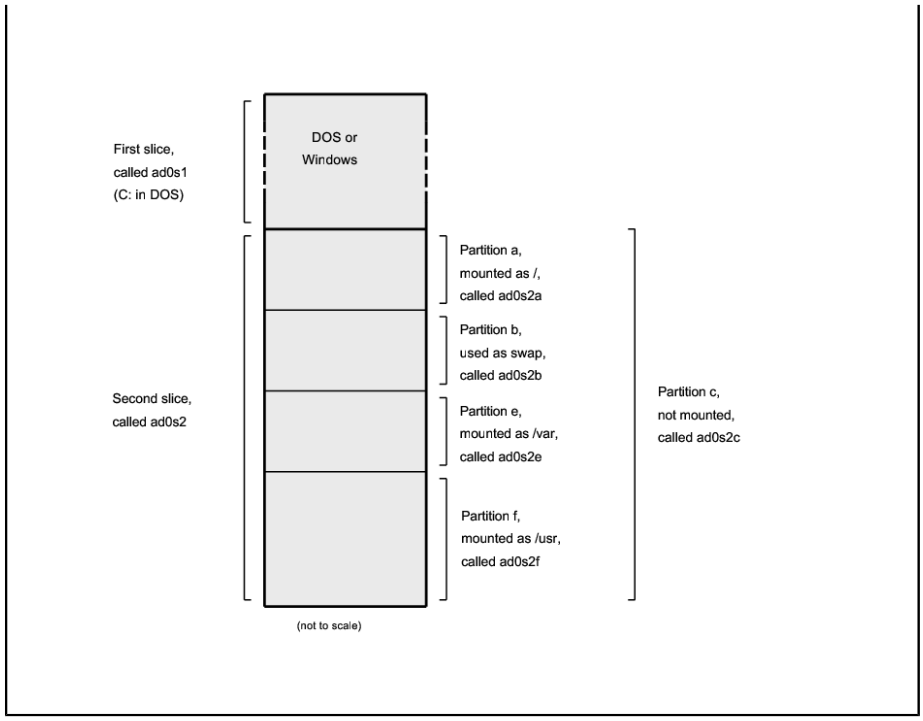
例4.1 ディスク名、スライス名、パーティション名のサンプル

記号	意味
ad0s1a	一番目の IDE ディスク (ad0) 上の一番目のスライス (s1) 内の一番目のパーティション (a)。
da1s2e	二番目の SCSI ディスク (da1) 上の二番目のスライス (s2) 内の五番目のパーティション (e)。

例4.2 ディスクの概念的構成

これはシステムに接続された一番目の IDE ディスクの FreeBSD から見た図を示します。ディスクサイズは 4 GB と仮定し、2 GB のスライス (MS-DOS® でいうパーティション) が二つあるとします。一番目のスライスは MS-DOS® ディスクの C: を含んでいます。そして、二番目のスライスは FreeBSD のディスクを含んでいます。これは FreeBSD インストーラが三つのデータパーティションと一つのスワップパーティションを作成した例です。

三つのパーティションはそれぞれファイルシステムを含んでいます。 a パーティションはルートファイルシステムに使用され、e パーティションは /var ディレクトリ階層に、f パーティションは /usr ディレクトリ階層に使用されるでしょう。



4.6. ファイルシステムのマウントとアンマウント

ファイルシステムは / をルート (根) とする木構造として考えると視覚的に理解しやすいでしょう。ルートディレクトリにある /dev や /usr、その他のディレクトリは枝に相当し、それらには、/usr/local などのように、さらに枝分かれすることができます。

さまざまな理由がありますが、 ディレクトリをいくつかの異なるファイルシステム上に構築するのが良いでしょう。たとえば /var には、log/ や spool/ など、さまざまな種類の一時ファイルを置くディレクトリがあるため、あふれてしまう可能性があります。ルートファイルシステムをあふれさせるのは得策ではありませんので、普通は /var を / から分離します。

また、次のような場合も、ディレクトリツリーを 別のファイルシステムに置く理由として良くあげられます。 それは、たとえば物理的に別のディスクにディレクトリツリーを置く場合、ネットワークファイルシステム (Network File System) や CDROM ドライブのような別の仮想ディスクに置くという場合です。

4.6.1. **fstab** ファイル

/etc/fstab に書かれているファイルシステムは (noauto オプションがなければ) 起動プロセスの途中で 自動的にマウントされます。

/etc/fstab ファイルは、次のような書式で書かれた行のリストになっています。

```
device          /mount-point  fstype     options      dumpfreq
   passno
```

device
: デバイスの名前 (存在していなければなりません)。「デバイス名」 に説明があります。

mount-point
: ファイルシステムがマウントするディレクトリの名前 (存在していなければなりません)。

fstype
: mount(8) に渡されるファイルシステムタイプ。 FreeBSD ファイルシステムのデフォルトは ufs です。

options
: 読み書きするファイルシステムには rw、読み込み専用のファイルシステムには ro を、必要な他のオプションの前に指定します。 よく使われるオプションは noauto で、起動時にはマウントされないファイルシステムに使います。その他のオプションは mount(8) マニュアルページに載っています。

dumpfreq
: これは dump(8) が使うもので、どのファイルシステムにダンプが必要なのかを決めます。この項目がなければ、0 であるものとみなされます。

passno
: これはファイルシステムをチェックする順番を決めます。ファイルシステムチェックを飛ばしたいファイルシステムには、passno を 0 に設定してください。ルートファイルシステム (どれよりも先にチェックしなければなりません) は passno を 1 に設定してください。他のファイルシステムの passno は 1 以上に設定してください。同じ passno のファイルシステムがあった場合、fsck(8) は可能であれば並行してファイルシステムのチェック を行なおうとします。

/etc/fstab ファイルの書式やオプションに関しての詳細は、fstab(5) をご覧ください。

4.6.2. mount コマンド

mount(8) コマンドは、ファイルシステムをマウントするために使われるものです。

基本的には、次のように使います。

```
# mount device mountpoint
```

mount(8) マニュアルページにはたくさんのオプションが書かれていますが、いちばんよく使われるのは次のものです。

`-a`
: `/etc/fstab` にある全てのファイルシステムをマウントします。例外は「noauto」の印がついているものと、`-t` フラグで除外されたものと、すでにマウントされているファイルシステムです。

`-d`
: 実際にマウントシステムコールする以外のすべてのことをします。 このオプションは `-v` フラグと組み合わせて使い、mount(8) が実際なにをしようとしているのか調べるのに便利です。

`-f`
: クリーンでないファイルシステムを強制的にマウントします (危険です)。もしくは、ファイルシステムのマウント状態を 読み書き可能から読み込みのみに変更するとき、書き込みアクセスを強制的に取り消します。

`-r`
: ファイルシステムを読み込み専用でマウントします。 これは `ro` (5.2 より前の FreeBSD では `rdonly` です) 引数を `-o` オプションに使うのと同じです。

`-t` *`fstype`*
: ファイルシステムを指定のファイルシステムタイプでマウントします。 または、`-a` を使った場合、指定したタイプのファイルシステムのみマウントします。

  デフォルトのファイルシステムタイプは「ufs」です。

`-u`
: ファイルシステムのマウントオプションを更新します。

`-v`
: 詳細な出力にします。

`-w`
: ファイルシステムを読み書き可能にマウントします。

`-o` には、次のようなオプションを複数カンマで区切って指定します。以下に挙げるのはその一部です。

noexec
: そのファイルシステム上のバイナリの実行を禁止します。 セキュリティのために有用なオプションです。

nosuid
: そのファイルシステム上の setuid や setgid フラグを解釈しません。これもセキュリティのために有用なオプションです。

4.6.3. umount コマンド

umount(8) コマンドは、パラメータとしてマウントポイントの一つ、デバイス名、もしくは -a や -A といったオプションを取ります。

いずれの形式でも -f で強制的なアンマウントを行ない、 -v で詳細な出力を出します。ただしほとんどの場合、-f は使わないほうがよいでしょう。強制的にファイルシステムをアンマウントすると、 計算機がクラッシュしたりファイルシステム上部のデータが 破壊されたりする恐れがあるためです。

オプション -a と -A はマウントされているファイルシステムすべてをアンマウントするのに使います。-t にファイルシステムタイプを指定すると、指定されたものだけがアンマウントされます。また、-A を使うとルートファイルシステムはアンマウントしません。

4.7. プロセス

FreeBSD はマルチタスクのオペレーティングシステムです。つまり、1つ以上のプログラムがあたかも同時に動いているかのように見える、 ということです。動作中のプログラムはそれぞれ プロセス と呼ばれます。コマンドを実行すると、最低でも1つの新しいプロセスがスタートします。 システムを正常に機能させるために常に動作しているシステムプロセスもたくさんあります。

各プロセスはプロセス ID、もしくは PID と呼ばれる数字でただ一つに識別されます。また、ファイルのように各プロセスには所有者とグループがあります。 所有者とグループの情報は、これまでに見たファイル許可属性を用い、そのプロセスが開けるファイルやデバイスを決定するために使われます。多くのプロセスには親プロセスもあります。親プロセスとは、そのプロセスをスタートさせたプロセスのことです。例えば、シェルにコマンドを打ち込んでいるときはシェルがプロセスで、 動かすコマンドもまたどれもプロセスです。 このようにして起動するプロセスはそれぞれシェルが親プロセスになります。これの例外は init(8) という特別なプロセスです。init は常に最初のプロセスなので、PID は必ず 1 になります。init は FreeBSD がスタートするときカーネルによって自動的に起動されます。

ps(1) と top(1) という2つのコマンドが システム上のプロセスを確認するために特に便利です。ps コマンドは現在動作中のプロセスのリストを見るために使い、PID やプロセ

スが使っているメモリの量、どういうコマンドラインで起動されたのか、などを表示させることができます。top コマンドは動作中の全てのプロセスを表示し、数秒ごとに表示を更新するので、計算機がなにをしているのかインタラクティブに知ることができます。

デフォルトでは、ps は動作中かつ所有者が自分のコマンドのみを表示します。例えば:

```
% ps
  PID  TT  STAT      TIME COMMAND
  298  p0  Ss     0:01.10 tcsh
 7078  p0  S      2:40.88 xemacs mdoc.xsl (xemacs-21.1.14)
37393  p0  I      0:03.11 xemacs freebsd.dsl (xemacs-21.1.14)
48630  p0  S      2:50.89 /usr/local/lib/netscape-linux/navigator-
linux-4.77.bi
48730  p0  IW     0:00.00 (dns helper) (navigator-linux-)
72210  p0  R+     0:00.00 ps
  390  p1  Is     0:01.14 tcsh
 7059  p2  Is+    1:36.18 /usr/local/bin/mutt -y
 6688  p3  IWs    0:00.00 tcsh
10735  p4  IWs    0:00.00 tcsh
20256  p5  IWs    0:00.00 tcsh
  262  v0  IWs    0:00.00 -tcsh (tcsh)
  270  v0  IW+    0:00.00 /bin/sh /usr/X11R6/bin/startx -- -bpp 16
  280  v0  IW+    0:00.00 xinit /home/nik/.xinitrc -- -bpp 16
  284  v0  IW     0:00.00 /bin/sh /home/nik/.xinitrc
  285  v0  S      0:38.45 /usr/X11R6/bin/sawfish
```

この例で分かるとおり、ps(1) の出力はいくつかの行に整形されています。PID は先ほど見たプロセス ID です。PID は 1 から順に 99999 まで割り当てられ、足りなくなると最初に戻って使い回されます (使用中の PID は割り当てられません) 。TT の列はプログラムが動いている tty を示します。差し当たって無視してもかまわないでしょう。STAT はプログラムの状態を示しますが、これもまた無視してよいでしょう。TIME はプログラムがその CPU 上で動いている時間の長さです— 通常はプログラムをスタートさせたときからの経過時間ではありません。CPU 上で時間を使う必要があるまでかなりの時間を費すようなプログラムもあるからです。 最後に、COMMAND はそのプログラムを起動するのに使われたコマンドラインとなります。

ps(1) は表示する情報を変えるためのオプションをたくさんサポートしています。いちばん便利なのは auxww でしょう。a は自分のプロセスだけではなく、動作中のプロセス全部についての情報を表示します。u はプロセスの所有者の名前をメモリ使用量と同様に表示します。x はデーモンプロセスについての情報を表示し、ww で、スクリーンに入りきらないほど長くなったコマンドラインでも省略せず、ps(1) に各プロセスの全コマンドラインを表示させます。

top(1) の出力も同様です。例は以下の通りです。

```
% top
last pid: 72257;  load averages:  0.13,  0.09,  0.03    up 0+13:38:33
  22:39:10
```

```
47 processes:  1 running, 46 sleeping
CPU states: 12.6% user,  0.0% nice,  7.8% system,  0.0% interrupt, ↺
79.7% idle
Mem: 36M Active, 5256K Inact, 13M Wired, 6312K Cache, 15M Buf, 408K ↺
Free
Swap: 256M Total, 38M Used, 217M Free, 15% Inuse

  PID USERNAME PRI NICE  SIZE    RES STATE    TIME   WCPU    CPU ↺
COMMAND
72257 nik       28   0  1960K  1044K RUN      0:00 14.86%  1.42% top
 7078 nik        2   0 15280K 10960K select   2:54  0.88%  0.88% ↺
xemacs-21.1.14
  281 nik        2   0 18636K  7112K select   5:36  0.73%  0.73% ↺
XF86_SVGA
  296 nik        2   0  3240K  1644K select   0:12  0.05%  0.05% ↺
xterm
48630 nik        2   0 29816K  9148K select   3:18  0.00%  0.00% ↺
navigator-linu
  175 root       2   0   924K   252K select   1:41  0.00%  0.00% ↺
syslogd
 7059 nik        2   0  7260K  4644K poll     1:38  0.00%  0.00% mutt
...
```

出力は2つのセクションに分かれています。ヘッダ (最初の 5 行です) は動作している最新のプロセスの PID、システムの平均負荷 (システムがどれくらい忙しいかの指標)、システムの稼働時間 (最後の再起動からの時間) と現在の時刻を示します。ヘッダの中の他の数字は動作中のプロセスの数 (この場合 47 ですね)、使われているメモリとスワップ領域の量、そしてシステムが異なる CPU 状態に消費した時間と関係します。

その下には ps(1) の出力と同じような情報を持った行が続きます。前と同様 PID にユーザ名、消費 CPU 時間と実行中のコマンドを知ることができます。top(1) を使うとデフォルトでプロセスが使っているメモリ容量も分かります。 メモリ使用量の欄は2項目に分かれており、 一方は合計使用量、 そしてもう一方は実使用量です—合計使用量はアプリケーションが必要としているメモリ量で、 実使用量はその時点で実際に使われているメモリ量です。この例では、Netscape® がだいたい 30 MB の RAM を必要としていますが、いまのところ 9 MB しか使っていないことが分かります。

top(1) は自動的に2秒ごとに画面を更新します。s オプションを使えば更新間隔を変更することができます。

4.8. デーモン、シグナルとプロセス終了

エディタを使っている場合、エディタを操作するのは簡単です。 ファイルを開く、などと動かせばよいのです。 このように操作できるのは、エディタにそういった機能があり、 かつエディタが端末に関連づけられているからです。一方、ユーザから始終入力があるように設計されていないプログラムもあり、 そういったプログラムは最初から端末と切り離されます。 例えば、ウェブサーバは一日中ウェブのリクエストばかり処理するので、 通常全く

入力を必要としません。サイトからサイトへとメールを転送するプログラムも、こういった種類のアプリケーションの一例です。

このようなプログラムは、デーモンと呼ばれます。　デーモンはギリシャ神話の登場人物で、善でも悪でもなく、大雑把にいうと、人間のために役立つことをしてくれる小さな妖精さんです。今日の便利なウェブサーバやメールサーバととてもよく似ていますね。このため、長い間 BSD のマスコットはスニーカーをはいてフォークを携えた かわいらしい姿のデーモンなのです。

通常デーモンとして動作するプログラムには末尾に「d」を持った名前をつける慣習があります。BIND は Berkeley Internet Name Domain ですが、実際実行されるプログラムは `named` という名前です。Apache ウェブサーバのプログラムは `httpd` と呼ばれ、ラインプリンタスプーリングデーモンは `lpd`、などなどです。これは単なる慣習で、しっかりがっちりとしたルールではありません。例えば、Sendmail アプリケーションの主なメールデーモンは `sendmail` という名前で、連想しそうな `maild` ではありません。

時々、デーモンプロセスと通信したいときがあります。一つの方法として、それ (に限らずどんな動作中のプロセスでも) に シグナルと呼ばれるものを送信する方法です。送信可能なシグナルはたくさんあります—特別な意味があるものもあれば、アプリケーションによって解釈されるものもありますし、アプリケーションがシグナルをどう解釈するかは そのアプリケーションの文章を読めば分かるでしょう。自分が持っているプロセスにしかシグナルを送ることはできません。他人のプロセスに kill(1) や kill(2) を使ってシグナルを送っても、許可されないでしょう。これの例外は `root` ユーザで、ルートユーザは誰のプロセスでもシグナルを送ることができます。

FreeBSD もアプリケーションにシグナルを送ることがあります。アプリケーションを下手に書くと、予想外のメモリにアクセスしようとするので、FreeBSD がプロセスに セグメンテーション違反 シグナル (`SIGSEGV`) を送ります。ある程度の時間が経ったら alarm(3) システムコールを使って警告してもらうようなアプリケーションには、警告シグナル (`SIGALRM`) が送信される、などです。

プロセスを止めるためには2つのシグナル、`SIGTERM` か `SIGKILL` を使います。`SIGTERM` は穏かにプロセスを終了させる方法です。プロセスはシグナルを受け取ることができ、終了させたいのだなということを理解し、開いているログファイルを全部を閉じ、一般的に終了前にしていたことを終えることができます。中断できない処理の途中だと、`SIGTERM` をプロセスが無視することもあるかもしれません。

プロセスは `SIGKILL` を無視することができません。これは、「なにをしていようが構わないから今すぐ止まれ」というシグナルです。プロセスに `SIGKILL` を送ると、FreeBSD はそのプロセスをそこで止めます[4]。

[4]正確ではありません—中断できないものはわずかながら存在します。例えば、プロセスがネットワーク上の別の計算機にあるファイルを読もうとして、その計算機がなんらかの理由 (電源を落とされたとか、ネットワークに

使う可能性のあるシグナルは、他に SIGHUP、SIGUSR1、と SIGUSR2 があります。これらは一般的な用途のシグナルで、 このシグナルが送信されたときアプリケーションによって別のことをします。

ウェブサーバの設定ファイルを変更したとしましょう—ウェブサーバに新しい設定を再読み込みさせたいですね。 httpd を止めて再起動することもできますが、そうするとウェブサーバは一瞬ながら停止してしまいますし、 ちょっとでも止まってほしくないこともあるでしょう。 ほとんどのデーモンは SIGHUP シグナルに対して設定ファイルを再読み込みする反応を返すよう書かれています。 従って、httpd を止めて再起動する代わりに、SIGHUP シグナルを送りましょう。これらのシグナルへの標準的な反応というものがないために、デーモンごとに行動が違うので、疑問があれば必ずそのデーモンの文書を読んでください。

kill(1) コマンドを使って送るシグナルはこの例をご覧ください。

手順4.1 プロセスにシグナルを送る

この例では、inetd(8) にシグナルを送る方法を示します。 inetd の設定ファイルは /etc/inetd.conf で、inetd は SIGHUP が送信されるとこの設定ファイルを再読み込みします。

1. シグナルを送りたいプロセスのプロセス ID を探します。それには ps(1) と grep(1) を使います。grep(1) コマンドは出力を検索するために使い、指定した文字列を探します。このコマンドは一般ユーザで実行しますが、inetd(8) は root で実行されているので、ps(1) には ax オプションを与える必要があります。

   ```
   % ps -ax | grep inetd
     198  ??  IWs    0:00.00 inetd -wW
   ```

   ということで、inetd(8) の PID は 198 です。grep inetd コマンドがこの出力に出てくる場合もあります。 それは、ps(1) が動作中のプロセスのリストを見つける方法によります。

2. kill(1) を使ってシグナルを送ります。inetd(8) は root で起動されているために、まず su(1) を使って root にならなければなりません。

   ```
   % su
   Password:
   # /bin/kill -s HUP 198
   ```

   大部分の UNIX® コマンドと同じく、 成功したら kill(1) は何の出力も表示しません。 自分のものではないプロセスにシグナルを送ると、 kill: *PID*: Operation

問題があるとか) でいなくなった場合、そのプロセスは「中断不可能」と言われます。 最終的にはそのプロセスはタイムアウトします。普通は2分後です。タイムアウトした直後、そのプロセスは終了します。

not permitted と表示されます。PID を打ち間違えると、悪いことに間違ったプロセスにシグナルを送ってしまうか、もしくは運がよければその時点で使われていない PID にシグナルを送ったことになり、kill: ***PID***: No such process と表示されます。

なぜ `/bin/kill` を使うんでしょう?

多くのシェルは `kill` コマンドを組み込みコマンドとして備えています。 つまり、`/bin/kill` を実行するのではなく、シェルが直接シグナルを送ります。 これはとても便利なのですが、シェルが違うと送るシグナルの名前の指定の仕方が違います。 シェルによって異なるシグナルの指定の仕方を全部覚えようとはせずに、`/bin/kill ...` コマンドを直接使うほうが簡単です。

他のシグナルの送り方はほとんど同じで、コマンドラインの `TERM` や `KILL` を必要に応じて変えるだけです。

重要

システム上のランダムプロセスを終了させるのはよくありません。特に、プロセス ID が 1 の init(8) は特別です。`/bin/kill -s KILL 1` を使うといとも簡単にシステムをシャットダウンさせることができます。Return を押す前に kill(1) を実行する引数を二重にチェックする癖をつけてください。

4.9. シェル

FreeBSD では日々の作業のほとんどは、「シェル」と呼ばれるコマンドラインインタフェイスを通して行われます。 シェルの主な仕事はコマンドを入力チャンネルから受け取り、そしてそれらを実行することです。 大部分のシェルはさらに組み込みの機能を持っていて、日々の作業、ファイル管理やファイル名の展開、コマンドライン編集、コマンドマクロ、環境変数などに便利です。FreeBSD には `sh` (Bourne Shell) や `tcsh` (高機能 C-shell) が含まれています。また、これ以外にも `zsh` や `bash` などたくさんのシェルが FreeBSD Ports Collection から利用可能です。

「あなたは、どのシェルを使いますか?」という質問は、 まったく趣味の問題です。 あなたが C のプログラマだったとすれば、 tcsh のような C 風のシェルの方が落ち着くかもしれません。Linux から来た人や UNIX® のコマンドラインインタフェイスになじみがなければ、 bash を試すのも良いでしょう。ポイントは、それぞれのシェルは、あなたの好みの作業環境で利用できる (もしくはできない) 独自の機能を持っているということ、そして、どのシェルを使うことにするかを決めるのはあなた自身だということです。

シェルの一般的な機能の一つに、ファイル名の補完があります。 コマンドやファイル名の最初の数文字を与えて Tab キーを押すことで、 シェルにコマンドやファイル名の残りの部分を自動的に補完させることができます。例をあげましょう。二つのファイル foobar, foo.bar が あったとします。ここで foo.bar の方を削除するには、rm fo[Tab].[Tab] と入力します。

するとシェルは rm foo[BEEP].bar と出力するでしょう。

[BEEP] のところはコンソールのベル (訳注: 通常はビープ音が鳴ります) です。これは複数のファイルがマッチしたため、 ファイル名の補完を完全に行なえなかったことを伝えています。foobar と foo.bar は 両方とも fo ではじまるため、補完できるのは foo までです。ここで . を入力して Tab を押せば、シェルはファイル名の残りの部分を補完できます。

もう一つあげられるシェルの特徴として、環境変数があります。 環境変数とは、シェルの環境変数空間におけるキーと値とのペアです。 この変数空間は、そのシェルから起動されたプログラムから参照でき、 それを利用してプログラムの設定を保存するのに利用されます。下の表は、一般的な環境変数とその意味を示したものです。

変数名	意味
USER	現在のログインユーザのユーザ名。
PATH	コロンで区切られた実行ファイル探索のための ディレクトリのリスト。
DISPLAY	接続する X11 ディスプレイのネットワーク名 (存在する場合のみ)。
SHELL	現在のシェル。
TERM	ユーザの端末種名。端末のケーパビリティを決定するのに使われる。
TERMCAP	種々の端末の機能を実現する端末のエスケープコードの データベースのエントリ。
OSTYPE	オペレーティングシステムの種別。 たとえば FreeBSD。
MACHTYPE	システムが動作している CPU のアーキテクチャ。

変数名	意味
EDITOR	ユーザの選んだテキストエディタ。
PAGER	ユーザの選んだテキストページャ。
MANPATH	コロンで区切られたマニュアルページ探索のための ディレクトリのリスト。

環境変数をセットする方法は、 それぞれのシェルごとに多少異なります。 たとえば、tcsh や csh 等の C シェルでは setenv を使います。sh や bash 等の Bourne シェルでは set と export を使います。たとえば csh か tcsh で EDITOR 環境変数の値を /usr/local/bin/emacs に セットするか変更するには、次のようにします。

```
% setenv EDITOR /usr/local/bin/emacs
```

Bourne シェルでは次のようになります。

```
% export EDITOR="/usr/local/bin/emacs"
```

ほとんどのシェルでは、コマンドライン中の変数名の前に $ 文字を置くことで、環境変数を展開させることができます。たとえば、echo $TERM は $TERM が セットされている内容を表示します。それはシェルが $TERM を展開して echo に渡しているからです。

シェルはさまざまな特殊文字を、特別なデータを表すものとして扱います。その特殊文字はメタキャラクタと呼ばれます。もっとも一般的なものは * で、これはファイル名に含まれる、あらゆる文字を表します。これらの特殊なメタキャラクタはファイル名の展開に使われます。たとえば、echo * と入力すると ls と入力したのとほとんど同じ結果を得られます。これはシェルが * とマッチするすべてのファイルを 受け取って echo のコマンドラインに渡し、表示するからです。

これらの特殊文字をシェルに解釈させないようにするため、 特殊文字の前にバックスラッシュ文字 (\) を置くことができます。echo $TERM は、あなたの端末が何にセットされているかを表示します。echo \$TERM は $TERM と そのまま表示します。

4.9.1. シェルの変更

シェルを変更する一番簡単な方法は chsh コマンドを使うことです。chsh を実行すると 環境変数 EDITOR で示されたエディタが立ち上がります。環境変数をセットしていなかった時は vi が立ち上がります。「Shell:」の行を適宜変更してください。

chsh に -s オプションをつけると、エディタを起動せずにシェルを変更することが可能です。たとえば、シェルを bash に変えたいなら、次のようにしてください。

```
% chsh -s /usr/local/bin/bash
```

注記

使おうと思っているシェルは必ず `/etc/shells` 中に書かれているものでなければなりません。 シェルを Ports Collection からインストールしていたのであれば、すでにそれは行なわれていますが、 手動でインストールした場合は、それを忘れずに行ってください。

たとえば、`bash` を手動で `/usr/local/bin` にインストールした場合 以下のようにする必要があります。

```
# echo "/usr/local/bin/bash" >> /etc/
shells
```

そして `chsh` を実行してください。

4.10. テキストエディタ

さまざまな FreeBSD の設定は、テキストファイルを編集することで行われます。そのため、テキストエディタの扱いに慣れると良いでしょう。FreeBSD には、基本システムの一部として二、三提供されるものと、Ports Collection から利用できる、たくさんのテキストエディタが用意されています。

最も学習が簡単なエディタは、easy editor の略で ee と呼ばれるものです。ee を立ち上げるには、コマンドラインから `ee` *`filename`* と入力します。ここで *`filename`* は、編集しようとしているファイルの名前です。たとえば、`/etc/rc.conf` を編集するには `ee /etc/rc.conf` と入力します。一旦 `ee` の中に入れば、エディタの機能を操作するコマンドはすべてディスプレイの上部に 表示されています。キャレット `^` 文字は キーボードの Ctrl キーを意味しますので、`^e` はキーのコンビネーション Ctrl+e を押すという意味になります。ee を終了するには Esc キーを押し、そして leave editor を選びます。ファイルが更新されていたときは、エディタは変更をセーブするかどうかプロンプトを出します。

FreeBSD には、基本システムの一部として vi、一方 Emacs や vim といった他のエディタは Ports Collection の一部として、より強力なテキストエディタが用意されています (`editors/emacs` , `editors/vim`)。これらのエディタはやや学習が複雑ですが、より強力で高い機能性を提供します。しかし、あなたが多量のテキストを編集することを考えているなら、vim や Emacs といった強力なエディタを習得することは、より多くの時間を節約することでしょう。

4.11. デバイスとデバイスノード

デバイスとはシステム上のハードウェアに関するものに対してよく使われる用語で、 ディスクやプリンタ、グラフィックカードやキーボードが含まれます。 FreeBSD が起動するとき、FreeBSD が表示しているものの大部分は検出されたデバイスです。 `/var/run/dmesg.boot` を眺めれば起動メッセージを読み直すことができます。

例えば、`acd0` は最初の IDE CDROM ドライブで、`kbd0` はキーボードを表します。

UNIX® オペレーティングシステムにおけるデバイスのほとんどは、 デバイスノードと呼ばれる `/dev` ディレクトリにあるスペシャルファイルを通してアクセスしなければなりません。

4.11.1. デバイスノードを作成する

新しいデバイスをシステムにつけ足したり、 追加デバイスのサポートをコンパイルして加えたりするときは、デバイスノードを作成しなければなりません。

4.11.1.1. **DEVFS** (デバイスファイルシステム: Device File System)

デバイスファイルシステム `DEVFS` は、グローバルファイルシステム名前空間の中のカーネルデバイス名前空間へのアクセスを提供します。 デバイスノードを作成したり変更したりするのではなく、`DEVFS` がこの特別なファイルシステムを管理するのです。

詳しくは devfs(5) マニュアルページをご覧ください。

4.12. さらに詳しい情報を得るには...

4.12.1. オンラインマニュアル

FreeBSD についてのもっとも包括的な文書は、 マニュアルページの形式になっているものです。 FreeBSD システム上のほとんどすべてのプログラムには、 基本的な操作方法とさまざまな引数を説明しているリファレンスマニュアルが添付されています。 これらのマニュアルは `man` コマンドで見ることができます。`man` コマンドの使い方は簡単です。

```
% man #####
```

`#####` のところには、知りたいコマンドの名前を入れます。たとえば `ls` コマンドについて知りたい場合には、次のように入力します。

```
% man ls
```

オンラインマニュアルは、セクション番号で分類されています。

1. ユーザコマンド

2. システムコールとエラー番号

3. C のライブラリ関数

4. デバイスドライバ

5. ファイル形式

6. ゲームや娯楽

7. さまざまな情報

8. システムの管理と操作のためのコマンド

9. カーネル開発者のための情報

時折、 同じトピックがオンラインマニュアルの複数のセクションに記載されている場合があります。たとえば、chmod ユーザコマンドと chmod() システムコールの場合がそれに該当します。この場合、man コマンドにセクション番号を与えることで、どちらを参照したいかを指定することができます。

```
% man 1 chmod
```

上のようにすれば、ユーザコマンド chmod のマニュアルページが表示されます。オンラインマニュアルの特定セクションへの参照は、慣習的に書かれている文書で括弧の中に示されます。すなわち、chmod(1) は chmod ユーザコマンドを、chmod(2) はシステムコールの方を示しています。

コマンドの名前を知っていて、 単純にその使い方を知りたい場合はここまでの説明で十分でしょう。しかし、もしコマンドの名前を思い出せない場合にはどうしたら良いのでしょうか? man に -k スイッチをつければ、コマンド解説 (description) の文章から、指定したキーワードを検索することができます。

```
% man -k mail
```

このコマンドにより、「mail」 というキーワードをコマンド解説に含むコマンドの一覧が表示されます。実際には、これは apropos コマンドを使う場合と同等の機能です。

それでは、/usr/bin にあるさまざまなコマンドすべてを見ていて、 それらが実際にどう働くのかが、まったく見当もつかないときには どうしたら良いでしょう? そのときは単純に、

```
% cd /usr/bin
% man -f *
```

とするか、あるいは同じ働きをする

```
% cd /usr/bin
% whatis *
```

としてください。

4.12.2. GNU の Info ファイル

FreeBSD には Free Software Foundation (FSF) によるアプリケーションや ユーティリティがたくさん含まれています。 これらのプログラムには、マニュアルページに加えて `info` ファイルと呼ばれる ハイパーテキスト形式の文書が付属しています。この文書は `info` コマンド、あるいは emacs をインストールしているなら emacs の info モードで読むことができます。

info(1) コマンドを使うには、単に次のように入力します。

```
% info
```

h と入力すると、簡単な手引きを読むことができます。クイックコマンドリファレンスは ? を入力してください。

第5章 アプリケーションのインストール - packages と ports

5.1. この章では

FreeBSD の基本システムには数多くのシステムツールが含まれています。 FreeBSD は、サードパーティ製のソフトウェアの導入を支援するために、 ソースコードをコンパイルしてインストールする Ports Collection と、コンパイル済みのバイナリをインストールする packages という相補的な 2 つの技術を提供しています。どちらのシステムを用いても、ローカルメディアやネットワーク上からソフトウェアをインストールできます。

この章を読むと、以下のことがわかります。

- packages と ports の違い
- FreeBSD に移植されたサードパーティ製のソフトウェアの探し方
- pkgng を用いてバイナリ package を管理する方法
- Ports Collection を用いてサードパーティ製のソフトウェアをソースコードから構築する方法
- インストール後の設定のために、 アプリケーションとともにインストールされたファイルを探す方法
- ソフトウェアのインストールに失敗した場合に、どうしたらよいか

5.2. ソフトウェアのインストール

UNIX® システムでは、サードパーティ製ソフトウェアの典型的なインストール手順は以下のようになります。

1. ソースコード、またはバイナリ形式で配布されているソフトウェアを探し出し、ダウンロードする。
2. 配布時のフォーマットからソフトウェアを取り出す。 一般的には compress(1), gzip(1) または bzip2(1) で圧縮された tarball です。
3. `INSTALL` または `README` ファイル、あるいは `doc/` サブディレクトのファイルからドキュメントを探しだし、ソフトウェアのインストール方法を調べる。

4. ソース形式でソフトウェアが配布されている場合はコンパイルを行う。 ここでは、`Makefile` の編集、または、`configure` スクリプトの実行を伴うことがあります。

5. ソフトウェアの動作を確認し、インストールする。

インストールしているソフトウェアパッケージが、FreeBSD を意識して移植されたものでなかったり、FreeBSD 上でテストされていなければ、正しくインストールが行われ、適切に動くようにコードを調べ、 編集する必要があるかもしれません。この文書を書いている時点では、24,000 を越えるサードパーティ製アプリケーションが利用可能です。

FreeBSD の package は、コンパイル済みのアプリケーションの全コマンド、各種設定ファイルやドキュメントを含んでいます。pkgng コマンドでは、`pkg install` といったコマンドで、package を扱うことができます。

FreeBSD port は、アプリケーションをソースコードからコンパイルする際の処理を自動化するように設計されたファイルの集まりです。port を構成するファイルは、自動的にアプリケーションをダウンロードし、展開、パッチ作業、コンパイル、そしてインストールを行うために必要な情報を含んでいます。

さらに ports システムは、FreeBSD の package 管理コマンドで扱うことのできる packages を生成できます。

packages と ports は依存関係を理解します。package または port を用いてアプリケーションをインストールすると、 依存するライブラリがまだインストールされていない場合には、最初にライブラリが自動的にインストールされます。

2 つの技術は類似していますが、packages と ports にはそれぞれ独自の特徴があります。それぞれのアプリケーションのインストールに対する必要要件に応じてどちらかを選択してください。

- 一般的に、あるアプリケーションの package の tarball は、 ソースコードを含む tarball より小さなサイズとなります。

- packages はコンパイルの時間を必要としません。 このことは、遅いシステム上で Mozilla, KDE, または GNOME といった大きなアプリケーションを扱う場合に重要となります。

- packages を用いれば、ソフトウェアのコンパイルに関する知識は必要ありません。

- packages は、通常最も多くのシステムで実行できるように、 非常に保守的な設定で構築されています。port からコンパイルすることで、コンパイルオプションを指定できます。

- アプリケーションのなかには、 どの機能をインストールするかをコンパイル時に設定するものがあります。たとえば、Apache は多種多様な ビルトインオプションを設定できます。

設定を区別するために、同じアプリケーションに対して 複数の packages が存在することがあります。たとえば、Ghostscript は Xorg がインストールされているかどうかにより、`ghostscript` package と `ghostscript-nox11` package が選択可能となっています。アプリケーションのコンパイルオプションが 1 つもしくは 2 つ以上になると、複数の packages を用意することは困難になります。

- ライセンス条項で、バイナリでの配布を禁止しているソフトウェアがあります。このようなソフトウェアはソースコードで配布される必要があり、エンドユーザがコンパイルしなくてはなりません。
- バイナリ配布を信用していない人や、潜在的な問題点を見つけ出すためにソースコードを読むことを好む人がいます。
- カスタマイズしたパッチを適用するためには、ソースコードが必要になります。

ports の更新状況を把握するために、FreeBSD ports メーリングリスト や FreeBSD ports bugs メーリングリスト を購読するとよいでしょう。

警告

アプリケーションをインストールする前に、 そのアプリケーションに関連したセキュリティ上の問題がないことを `http://vuxml.freebsd.org/` で確認するか、ports-mgmt/portaudit をインストールしてください。インストールしたら、`portaudit -F -a` と入力して、インストールされているアプリケーションに既知の脆弱性がないことを確認してください。

この章では、packages と ports を用いた FreeBSD 上での サードパーティ製ソフトウェアのインストール方法や管理方法について説明します。

5.3. ソフトウェアの探し方

FreeBSD 上で利用可能なアプリケーションのリストは常に増えています。 インストールするソフトウェアを探す方法はたくさん用意されています。

- FreeBSD ウェブサイトは、 利用可能なすべてのアプリケーションの最新の一覧を、検索できる形で http://www.FreeBSD.org/ja/ports/ において公開しています。ports はアプリケーションの名前や、ソフトウェアのカテゴリで検索出来ます。
- Dan Langille は、包括的な検索ユーティリティや Ports Collection にあるアプリケーションの変更点を追跡する FreshPorts.org を公開しています。 登録したユーザは、

監視している ports がアップデートされた時に、そのことを自動的にメールで知らせてくれるような、カスタマイズ可能な監視リストを使うことができます。

- アプリケーションの名前がわからなければ、Freecode.com のようなサイトでアプリケーションを探して下さい。その後、そのアプリケーションが ports で利用可能かどうかを FreeBSD サイトで調べて下さい。

- Ports Collection がすでにインストールされていれば、ports ツリーのローカルバージョンを調べることができます。port がどのカテゴリに分類されているのかを知りたければ、whereis(1) コマンドで調べることができます。`whereis ####` と入力してください。`####` の部分にはインストールを考えているプログラム名を入れます。

```
# whereis lsof
lsof: /usr/ports/sysutils/lsof
```

さらに、以下の例のように echo(1) を使って調べることもできます。

```
# echo /usr/ports/*/*lsof*
/usr/ports/sysutils/lsof
```

この方法では `/usr/ports/distfiles` 以下にダウンロードされたファイル名にもマッチします。

- また、Ports Collection に備わっている検索機能を利用して port を検索する方法もあります。この検索機能を利用するには、cd コマンドを用いて `/usr/ports` ディレクトリに移動し、`make search name=######` と入力してください。`######` の部分には検索したいソフトウェアの名前を入れてください。たとえば、`lsof` を探すには次のようにします。

```
# cd /usr/ports
# make search name=lsof
Port:   lsof-4.88.d,8
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
Maint:  ler@lerctr.org
Index:  sysutils
B-deps:
R-deps:
```

ヒント

Ports Collection に用意されている検索のメカニズムでは、インデックスファイルを利用して検索を行います。もし `INDEX` が必要であるというメッセージが表示されたら、`make fetchindex` を実行して、最新のインデックスファイルをダ

> ウンロードしてください。 `INDEX` が用意されれば、 `make search` で検索を実行できるでしょう。

「Path:」という行は、port がどこにあるかを示しています。

より絞られた情報を得るには、`quicksearch` と呼ばれる機能を使ってください。

```
# cd /usr/ports
# make quicksearch name=lsof
Port:   lsof-4.88.d,8
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
```

もっと詳しく検索するには、`make search key=string` または `make quicksearch key=string` と入力してください。*`string`* の部分には検索したいテキストを入れます。プログラムの名前がわからない場合でも、ある目的に関連した ports の検索に利用できるよう、テキストの部分には、コメント、説明文および依存情報を入れることができます。

`search` および `quicksearch` を使う場合には、検索文字列中の大文字と小文字を区別せずに検索が行われるので、「LSOF」を検索した結果は、「lsof」と同じ検索結果になります。

5.4. pkgng によるバイナリ package の管理

pkgng は、FreeBSD における伝統的な package 管理ツールの置き換えとなる次世代の管理ツールで、バイナリ packages をより早く、より簡単に管理できるようにする数多くの機能を提供します。

pkgng は ports-mgmt/portmaster や ports-mgmt/portupgrade などの port 管理ツールの置き換えではありません。これらのツールは、サードパーティ製ソフトウェアをバイナリ packages と Ports Collection の両形式からインストールできますが、一方で pkgng はバイナリ packages のみをインストールします。

5.4.1. pkgng 入門

FreeBSD 8.4 以降では、pkgng とマニュアルページをダウンロードし、インストールするためのブートストラップユーティリティが用意されています。

システムをブートストラップするためには、以下を実行してください。

```
# /usr/sbin/pkg
```

これより前のバージョンの FreeBSD では、かわりに Ports Collection または packages を用いてインストールする必要があります。

port をインストールするには以下を実行してください。

```
# cd /usr/ports/ports-mgmt/pkg
# make
# make install clean
```

package からインストールするには以下を実行してください。

```
# pkg_add -r pkg
```

pkgng をインストールしたら、以下のコマンドを実行して、package データベースをこれまでの伝統的なフォーマットから新しいフォーマットへと変換する必要があります。

```
# pkg2ng
```

このステップは、 サードパーティ製ソフトウェアがまだインストールされていないような、新しくインストールされた直後のシステムでは必要ありません。

重要

このステップは非可逆です。 一度 package データベースを pkgng フォーマットへと変換したら、伝統的な pkg_* ツールを使うべきではありません。

注記

package データベースを変換する際には、新しいバージョンへのデータ変換に伴ったエラーが出力されることがあります。 通常、これらのエラーは無視して構いませんが、pkg2ng 終了後に表示される、 変換に失敗したサードパーティ製ソフトウェアの一覧については、これらのソフトウェアを手動で再インストールする必要があります。

FreeBSD のバージョンが 10.*X* より前であれば、以下の行を /etc/make.conf に追加して、FreeBSD Ports Collection がソフトウェアの登録に、伝統的な package のフォーマットではなく、pkgng を用いるように設定してください。

```
WITH_PKGNG= yes
```

pkgng package 管理システムでは、ほとんどのコマンドに対して package リポジトリを使います。デフォルトの package リポジトリは `/usr/local/etc/pkg.conf` または `PACKAGESITE` 環境変数で定義されます。この環境変数は、設定ファイルを上書きします。

その他の pkgng の設定オプションは、pkg.conf(5) に記述されています。

pkgng の利用情報は、pkg(8) マニュアルページや、`pkg` を引数なしに実行すると表示されます。

各 pkgng コマンドの引数は、コマンドに固有なマニュアルページに記述されています。たとえば、`pkg install` のマニュアルページを読むには、以下のコマンドのどちらかを実行してください。

```
# pkg help install
```

```
# man pkg-install
```

以下の節では、pkgng を用いた通常のバイナリ package の管理について説明します。各コマンドでは、カスタマイズのために、多くのオプションが使われています。詳細や、他の例については、コマンドのヘルプやマニュアルページを参照してください。

5.4.2. インストールされている package の情報を入手する

オプションを使用しないで `pkg info` を実行すると、システムにインストールされているすべての package もしくは、ある特定の package の情報が得られます。

たとえば、インストールされている pkgng の情報を調べるには、以下のように実行してください。

```
# pkg info pkg
pkg-1.1.4_1
```

5.4.3. package のインストールと削除

バイナリ package をインストールするには、 以下のコマンドを使ってください。 ここで *`packagename`* は、インストールする package の名前です。

```
# pkg install packagename
```

このコマンドは、リポジトリデータを使用して、 インストールすべきソフトウェアのバージョン、および、インストールされていない依存ソフトウェアがあるかどうかを調べます。たとえば、curl をインストールするには以下を実行してください。

```
# pkg install curl
```

```
Updating repository catalogue
/usr/local/tmp/All/curl-7.31.0_1.txz          100% of 1181 kB 1380 ↺
kBps 00m01s

/usr/local/tmp/All/ca_root_nss-3.15.1_1.txz   100% of  288 kB 1700 ↺
kBps 00m00s

Updating repository catalogue
The following 2 packages will be installed:

        Installing ca_root_nss: 3.15.1_1
        Installing curl: 7.31.0_1

The installation will require 3 MB more space

0 B to be downloaded

Proceed with installing packages [y/N]: y
Checking integrity... done
[1/2] Installing ca_root_nss-3.15.5_1... done
[2/2] Installing curl-7.31.0_1... done
Cleaning up cache files...Done
```

新しい package と依存関係から追加された package は、インストール済み package 一覧に表示されます。

```
# pkg info
ca_root_nss-3.15.5_1 The root certificate bundle from the Mozilla ↺
Project
curl-7.31.0_1 Non-interactive tool to get files from FTP, GOPHER, ↺
HTTP(S) servers
pkg-1.1.4_6 New generation package manager
```

必要のなくなった packages は、`pkg delete` を使って削除できます。たとえば、以下のようにして削除できます。

```
# pkg delete curl
The following packages will be deleted:

 curl-7.31.0_1

The deletion will free 3 MB

Proceed with deleting packages [y/N]: y
[1/1] Deleting curl-7.31.0_1... done
```

5.4.4. インストールされている package のアップグレード

`pkg version` を用いて古くなった packages を見つけることができます。ローカルに ports ツリーがない場合には、pkg-version(8) は、リモートリポジトリのカタログを利用

します。そうでなければ、ローカルの ports ツリーを使って package のバージョンを同定します。

pkg upgrade を実行すると、インストールされている packages が最新のバージョンにアップグレードされます。このコマンドは、インストールされているソフトウェアのバージョンと、リポジトリのカタログから利用できるバージョンとを比較します。比較が終わったら、新しいバージョンが用意されているアプリケーションの一覧を表示します。アップグレードを行うのであれば、**y** を選択してください。キャンセルする場合には、**n** を選択してください。

5.4.5. インストールされている package の検証

時折、サードウェア製のアプリケーションに脆弱性が見つかることがあります。脆弱性を調べるために、pkgng は、検証機能を持っています。システムにインストールされているソフトウェアに既知の脆弱性がないかどうかを調べるには、以下のように実行してください。

```
# pkg audit -F
```

5.4.6. リーフ依存 ports の自動削除

package を削除すると、不必要な依存 ports が残されることがあります。依存のために導入され、現在は不必要になった package は、以下のようにすると自動的に検出され、削除されます。

```
# pkg autoremove
Packages to be autoremoved:
 ca_root_nss-3.13.5

The autoremoval will free 723 kB

Proceed with autoremoval of packages [y/N]: y
Deinstalling ca_root_nss-3.15.1_1... done
```

5.4.7. package データベースのバックアップ

伝統的な package 管理システムとは異なり、pkgng には package データベースをバックアップするメカニズムがあります。package データベースの内容を手動でバックアップするには、以下を実行してください。以下の *pkgng.db* を適切なファイル名に置き換えてください。

```
# pkg backup -d pkgng.db
```

さらに、pkgng は package データベースを毎日自動的にバックアップする periodic(8) スクリプトを含んでいます。この機能は、periodic.conf(5) の中で、`daily_backup_pkgng_enable` を YES に設定すると有効になります。

ヒント

スクリプトによる定期的な package データベースのバックアップを無効にするには、 periodic.conf(5) の中で、 `daily_backup_pkgdb_enable` を `NO` に設定してください。

過去にバックアップした package データベースの中身をリストアするには、以下を実行してください。

```
# pkg backup -r /path/to/pkgng.db
```

5.4.8. 古くなった package の削除

デフォルトでは、pkgng はキャッシュディレクトリにバイナリ packages を保存します。このディレクトリは、pkg.conf(5) の `PKG_CACHEDIR` 変数で定義されます。 `pkg upgrade` を使って packages をアップグレードする際には、 アップグレードされた package の古いバージョンは自動的には削除されません。

システムからこれらの古いバイナリ package を削除するには、以下を実行してください。

```
# pkg clean
```

5.4.9. package メタデータの変更

FreeBSD Ports Collection では、メジャーバージョン番号が変更になることがあります。これに対応するために、pkgng には、package の情報をアップデートするコマンドが組み込まれています。たとえば、`lang/php5` が、バージョン `5.4` を表すようになり、`lang/php5` を `lang/php53` と名前を変更する必要があるような場合に、有用です。

上記の例の package の情報を変更するには、以下のように実行してください。

```
# pkg set -o lang/php5:lang/php53
```

別の例として、lang/ruby18 を lang/ruby19 にアップデートするには、以下のようにしてください。

```
# pkg set -o lang/ruby18:lang/ruby19
```

最後の例として、`libglut` 共有ライブラリの情報を graphics/libglut から graphics/freeglut へと変更するには、以下のように実行してください。

```
# pkg set -o graphics/libglut:graphics/freeglut
```

注記

package の情報を変更したら、情報が変更された package に依存している packages を再インストールすることが重要となります。依存 packages を再インストールするには、以下のように実行してください。

```
# pkg install -Rf graphics/freeglut
```

5.5. Ports Collection の利用

Ports Collection とは、`/usr/ports` 以下に置かれる `Makefile`, 修正パッチ、説明文などの一連のファイルのことです。このファイルのセットは、アプリケーションをコンパイルして FreeBSD にインストールするのに用いられます。port を用いてアプリケーションをコンパイルできるようにするには、まず最初に Ports Collection をインストールする必要があります。FreeBSD のインストール時に Ports Collection をインストールしなかった場合に、以下の方法のどれかを用いてインストールしてください。

手順5.1 Portsnap を利用する方法

FreeBSD のベースシステムには、Portsnap が含まれています。 これはは Ports Collection を取得するための速くて使いやすく、 多くのユーザに推奨されるツールです。 このユーティリティは、FreeBSD のサイトに接続し、セキュリティキーを検証し、Ports Collection の最新版をダウンロードします。セキュリティキーは、ダウンロードしたすべてのファイルの検証に用いられます。

1. 圧縮された Ports Collection のスナップショットを `/var/db/portsnap` にダウンロードするには以下を実行してください。

   ```
   # portsnap fetch
   ```

2. 初めて Portsnap を使う時は、スナップショットをまず `/usr/ports` に展開してください。

   ```
   # portsnap extract
   ```

3. 上で示した Portsnap を初めて利用する際に行うコマンドを実行した後は、以下のコマンドで `/usr/ports` をアップデートしてください。

   ```
   # portsnap fetch
   # portsnap update
   ```

`fetch` を使う場合には、以下のよう `extract` または `update` を連続して行うことができます。

```
# portsnap fetch update
```

手順5.2 Subversion を用いる方法

ports ツリーの管理が必要な場合や、ローカルで変更点をメンテナンスする必要がある場合には、 Subversion を使って Ports Collection を取得する方法があります。Subversion のより詳細な説明については、 Subversion Primer を参照してください。

1. Subversion を使って ports ツリーをチェックアウトする前に、Subversion をインストールしておく必要があります。ports ツリーがすでにインストールされていれば、以下のようにして Subversion をインストールできます。

   ```
   # cd /usr/ports/devel/subversion
   # make install clean
   ```

   ports ツリーが利用できない場合や、package の管理に pkgng を使っているのであれば、package から Subversion をインストールできます。

   ```
   # pkg install subversion
   ```

2. ports ツリーをチェックアウトしてください。 パフォーマンスを良くするため、*`svn0.us-east.FreeBSD.org`* を Subversion ミラー の中から地理的に近い場所にあるミラーに置き換えてください。

   ```
   # svn checkout https:// svn0.us-east.FreeBSD.org /
   ports/head /usr/ports
   ```

3. Subversion で最初のチェックアウトを行ったら、必要に応じて `/usr/ports` をアップデートしてください。

   ```
   # svn update /usr/ports
   ```

Ports Collection はソフトウェアのカテゴリを表すディレクトリを持ち、各カテゴリには、各アプリケーションのサブディレクトリがあります。ports スケルトンとも呼ばれる各サブディレクトリには、プログラムを FreeBSD 上で正しくコンパイルしてインストールする方法を提供するファイルのセットが含まれています。それぞれの port スケルトンには、次のファイルおよびディレクトリが含まれています。

- `Makefile`: このファイルにはアプリケーションのコンパイル方法やシステムのどこにインストールするかを指定する命令文が含まれています。
- `distinfo`: このファイルには、その port を構築するためにダウンロードする必要があるファイルのファイル名と、チェックサム情報が含まれています。

- `files`: このディレクトリには FreeBSD 上でプログラムをコンパイルし、インストールするための修正パッチが含まれています。このディレクトリには、その port の構築に必要なその他のファイルが入る場合もあります。
- `pkg-descr`: このファイルにはプログラムに関する、より詳しい説明文が含まれます。
- `pkg-plist`: これは、その port によってインストールされる全ファイルのリストです。これにはプログラムを削除する際に、どのファイルを削除すれば良いのかを ports システムに伝える役割もあります。

これらの他に `pkg-message` や特殊な状況に対応するためのファイルを含む ports もあります。これらのファイルについての詳細および ports の一般的な説明については、port 作成者のためのハンドブック をご覧下さい。

port は実際のソースコード (`distfile` とも呼ばれます) を含んではいません。port の構築の展開部で、ダウンロードされたソースは自動的に `/usr/ports/distfiles` に保存されます。

5.5.1. ports のインストール

この節では、Ports Collection を利用してプログラムをインストールしたり、システムから削除したりする基本的な手順について説明します。利用可能な `make` のターゲットや環境変数についての詳細は ports(7) をご覧ください。

警告

いかなる port でも、構築する前には、前節に書かれているように、Ports Collection をアップデートしてください。サードパーティ製のソフトウェアをインストールすると、セキュリティの脆弱性を引き起こす可能性があります。その port に関連したセキュリティ上の問題がないことを、まずは `http://vuxml.freebsd.org/` で確認してください。または、`ports-mgmt/portaudit` をインストールしているのであれば、新しい port をインストールする前に、`portaudit -F -a` を実行してください。毎日のシステムのセキュリティ確認時に、自動的にセキュリティの検査およびデータベースの更新を行うようにこのコマンドを設定できます。詳しくは、portaudit および periodic(8) のマニュアルページを参照してください。

Ports Collection は、ネットワークに接続できることを想定しています。また、superuser の権限も必要となります。

freebsdmaill.com の FreeBSD Toolkit のようなサードパーティ製の DVD-ROM 製品の中には distfiles を収録しているものがあります。これらを使うとインターネットへの接続がなくても ports をインストールするのに使うことができます。DVD を `/cdrom` にマウントしてください。ほかのマウントポイントを使用したければ、`CD_MOUNTPTS` 変数を設定してください。ディスク上に必要な distfiles が存在すると、自動的に利用されます。port には DVD への収録を許可しないライセンス条項を持つものがあります。これにはダウンロード前に登録を必要としたり、再配布が禁止されているという理由があります。DVD に含まれていない port をインストールしたい場合には、ネットワークに接続する必要があります。

port をコンパイルしてインストールするには、インストールしたい port のディレクトリに移動してください。その後、プロンプトから `make install` と入力してください。すると、次のような出力が現われるはずです。

```
# cd /usr/ports/sysutils/lsof
# make install
>> lsof_4.88D.freebsd.tar.gz doesn't seem to exist in /usr/ports/
distfiles/.
>> Attempting to fetch from ftp://lsof.itap.purdue.edu/pub/tools/
unix/lsof/.
===>  Extracting for lsof-4.88
...
[extraction output snipped]
...
>> Checksum OK for lsof_4.88D.freebsd.tar.gz.
===>  Patching for lsof-4.88.d,8
===>  Applying FreeBSD patches for lsof-4.88.d,8
===>  Configuring for lsof-4.88.d,8
...
[configure output snipped]
...
===>  Building for lsof-4.88.d,8
...
[compilation output snipped]
...
===>  Installing for lsof-4.88.d,8

===>  Installing for lsof-4.88.d,8
...
[installation output snipped]
...
===>   Generating temporary packing list
===>   Compressing manual pages for lsof-4.88.d,8
===>   Registering installation for lsof-4.88.d,8
===>  SECURITY NOTE:
      This port has installed the following binaries which execute ↺
with
      increased privileges.
/usr/local/sbin/lsof
#
```

lsof は高い権限で動作するプログラムなので、インストールする時にセキュリティに関する警告が表示されます。インストールが終わったら、プロンプトが戻ります。

シェルによってはコマンドの実行ファイルを探す時間を短縮するために、環境変数 PATH に登録されている ディレクトリのコマンド一覧をキャッシュするものがあります。 tcsh シェルを使っているのであれば、フルパスを指定することなく新しくインストールしたコマンドを利用できるように、rehash を実行してください。sh シェルを使っているのであれば かわりに hash -r を実行してください。詳細については、あなたの使っているシェルのドキュメントをご覧ください。

インストールの間に、作業用ディレクトリが作成されます。このディレクトリにはコンパイル時に使用されるすべての一時ファイルが含まれています。 このディレクトリを削除することで、ディスク容量を節約でき、また port を新しいバージョンへアップデートする際に問題が起こる可能性を小さくします。

```
# make clean
===>  Cleaning for lsof-88.d,8
#
```

注記

port を構築する際に、make install clean と実行することで、これらの余分な手順を省くことができます。

5.5.1.1. ports のインストールのカスタマイズ

ports の中にはビルドオプションを指定できるものがあります。このオプションを指定することで、 アプリケーションの機能の一部を有効もしくは無効にできます。また、セキュリティオプションを設定したり、 その他のカスタマイズを行うことができます。 このようなアプリケーションには www/firefox , security/gpgme や mail/sylpheed-claws などがあります。port が他のカスタマイズ可能なオプションを持つ ports に依存する場合には、デフォルトでは、ユーザに port のオプションをメニューから選択させる設定のため、何度もユーザとの対話が起こり待たされることがあります。これを避けるには、まず最初に port スケルトンで make config-recursive を実行して設定を一括で行い、その後 make install [clean] を実行して port を構築してインストールしてください。

ヒント

config-recursive を実行する際、all-depends-list を実行すると、設定すべき ports の一覧を得ることができます。多

> くの場合は、すべての依存 ports のオプションが定義され、ports オプションの画面が表示されなくなり、すべてのオプションが設定されたことを確認できるまで `make config-recursive` を実行すると良いでしょう。

port の構築後、再びこのメニューを表示させてオプションの追加や削除、設定の変更を行う方法はたくさんあります。一つ目は port のディレクトリに `cd` で移動し、`make config` と入力する方法です。別の方法は `make showconfig` を使う方法です。他の方法は `make rmconfig` の実行です。このコマンドを実行すると選択されているすべてのオプションが削除され、設定をもう一度やり直すことができます。これらの方法や他の方法についての詳細は、ports(7) マニュアルで説明されています。

ports は、いくつかの環境変数を参照する fetch(1) を用いてソースファイルをダウンロードします。FreeBSD システムがファイアウォールの内側であったり、FTP/HTTP プロキシを使う場合には、`FTP_PASSIVE_MODE`, `FTP_PROXY`, `FTP_PASSWORD` の環境変数を設定することなります。対応している環境変数の一覧については fetch(3) をご覧ください。

インターネットに常時接続できないユーザのために `make fetch` コマンドが用意されています。このコマンドを `/usr/ports` で実行してすべての distfiles をダウンロードするか、`/usr/ports/net` といったカテゴリや、あるスケルトンにおいても実行できます。ある port がライブラリやその他の ports に依存している場合には、別のカテゴリの ports の distfiles はダウンロードされないことに注意してください。port が依存しているすべての distfiles をダウンロードしたければ、`make fetch-recursive` を使ってください。

めったにないことかもしれませんが、ローカルに distfiles のリポジトリがあるような場合に、`MASTER_SITES` 変数を変更することで `Makefile` で指定されているダウンロードの場所を 変更することができます。 設定する場合には、変更先を以下のようにして指定してください。

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE= \
ftp://ftp.organization.org/pub/FreeBSD/ports/
distfiles/  fetch
```

`WRKDIRPREFIX` 変数と `PREFIX` 変数を変更することで、作業ディレクトリやターゲットディレクトリをデフォルトのものから変更できます。

```
# make WRKDIRPREFIX=/usr/home/example/ports install
```

とすると、ports は `/usr/home/example/ports` でコンパイルされ、すべて `/usr/local` 以下にインストールされます。

```
# make PREFIX=/usr/home/example/local install
```

この場合、port のコンパイルは `/usr/ports` でおこない、`/usr/home/example/local` にインストールします。そして

```
# make WRKDIRPREFIX=../ports PREFIX=../local install
```

とすれば両者を組み合わせることが可能です。

これらを環境変数に設定する方法もあります。 どのように環境変数を設定するかについては、あなたの使っているシェルのマニュアルページを参照してください。

5.5.2. インストールした ports の削除

インストールされた ports は、`pkg delete` コマンドで削除できます。このコマンドの使用例は、「pkgng によるバイナリ package の管理」にあります。

あるいは、port のディレクトリにて `make deinstall` を実行することでも削除できます。

```
# cd /usr/ports/sysutils/lsof
make deinstall
===>  Deinstalling for sysutils/lsof
===>   Deinstalling
Deinstallation has been requested for the following 1 packages:

 lsof-4.88.d,8

The deinstallation will free 229 kB
[1/1] Deleting lsof-4.88.d,8... done
```

port が削除されるときに表示されるメッセージを読むことをお勧めします。もし削除した port に依存するアプリケーションがあった場合には、 その情報が表示されますが、port の削除は行われます。 そのようなケースでは、依存を直すためにアプリケーションを再インストールするとよいでしょう。

5.5.3. ports のアップグレード

ports のインストール後、時間が経過すると、Ports Collection で新しいバージョンのソフトウェアを利用できるようになります。この章では、どのようにしてアップグレードする必要のあるソフトウェアを判断するか、そしてアップグレードの方法について説明します。

インストールされている ports の新しいバージョンを利用できるかどうかを知るには、まず、 最新の ports ツリーがインストールされていることを確認してください。 これには、手順の 5.1 もしくは 5.2 で書かれているアップデートのコマンドを使ってください。その後、以下のコマンドを実行すると、現在利用可能なバージョンよりも古い ports の一覧が表示されます。

```
# pkg_version -l "<"
```

重要

アップグレードする前に /usr/ports/UPDATING を、ファイルの頭から、ports を最後にアップデートした日、 もしくはシステムをインストールをした日に最も近い日まで目を通してください このファイルには port をアップグレードする際にユーザが遭遇するであろう問題や、 追加で必要な作業などが記述されています。 例えば、ファイル形式の変更や設定ファイルの場所の変更、 前のバージョンと互換性がなくなったことなどが書かれています。 アップグレードする必要のある ports に関連した手順に注意し、 アップグレードする際にはこれらの手順に従ってください。

実際にアップグレードを行うには、 Portmaster もしくは Portupgrade を使ってください。

5.5.3.1. portmaster を用いた ports のアップグレード

ports-mgmt/portmaster package または port は、FreeBSD と共にインストールされているツールだけを使うので、インストールされている ports のアップグレードに推奨されているツールです。portmaster は、他の ports に依存せずに、「base」システムのツールのみを使うように設計されています。このアプリケーションは、どの ports をアップグレードすべきかの判断を、/var/db/pkg/ の情報を使って行います。port からこのユーティリティをインストールするには以下のようにしてください。

```
# cd /usr/ports/ports-mgmt/portmaster
# make install clean
```

Portmaster は、ports を 4 つのカテゴリに分類します。

- Root ports: 他の port に依存せず、他の port からも依存されない ports。
- Trunk ports: 他の port には依存しないが、他の port から依存されている ports。
- Branch ports: 他の port に依存し、他の port からも依存されている ports。
- Leaf ports: 他の port に依存するが、他の port からは依存されない ports。

これらのカテゴリの一覧や、アップデート可能な port の一覧を表示するには以下のようにしてください。

```
# portmaster -L
===>>> Root ports (No dependencies, not depended on)
```

```
===>>> ispell-3.2.06_18
===>>> screen-4.0.3
        ===>>> New version available: screen-4.0.3_1
===>>> tcpflow-0.21_1
===>>> 7 root ports
...
===>>> Branch ports (Have dependencies, are depended on)
===>>> apache22-2.2.3
        ===>>> New version available: apache22-2.2.8
...
===>>> Leaf ports (Have dependencies, not depended on)
===>>> automake-1.9.6_2
===>>> bash-3.1.17
        ===>>> New version available: bash-3.2.33
...
===>>> 32 leaf ports

===>>> 137 total installed ports
        ===>>> 83 have new versions available
```

以下のコマンドを使うと、古くなった ports をすべてアップデートします。

```
# portmaster -a
```

注記

Portmaster のデフォルトの設定では、 インストールされている port を削除する前にバックアップ用の package が作成されます。このバックアップは、新しいバージョンのインストールに成功すると削除されます。`-b` を使うと、Portmaster の自動的なバックアップの削除は行いません。`-i` を追加すると、Portmaster をインタラクティブモードで使用できます。このモードでは、各 port をアップグレードするかどうかの選択を対話的に行うことがでます。 多くのオプションが利用可能です。portmaster(8) のマニュアルページから、それらの使用方法に関する詳細な説明を読んでください。

アップグレードの過程でエラーに遭遇した場合には、`-f` を使ってすべての ports のアップグレードや再構築を行なってください。

```
# portmaster -af
```

Portmaster を使ってシステムに新しい ports をインストールしたり、新しい port のコンパイルやインストール前に依存するすべての port をアップグレードできます。この機能を使う時には、Ports Collection の場所を指定してください。

```
# portmaster shells/bash
```

5.5.3.2. Portupgrade を用いた ports のアップグレード

Portupgrade は、インストールした ports のアップグレードを行なうためのもう一つのユーティリティです。ports-mgmt/portupgrade package または port から利用できます。このユーティリティは ports を管理するために用いられるアプリケーションをインストールします。Ruby に依存します。port をインストールするには、以下を実行してください。

```
# cd /usr/ports/ports-mgmt/portupgrade
# make install clean
```

このユーティリティを使ってアップグレードを行う前に、`pkgdb -F` を使って、インストールされている ports の一覧を調べてください。矛盾が検出された場合には修復してください。

システムにインストールされている port の中で古くなったものをすべてアップデートするには `portupgrade -a` を実行してください。もし、すべての ports に対して個別にアップグレードするかどうかを確認したいのであれば、`-i` を追加してください。

```
# portupgrade -ai
```

ports で利用可能なすべてのアプリケーションではなく、ある特定のアプリケーションだけを更新したいのであれば、`portupgrade` *`pkgname`* を実行してください。アップグレードするアプリケーションが依存しているすべての ports をまず先に更新したい場合には、`-R` を使ってください。

```
# portupgrade -R firefox
```

`-P` オプションを使うと、portupgrade は `PKG_PATH` に登録されているローカルディレクトリから、利用可能な package を探します。ローカルに利用可能な packages が見つからなければ、リモートサイトから package のダウンロードを試みます。packages をローカルに見つけることができず、リモートサイトからもダウンロードできない場合には、portupgrade は ports からインストールを行ないます。ports を使用したくなければ、`-PP` オプションを指定してください。この最後のオプションを設定すると、もし package が利用できなければ Portupgrade は終了します。

```
# portupgrade -PP gnome2
```

また、ビルドやインストールを行なわず、distfiles または packages だけをダウンロードしたければ、`-F` オプションを指定してください。利用可能なすべてのオプションについては、portupgrade(1) のマニュアルを参照してください。

5.5.4. ports とディスク容量

Ports Collection を使い続けていると、そのうちディスクを食いつぶしてしまうでしょう。ports をビルドしてインストールした後、ports スケルトンで `make clean` を実行

すると、作業用の `work` ディレクトリを削除します。 Portmaster を使って port をインストールする場合には、`-K` を使わなければこのディレクトリは自動的に削除されます。 Portupgrade がインストールされている場合には、以下のコマンドはローカルの Ports Collection に見つかったすべての `work` ディレクトリを削除します。

```
# portsclean -C
```

さらに、時間が経つにつれ `/usr/ports/distfiles` には、古くなったソースファイルがたまっていきます。Portupgrade がインストールされている場合には、次のコマンドで、どの ports からも使われていない distfiles を削除できます。

```
# portsclean -D
```

Portupgrade を使って、システムにインストールされている port から使われていない distfiles をすべて削除するには、以下のコマンドを使ってください。

```
# portsclean -DD
```

もし Portmaster がインストールされているのであれば、以下を実行してください。

```
# portmaster --clean-distfiles
```

デフォルトでは、このコマンドはインタラクティブに設定されているため、 ユーザに対して distfile を削除すべきかどうかを確認するプロンプトが表示されます。

これらのコマンドに加え、`ports-mgmt/pkg_cutleaves` package または port は、必要なくなった ports を削除する作業を自動化します。

5.6. インストール後の作業

バイナリ package もしくは port のどちらを用いてソフトウェアをインストールするかに関わらず、 サードパーティ製のアプリケーションの多くは、 インストール後にある程度の設定を必要とします。 以下のコマンドや場所の情報は、 アプリケーションとともに何がインストールされたかを知るための助けとなるでしょう。

- 多くのアプリケーションでは、デフォルトの設定ファイルが、少なくとも一つは `/usr/local/etc` にインストールされます。この設定ファイルを確認し、必要に応じてシステムの要求に合うように編集してください。
- ドキュメントが付属しているアプリケーションは、 ドキュメントを `/usr/local/share/doc` にインストールします。また、多くのアプリケーションは、マニュアルページもインストールします。 これらのドキュメントは、 アプリケーションを使い続ける前に見ておくべきものです。
- ある種のアプリケーションでは、サービスを実行するためには、アプリケーションの起動前に、`/etc/rc.conf` に追加する必要があります。これらのアプリケーションで

は、通常、スタートアップスクリプトが `/usr/local/etc/rc.d` にインストールされます。詳細は、サービスの起動 をご覧ください。

- csh(1) のユーザは、`rehash` を実行して、シェルの `PATH` のバイナリリストを再構築してください。
- `pkg info` を使って、アプリケーションと共にインストールされたファイル、マニュアルページ、およびバイナリを調べることができます。

5.7. うまく動作しない ports に遭遇した場合には

port をうまくコンパイルできなかったりインストールできない場合には、以下を試してください。

1. その port に対する修正案が提出されていないかどうかを 障害報告 (Problem Report) データベース で調べてください。もし提案されていれば、その提案されている修正によって問題を解決できるかもしれません。
2. port の保守担当者に対応してもらいましょう。 port スケルトンで `make maintainer` と入力するか、port の `Makefile` を読み、保守担当者の電子メールアドレスを調べてください。保守担当者にメールを送る際には、port の `Makefile` の `$FreeBSD:` 行、そしてエラーが出力されるまでの出力ログを忘れずに添付してください。

注記

特定の保守担当者が存在せず、かわりに メーリングリスト が保守している ports があります。そのような場合には、メールアドレスは `<freebsd-listname@FreeBSD.org >` のようになります。メールを送る際には、このことに気をつけてください。

特に `<ports@FreeBSD.org >` が保守している ports には、保守担当者がいません。そのかわり、そのメーリングリストを購読する人々からなるコミュニティが、 修正や対応をおこなっています。もっとボランティアが必要です!

メールに対して返信がなければ、 FreeBSD 障害報告の書き方 に書かれている手順にしたがって、send-pr(1) を使ってバグレポートを提出してください。

3. 自分で直しましょう! ports システムに関する詳細な情報は port 作成者のためのハンドブック にあります。このセクションを読むと、壊れてしまった port を直したり、自分で作った port を提出したりできるようになります!

4.「pkgng によるバイナリ package の管理」 に書かれている手順にしたがって、package をインストールしてください。

第6章 X Window System

X.Org X11 サーバ対応: Ken Tom 、Marc Fonvieille.

6.1. この章では

FreeBSD では、ユーザに強力なグラフィカルインタフェースを提供するために X11 を採用しています。X11 は、Xorg (他のソフトウェアパッケージについてはここでは述べません) に実装された、フリーで利用できる X Window System です。FreeBSD におけるデフォルトかつ公式の X11 サーバは X.Org Foundation が開発している Xorg です。FreeBSD に似たライセンスに従って配布されています。

X11 がサポートするビデオハードウェアについての情報は、Xorg のウェブサイトをご覧ください。

この章を読めば以下のことがわかります。

- X Window System のさまざまなコンポーネントと、それらが互いにどのように連携しているか。
- X11 のインストールおよび設定方法
- さまざまなウィンドウマネージャのインストール方法
- X11 での TrueType® フォントの使い方
- GUI ログイン (XDM) の設定方法

この章を読み始める前に以下のことを理解しておく必要があります。

- サードパーティ製ソフトウェアのインストール方法 (5章アプリケーションのインストール - packages と ports)

6.2. 用語の説明

X Window System のさまざまなコンポーネントについての詳細や、それらがどのようにやり取りするかについてすべて理解する必要はありませんが、 これらのコンポーネントについて基本的なことを知っていると、強力な武器になるでしょう。

X サーバ
: X は最初からネットワークを意識してデザインされており、「クライアント - サーバ」モデルを採用しています。このモデルでは、「X サーバ」はキーボードやモニタ、マウスが接続されたコンピュータ上で動きます。このサーバはディスプレイの表示を管

理したり、キーボード、マウスからの入力を処理したり、タブレットやビデオプロジェクタ等の他の装置からの入出力を処理します。これは、ある人々を混乱させることがあります。X での用語は彼らが想定するものとは正反対だからです。彼らは「X サーバ」は地下にある大きなパワフルなマシンであり、「X クライアント」が自分たちのデスク上にあると想像するのです。

X クライアント
: XTerm や Firefox などの各 X アプリケーションは、「クライアント」になります。クライアントは「この座標にウィンドウを描いてください」といったメッセージをサーバへ送り、サーバは「ユーザが OK ボタンを押しました」といったメッセージを送り返します。

  家庭や小さなオフィスのような環境では、X サーバと X クライアントは通常同じコンピュータ上で動いています。X サーバを非力なコンピュータで動かし、X アプリケーションをより高性能なマシンで動かすことも可能です。この場合、X のクライアントとサーバの通信はネットワーク越しに行なわれます。

ウィンドウマネージャ
: X はスクリーン上でウィンドウがどのように見えるべきか、マウスでそれらをどうやって動かすか、ウィンドウ間を移動するのにどういうキーストロークを使うべきか、各ウィンドウのタイトルバーはどのように見えるべきか、クローズボタンを持つべきかどうか、といったことは規定しません。そのかわりに、X ではそういったことを「ウィンドウマネージャ」と呼ばれるアプリケーションに任せます。ウィンドウマネージャはたくさん あります。これらのウィンドウマネージャの見た目や使い勝手はそれぞれ異なっています。バーチャルデスクトップをサポートしているものもありますし、デスクトップを操作するキーストロークをカスタマイズできたり、「スタート」ボタンやそれに類するものを持っているものもあります。テーマに対応しており、デスクトップの見た目や使い勝手を完全に変えられるものもあります。ウィンドウマネージャは Ports Collection の `x11-wm` カテゴリに用意されています。

  それぞれのウィンドウマネージャは異なる設定機構を備えています。手で設定ファイルを編集しなければならないものや、設定作業のほとんどを GUI ツールで行うことができるものもあります。

デスクトップ環境
: KDE や GNOME は、デスクトップ環境です。これらは、共通のデスクトップのタスクを実行するための完全なアプリケーションスイートを含んでいます。オフィススイート、ウェブブラウザやゲームを含んでいるものもあります。

フォーカスポリシ
: ウィンドウマネージャは、マウスのフォーカスポリシに責任を持ちます。このポリシは、どのウィンドウがアクティブにキーストロークを 受け付けるようにするための方法を提供し、そして、どのウィンドウがアクティブなのかを示します。

  よく知られているフォーカスポリシは「click-to-focus」と呼ばれるものです。このポリシは、あるウィンドウ内でマウスをクリックすればそのウィンドウがアクティブにな

る、というものです。「focus-follows-mouse」ポリシでは、マウスポインタの下にいるウィンドウがフォーカスされるというものです。 フォーカスを変えるには他のウィンドウにマウスポインタを動かすだけです。 マウスがルートウィンドウに移動した時には、このウィンドウがフォーカスされます。「sloppy-focus」モデルでは、マウスがルートウィンドウに移動した時には、直前に使われていたウィンドウがフォーカスされています。sloppy-focus では、ポインタが別のウィンドウに移った時のみフォーカスが変わり、現在のウィンドウから出ただけでは変わりません。「click-to-focus」ポリシでは、マウスクリックによりアクティブなウィンドウが選択されます。ウィンドウは前面に表示され、他のすべてのウィンドウの前にきます。ポインタが別のウィンドウ上に移動した時でも、すべてのキーストロークがこのウィンドウに届きます。

それぞれのウィンドウマネージャは、それぞれのフォーカスポリシに対応しています。すべてのものは click-to-focus をサポートしていますし、多くのものは他の方法もサポートしています。どのフォーカスモデルを利用可能かどうかについては、ウィンドウマネージャのドキュメントをご覧ください。

ウィジェット
: ウィジェットはクリック可能であったり、 他の方法で操作可能なすべてのユーザインタフェース用アイテムを指す用語です。 ボタンやチェックボックス、ラジオボタン、アイコン、リスト、などがそうです。 ウィジェットツールキットはグラフィカルアプリケーションを作成するために使われます。KDE で使われている Qt や GNOME プロジェクトで使われている GTK+ といった有名なウィジェットセットがあります。そのため、アプリケーションのルックアンドフィールは、 アプリケーションを作成するのに使われたウィジェットツールキットに依存し、異なります。

6.3. X11 のインストール

FreeBSD における X11 の実装は、Xorg です。Xorg は X.Org Foundation がリリースしている X Window System のオープンソースな実装です。Xorg は XFree86™ 4.4RC2 と X11R6.6 のコードを元にしています。現在 FreeBSD Ports Collection で利用可能な Xorg のバージョンは 7.7 です。

Ports Collection から Xorg をビルドしインストールするには、以下のように入力します。

```
# cd /usr/ports/x11/xorg
# make install clean
```

注記

Xorg をすべてビルドするには、少なくとも 4 GB の空き容量が必要です。

package から直接 X11 をインストールすることもできます。pkg_add(1) ツールで扱われるバイナリパッケージが X11 についても提供されています。pkg_add(1) のリモート取得機能を利用する時は、package のバージョンは外してください。pkg_add(1) は自動的にそのアプリケーションの最新版を取得します。

ですから、Xorg の package を取得してインストールするには、次のように入力します。

```
# pkg_add -r xorg
```

注記

上記の例では、サーバ、クライアント、フォントなどを含んだ X11 ディストリビューション全体をインストールします。分割された X11 のパッケージや ports も利用できます。

最小の構成で X11 をインストールをするには、 x11/xorg-minimal をインストールしてください。

この章の残りでは、X11 をどのように設定すればよいか、また生産性の高いデスクトップ環境をどのように設定するかについて解説します。

6.4. X11 の設定

##: Christopher Shumway.

6.4.1. はじめに

X11 は、ほとんどの機器を自動で設定します。古かったり、一般的ではないような機器を使う場合には、設定を始める前に、 以下のハードウェア情報を集めておくと助けになるでしょう。

- モニタ同期周波数
- ビデオカードのチップセット
- ビデオカードのメモリ

画面の解像度とリフレッシュレートは、 モニタの水平および垂直同期周波数により決定されます。ほとんどすべてのモニタは、これらの値を自動検出します。これらの値を自動検出しないモニタもありますが、こういった仕様は、付属のマニュアルや、 製造元のウェブサイトから取得できます。

ビデオカードのチップセットも自動検出され、 ビデオドライバの選択に使われます。 正常に認識されない時のために、 どのチップセットが使われているかを知っておくとよいでしょう。

ビデオカードのメモリは、グラフィックアダプタがどの解像度とどの色数で動くことができるかを決めます。

6.4.2. X11 の設定

Xorg は、HAL を用いてキーボードやマウスを自動認識します。x11/xorg の依存として、sysutils/hal および devel/dbus port がインストールされますが、以下のようなエントリを `/etc/rc.conf` ファイルに追加し、有効にする必要があります。

```
hald_enable="YES"
dbus_enable="YES"
```

これらのサービスは、Xorg の設定や起動が行われる前に (手動や再起動によって) 起動している必要があります。

Xorg は、設定をすることなく

```
% startx
```

と入力するだけで起動します。

ある種のハードウェアは、自動設定で動作しなかったり、期待したような設定が行われないかもしれません。そのような場合には、手動で設定する必要があります。

注記

GNOME, KDE および Xfce などのデスクトップ環境では、 解像度などのスクリーンパラメータを簡単に設定できるツールがあります。したがって、デフォルトの設定が満足するようなものではない場合でも、これらのデスクトップ環境をインストールすることを考えているのであれば、デスクトップ環境をインストールして、スクリーン設定ツールを利用してください。

X11 の設定は複数の段階を経て行います。 まずは初期設定ファイルを作りましょう。スーパーユーザになって次のようにしてください。

```
# Xorg -configure
```

これにより、`/root` ディレクトリに `xorg.conf.new` という X11 の設定ファイルの雛形が生成されます (su(1) か直接ログインのどちらを利用したかが、受け継がれる

$HOME 環境変数に影響します)。X11 プログラムはシステム上のグラフィックハードウェアを検出し、そのハードウェア用の適切なドライバを読み込む設定ファイルを作ります。

次のステップは、作成した設定ファイルで Xorg がそのグラフィックハードウェアで動くことを確認することです。以下を実行してください。

```
# Xorg -config xorg.conf.new -retro
```

黒とグレーのグリッドと X のマウスポインタが現われればその設定は成功です。テストを終了するには、コマンドを実行した仮想コンソールに Ctrl+Alt+F***n*** (1 番目の仮想コンソールへは F1) と入力して切り替え、Ctrl+C と入力してください。

注記

Ctrl+Alt+Backspace キーの組み合わせでも Xorg を終了できます。このキー操作を利用する場合には、X 端末エミュレータで以下のコマンドを入力してください。

```
% setxkbmap -option ↺
terminate:ctrl_alt_bksp
```

または、hald 用のキーボード設定ファイル x11-input.fdi を作成して、/usr/local/etc/hal/fdi/policy ディレクトリに保存してください。 このファイルには以下を含める必要があります。

```
<?xml version="1.0" encoding="iso-8859-1"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.↺
keyboard">
   <merge key="input.x11_options.XkbOptions" ↺
type="string">terminate:ctrl_alt_bksp</merge>
    </match>
  </device>
</deviceinfo>
```

hald がこのファイルを読み込むように、コンピュータを再起動してください。

xorg.conf.new の ServerLayout もしくは ServerFlags セクションに、以下の行を追加する必要もあるでしょう。

```
        Option "DontZap" "off"
```

もしマウスが動作しなければ、 先へ進む前にマウスの設定を行う必要があります。 FreeBSD インストールの章の 「マウスの設定」 を参照してください。 さらに、 最近の Xorg では、デバイスの自動認識のため、 `xorg.conf` の `InputDevice` セクションは無視されます。 古い設定の記述を利用するには、 以下の行をファイルの `ServerLayout` もしくは、`ServerFlags` セクションに追加してください。

```
Option "AutoAddDevices" "false"
```

これで、以前のバージョンのように、入力デバイスを (キーボードレイアウトの変更のように) 必要なオプションを用いて設定できるようになります。

注記

すでに説明したように、デフォルトで hald デーモンがキーボードを自動的に認識します。 キーボードレイアウトやモデルを正しく認識しない場合でも、GNOME, KDE もしくは Xfce のようなデスクトップ環境が、キーボードの設定ツールを提供している可能性があります。しかしながら、setxkbmap(1) ユーティリティや hald の設定を利用することで、 キーボードのプロパティを直接設定できます。

たとえば、フランス語のレイアウトの PC 102 キーボードを使いたい場合には、hald のキーボード設定ファイル `x11-input.fdi` を作成し、`/usr/local/etc/hal/fdi/policy` ディレクトリに保存してください。 このファイルは以下を含んでいる必要があります。

```
<?xml version="1.0" encoding="iso-8859-1"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.↺
keyboard">
   <merge key="input.x11_options.XkbModel" ↺
type="string">pc102</merge>
   <merge key="input.x11_options.XkbLayout" ↺
type="string">fr</merge>
    </match>
  </device>
</deviceinfo>
```

このファイルがすでに存在する場合には、 キーボードの設定に関する部分をただ単にコピーし、ファイルに追加してください。

hald がこのファイルを読み込むように、コンピュータを再起動してください。

X 端末やスクリプトから以下のコマンドラインを実行することでも、同様に設定できます。

```
% setxkbmap -model pc102 -layout fr
```

`/usr/local/share/X11/xkb/rules/base.lst` ファイルは、利用可能なキーボード、レイアウトおよびオプションの一覧です。

次に `xorg.conf.new` を好みに合うように調整します。 emacs(1) や ee(1) のようなテキストエディタでファイルを開いてください。 モニタが古い場合や、通常とは異なるモデルでは、 同期周波数の自動認識に対応していないので、 これらの値を `xorg.conf.new` の `"Monitor"` セクションに次のように書き加えます。

```
Section "Monitor"
       Identifier   "Monitor0"
       VendorName   "Monitor Vendor"
       ModelName    "Monitor Model"
       HorizSync    30-107
       VertRefresh  48-120
EndSection
```

ほとんどのモニタは同期周波数の自動認識に対応しているので、 これらの値を手動で入力する必要はありません。自動認識に対応していないモニタでは、ダメージの可能性を避けるため、メーカーが提供している値のみを入力してください。

X はモニタが対応していれば DPMS (Energy Star) 機能を使うことができます。 xset(1) プログラムでタイムアウトをコントロールしたり、 強制的にスタンバイ、サスペンドや電源オフにすることができます。 モニタの DPMS 機能を有効にしたい場合は、 `"Monitor"` セクションに次の行を加えてください。

```
Option       "DPMS"
```

`xorg.conf.new` はエディタで開いたままにしておき、デフォルトの解像度と色数を好みで選びましょう。`"Screen"` セクションに以下のように書きます。

```
Section "Screen"
        Identifier "Screen0"
        Device     "Card0"
        Monitor    "Monitor0"
```

```
        DefaultDepth 24
        SubSection "Display"
                Viewport  0 0
                Depth     24
                Modes     "1024x768"
        EndSubSection
EndSection
```

`DefaultDepth` というキーワードは 実行時のデフォルトの色数について記述するためのものです。Xorg(1) のコマンドラインスイッチ `-depth` が使用された場合はそちらが優先されます。`Modes` というキーワードは、与えられた色数におけるデフォルトの解像度を記述しておくためのものです。ターゲットのシステムのグラフィックハードウェアによって定義されている、VESA スタンダードモードのみがサポートされていることに注意してください。上の例ではデフォルトの色数はピクセルあたり 24 ビットであり、この色数での解像度は 1024 ピクセル× 768 ピクセルです。

最後に、設定ファイルを保存し、上の例にあるようにテストしてみてください。

注記

トラブルシューティングの過程で手助けするツールのひとつに X11 のログファイルがあります。これには、X11 サーバが検知したデバイスそれぞれについての情報があります。Xorg のログファイル名は `/var/log/Xorg.0.log` という形式です。実際のログファイル名は `Xorg.0.log` から `Xorg.8.log` のように変わります。

すべてうまくいったなら、設定ファイルを Xorg(1) が見つけることができる共通の場所に置きます。これは、通常は `/etc/X11/xorg.conf` や `/usr/local/etc/X11/xorg.conf` です。

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

これで X11 の設定は完了です。startx(1) ユーティリティで Xorg を起動できます。また、xdm(1) を使って X11 サーバを起動することもできます。

6.4.3. 高度な設定

6.4.3.1. Intel® `i810` グラフィックチップセットの設定

Intel® i810 統合チップセットを設定するには、X11 にカードを制御させるために AGP プログラミングインタフェースである `agpgart` が必要になります。詳しくは、agp(4) ドライバのマニュアルページをご覧ください。

このドライバを用いることで、 他のグラフィックカードと同様に設定を行うことができるようになります。カーネルに agp(4) ドライバが組み込まれていないシステムでは、このモジュールを kldload(8) を使って読み込もうとしても動作しないことに注意してください。このドライバは、 起動時にカーネル内に存在するようにカーネル内部に組み込むか、 `/boot/loader.conf` を使わなければなりません。

6.4.3.2. ワイドスクリーンフラットパネルの追加

この章では、設定に関する幾分高度な知識を必要とします。 これまでに述べた標準ツールを使って設定に失敗する場合は、ログファイルを参照してください。ログファイルには、設定のために有用な情報が十分含まれています。テキストエディタを使用する必要があるでしょう。

現在のワイドスクリーン (WSXGA, WSXGA+, WUXGA, WXGA, WXGA+ など) は、16:10 や 10:9 形式、または (問題を含む可能性のある) 他のアスペクト比に対応しています。以下は、16:10 アスペクト比のスクリーン解像度の例です。

- 2560x1600
- 1920x1200
- 1680x1050
- 1440x900
- 1280x800

これらの解像度のひとつを以下のように `"Screen" #####` の `Mode` に追加してください。

```
Section "Screen"
Identifier "Screen0"
Device     "Card0"
Monitor    "Monitor0"
DefaultDepth 24
SubSection "Display"
 Viewport  0 0
 Depth     24
 Modes     "1680x1050"
EndSubSection
EndSection
```

Xorg は、I2C/DDC を通してワイドスクリーンの解像度に関する情報を取得できるので、モニタの周波数や解像度の範囲を把握しています。

もし、これらの `ModeLines` についての情報がドライバに存在しないのであれば、Xorg にヒントを与えなければならないでしょう。`ModeLine` を手動で設定するのに十分な情報

を /var/log/Xorg.0.log から得ることができます。以下のような情報を探してください。

```
(II) MGA(0): Supported additional Video Mode:
(II) MGA(0): clock: 146.2 MHz   Image Size:  433 x 271 mm
(II) MGA(0): h_active: 1680  h_sync: 1784  h_sync_end 1960 ↺
h_blank_end 2240 h_border: 0
(II) MGA(0): v_active: 1050  v_sync: 1053  v_sync_end 1059 ↺
v_blanking: 1089 v_border: 0
(II) MGA(0): Ranges: V min: 48  V max: 85 Hz, H min: 30  H max: 94 ↺
kHz, PixClock max 170 MHz
```

これは EDID と呼ばれる情報です。この情報を用いて ModeLine を作成するには、正しい順番に数字を入力するだけです。

```
 ModeLine <name> <clock> <4 horiz. timings> <4 vert. timings>
```

この例では "Monitor" セクションの ModeLine は以下のようになります。

```
Section "Monitor"
Identifier      "Monitor1"
VendorName      "Bigname"
ModelName       "BestModel"
ModeLine        "1680x1050" 146.2 1680 1784 1960 2240 1050 1053 ↺
1059 1089
Option          "DPMS"
EndSection
```

以上の簡単な編集作業が終わったら、新しいワイドスクリーンモニタ上で X が動作するでしょう。

6.5. X11 でのフォントの使用

##: Murray Stokely.

6.5.1. Type1 フォント

X11 に付いてくるデフォルトのフォントは、通常のデスクトップパブリッシングアプリケーションにとっては理想的とは言えない程度のものです。文字を大きくするとジャギーになりプロフェッショナルとは言えないようなものになりますし、 小さなフォントは頭が悪そうに見えます。しかし、世の中には質の高い Type1 (PostScript®) フォントがいくつかあり、 X11 ではそれらを簡単に利用することができます。 例えば、URW フォントコレクション (x11-fonts/urwfonts) には高品質の Type1 フォント (Times Roman®, Helvetica®, Palatino® など) が含まれています。freefont コレクション (x11-fonts/freefonts) にはもっとたくさんのフォントが含まれていますが、それらは Gimp のようなグラフィックソフトウェアで使用するためのものであり、スクリーンフォントとしては十分ではありません。さらに、X11 は簡単に TrueType® フォントを使うように設定することも可

能です。詳しくは、X(7) のマニュアルページか TrueType® フォントの節 を参照してください。

上記の Type1 フォントコレクションを Ports Collection から入れる場合には次のコマンドを実行してください。

```
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

freefont や他のコレクションでも同じようにします。X サーバがこれらのフォントを検出できるようにするには X サーバ設定ファイル (`/etc/X11/xorg.conf`) の適切な場所に次のような行を加えます。

```
FontPath "/usr/local/lib/X11/fonts/URW/"
```

別の方法としては、X のセッション中に次のようなコマンドラインを実行します。

```
% xset fp+ /usr/local/lib/X11/fonts/URW
% xset fp rehash
```

これは動くのですが、X のセッションが終了すると消えてしまいます。 消えないようにするには X の起動時に読み込まれるファイル (通常の `startx` セッションの場合は `~/.xinitrc` , XDM のようなグラフィカルなログインマネージャを通してログインする時は `~/.xsession`) に加えておきます。 三番目の方法は新しい `/usr/local/etc/fonts/local.conf` ファイルを使うことです。これに関しては アンチエイリアス 節をご覧ください。

6.5.2. TrueType® フォント

Xorg には、 TrueType® フォントのレンダリング機能が組み込まれています。 この機能を実現するために 2 つの異なるモジュールがあります。 ここでは、freetype の方が他のフォントレンダリングバックエンドと整合性が高いので、このモジュールを使うことにします。 freetype モジュールを使うためには `/etc/X11/xorg.conf` ファイルの `"Module"` セクションに以下の行を追加するだけです。

```
Load  "freetype"
```

さて、まずは TrueType® フォント用のディレクトリ (例えば `/usr/local/lib/X11/fonts/TrueType`) を作り、そこに TrueType® フォントをすべて放り込みましょう。 Macintosh® の TrueType® フォントは、そのままでは使うことができませんので注意してください。X11 で使うには UNIX®/MS-DOS®/Windows® 用のフォーマットでなければなりません。ファイルを置いたら ttmkfdir を使って `fonts.dir` ファイルを作り、X のフォントレンダラが新しいファイルがイントールされたことを分かるようにしてください。`ttmkfdir` は FreeBSD Ports Collection の x11-fonts/ttmkfdir からインストールできます。

```
# cd /usr/local/lib/X11/fonts/TrueType
# ttmkfdir -o fonts.dir
```

次に TrueType® フォントのディレクトリをフォントパスに追加します。上の Type1 フォントの場合と同じように、

```
% xset fp+ /usr/local/lib/X11/fonts/TrueType
% xset fp rehash
```

とするか、もしくは `xorg.conf` ファイルに `FontPath` 行を追加します。

これで終わりです。Gimp や Apache OpenOffice といったすべての X アプリケーションから TrueType® フォントを使うことができます。(高解像度なディスプレイで見るウェブページ上のテキストみたいな) とても小さなフォントや (StarOffice™ にあるような) 非常に大きなフォントもかなり綺麗に見えるようになることでしょう。

6.5.3. フォントのアンチエイリアス

: Joe Marcus Clarke.

`/usr/local/lib/X11/fonts/` と `~/.fonts/` にあるすべての X11 のフォントが、Xft に対応しているアプリケーションで自動的にアンチエイリアス表示できるようになりました。KDE, GNOME および Firefox のような最新のアプリケーションは、Xft に対応しています。

どのフォントがアンチエイリアスされるかを制御するため、もしくはアンチエイリアスの特性を設定するために、`/usr/local/etc/fonts/local.conf` ファイルを作成 (すでに存在しているのなら編集) します。多くの Xft フォントシステムの高度な機能をこのファイルを使って調整できます。この節ではいくつか簡単なところだけを紹介します。詳しくは、fonts-conf(5) をご覧ください。

このファイルは XML 形式でなければなりません。大文字小文字の区別に注意を払い、すべてのタグが正しく閉じられているか確認してください。 ファイルは一般的な XML ヘッダで始まり、DOCTYPE 定義と `<fontconfig>` タグがその後にきます。

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

すでに説明したように、`/usr/local/lib/X11/fonts/` と `~/.fonts/` にあるすべてのフォントは Xft 対応のアプリケーションで利用できます。これら 2 つ以外に別のディレクトリを追加したいのなら、`/usr/local/etc/fonts/local.conf` に以下のような行を追加します。

```
<dir>/path/to/my/fonts</dir>
```

新しいフォント、そして特に新しいフォントディレクトリを追加したら、以下のコマンドを実行してフォントキャッシュを再構築してください。

```
# fc-cache -f
```

アンチエイリアスをかけることによって境界が少しぼやけ、そのためにとても小さなテキストはさらに読みやすくなり、大きなフォントでは「ギザギザ」が消えるのです。しかし、普通のテキストにかけた場合には目が疲れてしまうこともあります。14 ポイント以下のサイズのフォントについて、アンチエイリアスをかけないようにするには次の行を加えます。

```
        <match target="font">
            <test name="size" compare="less">
                <double>14</double>
            </test>
            <edit name="antialias" mode="assign">
                <bool>false</bool>
            </edit>
        </match>
        <match target="font">
            <test name="pixelsize" compare="less" qual="any">
                <double>14</double>
            </test>
            <edit mode="assign" name="antialias">
                <bool>false</bool>
            </edit>
        </match>
```

いくつかの等幅フォントは、 アンチエイリアスをかけるとスペーシングがうまくいかなくなる場合があります。 特に KDE でその傾向があるようです。 解決策の一つとして、そういったフォントのスペーシングを 100 に設定する方法があります。そうするためには次の行を加えてください。

```
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>fixed</string>
            </test>
            <edit name="family" mode="assign">
                <string>mono</string>
            </edit>
        </match>
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>console</string>
            </test>
            <edit name="family" mode="assign">
                <string>mono</string>
            </edit>
        </match>
```

(これは固定サイズのフォントに "mono" という一般的な別名をつけます) そして以下を追加します。

```
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>mono</string>
            </test>
            <edit name="spacing" mode="assign">
                <int>100</int>
            </edit>
        </match>
```

Helvetica の様なある種のフォントは、アンチエイリアスすると問題が起こるでしょう。たいてい、フォントが縦に半分に切られて表示されます。 最悪の場合、アプリケーションがクラッシュします。これを回避するには、以下を `local.conf` に追加します。

```
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>Helvetica</string>
            </test>
            <edit name="family" mode="assign">
                <string>sans-serif</string>
            </edit>
        </match>
```

`local.conf` の編集を終えたら、ファイルの末尾が `</fontconfig>` タグで終わるようにしてください。これを行わなければ、変更は無視されるでしょう。

最後に、ユーザは自分だけの設定を各自の `.fonts.conf` ファイルに追加できます。これを行うためには、それぞれのユーザが単に `~/.fonts.conf` を作成するだけです。このファイルも XML 形式でなければなりません。

最後に一つ。LCD スクリーンではサブピクセルサンプリングが必要な場合があります。これは、基本的には (水平方向に分かれている) 赤、緑、青の各コンポーネントを別々に扱うことによって水平方向の解像度を良くするというもので、劇的な結果が得られます。これを有効にするには `local.conf` ファイルに次の行を加えます。

```
        <match target="font">
            <test qual="all" name="rgba">
                <const>unknown</const>
            </test>
            <edit name="rgba" mode="assign">
                <const>rgb</const>
            </edit>
        </match>
```

注記

ディスプレイの種類にもよりますが、`rgb` ではなく `bgr` や `vrgb`、もしくは `vbgr` の場合もあるので、試してみて最も良いものを使ってください。

6.6. X ディスプレイマネージャ

##: Seth Kingsley.

6.6.1. 概要

X ディスプレイマネージャ (XDM) は X Window System のオプショナルな一部分であり、ログインセッションの管理に用います。最低限の機能を実装した「X 端末」やデスクトップ、大規模なネットワークディスプレイサーバといった場面ではこれが有用です。X Window System はネットワークとプロトコルから独立しているため、ネットワークで繋がれた X のクライアントとサーバを動かすための設定はかなり幅が広くなります。XDM はどのディスプレイサーバに接続するかを選択でき、ログイン名とパスワードの組み合わせなど認証情報を入力できるグラフィカルなインタフェースを提供しています。

XDM がユーザに getty(8) (詳しくは「設定」をご覧ください) と同じ機能を提供することを考えてみてください。つまり、ディスプレイ上でシステムへのログインができ、ユーザの代わりにセッションマネージャ (通常は X のウィンドウマネージャ) を起動することができるのです。それから XDM は、ユーザが作業を終えてディスプレイからログアウトする合図を送ってきてプログラムが終了するのを待ちます。この時点で、XDM は次にログインするユーザのためにログイン画面や chooser 画面を表示できます。

6.6.2. XDM の使用

XDM を使用するには、x11/xdm port をインストールしてください (最近の Xorg のバージョンでは、デフォルトでインストールされません)。XDM のデーモンプログラムは `/usr/local/bin/xdm` にあります。このプログラムは `root` になればいつでも起動することができ、ローカルマシン上のディスプレイの管理を始めます。マシンをブートする際、いつも XDM を起動したい場合には、`/etc/ttys` にそのためのエントリを加えておくのが簡単です。このファイルのフォーマットや使用方法についての詳細は「`/etc/ttys` へのエントリの追加」を参照してください。デフォルトの `/etc/ttys` ファイルには仮想端末上で XDM デーモンを起動するための行:

```
ttyv8 "/usr/local/bin/xdm -nodaemon" xterm off secure
```

があります。このエントリはデフォルトでは無効になっており、有効にするには 5 番目のカラムを `off` から `on` にし、「`init` にファイル `/etc/ttys` の再読み込みをさせる」の指示に従って init(8) を再起動します。最初のカラムはこのプログラムが管理する端末の名前で、この場合 `ttyv8` になります。つまり、XDM は 9 番目の仮想端末で起動されるということです。

6.6.3. XDM の設定

XDM の設定用ディレクトリは `/usr/local/lib/X11/xdm` です。このディレクトリには XDM の振る舞いや見た目を変更するために用いられるいくつかのファイルがあります。だいたいは以下のような感じです。

ファイル	説明
`Xaccess`	クライアント認証のルールセット
`Xresources`	デフォルトの X リソース
`Xservers`	管理すべきリモートやローカルのディスプレイのリスト
`Xsession`	デフォルトのログイン時のセッションスクリプト
`Xsetup_*`	ログインインタフェースの前にアプリケーションを起動するためのスクリプト
`xdm-config`	このマシンで動いているすべてのディスプレイのグローバルな設定
`xdm-errors`	サーバプログラムによって生成されるエラー
`xdm-pid`	現在動いている XDM のプロセス ID

このディレクトリにはまた、XDM の動作中にデスクトップをセットアップするために用いられるスクリプトやプログラムがいくつかあります。 それぞれのファイルの目的を簡単に解説しましょう。正確な文法や使い方は xdm(1) に記述されています。

デフォルトの設定では、単純な四角のログインウィンドウがあり、 そこにはマシンのホスト名が大きなフォントで表示され、「Login:」と「Password:」のプロンプトがその下に表示されています。XDM スクリーンの見ためや使い勝手を変えるにはここから始めるのがいいでしょう。

6.6.3.1. Xaccess

XDM がコントロールするディスプレイに接続するためのプロトコルは X Display Manager Connection Protocol (XDMCP) と呼ばれます。このファイルにはリモートのマシンからの XDMCP 接続をコントロールするためのルールセットが書かれます。こ

れは、xdm-config を変更してリモートからのコネクションを待ち受けるようにしない限り無視されます。デフォルトでは、どのクライアントからの接続も拒否します。

6.6.3.2. Xresources

これはディスプレイの chooser とログインスクリーン用の application-defaults ファイルです。このファイルでログインプログラムの見た目を変更することができます。フォーマットは X11 のドキュメントで記述されている app-defaults ファイルのものと同じです。

6.6.3.3. Xservers

これは、chooser が選択肢として提供するリモートのディスプレイの一覧です。

6.6.3.4. Xsession

XDM でログインした後に実行されるデフォルトのセッションスクリプトです。 通常、各ユーザは ~/.xsession というカスタマイズしたセッションスクリプトを持っており、こちらが優先されます。

6.6.3.5. Xsetup_*

これらは chooser やログインインタフェースが表示される前に自動的に実行されます。それぞれのディスプレイに対して、Xsetup_ に続けてローカルのディスプレイ番号を付けた (たとえば Xsetup_0) 名前のついたスクリプトがあります。 典型的な使い方は xconsole のようなバックグラウンドで動かすプログラムを一つか二つ起動することです。

6.6.3.6. xdm-config

app-defaults の書式で書かれた、このインストレーションで管理されるすべてのディスプレイに適用される設定を保持しています。

6.6.3.7. xdm-errors

このファイルには XDM が起動しようとしている X サーバからの出力が書き出されます。 XDM が起動しようとしているディスプレイがなんらかの理由でハングした場合、このファイルのエラーメッセージを見てください。そういったメッセージは各ユーザの ~/.xsession-errors にもセッション毎に書き出されます。

6.6.4. ネットワークディスプレイサーバの起動

あるディスプレイサーバに他のクライアントが接続することができるようにするために、アクセスコントロールのルールを編集し、 コネクションリスナを有効にする必要があります。デフォルトでは保守的な設定になっています。 XDM がそういったコネクションを待ち受けるようにするためには xdm-config にある次の行をコメントアウトします。

```
! SECURITY: do not listen for XDMCP or Chooser requests
```

```
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort:     0
```

そして、XDM を再起動します。app-defaults ファイルにおけるコメントは「!」であっていつものような「#」ではないことに注意してください。アクセス制限はもっと厳しくしてもよいかもしれません。 `Xaccess` にある例を参考にしたり、詳細についてオンラインマニュアル xdm(1) を参照してください。

6.6.5. XDM の代わりになるもの

デフォルトの XDM に代わるものがいくつかあります。一つは KDM (KDE に付属しています) はその一つであり、この章の後ろで解説します。KDM ディスプレイマネージャは、ログイン時にウィンドウマネージャを選ぶことができるのに加え、 見た目もかなり綺麗にしてくれます。

6.7. デスクトップ環境

##: Valentino Vaschetto.

このセクションでは、FreeBSD 上の X で利用可能ないくつかのデスクトップ環境について解説します。「デスクトップ環境」とは、 単なるウィンドウマネージャから KDE や GNOME といったような完全なデスクトップアプリケーションスイートまでカバーします。

6.7.1. GNOME

6.7.1.1. GNOME について

GNOME はユーザフレンドリなデスクトップ環境で、ユーザはコンピュータを簡単に使ったり設定したりできるようになります。GNOME にはパネル (アプリケーションを起動したり状態を表示したりするもの)、デスクトップ (データやアプリケーションが置かれる場所)、標準的なデスクトップツールやアプリケーションのセット、 そしてアプリケーションが互いにうまくやり取りできるような仕組みが含まれています。他の OS や環境に慣れている人でも GNOME の提供するグラフィック環境であれば心地よく感じるでしょう。FreeBSD 上の GNOME に関するもっと詳しい情報は、FreeBSD GNOME Project のウェブサイトで見ることができます。ウェブサイトには、GNOME のインストール、設定、管理に関する非常に分かりやすい FAQ も用意されています。

6.7.1.2. GNOME のインストール

package や Ports Collection を利用してソフトウェアを簡単にインストールできます。

GNOME package をネットワークからインストールするには、 以下のようにするだけです。

```
# pkg_add -r gnome2
```

pkgng ユーザは、以下のコマンドを使用してください。

```
# pkg install gnome2
```

GNOME をソースから構築する場合、次のように ports ツリーを使いましょう。

```
# cd /usr/ports/x11/gnome2
# make install clean
```

GNOME を適切に動作させるには、/proc ファイルシステムをマウントする必要があります。以下を /etc/fstab に追加して、起動中に procfs(5) をマウントするように設定してください。

```
proc            /proc       procfs  rw  0   0
```

GNOME がインストールできたら、 デフォルトのウィンドウマネージャの代わりに GNOME を起動するように X サーバに指示しなければなりません。

GNOME を起動する最も簡単な方法は、GNOME ディスプレイマネージャ GDM を使うことです。GDM は、GNOME デスクトップの一部としてインストールされますが、デフォルトでは無効になっています。有効にするには、以下の行を /etc/rc.conf に追加してください。

```
gdm_enable="YES"
```

再起動すると、GDM が自動的に起動します。

GNOME のすべてのサービスを、GDM とともに起動するようにしておくと良いでしょう。このように設定するには、以下の行を /etc/rc.conf ファイルに追加してください。

```
gnome_enable="YES"
```

GNOME は .xinitrc という名前のファイルを適切に設定することで、コマンドラインから起動することもできます。自分の .xinitrc が既にある場合には、ウィンドウマネージャを起動する行を /usr/local/bin/gnome-session を起動するように変更するだけです。特にこのファイルを用意していない場合には、次のようにすれば十分でしょう。

```
% echo "/usr/local/bin/gnome-session" > ~/.xinitrc
```

次に、startx と入力すれば GNOME デスクトップ環境が起動します。

注記

もし XDM のような古いディスプレイマネージャを使っているなら、 この方法ではうまくいきません。その代わり、実行可能な .xsession というファイルを作成し、 同じコマンドを起動するよ

うにします。そのためには、このファイルを編集してウィンドウマネージャを /usr/local/bin/gnome-session で置き換えます。

```
% echo "#!/bin/sh" > ~/.xsession
% echo "/usr/local/bin/gnome-session" >> ~/.xsession
% chmod +x ~/.xsession
```

もう一つの方法は、 ログイン時にウィンドウマネージャを選択できるようにディスプレイマネージャを設定することです。KDE の詳細についてのセクションで KDE のディスプレイマネージャである KDM を使ってどのようにすればいいのかを解説しています。

6.7.2. KDE

6.7.2.1. KDE について

KDE は最近の簡単に使えるデスクトップ環境です。KDE によりユーザは以下のようなメリットを享受します。

- 美しい現代風のデスクトップ
- ネットワーク透過なデスクトップ
- KDE デスクトップやそのアプリケーションを使う際に、ヘルプへのアクセスを便利で統一されたものにする統合されたヘルプシステム
- すべての KDE アプリケーションで統一された見ためや使い勝手
- 標準化されたメニュー、ツールバー、キーバインディング、カラースキームなど
- 国際化: KDE は 55 を越える言語で利用可能
- 一元的で、首尾一貫した、ダイアログベースのデスクトップ設定
- 膨大な数の KDE アプリケーション

KDE には Konqueror と呼ばれるウェブブラウザも付属しており、これは UNIX® システム上の他のウェブブラウザの強力な競争相手です。KDE の詳細については KDE のウェブサイト をご覧ください。KDE に関する FreeBSD 特有の情報とリソースについては、KDE/FreeBSD initiative のウェブサイトをご覧ください。

6.7.2.2. KDE のインストール

GNOME や他のデスクトップ環境とまったく同じように、package や Ports Collection から簡単にインストールできます。

KDE 4 package をネットワーク越しにインストールするには次のようにします。

```
# pkg_add -r kde4
```

pkg_add(1) は自動的にアプリケーションの最新版を取得します。

pkgng ユーザは、以下のコマンドを使用してください。

```
# pkg install kde4
```

KDE をソースから構築するには、次のように ports ツリーを使いましょう。

```
# cd /usr/ports/x11/kde4
# make install clean
```

port を最初にインストールするときには、 オプションを選択するメニューが表示されます。デフォルトが推奨されます。

KDE 4 は大きなアプリケーションのため、 高速のコンピュータでもコンパイルに長い時間がかかります。

KDE がインストールできたら X サーバに、デフォルトのウィンドウマネージャの代わりにこのアプリケーションを立ち上げるように教えます。`.xinitrc` ファイルを次のように編集しましょう。

```
% echo "exec /usr/local/kde4/bin/startkde" > ~/.xinitrc
```

さぁ、これで `startx` でいつ X Window System を立ち上げても KDE がデスクトップになります。

XDM のようなディスプレイマネージャを使っている場合、設定は少し異なります。代わりに `.xsession` ファイルを編集しましょう。KDM 用の説明はこの章の後のほうにあります。

6.7.3. KDE の詳細について

さぁ、KDE のインストールができました。 ほとんどのことはヘルプページを見たりいろんなメニューをつつけばわかるでしょう。Windows® や Mac® のユーザにも簡単なはずです。

KDE の最も良いリファレンスはオンラインドキュメントです。KDE には独自のウェブブラウザ Konqueror が付属していますし、膨大な数の便利なアプリケーションや詳しいドキュメントも用意されています。 このセクションの残りではなんとなく使っているだけでは理解し難い技術的なところを解説します。

6.7.3.1. KDE ディスプレイマネージャ

マルチユーザシステムの管理者であれば、ユーザを迎えるにあたってグラフィカルなログイン画面が欲しいと思うかもしれません。前の章で述べたように XDM はそのために使

えますが、KDE にはその代わりになる KDM が付いています。見た目がより魅力的で、ログイン時のオプションもたくさんあります。特に (メニューを使って) ログイン後に使うデスクトップ環境 (KDE や GNOME など) を簡単に選ぶことができます。

KDE 4 では、procfs(5) をマウントする必要があります。`/etc/rc.conf` に以下の行を追加してください。

```
kdm4_enable="YES"
```

6.7.4. Xfce

6.7.4.1. Xfce について

Xfce は GNOME で使われている GTK+ ツールキットをベースにしたデスクトップ環境ですが、ずっと軽いので、シンプルで効率的でありながら使いやすく設定が簡単なデスクトップ環境を求める人に向いています。見ためは商用 UNIX® システムが採用している CDE にかなり似ています。Xfce の機能をいくつか以下に挙げます。

- シンプルで使いやすいデスクトップ
- マウスのドラッグアンドドロップなどですべての設定が可能
- CDE に似たメインパネルとメニューやアプレット、アプリケーションランチャ
- 統合されたウィンドウマネージャ、ファイルマネージャ、サウンドマネージャと GNOME 準拠のモジュールなど
- (GTK+ なので) テーマをサポート
- 速くて軽くて効率的: 古いマシンや遅いマシン、メモリの限られたマシンに向いています

Xfce に関する詳しい情報は Xfce のウェブサイト で得られます。

6.7.4.2. Xfce のインストール

Xfce をネットワークからインストールするには、次のように実行してください。

```
# pkg_add -r xfce4
```

pkgng ユーザは、以下のコマンドを使用してください。

```
# pkg install xfce4
```

また、Ports Collection を利用してソースからインストールすることも可能です。

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```

インストール後、次回 X を起動した時に Xfce が立ち上がるように設定します。次のようにしてください。

```
% echo "/usr/local/bin/startxfce4" > ~/.xinitrc
```

X を次に起動すると Xfce が立ち上がります。XDM のようなディスプレイマネージャを使っている場合には、GNOME のセクションに書いてあるように `.xsession` ファイルを作成し、`/usr/local/bin/startxfce4` を起動するように追加します。または KDM のセクションにあるように、ログイン時にディスプレイマネージャからデスクトップ環境を選択できるように設定してください。

パート II. 日々の生活

第 1 部では基礎的なことがらを説明したので、FreeBSD ハンドブックの第 2 部では FreeBSD でよく使われる機能について説明します。各章の内容は以下のとおりです。

- ブラウザ、生産的なツール、ドキュメントビューアといった、 人気があって便利なデスクトップアプリケーションの紹介
- FreeBSD で利用可能なマルチメディアツールの紹介
- システムで特別な機能を有効にするために、カスタムカーネルを構築する手順の説明
- デスクトップおよびネットワーク接続両方のプリンタの設定に関する、 印刷システムの詳細な説明
- FreeBSD システムで Linux アプリケーションを実行する方法

これらの章では、読み飛ばしを推奨しているものもあります。 これについてはそれぞれの章の始めにある概要に書かれています。

目次

7. デスクトップアプリケーション 245
    7.1. この章では 245
    7.2. ブラウザ 246
    7.3. 生産的なアプリケーション 251
    7.4. ドキュメントビューア 255
    7.5. 財務管理ソフトウェア 257
8. マルチメディア 261
    8.1. この章では 261
    8.2. サウンドカードの設定 262
    8.3. MP3 オーディオ 267
    8.4. ビデオ再生 270
    8.5. TV カードの設定 276
    8.6. MythTV 278
    8.7. 画像スキャナ 279
9. FreeBSD カーネルのコンフィグレーション 285
    9.1. この章では 285
    9.2. なぜカスタムカーネルを作るか? 285
    9.3. システムのハードウェアについて知る 287
    9.4. コンフィグレーションファイル 288
    9.5. カスタムカーネルの構築とインストール 290
    9.6. 問題が起きた場合には 291
10. プリンタの利用 293
    10.1. この章では 293
    10.2. はじめに 293
    10.3. 基本的な設定 295
    10.4. プリンタ設定上級編 311
    10.5. プリンタを使う 348
    10.6. 標準スプーラの代替品 357
    10.7. トラブルシューティング 358
11. Linux バイナリ互換機能 365
    11.1. この章では 365
    11.2. インストール 366
    11.3. Mathematica のインストール 370
    11.4. Oracle のインストール 372
    11.5. SAP R/3 (4.6B - IDES) のインストール 376
    11.6. 高度なトピックス 395

第7章 デスクトップアプリケーション

7.1. この章では

FreeBSD は性能や安定性によりサーバとして人気がある一方で、 日々のデスクトップとしての利用にも適しています。 packages や ports から 24,000 を超えるアプリケーションを利用できるので、 さまざまなアプリケーションを動かせるようにカスタマイズしたデスクトップを作り上げることができます。この章では、ウェブブラウザ、生産的なソフトウェア、ドキュメントビューア、および財務管理ソフトウェアといった、数多くのデスクトップアプリケーションのインストール方法について説明します。

注記

一から構築するのではなく、 事前に構築されたデスクトップバージョンの FreeBSD をお望みのユーザは、pcbsd.org ウェブサイトをご覧ください。

この章の読者は、以下のことを理解しておく必要があります。

- package または ports を用いたサードパーティ製ソフトウェアのインストール方法 (5章アプリケーションのインストール - packages と ports)。
- X およびウィンドウマネージャのインストール方法 (6章X Window System)。

マルチメディア環境を整える方法については 8章マルチメディア を参照してください。

訳注

この文書は英語で書かれている原文をそのまま邦訳したものです。 必ずしも各アプリケーションで日本語が扱えるとは限らないことに注意してください。 日本語に対応したアプリケーションは、Ports Collection の japanese ディレクトリにあるかもしれません。

7.2. ブラウザ

FreeBSD では Web ブラウザは事前にインストールされていません。そのかわり、Ports Collection の www カテゴリには数多くの Web ブラウザ が用意されており、多くのプログラムを packages からインストールしたり、Ports Collection からコンパイルできます。

KDE や GNOME デスクトップ環境には、それぞれ HTML ブラウザが用意されています。これらのデスクトップ環境を設定するための情報については「デスクトップ環境」を参照してください。

軽量なブラウザには、www/dillo2, www/links, および www/w3m といったものがあります。

この節では、広く使われている以下の web ブラウザのインストール方法について説明します。もし、アプリケーションがリソースを大量に消費したり、ports からのコンパイルに時間がかかったり、他の ports に大きく依存する場合には、そのことについても触れます。

アプリケーション名	必要なリソース	port からのインストール	備考
Firefox	中	重	FreeBSD, Linux® および地域化されたバージョンを利用できます。
Opera	少	軽	FreeBSD と Linux® 版を利用できます。
Konqueror	中	重	KDE ライブラリを必要とします。
Chromium	中	重	Gtk+ を必要とします。

7.2.1. Firefox

Firefox は、FreeBSD に完全に対応したオープンソースのブラウザです。標準に準拠した HTML 表示エンジン、タブブラウジング、ポップアップブロック、拡張性、高い安全性などが特徴です。Firefox は Mozilla のコードベースから派生したブラウザです。

最新の Firefox の package をインストールするには以下のように入力してください。

```
# pkg install firefox
```

Firefox 延長サポート版 (ESR: Extended Support Release) を利用したい場合には、かわりに以下のように入力してください。

```
# pkg install firefox-esr
```

ローカライズ版は、www/firefox-i18n および www/firefox-esr-i18n から利用できます。

かわりにソースコードから希望の firefox をコンパイルすることもできます。この例では www/firefox をビルドしますが、`firefox` の部分は、インストールする ESR やローカライズに置き換えることもできます。

```
# cd /usr/ports/www/firefox
# make install clean
```

7.2.1.1. Firefox と Java™ プラグイン

Firefox をインストールしただけでは、Java™ には対応していません。java/icedtea-web は、Java アプレットを実行するための、フリーソフトの web ブラウザのプラグインです。package からインストールできます。port からコンパイルするには、以下のようにしてください。

```
# cd /usr/ports/java/icedtea-web
# make install clean
```

ports をコンパイルする際のコンフィグレーションオプションは、デフォルトのままにしてください。

インストールしたら、ブラウザを起動し、ロケーションバーに `about:plugins` と入力して Enter を押してください。インストールされているプラグイン一覧のページが表示されます。Java™ プラグインが一覧に追加されているでしょう。

プラグインが一覧に追加されていない場合には、ユーザごとに以下のコマンドを実行し、ブラウザを再起動してください。

```
% ln -s /usr/local/lib/IcedTeaPlugin.so \
  $HOME/.mozilla/plugins/
```

7.2.1.2. Firefox と Adobe® Flash® プラグイン

FreeBSD 用の Adobe® Flash® プラグインは提供されていませんが、Linux® 版のプラグインを実行するためのソフトウェア wrapper を利用できます。この wrapper は、RealPlayer® プラグインといった他のブラウザのプラグインにも対応しています。

プラグインをインストールして有効にするには、以下を実行してください。

1. www/nspluginwrapper port をインストールしてください。ライセンスの制限のため package は利用できません。この port は、emulators/linux_base-f10 という大きな port を必要とします。

2. www/linux-f10-flashplugin11 port をインストールしてください。ライセンスの制限により、package は利用できません。

3. ```
   # ln -s /usr/local/lib/browser_plugins/linux-f10-
   flashplugin/libflashplayer.so \
     /usr/local/lib/browser_plugins/
   ```

   `/usr/local/lib/browser_plugins` ディレクトリが存在しない場合には、作成してください。

4. プラグインを最初に使う前に、ユーザごとに以下を実行しておく必要があります。

   ```
   % nspluginwrapper -v -a -i
   ```

   プラグインの port がアップデートされ、再インストールされた際には、各ユーザは以下を実行する必要があります。

   ```
   % nspluginwrapper -v -a -u
   ```

   ブラウザを起動し、ロケーションバーに `about:plugins` と入力して Enter を押してください。現在利用可能なプラグインの一覧が表示されます。

7.2.1.3. Firefox と Swfdec Flash® プラグイン

Swfdec は Flash® アニメーションのデコーダーおよびレンダラーです。 Swfdec-Mozilla は、Firefox ブラウザにおいて SWF ファイルを実行するためのプラグインで、Swfdec ライブラリを利用しています。

package をインストールするには、以下を実行してください。

```
# pkg install swfdec-plugin
```

package を利用できない場合は、Ports Collection からコンパイルしてインストールしてください。

```
# cd /usr/ports/www/swfdec-plugin
# make install clean
```

その後、ブラウザを再起動してプラグインを有効にしてください。

7.2.2. Opera
```

Opera は、機能満載で、標準規格に準拠したブラウザで、軽量で高速です。メールおよびニュースリーダや、IRC クライアント、RSS/Atom フィードリーダ、その他いろいろなものが組み込まれています。FreeBSD ネイティブ版と、Linux® バイナリ互換機能によって動作するバージョンを利用できます。

以下のコマンドは、FreeBSD 版の Opera package をインストールします。Linux® 版をインストールするには、`opera` を `linux-opera` に置き換えてください。

```
# pkg install opera
```

あるいは Ports Collection からインストールしてください。以下のコマンドで、ネイティブ版がインストールされます。

```
# cd /usr/ports/www/opera
# make install clean
```

Linux® 版の Opera をインストールするには、`opera` を `linux-opera` に置き換えてください。

Adobe® Flash® プラグインをインストールするには、まず www/linux-f10-flashplugin11 port をコンパイルしてください。package は、ライセンスの制限により利用できません。その後、www/opera-linuxplugins port もしくは package をインストールしてください。以下の例では、両方のアプリケーションを ports からコンパイルします。

```
# cd /usr/ports/www/linux-f10-flashplugin11
# make install clean
# cd /usr/ports/www/opera-linuxplugins
# make install clean
```

インストールが終了したら、プラグインがインストールされたかどうかをブラウザを起動して確認してください。ロケーションバーに `about:plugins` と入力して Enter を押してください。現在利用可能なプラグインの一覧が表示されます。

Java™ プラグインを追加するには、「Firefox と Java™ プラグイン」に書かれている手順に従ってください。

7.2.3. Konqueror

Konqueror はブラウザであると同時に、ファイルマネージャおよびマルチメディアビューアの役割も果たします。x11/kde4-baseapps package または port に含まれています。

Konqueror は、KHTML とともに、WebKit にも対応しています。WebKit は Chromium など最近のブラウザの多くで採用されているレンダリングエンジンです。

FreeBSD の Konqueror で WebKit を使うには、www/kwebkitpart package または port をインストールしてください。以下の例では、port をコンパイルします。

```
# cd /usr/ports/www/kwebkitpart
# make install clean
```

Konqueror で、WebKit を有効にするには、「Settings」,「Configure Konqueror」をクリックしてください。「General」の設定ページにおいて、「Default web browser engine」の隣の ドロップダウンメニューをクリックし、「WebKit」を「KHTML」に変更してください。

Konqueror は Flash® にも対応しています。Konqueror に Flash® を導入するための「How To」ガイドが `http://freebsd.kde.org/howtos/konqueror-flash.php` にあります。

7.2.4. Chromium

Chromium は、 オープンソースのブラウザのプロジェクトで、 より安全かつより高速、より安定したウェブブラウジングを目指しています。 Chromium は、タブブラウジング、ポップアップブロック、拡張機能などの機能を持っています。 Chromium は、Google Chrome ウェブブラウザがベースとしているオープンソースのプロジェクトです。

Chromium は、以下のように入力することで package からインストールできます。

```
# pkg install chromium
```

または、Ports Collection を用いて ソースから Chromium をコンパイルしてインストールできます。

```
# cd /usr/ports/www/chromium
# make install clean
```

注記

Chromium の実行可能ファイルは、 `/usr/local/bin/chrome` です。`/usr/local/bin/chromium` ではありません。

7.2.4.1. Chromium と Java™ プラグイン

Chromium をインストールしただけでは、Java™ には対応していません。Java™ プラグインをインストールするには、「Firefox と Java™ プラグイン」に書かれている手順に従ってください。

Java™ のサポートをインストールしたら、 Chromium を起動し、アドレスバーに、 `about:plugins` と入力してください。 インストールされているプラグインの一覧に IcedTea-Web が追加されているはずです。

Chromium に IcedTea-Web プラグインが追加されていない場合には、以下のコマンドを実行し、ブラウザを再起動してください。

```
# mkdir -p /usr/local/share/chromium/plugins
# ln -s /usr/local/lib/IcedTeaPlugin.so \
  /usr/local/share/chromium/plugins/
```

7.2.4.2. Chromium と Adobe® Flash® プラグイン

Chromium と Adobe® Flash® の設定は、「Firefox と Adobe® Flash® プラグイン」に書かれている手順と同じです。Chromium は、他のブラウザのプラグインを用いることができるので、追加の設定は必要ありません。

7.3. 生産的なアプリケーション

生産的なアプリケーションということになると、 新しいユーザはしばしばオフィススイートや、 使いやすい文書作成ソフトウェアを求めるでしょう。 デフォルトの生産的なアプリケーションはありませんが、 KDE のような デスクトップ環境 はオフィススイートを提供しています。インストールされているウィンドウマネージャにかかわらず、FreeBSD では、いくつものオフィススイート、グラフィカルな文書作成ソフトウェアを利用できます。

この節では、 以下の人気のある生産的なソフトウェアのインストール方法について説明します。もし、アプリケーションがリソースを大量に消費したり、ports からのコンパイルに時間がかかったり、もしくは他の ports に大きく依存する場合には、そのことについても触れます。

アプリケーション名	必要なリソース	port からのインストール	実行に必要となる主な環境
Calligra	少	重	KDE
AbiWord	少	軽	Gtk+ または GNOME
Gimp	少	重	Gtk+
Apache OpenOffice	多	莫大	JDK™ および Mozilla
LibreOffice	やや多	莫大	Gtk+ または KDE/GNOME または JDK™

7.3.1. Calligra

KDE デスクトップには、 KDE 環境以外でも利用可能なオフィススイートがあります。Calligra には、他のオフィススイートと同様に、 標準的なアプリケーションが含まれています。Words は文書作成ソフトウェア、Sheets は表計算ソフトウェア、Stage はプレゼンテーションソフトウェア、そして Karbon は図形描画ソフトウェアです。

FreeBSD では package または port から editors/calligra をインストール出来ます。package からインストールするには次のようにします。

```
# pkg install calligra
```

package を入手できない場合は、かわりに Ports Collection を利用してください。

```
# cd /usr/ports/editors/calligra
# make install clean
```

7.3.2. AbiWord

AbiWord は、Microsoft® Word のような見た目や操作感を持つフリーの文書作成ソフトウェアです。速く、多くの機能を持ち、ユーザフレンドリです。

AbiWord は、Microsoft® `.rtf` のような独自仕様を含む多くの形式のファイルを読み書きできます。

AbiWord package をインストールするには、以下のようにしてください。

```
# pkg install abiword
```

package を入手できない場合は、Ports Collection からコンパイルしてください。

```
# cd /usr/ports/editors/abiword
# make install clean
```

7.3.3. GIMP

画像を描画したり写真を修正することに関して、 GIMP は洗練された編集プログラムです。 単純にお絵かきソフトウェアとして使うこともできますし、 高品質な写真の加工ツールとしても使えます。多くのプラグインに対応しており、スクリプトインタフェースを特徴としています。GIMP はさまざまな形式のファイルを読み書きでき、スキャナやタブレットとのインタフェースにも対応しています。

package をインストールするには、以下のようにしてください。

```
# pkg install gimp
```

もしくは、Ports Collection を利用してください。

```
# cd /usr/ports/graphics/gimp
# make install clean
```

Ports Collection の graphics カテゴリ (freebsd.org/ja/ports/graphics.html) には、GIMP に関連したプラグイン、ヘルプファイルおよびユーザマニュアルなどがあります。

7.3.4. Apache OpenOffice

Apache OpenOffice は、Apache Software Foundation のインキュベータプロジェクトとして開発が行われているオープンソースのオフィススイートです。Apache OpenOffice は、完全なオフィススイートに必須のアプリケーション (文書作成ソフトウェア、表計算ソフトウェア、プレゼンテーションソフトウェア、そして図形描画ソフトウェア) をひととおり揃えています。ユーザインタフェースは他のオフィススイートと似ており、広く用いられているさまざまな形式のファイルを読み書きできます。多くの言語で利用でき、インタフェース、スペルチェッカ、辞書は国際化されています。

Apache OpenOffice の文書作成ソフトウェアは、ネイティブの XML ファイル形式を採用することでポータビリティや柔軟性を高めています。表計算ソフトウェアにはマクロ機能があり、外部データベースと接続することもできます。Apache OpenOffice は、Windows®, Solaris™, Linux®, FreeBSD および Mac OS® X において安定してネイティブに動作しています。Apache OpenOffice についてのより詳しい情報は、openoffice.org をご覧ください。また、porting.openoffice.org/freebsd/ から、FreeBSD 特有の情報を参照してください。

Apache OpenOffice package をインストールするには、以下のように入力してください。

```
# pkg install apache-openoffice
```

package をインストールしたら、以下のコマンドを入力して Apache OpenOffice を起動してください。

```
% openoffice-X.Y.Z
```

ここで *X.Y.Z* は、インストールされている Apache OpenOffice のバージョン番号です。Apache OpenOffice の初回起動時に、いくつかの質問が行われ、ユーザのホームディレクトリに `.openoffice.org` フォルダが作成されます。

希望の Apache OpenOffice の packages を利用できない場合には、port を利用する方法もあります。しかしながら、コンパイルには大きなディスクスペースと、本当にかなり長い時間を必要とします。

```
# cd /usr/ports/editors/openoffice-4
# make install clean
```

注記

地域化されたバージョンをビルドするには、 上記のコマンドの代わりに以下を実行して下さい。

```
# make LOCALIZED_LANG= your_language ↺
install clean
```

your_language を正しい言語 ISO コードに置き換えてください。サポートされている言語コードは、同じ port ディレクトリにある `files/Makefile.localized` に書かれています。

7.3.5. LibreOffice

LibreOffice は、documentfoundation.org が開発しているフリーソフトウェアのオフィススイートです。他のメジャーなオフィススイートと互換性があり、さまざまなプラットフォームで利用できます。 OpenOffice.org からの新しいフォークで、 完全なオフィススイートに必須のアプリケーション (文書作成ソフトウェア、表計算ソフトウェア、 プレゼンテーションソフトウェア、図形描画ソフトウェア、 データベース管理ソフトウェア、数式エディタ) をすべて揃えています。多くの言語で利用でき、インタフェース、スペルチェッカ、辞書は国際化されています。

LibreOffice の文書作成ソフトウェアは、 ネイティブのファイル形式に XML を採用することで ポータビリティや柔軟性を高めています。表計算ソフトウェアにはマクロ機能があり、 外部データベースと接続することもできます。 LibreOffice は、 Windows®, Solaris™, Linux®, FreeBSD, Mac OS® X において安定してネイティブに動作しています。LibreOffice についての詳しい情報は、libreoffice.org をご覧ください。

英語版の LibreOffice package をインストールするには、以下のように入力してください。

```
# pkg install libreoffice
```

Ports Collection の edtors カテゴリ (freebsd.org/ja/ports/editors.html) カテゴリには、地域化された LibreOffice が用意されています。地域化された package をインストールするには、`libreoffice` を地域化された package 名に置き換えてください。

package をインストールしたら、以下のコマンドで LibreOffice を起動してください。

```
% libreoffice
```

初回起動時には、いくつかの質問が行われ、 ユーザのホームディレクトリに `.libreoffice` フォルダが作成されます。

希望の LibreOffice の packages を利用できない場合には、port からコンパイルする方法もあります。しかしながら、コンパイルには大きなディスクスペースと、 本当にかなり長い時間を必要とします。以下の例では、英語版をコンパイルします。

```
# cd /usr/ports/editors/libreoffice
# make install clean
```

注記

地域化されたバージョンをビルドしたいのなら、 希望の言語の port ディレクトリに `cd` コマンドで移動してください。対応している言語は、Ports Collection の editors カテゴリ (freebsd.org/ja/ports/editors.html) にあります。

7.4. ドキュメントビューア

UNIX® の出現以降、いくつかの新しい文書形式が広く使われるようになりました。基本システムには、それらの文書が要求するビューアがないかもしれません。 この節ではそれらのドキュメントビューアのインストール方法について説明します。

アプリケーション名	必要なリソース	port からのインストール	実行に必要になる主な環境
Xpdf	少	軽	FreeType
gv	少	軽	Xaw3d
GQview	少	軽	Gtk+ または GNOME
ePDFView	少	軽	Gtk+ または GNOME
Okular	少	重	KDE

7.4.1. Xpdf

FreeBSD 向けの軽い PDF ビューアを使いたいのなら Xpdf を試してみてください。これは少ないリソースで動作するビューアで、軽くて効率的です。標準の X フォントを利用し、他の X ツールキットを必要としません。

Xpdf の package をインストールするには次のコマンドを入力してください。

```
# pkg install xpdf
```

package を入手できない場合は、Ports Collection を利用してください。

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

インストールが完了したら xpdf を起動してください。メニューを表示するにはマウスの右ボタンを押してください。

7.4.2. gv

gv は PostScript® と PDF のビューアです。これは ghostview をベースとしていますが、Xaw3d ウィジットツールキットによってより良い外観になっています。gv は向きや用紙のサイズ、拡大縮小、アンチエイリアスなどたくさんの設定可能な機能を持っています。ほとんどすべての操作をキーボードかマウスのどちらかだけで行なうことができます。

package から gv をインストールするには次のようにします。

```
# pkg install gv
```

package を利用できない場合には、Ports Collection を使ってください。

```
# cd /usr/ports/print/gv
# make install clean
```

7.4.3. GQview

GQview は画像管理ソフトウェアです。 クリックひとつで画像ファイルを開いたり、外部エディタを起動したり、サムネイル画像を作成できます。また、スライドショーや基本的なファイル操作機能も備えており、 画像のコレクションの管理や、 重複したファイルを見つけることが簡単にできます。 GQview は全画面表示、 および国際化にも対応しています。

GQview package をインストールするには次のコマンドを入力してください。

```
# pkg install gqview
```

package を入手できない場合は、Ports Collection を利用してください。

```
# cd /usr/ports/graphics/gqview
# make install clean
```

7.4.4. ePDFView

ePDFView は軽量な PDF ドキュメントビューアです。このビューアは、Gtk+ および Poppler ライブラリのみを使います。 このソフトウェアは、現在開発中ですが、ほぼすべ

ての PDF ファイル (暗号化されたものを含む) を開くことが可能で、ドキュメントのコピーを保存でき、CUPS を用いた印刷にも対応しています。

package から ePDFView をインストールするには以下のようにしてください。

```
# pkg install epdfview
```

package が利用できないようでしたら、Ports Collection を使ってインストールしてください。

```
# cd /usr/ports/graphics/epdfview
# make install clean
```

7.4.5. Okular

Okular は、KDE の KPDF をベースとした一般的なドキュメントビューアです。このビューアは、PDF, PostScript®, DjVu, CHM, XPS, および ePub といった、多くの形式のファイルを開くことができます。

package で Okular をインストールするには、以下のようにしてください。

```
# pkg install okular
```

package が利用できないようでしたら、Ports Collection を使ってインストールしてください。

```
# cd /usr/ports/graphics/okular
# make install clean
```

7.5. 財務管理ソフトウェア

FreeBSD のデスクトップで個人的な財務管理ができるように、強力で簡単に使えるアプリケーションが用意されています。それらのアプリケーションの中には Quicken や Excel などの広く行き渡った形式のファイルと互換性があるものもあります。

この節では次のアプリケーションについて説明します。

アプリケーション名	必要なリソース	port からのインストール	実行に必要になる主な環境
GnuCash	少	重	GNOME
Gnumeric	少	重	GNOME
KMyMoney	少	重	KDE

7.5.1. GnuCash

GnuCash は、GNOME の一部で、使いやすくかつ強力なアプリケーションとしてエンドユーザに提供されています。GnuCash を使えば、収入や支出、銀行口座、あるいは株を管理できます。直観的なインタフェースを特徴としていますが、高度な機能も提供しています。

GnuCash は洗練された登録機能、階層構造の勘定システム、多くのキーボードショートカット、自動補完機能を提供しています。単一のトランザクションをより小さな要素に分解できます。GnuCash は、Quicken の QIF ファイルの読み込みやマージができます。また、国際的な日付および通貨形式も扱えます。

GnuCash package をインストールするには次のようにしてください。

```
# pkg install gnucash
```

package が手に入らなければ、Ports Collection を使ってください。

```
# cd /usr/ports/finance/gnucash
# make install clean
```

7.5.2. Gnumeric

Gnumeric は、GNOME コミュニティによって開発されている表計算ソフトウェアです。セルの書式に従ってユーザの入力を自動的に推測する便利な機能や、多くのシーケンスに対する自動補完機能があります。Excel, Lotus 1-2-3, Quattro Pro といった広く行き渡っている多くの形式のファイルを読みこめます。多くの関数を内蔵しており、数値、通貨、日付、時間などのよく使うセルの書式が利用できます。

Gnumeric package をインストールするには次のように入力してください。

```
# pkg install gnumeric
```

package が手に入らなければ、Ports Collection を使ってください。

```
# cd /usr/ports/math/gnumeric
# make install clean
```

7.5.3. KMyMoney

KMyMoney は、KDE コミュニティが作成している個人用財務管理アプリケーションです。KMyMoney は、商用の個人用財務管理ソフトウェアに見られる重要な機能を提供することを目指しています。 また、使いやすい複式簿記機能も特徴です。 KMyMoney は標準の Quicken QIF ファイルをインポート可能で、投資履歴や複数通貨の取扱い、財政状況のレポートを提供します。

package から KMyMoney をインストールするには次のようにします。

```
# pkg install kmymoney-kde4
```

package が手に入らない場合は、Ports Collection を使ってください。

```
# cd /usr/ports/finance/kmymoney-kde4
# make install clean
```

第8章 マルチメディア

編集: Ross Lippert.

8.1. この章では

FreeBSD は数多くの種類のサウンドカードに対応しており、FreeBSD システムで原音に忠実な出力を楽しむことができます。これには録音機能と、MPEG Audio Layer 3 (MP3) や Waveform Audio File (WAV), Ogg Vorbis などをはじめとした多くの形式の音楽の再生機能が含まれます。加えて FreeBSD の Ports Collection には、録音した音楽を編集したり、音響効果を加えたり、接続された MIDI 機器を制御するためのアプリケーションが用意されています。

FreeBSD ではビデオファイルおよび DVD の再生もできます。FreeBSD の Ports Collection には、さまざまなビデオメディアをエンコード、変換、再生するアプリケーションが用意されています。

この章では FreeBSD 上でサウンドカード、ビデオの再生、TV チューナカード、スキャナを設定する方法について説明します。また、これらのデバイスを使うためのアプリケーションについても説明します。

この章を読むと、以下のことがわかります。

- FreeBSD でのサウンドカードの設定方法
- サウンドの設定に関するトラブルシューティング
- MP3 およびその他の形式の音声を再生、エンコードする方法
- FreeBSD システムでのビデオ再生の準備
- DVD, `.mpg` および `.avi` ファイルを再生する方法
- CD および DVD の情報をファイルに抽出する方法
- TV カードの設定方法
- MythTV を FreeBSD にインストールして設定する方法
- 画像スキャナの設定方法

この章を読む前に、以下のことを理解しておく必要があります。

- アプリケーションのインストール方法 (5章アプリケーションのインストール - packages と ports)

8.2. サウンドカードの設定

##: Moses Moore.
###: Marc Fonvieille.

設定をはじめる前に、サウンドカードのモデル、そのカードが使用しているチップを確認してください。FreeBSD は サウンドカードに幅広く対応しています。使用しているカードが対応しているかどうか、どの FreeBSD ドライバを使うかについて、ハードウェアノート の対応オーディオデバイスの一覧を確認してください。

サウンドデバイスを使うためには、デバイスドライバを読み込まなければいけません。もっとも簡単な方法は kldload(8) を使ってサウンドカードのカーネルモジュールを読み込むことです。 次の例は、Intel 仕様のビルトインオーディオチップセットのドライバを読み込む例です。

```
# kldload snd_hda
```

このドライバを起動時に読み込むように設定するためには、`/boot/loader.conf` にドライバを追加してください。このドライバの場合は以下の行になります。

```
snd_hda_load="YES"
```

他に利用可能な読み込み可能なサウンドモジュールは `/boot/defaults/loader.conf` に記載されています。どのドライバを利用すればいいか確かでなければ、`snd_driver` モジュールを読み込んでください。

```
# kldload snd_driver
```

`snd_driver` モジュールは、一般に使用されるカードに対応したドライバをまとめて一度に読み込むメタドライバです。このドライバを使用すれば、速やかに正しいドライバを探し出すことができます。`/boot/loader.conf` ファイルを使用して、すべてのサウンドドライバを読み込むこともできます。

`snd_driver` メタドライバの読み込み後に、どのドライバがサウンドカードに選択されたのかを知るには、`cat /dev/sndstat` と入力してください。

8.2.1. サウンドに対応したカスタムカーネルを設定する

この節は、 サウンドカードのドライバをカーネルへ静的に組み込もうと考えているユーザ向けです。カーネル再構築の詳細は 9章FreeBSD カーネルのコンフィグレーション を参照してください。

サウンドに対応したカスタムカーネルを使うときには、 オーディオフレームワークドライバをカーネルコンフィグレーションファイルに追加してください。

```
device sound
```

次に、サウンドカードに対応したドライバを追加します。 前節の Intel 仕様のビルトインオーディオチップセットの例では、 カスタムカーネルコンフィグレーションファイルに以下の行を追加してください。

```
device snd_hda
```

ドライバのマニュアルページを読んで、ドライバが使用するデバイス名を調べてください。

PnP 非対応の ISA サウンドカードでは、 IRQ および I/O ポートの設定を `/boot/device.hints` に指定する必要があるかもしれません。システムの起動時に、loader(8) はこのファイルを読み、設定情報をカーネルに渡します。たとえば、PnP 非対応の古い Creative SoundBlaster® 16 (ISA 接続) には `snd_sb16` とともに snd_sbc(4) ドライバを使用します。このカードを使用する場合には、カーネルコンフィグレーションファイルに以下の行を追加してください。

```
device snd_sbc
device snd_sb16
```

もしカードが `0x220` I/O port と IRQ `5` を使用している場合には、 `/boot/device.hints` に以下の行を追加してください。

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

この例では、I/O ポートに `0x220` を、IRQ に `5` を使用します。

`/boot/device.hints` に用いるべき構文は、sound(4) および、サウンドカードの各ドライバのマニュアルページに記載されています。

これまでの設定はデフォルトのものです。 カードを使用する状況によっては、 IRQ やその他の設定を変更する必要があるかもしれません。 このカードについての詳細は、snd_sbc(4) をご覧ください。

8.2.2. サウンドのテスト

必要となるモジュールを読み込むか、カスタムカーネルで再起動すると、 サウンドカードが検出されます。確認をするには、`dmesg | grep pcm` と実行してください。この例は、ビルトイン Conexant CX20590 チップセットを搭載したシステムのものです。

```
pcm0: <NVIDIA (0x001c) (HDMI/DP 8ch)> at nid 5 on hdaa0
pcm1: <NVIDIA (0x001c) (HDMI/DP 8ch)> at nid 6 on hdaa0
pcm2: <Conexant CX20590 (Analog 2.0+HP/2.0)> at nid 31,25 and 35,27 ↺
on hdaa1
```

サウンドカードの状態は、以下のコマンドを使用して確認することもできます。

```
# cat /dev/sndstat
FreeBSD Audio Driver (newpcm: 64bit 2009061500/amd64)
Installed devices:
pcm0: <NVIDIA (0x001c) (HDMI/DP 8ch)> (play)
pcm1: <NVIDIA (0x001c) (HDMI/DP 8ch)> (play)
pcm2: <Conexant CX20590 (Analog 2.0+HP/2.0)> (play/rec) default
```

この出力は、サウンドカードによって異なります。`pcm` デバイスがなければ、適切なデバイスドライバが読み込まれているか、 カーネルに追加されてコンパイルされているかどうかを確認してください。次の節では、良くある問題とその解決方法をリストアップしています。

すべてうまくいけば、サウンドカードが FreeBSD で機能するでしょう。 CD または DVD ドライブのオーディオ出力端子がサウンドカードと適切に接続されていれば、cdcontrol(1) を使ってドライブ内のオーディオ CD を再生できます。

```
% cdcontrol -f /dev/acd0 play 1
```

警告

オーディオ CD は特別なエンコーディングが行われているため、mount(8) を使ってマウントすべきではありません。

audio/workman のように、よりよいインタフェースを提供するさまざまなアプリケーションがあります。audio/mpg123 port をインストールして MP3 オーディオファイルを聞くことができます。

手っ取り早くカードをテストするには、`/dev/dsp` デバイスにデータを送ってみてください。

```
% cat filename > /dev/dsp
```

ここで *`filename`* は、どのような形式のファイルでも構いません。 このコマンドラインを実行すると雑音が発生するはずです。これにより、サウンドカードが動作していることを確認できます。

注記

`/dev/dsp*` デバイスノードは、 必要に応じて自動的に作成されます。 デバイスノードが使用されていない場合には存在せず、ls(1) の出力に表示されません。

8.2.3. サウンドカードの問題についてのトラブルシューティング

表 8.1 は、良くあるエラーメッセージとその解決法の一覧です。

表8.1 良くあるエラーメッセージ

エラー	解決方法
sb_dspwr(XX) timed out	使用する I/O ポートが適切に設定されていません。
bad irq XX	使用する IRQ が正しく設定されていません。サウンドカードの IRQ と設定した IRQ が同じかどうか確かめてください。
xxx: gus pcm not attached, out of memory	デバイスを使用するのに十分なメモリを確保できません。
xxx: can't open /dev/dsp!	`fstat | grep dsp` と入力して、他のアプリケーションがデバイスを使用しているか調べてください。 注目すべきトラブルメーカは esound と KDE のサウンド機能です。

最近のグラフィックカードの中には、HDMI を利用するため、グラフィックカード自身がサウンドカードを持つものがあります。このようなサウンドデバイスには、時としてサウンドカードより若い番号が付けられることがあります。そのような場合には、サウンドカードをデフォルトプレイバックデバイスとして利用できません。このことが原因かどうかを確認するには、dmesg を実行して `pcm` を探してください。以下のような出力を得るかもしれません。

```
...
hdac0: HDA Driver Revision: 20100226_0142
hdac1: HDA Driver Revision: 20100226_0142
hdac0: HDA Codec #0: NVidia (Unknown)
hdac0: HDA Codec #1: NVidia (Unknown)
hdac0: HDA Codec #2: NVidia (Unknown)
hdac0: HDA Codec #3: NVidia (Unknown)
pcm0: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 0 nid 1 on ↺
hdac0
pcm1: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 1 nid 1 on ↺
hdac0
pcm2: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 2 nid 1 on ↺
hdac0
pcm3: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 3 nid 1 on ↺
hdac0
hdac1: HDA Codec #2: Realtek ALC889
pcm4: <HDA Realtek ALC889 PCM #0 Analog> at cad 2 nid 1 on hdac1
pcm5: <HDA Realtek ALC889 PCM #1 Analog> at cad 2 nid 1 on hdac1
pcm6: <HDA Realtek ALC889 PCM #2 Digital> at cad 2 nid 1 on hdac1
pcm7: <HDA Realtek ALC889 PCM #3 Digital> at cad 2 nid 1 on hdac1
```

```
...
```

この例では、グラフィックカード (NVidia) には、サウンドカード (Realtek ALC889) より若い番号が付けられています。 サウンドカードをデフォルトのプレイバックデバイスとして利用するには、hw.snd.default_unit をプレイバックで使用するユニット番号に変更してください。

```
# sysctl hw.snd.default_unit= n
```

ここで、n は使用するサウンドデバイスの番号です。 この例では 4 です。 /etc/sysctl.conf に以下の行を入れると、設定の変更が常に反映されるようになります。

```
hw.snd.default_unit=4
```

8.2.4. 複数音源の利用

##: Munish Chopra.

同時に再生することのできる音源を複数実装していることは、 多くの場合望ましいことです。FreeBSD では、「仮想サウンドチャネル」を使ってカーネル内でサウンドを合成することにより、サウンドカードの再生を多重化することができます。

仮想チャネルの数を決めるのに三つの sysctl(8) 変数を設定できます。

```
# sysctl dev.pcm.0.play.vchans=4
# sysctl dev.pcm.0.rec.vchans=4
# sysctl hw.snd.maxautovchans=4
```

この例では四つの仮想チャネルを設定しています。 これは通常利用する上で十分実用的な数です。dev.pcm.0.play.vchans=4 と dev.pcm.0.rec.vchans=4 は、デバイスが取り付けられた後で設定できます。これらは pcm0 が再生や録音のために持っている仮想チャネルの数です。hw.snd.maxautovchans は、kldload(8) を用いて認識された新しいデバイスの仮想チャネル数です。pcm モジュールはハードウェアドライバとは独立して読み込むことができるので、hw.snd.maxautovchans は、オーディオデバイスが取り付けられた時に、デバイスに与えられる仮想チャネルの数を表しています。より詳細な情報については pcm(4) を参照してください。

注記

デバイスを使用しているときに仮想チャンネルの数を変更することはできません。 まず、ミュージックプレーヤやサウンドデーモンといった デバイスを使用しているすべてのプログラムを終了してください。

/dev/dsp0 を必要とするプログラムが意識しなくても、適切な pcm デバイスが自動的に設定されます。

8.2.5. ミキサチャネルの初期値を設定する

##: Josef El-Rayes.

各ミキサチャネルの初期値は pcm(4) ドライバのソースコードにハードコーディングされています。 mixer(8) および他のサードパーティ製のアプリケーションやデーモンによって、 サウンドカードのミキサレベルを変更できますが、 永続的な解決方法ではありません。そのかわり以下の例のように、適切な値を /boot/device.hints ファイルに記述することによって、ドライバレベルでミキサの初期値を設定することができます。

```
hint.pcm.0.vol="50"
```

この例では、pcm(4) が読み込まれたと同時に、ボリュームチャネルの初期値を 50 に設定します。

8.3. MP3 オーディオ

##: Chern Lee.

この節では、FreeBSD で利用できる MP3 プレイヤや、オーディオ CD トラックを吸い出す方法、および MP3 のエンコード、デコードの方法について説明します。

8.3.1. MP3 プレイヤ

XMMS は 人気のあるグラフィカルな MP3 プレイヤです。Winamp スキンや追加のプラグインに対応しています。XMMS のプレイリスト、グラフィックイコライザ等のインタフェースは直感的です。Winamp を使いなれている人は簡単に XMMS を使えるでしょう。FreeBSD では、XMMS は multimedia/xmms の port または package からインストールできます。

audio/mpg123 package もしくは port は、は代替となる コマンドライン上の MP3 プレイヤです。インストールしたら、再生する MP3 ファイルをコマンドラインから指定してください。もしシステムが、複数のオーディオデバイスを搭載しているのであれば、サウンドデバイスを同様に指定してください。

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatesHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
        version 1.18.1; written and copyright by Michael Hipp and ↺
others
        free software (LGPL) without any warranty but with best ↺
wishes

Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

他の MP3 プレイヤも Ports Collection から利用できます。

8.3.2. CD オーディオトラックの抽出

CD 全体または CD トラックを MP3 に変換する前に、CD 上のオーディオデータをハードディスク上に抽出する必要があります。これは raw CD Digital Audio (CDDA) データを WAV ファイルにコピーすることで行われます。

sysutils/cdrtools スイートからインストールされる `cdda2wav` ツールを用いて、CD からオーディオデータを抽出できます。

CD をドライブにいれて次のコマンドを `root` 権限で実行すると、CD 全体をトラックごとに個々の WAV ファイルに抽出できます。

```
# cdda2wav -D 0,1,0 -B
```

この例では、`-D 0,1,0` は SCSI デバイス `0,1,0` が抽出する CD を表します。`cdrecord -scanbus` を使って、システムの適切なデバイスパラメータを取得してください。

個々のトラックを抽出するには、次のように `-t` でトラックを指定してください。

```
# cdda2wav -D 0,1,0 -t 7
```

範囲を指定して、一番目から七番目のトラックまで抽出したい場合、次のようにします。

```
# cdda2wav -D 0,1,0 -t 1+7
```

ATAPI (IDE) CDROM ドライブから抽出するには、SCSI ユニット番号をデバイス名に置き換えて指定します。たとえば IDE ドライブから七番目のトラックを抽出するには、次のようにします。

```
# cdda2wav -D /dev/acd0 -t 7
```

または、「オーディオ CD の複製」で説明されているように、`dd` を使って ATAPI ドライブ上のオーディオトラックを展開できます。

8.3.3. MP3 のエンコードとデコード

lame は、ポピュラーな MP3 エンコーダです。audio/lame port からインストールできます。特許の問題から、package は利用できません。

次のコマンドを実行すると、抽出した WAV ファイル *audio01.wav* を使って *audio01.mp3* に変換します。

```
# lame -h -b 128 --tt "##" --ta "#######" --tl "#####" \
--ty "#" --tc "####" --tg "####" audio01.wav audio01.mp3
```

ここで指定している 128 kbits は、MP3 の標準のビットレートです。160 kbits または 192 kbits のビットレートは、さらに高音質を提供します。ビットレートが高くなるにつれて作成される MP3 ファイルは多くのディスク領域を消費します。`-h` オプションを指定すると「低速高品質」モードとなります。`--t` ではじまるオプションは ID3 タグを設定します。このタグにはたいてい曲の情報が含まれており、MP3 ファイルに格納されます。Lame のマニュアルを参照すれば、他のエンコーディングのオプションが見つかるでしょう。

MP3 からオーディオ CD を作成するには、まず非圧縮のファイル形式に変換しなければなりません。XMMS は WAV 形式へ変換できますが、mpg123 は raw Pulse-Code Modulation (PCM) オーディオデータに変換します。

mpg123 を使って `audio01.mp3` を変換するには、PCM ファイルを指定してください。

```
# mpg123 -s audio01.mp3 > audio01.pcm
```

XMMS を使って MP3 を WAV 形式に変換するには、以下の手順に従ってください。

手順8.1 XMMS を使って WAV 形式に変換する

1. XMMS を起動します。
2. 右クリックで XMMS メニューを表示します。
3. `Options` から `Preferences` を選択します。
4. Output Plugin を「Disk Writer Plugin」に変更します。
5. `Configure` を押します。
6. 非圧縮ファイルを書き出すディレクトリを入力、または選択します。
7. 普段通り XMMS へ MP3 ファイルを読み込みます。音量は 100% でイコライザの設定はオフにします。
8. `Play` を押します。XMMS は MP3 を再生しているかのように表示しますが、音声はきこえません。実際には MP3 をファイルに出力しています。
9. 終了したら、再び MP3 を聴けるように Output Plugin を以前のように元に戻すのを忘れないでください。

WAV と PCM 形式は、cdrecord で利用できます。WAV ファイルを使用する場合、それぞれのトラックの先頭に小さなノイズが入るのに気づくでしょう。これは WAV ファイルのヘッダ情報です。audio/sox port または package を使うとヘッダ情報を削除できます。

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

FreeBSD での CD 作成の詳しい情報は「光メディア (CD & DVD) の作成と使用」を参照してください。

8.4. ビデオ再生

##: Ross Lippert.

ビデオ再生のための設定をはじめる前に、ビデオカードのモデルおよびチップセットを確認する必要があります。 Xorg はさまざまなビデオカードに対応していますが、 すべてのカードがビデオ再生に性能を発揮できるとは限りません。 利用しているビデオカードの Xorg サーバが対応している拡張機能のリストを得るには、 Xorg を実行中に `xdpyinfo` を実行してください。

さまざまなプレイヤやオプションを試すのに、テストファイルとして小さな MPEG ファイルを用意しておくのはよい考えです。いくつかの DVD アプリケーションは DVD メディアを `/dev/dvd` として初期設定しているか、ハードコーディングしているので、次のように適切なデバイスにシンボリックリンクを張っておくと便利かもしれません。

```
# ln -sf /dev/cd0 /dev/dvd
```

devfs(5) の仕様により、 このように手動で作成されたリンクはシステムを再起動すると消えてしまいます。 システムの起動時にこれらのシンボリックリンクを自動的に作成するには、`/etc/devfs.conf` に下記の設定を追加してください。

```
link cd0 dvd
```

特別な機能を必要とする DVD の抽出には、DVD デバイスへの書き込み権限が必要です。

Xorg インタフェースの使う共有メモリを拡張するために、以下の sysctl(8) 変数の値を増やすことが推奨されています。

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

8.4.1. ビデオ機能の決定

Xorg においてビデオ表示性能を改善する方法はいくつかあり、正しく動作するかどうかはハードウェアに大きく依存しています。下記に説明したどの方法でも、ハードウェアが変わると品質が変わるでしょう。

よく知られたビデオインタフェースは次の通りです。

1. Xorg: 共有メモリを用いた通常の出力

2. XVideo: 特別なアクセラレータによって、 drawable オブジェクトに直接ビデオを表示する Xorg インタフェースの拡張機能です。 この拡張を使うことで廉価なコン

ピュータでも高品質の再生が可能になります。次の節では、この拡張が動作していることの確認方法について説明します。

3. SDL: Simple Directmedia Layer は、さまざまなオペレーティングシステムの間でサウンドとグラフィックスを効果的に利用したクロスプラットホームアプリケーションを開発することを目的としたレイヤです。SDL はハードウェアに対する低レベルの抽象的概念を提供し、時には Xorg インタフェースを使用するよりも効果的なことがあります。FreeBSD では、SDL は、devel/sdl20 package または port によりインストールできます。

4. DGA: Direct Graphics Access は、プログラムが Xorg サーバを介せず直接フレームバッファを変更することを可能にする Xorg の拡張機能です。低レベルのメモリマッピングが実行できることを期待しているので、この機能を使うプログラムは `root` 権限で実行されなければなりません。DGA 機能拡張は dga(1) によってテストとベンチマークができます。`dga` 実行中はキーボードを押せばいつでもディスプレイ色が変更されます。中止するには q を押します。

5. SVGAlib: 低レベルコンソールグラフィックレイヤ

8.4.1.1. XVideo

この拡張機能が動作しているかどうかを調べるには、`xvinfo` を使います。

```
% xvinfo
```

以下のような結果が得られたならば、カードは XVideo に対応しています。

```
X-Video Extension version 2.2
  screen #0
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
    supported visuals:
      depth 16, visualID 0x22
      depth 16, visualID 0x23
    number of attributes: 5
      "XV_COLORKEY" (range 0 to 16777215)
              client settable attribute
              client gettable attribute (current value is 2110)
      "XV_BRIGHTNESS" (range -128 to 127)
              client settable attribute
              client gettable attribute (current value is 0)
      "XV_CONTRAST" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_SATURATION" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_HUE" (range -180 to 180)
              client settable attribute
```

```
              client gettable attribute (current value is 0)
    maximum XvImage size: 1024 x 1024
    Number of image formats: 7
      id: 0x32595559 (YUY2)
        guid: 59555932-0000-0010-8000-00aa00389b71
        bits per pixel: 16
        number of planes: 1
        type: YUV (packed)
      id: 0x32315659 (YV12)
        guid: 59563132-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x30323449 (I420)
        guid: 49343230-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x36315652 (RV16)
        guid: 52563135-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x3e0, 0x7c00
      id: 0x35315652 (RV15)
        guid: 52563136-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x7e0, 0xf800
      id: 0x31313259 (Y211)
        guid: 59323131-0000-0010-8000-00aa00389b71
        bits per pixel: 6
        number of planes: 3
        type: YUV (packed)
      id: 0x0
        guid: 00000000-0000-0000-0000-000000000000
        bits per pixel: 0
        number of planes: 0
        type: RGB (packed)
        depth: 1
        red, green, blue masks: 0x0, 0x0, 0x0
```

リストにある形式、YUV2, YUV12 などが XVideo のすべての実装で存在するとは限りません。対応している形式が少ないために、あるプレイヤでは悪影響が出るかもしれないことにも注意してください。

出力が以下のような場合、

```
X-Video Extension version 2.2
screen #0
```

```
no adaptors present
```

カードはおそらく XVideo に対応していないのでしょう。このことはディスプレイでビデオを表示するのに、ビデオカードおよびプロセッサによっては、計算上の要求を満たすことがより困難になることを意味します。

8.4.2. ビデオを扱う ports および packages

この節では Ports Collection で利用可能な、ビデオの再生に使用できるソフトウェアについて紹介します。

8.4.2.1. MPlayer および MEncoder

MPlayer はコマンドラインのビデオプレイヤで、高速性と柔軟性をもたらすグラフィカルなインタフェースも持っています。MPlayer の他のグラフィカルなフロントエンドも Ports Collection からインストールできます。

MPlayer は multimedia/mplayer package または port からインストールできます。いくつかのコンパイル時のオプションを設定することができ、また、構築の際にさまざまなハードウェアのチェックがおこなわれます。そのため、package からインストールを行わず、port から構築することを好むユーザもいます。

port を構築する際に、メニューのオプションは、port にコンパイル時にオプションとしてどの形式に対応するかを決定するため、 見ておく必要があります。 オプションが選択されていなければ、 MPlayer はその形式のビデオ形式を表示することは出来ません。 矢印キーとスペースキーを使って必要な形式を選択してください。 選択が終わったら、Enter を押して、port の構築とインストールを続けてください。

デフォルトでは、この packege または port は、`mplayer` コマンドラインユーティリティと `gmplayer` グラフィカルユーティリティを構築します。ビデオをエンコードする必要があれば、multimedia/mencoder port をコンパイルしてください。ライセンスの制限のため、MEncoder の package は利用できません。

MPlayer を初めて起動すると、各自のホームディレクトリ内に `~/.mplayer` が作成されます。このサブディレクトリには、ユーザ固有の設定ファイルのデフォルトバージョンが含まれています。

この節では、一般的な使用法についてのみ説明します。 数多くのオプションの完全な説明については、mplayer(1) のマニュアルに記載されています。

`testfile.avi` というファイルを再生するには、以下の例のように、`-vo` とともに、ビデオインタフェースを指定してください。

```
% mplayer -vo xv testfile.avi
```

```
% mplayer -vo sdl testfile.avi
```

```
% mplayer -vo x11 testfile.avi
```

```
# mplayer -vo dga testfile.avi
```

```
# mplayer -vo 'sdl:dga' testfile.avi
```

ビデオ再生の相対的性能は多くの要因に依存し、 ハードウェアに応じて著しく変わると思われるので、これらのオプションをすべて試してみる価値はあるでしょう。

DVD を再生するには、*testfile.avi* を dvd://*N* -dvd-device *DEVICE* に置き換えてください。<N> には再生するタイトル番号を、*DEVICE* は DVD のデバイスノードを指定します。たとえば、/dev/dvd から 2 番目のタイトルを再生するには以下のようにします。

```
# mplayer -vo xv dvd://3 -dvd-device /dev/dvd
```

注記

デフォルトの DVD デバイスは、 MPlayer port の構築時に WITH_DVD_DEVICE=/path/to/desired/device を追加することでで定義できます。デフォルトでは、デバイスは /dev/cd0 です。詳細はこの port の Makefile.options をご覧ください。

停止、休止、再生などをするにはキーバインディングを使ってください。キーバインディングの一覧を見るには、mplayer -h を実行するか、もしくは、mplayer(1) を読んでください。

再生に関する追加のオプションがあります。全画面モードにする -fs -zoom オプションと、性能を向上させる -framedrop オプションです。

よく使用するオプションについては、各ユーザの .mplayer/config に以下のように追加してください。

```
vo=xv
fs=yes
zoom=yes
```

mplayer を使って、DVD タイトルを .vob に抽出できます。DVD から 2 番目のタイトルをダンプするには次のようにします。

```
# mplayer -dumpstream -dumpfile out.vob dvd://2 -dvd-
device /dev/dvd
```

出力された out.vob ファイルは MPEG 形式です。

UNIX® ビデオについて、 高レベルのノウハウを得たいと考えている方は mplayerhq.hu/DOCS をご覧ください。技術的な情報があります。このドキュメントは、バグを報告する前に、読むべきものです。

`mencoder` を使う前に、mplayerhq.hu/DOCS/HTML/en/mencoder.html を読んでオプションに慣れておくのはよい考えです。品質向上、低ビットレート、形式変換をする方法が無数にあります。 これらの要素の調節具合で、性能が良かったり悪かったりするなど、結果に違いが出るかもしれません。コマンドラインオプションを不適切に組合せると、`mplayer` でさえ再生できない出力ファイルを作成してしまいます。

はじめは単純なファイルのコピーです。

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

したがって、単にファイルを抽出したいときには、`mplayer` に `-dumpfile` をつけます。

`input.avi` を音声に MPEG3 エンコードを使用して MPEG4 コーデックに変換するには、まず最初に audio/lame port をインストールしてください。ライセンスの制限により、package は利用できません。インストールしたら、以下のように入力してください。

```
% mencoder input.avi -oac mp3lame -lameopts br=192 \
  -ovc lavc -lavcopts vcodec=mpeg4:vhq -o  output.avi
```

これは `mplayer` や `xine` といったアプリケーションで再生可能な出力ファイルを作成します。

DVD タイトルを直接再エンコードするためには、上記のコマンドラインの *`input.avi`* を `dvd://1 -dvd-device /dev/dvd` に置き換えて、`root` 権限で実行します。 期待する結果を得るには何度か繰り返すことになるので、 かわりにタイトルをファイルにダンプして、ファイルに対して作業することをおすすめします。

8.4.2.2. xine ビデオプレイヤ

xine は、再利用可能な基本ライブラリと、プラグインで拡張できる実行可能なモジュールを提供するビデオプレイヤです。multimedia/xine package または port からインストールできます。

実用上、xine を使用するには高速なビデオカードとともに高速な CPU があるか、またはビデオカードが XVideo 拡張に対応している必要があります。XVideo インタフェースとともに xine ビデオプレイヤを使うのが最良です。

デフォルトでは、xine プレイヤは GUI 付きで起動するでしょう。メニューを使用して特定のファイルを開くことができます。

xine は、再生するファイル名を指定することで、コマンドラインから実行することもできます。

```
% xine -g -p mymovie.avi
```

xine-project.org/faq には、より多くの情報やトラブルシューティングがあります。

8.4.2.3. Transcode ユーティリティ

Transcode は、ビデオおよびオーディオファイルを再エンコードするためのツール一式です。Transcode を使えば、stdin/stdout ストリームインタフェースとともにコマンドラインツールを用いることで、ビデオファイルの統合や、壊れたファイルの修復ができます。

FreeBSD では、Transcode は、multimedia/transcode package もしくは port からインストールできます。多くのユーザは port からコンパイルすることを好みます。port では、 コンパイルで有効にするサポートやコーデックを指定するコンパイルオプションのメニューを利用できるためです。オプションを選択しないと、Transcode は、その形式をエンコード出来ないでしょう。矢印キーとスペースバーを使って、必要とするフォーマットを選択してください。選択が終わったら、Enter を押して、port のコンパイルとインストールを続けてください。

この例では、DivX ファイルを PAL MPEG-1 (PAL VCD) に変換する使用例を示します。

```
% transcode -i
input.avi -V --export_prof vcd-pal -o output_vcd
% mplex -f 1 -o output_vcd.mpg output_vcd.m1v ↵
output_vcd.mpa
```

作成された MPEG ファイル、*output_vcd.mpg* は、MPlayer を使って再生できます。また、multimedia/vcdimager および sysutils/cdrdao といったユーティリティを使って、ファイルを CD メディアに書き込むことでビデオ CD も作成できます。

`transcode` のマニュアルページに加え、transcoding.org/cgi-bin/transcode から、更なる情報や使用例を得てください。

8.5. TV カードの設定

##: Josef El-Rayes.
##: Marc Fonvieille.

TV カードを使用することで、TV 放送をコンピュータで見ることができます。これらの多くのカードは RCA または S-video 入力端子を備えており、FM ラジオチューナを装備したカードもあります。

FreeBSD は、Brooktree Bt848/849/878/879 をビデオキャプチャチップに採用した PCI TV カードに bktr(4) ドライバで対応しています。このドライバは、ほとんどの Pinnacle PCTV ビデオカードに対応しています。TV カードを購入する前に、対応しているチューナの一覧について、bktr(4) を参照してください。

8.5.1. ドライバを読み込む

カードを使用するには、bktr(4) ドライバを読み込む必要があります。起動時に自動的に読み込むためには、`/boot/loader.conf` に以下の行を追加してください。

```
bktr_load="YES"
```

あるいは、カスタムカーネルに TV ビデオカードへのサポートを静的に組み込むこともできます。この場合には、次の行をカーネルコンフィギュレーションファイルに追加してください。

```
device  bktr
device iicbus
device iicbb
device smbus
```

カードコンポーネントは I2C バス経由で連結されているため、bktr(4) ドライバに加えてこれらのデバイスが必要になります。編集したら新しいカーネルを構築し、インストールします。

チューナが適切に検出されたかどうかを確認するため、システムを再起動してください。起動時のメッセージに TV カードが以下のように認識されるでしょう。

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device ↺
10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

これらのメッセージはハードウェアに応じて異なります。 必要であれば、sysctl(8) や、カーネルコンフィギュレーションファイルオプションで、 検知されたいくつかのパラメータを変更できます。たとえば、チューナを Philips SECAM チューナとして検知されるようにするには、カーネルコンフィギュレーションファイルに以下の行を追加します。

```
options OVERRIDE_TUNER=6
```

または、直接 sysctl(8) を使用して変更します。

```
# sysctl hw.bt848.tuner=6
```

利用可能な sysctl(8) パラメータおよびカーネルオプションについては bktr(4) を参照してください。

8.5.2. 便利なアプリケーション

TV カードを使用するためには、 以下のアプリケーションの一つをインストールする必要があります。

- multimedia/fxtv はウィンドウ内に TV 映像を映します。画像/音声/ビデオを取り込むこともできます。
- multimedia/xawtv も同様の機能を持った TV アプリケーションです。
- audio/xmradio は TV カードに搭載された FM ラジオチューナを使用するためのアプリケーションです。

他にも多くのアプリケーションが FreeBSD の Ports Collection に収録されています。

8.5.3. トラブルシューティング

TV カードに関する問題が起きたときには、bktr(4) が本当にビデオキャプチャチップおよびチューナに対応しているか、 オプションが正しく設定されているかどうかをまず確認してください。TV カードに関するサポートや質問に関しては、freebsd-multimedia メーリングリストを参照してください。

8.6. MythTV

MythTV は、広く使われているオープンソースの Personal Video Recorder (PVR) アプリケーションです。 この節では、FreeBSD に MythTV をインストールし、 設定する方法について説明します。 MythTV の使用法に関するより詳細な情報については、mythtv.org/wiki をご覧ください。

MythTV は、フロントエンドおよびバックエンドを必要とします。これらは、同じシステム上でも、異なるコンピュータ上でも動かすことが可能です。

フロントエンドについては、multimedia/mythtv-frontend package または port から FreeBSD にインストールできます。 6章X Window System で説明されているように、 Xorg をインストールして設定する必要もあります。 このシステムは X-Video Motion Compensation (XvMC) に対応し、 オプションとして、Linux Infrared Remote Control (LIRC)-互換のリモートに対応したビデオカードを持っていることが理想的です。

FreeBSD にバックエンドとフロントエンドの両方をインストールするには、multimedia/mythtv package または port を使ってください。MySQL™ データベースサーバも必要となりますが、自動的に依存でインストールされます。オプションで、チューナカードと録音したデータを保存するためのストレージが必要です。

8.6.1. ハードウェア

MythTV は、 エンコーダやチューナなどのビデオ入力デバイスへのアクセスに Video for Linux (V4L) を用います。 FreeBSD では、USB DVB-S/C/T カードにおいて最もよく動作します。なぜならば、このカードは、V4L ユーザランドアプリケーションを提供する multimedia/webcamd package または port により良くサポートされてい

るためです。 webcamd により対応している Digital Video Broadcasting (DVB) カードは、MythTV で動作するはずです。 動作することが知られているカードの一覧が wiki.freebsd.org/WebcamCompat にあります。 Hauppauge カードのドライバもまた、multimedia/pvr250 および multimedia/pvrxxx port として利用可能ですが、標準的ではないドライバのインタフェースを提供しており、0.23 より後の MythTV では動作しません。ライセンスの制限により、package は利用できません。そのため、これらの ports はコンパイルをしなければなりません。

wiki.freebsd.org/HTPC ページは、DVB ドライバのすべての一覧を提供しています。

8.6.2. MythTV バックエンドの設定

port を使って MythTV をインストールしてください。

```
# cd /usr/ports/multimedia/mythtv
# make install
```

インストールが終わったら、MythTV データベースを設定してください。

```
# mysql -uroot -p < /usr/local/share/mythtv/database/
mc.sql
```

その後、バックエンドを設定してください。

```
# mythtv-setup
```

最後にバックエンドを起動してください。

```
# echo 'mythbackend_enable="YES"' >> /etc/rc.conf
# service mythbackend start
```

8.7. 画像スキャナ

##: Marc Fonvieille.

FreeBSD では、画像スキャナに対するアクセスは SANE (Scanner Access Now Easy) によって実現されており、FreeBSD の Ports Collection で提供されています。SANE はスキャナのハードウェアにアクセスするために FreeBSD デバイスドライバを使用します。

FreeBSD は SCSI 接続および USB 接続のスキャナのどちらにも対応しています。 スキャナのインタフェースに依存して、異なるドライバが必要となります。 設定を始める前に、 SANE がスキャナに対応していることを確認してください。 対応しているスキャナに関してのより詳細な情報については、http://www.sane-project.org/sane-supported-devices.html をご覧ください。

この節では、FreeBSD がどのようにしてスキャナを認識するかについて説明します。その後、FreeBSD システム上で SANE を設定して使用する方法の概要について説明します。

8.7.1. スキャナの確認

GENERIC カーネルには USB スキャナに対応するためのデバイスドライバが搭載されています。カスタムカーネルを使用する際には、以下の行がカーネルコンフィグレーションファイルにあることを確認してください。

```
device usb
device uhci
device ohci
device ehci
```

USB スキャナが認識されたかを確認するには、スキャナを接続して、dmesg を利用し、システムメッセージバッファで、スキャナが認識されているかどうかを確認してください。認識されていたら、以下のようなメッセージが表示されます。

```
ugen0.2: <EPSON> at usbus0
```

この例では、EPSON Perfection® 1650 USB スキャナが /dev/ugen0.2 上で認識されています。

スキャナのインタフェースが SCSI であれば、どの SCSI コントローラボードを使用するかを知ることが重要です。使用する SCSI チップセットによって、カスタムカーネルコンフィグレーションファイルを調整する必要があります。GENERIC カーネルは、一般に使用される SCSI コントローラのほとんどに対応しています。/usr/src/sys/conf/NOTES ファイルを読んで、適切な行をカーネルコンフィグレーションファイルに追加してください。また、SCSI アダプタドライバに加えて、以下の行をカスタムカーネルコンフィグレーションファイルに記述する必要があります。

```
device scbus
device pass
```

デバイスがメッセージバッファに出力されていることを確認してください。

```
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers
```

システムを起動する際にスキャナの電源を入れてなければ、camcontrol を使用して SCSI バスをスキャンし、以下のように手動でデバイスを検出させることもできます。

```
# camcontrol rescan all
Re-scan of bus 0 was successful
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

すると、スキャナは SCSI デバイスの一覧に現れるでしょう。

```
# camcontrol devlist
<IBM DDRS-34560 S97B>              at scbus0 target 5 lun 0 ↺
(pass0,da0)
<IBM DDRS-34560 S97B>              at scbus0 target 6 lun 0 ↺
(pass1,da1)
<AGFA SNAPSCAN 600 1.10>           at scbus1 target 2 lun 0 (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00>     at scbus2 target 0 lun 0 ↺
(pass2,cd0)
```

FreeBSD における SCSI デバイスについての詳細は、scsi(4) および camcontrol(8) をご覧ください。

8.7.2. SANE の設定

SANE システムは、二つの部分、すなわちバックエンド (graphics/sane-backends) とフロントエンド (graphics/sane-frontends もしくは、graphics/xsane) に分割されています。 バックエンドはスキャナに対するアクセスを提供します。 どのバックエンドが画像スキャナに対応しているかについては、http://www.sane-project.org/sane-supported-devices.html を参照してください。 フロントエンドはグラフィカルなスキャニングインタフェースを提供します。graphics/sane-frontends は、xscanimage をインストールし、一方、graphics/xsane は、xsane をインストールします。

graphics/sane-backends port または package をインストールしたら、`sane-find-scanner` コマンドを使用して、SANE システムで検出されているスキャナを確認してください。

```
# sane-find-scanner -q
found SCSI scanner "AGFA SNAPSCAN 600 1.10" at /dev/pass3
```

この出力から、スキャナインタフェースの種類と システムに接続されているスキャナが使用するデバイスノードがわかります。 ベンダ名や製品のモデル名は表示されないかも知れません。

注記

いくつかの USB スキャナではファームウェアを読み込む必要がある場合があります。 詳細については、sane-find-scanner(1) および sane(7) を参照してください。

次に、スキャナがフロントエンドで認識されるか調べてください。SANE のバックエンドには `scanimage` が付属します。このコマンドを使用すると、デバイスの一覧を表示したり

画像を取得することができます。スキャナデバイスの一覧を表示するには、`-L` オプションを使ってください。以下の最初の例は、SCSI スキャナ用のもので、次の例は、USB スキャナ用のものです。

```
# scanimage -L
device `snapscan:/dev/pass3' is a AGFA SNAPSCAN 600 flatbed scanner
# scanimage -L
device 'epson2:libusb:/dev/usb:/dev/ugen0.2' is a Epson GT-8200 ↺
flatbed scanner
```

2 番目の出力の中で、`'epson2:libusb:/dev/usb:/dev/ugen0.2'` がスキャナが使用するバックエンド名 (`epson2`) および `/dev/ugen0.2` は、デバイスノードです。

`scanimage` がスキャナの認識に失敗した場合には、以下のようなメッセージが表示されます。

```
# scanimage -L

No scanners were identified. If you were expecting something ↺
different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the ↺
documentation
which came with this software (README, FAQ, manpages).
```

このような場合には、`/usr/local/etc/sane.d/` にあるバックエンドの設定ファイルを編集して、使用するスキャナデバイスを設定してください。例えば、認識されなかったスキャナのモデルが、EPSON Perfection® 1650 で、`epson2` バックエンドを使っているのであれば、`/usr/local/etc/sane.d/epson2.conf` を編集してください。編集作業を行う際には、使用するインタフェースとデバイスノードを指定する行を追加します。この例では、以下の行を追加します。

```
usb /dev/ugen0.2
```

編集を保存し、適切なバックエンド名とデバイスノードでスキャナが認識されたかどうかを確認してください。

```
# scanimage -L
device 'epson2:libusb:/dev/usb:/dev/ugen0.2' is a Epson GT-8200 ↺
flatbed scanner
```

`scanimage -L` を実行してスキャナが認識されたことがわかれば、設定は終了です。スキャナを使用する準備ができました。

`scanimage` を使用してコマンドラインから画像を取得することができますが、GUI を使用して画像を取得できるとより望ましいでしょう。graphics/sane-frontends package

および port は、シンプルですが、効率的なグラフィカルインタフェース xscanimage をインストールします。

一方、graphics/xsane package または port からインストールされる xsane は、広く使われているもう一つのグラフィカルなスキャニングフロントエンドです。Xsane には、さまざまなスキャニングモード、色補正、バッチスキャンなど先進的な機能があります。これらのアプリケーションの両方とも GIMP のプラグインとして使用することができます。

8.7.3. スキャナの許可属性

スキャナにアクセスするには、ユーザはスキャナが使用するデバイスノードへの読み込み権限と書き込み権限が必要です。 今回の例では、USB スキャナは `/dev/ugen0.2` デバイスノードを使用しています。 このデバイスノードは、 `/dev/usb/0.2.0` へのシンボリックリンクです シンボリックリンクとデバイスノードは、 それぞれ `wheel` および `operator` グループが所有しています。ユーザをこれらのグループに加えると、スキャナを使用できるようになりますが、ユーザを `wheel` に追加することは、セキュリティの観点からお勧めできません。良い方法は、スキャナデバイスにアクセスできるグループを作成することです。

この例では、*usb* という名前のグループを作成します。

```
# pw groupadd usb
```

その後、シンボリックリンク `/dev/ugen0.2` および、`/dev/usb/0.2.0` デバイスノードに対して、 `usb` グループが利用できるように書き込みの許可属性 `0660` または `0664` を設定してください。`/etc/devfs.rules` に次の行を追加すれば設定できます。

```
[system=5]
add path ugen0.2 mode 0660 group usb
add path usb/0.2.0 mode 0666 group usb
```

最後に、スキャナを利用するユーザを *usb* グループに追加してスキャナを利用できるようにしてください。

```
# pw groupmod usb -m joe
```

詳細については、pw(8) をご覧ください。

第9章 FreeBSD カーネルのコンフィグレーション

9.1. この章では

カーネルは FreeBSD オペレーティングシステムの中核をなすものです。 カーネルは、メモリ管理、セキュリティ制御の強制、ネットワーク、ディスクアクセスなどを担っています。 FreeBSD の大部分は動的に構成することができるようになっていますが、まだ、時にはカスタムカーネルを設定してコンパイルする必要があります。

この章では、以下のことを扱っています。

- いつカスタムカーネルの構築が必要になるか。
- ハードウェア一覧の作成方法。
- カーネルコンフィグレーションファイルのカスタマイズの方法。
- カーネルコンフィグレーションファイルから新しいカーネルを構築する方法。
- 新しいカーネルのインストール方法。
- うまく行かないときの問題解決法。

この章で表示されているすべてのコマンドは、`root` 権限で実行する必要があります。

9.2. なぜカスタムカーネルを作るか?

伝統的に、FreeBSD はモノリシック (monolithic) カーネルを使っていました。このカーネルは、単一の巨大なプログラムで、扱えるデバイスは固定されていて、カーネルの振る舞いを変えたければ構築してコンピュータを再起動し、新しいカーネルを動かさなれければなりませんでした。

今日では、FreeBSD カーネルのかなりの機能はモジュールに含まれるようになり、 必要に応じて動的にカーネルに組み込んだり外したりできるようになりました。 この移行により、 動作しているカーネルが新しいハードウェアに迅速に対応したり、 カーネルに新たな機能を取り入れられるようになります。 このようなカーネルは、モジュラ (modular) カーネルと呼ばれます。

しかしながら、 いまだにいくらかは静的にカーネルを構成する必要があります。 機能がカーネルとあまりに密接に結びついているため、 動的に組み込むことができない場合が

あるためです。 環境によっては、セキュリティの観点から、 カーネルモジュールを読み込んだり外すことができず、 必要となる機能を静的にカーネルにコンパイルしなければならない場合もあります。

システムに合わせたカーネルを構築することは、多くの場合、 高度な知識を持つ BSD ユーザが避けて通ることのできない通過儀礼です。 この作業は多くの時間を必要としますが、FreeBSD システムに利益をもたらします。 広範囲のハードウェアをサポートしなければならない GENERIC カーネルとは異なり、カスタムカーネルは、 使用しているコンピュータのハードウェアのみをサポートするように、必要のない機能を省くことができます。これは、次にあげるような利益をもたらします。

- 素早く起動します。カーネルはシステム上にあるハードウェアしか検出しないので、システムの起動にかかる時間を短くできます。

- メモリの消費量を減らすことができます。システムに合わせたカーネルは、使用しない機能やデバイスドライバを含まないので、大抵 GENERIC カーネルより少ないメモリしか消費しません。カーネルコードは常に物理メモリ上に存在し、アプリケーションはその容量分のメモリを使用できないので、 これは重要なことです。 したがって、メモリが少ないシステムでは、カーネルの再構築は重要です。

- 追加のハードウェアをサポートします。 カスタムカーネルは、GENERIC カーネルに存在しないデバイスのサポートを追加することができます。

カスタムカーネルを構築する前に、再構築する理由を考えてください。 ある特定のハードウェアに対応する必要がある場合に、 そのハードウェアに対応するためのモジュールがすでに用意されていることがあります。

カーネルモジュールは /boot/kernel にあります。モジュールによっては kldload(8) により、すでに実行中のカーネルに動的に読み込まれています。ほとんどのカーネルドライバには、 読み込み可能なモジュールやマニュアルページが用意されています。 たとえば、ath(4) ワイヤレスイーサネットドライバのマニュアルページには以下のような記述があります。

```
Alternatively, to load the driver as a module at boot time, place the
following line in loader.conf(5):

    if_ath_load="YES"
```

/boot/loader.conf に if_ath_load="YES" を追加すると、 起動時にモジュールが読み込まれるようになります。

対応するモジュールが /boot/kernel に存在しないこともあります。特定のサブシステムでは、ほとんど多くの場合存在しません。

9.3. システムのハードウェアについて知る

カーネルコンフィグレーションファイルの編集を始める前に、 コンピュータのハードウェア一覧を作成すると良いでしょう。 デュアルブートシステムでは、 現在インストールされている別のオペレーティングシステムの設定を調べることで、一覧を作成できます。たとえば、Microsoft® の デバイスマネージャ は、インストールされているデバイスに関する情報を持っています。

注記

Microsoft® Windows® のバージョンによっては、システム アイコンを使って、デバイスマネージャ にアクセスできます。

インストールされているオペレーティングシステムが FreeBSD だけであれば、dmesg(8) を使い、 起動時に検出されたハードウェアの一覧を調べてください。FreeBSD のほとんどのデバイスドライバにはマニュアルページが用意され、対応しているハードウェアの一覧を提供しています。たとえば、以下の行は、psm(4) ドライバがマウスを検出したことを示しています。

```
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: [ITHREAD]
psm0: model Generic PS/2 mouse, device ID 0
```

このハードウェアはシステムに存在するので、 カスタムカーネルコンフィグレーションファイルからこのドライバを外さないでください。

`dmesg` が起動時の検出結果を表示しない場合には、 かわりに `/var/run/dmesg.boot` で出力を確認してください。

ハードウェアを見つけるためのもうひとつのツールは、より冗長な出力を行う pciconf(8) です。たとえば、以下のようになります。

```
% pciconf -lv
ath0@pci0:3:0:0:        class=0x020000 card=0x058a1014 ↺
chip=0x1014168c rev=0x01 hdr=0x00
    vendor     = 'Atheros Communications Inc.'
    device     = 'AR5212 Atheros AR5212 802.11abg wireless'
    class      = network
    subclass   = ethernet
```

この出力は、`ath` ドライバがワイヤレスイーサネットデバイスにあることを示しています。

man(1) を `-k` フラグで実行すると、有用な情報を得ることができます。たとえば、ある特定の単語を含むマニュアルページの一覧を表示するには、以下のようになります。

```
# man -k Atheros
```

```
ath(4)                  - Atheros IEEE 802.11 wireless network ↺
driver
ath_hal(4)              - Atheros Hardware Access Layer (HAL)
```

ハードウェアの一覧を作成したら、 この一覧を利用して、 カスタムカーネルのコンフィグレーションファイルを編集している時に、 インストールされているハードウェアのドライバが削除されていないことを確認してください。

9.4. コンフィグレーションファイル

カスタムカーネルのコンフィグレーションファイルを作成し、 カスタムカーネルを構築するには、FreeBSD の全ソースツリーがまずインストールされている必要があります。

もし `/usr/src/` が存在していなかったり、空であれば、カーネルのソースはインストールされていません。「Subversion を使う」で説明した Subversion を使ってソースをインストールしてください。

ソースをインストールしたら、 `/usr/src/sys` を確認して下さい。 このディレクトリには、いくつものサブディレクトリがあります。その中には、サポートされている各アーキテクチャ `amd64`, `i386`, `ia64`, `pc98`, `powerpc` および `sparc64` のサブディレクトリがあります。 各アーキテクチャのディレクトリ内部にあるファイルはすべてそのアーキテクチャでのみ使用されます。残りのコードは、アーキテクチャに依存しない、すべてのプラットフォームで共有されるコードです。サポートされている各アーキテクチャには、`conf` サブディレクトリがあり、そのアーキテクチャ用の `GENERIC` カーネルコンフィグレーションファイルが用意されています。

この `GENERIC` は編集しないでください。 かわりに、このファイルを別名でコピーし、コピーを編集してください。慣習として、この名前はすべて大文字でつづられます。もし、いくつかの異なるハードウェアの FreeBSD マシンを扱うなら、この名前にホスト名を含めるとよいでしょう。ここでは、例として `MYKERNEL` という名前の `amd64` アーキテクチャ用の `GENERIC` コンフィグレーションファイルのコピーを作成します。

```
# cd /usr/src/sys/ amd64 /conf
# cp GENERIC MYKERNEL
```

これで、*`MYKERNEL`* を ASCII テキストエディタで編集できます。初心者に対してより簡単なエディタである ee も FreeBSD とともにインストールされていますが、デフォルトのエディタは vi です。

コンフィグレーションファイルのフォーマットはシンプルです。各行はデバイスやサブシステム、引数、または簡単な説明を含んでいます。 `#` に続くテキストはすべてコメントとして扱われ、無視されます。カーネルからデバイスもしくはサブシステムのサポートを外すに

は、対応する行の最初に # を入れてください。理解していない行に対しては、# を追加したり削除しないでください。

警告

デバイスやオプションのサポートを外すことは簡単で、 その結果、カーネルを壊すことがあります。 たとえば ata(4) ドライバをカーネルコンフィグレーションファイルから除くと、 ATA ディスクドライバを用いているシステムは起動しません。 確信が持てないものについては、カーネルにサポートを残したままにしてください。

このファイルで与えられる説明の他に、そのアーキテクチャの GENERIC と同じディレクトリにある NOTES にも説明があります。アーキテクチャに依存しないオプションについては、/usr/src/sys/conf/NOTES をご覧ください。

ヒント

カーネルコンフィグレーションファイルの編集を終えたら、 ファイルのバックアップを /usr/src 以外の場所に保存してください。

または、カーネルコンフィグレーションファイルは他の場所において、シンボリックリンクを張る方法もあります。

```
# cd /usr/src/sys/amd64/conf
# mkdir /root/kernels
# cp GENERIC /root/kernels/MYKERNEL
# ln -s /root/kernels/MYKERNEL
```

コンフィグレーションファイルでは include ディレクティブを利用できます。 コンフィグレーションファイルに他のファイルを取り込むことができるので、すでに存在するファイルに対する小さな変更の管理が簡単にできます。 オプションやドライバの追加が少しだけの場合には、以下の例のように GENERIC からの差分による管理が可能になります。

```
include GENERIC
ident MYKERNEL

options         IPFIREWALL
options         DUMMYNET
options         IPFIREWALL_DEFAULT_TO_ACCEPT
options         IPDIVERT
```

この方法では、ローカルのコンフィグレーションファイルには、ローカルにある GENERIC カーネルとの差分が記述されています。 アップグレードが行われると、 GENERIC に追加された新しい機能は、 (nooptions や nodevice によって外されない限り) ローカルのカーネルにも反映されます。コンフィグレーションの構成要素に関する包括的な一覧と説明は config(5) にあります。

注記

利用可能なすべてのオプションを含むファイルを構築するには、以下のコマンドを root 権限で実行してください。

```
# cd /usr/src/sys/ arch/conf && make LINT
```

9.5. カスタムカーネルの構築とインストール

カスタムコンフィグレーションファイルを編集して保存したら、 カーネルのソースコードを以下の手順でコンパイルしてください。

手順9.1 カーネルの構築

1. 以下のディレクトリに移動してください。

   ```
   # cd /usr/src
   ```

2. カスタムコンフィグレーションファイルの名前を指定して新しいカーネルをコンパイルします。

   ```
   # make buildkernel KERNCONF= MYKERNEL
   ```

3. 指定したカーネルコンフィグレーションファイルでコンパイルされた新しいカーネルをインストールします。 以下のコマンドは、新しいカーネルを /boot/kernel/kernel に、今までのカーネルを /boot/kernel.old/kernel という名前で保存します。

   ```
   # make installkernel KERNCONF= MYKERNEL
   ```

4. 新しいカーネルを使うために、システムをシャットダウンして再起動してください。うまく行かない場合は、カーネルが起動しない を参照してください。

デフォルトでは、カスタムカーネルを構築すると、 すべてのカーネルモジュールが再構築されます。カーネルのアップデートをより早く行いたい、または、 カスタムモジュールのみ

を構築したいといった場合は、 カーネルの構築を開始する前に、以下のように `/etc/make.conf` を編集してください。

例として、以下の変数は、デフォルトのすべてのモジュールを構築する設定を変更し、構築するモジュール一覧を指定します。

```
MODULES_OVERRIDE = linux acpi
```

また、以下の変数は、構築を行わないモジュールを指定します。

```
WITHOUT_MODULES = linux acpi sound
```

他の変数については、make.conf(5) を参照してください。

9.6. 問題が起きた場合には

カスタムカーネルを作る際に起こりうるトラブルは、次の 4 種類に分けられます。

`config` コマンドの失敗
: `config` で失敗した時には、 トラブルの起きた行番号が出力されます。 たとえば、次のように出力された場合には、 17 行目が正しく入力されているかどうか、`GENERIC` や `NOTES` と比較して修正してください。

```
config: line 17: syntax error
```

`make` コマンドの失敗
: `make` が失敗した場合には、通常、カーネルコンフィグレーションファイルにおいて、`config` がとらえられなかったような間違いをしています。コンフィグレーションファイルを見直してください。 それでも問題を解決することができなければ、FreeBSD general questions メーリングリスト へカーネルコンフィグレーションファイルを添付して送ってください。

カーネルが起動しない
: 新しいカーネルが起動しなかったり、 デバイスの認識をしない場合でもあわてないでください! さいわい、FreeBSD には利用できないカーネルから復帰する洗練されたメカニズムがあります。 FreeBSD のブートローダで起動したいカーネルを選択してください。 システムの起動メニューが表示されている時に、「Escape to a loader prompt」 オプションを選択するとアクセスできます。 プロンプトで `boot kernel.old` か他の正常に起動するカーネルを入力してください。

  問題のないカーネルで起動した後、コンフィグレー ションファイルを調べ、再び構築を試みてください。`/var/log/messages` にはすべての成功した起動時のカーネルメッセージの記録があり、 これは問題を解決するための助けになる情報の一つでしょう。また、dmesg(8) は現在の起動時のカーネルメッセージを出力します。

注記

カーネルの構築中にトラブルが起きた時には、次回の構築で消されないように、GENERIC のコピーや他の正常に起動するカーネルを別の名前で保存するようにしてください。
kernel.old は新しいカーネルをインストールする時に、その一つ前にインストールした、うまく動かないかもしれないカーネルで上書きされてしまうため、起動するカーネルを保存しておくことは重要です。できる限り早く以下のようにして、正しく起動するカーネルを含むディレクトリ名に変更してください。

```
# mv /boot/kernel /boot/kernel.bad
# mv /boot/kernel.good /boot/kernel
```

カーネルは動きますが ps(1) は動きません!
: システムユーティリティの構築されたバージョンと異るバージョンのカーネルをインストールした場合、たとえば -CURRENT のソースから構築したカーネルを -RELEASE システム上にインストールするような場合には、ps(1) や vmstat(8) のような多くのシステムステータスコマンドは動かなくなります。修正するには、カーネルと同じバージョンのソースツリーで world を再構築し、インストール してください。カーネルとそれ以外で異なるバージョンを組み合わせてオペレーティングシステムを使用することは推奨されていません。

第10章 プリンタの利用

寄稿: Sean Kelly.
改訂:Jim Mock.

10.1. この章では

FreeBSD　は古いインパクトプリンタから最新のレーザープリンタまで幅広いプリンタが利用でき、実行しているアプリケーションから高品質な印刷出力が行えます。

FreeBSD はネットワーク上のプリンタサーバとして動作するように設定することもできます。この機能は、他の FreeBSD コンピュータや、Windows® や Mac OS® ホストから印刷ジョブを受け取ることができます。FreeBSD は印刷ジョブを 1 つずつ処理することを保証します。また、どのユーザやマシンが最も多く印刷しているかの統計を取り、どの印刷物が誰の物か表示する「バナー」ページの作成などを行うことができます。

この章を読めば以下のことがわかります。

- FreeBSD プリントスプーラの設定方法。
- 入力ドキュメントをプリンタが扱える印刷フォーマットへ変換するなどといった、特別な印刷ジョブを別に取り扱うための印刷フィルタのインストール方法。
- 印刷物へのヘッダやバナーの適用方法。
- 他のコンピュータに接続されたプリンタで印刷する方法。
- ネットワークに直接接続されたプリンタで印刷する方法。
- 印刷ジョブの上限サイズや特定のユーザからの印刷拒否といった、プリンタの制限の制御方法。
- 印刷の統計とプリンタの使用状況の取得方法。
- 印刷問題のトラブルシューティング方法。

この章を読み始める前に以下を済ませておいてください。

- 新しいカーネルの設定とインストール方法について理解すること (9章FreeBSD カーネルのコンフィグレーション)。

10.2. はじめに

FreeBSD でプリンタを使うために、それらを LPD スプーリングシステム、または単に LPD としても知られる Berkeley ラインプリンタスプーリングシステムで動作するように

設定できます。これは FreeBSD での標準的なプリンタ制御システムです。この章では、LPD を紹介し、その設定方法について説明します。

あなたがすでに LPD やその他のプリンタスプーリングシステムに詳しいのなら、基本的な設定 まで読み飛ばしてもかまいません。

LPD はホストのプリンタに関するあらゆることを制御します。ここで言う制御としては、次のことがあげられます。

- ホストに接続されたプリンタ、あるいはネットワーク上の他ホストに接続されたプリンタに対するアクセス制御を行ないます。
- ファイルをプリントする要求に対して許可を与えます。この要求は特にジョブと呼ばれています。
- 各々のプリンタのキューを管理することにより、 複数のユーザがあるプリンタに対して同時にアクセスすることを防ぎます。
- ヘッダページ (バナーまたは バーストページとしても知られています) をプリントすることができます。これにより、プリントアウトの山の中から自分がプリントしたジョブを見つけやすくなります。
- シリアルポートに接続したプリンタ用に通信パラメータを管理します。
- ネットワーク経由で他のホスト上の LPD スプーラにジョブを送ることができます。
- 様々なプリンタ言語やプリンタの能力に応じてジョブの形式を整えるため、 特別なフィルタを起動することができます。
- プリンタの使用に対して課金を行なうことができます。

設定ファイル (`/etc/printcap`) を通して、専用のフィルタプログラムを用いることにより、多種多様なプリンタ機器に対して、上述の機能の全部または一部を LPD システムに行なわせることができます。

10.2.1. どうしてスプーラを使うべきなのか

あなたのシステムを利用するのがあなた一人だけだとしても、 スプーラは有用ですし、使用すべきです。その理由は以下のとおりです。

- LPD はジョブをバックグラウンドで処理します。データがプリンタに送信されるまで待つ必要がなくなります。
- LPD ではジョブをフィルタを通してプリントすることが簡単にできます。これにより、印刷物のヘッダに時刻や日付を入れたり、特別なファイル形式 (TeX の DVI ファイルなど) をプリンタが処理できる形式に変更することができ、これらの作業を手動で行なう必要がなくなります。

- プリント処理を行なうフリー、 または商用のプログラムのほとんどは、 システムのスプーラとやりとりするように作られています。スプーリングシステムをセットアップすることで、今後加えるかもしれない、あるいは、すでに持っている別のソフトウェアをより簡単にサポートすることができるでしょう。

10.3. 基本的な設定

LPD スプーリングシステムを用いてプリンタを使用するためには、プリンタ機器と LPD 用ソフトウェアの両方を準備する必要があります。本文書では次の二段階のレベルに分けて説明をします。

- プリンタを接続する方法、プリンタにどのように通信するかを LPD に指示する方法や、プレインテキストをプリンタで印字する方法については、 プリンタの簡単な設定をご覧ください。

- 様々な形式のファイルを印字する方法、ヘッダページを印字する方法、ネットワーク経由でプリンタに印字する方法、プリンタを制御する方法、プリンタの使用に対する課金を行なう方法についてはプリンタ設定上級編をご覧ください。

10.3.1. プリンタ設定導入編

この節では、プリンタ機器やプリンタを使用するための LPD 用ソフトウェアを設定する方法について述べます。この節の概要は次のとおりです。

- プリンタ機器の設定では、 プリンタをコンピュータに接続するためのヒントがいくつか書かれています。

- ソフトウェアの設定では、LPD のスプーラ設定ファイル (`/etc/printcap`) の設定方法について書かれています。

データをプリンタに送るのにコンピュータのローカルインタフェースではなく、ネットワークプロトコルを使用する場合は、 ネットワークにおけるデータストリームインタフェースを持つプリンタをご覧ください。

この節のタイトルは 「プリンタ設定導入編」 ですが、 実際の設定はかなり複雑です。プリンタをコンピュータに接続し、 LPD スプーラを起動させることは一番困難な作業です。 ヘッダページを出力させたり課金したりするオプションの設定は、 一度プリンタがうまく動くようになればとても簡単です。

10.3.1.1. プリンタ機器の設定

この節では、プリンタに PC を接続するための様々な方法について説明しています。 ここでは、ポートやケーブルの種類、FreeBSD がプリンタとの通信に必要なカーネルコンフィグレーションについても言及しています。

もしプリンタが既に接続されていて、 他のオペレーティングシステム上でプリンタからの印字に成功している場合は、ソフトウェアの設定まで読み飛ばすことが多分できるでしょう。

10.3.1.1.1. ポートとケーブル

今日 PC 用に売られているプリンタには通常、次の 3 つのインタフェースのうち、どれか 1 つ以上がついてきます。

- シリアルインタフェース (RS-232 または COM ポートとも呼ばれます) は、コンピュータにあるシリアルポートを使ってプリンタにデータを送信します。 シリアルインタフェースはコンピュータ業界で共通して使用されています。 そのケーブルは容易に手に入りますし、簡単に自作することもできます。 シリアルインタフェースの場合は時々、 特別なケーブルや何か複雑な通信方式選択の設定が必要になることがあります。ほとんどの PC のシリアルポートは通信速度が最大で 115200 bps であり、大きな画像を印刷するのには実用的ではありません。

- パラレルインタフェースではプリンタにデータを送信するために、コンピュータにあるパラレルポートを使用します。パラレルインタフェースは PC 業界ではよく使われており、RS-232 シリアルよりも速いです。ケーブルの入手は容易ですが、自作するのはシリアルよりも困難です。パラレルインタフェースには通常、通信方式の選択はなく、設定は極めて単純です。

  パラレルインタフェースは「セントロニクス」インタフェースとして知られています。これは、プリンタ用のコネクタタイプとして採用された後に名付けられました。

- USB インタフェースは、Universal Serial Bus (汎用シリアルバス) の略で、パラレルや RS-232 シリアルよりさらに速く動作します。ケーブルは単純で安価です。USB は、印刷目的には RS-232 シリアルやパラレルよりも向いていますが、UNIX® システムでは十分対応されていません。この問題を回避する手としては、多くのプリンタがそうですが、USB とパラレルの両方のインタフェースを備えたプリンタを購入することが挙げられます。

パラレルインタフェースでは、普通は (コンピュータからプリンタへの) 単方向通信のみを行なうのに対して、 シリアルおよび USB インタフェースは双方向通信を行ないます。FreeBSD でも IEEE1284 準拠のケーブルを使えば、最近のパラレルポート (EPP や ECP) とプリンタの多くで双方向通信を行なうことができます。

パラレルポート経由のプリンタとの双方向通信には、 通常 2 つの方法のどちらかが使われます。一つ目の方法は、 プリンタが使用しているプロプライエタリな言語を話す FreeBSD 用に作成されたプリンタドライバを使うものです。 これはインクジェットプリンタではよく使われる方法で、 インクの残量やその他の状態の情報を知らせるのに使えます。二つ目の方法は、プリンタが PostScript® に対応している時に使われます。

PostScript® ジョブは、実際にはプリンタに送信されるプログラムです。 印字作業を行う必要は必ずしありませんし、 プログラムの結果を直接コンピュータに返してもよいのです。PostScript® プリンタでは双方向通信を使って PostScript® プログラムのエラー

や紙づまりといった問題をコンピュータに報告します。 ユーザはそれらの情報を知りたいと思うかも知れません。 また、PostScript® プリンタで課金作業をもっとも効率よく行なうためには、双方向通信が必要となります。この方法ではまず、プリンタの現在のページカウント (起動してから今まで何枚の紙を印字したか) の情報を得ます。 次に、ユーザのジョブを実行し、終了後、再びページカウントを得ます。この二つの数を差によって、課金対象となる紙の枚数を知ることができるのです。

10.3.1.1.2. パラレルポート

プリンタをパラレルインタフェースを使って接続する場合は、セントロニクスケーブルでプリンタとコンピュータを接続してください。 詳しい説明はプリンタやコンピュータに付属する説明書に書かれているはずです。

その際、どのパラレルポートを使用したかを覚えておいてください。FreeBSD では最初のポートは `ppc0`、二番目が `ppc1` であり、三番目以降も同様に続きます。プリンタのデバイス名にも同じ形式が使われており、最初のパラレルポートに接続されたプリンタは `/dev/lpt0` などとなります。

10.3.1.1.3. シリアルポート

シリアルインタフェースを使ってプリンタを使う場合は、適切なシリアルケーブルでプリンタとコンピュータを接続してください。詳しい説明はプリンタ、コンピュータ、あるいは両方に付属する説 明書に書かれているはずです。

「適切なシリアルケーブル」が良くわからないときは、次のどれかを試してみてください。

- モデム用ケーブルでは、 それぞれのピンは他方のコネクタの対応するピンと線でつながっています。このタイプのケーブルは「DTE-DCE」間ケーブルとしても知られています (訳注: 日本ではストレートケーブルという名前で売られています)。

- ヌルモデム用ケーブルでは、あるピンは対応するピンとを接続していますが、あるピン (たとえば、データ送信用とデータ受信用のピン) が交差して接続したり、いくつかのピンは内部で短絡していたりします。このタイプのケーブルは、「DTE-DTE」間ケーブルと呼ばれています (訳注: 日本ではクロスケーブルという名前で売られています)。

- A シリアルプリンタ用ケーブルは、ある特定のプリンタで必要とされるものです。ヌルモデムケーブルと似ていますが、内部で短絡させる代わりに、ある信号を他方側に送るために使用しています。

この他に、プリンタ用の通信パラメータを設定する必要があります。通常、プリンタのフロントパネルや DIP スイッチによって制御します。コンピュータとプリンタの双方で設定できる最高の通信速度 [bps] (ビット/秒、 ボーレートと示されているときもある) を選んでください。そして、データビット (7 または 8)、パリティ (偶/奇/なし)、ストップビット (1 または 2) を選んでください。 そして、フローコントロールの有無 (制御なし、または XON/XOFF (「イン・バンド」または「ソフトウェア」フローコントロールとも呼ばれる)) を選びます。以下に続くソフトウェアの設定のために、ここでの設定を覚えておいてください。

10.3.1.2. ソフトウェアの設定

本節では FreeBSD の LPD スプーリングシステムで印字をおこなうために 必要となるソフトウェアの設定について説明しています。

本節の概要は次のようになります。

1. プリンタで使用するポートのために、必要があれば、カーネルの書き変えをおこないます。「カーネルの変更」で、このためにしなくてはならないことを説明しています。

2. パラレルポートを使用している場合は、パラレルポートのための通信モードを設定します。詳細は、「 パラレルポートの通信モードを設定する」で説明しています。

3. オペレーティングシステムからプリンタにデータが送ら れているかをテストします。「プリンタとの通信状況を調べる」で、 どのようにテストするかの提案をいくつかおこなっています。

4. ファイル/etc/printcap を変更し、LPD の設定をおこないます。この節で、どのように変更するかを説明しています。

10.3.1.2.1. カーネルの変更

オペレーティングシステムのカーネルの コンパイルをおこなうことによって、 指定されたデバイスが機能するようになります。シリアル、 または、パラレルインタフェースをプリンタで使用する場合、 必要なデバイスがこの指定の中に含まれていなくてはなりません。 したがって、必要なデバイスがカーネルに組み込まれていない場合、追加のシリアル、または、パラレルポートをサポートするために、カーネルの再コンパイルが必要となるかもしれません。

シリアルポートが現在使用しているカーネルで サポートされているかどうかを調べるためには、次のように入力します。

```
# grep sioN /var/run/dmesg.boot
```

ここで、*N* はシリアルポートの番号を示し、この番号は 0 から始まります。次のような出力があった場合、カーネルはそのポートをサポートしています。

```
sio2 at port 0x3e8-0x3ef irq 5 on isa
 sio2: type 16550A
```

パラレルポートが現在使用しているカーネルで サポートされているかどうかを調べるためには、次のように入力します。

```
# grep ppcN /var/run/dmesg.boot
```

ここで、*N* はパラレルポートの番号を示し、この番号は 0 から始まります。次のような出力があった場合、カーネルはそのポートをサポートしています。

```
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/8 bytes threshold
```

上記の出力が得られない場合、プリンタを使うため、 オペレーティングシステムにパラレル、または、シリアルポートを認識し、使用できるようにするためには カーネルを変更する必要があります。

シリアルポートをサポートさせるには、「 FreeBSD カーネルのコンフィグレーション」の節をご覧く ださい。パラレルポートをサポートさせる場合も、その節と、あわせて、この節に続く節もご覧ください。

10.3.1.3. パラレルポートの通信モードを設定する

パラレルインタフェースを使用している場合、FreeBSD では、 割り込み駆動型にするか、 プリンタとの通信の状況をカーネルに監視させるかのいずれかを選択できます。FreeBSD の汎用プリンタデバイスドライバ (lpt(4)) は ppbus(4) システムを利用しています。これは ppc(4) ドライバを使ってパラレルポートのチップセットを制御します。

- GENERIC カーネルでは割り込み駆動方式がデフォルトになっています。 この方式では、 オペレーティングシステムはプリンタがデータを受け付けられるかどうかを調べるために、IRQ ラインを一つ使用します。

- 監視方式では、 オペレーティングシステムにプリンタがもっとデータを受け付けられるかどうかを繰り返し尋ねるように指示します。 そして、受け付けるという応答を受けたとき、カーネルはさらなるデータを送信します。

割り込み駆動方式は一般的にいくらか高速になりますが、貴重な IRQ ラインを一つ消費します。 HP の新しいプリンタの一部には、明らかに何かしらのタイミングの問題 (まだ正確にはわかっていません) で割り込みモードでは正常に動作しないものがあると言われています。これらのプリンタにはポーリングモードが必要になります。どちらかうまく機能する方を使ってください。一部のプリンタはどちらの方式でも動作しますが、割り込みモードでは苦痛を感じるほど低速です。

通信モードを設定するためには 2 つの方法があります。1 つはカーネルを変更することで、もう一つは lptcontrol(8) プログラムを使用する方法です。

カーネルを設定することによって、通信モードを変更する。

1. カーネルコンフィグレーションファイルを変更します。`ppc0` のエントリを探してください。2 番目のパラレルポートを設定するときは、代わりに `ppc1` を使います。以下、3 番目のポートは `ppc2` となっていきます。

   - 割り込み駆動方式にする場合は、`/boot/device.hints` ファイルの以下の行を編集して、*N* を適切な IRQ 番号に置き換えてください。

     ```
     hint.ppc.0.irq="N"
     ```

カーネルの設定ファイルには ppc(4) ドライバも入れなければなりません。

```
device ppc
```

- ポーリングモードを使用する場合は、/boot/device.hints ファイルの以下の行を削除してください。

```
hint.ppc.0.irq="N"
```

場合によっては、これだけでは FreeBSD でポートをポーリングモードにするには十分ではないことがあります。多くの場合これは acpi(4) ドライバと併せて動作します。これはデバイスのプローブとアタッチを行うので、プリンタポートへのアクセスモードを制御できます。問題を修正するために acpi(4) の設定を確認してください。

2. ファイルをセーブし、config プログラムを起動し、カーネルの構築、インストールをおこないます。そして、リブートしてください。詳細は、「FreeBSDカーネルのコンフィグレーション」を参照 してください。

lptcontrol(8) で通信モードを設定する場合

1. lptN をイベント駆動方式に設定する場合は、次のように入力します。

```
# lptcontrol -i -d /dev/lptN
```

2. lptN を監視方式に設定する場合は、次のように入力します。

```
# lptcontrol -p -d /dev/lptN
```

これらのコマンドを /etc/rc.local ファイルに追加 しておくと、システムをブートする度に通信モードを設定する ことができます。詳細については、lptcontrol(8) をご覧ください。

10.3.1.4. プリンタとの通信状況を調べる

スプーリングシステムの設定に進む前に、オペレーティング システムがプリンタにデータを送ることに成功しているかどうか を確かめるべきでしょう。これにより、印字がうまくいかないと き、プリンタとの通信が問題なのか、スプーリングシステムが問 題なのかを分けて調べることがかなり容易になります。

プリンタをテストするためには、プリンタに何かのテキストを送 信してみます。送信した文字をすぐに印字してくれるプリンタに は、lptest(1) コマンドを使うと有用です。このコマンドは印 字可能な 96 文字の ASCII 文字すべてを 96 行生成します。

PostScript® (または他の言語に対応した) プリンタの場合 は、もっと巧妙なテストが必要になります。次のような、簡単な PostScript® プログラムを使えば十分でしょう。

```
%!PS
100 100 moveto 300 300 lineto stroke
310 310 moveto
/Helvetica findfont 12 scalefont setfont
(Is this thing working?) show
showpage
```

上の PostScript® コードはファイルに保存し、以降の節で例として示されているように利用することができます。

注記

このドキュメントでプリンタ用言語を参照するときは、PostScript® のような言語を仮定しており、Hewlett Packard の PCL は考慮していません。PCL は非常に機能的なの ですが、プレインテキストにエスケープシーケンスを混ぜること ができます。PostScript® ではプレインテキストを直接印字 することはできません。 このような種類のプリンタ言語に対しては、特別な対応をおこなわなければなりません。

10.3.1.4.1. パラレルポートのプリンタとの接続を調べる

この節では、FreeBSD がパラレルポートに接続されたプリ ンタと通信できているかどうかを調べる方法について説明し ています。

パラレルポートのプリンタをテストするために

1. su(1) コマンドで `root` になります。
2. プリンタにデータを送ります。
   - プリンタがプレインテキストを印字できる場合、lptest(1) コマンドを使います。次のように入力してください。

     ```
     # lptest > /dev/lptN
     ```

     ここで、*N* はパラレルポートの番号で、番号は 0 から始まります。
   - プリンタが PostScript® か他のプリンタ 言語を使用している場合、そのプリンタに簡単なプロ グラムを送信してください。次のように入力します。

     ```
     # cat > /dev/lptN
     ```

     そして、一行一行、 プログラムを慎重に入力して 下さい。RETUREN または ENTER キーを入力してしま うと、その行は編集できなくなります。プログラムの

入力が終わったら、CONTROL+D か、あなたが設定して いるファイル終了のキーを押してください。

もしくは、プログラムを入力したファイルがある 場合は、次のように入力してください。

```
# cat file > /dev/lptN
```

ここで、*file* はプログラムが格納されていて、プリンタに送信するファイルの名前です。

これで何かが印刷されるはずです。 印字されたテキストがおかしくても心配は無用です。それについては、後で修正します。

10.3.1.4.2. シリアルポートのプリンタとの接続を調べる

この節では、FreeBSD がシリアルポートに接続されたプリ ンタと通信できているかどうかを調べる方法について述べられ ています。

シリアルポートのプリンタをテストするために

1. su(1) コマンドで root になります。
2. /etc/remote ファイルを編集します。次のエントリを加えてください。

   ```
   printer:dv=/dev/port :br#bps-rate :pa=parity
   ```

   ここで、*port* シリアルポート (ttyu0、ttyu1 など) のデバイスエントリで、*bps-rate*は プリンタとの通信の転送速度[bit/秒]、*parity* はプリ ンタとの通信で必要とされるパリティ (even、odd、none、zeroのいずれか) を表わしていま す。

   次の例は、プリンタをシリアルケーブルでパリティなし、転送速度 19200 bps で第 3 番目のシリアルポートに接続した場 合です。

   ```
   printer:dv=/dev/ttyu2 :br#19200:pa=none
   ```

3. tip(1) コマンドでプリンタと接続します。次のように入力してください。

   ```
   # tip printer
   ```

   これがうまくいかなかった場合は、/etc/remote を編集して、/dev/ttyuN の代わりに /dev/cuaaN を試してみてください。

4. プリンタにデータを送ります。

   - プリンタがプレインテキストを印字できる場合、lptest(1) コマンドを使います。次のように入力してください。

```
% $lptest
```

- プリンタが PostScript® か他のプリンタ言語を使用している場合、そのプリンタに簡単なプログラムを入力します。一行一行、プログラムを慎重に入力してください。バックスペースキーや他の編集用のキーは、プリンタの制御コードに割り当てられているかもしれません。プログラムが終了したことをプリンタに伝えるための特別なファイル終了キーを 入力する必要があるかもしれません。PostScript® プリンタの場合、CONTROL+D を入力します。

  もしくは、プログラムを入力したファイルがある場合は、次のように入力してください。

  ```
  % >file
  ```

  ここで、*file* はプログラムが格納されているファイル名です。tip(1) コマンドでファイルを送信した後は、ファイル終了を表わすキーを入力する必要があります。

これで何かがプリントされることでしょう。印字されたテキ ストがおかしくても心配しなくても構いません。それについては、後で修正します。

10.3.1.5. スプーラに許可を与える: /etc/printcap ファイル

ここまでで、プリンタはコンピュータに接続され、(必要なら) プリンタと通信できるようにカーネルを変更し、簡単なデータをプリンタに送信することができているはずです。これで、LPD にプリンタへのアクセスを 制御させる設定をおこなう準備が整いました。

LPD の設定は /etc/printcap を編集することでおこないます。LPD スプーリングシステムは スプーラが使われる毎にこのファイルを参照します。そのため、ファイルを更新するとすぐにその変更が反映されます。

printcap(5) ファイルの書式は簡単です。/etc/printcap の編集はお好みのテキストエディタをお 使いください。このファイルの書式は、/usr/share/misc/termcap や /etc/remote といった他のケイパビリティファイルと一致しています。この書式 についての詳細な情報については cgetent(3) をご覧ください。

スプーラの単純な設定法は、次のステップでおこないます。

1. プリンタに名前 (と簡単な別名 2 ～ 3 個) を付け、それを /etc/printcap ファイルに記述します。これについては、「 プリンタに名前を付ける」を参照してください。

2. sh の項目を追加することで、ヘッダページの出力を禁止します (デフォルトは許可)。これについては、「 ヘッダページの印字を禁止する」を参照してください。

3. スプール用のディレクトリを作成し、その位置を sd 項目で指定します。これについては、「 スプーリングディレクトリの作成」を参照してください。

4. プリンタを使用するために /dev エントリを設定し、/etc/printcap の lp 項目でそのエントリを指定します。これについては、「 プリンタデバイスの特定」を参照してください。プリンタをシリアルポートに接続した場合は、ms# の項目を設定する必要があります。こちらについては、「 スプーラのための通信パラメータの設定」を参照してください。

5. プレインテキスト用の入力フィルタのインストールをおこないます。「テキストフィルタのインストール」を参照してください。

6. lpr(1) コマンドで何かを印字することで設定のテストをおこないます。印字してみよう と トラブルシューティング を参照してください。

注記

PostScript® プリンタのような、プリンタ言語を使用しているプリンタには、プレインテキストを直接印字させることができません。上にアウトラインを示し、 以下の節で説明する簡単な設定方法の説明では、 そのようなプリンタを設置している場合は、 プリンタが認識できるファイルだけを印字の対象としているという 仮定をしています。

多くの場合、 利用者はシステムに設置されているプリンタすべてで プレインテキストが印字できることを期待しています。 印字作業をおこなうために LPD のインタフェースを利用するプログラムでも、 通常、そのような仮定を置きます。 プリンタ言語を使用するプリンタを設置しており、 そのプリンタ言語で記述されたジョブと、 これに加えて、 プレインテキストのジョブも印字できるようにしたいならば、 上で示した簡単な設定方法に加えて、さらなる設定をおこなうことを強くお勧めします。すなわち、自動的にプレインテキストから PostScript® (もしくは、他のプリンタ言語) に変換するプログラムをインストールしてください。「 プレインテキストのジョブを PostScript® プリンタで印字する」で、それをどのようにおこなえばよいのかが説明されています。

訳注

日本語を印字したい場合は、プリンタ言語を使用し ていない「日本語プリンタ」についても、プリンタ固有のエスケープシーケンスを送る必要があります。また、漢字コードをプリン タが設定しているものに変換したりする必要があり、 各プリンタ毎に、日本語用のフィルタが必要になります。

10.3.1.5.1. プリンタに名前を付ける

最初の (簡単な) ステップで、プリンタの名前を考えます。プリンタには別名をいくつか付けることもできるので、機能的な名前 でも風変わりな名前でもどちらを選んでもまったく問題はありません。

少なくとも1つのプリンタには、 /etc/printcap の中で、 lp という別名を持たせるべきでしょう。この名前はデフォルトのプリンタ名になっています。ユーザが環境変数 PRINTER を設定しておらず、かつ、LPD コマンドのコマンドラインで プリンタの名前が指定されていない場合、lp がデフォルトのプリンタ名となり、そのプリンタに出力されます。

それから、これは共通の慣習ですが、 プリンタの最後の別名には、 メーカーやモデル名を含むプリンタの完全な名称をつけることに なっています。

名前と別名のいくつかを決めたら、/etc/printcap ファイルに設定します。プリンタ名は一番左のカラムから書き始めます。別名はそれぞれ縦棒によって区切られ、最後の別名の後ろにコロンを置きます。

次の例では、2 台のプリンタ (Diablo 630 ラインプリンタと Panasonic KX-P4455 PostScript® レーザライタプリンタ) が定義 されている /etc/printcap のスケルトンを記しています。

```
#
#  /etc/printcap for host rose
#
rattan|line|diablo|lp|Diablo 630 Line Printer:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:
```

この例では、最初のプリンタに rattan という名前と別名として、line、diablo、lp そして Diablo 630 Line Printer が付けられています。別名とし て lp があるので、このプリンタはデフォルトのプリンタとなっ ています。2 番目は bamboo と名付けられ、別名として、ps と PS、S、panasonic 、Panasonic KX-P4455 PostScript v51.4 が付けられています。

10.3.1.5.2. ヘッダページの印字を禁止する

LPD スプーリングシステムでは、 デフォルトでジョブ毎に ヘッダページを印字します。ヘッダページにはジョブを要求したユーザ名、 ジョブが送られたホスト名、そして、ジョブの名前が素晴 らしい大きな文字で印字されています。 残念なことに、この余分なテキストすべてが、簡単なプリンタ設定法のデバッグの際に紛れ込んできてしまいます。このため、ヘッダページの出力を禁止しておきます。

ヘッダページの出力を禁止するには、 /etc/printcap にあるプリンタのエントリに sh の項目を追加します。次に、sh を加えた /etc/printcap の例を示します。

```
#
```

```
#  /etc/printcap for host rose - no header pages anywhere
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:
```

この書式を正しく使うための注意をしておきます。 最初の行は左端のカラムから始めます。 それに続く行は字下げします。最後の行以外のすべての行は、 行末にバックスラッシュを記述します。

10.3.1.5.3. スプーリングディレクトリの作成

スプーラの簡単な設定の次のステップでは、 スプーリングディレクトリを作成します。 プリンタに送られるジョブは、その印字が終了するまでこのディレクトリに置かれます。また、他のたくさんのスプーラもこのディレクトリにファイルを置きます。

様々な事情によりスプーリングディレクトリは、通常、慣例 として `/var/spool` の下に置きます。また、スプーリングディレクトリの内容は バックアップをする必要はありません。mkdir(1) によってディレクトリを 作るだけでスプーリングディレクトリの復旧は完了します。

スプーリングディレクトリの名前は、これも慣例ですが、 次のようにプリンタの名前と同じにします。

```
# mkdir /var/spool/printer-name
```

しかしながら、ネットワーク上に使用可能なプリンタがたく さんあるならば、LPD で印字するための専用のディレクトリにスプーリングディレクトリを置きたくなるかもしれません。 例に出てきたプリンタ `rattan` と `bamboo` について、この方式を採用すると、次のようになります。

```
# mkdir /var/spool/lpd
# mkdir /var/spool/lpd/rattan
# mkdir /var/spool/lpd/bamboo
```

注記

各ユーザが印字するジョブのプライバシを守りた いと考えているならば、スプーリングディレクトリを保護し て、これを誰からでもアクセスできないようにしたいと思う かもしれません。スプーリングディレクトリは、 `daemon` ユーザと `daemon` グループに所有され、読み込み、書き込み、検 索可能であり、他からはアクセスできないようにするべきで す。例題のプリンタに対して、次のようにすることにしましょ う。

```
# chown daemon:daemon /var/spool/lpd/
rattan
# chown daemon:daemon /var/spool/lpd/
bamboo
# chmod 770 /var/spool/lpd/rattan
# chmod 770 /var/spool/lpd/bamboo
```

最後に、/etc/printcap ファイルで、これらのディレクトリの位置を LPD に伝える必要があります。スプーリングディレクトリのパス名は sd 項目で指定します。

```
#
#  /etc/printcap for host rose - added spooling directories
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan :

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo :
```

プリンタ名が最初のカラムから始まっており、 そのプリンタに関して記述される他の項目は字下げされていること、 各行がバックスラッシュで終わっていることに注意してください。

sd によりスプーリングディレクトリが指定されていない場合、スプーリングシステムは /var/spool/lpd をデフォルト値として使用します。

10.3.1.5.4. プリンタデバイスの特定

プリンタ機器の設定 の節では、FreeBSD でプリンタとの通信に使用されるポートおよび /dev ディレクトリ内のエントリを特定します。そして、LPD にその情報を伝えます。印字するジョブを受け取ると、スプーリングシステムは、 (プリンタにデータを渡す義務がある) フィルタプログラムに代わって指定されたデバイスをオープンします。

/etc/printcap ファイルで lp 項目を使って /dev エントリを記入します。

ここでの例では、rattan は 1 番目のパラレルポートに、bamboo は 6 番目のシリアルポートに接続されていることにしましょう。このとき、/etc/printcap には 次のようになります。

```
#
#  /etc/printcap for host rose - identified what devices to use
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan :\
        :lp=/dev/lpt0 :
```

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo :\
        :lp=/dev/ttyu5 :
```

/etc/printcap でプリンタの lp 項目が指定されていない場合は、LPD はデフォルトとして /dev/lp を使用します。/dev/lp は、現在の FreeBSD には存在していません。

設置したプリンタがパラレルポートに 接続されている場合は、「 テキストフィルタのインストール」まで読み飛ばしてください。そうでない場合は、次節の説明に続いてください。

10.3.1.5.5. スプーラのための通信パラメータの設定

シリアルポートにプリンタを接続した場合、LPD は、プリンタにデータを送信するフィルタプログラムに代わり、通信速度やパリティ、その他のシリアル通信パラメータを設定することができます。このことによる利点は、

- /etc/printcap を編集するだけで、様々な通信パラメータを試してみることができます。フィルタプログラムを再コンパイルする必要はありません。
- スプーリングシステムで、シリアル通信の設定が異なっているかもしれない複数のプリンタに 同じフィルタプログラムを使うことが可能になります。

次の /etc/printcap の項目で、lp で指定された デバイスのシリアル通信パラメータを制御できます。

br#*bps-rate*
: デバイスの通信速度を *bps-rate* に設定します。ここで、*bps-rate* は 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 [bit/秒] のいずれかです。

ms#*stty-mode*
: デバイスをオープンした後にターミナルデバイスのオプションを設定します。 利用できるオプションについては stty(1) を参照してください。

lp で指定されたデバイスをオープンするとき、LPD は ms# で指定されたデバイスの特性を設定します。特に関係があるのは、parenb, parodd, cs5, cs6, cs7, cs8, cstopb, crtscts, ixon モードです。これらは stty(1) のマニュアルページで説明されています。

例題のプリンタで6番目のシリアルポートに接続された　プリンタの設定を追加してみましょう。通信速度は 38400bps に設定します。モードとして、-parenb でパリティ無し、cs8 で 8 ビットキャラクタ、clocal でモデム制御無し、そして crtscts でハードウェアフロー制御を設定します。

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
```

```
        :sh:sd=/var/spool/lpd/bamboo :\
        :lp=/dev/ttyu5 :ms#-parenb cs8 clocal crtscts:
```

10.3.1.5.6. テキストフィルタのインストール

ここまでで、プリンタにジョブを送るために使うテキストフィルタを LPD に設定する準備が整いました。テキストフィルタとは、入力フィルタとしても知られていますが、印字するジョブがあるときに LPD が起動するプログラムです。LPD がプリンタのためにテキストフィルタを起動するとき、LPD はフィルタの標準入力からプリントするジョブを入力し、フィルタの標準出力に項目 `lp` で指定されたプリンタデバイスを接続します。フィルタは、標準入力からジョブを読み込み、プリンタのための必要な変換をおこなった後、その結果を標準出力に出力する、これにより印字がなされることを期待されています。テキストフィルタについての更に詳しい情報については、「フィルタはどのように機能しているか」をご覧ください。

ここでの簡単なプリンタ設定では、プリンタにジョブを送るため、`/bin/cat` を実行するだけの簡単なシェルスクリプトで間に合います。FreeBSD に標準で付属している `lpf` というフィルタでは、バックスペース文字を使った下線引きの動作をおこなう文字ストリームをうまく扱うことができないプリンタのための代替処理をおこなってくれます。もちろん、他のどんなフィルタプログラムを使っても構いません。フィルタ `lpf` については、「テキストフィルタ lpf」で詳しく説明します。

最初に、簡単なテキストフィルタであるシェルスクリプト `/usr/local/libexec/if-simple` を作ってみましょう。次のテキストをお好みのテキストエディタでファイルに書き込んでください。

```
#!/bin/sh
#
# if-simple - Simple text input filter for lpd
# Installed in /usr/local/libexec/if-simple
#
# Simply copies stdin to stdout.  Ignores all filter arguments.

/bin/cat && exit 0
exit 2
```

そして、このファイルを実行可能にします。

```
# chmod 555 /usr/local/libexec/if-simple
```

LPD にこのテキストフィルタを使うことを設定するためには、`/etc/printcap` に `if` 項目を使って指定します。これまでの `/etc/printcap` の例のプリンタ 2 台に、このフィルタを加えてみましょう。

```
#
#  /etc/printcap for host rose - added text filter
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
```

```
        :sh:sd=/var/spool/lpd/rattan :\ :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple :

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo :\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:\
        :if=/usr/local/libexec/if-simple :
```

注記

`if-simple` スクリプトのコピーが `/usr/share/examples/printing` ディレクトリにあります。

10.3.1.5.7. LPD の起動

lpd(8) は `lpd_enable` 変数に従って `/etc/rc` から実行されます。この変数の デフォルト値は `NO` です。まだ そうしていなかったならば

```
lpd_enable="YES"
```

の行を `/etc/rc.conf` に追加して 計算機を再起動するか、そのまま lpd(8) を 起動してください。

```
# lpd
```

10.3.1.5.8. 印字してみよう

簡単な LPD 設定も終わりにたどり着きました。 残念ながら、設定はこれでおしまいというわけではありません。なぜなら、さらに、設定をテストし、すべての問題点を解決しなくてはならないからです。 設定をテストするために、 何かを印字してみましょう。LPD システムで印字をするためには、 lpr(1) コマンドを使います。このコマンドは、 印字するためのジョブを投入する働きをします。

lpr(1) コマンドを「 プリンタとの通信状況を調べる」で紹介した、あるテスト用のテキストを生成してくれる lptest(1) プログラムと一緒に使うこともできます。

簡単な LPD 設定のテスト

次のように入力してください。

```
# lptest 20 5 | lpr -Pprinter-name
```

ここで、*printer-name* は `/etc/printcap` で指定したプリンタ名 (もしくはその別名) です。デフォルト のプリンタを使用する場合は、`-P` 引数を付けないで lpr(1) を打ち

込んでください。もう一度述べますが、PostScript® を期待しているプリンタをテストするならば、lptest(1) を使う代わりに PostScript® で書かれた プログラムをプリンタに送ってください。プログラムを送るためには、プログラムをファイルに格納して、`lpr file` と打ち込みます。

PostScript® プリンタの場合、 送信したプログラムによる結果が得られるでしょう。lptest(1) を使った場合は、以下のような結果が見られるでしょう。

```
!"#$%&'()*+,-./01234
"#$%&'()*+,-./012345
#$%&'()*+,-./0123456
$%&'()*+,-./01234567
%&'()*+,-./012345678
```

更にプリンタをテストしたい場合は、 (言語ベースのプリンタのための) もっと大きなプログラムを送信するか、引数を変えて lptest(1) を実行します。たとえば、`lptest 80 60` で、それぞれ 80 文字の行を 60 行生成します。

プリンタがうまく動かなかった場合は、次の節、「 トラブルシューティング」をご覧ください。

10.4. プリンタ設定上級編

この節では、特殊な形式のファイルを印字するためのフィルタ、 ヘッダページ、ネットワーク越しのプリンタへの印字、そして、 プリンタ使用の制限や課金について説明しています。

10.4.1. フィルタ

LPD は、ネットワークプロトコル、キュー、アクセス制御などの 印刷にかかわるさまざまな点を扱いますが、 実際の作業のほとんどは フィルタによっておこなわれています。 フィルタは、プリンタと通信し、 プリンタのデバイス依存性や特殊な要求を扱うプログラムです。簡単なプリンタ設定では、プレインテキストのためのフィルタをインストールしました。このプレインテキストフィルタは、ほとんどのプリンタで機能する極めて単純なものでした (「 テキストフィルタのインストール」を参照)。

しかしながら、形式変換やプリンタ課金、特定のプリンタの癖、など をうまく利用するためには、フィルタがどのように機能するかという ことを理解しておくべきです。これらの側面を扱うことは、最終的には、フィルタの責任であるからです。そして、これは悪い情報ですが、ほとんどの場合において、あなた自身が フィルタを供給する必要があるということです。また都合のよいことには、 たくさんのフィルタが一般的に利用できるということです。もしフィルタがなかったとしても、普通はフィルタを作るのは簡単です。

FreeBSD にも、プレインテキストを印字させることができる `/usr/libexec/lpr/lpf` というフィルタが 1 つ付いています (このフィルタはファイルに含まれるバックス

ペースやタブを扱います。また、課金をすることもできますが、できることはこれだけしかありません)。いくつかのフィルタとフィルタの構成要素は FreeBSD Ports Collection にもあります。

この節で述べることは次の通りです。

- 「 フィルタはどのように機能しているか」では、 印字の過程におけるフィルタの役割を概説します。 この節を読むことで、LPD がフィルタを使うときに、「見えないところで」 何が起こっているかが理解できるでしょう。このことを知っておくと、 プリンタそれぞれに様々なフィルタをインストールしたときに 遭遇するかもしれない問題を予期したり、デバッグするときに役立つでしょう。

- LPD は、すべてのプリンタがデフォルトでプレインテキストを印字できることを期待しています。これは、プレインテキストを直接印字できない PostScript® (または他の言語対応の) プリンタで問題になります。「 プレインテキストのジョブを PostScript® プリンタで印字する」で、この問題を克服する方法について述べます。PostScript® プリンタをお持ちの方は、この節をお読みになることをおすすめします。

- PostScript® は様々なプログラムのための有名な出力形式です。 PostScript® のコードを直接書いてしまう人すらいます。 残念ながら、PostScript® プリンタは高価です。「非 PostScript® プリンタによる PostScript® のシミュレート」節では、PostScript® データを非 PostScript® プリンタに受けつけさせ、印字させるために、どのようにしてプリンタ用のテキストフィルタをさらに変更すればよいのか、ということについて説明しています。PostScript® プリンタを持っていない方は、この節をお読みになることをおすすめします。

- 「 変換フィルタ」では、図形や組版データといった特定のファイル形式を、プリンタが理解できる形式へ変換する作業を自動的におこなわせる方法について述べます。 この節を読むと、troff のデータを印字するには `lpr -t`, または、TeX DVI を印字するには `lpr -d`、ラスタイメージデータを印字するには `lpr -v`、などといったようにユーザが入力することができるように プリンタの設定をおこなうことができます。この節もお読みになることをお薦めします。

- 「出力フィルタ」では、あまり使われない LPD の機能のすべて、すなわち、出力フィルタに関することが記述されています。ヘッダページ (「 ヘッダページ」参照) を印字させていない場合は、多分、この節は飛ばしても構わないでしょう。

- 「テキストフィルタ lpf」では、`lpf` についての説明が、ほぼ完全におこなわれています。これは FreeBSD に付属するラ インプリンタ (または、ラインプリンタのように動作するレーザプリンタ) のための、単純なテキストフィルタです。プレインテキストを印字したことに対して課金をおこなう方法が 至急必要な場合、もしくは、バックスペース文字を印字しようと すると煙を発するプリンタを持っている場合は、絶対に `lpf` を検討するべきです。

注記

以下で述べられているさまざまなスクリプトは、`/usr/share/examples/printing` ディレクトリにあります。

10.4.1.1. フィルタはどのように機能しているか

既に言及したように、フィルタとは、プリンタにデータを送る際に、 デバイスに依存した部分を取り扱うために LPD によって起動される実行プログラムです。

LPD がジョブ中のファイルを印字しようとするとき、LPD はフィルタプログラムを起動します。このとき、 フィルタの標準入力を印字するファイルに、 標準出力をプリンタに、そして、標準エラー出力を エラーログファイル (`/etc/printcap` 内の `lf` 項目で指定されたファイル、または、指定されていない場合は、デフォルトとして `/dev/console`) にセットします。

LPD が起動するフィルタと、その引数が何であるかは、`/etc/printcap` ファイルの内容と、ジョブの起動時にユーザが指定した lpr(1) コマンドの引数に依存しています。たとえば、ユーザが `lpr -t` と入力した場合は、LPD は出力先のプリンタ用の `tf` 項目で指定されている troff 用のフィルタを起動させるでしょう。ユーザがプレインテキストの印字を指示したときは、`if` で指定されたフィルタが起動されるでしょう (このことはほとんどの場合にあてはまります。詳細については、「 出力フィルタ」をご覧ください)。

`/etc/printcap` で指定可能なフィルタは次の3種類があります。

- テキストフィルタ (LPD のドキュメントでは紛らわしいことに 入力フィルタと呼んでいますが) は一般のテキストの印字を扱います。これはデフォルトのフィルタと 考えてください。LPD では、すべてのプリンタに対して、デフォルトでプレインテキストが印字できることを期待しています。 さらに、バックスペースやタブを正しく扱い、また、 他の特殊な文字が入力されてもプリンタに混乱を来さないように するのはテキストフィルタの仕事であると考えています。 プリンタの使用に対して課金をしなくてはならない環境にあるときは、テキストフィルタが印字したページ数を数える作 業もしなくてはなりません。この作業は、通常、印字した行 数を数え、これをプリンタが 1 ページ当たりに印字できる行 数と比較することでおこなわれます。 テキストフィルタは、次のような引数を付けて起動されます。

  `filter-name` [-c] -w *`width`* -l *`length`* -i *`indent`* -n *`login`* -h *`host`* *`acct-file`*

  ここで、

`-c`
: `lpr -l` によってジョブが入力されたときに与えられます。

`width`
: `/etc/printcap` で指定された `pw` (page width) 項目の値が与えられます。デフォルトは、132 です。

`length`
: `pl` (page length) 項目で指定された値が与えられます。デフォルトは 66 です。

`indent`
: `lpr -i` によって与えられた字下げの量で、デフォルトは 0 です。

`login`
: ファイルを印字したユーザのアカウント名が 与えられます。

`host`
: ジョブが入力されたホスト名が 与えられます。

`acct-file`
: `af` 項目で指定されている課金データファイル の名前が与えられます。

- 変換フィルタは、 特定のファイル形式をプリンタ が紙に印字できるようなものに変換します。たとえば、 プリンタで ditroff 組版データを直接印字することはできません。しかし、ditroff データをプリンタが消化し、 印字することができる形式へ変換するために、ditroff ファイル用フィルタをインストールすることができます。「 変換フィルタ」で、これらに関するすべてについて説明します。 プリンタの課金をする必要がある場合は、変換フィルタでも印字ページを数える作業が必要となります。変換フィルタは次の引数をとって起動されます。

  `filter-name` -x *`pixel-width`* -y *`pixel-height`* -n *`login`* -h *`host`* *`acct-file`*

  ここで、*`pixel-width`* は、`px` 項目で指定された値 (デフォルトは 0)、*`pixel-height`* は、`py` 項目で指定された値 (デフォルトは 0) です。

- 出力フィルタは、テキストフィルタが指定されて おらず、かつ、ヘッダページの出力が許可されている場合にのみ使われます。「 出力フィルタ」で、これらのことについて説明します。出力フィルタに対する引数は次の 2 つだけです。

  `filter-name` -w *`width`* -l *`length`*

  ここで、`-w` と `-l` は、テキストフィルタの場合と同じです。

フィルタは、次に示す終了状態をもってプログラムを exit するべきです。

exit 0
　フィルタがファイルを正常に印字した場合。

exit 1
　フィルタはファイルの印字に失敗したが、LPD に再度ファイルの印字を試みて欲しい場合。この終了状態で終了した場合、LPD はフィルタを再スタートします。

exit 2
　フィルタはファイルの印字に失敗し、かつ、LPD に再出力を試みて欲しくない場合。この場合、LPD はそのファイルを放棄します。

FreeBSD に付属するテキストフィルタ `/usr/libexec/lpr/lpf` は、FORM FEED 文字が送られたときやプリンタ使用に対する課金をどのようにするかを決定するために、ページ幅やページ長の引数を利用します。また、課金用のエントリを作成するため、ログイン名、ホスト名、課金ファイル名の引数を利用します。

もし、フィルタの購入を検討しているならば、LPD と互換性があるかどうかを確認してください。もしそうならば、上述の引数リストをサポートしていなければなりません。一般向けの使用のためにフィルタを作成する計画をしている場合は、同じ引数リストと終了コードをサポートしてください。

10.4.1.2. プレインテキストのジョブを PostScript® プリンタで印字する

コンピュータと PostScript® (または、他の言語に対応した) プリンタをあなたしか使用しない場合は、プリンタにプレ インテキストを絶対に送らない、そして、プリンタにプレインテキストを送りたがっている 様々なプログラムの機能を決して使わないことにしてください。そうすれば、この節に書かれたことに心を煩わせる必要はまったくなくなります。

しかし、PostScript® とプレインテキストの両方のジョブをプリンタへ送りたいと思っている場合は、プリンタ設定についての要求が増えるでしょう。両者をプリンタへ送信するためには、到着したジョブがプレインテキストであるか PostScript® であるかを検出するテキストフィルタが必要です。PostScript® のジョブはすべて `%!` で始まらなければならないことになっています (他のプリンタ言語に関しては、プリンタのドキュメントをご覧ください)。ジョブの最初の 2 文字がこれならば、PostScript® であることが分かります。したがって、ジョブのそれ以降の部分をプリンタに直接送ることができます (訳注: PostScript® では、% 以降はコメントとして扱われるので、最初の %! の行を読み捨てても問題はない)。最初の2文字が %! でない場合は、フィルタはテキストを PostScript® に変換し、その結果を使って印字をおこないます。

この作業をどうやってやればよいのでしょうか。

シリアルポートにプリンタを接続した場合は、`lprps` をインストールすることをお勧めします。`lprps` は PostScript® 用のフィルタで、プリンタとの双方向通信をおこないます。このフィルタでは、プリンタからの冗長な情報を得ることで、プリンタの状況を示すファイルが更新されていきます。したがって、ユーザや管理者は (トナー残量少や 紙

詰まりといった) プリンタの状況を正確に知ることができます。しかし、もっと重要なことは、`psif` と呼ばれるプログラムが含まれているということです。このプログラムは、入力されたジョブがプレインテキストかどうかを検出し、これを PostScript® に変換するために、`textps` (`lprps` に付属する別のプログラム) を呼び出します。そして、このジョブをプリンタに送るために、`lprps` が使われます。

`lprps` は FreeBSD Ports Collection に含まれています (Ports Collection を参照してください)。紙のサイズに合わせて print/lprps-a4 または print/lprps-letter port をインストールしてください。`lprps` をインストールした後は、`lprps` の一部である `psif` プログラムのパス名を指定するだけです。Ports Collection から `lprps` をインストールしたときは、`/etc/printcap` の中のシリアル接続した PostScript® プリンタのエントリに対して、次を使ってください。

```
:if=/usr/local/libexec/psif :
```

LPD にプリンタをリード・ライトモードでオープンさせるために、`rw` 項目も指定すべきです。

パラレルポート接続の PostScript® プリンタの場合 (すなわち、`lprps` が 必要としているプリンタとの双方向通信ができない)、テキストフィルタとして次のシェルスクリプトを使うことができます。

```
#!/bin/sh
#
#  psif - Print PostScript or plain text on a PostScript printer
#  Script version; NOT the version that comes with lprps
#  Installed in /usr/local/libexec/psif
#

IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

if [ "$first_two_chars" = "%!" -]; then
   #
   #  PostScript job, print it.
   #
   echo "$first_line" && cat && printf "\004" && exit 0
   exit 2
else
   #
   #  Plain text, convert it, then print it.
   #
   ( echo "$first_line"; cat ) | /usr/local/bin/textps && printf
 "\004" && exit 0
   exit 2
fi
```

上記のスクリプトにおいて、`textps` はプレインテキストから PostScript® へ変換するために別にインストールしたプログラムです。 テキストから PostScript® へ変

換するのには、 お好みのどんなプログラムでも使うことができます。FreeBSD Ports Collection (Ports Collection を参照してください) には、a2ps と呼ばれるテキストから PostScript® に変換するプログラムが入っています。

10.4.1.3. 非 PostScript® プリンタによる PostScript® のシミュレート

PostScript® は質の高い組版と印字をおこなうための 事実上の標準です。しかしながら、PostScript® は、高価な標準です。ありがたいことに、 Aladdin Enterprises から Ghostscript と呼ばれる、PostScript® 互換の動作をするフリーのプログラムが出されていて、FreeBSD で動きます。Ghostscript はほとんどの PostScript® ファイルを読むことができ、これらの各ページを多くのブランドの非 PostScript® プリンタを含む 様々なデバイス用に変換することができます。 Ghostscript をインストールし、プリンタ用の特別なテキストフィルタを使うことによって、非 PostScript® プリンタをあたかも本物の PostScript® プリンタであるかのように動作させることができます。

Ghostscript は FreeBSD Ports Collection に入っています。複数のバージョンがありますが、最も良く使われているバージョンは print/ghostscript-gpl です。

PostScript® プリンタをシミュレートさせる場合は、 テキストフィルタに PostScript® ファイルを印字しようとしているかどうかを検出させます。 PostScript® ファイルでない場合は、 フィルタはそのファイルを直接プリンタに送ります (訳注: テキストファイルを直接印字できない場合は、もちろん、変換フィルタを通す必要があります)。PostScript® の場合は、まず、Ghostscript を使い、ファイルをそのプリンタが理解できる形式へ変換します。

次の例のスクリプトは、Hewlett Packard DeskJet 500 プリンタ用 のテキストフィルタです。他のプリンタで用いるときは、-sDEVICE 引数を gs (Ghostscript) コマンドに変えてください (gs -h と入力すると、現在インストールされている Ghostscript でサポートされているデバイスのリストが得られます)。

```
#!/bin/sh
#
#  ifhp - Print Ghostscript-simulated PostScript on a DeskJet 500
#  Installed in /usr/local/libexec/ifhp

#
#  Treat LF as CR+LF (to avoid the "staircase effect" on HP/PCL
#  printers):
#
printf "\033&k2G" || exit 2

#
#  Read first two characters of the file
#
IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`
```

```
if [ "$first_two_chars" = "%!" -]; then
    #
    #  It is PostScript; use Ghostscript to scan-convert and print ↺
it.
    #
    /usr/local/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=djet500 \
      -sOutputFile=- - && exit 0

else
    #
    #  Plain text or HP/PCL, so just print it directly; print a ↺
form feed
    #  at the end to eject the last page.
    #
    echo "$first_line" && cat && printf "\033&l0H" &&
    exit 0
fi

exit 2
```

最後に、`if` 項目を通して、LPD にこのフィルタを教えてやる必要があります。

```
:if=/usr/local/libexec/ifhp :
```

これでおしまいです。`lpr plain.text` とか `lpr whatever.ps` と入力してみましょう。どちらも正常に印字されるはずです。

訳注

日本語を印字する場合は、日本語対応の Ghostscript が必要です。日本語対応版の Ghostscript も Ports Collection に入っています。

10.4.1.4. 変換フィルタ

「プリンタ設定導入編」に書かれた簡単な設定が完了したら、最初に、やってみたいと思うことは、多分 (プレイン ASCII テキストに加えて) 好みのファイル形式のための変換フィルタをインストールすることでしょう。

10.4.1.4.1. なぜ、変換フィルタをインストールするのか?

変換フィルタによって、　様々な種類のファイルを印字することが簡単になります。たとえば、TeX 組版システムでたくさんの仕事をしたと仮定しましょう。そして、PostScript® プリンタが接続 されているとします。 すると、TeX で DVI ファイルを作成する度に、DVI ファイルを印字するために、これを PostScript® ファイルに変換する必要があります。このコマンドは次のようになるでしょう。

```
% dvips seaweed-analysis.dvi
% lpr seaweed-analysis.ps
```

DVI ファイル用の変換フィルタがインストールしてあると、LPD に変換を肩代わりさせることで毎回毎回 おこなわなければならなかった面倒な変換作業を省くことができます。つまり、DVI を生成したら、次のようなコマンドを入力するだけで、これが印字されます。

```
% lpr -d seaweed-analysis.dvi
```

LPD に DVI ファイルの変換をさせるためには、`-d` オプション を指定します。変換オプションのリストは「 整形と変換に関するオプション」に載せてあります。

変化のオプションのそれぞれをプリンタに サポートさせるためには、変換フィルタをインストールし、そのパス名を `/etc/printcap` の中で指定しなくてはなりません。変換フィルタは、プレインテキストを印字する代わりに、フィルタはファイルを プリンタが理解できる形式に変換するところを除けば、「プリンタの簡単な設定」で説明したテキストファイル (「 テキストフィルタのインストール」を見て下さい) に似ています。

10.4.1.4.2. どの変換フィルタをインストールすべきか?

使いたいと思う変換フィルタをインストールすべきです。DVI のデータを頻繁に印字するならば、DVI 変換フィルタ をインストールするのが適切でしょう。印字しなくてはなら ない troff を大量に抱えている場合は、多分、troff フィルタが欲しくなるはずです。

次の表は、LPD で動作するフィルタと、`/etc/printcap` ファイルでのエントリする項目、そして、`lpr` コマンドで呼び出す方法をまとめたものです。

ファイル形式	**/etc/printcap** 項目	**lpr** オプション
cifplot	`cf`	`-c`
DVI	`df`	`-d`
plot	`gf`	`-g`
ditroff	`nf`	`-n`
FORTRAN text	`rf`	`-f`
troff	`tf`	`-f`
raster	`vf`	`-v`
プレインテキスト	`if`	なし、`-p`、または `-l`

先の例のように、`lpr -d` を使うためには、出力先のプリンタの `/etc/printcap` 内のエントリで、`df` 項目が必要であることが分かります。

反論はあるかも知れませんが、FORTRAN テキストや plot のような形式は、多分、廃れてていくでしょう。あなたのサイトで、自前のフィルタをインストールするだけで、プリン

トオプションのいくつか、あるいは、全部に新しい意味を与えることができます。たとえば、Printerleaf ファイル (Interleaf デスクトップパブリッシングプログラムによるファイル) を直接印字したいとします。そして、Printerleaf 用の変換フィルタを gf 項目で 指定したパスにインストールすれば、lpr -g の意味は「Printerleaf ファイルを印字する」意味だとユーザに教えることができます。

10.4.1.4.3. 変換フィルタのインストール

変換フィルタは FreeBSD の基本システムのインストールとは別にインストールするプログラムなので、変換フィルタは、/usr/local ディレクトリの下に置くべきでしょう。フィルタは LPD だけが実行する特別なプログラム、すなわち、一般ユーザが実行する必要すらないプログラムなので、 /usr/local/libexec ディレクトリに置くのが普通です。

変換フィルタを使用可能にするためには、/etc/printcap の目的のプリンタの適切な項目に フィルタがあるパス名を指定します。

DVI 変換フィルタをプリンタ bamboo のエントリに加えてみましょう。プリンタ bamboo の df 項目を新たに加えた /etc/printcap ファイルの例を以下に再掲します。

```
#
#  /etc/printcap for host rose - added df filter for bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan :\
        :lp=/dev/lpt0 :\
        :if=/usr/local/libexec/if-simple :

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo :\
        :lp=/dev/ttyu5 :ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif :\
        :df=/usr/local/libexec/psdf :
```

DVI フィルタは /usr/local/libexec/psdf という 名前のシェルスクリプトです。このスクリプトは次のようになっています。

```
#!/bin/sh
#
#  psdf - DVI to PostScript printer filter
#  Installed in /usr/local/libexec/psdf
#
#  Invoked by lpd when user runs lpr -d
#
exec /usr/local/bin/dvips -f | /usr/local/libexec/lprps "$@"
```

このスクリプトでは、dvips をフィルタモード (引数 -f) で、 標準入力上で起動しています。標準入力は印字するジョブです。 それから、PostScript® プリンタ用フィルタ

`lprps` (これについては「 プレインテキストのジョブを PostScript® プリンタで印字する」を参照してください) を LPD に与えられた引数を付けて起動します。`lprps` はこれらの引数を印字されたページ分の課金をおこなうために使われます。

10.4.1.4.4. 変換フィルタのその他の例

変換フィルタのインストールには決まったステップがないので、この節では、例をもっと挙げることにします。 これを自分でフィルタを作る際のガイドにしてください。 適当な例があったら、それをそのまま使ってください。

次のスクリプト例は、Hewlett Packard LaserJet III-Si のための、raster (ええと‥実は、GIF ファイル) 用の変換フィルタです。

```
#!/bin/sh
#
#  hpvf - Convert GIF files into HP/PCL, then print
#  Installed in /usr/local/libexec/hpvf

PATH=/usr/X11R6/bin:$PATH; export PATH

giftopnm | ppmtopgm | pgmtopbm | pbmtolj -resolution 300 \
    && exit 0 \
    || exit 2
```

ここでは、GIF ファイルから PNM (portable anymap) 形式に変換し、次に PGM (portable graymap) 形式に変換してから、LaserJet/PCL-互換データに変換しています。

上記のフィルタを使うプリンタのためのエントリを付け加えた `/etc/printcap` ファイルは次のようになります。

```
#
#  /etc/printcap for host orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak :mx#0:\
        :if=/usr/local/libexec/hpif :\
        :vf=/usr/local/libexec/hpvf :
```

次のスクリプトは、PostScript® プリンタ `bamboo` のための groff 組版システムの troff データのための変換フィルタです。

```
#!/bin/sh
#
#  pstf - Convert groff's troff data into PS, then print.
#  Installed in /usr/local/libexec/pstf
#
exec grops | /usr/local/libexec/lprps "$@"
```

上記のスクリプトではプリンタとの通信をおこなうため、 lprps をまた利用しています。プリンタがパラレルポートに接続されている場合は、代わりに、 次のスクリプトを使うかもしれません。

```
#!/bin/sh
#
#  pstf - Convert groff's troff data into PS, then print.
#  Installed in /usr/local/libexec/pstf
#
exec grops
```

これで完成しました。次に、フィルタを使用可能にするため に /etc/printcap に加える必要があるエントリを示します。

```
:tf=/usr/local/libexec/pstf :
```

次の例をみたら、FORTRAN のベテランは赤面するかもしれません。この FORTRAN テキストフィルタは、 プレインテキストを直接印字できるすべてのプリンタで利用できます。このフィルタをプリンタ teak にインストールすることにしましょう。

```
#!/bin/sh
#
# hprf - FORTRAN text filter for LaserJet 3si:
# Installed in /usr/local/libexec/hprf
#

printf "\033&k2G" && fpr && printf "\033&l0H" && exit 0
exit 2
```

そして、このフィルタを使用可能にするため、以下の行を /etc/printcap のプリンタ teak のエントリに加えます。

```
:rf=/usr/local/libexec/hprf :
```

これが最後の、そして、若干複雑な例です。前に紹介した LaserJet プリンタ teak に、DVI フィルタを加える ことにしましょう。最初に、 簡単な部分をおこないます。すなわち、DVI フィルタの位置を /etc/printcap に書き加えます。

```
:df=/usr/local/libexec/hpdf :
```

さて、難しい部分であるフィルタの作成をおこないます。 このために、DVI から LaserJet/PCL への変換プログラムが必要です。FreeBSD の Ports Collection (Ports Collection を参照してください) には、それがあります。dvi2xx というのがその port の名前です。これをインストールすると、必要なプログラム dvilj2p が使えます。このプログラムは DVI を LaserJet IIp、LaserJet III、そして LaserJet 2000 の互換コードへ変換してくれます。

dvilj2p はフィルタ hpdf を極めて複雑にしています。 なぜなら、dvilj2p は標準入力からデータを読み込むことができないからです。このプログラムを働かせるために

は、ファイル名が必要です。もっと悪いことに、ファイル名は `.dvi` で終わっている必要があり、標準入力の代わりに、`/dev/fd/0` を使うのは問題があります。この問題は、(`.dvi` で終わる) 一時的なファイル名から`/dev/fd/0` に (シンボリックな) リンクを張る ことで回避することができます。これで、`dvilj2p` に強制的に標準入力からデータを読み込ませることができます。

もう1つの問題は、一時的なリンクを張るために `/tmp` ディレクトリを使うことができないという事実です。シンボリックリンクはユーザ、グループが `bin` であるユーザに所有されています。フィルタはユーザ `daemon` として起動します。そして、`/tmp` ディレクトリはスティッキービットが立っています。フィルタはリンクを作ることができます。しかし、リンクは別のユーザに所有されているため、作業が終了したとき、このリンクを削除することができません。

その代わりに、シンボリックリンクは現在の作業ディレクトリ、すなわち、スプーリングディレクトリ (`/etc/printcap` の `sd` 項目で指定する) に作ることにします。フィルタが作業するにはここの場所は完璧な場所で、なぜなら、特に、スプーリングディレクトリのディスクの空き容量は (ときどき) `/tmp` ディレクトリよりもたくさんあるからです。

以下に示すのが最後のフィルタです。

```
#!/bin/sh
#
#  hpdf - Print DVI data on HP/PCL printer
#  Installed in /usr/local/libexec/hpdf

PATH=/usr/local/bin:$PATH; export PATH

#
#  Define a function to clean up our temporary files.  These exist
#  in the current directory, which will be the spooling directory
#  for the printer.
#
cleanup() {
   rm -f hpdf$$.dvi
}

#
#  Define a function to handle fatal errors: print the given message
#  and exit 2.  Exiting with 2 tells LPD to do not try to reprint the
#  job.
#
fatal() {
    echo "$@" 1>&2
    cleanup
    exit 2
}

#
#  If user removes the job, LPD will send SIGINT, so trap SIGINT
#  (and a few other signals) to clean up after ourselves.
```

```
#
trap cleanup 1 2 15

#
#  Make sure we are not colliding with any existing files.
#
cleanup

#
#  Link the DVI input file to standard input (the file to print).
#
ln -s /dev/fd/0 hpdf$$.dvi || fatal "Cannot symlink /dev/fd/0"

#
#  Make LF = CR+LF
#
printf "\033&k2G" || fatal "Cannot initialize printer"

#
#  Convert and print.  Return value from dvilj2p does not seem to be
#  reliable, so we ignore it.

#
dvilj2p -M1 -q -e- dfhp$$.dvi

#
#  Clean up and exit
#
cleanup
exit 0
```

10.4.1.4.5. 自動変換: その他の変換フィルタ

ここまでに述べてきたフィルタによって、 印字環境の能率が上がったことと思います。しかし、これはどのフィルタを使うかを (lpr(1) のコマンドライン上で) ユーザが指定しなくてはならないという代価を支払って実現されています。 コンピュータの事情にあまり詳しくないユーザにとって、フィルタのオプションを指定させられるということは いらいらさせられるものになるでしょう。更に悪いことに、 間違ったフィルタオプションを指定されると、間違った形式のファイルがそのフィルタに適用されることになり、 その結果、何百枚もの紙を吐き出すことになるかもしれません。

そのような結果になるならば、変換フィルタをインストールするよりもむしろ、テキストフィルタ (これがデフォルトフィルタなので) に印字するよう要求されたファイルの形式を検出させ、自動的に、適切な変換フィルタを起動するようにしたいと思うかもしれません。ここでは `file` コマンドのようなツールを役立たせることができます。 もちろん、いくつかのファイル形式の違いを見分けることは難しいことでしょう。 そして、もちろん、それらのファイルに対しては、変換フィルタを提供するだけで済ますこともできるのです。

FreeBSD Ports Collection には、`apsfilter` (print/apsfilter) と呼ばれる自動変換をおこなうテキストフィルタがあります。 このフィルタは プレインテキス

ト、PostScript®, DVI など、ほとんどすべてのファイル形式を検出し、適当な変換をおこなった後、データを印字することができます。

10.4.1.5. 出力フィルタ

LPD スプーリングシステムでは、ここまでにまだ取り上げていないフィルタ形式、出力フィルタをサポートしています。出力フィルタは、テキストフィルタのように、プレインテキストのみを印字するために意図されたものですが、非常に簡単化されています。テキストフィルタを用いずに、出力フィルタを使っている場合は、次のようになります。

- LPD はジョブ中の各ファイルに一度ではなく、ジョブ全体に対して一度だけ出力フィルタを起動します。
- LPD は出力フィルタに対し、ジョブ中のファイルの先頭や末尾を特定するための対策を 一切おこなっていません。
- LPD はユーザのログイン名やホスト名をフィルタに渡しません。したがって、課金の処理をおこなうことは考えていません。 実際、出力フィルタには、以下2つの引数しか与えられません。

  `filter-name -w`*`width`* `-l`*`length`*

  ここで、*`width`* は対象となるプリンタの `pw` 項目、*`length`* は `pl` 項目に指定された数です。

出力フィルタの簡便さに誘惑されてはいけません。もし、ジョブ中のそれぞれのファイルに別のページ番号を付加しようとしても、出力フィルタはうまく動作しないでしょう。そのような動作を期待しているならば、(入力フィルタとしても知られている) テキストフィルタを使ってください。詳しくは、「テキストフィルタのインストール」をご覧ください。さらに、出力フィルタは、実のところ、もっと複雑になっています。まず、特殊なフラグ文字を検出するために、フィルタに送られてくるバイトストリームを検査する必要があります。また、LPD に代わって、自分自身にシグナルを送らなければなりません。

しかしながら、ヘッダページの印字をおこないたくて、エスケープシーケンスやヘッダページを印字できるようにするその他の初期化文字列を送信する必要がある場合は、 出力ファイルが必要です。(しかし、ヘッダページを要求したユーザに対して課金しようとするのもまた無駄なことです。LPD は出力フィルタにユーザやホストの情報を渡しません)。

1 台のプリンタに対し、LPD では出力フィルタとテキストやその他のフィルタを両方使うことができます。このような場合、LPD はヘッダページ (「 ヘッダページ」を参照してください) だけを印字させるために、出力フィルタを起動させます。それから LPD では、出力フィルタに 2 バイトの文字 (ASCII 031 の次に ASCII 001) を送ることで、出力フィルタが自分自身を停止することを期待しています。2 バイト (031, 001) が出力フィルタに送られたとき、出力フィルタは自分自身にシグナル `SIGSTOP` を送ることによって停止するはずです。LPD がその他のフィルタを動かし終わると、出力フィルタにシグナル `SIGCONT` を送って、出力フィルタを再起動します。

出力フィルタがあり、テキストフィルタがない場合、LPD はプレインテキストジョブを扱う場合に、出力フィルタを使います。前述したように、出力フィルタでは、ジョブ中の各ファイルの間に FORM FEED 文字や紙を送る他の文字を入れることはしません。この動作は多分、あなたが求めているものとは異なっているでしょう。ほとんどの場合において、テキストフィルタが必要なはずです。

プログラム `lpf` は、テキストフィルタの項で既に紹介しましたが、出力フィルタとしても動作させることができます。もし、簡便で極悪な出力フィルタが必要で、かつ、バイトストリームを検査したりシグナルを送るコードを書きたくないときには、`lpf` をお試しください。あるいは、プリントが要求する初期化コードを送るために、`lpf` をシェルスクリプトに包んで使うこともできます。

10.4.1.6. テキストフィルタ `lpf`

プログラム `/usr/libexec/lpr/lpf` は、FreeBSD の バイナリ配布に付属しているテキストフィルタ (入力フィルタ) で、出力を字下げしたり (`lpr -i` でジョブが入力さ れたとき)、文字を未処理のままプリンタに送ったり (`lpr -l` でジョブが入力されたとき)、ジョブ中のバックスペースやタブの印字位置を調節したり、印字したページに対して課金したりすることができます。また、このフィルタは出力フィルタとしても動作させることができます。

`lpf` フィルタは多くの印字環境において使用することに適しています。このフィルタには、プリンタに初期化文字列を送る機能はありませんが、必要とされる初期化をおこない、それから `lpf` を実行させるためのシェルスクリプトを作成するのはたやすいことです。

`lpf` に対して、印字ページへの課金を正確におこなわせるためには、`/etc/printcap` ファイルの中の `pw` と `pl` の項目に正確な値を入れておく必要があります。これらの値は、どのくらいの量のテキストがページにフィットするか、また、ユーザのジョブが何ページあるのかを調べるために使われます。プリンタの課金についての詳しい情報については、「プリンタの利用に対する課金」をご覧ください。

10.4.2. ヘッダページ

あなたが管理するシステムのユーザが たくさんおり、ユーザ全員が様々なプリンタを使用する場合、多分、必要悪であるヘッダページを 印字させることを検討したいと思うかもしれません。

ヘッダページは、バナー とか バーストページ としても知られていますが、出力されたジョブが誰によるものなのかを特定させる働きがあります。印字結果の山の中において、ユーザのジョブによって印字された本物のドキュメント部分よりも際立たせるために、ヘッダページは、通常、多分、縁が装飾されている大きな太文字で印字されます。ヘッダページにより、ユーザは自分が出したジョブがどこにあるのかをすばやく見つけることができます。ヘッダページの欠点は、明らかに、すべてのジョブに対して、紙が 1 枚余分に印字されるということです。この紙の有効期間は短く、2 ～ 3 分も続きません。最終

的に、これらの紙は再利用紙入れの中かくずの山に入れられることでしょう (ヘッダページはジョブ中の各ファイル毎に印字されるのではなく、 ジョブ毎に印字されるということに注意してください。したがって、紙の消費はそれほどひどくはないかもしれません)。

もし、プリンタがプレインテキストを直接印字できるならば、LPD システムは印字物に対して自動的にヘッダページを付けることができます。 PostScript® プリンタを使っている場合は、 ヘッダページを生成する外部プログラムが必要になります。これについては、「PostScript® プリンタでのヘッダページ」をご覧ください。

10.4.2.1. ヘッダページの印字を許可する

「プリンタ設定導入編 」節では、/etc/printcap ファイルの sh (``suppress header'' :「ヘッダを供給しない」という意味) を指定して、ヘッダページの印字を止めていました。プリンタでのヘッダページの印字を許可するには、sh 項目を取り除くだけでよいのです。

とても簡単そうに見えるけど、本当かな?

それは本当です。プリンタに初期化文字列を送るための 出力フィルタを用意しなくてはならないかもしれません。次に、Hewlett Packard PCL 互換プリンタの例を挙げます。

```
#!/bin/sh
#
#  hpof - Output filter for Hewlett Packard PCL-compatible printers
#  Installed in /usr/local/libexec/hpof


printf "\033&k2G" || exit 2
exec /usr/libexec/lpr/lpf
```

of 項目に出力フィルタのパス名を指定してください。 詳細については、「出力フィルタ」節 をご覧ください。

次に、以前紹介したプリンタ teak のための /etc/printcap ファイルの例を示します。ここでは、ヘッダページの印字を許可し、上記の出力フィルタを追加しました。

```
#
#  /etc/printcap for host orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak :mx#0:\
        :if=/usr/local/libexec/hpif :\
        :vf=/usr/local/libexec/hpvf :\
        :of=/usr/local/libexec/hpof :
```

さて、ユーザが teak からジョブを印字させたとき、それぞれのジョブ毎にヘッダページが印字されます。もし、ユーザが印字物を探すのに時間を費やしたいと思うなら、lpr -

h によってジョブを入力することで、ヘッダページの印字を止めることができます。これ以外の lpr(1) のオプションについては、「 ヘッダページ用オプション」節をご覧ください。

注記

LPD では、ヘッダページの最後に、FORM FEED 文字が印字されます。プリンタに紙排出をさせるために、別な文字、もしくは、別な文字列が利用されている場合は、`/etc/printcap` 中の `ff` 項目で指定することができます。

10.4.2.2. ヘッダページを制御する

ヘッダページの印字が許可されていると、LPD は 長いヘッダを作ります。これには、紙全面に大きな文字でユーザ名、ホスト名、 ジョブ名が書かれています。次に、このヘッダページの例を示 します (`kelly` がジョブ名「outline」を `rose` というホストから印字された場合)。

```
k                 ll       ll
k                  l        l
k                  l        l
k   k     eeee     l        l     y    y
k  k     e    e    l        l     y    y
k k      eeeeee    l        l     y    y
kk k     e         l        l     y    y
k   k    e    e    l        l     y   yy
k    k    eeee    lll      lll     yyy y
                                       y
                                      y    y
                                       yyyy


                           ll
                  t         l        i
                  t         l
 oooo    u    u   ttttt     l       ii     n nnn     eeee
o    o   u    u     t       l        i     nn   n   e    e
o    o   u    u     t       l        i     n    n   eeeeee
o    o   u    u     t       l        i     n    n   e
o    o   u   uu     t  t     l        i     n    n   e    e
 oooo     uuu u      tt     lll      iii    n    n    eeee
```

```
      r rrr     oooo     ssss     eeee
      rr   r   o    o   s    s   e    e
      r        o    o    ss      eeeeee
      r        o    o      ss    e
      r        o    o   s    s   e    e
      r         oooo     ssss     eeee




                                              Job:  outline
                                              Date: Sun Sep 17 ↺
11:04:58 1995
```

LPD はこのテキストの終わりに FORM FEED 文字を加えます ので、ジョブは新しいページから開始されます (ただし、`/etc/printcap` で出力先のプリンタのエントリに `sf` (suppress form feeds) が指定されているときはこ の限りではありません)。

お望みならば、LPD に短いヘッダページを出力させることもできます。 この場合は、`/etc/printcap` ファイルの中で `sb` (short banner) を指定してください。ヘッダページは次のようになります。

```
rose:kelly  Job: outline  Date: Sun Sep 17 11:07:51 1995
```

デフォルトでは、LPD はヘッダページを最初に印字し、次にジョブの印字をおこないます。この順番を逆にするときは、`/etc/printcap` で `hl` (header last) を指定してください。

10.4.2.3. ヘッダページに対する課金

LPD に備わっているヘッダページ出力機能を使うと、入力されたジョブに対して課金をおこなうことができても、 ヘッダページは無料で提供しなくてはならない、 という特有のやり方を強要されます。

なぜでしょうか。

出力フィルタは単なる外部プログラムなので、 課金をするための制御をおこなうとすれば、それはヘッダページを印字するときですが、出力フィルタには、ユーザ名とホスト名の情報や課金情報を格納するファイルがどれな のかということが知らされません。それゆえ、出力ファイルには、 誰にプリンタ利用の課金をおこなえばよいのかが分からないのです。テキストフィルタやその他の変換フィルタ (これらのフィルタはユーザやホストの情報が知らされます) が出力ページの枚数に「1 ページ分水増しする」だけでは十分ではありません。なぜなら、ユーザは `lpr -h` に よってヘッダページの出力を止めることがで

きるからです。やみくもに 1 ページを水増しすると、印字されてもいないヘッダページに対する 料金をとることになります。基本的に、`lpr -h` は環境に優しい心を持つユーザに好まれるオプションですが、これを使うように奨励することもできません。

各々のフィルタに独自のヘッダページを生成させる (その結果、ヘッダページに課金することができる) という方法でも十分であるとはいえません。この場合、LPD はフィルタに `-h` の情報を送りませんので、`lpr -h` によってヘッダページを印字しないオプションを選択したとしても、依然としてヘッダページは印字され、その分の課金がおこなわれてしまいます。

では、どのような選択肢があるのでしょうか。

ヘッダページへの課金に関しては、次のことができます。

- LPD のやり方を受け入れ、ヘッダページは無料とする。
- LPRng などの LPD の代替品をインストールする。LPD と入れ替えが可能な他のスプーリングソフトウェアに関しては、標準スプーラの代替品 をご覧ください。
- スマートな 出力フィルタを作成する。通常、 出力フィルタはプリンタを初期化するか、単純な文字列変換をする程度の働きしかしません。(テキスト (入力) フィルタがない場合) 出力フィルタはヘッダページとプレインテキストの印字をおこなうのに適しています。プレインテキストを印字するためのテキストフィルタがない場合、LPD はヘッダページを印字するためだけの目的で出力フィルタを起動します。 そして、LPD が生成するヘッダページのテキストを解析することにより、 出力フィルタはヘッダページに課金するために必要なユーザ名と ホスト名を取得することができます。この方式の唯一の問題点は、 出力フィルタは課金情報を格納するデータファイルの名前を知ることができないということです (`af` 項目で指定されたファイル名は 出力ファイルに渡されません)。しかし、既知の 名前の課金データファイルを使うのならば、その名前を出力フィルタのプログラム中に埋め込むことができます。解析の手順を簡単にするためには、`/etc/printcap` で `sh` 項目 (短いヘッダを指定) を使うとよいでしょう。 そしてまた、ここまでの方法は少なからぬトラブルを生じさせるかもしれません。そうなれば、もちろんユーザはヘッダページを無料で 提供してくれる気前のよいシステム管理者に感謝することでしょう。

10.4.2.4. PostScript® プリンタでのヘッダページ

これまでに述べたように、LPD ではプレインテキストのヘッダページをたくさんのプリンタに合うように生成することができます。 残念ながら、PostScript® プリンタは、 プレインテキストを直接印字することができません。ですから、LPD のヘッダページ機能はまったく、あるいはほとんどの場合、役に立ちません。

ヘッダページを出力するための自明な方法の1つに、 すべての変換フィルタとテキストフィルタにヘッダページを生成させる方法があります。 フィルタは、 適切なヘッダページを生成するために、ユーザ名とホスト名の引数を使うべきです。この方法の欠点は、いつでも、`lpr -h` によってジョブが入力された場合でさえも、ヘッダページが印字されるということです。

この方法で試してみましょう。次のスクリプトは、3 つの引数 (ユーザ のログイン名、ホスト名、ジョブ名) をとり、簡単な PostScript® 用 のヘッダページを生成します。

```
#!/bin/sh
#
#  make-ps-header - make a PostScript header page on stdout
#  Installed in /usr/local/libexec/make-ps-header
#

#
#  These are PostScript units (72 to the inch).  Modify for A4 or
#  whatever size paper you are using:
#
page_width=612
page_height=792
border=72

#
#  Check arguments
#
if [ $# -ne 3 -]; then
    echo "Usage: `basename $0` <user> <host> <job>" 1>&2
    exit 1
fi

#
#  Save these, mostly for readability in the PostScript, below.
#
user=$1
host=$2
job=$3
date=`date`

#
#  Send the PostScript code to stdout.
#
exec cat <<EOF
%!PS

%
%  Make sure we do not interfere with user's job that will follow
%
save

%
%  Make a thick, unpleasant border around the edge of the paper.
%
$border $border moveto
$page_width $border 2 mul sub 0 rlineto
0 $page_height $border 2 mul sub rlineto
currentscreen 3 -1 roll pop 100 3 1 roll setscreen
$border 2 mul $page_width sub 0 rlineto closepath
0.8 setgray 10 setlinewidth stroke 0 setgray
```

```
%
%  Display user's login name, nice and large and prominent
%
/Helvetica-Bold findfont 64 scalefont setfont
$page_width ($user) stringwidth pop sub 2 div $page_height 200 sub ↺
moveto
($user) show

%
%  Now show the boring particulars
%
/Helvetica findfont 14 scalefont setfont
/y 200 def
[ (Job:) (Host:) (Date:) -] {
200 y moveto show /y y 18 sub def
} forall

/Helvetica-Bold findfont 14 scalefont setfont
/y 200 def
[ ($job) ($host) ($date) -] {
        270 y moveto show /y y 18 sub def
} forall

%
%  That is it
%
restore
showpage
EOF
```

そして、変換フィルタやテキストフィルタがそれぞれ、 最初にこのスクリプトを起動することで、ヘッダページが出力され、それから、ユーザのジョブの印字をおこないます。次に、このドキュメントの始めのほうで紹介した DVI 変換フィルタを、ヘッダページを印字するように変更したものを示します。

```
#!/bin/sh
#
#  psdf - DVI to PostScript printer filter
#  Installed in /usr/local/libexec/psdf
#
#  Invoked by lpd when user runs lpr -d
#

orig_args="$@"

fail() {
    echo "$@" 1>&2
    exit 2
}

while getopts "x:y:n:h:" option; do
    case $option in
```

```
        x|y)  -;; # Ignore
        n)    login=$OPTARG -;;
        h)    host=$OPTARG -;;
        *)    echo "LPD started `basename $0` wrong." 1>&2
              exit 2
              -;;
    esac
done

[ "$login" -] || fail "No login name"
[ "$host" -] || fail "No host name"

( /usr/local/libexec/make-ps-header $login $host "DVI File"
  /usr/local/bin/dvips -f ) | eval /usr/local/libexec/lprps ↺
$orig_args
```

このフィルタがユーザ名やホスト名を決定するために 引数リストをどのように解析しなくてはならないかという点に注意してください。この解析方法は他の変換フィルタに対しても同様です。しかしながら、テキストフィルタについては、引数の設定が少し異なっています (これについては、「 フィルタはどのように機能しているか」をご覧ください)。

前述の通り、上記の手法は、極めて単純なのにも関らず、`lpr` で「ヘッダページを印字しない」オプション (`-h` オプション) が使えなくなっています。ユーザが森林資源を (あるいは、 ヘッダページが課金されているならば、その僅かな金額を)、 節約したいと望んでいる場合でも、すべてのフィルタがすべてのジョブ毎にヘッダページを印字 することになっているので、節約することはできません。

ジョブ毎に印字されるヘッダページを ユーザが抑制できるようにするためには、「 ヘッダページに対する課金」で紹介したトリックを 使う必要があります。すなわち、LPD が生成するヘッダページの解析をおこない、PostScript® 版のヘッダページを出力させる出力フィルタを作るのです。この場合、ユーザが `lpr -h` でジョブを入力すると、LPD はヘッダページを生成しなくなり、また、 出力フィルタも起動されません。そうでないならば、作成した出力フィルタが LPD からのテキストを読み込み、ヘッダページを印字する適当な PostScript® のコードがプリンタに送られるでしょう。

PostScript® プリンタがシリアルポートに接続されている場合、 出力フィルタとして `lprps` を、上記の動作をおこなうものとして `psof` を使うことができます。ただし、`psof` はヘッダページに対して課金をおこないませんので注意してください。

10.4.3. リモートプリンタからの出力

FreeBSD では、ネットワーク越しの印字、すなわち、 ジョブをリモートプリンタに送ることをサポートしています。リモートプリンタからの出力をするには、一般に、次の 2 つを参照してください。

- リモートホストに接続されたプリンタにアクセスする方法。 プリンタがあるホストのシリアル、または、パラレルインタフェースに接続されている場合、ネットワーク上の他のホ

ストからこのプリンタにアクセスできるように LPD を設定します。「リモートホストに 接続されたプリンタ」でどのようにするかを説明します。

- ネットワークに直接接続されているプリンタにアクセスする方法。プリンタに、旧来のシリアル、または、 パラレルインタフェースに加えて (もしくは、これらに代わって) ネットワーク用のインタフェースがある場合。そのようなプリンタは次のように動作するでしょう。

  - そのプリンタが LPD のプロトコルを理解でき、リモートホストからのジョブを キューに入れることさえできる場合。この場合、プリンタは、LPD が起動している一般のホストのように振る舞います。そのようなプリンタを設定するために、「 リモートホストに接続されたプリンタ」と同様の手順をおこなってください。

  - そのプリンタが、 データストリームによるネットワーク接続をサポートしている場合。この場合、ネットワーク上の1つのホストとしてプリンタを「接続」します。このホストは、ジョブをスプーリングする責任を負い、スプーリングされたジョブはプリンタに送られます。 そのようなプリンタをインストールするためのいくつかの提案が 「 ネットワークにおけるデータストリームの インタフェースを持つプリンタ」にあります。

10.4.3.1. リモートホストに接続されたプリンタ

LPD スプーリングシステムでは LPD (または LPD 互換のシステム) が起動している他のホストへジョブを送る機能が 始めからサポートされています。この機能により、あるホストに接続されたプリンタへ、 他のホストからアクセスできるようになります。また、LPD プロトコルを理解するネットワークインタフェースを持ったプリンタに対しても、 この機能は働きます。

リモートプリンタへの出力を許可するためには、最初に、 あるホスト (これを、 プリンタホストと呼びます) にプリンタを接続します。そして、「 プリンタ設定導入編」 に書かれた簡単なプリンタの設定をおこなってください。必要ならば、「プリンタ設定上級編」にある、更に進んだ設定をおこなってください。そして、 そのプリンタをテストしてうまく動作することを確認し、LPD に許可した機能がうまく働くかどうかを見てください。さらに ローカルホストが プリンタホストの LPD サービスの使用を許可されているか確認して下さい (「 リモートホストからの利用を制限する 」参照)。

LPD 互換のネットワークインタフェースを持つプリンタを使用している場合は、 そのプリンタ自身が以下で説明する プリンタホストになります。そして、プリンタ名とは、そのプリンタに設定した名前のことを指します。これについては、プリンタ、および (または)、プリンタのネットワークインタフェースに付属するドキュメントを参照してください。

ヒント

ヒューレット・パッカード社の Laserjet シリーズを使用している場合には、プリンタ名を `text` とすると、自動的に LF から CRLF

への変換が行なわれます。そのため、hpif スクリプトは必要ありません。

次に、 そのプリンタにアクセスしたいと思っている他ホストにおいて、 そのホストの /etc/printcap ファイルに次にあげるエントリを作ります。

1. 名前のエントリ。どんな名前でもよいのですが、簡単のため、多分、プリンタホストで設定されたプリンタ名や別名と同じものを使いたいと思うでしょう。

2. lp 項目で指定されるデバイスは明示的に空にします (:lp=: とします)。

3. スプーリングディレクトリを作成し、sd 項目でその位置を指定します。LPD では、プリンタホストにジョブを送信するまでの間、このディレクトリにジョブを格納します。

4. rm 項目でプリンタホストの名前を指定します。

5. rp 項目で プリンタホストに接続したプリンタ名を指定します。

これで終わりです。 変換フィルタやページの大きさやその他の事項を /etc/printcap に加える必要はありません。

次に、 リモートホストに接続されたプリンタで印字するための設定例を示します。 ホスト rose には 2 台のプリンタ bamboo と rattan が接続されています。これらのプリンタをホスト orchid のユーザが使えるようにしましょう。最初に orchid の /etc/printcap を示します (このファイルは、「 ヘッダページの出力を許可する」で参照することができます)。このファイルには、既に、プリンタ teak 用のエントリがありました。以下では、これに、ホスト rose にある2台のプリンタ用のエントリが加えられています。

```
#
#  /etc/printcap for host orchid - added (remote) printers on rose
#

#
#  teak is local; it is connected directly to orchid:
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0 :sd=/var/spool/lpd/teak :mx#0:\
        :if=/usr/local/libexec/ifhp :\
        :vf=/usr/local/libexec/vfhp :\
        :of=/usr/local/libexec/ofhp :

#
#  rattan is connected to rose; send jobs for rattan to rose:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
```

```
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan :

#
#  bamboo is connected to rose as well:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo :
```

orchid で必要となる作業はスプーリングディレクトリを作ることだけです。

```
# mkdir -p /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chown daemon:daemon /var/spool/lpd/rattan /var/spool/
lpd/bamboo
```

これで、orchid のユーザが rattan と bamboo で印字することができるようになりました。たとえば、orchid のユーザが次のように入力したとします。

```
% lpr -P bamboo -d sushi-review.dvi
```

すると、orchid の LPD システムは、ジョブをスプーリングディレクトリ /var/spool/lpd/bamboo にコピーし、これが DVI ファイルを印字するジョブであることを記録します。ホスト rose の bamboo スプーリングディレクトリに十分な容量が確保でき次第、両者の LPD は、ジョブのファイルを rose に転送します。このファイルは、そのすべてが印字されるまで、rose のキューに留まります。(bamboo は PostScript® プリンタなので) DVI から PostScript® への変換は rose でおこなわれます。

10.4.3.2. ネットワークにおけるデータストリームの インタフェースを持つプリンタ

プリンタのネットワークインタフェースカードは、2 種類に分類することができます。1 つはスプーラをエミュレートするもの (高価) で、もう 1 つはシリアルやパラレルポートを使うように プリンタにデータを送ることができるだけのもの (安価) です。この節では、後者の使い方を説明します。前者のプリンタは、前節「 リモートホストに接続されたプリンタ」の方法が適用できます。

/etc/printcap ファイルでは、 シリアルかパラレルのインタフェースのどちらを使うのか、 そして、(シリアルインタフェースを使う場合) そのボーレートはいくらであるか、フロー制御は使うのか、タブのための遅延を加えるのか、改行文字を変換するかなどの指定をおこなうことができます。しかし、TCP/IP や他のネットワークポートからデータを受け取るプリンタを 接続するための指定をおこなうことはでき ません。

ネットワーク接続されたプリンタにデータを送るためには、 テキストフィルタと変換フィルタから呼び出すことができる 通信プログラムを開発する必要があります。以下に、そのようなプログラムの例を示します。スクリプト netprint では、標準入力から印字データをすべて受け取り、ネットワーク接続されたプリンタにこれを送ります。netprint の最初

の引数でプリンタのホスト名を、2 番目の引数で接続するポート番号を指定します。このプログラムでは単方向通信 (FreeBSD からプリンタ) のみをサポートしていることに注意してください。ネットワークプリンタの多くは双方向通信をサポートしていますので、その恩恵 (プリンタの状態を得たり、課金をおこなうなど) にあずかりたいと思われるかもしれません。

```
#!/usr/bin/perl
#
#  netprint - Text filter for printer attached to network
#  Installed in /usr/local/libexec/netprint
#

$#ARGV eq 1 || die "Usage: $0 <printer-hostname> <port-number>";

$printer_host = $ARGV[0];
$printer_port = $ARGV[1];

require 'sys/socket.ph';

($ignore, $ignore, $protocol) = getprotobyname('tcp');
($ignore, $ignore, $ignore, $ignore, $address)
    = gethostbyname($printer_host);

$sockaddr = pack('S n a4 x8', &AF_INET, $printer_port, $address);

socket(PRINTER, &PF_INET, &SOCK_STREAM, $protocol)
    || die "Can't create TCP/IP stream socket: $!";
connect(PRINTER, $sockaddr) || die "Can't contact $printer_host: $!";
while (<STDIN>) { print PRINTER; }
exit 0;
```

このスクリプトは、 様々なフィルタが利用することができます。仮に、Diablo 750-N ラインプリンタを持っており、 これがネットワークに接続されているとしましょう。 プリンタはポート番号 5100 にて印字するデータを受け取ります。 プリンタのホスト名は `scrivener` とします。このとき、このプリンタのテキストフィルタは次のようになります。

```
#!/bin/sh
#
#  diablo-if-net - Text filter for Diablo printer `scrivener' ↺
listening
#  on port 5100.  Installed in /usr/local/libexec/diablo-if-net
#

exec /usr/libexec/lpr/lpf "$@" | /usr/local/libexec/netprint ↺
scrivener 5100
```

10.4.4. プリンタの利用に制約を与える

本節では、プリンタの利用に制約を与えるための情報を記しています。LPD システムでは、プリンタ (ローカル、リモートのいずれに接続されていても) にアクセスできる人を制

限する機能、 複数部のコピーの印字の可否を制御する機能、 ジョブのサイズの最大値やプリンタキューに入る ジョブの最大個数を制御する機能を提供しています。

10.4.4.1. 複数部のコピーの印字を制限する

LPD システムではユーザが複数部のコピーの印字を簡単におこなう 機能を提供しています。ユーザが、(たとえば) `lpr -#5` コマンドを使ってジョブを印字すると、ジョブのそれぞれのファイルのコピーを 5 部得ることができます。これがよい機能であると思うかどうかは人それぞれでしょう。

複数部のコピーの印字によってプリンタが 必要以上に消耗してしまうと感じるならば、`/etc/printcap` ファイルに `sc` 項目を加えてください。これにより、lpr(1) の `-#` オプションの使用が禁止されます。このオプションが指定されているにも関らず、`-#` オプションを使うと、次のようなメッセージが表示され、このオプションの利用できない旨を伝えます。

```
lpr: multiple copies are not allowed
```

リモートホストからプリンタをアクセスできる 設定にしている場合 (この 設定については、「 リモートホストに接続されたプリンタ」 をご覧ください)、そのリモートホストの `/etc/printcap` にも同じように `sc` 項目を追加する必要があることに注意してください。そうしないと、ユーザは別なホストから複数部のコピーの 印字をすることができてしまいます。

例を使って説明しましょう。次に示す `/etc/printcap` ファイルは、ホスト `rose` のものです。プリンタ `rattan` は極めて頑丈なので、 複数部のコピーの印字は許可されています。しかし、レーザプリンタの `bamboo` はもう少しデリケートで、このプリンタから複数部のコピーを印字することを `sc` 項目を追加することで禁止しています。

```
#
#  /etc/printcap for host rose - restrict multiple copies on bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan :\
        :lp=/dev/lpt0 :\
        :if=/usr/local/libexec/if-simple  :

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo :sc:\
        :lp=/dev/ttyu5 :ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif :\
        :df=/usr/local/libexec/psdf :
```

さらに、orchid の `/etc/printcap` にも

sc

項目を追加する必要があります (orchid でこの編集をおこなっているときに、ついでに、プリンタ teak でも複数部のコピーの印字を禁止することにしましょう)。

```
#
#  /etc/printcap for host orchid - no multiple copies for local
#  printer teak or remote printer bamboo

teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0 :sd=/var/spool/lpd/teak :mx#0:sc:\
        :if=/usr/local/libexec/ifhp :\
        :vf=/usr/local/libexec/vfhp :\
        :of=/usr/local/libexec/ofhp :

rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan :

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo :sc:
```

sc 項目を指定することにより、lpr -# の使用を防ぐことができます。しかし、この状態では lpr(1) を複数回起動したり、1 回のジョブで次のように同じファイルを複数個指定することを防ぐまでには至っていません。

```
% lpr forsale.sign forsale.sign forsale.sign forsale.↺
sign forsale.sign
```

このような悪用を防ぐ方法は (その指示を無視することも含めて) たくさんあります。各自で調べてみてください。

10.4.4.2. プリンタを使用できる人を限定する

それぞれのプリンタを使用できる人を限定するには、UNIX® の グループ権限のメカニズムを利用し、さらに、/etc/printcap で rg 項目を指定することでおこないます。あるプリンタにアクセスさせてもよいと思うユーザすべてを グループのどれかに入れてください。そして、そのグループ名を rg で指定します。

このとき、そのグループに含まれないユーザ (root も含みます) がプリントしようとすると、次のようなメッセージが表示されます。

```
lpr: Not a member of the restricted group
```

sc (suppress multiple copies : 複数部のコピーの印字を禁止する) を指定するときと同様に、rg が指定されたプリンタがリモートホストからもアクセスでき (この設定については、「 リモートホストに接続されたプリンタ」をご覧ください)、かつ、そのホストでもプリンタを使用できる人を限定するのが 妥当であると思う場合は、そのホストの /etc/printcap にも rg 指定をおこなう必要があります。

たとえば、プリンタ rattan は誰でも利用できるが、bamboo はグループ artists に属している人のみが利用できるようにしてみましょう。 以下に、もうお馴染みとなったホスト rose の /etc/printcap を示します。

```
#
#  /etc/printcap for host rose - restricted group for bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan :\
        :lp=/dev/lpt0 :\
        :if=/usr/local/libexec/if-simple :

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo :sc:rg=artists:\
        :lp=/dev/ttyu5 :ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif :\
        :df=/usr/local/libexec/psdf :
```

これ以外の /etc/printcap ファイル (ホスト orchid のもの) はそのままにしておくことにします。もちろん、orchid のユーザは全員 bamboo を利用することができます。これは、orchid には特定のユーザのみにしかアクセスさせておらず、そのユーザにはプリンタを利用させたいと思っているからなのかもしれませんし、そうでないかもしれません。

注記

1台のプリンタを複数グループのユーザに利用させることはできません。

10.4.4.3. 入力可能なジョブのサイズを制限する

たくさんのユーザからプリンタが利用される場合には、多分、 ユーザが印字要求を出すことができるファイルのサイズに 上限値を置く必要が生じるでしょう。結局のところ、 スプーリングディレクトリ が置かれているファイルシステムの空き容量がその 上限値になる訳ですが、 あるユーザがこれを独占的に使用すること避けるために、 他ユーザからのジョブ用の空き容量を確保する必要もあります。

LPD では、mx 項目を指定することにより、ジョブ中の個々のファイルのサイズの上限値を制限する機能を提供しています。指定される ファイルサイズの単位は BUFSIZ ブロックで、1 BUFSIZ ブロックは 1024バイトを表わします。この mx 項目の値として 0 が指定されると、ファイルサイズの制限はなくなります。mx が指定されない場合は、デフォルトの制限として 1000 ブロックが使われます。

注記

この制限はジョブ中の各 ファイルに対して適用されるものであり、ジョブ全体のサイズ を制限するものではありません。

ところで、プリンタに設定された上限値を超えるファイルサイズの ファイルが入力された場合でも、LPD はこれを拒否しません。その代わりに、このファイルは、 その先頭から上限値のファイルサイズまでしかキューに入れられません。 そして、その部分までが印字され、残りの部分は捨てられます。これが正しい動作といえるのかどうかは議論の余地があるところです。

それでは、設定例に登場しているプリンタ `rattan` と `bamboo` の印字可能なファイルサイズに制限を加えてみましょう。`artists` グループの人達が作る PostScript® ファイルのサイズは 巨大になる傾向があるので、上限値を 5M バイトとします。それから、プレインテキスト用のラインプリンタは無制限とします。

```
#
#  /etc/printcap for host rose
#

#
#  No limit on job size:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:mx#0:sd=/var/spool/lpd/rattan :\
        :lp=/dev/lpt0 :\
        :if=/usr/local/libexec/if-simple  :

#
#  Limit of five megabytes:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo :sc:rg=artists:mx#5000:\
        :lp=/dev/ttyu5 :ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif :\
        :df=/usr/local/libexec/psdf :
```

この場合もそうですが、この制限はローカル (ホスト rose) のユーザのみに適用されます。 リモートホストからプリンタを利用できるように設定している場合は、 そのリモートホストのユーザはこの制限を受けません。 これらのユーザにも制限を加える場合は、リモートホストの `/etc/printcap` の `mx` を指定する必要があります。リモートホストから印字するための詳しい情報については、「 リモートホストに接続されたプリンタ」を参照してください。

リモートホストに接続されたプリンタへのジョブの サイズを制限する特別な方法は他にもあります。これについては、「 リモートホストからの利用を制限する」を参照してください。

10.4.4.4. リモートホストからの利用を制限する

LPD スプーリングシステムでは、リモートホストから要求されたジョブの印字を制限するための方法がいくつか提供されています。

ホストの制限
: ローカルの LPD が印字要求を受け付けるリモートホストは、ファイル `/etc/hosts.equiv` と `/etc/hosts.lpd` によって制御することができます。LPD では、あるホストから印字の要求がきたとき、 このホストの名前がこれら 2 つのファイルのどちらかに含まれている かどうかを調べます。これが含まれていない場合は、LPD はこの要求を拒否します。

  これらのファイルの形式は単純です。 各行にホストの名前を 1つずつ書いていきます。ファイル `/etc/hosts.equiv` の方は ruserok(3) プロトコルでも利用され、rsh(1) や rcp(1) といったプログラムの動作に影響するので注意が必要です。 `/etc/hosts.equiv` の記述は慎重におこないましょう。

  例として、以下にホスト `rose` の `/etc/hosts.lpd` を示します。

```
orchid
violet
madrigal.fishbaum.de
```

  この例では、`rose` はホスト `orchid`, `violet` そして `madrigal.fishbaum.de` からの要求を受け付けることになります。 その他のホストが `rose` の LPD にアクセスしようとしても、LPD はそのジョブを拒否します (訳注: 拒否されるのは、そのホストが `/etc/hosts.equiv` にも含まれていない場合です)。

サイズの制限
: スプーリングディレクトリがある ファイルシステムに残しておく必要がある 空き容量の大きさを制御することができます。 ローカルプリンタ用のスプーリングディレクトリに `minfree` という名前のファイルを作成します。そして、 そのファイルの中にリモートホストからのジョブの 要求を受け付けるために必要な空き容量のディスクブロックサイズ (1 ディスクブロック = 512 バイト) を記します。

  これで、 リモートホストのユーザにファイルシステムを満杯にされないことが保証されます。 この機能を使うと、 ローカルホストのユーザに対してある種の優先権を与えることもできます。 ローカルホストのユーザは、 `minfree` ファイルで指定された値よりもディスクの空き容量が下回った後でもずっと、 ジョブをキューに入れることができるのです。

たとえば、プリンタ bamboo 用の minfree を作ってみましょう。このプリンタのスプーリングディレクトリを調べるために、/etc/printcap を調べてみましょう。以下に、bamboo のエントリ部分を示します。

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo :sc:rg=artists:mx#5000:\
        :lp=/dev/ttyu5 :ms#-parenb cs8 clocal ↺
crtscts:rw:mx#5000:\
        :if=/usr/local/libexec/psif :\
        :df=/usr/local/libexec/psdf :
```

スプーリングディレクトリは sd 項目で指定されます。LPD がリモートホストからのジョブを受け付けるために必要な ファイルシステムの空き容量を 3M バイト (= 6144 ディスクブロック) にすることにしましょう。

```
# echo 6144 > /var/spool/lpd/bamboo/minfree
```

利用ユーザの制限

/etc/printcap の rs 項目を指定することで、ローカルプリンタを利用できるリモートホストのユーザを制限することができます。ローカルホストに接続されたプリンタ用のエントリに rs 項目が指定されている場合、LPD は、印字を要求したユーザのアカウントと同じログイン名が ローカルホストに登録されている場合に限り、そのジョブを受け付けます。それ以外のジョブを LPD は拒否します。

この機能は、(たとえば) 複数の部署がネットワークを共有しており、 この内のあるユーザが部署の境界を越えて活動している場合には特に有用です。 そのようなユーザに対して、システムのアカウントを与えるだけで、 これらのユーザは自分が所属する部署のシステムから そのシステムに接続されているプリンタを使用することができます。 これらのユーザにはむしろ、 プリンタの使用だけを認め、 その他のコンピュータ資源を利用させたくないときは、 それらのユーザにはホームディレクトリを与えず、 ログインシェルはシェルとしては何の役にも立たない /usr/bin/false などを指定して、これらのユーザのアカウントはプリンタ用の「形式的な」ものとします。

10.4.5. プリンタの利用に対する課金

という訳で、印字するためには料金をとることが必要です。取らない理由などありましょうか。紙やインクにはお金がかかります。そして、プリンタの維持費もかかります。プリンタには可動部分が搭載されており、 これらの部分は壊れやすいという傾向があります。 プリンタや、その利用形態、維持費について調査をし、1 ページ (1 フィート、1 メートルなど) 当たりにかかるコストを調べておいてください。これに基づき、プリンタの利用に対する課金を、実際に、どのように始めればよいのでしょうか。

さて、残念ながら、この部分に関しては LPD スプーリングシステムはほとんど役に立ちません。 課金は使用しているプリンタの種類、印字するもののファイルの形式、 プリンタの利用に対する課金での あなた自身の要求に大きく左右されます。

課金システムを実現するためには、プリンタのテキストフィルタ (プレインテキストのジョブに対して課金するため) と変換フィルタ (その他のファイル形式に対して課金するため) を変更して、印字したページを数えたり、プリンタに印字したページ数を取得するための要求を送る必要があります。 ただし、出力フィルタのみを利用している場合は、 課金をおこなうことができません。フィルタに関しては、「 フィルタ」をご覧ください。

一般に、課金方式には次の 2 つがあります。

- 定期的に課金する方法 はよく利用される方法です。この理由は、 恐らく比較的簡単に実現できるからです。 誰かがジョブを印字する度に、フィルタはそのユーザ名、ホスト名、印字したページ数を課金データファイルに記録します。 毎月、毎学期、毎年、その他お好みの時期に、 各プリンタの課金用ファイルを集め、 それぞれのユーザが印字したページ数を合計して その分の課金をおこないます。 次回の課金期間をデータを 0 にして課金を再開するために、 すべてのログファイルを削除します。

- 利用毎に課金する方法 はあまり利用されていません。これは、 実現するのが比較的難しいからです。この方式では、プリンタを使用したらすぐに、フィルタがユーザにその利用に対する課金をおこないます。 ディスククォータのように、課金作業は瞬時におこなわれます。この方式では、ユーザのアカウントが赤字になる場合に、ユーザが印字をおこなうことを拒否することができます。 また、ユーザに 「プリンタ版 quota」 を調べたり、 調整したりする方法を提供したいと思うかもしれ ません。これを実現するためには、ユーザとその quota を追跡するために、 あるデータベース用のコードが必要となります。

LPD スプーリングシステムでは、 どちらの方式にも簡単に対応できます。(ほとんどの場合は) フィルタを用意しなければならないので、 課金作業のためのコードも用意しなければなりません。 しかし、明るい面もあります。 それは、課金方式に関して、非常に大きな柔軟性が与えられたということです。 たとえば、「定期的に課金する方法」か、「利用毎に課金する方法」のどちらかを選びまず、そして、 どんな情報 (ユーザ名、ホスト名、ジョブのタイプ、印字された頁数、 使用した紙の大きさ、印字をするために要した時間など) をログに記録するかを決めます。 以上のことをおこなうには、上記の情報を保持するために、フィルタを変更しなくてはなりません。

10.4.5.1. 手軽なプリンタ課金方法

FreeBSD には、「定期的に課金する方法」による課金を すぐに設定できるように、2 個のプログラムを添付しています。その内の1つはテキストフィルタ `lpf` で、これについては、「 テキストフィルタ lpf」をご覧ください。もう1つは、 pac(8) で、 これはプリンタの課金データファイルからのエントリを集め、これを合計するプログラムです。

「 フィルタはどのように機能しているか」で述べたように、 LPD ではテキストフィルタや変換フィルタを起動しますが、 そのコマンドラインで使用している課金データファイル

の名前が指定されます。 両フィルタはこの引数を使って、 どの課金データファイルのエントリに書き込めばよいのかを知ることができます。 このファイルの名前は `/etc/printcap` 中の `af` 項目によって指定されます。このファイルが絶対パ スで指定されない場合は、スプーリングディレクトリからの相対パスとして扱われます。

LPD は、紙のページの幅と行数 (`pw` と `pl` 項目で 指定される) を引数として `lpf` を起動します。`lpf` フィルタでは、何ページ印字したかを決定するためにこれらの引数を使用します。ファイルをプリンタに送った後、課金情報を課金データファイルに書き込みます。このファイルは次のようになります。

```
2.00 rose:andy
3.00 rose:kelly
3.00 orchid:mary
5.00 orchid:mary
2.00 orchid:zhang
```

課金データファイルはプリンタ毎に分けて作るべきです。これは、`lpf` にはファイルをロックする機構が組み込まれていないためです。したがって、`lpf` が 2 つ起動されたとき、同じファイルに同時に書き込みをおこなった場合、お互いのエントリを破壊してしまうかもしれません。 課金用ファイルを各プリンタ毎に確実に分けるには、 `/etc/printcap` 中の `af=acct` 項目を使います。そうすれば、それぞれの課金用ファイルがプリンタのスプーリングディレクトリに、`acct` という名称で作成されます。

プリンタの利用に対してユーザに課金する準備ができたら、pac(8) プログラムを実行してください (課金したいプリンタのスプーリングディレクトリに移動した後、`pac` と入力してください)。次のような、ドル中心主義の課金リストが表示されます (訳注: ドル中心主義という表現は、表示がドルで出ることへの著者の皮肉でしょう。セントがあるので小数点以下が表示されますが、この機能も日本では邪魔ですね)。

```
  Login               pages/feet   runs    price
orchid:kelly               5.00    1   $  0.10
orchid:mary               31.00    3   $  0.62
orchid:zhang               9.00    1   $  0.18
rose:andy                  2.00    1   $  0.04
rose:kelly               177.00  104   $  3.54
rose:mary                 87.00   32   $  1.74
rose:root                 26.00   12   $  0.52

total                    337.00  154   $  6.74
```

pac(8) が受け付ける引数には次のようなものがあります。

`-P`*`printer`*
: プリンタ *`printer`* の利用に対する課金リストを作成します。このオプションは、`/etc/printcap` の `af` が絶対パスで指定されていた場合に限り、動作します。

`-c`
: ユーザ名のアルファベット順ではなく、課金額の低い順にリストを並べます。

`-m`
: 課金データファイルにあるホスト名を無視します。このオプションを使用すると、ホスト `alpha` のユーザ `smith` とホスト `gamma` のユーザ `smith` は同一人物として扱われます。このオプションが指定されない場合は、両者は別なユーザとして扱います。

`-p`*`price`*
: `/etc/printcap` の `pc` 項目で指定された値、または、デフォルトの値 (2 セント) に代わり、紙1ページ、または、1フィート当たりの価格を指定します。*`price`* として、浮動小数点数を指定することができます。

`-r`
: リストの並べる順番を逆順にします。

`-s`
: 課金リストを作成し、課金データファイルを削除します。

`name …`
: ユーザ *`names`* に対する課金情報のみを表示します。

pac(8) が生成するデフォルトのリストには、各ホストのユーザ別に印字ページ数が表示されます。 (ユーザがサイト内のすべてのホストを使用できるため) ホスト名の情報が意味を持たない場合、`pac -m` を実行してください。次のようなリストが得られます。

```
  Login               pages/feet   runs    price
andy                        2.00    1   $  0.04
kelly                     182.00  105   $  3.64
mary                      118.00   35   $  2.36
root                       26.00   12   $  0.52
zhang                       9.00    1   $  0.18

total                     337.00  154   $  6.74
```

課金額を決めるために、pac(8) は `/etc/printcap` ファイルの `pc` 項目で指定された値 (デフォルト値は 200、すなわち 1 ページ当たり 2 セント) を使います。この項目で、印字物に課金したい ファと思う 1 ページ当たり、または、1 フィート当たりの価格を 100 分の 1 セント単位で指定します。pac(8) を `-p` オプション付きで起動すると、この値を置き換えることができます。この `-p` オプションで指定する額の単位は、100 分の 1 セント単位ではなく、ドル単位です。たとえば、次の指定では、1 ページ当たりの単価が 1 ドル 50 セントになります。

```
# pac -p1.50
```

このオプションを使うと、実際の課金額を集計することができます。

最後に、`pac -s` を起動すると、課金情報は課金データ累計ファイルに保存されます。このファイルの名前は、プリンタの課金データファイルの後ろに `_sum` を付けたものとな

ります。そして、 課金データファイルは削除されます。次に pac(8) が起動されると、 その時点までの累計金額を得るために、課金データ累計ファイルが読み込まれ、通常の課金データファイルからの情報に加算されます。

10.4.5.2. 印字されたページ数をどのように数えるか?

課金を、リモートホストからの印字でさえも、正確におこなうためには、ジョブで使用された紙が何ページであるかを特定できる必要があります。 このことは、プリンタ利用に対する課金をおこなう上の根本的な問題です。

プレインテキストのジョブの場合、 問題を解決するのはさほど難しくはありません。 ジョブが何行であったかを数え、プリンタがサポートしている紙 1 ページに印字できる最大の行数と比較すればよいのです。 重ね打ちするために利用されるファイル中のバックスペース文字や、 物理的に複数の行に渡る長い論理行に対する取り扱いを忘れずにおこなってください。

(「テキストフィルタ lpf」で紹介した) テキストフィルタ `lpf` では、課金をおこなうときに、これらの取り扱いをおこなってくれます。 課金をおこなうために必要なテキストフィルタを作成している方は、`lpf` のソースコードが参考になるでしょう。

これに対して、他のファイル形式の処理はどのようにすれば よいのでしょうか。

まず、DVI から LaserJet, または、DVI から PostScript® への変換の場合、フィルタが `dvilj` や `dvips` の 出力メッセージを解析することで、何ページ分の変換がおこなわれたかを知ることができます。他のファイル形式とその変換プログラムに関しても、同様のことができるかもしれません。

しかし、この方式には問題点があります。それは、 変換されたページがすべて印字されるとは限らないということです。 たとえば、プリンタが紙詰まりを起こしたり、トナー切れになったり、 はたまた、爆発したりするかもしれません。 そのような状況により印字が途中で中止されたとしても、この方式では、ユーザは全ページ分の料金を課されてしまうのです。

それでは、どのような対策をたてることができるのでしょうか。

正確な 課金をおこなうための唯一の確実な方法は、 何ページ印字したのかを知らせることができるプリンタを入手し、これをシリアルポートかネットワークに接続することです。ほとんどすべての PostScript® プリンタではこの概念がサポートされています。 他のプリンタも同様です (Imagen レーザプリンタをネットワーク接続するなど)。 それぞれのプリンタのフィルタを、ジョブを印字した後で印字ページ数を得るように変更してください。そして、課金情報はここで得られた値のみに 基づいて記録してください。行数を数えたり、エラーが生じやすいファイルの調査は必要とされません。

もちろん、気前よく印字料金をすべて無料にすることもできます。

10.5. プリンタを使う

この節では、FreeBSD で設定したプリンタを使う方法について説明します。 ここでは、ユーザレベルでのコマンドを概説します。

lpr(1)
: 印字をおこないます。

lpq(1)
: プリンタキューを調べます。

lprm(1)
: プリンタキューにあるジョブを削除します。

また、「プリンタの管理」節で説明されている管理者用コマンド lpc(8) もあり、プリンタやそのキューの制御のために用いられています。

lpr(1)、lprm(1)、そして lpq(1) の 3 コマンドは、`-P`*`printer-name`* オプションをとり、これによって、`/etc/printcap` のように操作の対象となる プリンタやキューを指定します。これによって、様々なプリンタに対してジョブを送る、取り消す、調査することができます。 `-P` が使われなかった場合は、これらのコマンドは `PRINTER` 環境変数で指定されたプリンタを使用します。そして、`PRINTER` 環境変数がなかった場合は、これらのコマンドはデフォルトのプリンタ `lp` を使います。

以下では、デフォルトプリンタ という用語が意味するプリンタは、`PRINTER` 環境変数で指定されたプリンタ、もしくは、`PRINTER` 環境変数がない場合は、`lp` という名前のプリンタです。

10.5.1. 印字する

ファイルを印字するためには、次のように入力してください。

```
% lpr filename ...
```

これにより、 入力されたファイルのそれぞれをデフォルトのプリンタ から印字します。ファイル名が与えられなかった場合、 lpr(1) は標準入力から印字するデータを読み込みます。たとえば、次のコマンドにより、ある重要なシステムファイルが印字されます。

```
% lpr /etc/host.conf /etc/hosts.equiv
```

印字させるプリンタを選択するためには、次のように入力します。

```
% lpr -P printer-name filename ...
```

次の例では、プリンタ `rattan` に、 カレントディレクトリにあるファイルの詳細なリストを印字しています。

```
% ls -l | lpr -P rattan
```

上記の lpr(1) コマンドではファイル名の指定がないので、lpr は標準入力から印字するデータ、この場合、ls -l コマンドの出力、を読み込みます。

lpr(1) コマンドでは、出力の整形を制御したり、ファイル変換を適用したり、複数部数のコピーを作成したり、などといた様々な幅広いオプションを受け付けることもできます。詳細については、「 その他の印字オプション」をご覧ください。

10.5.2. ジョブの処理状況を調べる

lpr(1) コマンドを使って印字をする場合、プリントしようと するデータは「プリントジョブ」と呼ばれる箱に一緒に置かれ、これが LPD スプーリングシステムに送られます。プリンタにはそれぞれジョブ用のキューがあり、送られてきたジョブはあなたや他のユーザからの別のジョブと一緒にそのキューで並んで、処理される順番を待ちます。プリンタは到着順にこれらのジョブの印字をおこないます。

デフォルトプリンタのキューの状態を表示するには、 lpq(1) と入力します。プリンタを指定するときは、-P オプションを使います。たとえば、次のコマンド

```
% lpq -P bamboo
```

は、プリンタ bamboo のキューの状態を表示します。この lpq コマンドの出力結果の例を次に示します。

```
bamboo is ready and printing
Rank   Owner    Job  Files                                 Total Size
active kelly    9    /etc/host.conf, /etc/hosts.equiv      88 bytes
2nd    kelly    10   (standard input)                      1635 bytes
3rd    mary     11   ...                                   78519 bytes
```

この例では、bamboo のキューに 3 つのジョブがあることが分かります。 最初のジョブはユーザ kelly からのものであり、「ジョブ番号」9 が割り当てられています。プリンタのすべてのジョブには一意なジョブ番号が付けられています。 ほとんどの場合、このジョブ番号は無視することができますが、ジョブをキャンセルするときにはこの番号が必要になります。このことの詳細については、「ジョブの削除 」をご覧ください。

ジョブ番号 9 のジョブは 2 つのファイルを処理します。すなわち、 lpr(1) のコマンドラインに複数のファイル名が与えられたときは、1つのジョブとして扱われるのです。このジョブは、現在、 アクティブジョブ (「Rank」 の欄の active という後に注目) になっています。これは、プリンタからそのジョブが現在印字されているはずであることを意味しています。 2 番目のジョブでは、 lpr(1) コマンドに標準入力からデータが与えられています。3番目のジョブはユーザ mary から与えられました。このジョブのサイズはとても大きくなっています。 彼女がプリントしようとしたファイルのパス名はここで表示させるには長すぎるため、 lpq(1) コマンドはドットを 3 つだけ表示しています。

lpq(1) からの出力で一番最初の行もまた有益な情報を与えています。この行から、プリンタが現在何をしているか (あるいは、少なくとも LPD がプリンタがしていると思っていること) が分かります。

lpq(1) コマンドは `-l` オプションもサポートしています。これにより、詳しい情報が表示されます。`lpq -l` の実行例を次に示します。

```
waiting for bamboo to become ready (offline ?)
kelly: 1st                              [job 009rose]
       /etc/host.conf                   73 bytes
       /etc/hosts.equiv                 15 bytes

kelly: 2nd                              [job 010rose]
       (standard input)                 1635 bytes

mary: 3rd                               [job 011rose]
      /home/orchid/mary/research/venus/alpha-regio/mapping 78519 ↺
bytes
```

10.5.3. ジョブの削除

印字するようジョブを 送った後で印字を中断したくなったときは、lprm(1) コマンドで、キューの中からそのジョブを削除することができます。 大抵の場合、アクティブジョブでさえも lprm(1) を使って削除することができますが、そのジョブの一部またはすべてが印字されてしまうかもしれません。

デフォルトプリンタへのジョブを削除するためには、最初に、lpq(1) を使ってそのジョブ番号を調べます。すなわち、それから、次のように入力して、ジョブを削除します。

```
% lprm job-number
```

特定のプリンタへのジョブを削除するときは、`-P` オプションを使ってそのプリンタを指定します。たとえば、プリンタ `bamboo` のキューからジョブ番号 10 のジョブを削除するには次のようにします。

```
% lprm -P bamboo 10
```

lprm(1) コマンドには略記法がいくつかあります。

lprm -
: あなたが (デフォルトプリンタへ) 送ったジョブをすべて削除します。

lprm *user*
: ユーザ *user* が (デフォルトプリンタへ) 送ったジョブをすべて削除します。他のユーザのジョブを削除できるのはスーパユーザだけです。あなたは、あなた自身のジョブしか削除することはできません。

lprm
: ジョブ番号もユーザ名もシンボル -も指定されないときは、lprm(1) は現在のアクティブジョブを、そのジョブを送ったのがあなた自身であるときに限り、デフォルトプリンタから削除します。ただし、スーパユーザは任意のアクティブジョブを削除することができます。

上記の略記法をデフォルトプリンタではなく 特定のプリンタに対しておこなうときは、-P オプションでそのプリンタを指定するだけよいのです。たとえば、 プリンタ rattan のキューへあなたが送ったジョブを すべて削除するためには次のようにします。

```
% lprm -P rattan -
```

注記

ネットワーク環境で作業をしている場合、 あるホストから送られたプリンタジョブは、これを送ったホストで lprm(1) を使った場合に限って、これを削除することができます。他のホストで同じプリンタを使えたとしても、このジョブを削除することはできません。次の例では、他ホストからジョブを削除することを試みています。

```
% lpr -P rattan myfile
% rlogin orchid
% lpq -P rattan
Rank   Owner      Job  Files
  Total Size
active seeyan     12    ...
  49123 bytes
2nd    kelly      13   myfile
  12 bytes
% lprm -P rattan 13
rose: Permission denied
% logout
% lprm -P rattan 13
dfA013rose dequeued
cfA013rose dequeued
```

10.5.4. その他の印字オプション

lpr(1) コマンドには、テキストの整形や、図や他のファイル形式の変換、複数部コピーの生成、 ジョブの扱いなどを制御することができます。 この節では、これに関するオプションについて記しています。

10.5.4.1. 整形と変換に関するオプション

以下の lpr(1) 用のオプションはジョブにおける ファイルの整形の制御に関するものです。 このオプションは、ジョブにプレインテキストが含まれない場合や pr(1) ユーティリティを使ってプレインテキストを整形する場合に用いてください。

次の例では、プリンタ `bamboo` に (TeX 組版システムによる) DVI ファイル `fish-report.dvi` を印字しています。

```
% lpr -P bamboo -d fish-report.dvi
```

このオプションは、 ジョブに含まれるすべてのファイルに対して適用されます。 したがって、1 つのジョブに (たとえば) DVI ファイルと ditroff ファイルを混在させることはできません。その代わりに、 ファイルを形式毎に別々のジョブに分け、 それぞれのジョブでその形式用の変換オプションを使って印字してください。

注記

`-p` と `-T` を除くすべてのオプションを使用 するためには、出力先プリンタ用の変換フィルタが必要です。たとえば、 `-d` オプションを使用するには、DVI 用の変換フィルタが必要 です。詳細については、「 変換フィルタ」で説明しています。

`-c`
: cifplot ファイルを印字します。

`-d`
: DVI ファイルを印字します。

`-f`
: FORTRAN プログラムを印字します。

`-g`
: plot のデータを印字します。

`-i` *`number`*
: 出力に対して、*`number`* カラム分の字下げをおこないます。*`number`* が省略されると、8 カラム分字下げされます。このオプションはある変換フィルタと一緒の指定されたときのみに機能します。

注記

`-i` と数字の間に空白を入れてはいけません。

`-l`
制御文字を含む文字通りのテキストデータを印字します。

`-n`
ditroff (device independent troff) データを印字します。

-p
印字する前に pr(1) によってプレインテキストを整形します。詳細については pr(1) をご覧ください。

`-T` ***`title`***
pr(1) コマンドにより生成されるヘッダを、ファイル名の代わりに ***`title`*** とする。このオプションは、`-p` と一緒に使ったときのみ機能する。

`-t`
troff データを印字します。

`-v`
ラスタのデータを印字します。

次の例では、ls(1) のマニュアルを美しく整形したものをデフォルトプリンタで印字しています。

```
% zcat /usr/share/man/man1/ls.1.gz | troff -t -man | lpr
 -t
```

zcat(1) コマンドで ls(1) のマニュアルのソースファイルの圧縮を復元し、これを troff(1) コマンドに渡しています。これによりソースファイルが整形され GNU troff の形式となります。その結果は lpr(1) に渡され、LPD スプーラへジョブの要求が発せられます。lpr(1) には `-t` オプションが使われているため、スプーラでジョブを印字したときに GNU troff の形式から、デフォルトプリンタが解釈できる形式へと変換されます。

10.5.4.2. ジョブに関するオプション

以下のオプションは、lpr(1) によって、そのジョブを特殊な扱いにするよう LPD に指示するためのものです。

-# *copies*

ジョブに含まれるファイルのそれぞれを 1 部だけ印字するのではなく、*copies* 部のコピーを生成させるものです。管理者によっては、 プリンタの消耗を避け、コピー機による複製を奨励するために このオプションの使用が禁止されているかもしれません。これに関しては、「 複数部のコピーの印字を制限する 」をご覧ください。

次の例では、デフォルトプリンタで `parser.c` を 3 部コピーし、次に、`parser.h` を 3 部コピーしています。

```
% lpr -#3 parser.c parser.h
```

-m

印字ジョブが完了した後で、メールを送ります。このオプションを付けると、LPD システムはジョブの処理が終了したときに、あなたのアカウントにメールを送ります。メールのメッセージには、ジョブが正常終了したのか、あるいは、 何か異常があり、(しばしば) その異常が何であったのかが書かれています。

-s

印字ファイルをスプールディレクトリにコピーせず、代わりに、シンボリックリンクを作成するよう指示します。

印字させるジョブのサイズが大きいとき、 このオプションを使うと便利かもしれません。このオプションにより、 スプー ルディレクトリの容量が節約されます (それに、 巨大なジョブのお陰でスプールディレクトリのあるファイルシステムの空き容量がなくなってしまうかもしれません)。 さらに、LPD がいちいちすべてのデータをコピーする必要がなくなりますので、時間の節約にもなります。

ただし、欠点もあります。LPD はオリジナルのファイルを直接参照するので、印字が終了するまでそのファイルを変更したり削除することができません。

注記

リモートのプリンタで印字している場合、 LPD は、結局のところ、 ローカルホストからリモートホストにファイルをコピーする必要があります。 したがって、`-s` オプションはローカルのスプーリングディレクトリの空き容量を節約するだけで、リモート側では節約されません。 それでも、このオプションは有用です。

-r

ジョブに含まれるファイルを、スプーリングディレクトリに ファイルをコピーした後に削除します。もしくは、`-s` オプションと一緒に使われた場合は、印字終了後に削除されます。このオプションの使用には十分注意して下さい。

10.5.4.3. ヘッダページ用オプション

以下のオプションにより、ジョブのヘッダページに通常印字さ れるテキストを lpr(1) に調整させることができます。対象のプリンタからヘッダページが出力されない場合は、これらのオプションは何の効力も持ちません。 ヘッダページの設定に関する情報については、「 ヘッダページ」を参照してください。

-C *text*
: ヘッダページに印字されるホスト名を *text* に置き換えます。なお、ホスト名の場所には、通常、ジョブの要求があったホストの名前が印字されます。

-J *text*
: ヘッダページに印字されるジョブ名を *text* に置き換えます。ジョブ名の場所には、通常、ジョブの最初のファイル名、 または、標準入力からデータが印字されたときは `stdin` が印字されます。

-h
: ヘッダページの出力を禁止します。

注記

サイトによっては、 そのヘッダページの生成方法により、 このオプションの効果が現れないかもしれません。 詳細は、「 ヘッダページ」をご覧ください。

10.5.5. プリンタの管理

プリンタの管理者として、プリンタの設置、設定、 そして、それらのテストをおこなう必要がありました。 lpc(8) コマンドにより、これまでとは別な管理方法がプリンタと対話的におこなわれます。 lpc(8) により、次のことが可能となります。

- プリンタの起動、停止をおこなう。
- キューへの入力の許可、禁止をおこなう。
- それぞれのキューにあるジョブの順番を変更する。

最初に用語に関する注意をしておきます。プリンタが停止しているとは、キューの中にあるどのジョブも印字されることがない状態 を言います。この状態においても、 ユーザはまだジョブの要求をおこなうことができますが、これらのジョブはキューの中で、プリンタがスタートする状態になるまで、 あるいは、キューの内容が削除されるまで待たされることになります。

キューが禁止状態にあると、(`root` 以外の) すべてのユーザがプリンタにジョブを要求することができません。キューが許可状態にある場合は、ジョブの入力が許可されます。

キューが禁止状態にある場合でも、プリンタをスタートす る状態にすることは可能です。この場合は、キューが空になるまで、キュー内のジョブの印字が続けられます。

一般的に、lpc(8) コマンドを使用するには root 権限を持っている必要があります。一般のユーザも lpc(8) コマンドを使うことはできますが、プリンタの状態を取得することとハングしたプリンタ を再スタートすることだけに使用が制限されています。

以下に、 lpc(8) コマンドに関する説明の要約を述べます。 ほとんどのコマンドでは、操作対象となるプリンタを指定するため *printer-name* 引数を与えます。*printer-name* の代わりに all が与えられると、操作は /etc/printcap 内にある全プリンタに対しておこなわれることになります。

abort printer-name
: 現在のジョブをキャンセルし、プリンタを停止させます。 キューが許可状態にある場合は、ユーザはまだジョブを入力することができます。

clean printer-name
: プリンタのスプーリングディレクトリから、ジョブの古いファイルを削除します。状況によって、とりわけ、印字途中でエラーが発生していたり、管理操作が頻発していた場合には、ジョブで作られたファイルを LPD が完全に削除しないことがあります。このコマンドでは、スプーリングディレクトリに入っていないファイルを見つけ出し、それを削除しています。

disable printer-name
: キューに新しいジョブを入れることを禁止します。 プリンタが動作しているときは、キューに残っているジョブの印字は続けられます。ただし、キューが禁止状態にあったとしても、スーパーユーザ (root) は常にジョブを入力することができます。

  このコマンドは、新しいプリンタやフィルタを設置している間に使用すると有用です。すなわち、キューを禁止状態にしておくと、root によるジョブのみが入力されます。そして、他のユーザは、テストが完了し、enable コマンドでキューが再度許可状態になるまで、ジョブの入力はできなくなります。

down printer-name message
: プリンタをダウンさせます。これは、disable をおこなった後で、stop をおこなった場合と等価になります。 *message* は、ユーザが lpq(1) コマンドでプリンタのキューの状態を調べたり、lpc status でプリンタの状態を調べたときに、プリンタの状況として表示されるメッセージです。

enable printer-name
: プリンタのキューを許可状態にします。 ユーザはジョブの入力ができるようになりますが、プリンタがスタートの状態になるまでは、プリンタからは何も印字されません。

help command-name
: *command-name* コマンドのヘルプメッセージを表示します。*command-name* が指定されなかった場合は、利用できるコマンドの要約が表示されます。

`restart printer-name`
: プリンタをスタートさせます。通常のユーザは、 LPD がある異常な状況でハングしたときに限り、 このコマンドを使用することができます。しかし、 `stop` または `down` コマンドにより、 停止状態にあるプリンタをスタートさせることはできません。 `restart` コマンドは、`abort` の後に `start` をおこなったことと同じになります。

`start printer-name`
: プリンタをスタートさせます。プリンタのキューにあるジョブを印字することでしょう。

`stop printer-name`
: プリンタを停止します。プリンタは、現在のジョブを終了させ、そして、キューにあるその他のジョブは印字しません。 プリンタが停止状態にあったとしても、まだ、 許可状態にあるキューに対して、ジョブを送ることができます。

`topq printer-name job-or-username`
: *`printer-name`* のキューに対して、ジョブ番号 *`job`* のジョブ、または、ユーザ *`username`* から送られたジョブを置き換えて、キューの先頭に持ってきます。このコマンドに関しては、*`printer-name`* の代わりに `all` を使用することはできません。

`up printer-name`
: プリンタをアップ状態にします。これの反対のコマンドが `down` です。`start` の次に `enable` をおこなったことと等しくなります。

コマンドラインから上記のコマンドを入力すると、lpc(8) はこれを受け付けます。コマンドが入力されなかった場合は、lpc(8) は対話モードに入り、`exit`、`quit`、または、ファイル終端文字が入力されるまでコマンドの入力ができます。

10.6. 標準スプーラの代替品

このマニュアルを最初から通読されている方ならば、ここまでで、 FreeBSD 付属の LPD スプーリングシステムに関して知っておくべきことすべてを学ばれたことと思います。多分、このシステムにあるたくさんの欠点について認識できたことでしょう。そこから「(FreeBSD 上で動作する) スプーリングシステムには他にどのようなものがあるのか」という疑問が自然と湧いてきます。

LPRng
: 「次世代 LPR」を称するLPRng は、PLP を完全に書き換えたものです。Patrick Powell と Justin Mason (PLP の主要な管理者) が共同で LPRng を作成しました。LPRng の本サイトは `http://www.lprng.org/` です。

CUPS
: CUPS (the Common UNIX Printing System) は、UNIX® ベースのオペレーティングシステムに対して、移植性の高い印刷レイヤを提供します。CUPS は Easy

Software Products によって、すべての UNIX® ベンダとユーザに、標準的な印刷ソリューションを普及するために開発されています。

CUPS は、プリントジョブとキューを管理する基盤として Internet Printing Protocol (IPP) を使っています。機能は限定されますが、ラインプリンタデーモン (LPD)、サーバーメッセージブロック (SMB) や AppSocket (JetDirect とも呼ばれています) プロトコルにも対応しています。CUPS は、UNIX® に現実的なプリント機能を備えるため、ネットワークプリンタの検索、PostScript プリンタ記述言語 (PPD) に基づいた印刷オプションを追加します。

CUPS のメインサイトは `http://www.cups.org/` です

HPLIP
: HPLIP (the HP Linux® Imaging and Printing system) は、HP アプライアンス用に HP が開発した、プリンタ、スキャナ、ファックスへの対応のためのプログラム群です。このプログラムでは、印刷機能において CUPS 印刷システムをバックエンドとして利用しています。

  HPLIP のメインサイトは、`http://hplipopensource.com/hplip-web/index.html` です。

10.7. トラブルシューティング

lptest(1) を使った簡単なテストをおこなった結果、正しい出 力を得られずに、以下に示すような出力が得られるかもしれません。

しばらくしたら出力される、または、紙の全体が出てこない
: プリンタは上で示されたような印字を おこなったのですが、しばらくして止まってしまい、動かなくなってしまいました。印字された結果をプリンタから取り出すためには、プリンタにある PRINT REMAINING ボタン、または、FORM FEED ボタンを押す必要があるようです。

  この場合は、おそらくジョブはプリントをする前に 更にデータが送られてこないか待ち続けているのでしょう。この問題を解決するためには、プリンタに FORM FEED 文字 (あるいは特定の必要な文字コード) を 送るテキストフィルタを使ってください。プリンタ内部に残ったデータをプリンタにすぐに印字させるには、普通はこれで十分です。次のジョブが前のジョブの最終ページの中央の どこかから印字を開始させないためにも、紙の途中で印字のジョブが終了したかどうかを確認するのは有益です。

  シェルスクリプト `/usr/local/libexec/if-simple` を次のように変更して、プリンタへジョブを送信した後に FORM FEED 文字を印字させるようにします。

```
#!/bin/sh
#
```

```
# if-simple - Simple text input filter for lpd
# Installed in /usr/local/libexec/if-simple
#
# Simply copies stdin to stdout.  Ignores all filter arguments.
# Writes a form feed character (\f) after printing job.

/bin/cat && printf "\f" && exit 0
exit 2
```

「階段効果」が現れた
出力された紙には次のように印字されていました。

```
!"#$%&'()*+,-./01234
                "#$%&'()*+,-./012345
                            #$%&'()*+,-./0123456
```

あなたは「階段効果」の新たなる犠牲者になってしまいました。この原因は、改行を表わすべき文字がなんであるか の解釈が混乱していることにあります。UNIX® スタイルのオペレーティングシステムでは、改行文字は ASCII コード 10 の line feed (LF) の 1 文字が使われています。MS-DOS® や OS/2® などは ASCII コード 10 の LF と、ASCII コード 13 の文字 (carriage return または CR) をペアで使います (訳注: Macintosh では CR のみで表現されています)。大抵のプリンタでは、改行を表わすために MS-DOS® の慣習にしたがいます。

FreeBSD で印字する場合、印字したテキストは LF 文字だけ が使われていました。プリンタでは LF 文字を見つけると、紙を 1 行分送り出しました。しかし、次の文字を印字するた めの紙の水平方向の位置は維持されました。すなわち、CR 文字が意味することは、次の文字を印字する位置を紙の左端に動かすことです。

FreeBSD がプリンタに動作をして欲しいと思っている動作を以下に示します。

プリンタが CR を受け取ったとき	CR 動作 (復帰) をおこなう
プリンタが LF を受け取ったとき	CR + LF 動作 (復帰、改行) をおこなう

このように動作させるための方法がいくつかあります。

- これらの文字の解釈を変えるために、 プリンタの設定スイッチかコントロールパネルを操作する方法。どのようにして設定をするかはプリンタのマニュアルを参照してください。

注記

FreeBSD 以外のオペレーティングシステムを切り替えて使う場合、CR と LF 文字の解釈をそのオペレーティングシステムで使われているようにプリンタを 再設定する必要があ

るかもしれません。 以下に示す解決方法のいずれかを 選ぶのがよいかもしれませんね。

- 自動的に LF を CR+LF に変換してくれる FreeBSD 用のシリアルドライバを入手する方法。もちろん、このドライバはプリンタ専用に接続される シリアルポートのみで動作します。この機能を許可するためには、ms# 項目を使い、対象プリンタの /etc/printcap ファイルでonlcr モードを設定します。

- LF 文字の扱いを一時的に変更するための エスケープコード をプリンタに送る方法。 プリンタがサポートしているかもしれないエスケープコード については、プリンタのマニュアルを参照してください。 適切なエスケープコードが見つかったら、最初にそのコードを送り、次にプリントジョブを送信 するようにテキストフィルタを変更してください。

  次に、Hewlett Packard 社の PCL エスケープコードに対応しているプリンタのための テキストフィルタの例を示します。このフィルタでは、プリンタ に LF 文字を LF と CR の2文字として扱わせます。その後に、プリンタにジョブを送ります。最後に、ジョブの最終ページの紙を排出するため、FROM FEED 文字を送ります。このフィルタは Hewlett Packard 社のほとんどすべてのプリンタで機能するはずです。

```
#!/bin/sh
#
# hpif - Simple text input filter for lpd for HP-PCL based ↺
printers
# Installed in /usr/local/libexec/hpif
#
# Simply copies stdin to stdout.  Ignores all filter arguments.
# Tells printer to treat LF as CR+LF.  Ejects the page when ↺
done.

printf "\033&k2G" && cat && printf "\f" && exit 0
exit 2
```

  ホスト orchid の /etc/printcap の例を以下に示します。ここには、一番目のパラレルポートにプリンタ (Hewlett Packard LaserJet 3Si) が一台接続されており、そのプリンタ名は teak です。

```
#
#  /etc/printcap for host orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak :mx#0:\
        :if=/usr/local/libexec/hpif :
```

.

訳注

LF を CR+LF に置き換える cat コマンドを作る方法も当然考えられます。そして、このコマンドと、if-simple の cat の部分を置き換えればよいわけです。 具体的にどのようにするかは、読者への練習問題としましょう。

各行が重ね書きされてしまった
: プリンタは紙送りをまったくしませんでした。 テキストすべての行がある行の上で重ねて印字されてしまいました。

  この問題は、 階段現象とは 「正反対」 な問題で、 ほとんどまれにしか起こりません。FreeBSD では行末として扱われる LF 文字が、紙の左端に印字位置を復帰しますが、紙送りはしない CR 文字として扱われています。

  プリンタの設定スイッチかコントロールパネルを使って、LF と CR の文字を次のような解釈をするようにしてください。

プリンタが受け取ったとき	プリンタがおこなう
CR	CR 動作 (復帰)
LF	CR + LF (復帰、改行)

訳注

LF を CR+LF に置き換える cat コマンドを作る方法も当然考えられます。 そして、このコマンドと、 `if-simple` の cat の部分を置き換えればよいわけです。 具体的にどのようにするかは、読者への練習問題としましょう。

プリンタが文字を紛失してしまう
: 印字しているのですが、各行の 2 ～ 3 文字が印字されません。プリンタを動かせば動かすほど、もっとたくさんの文字が紛失されていき、この問題は更に悪くなっていくかもしれませんでした。

  この問題は、シリアルポートを通してコンピュータから送られてくるデータの速度に、プリンタがついていけないことに起因します (この問題は、パラレルポートに接続された プリンタでは発生することはありません)。この問題を克服する方法が2つあります。

- プリンタが XON/XOFF のフロー制御をサポート している場合は、項目 `ms#` で `ixon` モードをセットして、FreeBSD にこの機能を使用させてください。

- プリンタが Request to Send / Clear to Send ハードウェアハンドシェイク (通称 `RTS/CTS`) をサポートして いる場合は、項目 `ms#` で `crtscts` モードをセットして下さい。それから、プリンタとコンピュータを接続しているシリアルケーブルが ハードウェアフロー制御用に正しく配線されたものかどうかを確認してください。

プリンタは意味不明な文字列を印字した
: プリンタはランダムなゴミのように 見えるものを印字しましたが、意図したテキストは印字してくれませんでした。

  この問題は、通常、シリアルポートに接続したプリンタでの 通信パラメータの誤りからくる前項とは別の症状です。`br` 項目の通信速度と `ms#` 項目を再確認してください。また、プリンタでの設定が `/etc/printcap` ファイルで設定した 内容と一致しているかどうかも確認してください。

訳注

simple-if のような単純なフィルタだけの状態で、日本語を含むテキストを印字しようとした場合にも、 シリアルポート、パラレルポートの使用に関係なく、 このような症状は見られます。日本語プリンタの場合、 漢字コードそのもの を送信しただけでその漢字を印字してくれるものは、少なくとも訳者は見たことがありません。 漢字を印字するための制御 コードを別途送信するフィルタが必要となります。 また、そのようなフィルタを使用していても、 そのフィルタが想定してる漢字コードと異なった文書を プリントしようとしたときもこのような症状は出ます。 もちろん、これはプリンタ用の 言語を持たないプリンタの話で、PostScript® プリンタ などにプレインテキストを送信しても、日本語対応、 非対応に関らず、意味不明な文字列が印字される (もしくは、何も印字されない) ことでしょう。

何も起きない
: もしプリンタが何の動作もしないのであれば、 ハード的な問題ではなく、多分 FreeBSD の中に問題があります。`/etc/printcap` ファイルで、デバッグしているプリンタのエントリに (`lf` 項目で) ログファイルを取るように 設定を追加してください。たとえば、プリンタ `rattan` 用のエントリの項目 `lf` は次のようになります。

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan :\
```

```
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple :\
        :lf=/var/log/rattan.log
```

次に、もう一度印字をおこなってみます。そして、 発生したと思われるエラーメッセージを見るためにログファイル (上記の例では、/var/log/rattan.log) を調べます。そこで見られたメッセージを元に、問題を解決してみてください。

項目 lf が指定されていない場合、LPD はデフォルトのログファイルとして /dev/console を使います。

第11章 Linux バイナリ互換機能

再構成と一部の更新: Jim Mock.
オリジナルの文書を執筆: Brian N. Handy 、Rich Murphey.

訳: はらだ きろう <kiroh@jp.FreeBSD.org >、1996 年 9 月 24 日

11.1. この章では

FreeBSD は、Linux を含む Unix-like なオペレーティングシステムとのバイナリ互換機能を提供しています。現時点では、一体なぜ FreeBSD が Linux バイナリを実行できるようにならなければならないのか自問しているのではないでしょうか? 答えはきわめて簡単です。 Linux は現在コンピュータの世界では最も「ホットなモノ」なのでたくさんの会社や開発者たちが Linux のためだけに開発を行なっています。そのため、残された私たち FreeBSD ユーザは彼らに対して FreeBSD ネイティブなアプリケーションも出すように言うしかないのです。問題は、FreeBSD バージョンも出した場合にどれくらいの数のユーザーが使うのかわからない、ということであり、そのため Linux 版のみを開発しているということなのです。そこで FreeBSD では Linux バイナリ互換機能が役に立つのです。

簡単に言ってしまえば、この機能により全ての Linux アプリケーションの 90% が修正なしに FreeBSD 上で起動できます。 この中には Star Office や Linux 版の Netscape、Adobe Acrobat、 RealPlayer 5 と 7、 VMWare、Oracle、WordPerfect、Doom、Quake などがあります。また、ある状況においては Linux バイナリを Linux で動かすよりも FreeBSD で動かすほうが良いパフォーマンスが出るという報告もあります。

しかしながら、いくつかの Linux に特有な OS の機能は FreeBSD ではサポートされていません。例えば、Linux の `/proc` ファイルシステムを過度に使うような Linux バイナリは FreeBSD では動きません (FreeBSD の `/proc` ファイルシステムとは異なるのです) し、仮想 8086 モードを有効にするような i386 特有の呼び出しも動きません。

この章を読むと、以下のことがわかります。

- Linux バイナリ互換機能を有効にする方法。
- Linux 共有ライブラリを追加する方法。
- Linux アプリケーションを FreeBSD システムにインストールする方法
- FreeBSD における Linux 互換機能の実装の詳細。

この章を読む前に、以下のことを理解しておく必要があります。

- サードパーティ製ソフトウェアのインストール方法 (5章アプリケーションのインストール - packages と ports)

11.2. インストール

Linux バイナリ互換機能は、デフォルトでは有効ではありません。この機能を有効にする最も簡単な方法は、`linux` KLD オブジェクト (「Kernel LoaDable object」) を読み込むことです。コマンドプロンプトで `linux` と打つだけでモジュールを読み込むことができます。

Linux 互換機能を常に有効にする場合には、`/etc/rc.conf` に以下の行を追加してください。

```
linux_enable="YES"
```

この設定により、`/etc/rc.i386` では次のような操作が行なわれます。

```
# Start the Linux binary compatibility if requested.
#
case ${linux_enable} in
[Yy][Ee][Ss])
 echo -n ' linux'; linux > /dev/null 2>&1
 ;;
esac
```

望みの KLD モジュールがロードされているか確認したい時には kldstat(8) コマンドを利用します。

```
% kldstat
Id Refs Address    Size     Name
 1    2 0xc0100000 16bdb8   kernel
 7    1 0xc24db000 d000     linux.ko
```

何らかの理由で Linux KLD をロードしたくない、あるいはロードできないような場合には、`options LINUX` をカーネルの設定ファイルに指定して、Linux バイナリ互換機能をカーネルにスタティックリンクしてください。そして、9章FreeBSD カーネルのコンフィグレーション の記述にしたがって新しいカーネルをインストールしてください。

11.2.1. Linux ランタイムライブラリのインストール

これは、linux_base port を用いるか、もしくは 手動で インストールします。

11.2.1.1. linux_base の port を用いたインストール

ランタイムライブラリをインストールするには最も簡単な方法です。Ports Collection から他の port をインストールするのと全く同じようにできます。

```
# cd /usr/ports/emulators/linux_base
# make install distclean
```

これで Linux バイナリ互換機能が使えるはずです。いくつかのプログラムはシステムライブラリのマイナーバージョンが違うと文句を言うかもしれませんが一般的には大した問題ではありません。

11.2.1.2. 手動でのライブラリのインストール

「ports」コレクションをインストールしていない場合、代わりに手動でライブラリをインストールすることができます。プログラムが必要とする Linux のシェアードライブラリとランタイムリンカが必要です。また Linux ライブラリ用の「shadow root」ディレクトリ、`/compat/linux` を作成する必要があります。FreeBSD で動作する Linux プログラムが使用するシェアードライブラリは、まずこのファイルツリーから検索されます。例えば、Linux のプログラムが `/lib/libc.so` をロードしようとした場合には、FreeBSD はまず `/compat/linux/lib/libc.so` を開こうとします。これが存在しなかった場合には、次に `/lib/libc.so` を試します。シェアードライブラリは、Linux の `ld.so` が報告するパスではなく、`/compat/linux/lib` 以下にインストールする必要があります。

Linux のプログラムが必要とするシェアードライブラリを探す必要があるのは、FreeBSD のシステムに Linux のプログラムをインストールする最初の数回だけでしょう。それが過ぎれば、十分な Linux のシェアードライブラリがシステムにインストールされ、新しくインストールした Linux のバイナリも余計な作業をせずに動作させることができるようになります。

11.2.1.3. シェアードライブラリの追加

`linux_base` port をインストールした後に、アプリケーションが必要なライブラリが存在しないというエラーを出したらどうしたらよいでしょうか? Linux のバイナリがどのシェアードライブラリを必要とし、そしてどこで入手できるか、どのように探したらよいでしょうか? 基本的には、以下の 2 種類の方法があります (以下の手順に従う場合には、必要なインストール作業をおこなう FreeBSD システム上で root として作業をおこなう必要があります)。

Linux システムにアクセス可能ならば、そのアプリケーションがどういうシェアードライブラリを必要としているのか調べ、単に FreeBSD にそのライブラリをコピーするだけです。次の例を見てみましょう。

FTP を使って Doom の Linux バイナリを取ってきて、アクセスできる Linux システムに置いたとしましょう。次のように `ldd linuxdoom` とするだけでどのシェアードライブラリが必要かチェックできます。

```
% ldd linuxxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

最後のカラムに表示されているすべてのファイルを持って来て、/compat/linux の下に置き、 最初のカラムに示されるファイル名にシンボリックリンクを張ります。 すなわち、FreeBSD システムでは以下のようなファイルが必要となります。

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29 /compat/linux/lib/libc.so.4 -> ↺
libc.so.4.6.29
```

注記

最初のカラムに表示されているファイルとメジャーバージョンが同じ Linux シェアードライブラリを既にインストールしている場合は、 新たにコピーする 必要はありません。既にあるライブラリで動作するはずです。 ただ、新しいバージョンのものをコピーすることをお奨めします。 新しいライブラリにシンボリックリンクを変更したら、古いライブラリは削除してかまいません。

```
/compat/linux/lib/libc.so.4.6.27
/compat/linux/lib/libc.so.4 -> libc.↺
so.4.6.27
```

従って、以上のようなライブラリがインストールされており、新しいバイナリに対する ldd の出力が以下のようになる場合を考えます。

```
libc.so.4 (DLL Jump 4.5pl26) -> ↺
libc.so.4.6.29
```

このように最後の番号が 1 つか 2 つ古いだけならば、普通は /lib/libc.so.4.6.29 をコピーする必要はありません。わずかに古いライブラリでもプログラムは動作するはずだからです。もちろん、以下のように新しいライブラリと置き換えても構いません。

```
/compat/linux/lib/libc.so.4.6.29
```

```
/compat/linux/lib/libc.so.4 -> libc.↺
so.4.6.29
```

注記

シンボリックリンクのメカニズムは Linux バイナリにのみ必要なことに注意してください。FreeBSD のランタイムリンカはメジャーリビジョン番号の一致したライブラリを検索するので、ユーザが気にする必要はありません。

11.2.2. Linux の ELF バイナリのインストール

ELF のバイナリを使うためには、「マークをつける (branding)」 作業が必要になります。マークのない ELF バイナリを実行しようとすると以下のようなエラーメッセージを受けとってしまうことでしょう。

```
% ./my-linux-elf-binary
ELF binary type not known
Abort
```

カーネルが FreeBSD の ELF バイナリと Linux のバイナリとを 見分けられるようにするためには、brandelf(1) ユーティリティを以下のようにして使ってください。

```
% brandelf -t Linux my-linux-elf-binary
```

今では GNU のツールたちが ELF バイナリに自動的に適切なマークを付加するようになったので、今後はこの作業もだんだんと必要なくなってゆくでしょう。

11.2.3. ホストネームリゾルバの設定

DNS がうまく動作しなかったり、以下のようなエラーメッセージが表示され る場合は、`/compat/linux/etc/host.conf` ファイルを設定する必要があります。

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

ファイルの内容を以下のように設定してください。

```
order hosts, bind
multi on
```

ここで、order は `/etc/hosts` を最初に検索し、次に DNS を検索するように指定します。`/compat/linux/etc/host.conf` がインストールされていない場合、Linux

アプリケーションは FreeBSD の `/etc/host.conf` を使用しようとして、文法の違いによる警告を出力します。`/etc/resolv.conf` を利用してネームサーバの設定をしていない場合には、`bind` を削除してください。

11.3. Mathematica のインストール

Mathematica 4.X 用に Murray Stokely がアップデートし、Bojan Bistrovic **<bojanb@physics.odu.edu >** がマージしました。

この章では、Mathematica 4.X Linux 版の FreeBSD へのインストールについて説明します。

Linux 版の Mathematica は FreeBSD においても完璧に動きます。ただ、実行する際に Linux ABI を用いる必要があることを FreeBSD に教えるために、Wolfram によって出荷されているバイナリにマーク付け (branded) をする必要があります。

Mathematica や Mathematica for Students の Linux 版は Wolfram (http://www.wolfram.com/) から直接注文することができます。

11.3.1. Linux バイナリへのマーク付け (branding)

Linux 用バイナリは Wolfram の Mathematica CD-ROM の `Unix` ディレクトリにあります。 インストーラーを起動する前にこのディレクトリをローカルディスクにコピーし、brandelf(1) により Linux バイナリにマークを付けます。

```
# mount /cdrom
# cp -rp /cdrom/Unix/ /localdir/
# brandelf -t Linux /localdir/Files/SystemFiles/Kernel/
Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/
FrontEnd/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/
Installation/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/
Graphics/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/
Converters/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/
LicenseManager/Binaries/Linux/mathlm
# cd /localdir/Installers/Linux/
# ./MathInstaller
```

また以下のようにすると、マーク付けされていない ELF バイナリすべての扱いを、デフォルトで Linux バイナリとすることが可能です。

```
# sysctl -w kern.fallback_elf_brand=3
```

これは FreeBSD システムに対して、マーク付けされていない ELF バイナリが Linux ABI を利用するように設定します。こうすることで、CDROM から直接インストーラを実行することが可能になります。

11.3.2. Mathematica パスワードの取得

Mathematica を起動する前に Wolfram から自分の「マシン ID」に対応したパスワードを取得しなければいけません。

一旦 Linux 互換ランタイムライブラリをインストールし、Mathematica を展開すれば Install ディレクトリにある `mathinfo` プログラムを起動して「マシン ID」を得ることができます。このマシン ID は、最初に見つかったイーサネットカードの MAC アドレスをベースに生成されます。

```
# cd /localdir/Files/SystemFiles/Installation/Binaries/
Linux
# mathinfo
disco.example.com 7115-70839-20412
```

電子メールや電話、FAX などで Wolfram に登録する時にはこの「マシン ID」を渡します。 するといくつかの数字から構成されるパスワードが返されるので、 他の Mathematica プラットホームでするのと全く同じように最初に Mathematica を立ち上げる時にその情報を入力します。

11.3.3. ネットワーク経由での Mathematica フロントエンドの起動

Mathematica は標準フォントセットにない特別な記号 (積分記号、総和記号、ギリシャ文字など) を表示するために特殊なフォントを使用します。 X プロトコルは、これらのフォントが ローカルマシンに インストールされていることを要求します。 これはつまり、ローカルマシンに (CD-ROM や Mathematica がインストールされているホストマシンから) そのフォントをコピーしなければならないということです。 これらのフォントは通常、CD-ROM の `/cdrom/Unix/Files/SystemFiles/Fonts` か、もしくはハードディスクの `/usr/local/mathematica/SystemFiles/Fonts` に置かれており、実際に使用されるフォントは `Type1` と `X` のサブディレクトリに格納されています。これらを利用するには次のような二つ方法があります。

一つは、フォントファイルをすべて `/usr/X11R6/lib/X11/fonts/` 以下にある既存のフォントディレクトリにコピーする方法です。この場合、`fonts.dir` にフォント名を追加し、先頭行のフォント総数を変更することも必要になります。あるいは、フォントをコピーしたディレクトリで `mkfontdir` を実行するだけでもかまいません。

もう一つの方法は、`/usr/X11R6/lib/X11/fonts/` にフォントディレクトリごとコピーする方法です。

```
# cd /usr/X11R6/lib/X11/fonts
# mkdir X
# mkdir MathType1
# cd /cdrom/Unix/Files/SystemFiles/Fonts
# cp X/* /usr/X11R6/lib/X11/fonts/X
# cp Type1/* /usr/X11R6/lib/X11/fonts/MathType1
# cd /usr/X11R6/lib/X11/fonts/X
# mkfontdir
# cd ../MathType1
# mkfontdir
```

そして、フォントパスに新しいフォントディレクトリを追加します。

```
# xset fp+ /usr/X11R6/lib/X11/fonts/X
# xset fp+ /usr/X11R6/lib/X11/fonts/MathType1
# xset fp rehash
```

XFree86 サーバを使用しているなら、`/etc/XF86Config` に加えることでこれらのフォントを自動的に読み込むことができます。

`/usr/X11R6/lib/X11/fonts/Type1` という ディレクトリが存在していない場合には、上記例の `MathType1` を `Type1` とすることができます。

11.4. Oracle のインストール

##: Marcel Moolenaar.

11.4.1. はじめに

このドキュメントでは Oracle 8.0.5 と Oracle 8.0.5.1 Enterprise Edition の Linux 版を FreeBSD にインストールするための手順を解説します。

11.4.2. Linux 環境のインストール

まずは Ports Collection から `linux_base` と `linux_devtools` をインストールしてください。これらの ports は FreeBSD 3.2 のリリース後にコレクションに加えられました。もし FreeBSD 3.2 もしくはそれよりも古いものを使っている場合は ports コレクションをアップデートしましょう。ついでに FreeBSD をアップデートするのもいいでしょう。もし `linux_base-6.1` や `linux_devtools-6.1` でうまくいかなければ 5.2 を試してみてください。

もし賢いエージェント (intelligent agent) を起動したいなら Red Hat Tcl パッケージ `tcl-8.0.3-20.i386.rpm` もインストールする必要があるでしょう。公式の RPM パッケージをインストールするには一般的に次のようにします。

```
# rpm -i --ignoreos --root /compat/linux --dbpath /var/
lib/rpm package
```

パッケージのインストール時にエラーが出てはいけません。

11.4.3. Oracle 環境の構築

Oracleをインストールする前に、適切な環境を設定する必要があります。 このドキュメントでは、 Oracle のインストールガイドに書いてあるようなことではなく FreeBSD で Linux 用 Oracle を動かすために特別に必要なことのみを解説します。

11.4.3.1. カーネルのチューニング

Oracle インストールガイドにあるように、 シェアードメモリーの最大サイズを設定しなければいけません。FreeBSD では SHMMAX を使わないようにしてください。SHMMAX は単に SHMMAXPGS と PGSIZE から計算されるだけなのです。 従って、SHMMAXPGS を使うようにしましょう。インストールガイドに記述されている他のオプションは使えます。例えば以下のようにします。

```
options SHMMAXPGS=10000
options SHMMNI=100
options SHMSEG=10
options SEMMNS=200
options SEMMNI=70
options SEMMSL=61
```

これらのオプションを意図した Oracle の使い方に合わせて設定してください。

また、 次のオプションがカーネルのコンフィギュレーションファイルにあることも確認します。

```
options SYSVSHM #SysV shared memory
options SYSVSEM #SysV semaphores
options SYSVMSG #SysV interprocess communication
```

11.4.3.2. Oracle 用アカウント

他のアカウントを作るのと同じように Oracle 用のアカウントを作ります。Oracle 用のアカウントに特別なのは Linux のシェルを割り当てるところだけです。/etc/shells に /compat/linux/bin/bash を加え、Oracle 用のアカウントに設定します。

11.4.3.3. 環境設定

ORACLE_HOME や ORACLE_SID といった通常の Oracle 用の変数の他に次の変数も設定しなければなりません。

変数	値
LD_LIBRARY_PATH	$ORACLE_HOME/lib

変数	値
CLASSPATH	$ORACLE_HOME/jdbc/lib/classes111.zip
PATH	/compat/linux/bin /compat/linux/sbin /compat/linux/usr/bin /compat/linux/usr/sbin /bin /sbin /usr/bin /usr/sbin /usr/local/bin $ORACLE_HOME/bin

全ての環境変数は .profile で設定することをお勧めします。完璧なサンプルは以下の通りです。

```
ORACLE_BASE=/oracle; export ORACLE_BASE
ORACLE_HOME=/oracle; export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=386x; export ORACLE_TERM
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export CLASSPATH
PATH=/compat/linux/bin:/compat/linux/sbin:/compat/linux/usr/bin:/
compat/linux/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:
$ORACLE_HOME/bin
export PATH
```

11.4.4. Oracle のインストール

インストーラーを起動する前に、/var/tmp に .oracle という名前のディレクトリを作る必要がありますが、これは Linux エミュレーターにおけるちょっとした不整合のためです。このディレクトリは誰でもが書けるか、もしくは oracle ユーザーのものにしておきます。これで特に問題なく Oracle がインストールできるでしょう。もし問題が起こったら、まずは Oracle の配布物や設定をチェックしてください。Oracle のインストールが終わったら次の二つのサブセクションで解説するパッチを当てます。

よくあるトラブルは、TCP プロトコルアダプターが正しくインストールされていないことです。そのため、一切 TCP リスナーを起動することができないのです。次の操作はこの問題を解決するのに役立ちます。

```
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk ntcontab.o
# cd $ORACLE_HOME/lib
# ar r libnetwork.a ntcontab.o
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk install
```

もう一度 root.sh を起動するのを忘れないように!

11.4.4.1. root.sh へのパッチ

Oracle をインストールする時、root で行なう必要のあるいくつかの操作は root.sh と呼ばれるシェルスクリプトに記録されます。root.sh は orainst ディレクトリにあります。次のパッチを root.sh に当てて 正しい場所にある chown コマンドを使うようにするか、代わりに Linux ネイティブなシェルのもとでスクリプトを走らせましょう。

```
*** orainst/root.sh.orig Tue Oct 6 21:57:33 1998
--- orainst/root.sh Mon Dec 28 15:58:53 1998
***************
*** 31,37 ****
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/bin/chown
#
# Define variables to be used in this script
--- 31,37 ----
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/usr/sbin/chown
#
# Define variables to be used in this script
```

CD-ROM からのインストールでない場合は root.sh のソースにパッチを当ててもいいでしょう。rthd.sh という名前でソースツリーの orainst というディレクトリにあります。

11.4.4.2. genclntsh へのパッチ

genclntsh スクリプトは一つの共有クライアントライブラリを生成するのに用いられます。 これはデモを作る時に使われます。 PATH の定義をコメントアウトするために次のパッチを当ててください。

```
*** bin/genclntsh.orig Wed Sep 30 07:37:19 1998
--- bin/genclntsh Tue Dec 22 15:36:49 1998
***************
*** 32,38 ****
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
--- 32,38 ----
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
```

```
! #PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
```

11.4.5. Oracle の起動

インストラクションに従えば、Linux でと同じように Oracle を起動できるでしょう。

11.5. SAP R/3 (4.6B - IDES) のインストール

##: Holger Kill.
SGML ####: Valentino Vaschetto.

11.5.1. まえがき

この文書は、SAP R/3 4.6B IDES-System with Oracle 8.0.5 for Linux を FreeBSD 4.3 上にインストールする方法を、FreeBSD 4.3-STABLE と Oracle 8.0.5 のインストール方法と一緒に説明しています。

この文書では、重要な手順をすべて詳しく説明しようとしていますが、 Oracle および SAP R/3 のインストールの手引きを置き換えるものではありません。

SAP や Oracle に特有の質問については、SAP R/3 の Linux 版についてくる文書や Oracle と SAP OSS が出している資料をご覧ください。

11.5.2. ソフトウェア

SAP のインストールには以下の CDROM を使いました。

名称	番号	説明
KERNEL	51009113	SAP Kernel Oracle / Installation / AIX, Linux, Solaris
RDBMS	51007558	Oracle / RDBMS 8.0.5.X / Linux
EXPORT1	51010208	IDES / DB-Export / Disc 1 of 6
EXPORT2	51010209	IDES / DB-Export / Disc 2 of 6
EXPORT3	51010210	IDES / DB-Export / Disc3 of 6
EXPORT4	51010211	IDES / DB-Export / Disc4 of 6

名称	番号	説明
EXPORT5	51010212	IDES / DB-Export / Disc5 of 6
EXPORT6	51010213	IDES / DB-Export / Disc6 of 6

更に、Oracle 8 Server (Pre-production バージョン 8.0.5 for Linux, カーネルバージョン 2.0.33) の CD も使いましたが、これは必ずしも必要ではありません。また、もちろん FreeBSD 4.3-STABLE (4.3-RELEASE から間もない時期のものです) も利用しました。

11.5.3. SAP ノート

以下のノートは、SAP R/3 をインストールする前に読んでおくべき、もしくはインストールの際に役立つことが分かっているものです。

番号	題名
0171356	SAP Software auf Linux: grundlegenden Anmerkungen
0201147	INST: 4.6C R/3 Inst. on UNIX - Oracle
0373203	Update / Migration Oracle 8.0.5 --> 8.0.6/8.1.6 LINUX
0072984	Release of Digital UNIX 4.0B for Oracle
0130581	R3SETUP step DIPGNTAB terminates
0144978	Your system has not been installed correctly
0162266	Questions and tips for R3SETUP on Windows NT / W2K

11.5.4. ハードウェア要件

以下の機器であれば SAP R/3 System (4.6B) を動かすのに十分です。

構成部品	4.6B	4.6C
プロセッサ	2 × 800MHz Pentium III	2 × 800MHz Pentium III
メモリ	1GB ECC	2GB ECC
ハードディスク容量	50-60GB (IDES)	50-60GB (IDES)

業務で利用するなら、大容量キャッシュメモリを備えた Xeon プロセッサ、高速なディスク (SCSI, ハードウェア RAID コントローラ)、USV および ECC-RAM を推奨します。ハードディスク容量が大きいのは、設定済みの IDES システムが、インストール時に 27 GB のデータベースファイルを作成するためです。通常、インストール後には一部のテーブル領域を拡張する必要があります。

私は、800MHz の Pentium III を 2 個、Adaptec 29160 Ultra160 SCSI アダプタ (40/80 GB DLT テープドライブと CDROM にアクセスするため)、Mylex AcelleRAID (2 チャンネル、ファームウェアバージョン 6.00-1-00、32MB RAM) を積んだデュアルプロセッサボードを使いました。Mylex RAID コントローラには 17GB のハードディスク 2 台 (ミラー) と 36GB のハードディスク (RAID 5) 4 台が接続されています。

11.5.5. FreeBSD 4.3-STABLE のインストール

まず、FreeBSD 4.3-STABLE をインストールしました。FTP 経由のデフォルトインストールを行いました。

11.5.5.1. FTP 経由のインストール

ディスクイメージ kern.flp と mfsroot.flp を入手して、フロッピーディスクに書き込んでください (私は ftp7.de.freebsd.org から入手しました。適切なミラーサイトを選んでください)。

```
# dd if=kern.flp of=/dev/fd0
# dd if=mfsroot.flp of=/dev/fd0
```

2 つのイメージファイルそれぞれについて、 別のディスクを使うようにしてください。それから kern.flp を書き込んだフロッピーディスクで起動して、指示に従ってください。私は、次のようにディスクを分割しました。

ファイルシステム	容量 (1kB ブロック)	容量 (GB)	マウントポイント
/dev/da0s1a	1.016.303	1	/
/dev/da0s1b		6	<swap>
/dev/da0s1e	2.032.623	2	/var
/dev/da0s1f	8.205.339	8	/usr
/dev/da1s1e	45.734.361	45	/compat/linux/oracle
/dev/da1s1f	2.032.623	2	/compat/linux/sapmnt
/dev/da1s1g	2.032.623	2	/compat/linux/usr/sap

はじめに、Mylex のソフトウェアで論理ドライブ 2 台を設定して初期化しなければなりませんでした。このソフトウェアはボードに組み込まれていて、PC 起動時に動かせます。

このディスクの割り当て方は SAP の推奨と少し異なることに注意してください。SAP は oracle (その他) サブディレクトリを別途マウントするように提案していますが、私は簡便であることを優先して、実際にサブディレクトリとして作成するだけにしました。

11.5.5.2. 最新の STABLE のソースコードの入手

FreeBSD 4.3-STABLE 以降では、最新のソースコードは楽に手に入ります。それより前のバージョンの FreeBSD では、/etc/cvsup に自分で書いたスクリプトを置いていました。FreeBSD 4.3 で CVSup を設定するのは簡単です。`root` で以下を実行してください。

```
# cp /etc/defaults/make.conf /etc/make.conf
# vi /etc/make.conf
```

`/etc/make.conf` ファイルで、以下の項目を有効にしておかねばなりません。

```
SUP_UPDATE=    yes
SUP=           /usr/local/bin/cvsup
SUPFLAGS=      -g -L 2
SUPHOST=       cvsup8.FreeBSD.org
SUPFILE=       /usr/share/examples/cvsup/stable-supfile
PORTSSUPFILE=  /usr/share/examples/cvsup/ports-supfile
DOCSUPFILE=    /usr/share/examples/cvsup/doc-supfile
```

SUPHOST の値は適切なものに変更してください。`/usr/share/examples/cvsup` にある supfile で十分でしょう。もしもすべてのドキュメントを持ってくるのが嫌なら、DOCSUPFILE の項は無効なままにしておいてください。ここまでくれば、cvsup を起動して stable の最新のソースコードを入手するのはたやすいことです。

```
# cd /usr/src
# make update
```

11.5.5.3. `make world` と新しいカーネル

最初にすべきことは、ソースコードのインストールです。root ユーザで次の作業を行ってください。

```
# cd /usr/src
# make world
```

最後まで実行できたら、新しいカーネルの作成と設定を行えます。通常はここでカーネルコンフィグレーションファイルをカスタマイズします。コンピュータに `troubadix` という名前をつけているので、コンフィグレーションファイルを `troubadix` という名称にするのが自然です。

```
# cd /usr/src/sys/i386/conf
# cp GENERIC TROUBADIX
# vi TROUBADIX
```

この段階で、どのドライバを使うか、使わないかといったことを決定できます。詳しい説明は、適当な文書や LINT ファイルをご覧ください。

また、後述するパラメータをいれることもできます。新しいカーネルを作成するには次のようにしてください。

```
# cd /usr/src/sys/i386/conf
# config TROUBADIX
# cd /usr/src/sys/compile/TROUBADIX
# make depend
# make
# make install
```

make install が無事終わったら、コンピュータを再起動して新しいカーネルを動かしてください。

11.5.6. Linux 環境のインストール

(2001 年 5 月 2 日時点の 4.3-STABLE 用に) 必要な RPM ファイルをダウンロードするのに多少問題があったので、以下の場所も (他がすべてだめで、ここが古くなってしまっていなければ) 試してみるとよいかもしれません。

- ftp7.de.freebsd.org/pub/FreeBSD/distfiles/rpm
- ftp.redhat.com/pub/redhat/linux/6.1/en/os/i386/RedHat/RPMS

11.5.6.1. Linux ベースシステムのインストール

まず、Linux のベースシステムを (root で) インストールしなければなりません。

```
# cd /usr/ports/emulators/linux_base
# make package
```

11.5.6.2. Linux 開発環境のインストール

次に、必要なら Linux 開発環境をインストールします。

```
# cd /usr/ports/devel/linux_devtools
# make package
```

11.5.6.3. 必要な RPM のインストール

R3SETUP プログラムを動かすには、pam サポートが必要になります。これが更に他の package を必要としているので、私は結局いくつもの package をインストールすることになりました。それでも、pam は足りない package があると文句をいうので、強制的にインストールしたら動きました。pam 以外にインストールした package が本当に必要なのか、pam package をインストールするのに十分だったのかよくわかっていません。

なんにせよ、これが私がインストールした package 一覧です。

- cracklib-2.7-5.i386.rpm
- cracklib-dicts-2.7-5.i386.rpm
- pwdb-0.60-1.i386.rpm
- pam-0.68-7.i386.rpm

これらの package は、以下のコマンドでインストールしました。

```
# rpm -i --ignoreos --root /compat/linux --dbpath /var/
lib/rpm <package_name>
```

ただし、pam パッケージは次のようにして強制的にインストールしました。

```
# rpm -i --ignoreos --nodeps --root /compat/linux --
dbpath /var/lib/rpm pam-0.68-7.i386.rpm
```

Oracle が intelligent agent を動作させるには、次の RedHat Tcl package を (FreeBSD ハンドブックにあるように) インストールしなければなりませんでした。`tcl-8.0.5-30.i386.rpm` (そうしないと、Oracle インストール中の再リンクがうまくゆきません)。Oracle の再リンクについては他にも問題がありますが、私が理解している限りでは Oracle-Linux の問題で、FreeBSD に特有のものではありません。

11.5.7. SAP/R3 環境の構築

11.5.7.1. 必要となるファイルシステムとマウントポイントの作成

単純なインストールでは、次のファイルシステムを作成すれば事足ります。

マウントポイント	サイズ (GB)
/compat/linux/oracle	45 GB
/compat/linux/sapmnt	2 GB
/compat/linux/usr/sap	2 GB

また、私は FreeBSD が正しいパスを見付けられるように、いくつかリンクを作成しました。

```
# ln -s /compat/linux/oracle /oracle
# ln -s /compat/linux/sapmnt /sapmnt
# ln -s /compat/linux/usr/sap /usr/sap
```

11.5.7.2. ユーザとディレクトリの作成

SAP R/3 には、2 名のユーザと 3 つのグループが必要です。ユーザ名は 3 文字からなる SAP システム ID (SID) に準じています。SID の中には SAP が予約しているもの

があります (たとえば SAP や NIX です。すべて網羅した一覧は SAP のドキュメントをご覧ください)。IDES のインストールでは、IDS を使いました。したがって、次のグループができます (グループ ID は異なっていてもかまいません。これは単に私がインストールした時に使用した値です)。

グループ ID	グループ名	説明
100	dba	データベース管理者
101	sapsys	SAP システム
102	oper	データベースオペレータ

デフォルトの Oracle のインストールでは、`dba` グループだけが使われます。`oper` グループとして、`dba` を使う人もいます。(詳しくは Oracle と SAP のドキュメントを参照してください)。

以下のユーザも必要になります。

ユーザ ID	ユーザ名	一般名称	グループ	追加グループ	説明
1000	idsadm	<sid>adm	sapsys	oper	SAP 管理者
1002	oraids	ora<sid>	dba	oper	DB 管理者

`adduser` でユーザを追加するには、 以下の項目が必要になります (シェルとホームディレクトリを控えてください)。SAP 管理者については、

```
Name: idsadm            <sid>adm
Password: ******
Fullname: SAP IDES Administrator
Uid: 1000
Gid: 101 (sapsys)
Class:
Groups: sapsys dba
HOME: /home/idsadm      /home/<sid>adm
Shell: /bin/sh
```

そしてデータベース管理者については、

```
Name: oraids          ora<sid>
Password: ******
Fullname: Oracle IDES Administrator
Uid: 1002
Gid: 100 (dba)
Class:
Groups: dba
HOME: /oracle/IDS     /oracle/<sid>
Shell: /bin/sh
```

あなたが dba と oper の両方のグループを使っていたら、ここには oper グループもいれるべきです。

11.5.7.3. ディレクトリ作成

以下のディレクトリは、普通は別のファイルシステムとして作成します。どうするかは、あなたの要件に応じて決めてください。結局はすべてが同じ RAID 5 におかれるので、私は単なるディレクトリにしました。

まず、一部のディレクトリに (root ユーザで) 所有者と権限を設定しましょう。

```
# chmod 775 /oracle
# chmod 777 /sapmnt
# chown root:dba /oracle
# chown idsadm:sapsys /compat/linux/usr/sap
# chmow 775 /compat/linux/usr/sap
```

第二に、ユーザ ora<sid> 権限でディレクトリを作成します。すべて /oracle/IDS のサブディレクトリになります。

```
# su - oraids
# mkdir mirrlogA mirrlogB origlogA origlogB
# mkdir sapdata1 sapdata2 sapdata3 sapdata4 sapdata5 ↺
sapdata6
# mkdir saparch sapreorg
# exit
```

第三段階では、idsadm (<sid>adm) ユーザ権限でディレクトリを作成します。

```
# su - idsadm
# cd /usr/sap
# mkdir IDS
# mkdir trans
# exit
```

11.5.7.4. /etc/services 内の項目

SAP R/3 は、/etc/services にいくつか項目を必要としますが、FreeBSD にインストールする場合は正しく設定されません。次の項目を追加してください (少なくとも、インスタンス番号に対応する項目が必要で、この場合は 00 です。dp, gw, sp および ms について 00 から 99 まで項目を追加してもなんの問題もないでしょう)。

```
sapdp00  3200/tcp # SAP Dispatcher.      3200 + ########
sapgw00  3300/tcp # SAP Gateway.         3300 + ########
sapsp00  3400/tcp #                      3400 + ########
sapms00  3500/tcp #                      3500 + ########
sapmsIDS 3600/tcp # SAP Message Server.  3600 + ########
```

11.5.7.5. 必要となるロケール

SAP は、デフォルトの RedHat のインストールでは入らないロケールを少なくとも 2 つ必要とします。SAP は、必要な RPM を FTP サーバからダウンロードする形で提供しています (OSS アクセス契約している顧客でなければアクセスできません)。必要な RPM の一覧は note 0171356 をご覧ください。

適切なリンクを (たとえば de_DE と en_US から) 作成するという手もありますが、業務用システムでそうすることはお勧めしません (いまのところ、IDES システムで問題なく動いてはいますが)。必要なのは、次のロケールです。

```
de_DE.ISO-8859-1
en_US.ISO-8859-1
```

このロケールがないと、インストール時になんらかの問題が起きるでしょう。それを (たとえば CENTRDB.R3S ファイルで問題の段階を OK にして) 無視してしまうと、追加でなにかしないと SAP システムにログオンできないでしょう。

11.5.7.6. カーネルのチューニング

SAP R/3 システムには多くのリソースが必要です。そのため、私はカーネルコンフィグレーションファイルに 以下のパラメータを追加しました。

```
# Set these for memory pigs (SAP and Oracle):
options MAXDSIZ="(1024*1024*1024)"
options DFLDSIZ="(1024*1024*1024)" # System V options needed.
options SYSVSHM #SYSV-style shared memory
options SHMMAXPGS=262144 #max amount of shared mem. pages
options SHMMNI=256 #max number of shared memory ident if.
options SHMSEG=100 #max shared mem.segs per process
options SYSVMSG #SYSV-style message queues
options MSGSEG=32767 #max num. of mes.segments in system
options MSGSSZ=32 #size of msg-seg. MUST be power of 2
options MSGMNB=65535 #max char. per message queue
options MSGTQL=2046 #max amount of msgs in system
options SYSVSEM #SYSV-style semaphores
options SEMMNU=256 #number of semaphore UNDO structures
options SEMMNS=1024 #number of semaphores in system
options SEMMNI=520 #number of semaphore indentifiers
options SEMUME=100 #number of UNDO keys
```

SAP についてくる文書に最小値が指定されています。Linux に関する記述はないので、詳しいことは HP-UX (32-bit) の節を参照してください。

11.5.8. SAP R/3 のインストール

11.5.8.1. SAP CDROM の準備

インストールの際にマウントしてアンマウントしなければならない CDROM が山ほどあります。十分な数の CDROM ドライブを持っていれば、すべてマウントしてしまえばよいで

しょう。私は、CDROM の中身をそれぞれに対応するディレクトリにコピーすることにしました。

```
/oracle/IDS/sapreorg/<cd-name>
```

ここで、<cd-name> は KERNEL, RDBMS, EXPORT1, EXPORT2, EXPORT3, EXPORT4, EXPORT5 および EXPORT6 のいずれかです。すべてのファイル名を大文字にするか、さもなければ、マウントする際に -g オプションをつけるべきです。次のコマンドを使ってください。

```
# mount_cd9660 -g /dev/cd0a /mnt
# cp -R /mnt/* /oracle/IDS/sapreorg/<cd-name>
# umount /mnt
```

11.5.8.2. インストールスクリプトの実行

まず、インストールディレクトリを用意しなければなりません。

```
# cd /oracle/IDS/sapreorg
# mkdir install
# cd install
```

その後、インストールスクリプトを起動すると、 関連するファイルがほぼすべてインストールディレクトリにコピーされます。

```
/oracle/IDS/sapreorg/KERNEL/UNIX/INSTTOOL.SH
```

これは、完全にカスタマイズされた SAP R/3 デモシステム付きの IDES のインストールなので、EXPORT CD は 3 枚ではなく 6 枚あります。この時点では、インストールテンプレート CENTRDB.R3S は標準の中心インスタンス (R/3 とデータベース) をインストールするためのもので、 IDES 中心インスタンスをインストールするためのものではありません。ですから、それに対応する CENTRDB.R3S ファイルを EXPORT1 ディレクトリからコピーしてください。さもないと、R3SETUP は EXPORT CD を 3 枚しか要求しません。

11.5.8.3. R3SETUP の起動

LD_LIBRARY_PATH が正しく設定されていることを確認してください。

```
# export LD_LIBRARY_PATH=/oracle/IDS/lib:/sapmnt/IDS/
exe:/oracle/805_32/lib
```

インストールディレクトリから root ユーザで R3SETUP を起動してください。

```
# cd /oracle/IDS/sapreorg/install
# ./R3SETUP -f CENTRDB.R3S
```

スクリプトがいくつか質問してきます (デフォルトが括弧に囲まれて表示され、その後に実際に入力します)。

質問	デフォルト	入力
Enter SAP System ID (SAP システム ID を入力してください)	[C11]	IDS<ret>
Enter SAP Instance Number (SAP インスタンス番号を入力してください)	[00]	<ret>
Enter SAPMOUNT Directory (SAPMOUNT ディレクトリを入力してください)	[/sapmnt]	<ret>
Enter name of SAP central host (SAP 中心ホストの名称を入力してください)	[troubadix.domain.de]	<ret>
Enter name of SAP db host (SAP データベースホストの名称を入力してください)	[troubadix]	<ret>
Select character set (文字集合を選択してください)	[1] (WE8DEC)	<ret>
Enter Oracle server version (Oracle サーバのバージョンを入力してください) (1) Oracle 8.0.5, (2) Oracle 8.0.6, (3) Oracle 8.1.5, (4) Oracle 8.1.6		1<ret>
Extract Oracle Client archive (Oracle クライアントアーカイブを展開しますか)	[1] (はい、展開します)	<ret>
Enter path to KERNEL CD (KERNEL CD へのパスを入力してください)	[/sapcd]	/oracle/IDS/sapreorg/KERNEL
Enter path to RDBMS CD (RDBMS CD へのパスを入力してください)	[/sapcd]	/oracle/IDS/sapreorg/RDBMS

質問	デフォルト	入力
Enter path to EXPORT1 CD (EXPORT1 CD へのパスを入力してください)	[/sapcd]	/oracle/IDS/sapreorg/EXPORT1
Directory to copy EXPORT1 CD (EXPORT1 CD をコピーするディレクトリ)	[/oracle/IDS/sapreorg/CD4_DIR]	<ret>
Enter path to EXPORT2 CD (EXPORT2 CD へのパスを入力してください)	[/sapcd]	/oracle/IDS/sapreorg/EXPORT2
Directory to copy EXPORT2 CD (EXPORT2 CD をコピーするディレクトリ)	[/oracle/IDS/sapreorg/CD5_DIR]	<ret>
Enter path to EXPORT3 CD (EXPORT3 CD へのパスを入力してください)	[/sapcd]	/oracle/IDS/sapreorg/EXPORT3
Directory to copy EXPORT3 CD (EXPORT3 CD をコピーするディレクトリ)	[/oracle/IDS/sapreorg/CD6_DIR]	<ret>
Enter path to EXPORT4 CD (EXPORT4 CD へのパスを入力してください)	[/sapcd]	/oracle/IDS/sapreorg/EXPORT4
Directory to copy EXPORT4 CD (EXPORT4 CD をコピーするディレクトリ)	[/oracle/IDS/sapreorg/CD7_DIR]	<ret>
Enter path to EXPORT5 CD (EXPORT5 CD へのパスを入力してください)	[/sapcd]	/oracle/IDS/sapreorg/EXPORT5
Directory to copy EXPORT5 CD (EXPORT5 CD をコピーするディレクトリ)	[/oracle/IDS/sapreorg/CD8_DIR]	<ret>
Enter path to EXPORT6 CD (EXPORT6 CD へのパスを入力してください)	[/sapcd]	/oracle/IDS/sapreorg/EXPORT6

質問	デフォルト	入力
Directory to copy EXPORT6 CD (EXPORT6 CD をコピーするディレクトリ)	[/oracle/IDS/sapreorg/CD9_DIR]	<ret>
Enter amount of RAM for SAP + DB (SAP + DB に割り当てる RAM の合計を入力してください)		850<ret> (MB)
Service Entry Message Server	[3600]	<ret>
Enter Group-ID of sapsys (sapsys のグループ ID を入力してください)	[101]	<ret>
Enter Group-ID of oper (oper のグループ ID を入力してください)	[102]	<ret>
Enter Group-ID of dba (dba のグループ ID を入力してください)	[100]	<ret>
Enter User-ID of <sid>adm (<sid>adm のユーザ ID を入力してください)	[1000]	<ret>
Enter User-ID of ora<sid> (ora<sid> のユーザ ID を入力してください)	[1002]	<ret>
Number of parallel procs (並列プロセスの数)	[2]	<ret>

CD をそれぞれ別の場所にコピーしないと、SAP インストーラは (CD 内の LABEL.ASC ファイルで識別される) 必要な CD を見つけられず、CD を挿入 / マウントして承認するかマウントパスを入力するよう求めてくるでしょう。

CENTRDB.R3S ファイルには、間違いがないわけではありません。 私の場合は、EXPORT4 を再度要求されました (が、正しいキーを表示していたので (6_LOCATION、それから 7_LOCATION など))、正しい値を入力して進めて問題ありませんでした。いらいらしないようにしてください。

後で触れるいくつかの問題を除けば、Oracle データベースソフトウェアをインストールするところまでは 素直に進むはずです。

11.5.9. Oracle 8.0.5 のインストール

Linux と Oracle DB で起こりうる問題についての SAP-Notes と Oracle の Readme を読むようにしてください。すべてではないにしても、ほとんどは互換性のないライブラリに起因しています。

Oracle のインストールについて詳しいことは、Oracle のインストール の章をご覧ください。

11.5.9.1. orainst による Oracle 8.0.5 のインストール

Oracle 8.0.5 を利用する予定なら、無事に再リンクするのにライブラリがいくつか追加で必要です。なぜなら、Oracle 8.0.5 は古い glibc (RedHat 6.0) とリンクされていますが、RedHat 6.1 は新しい glibc を利用しているからです。ですから、リンクが行われることを保証するには、次の追加パッケージをインストールしなければなりません。

```
compat-libs-5.2-2.i386.rpm

compat-glibc-5.2-2.0.7.2.i386.rpm

compat-egcs-5.2-1.0.3a.1.i386.rpm

compat-egcs-c++-5.2-1.0.3a.1.i386.rpm

compat-binutils-5.2-2.9.1.0.23.1.i386.rpm
```

詳しくは、対応する SAP-Note や Oracle Readme をご覧ください。他に選択肢がなければ (私がインストールした時は、確認している暇がありませんでした)、元のバイナリまたは再リンクされたバイナリを、元の RedHat システムから持ってきて使うこともできるでしょう。

インテリジェントエージェントをコンパイルするために、RedHat Tcl パッケージがインストールされていなければなりません。`tcl-8.0.3-20.i386.rpm` が入手できないなら、RedHat 6.1 用の `tcl-8.0.5-30.i386.rpm` など新しいものでも構いません。

再リンク以外は、インストールは簡単です。

```
# su - oraids
# export TERM=xterm
# export ORACLE_TERM=xterm
# export ORACLE_HOME=/oracle/IDS
# cd /ORACLE_HOME/orainst_sap
# ./orainst
```

ソフトウェアのインストールが終わるまで、すべての画面で Enter を押して承認してください。ただし、Oracle On-Line Text Viewer は現時点では Linux 版がないので選択を外してください。その後、Oracle は利用可能な `gcc`, `egcs` や `i386-redhat-linux-gcc` ではなく、`i386-glibc20-linux-gcc` で再リンクしようとします。

時間が限られていたので、私は Oracle 8.0.5 PreProduction リリースのバイナリを利用することにしました。最初 RDBMS-CD 収録のバージョンを動かそうとして失敗し、その時点では正しい RPM を見つけて利用するのはまるで悪夢のようだったからです。

11.5.9.2. Linux (Kernel 2.0.33) 用の Oracle 8.0.5 Pre-Production リリースのインストール

これをインストールするのは簡単です。CD をマウントして、インストーラを起動してください。インストーラが Oracle ホームディレクトリの位置をたずねて、そこにすべてのバイナリをコピーします。もっとも、私はその前の RDBMS インストールを試した時の残りを削除していませんでした。

この後、Oracle データベースは問題なく起動できました。

11.5.10. SAP R/3 のインストールの続行

まず、`idsamd` (<sid>adm) ユーザと `oraids` (ora<sid>) ユーザの環境設定を確認してください。両者とも、`.profile`, `.login` と `.cshrc` をもっていて、いずれも `hostname` を使っているはずです。そのシステムのホスト名が完全修飾名であれば、すべてのファイルで `hostname` を `hostname -s` に変更する必要があります。

11.5.10.1. データベースの読み込み

その後、(exit を選択したかどうかによって) R3SETUP を再起動または継続実行します。R3SETUP は、テーブル領域を作成し、EXPORT1 から EXPORT6 までデータを R3load でデータベースに読み込みます (これは IDES システムです。違う場合は、EXPORT1 から EXPORT3 だけになります)。

データベースの読み込みが完了したら (数時間に渡るかもしれません)、何かパスワードを要求されます。テスト用にインストールしたのであれば、周知のデフォルトパスワードを使ってもよいでしょう (セキュリティが問題になるなら、異なるものを使ってください!)。

質問	入力
Enter Password for sapr3 (sapr3 のパスワードを入力してください)	sap<ret>
Confirum Password for sapr3 (sapr3 のパスワードを再入力してください)	sap<ret>
Enter Password for sys (sys のパスワードを入力してください)	change_on_install<ret>

質問	入力
Confirm Password for sys (sys のパスワードを再入力してください)	change_on_install<ret>
Enter Password for system (system のパスワードを入力してください)	manager<ret>
Confirm Password for system (system のパスワードを再入力してください)	manager<ret>

ここで、`dipgntab` で少し問題がありました。

11.5.10.2. Listener

次のように、Oracle-Listener を `oraids` (ora<sid>) ユーザで起動してください。

```
umask 0; lsnrctl start
```

そうしないと、ソケットが正しいパーミッションにならないので、ORA-12546 が発生するかもしれません。SAP note 072984 をご覧ください。

11.5.11. インストール後の手順

11.5.11.1. SAP R/3 ライセンスキーの要求

一時ライセンスは 4 週間だけ有効なので、これが必要になります。オペレーティングシステムを正しく入力するようにしてください: (X) Other (その他): FreeBSD 4.3 Stable。まず、ハードウェアキーを入手してください。`idsadm` ユーザでログオンし、`saplicense` を呼び出してください。

```
# /sapmnt/IDS/exe/saplicense -get
```

`saplicense` をオプションなしで起動すると、オプション一覧を出力します。ライセンスキーを受け取ったら、次のようにしてインストールできます。

```
# /sapmnt/IDS/exe/saplicense -install
```

そうしたら、次の値を入力します。

```
SAP SYSTEM ID   = <SID, 3 chars>
CUSTOMER KEY    = <hardware key, 11 chars>
INSTALLATION NO = <installation, 10 digits>
EXPIRATION DATE = <yyyymmdd, usually "99991231">
LICENSE KEY     = <license key, 24 chars>
```

11.5.11.2. ユーザの作成

クライアント 000 内でユーザを作成します (クライアント 000 内で行わなければなりませんが、ユーザ `sap*` や `ddic` 以外のユーザでなければならないタスクのためです)。

ユーザ名として、私は通常 *wartung* (または 英語で *service*) を選択しています。必要となるプロファイルは sap_new と sap_all です。セキュリティを高めるため、すべてのクライアントのデフォルトユーザのパスワードを変更すべきです (ユーザ sap* と ddic を含みます)。

11.5.11.3. トランスポートシステム、プロファイル、操作モードなどの設定

クライアント 000 内で、ddic と sap* 以外のユーザについて、少なくとも次のことを行ってください。

作業	トランザクション
トランスポートシステムの設定。たとえば、スタンドアロントランスポートエンティティ	STMS
システム用のプロファイルの作成・編集	RZ10
操作モードとインスタンスの維持管理	RZ04

これらのインストール後の手順は、それ以外も含めて SAP インストールガイドにすべて書かれています。

11.5.11.4. Edit init<sid>.sap (initIDS.sap)

/oracle/IDS/dbs/initIDS.sap ファイルには SAP バックアッププロファイルが含まれます。ここで、使用するテープの容量、圧縮の種類などを定義しなければなりません。sapdba や brbackup と共に動かすため、以下の値を変更しました。

```
compress = hardware
archive_function = copy_delete_save
cpio_flags = "-ov --format=newc --block-size=128 --quiet"
cpio_in_flags = "-iuv --block-size=128 --quiet"
tape_size = 38000M
tape_address = /dev/nsa0
tape_address_rew = /dev/sa0
```

説明:

compress 私が使っているテープは HP DLT1 で、ハードウェア圧縮を行います。

archive_function この変数は、Oracle のアーカイブログを保存する際のデフォルトの振る舞いを定義します。新規のログファイルはテープに保存され、保存済みのログファイルは再度保存された後、削除されます。 これは、もし、データベースを復旧しなければならなくなったのに、 アーカイブテープのどれかが駄目になっていたという場合にも問題を回避します。

cpio_flags デフォルトはブロックサイズを 5120 バイトに設定する -B が使われます。DLT テープについては、HP はブロックサイズに 32 キロバイト以上を推奨していま

すので、64 キロバイトに設定する --block-size=128 を用いました。65535 より多くの inode 番号があるので、--format=newc が必要になります。最後のオプション --quiet は、`cpio` が保存したブロック数を出力すると `brbackup` が文句をいうので必要になります。

cpio_in_flags は、データをテープから読み込むのに必要となるフラグです。データ形式は自動的に認識されます。

tape_size これは通常、テープの raw (生の) 容量を表します。(ハードウェア圧縮を利用しているので) 安全のため、

value は実際の値よりもわずかに小さくしています。

tape_address `cpio` と組み合わせて用いる巻き戻し不可なデバイス。

tape_address_rew cpio と組み合わせて用いる巻き戻し可能なデバイス。

11.5.12. インストール中の問題

11.5.12.1. R3SETUP 中の OSUSERSIDADM_IND_ORA エラー

この段階で R3SETUP がエラーを報告したら、CENTRDB.R3S ファイルを編集します。[OSUSERSIDADM_IND_ORA] という項目を探して、以下の値を変更します。

```
HOME=/home/idsadm (was empty)
STATUS=OK (had status ERROR)
```

以下のように、R3SETUP を再起動します。

```
# ./R3SETUP -f CENTRDB.R3S
```

11.5.12.2. R3SETUP 中の OSUSERDBSID_IND_ORA

この段階で R3SETUP がエラーを報告する可能性もあります。CENTRDB.R3S を編集して、[OSUSERDBSID_IND_ORA] という項目を探し、その項目内の次の値を変更してください。

```
STATUS=OK
```

その後、R3SETUP を再起動します。

```
# ./R3SETUP -f CENTRDB.R3S
```

11.5.12.3. Oracle のインストール中に oraview.vrf ファイルがみつからないという (FILE NOT FOUND) エラー

インストールを開始する前に Oracle On-Line Text Viewer の選択を外すのを忘れています。この選択肢は現在 Linux にはないにもかかわらず、インストール対象に選ばれ

ています。Oracle のインストールメニューでこの製品を選択しているのを外して、インストールを再開してください。

11.5.12.4. R3SETUP, RFC または SAPGUI 起動時に TEXTENV_INVALID エラー

このエラーが起きた場合は、正しいロケールがありません。SAP note 0171356 にインストールしておかなければならない RPM が挙げられています (たとえば、RedHat 6.1 なら saplocales-1.0-3, saposcheck-1.0-1)。R3SETUP がこれに関係するエラーを出してもすべて無視して、対応する状態を ERROR から OK に (CENTRDB.R3S で) 変更して R3SETUP を再起動していると、SAP システムは正しく設定されず、システムを起動できたとしても、`sapgui` でシステムに接続することはできません。古い Linux 版 `sapgui` で接続しようとすると、次のメッセージがでます。

```
Sat May 5 14:23:14 2001
*** ERROR => no valid userarea given [trgmsgo. 0401]
Sat May 5 14:23:22 2001
*** ERROR => ERROR NR 24 occured [trgmsgi. 0410]
*** ERROR => Error when generating text environment. [trgmsgi. 0435]
*** ERROR => function failed [trgmsgi. 0447]
*** ERROR => no socket operation allowed [trxio.c 3363]
Speicherzugriffsfehler
```

これは、SAP R/3 がロケールを割り当てられないか、正しく設定されていない (データベーステーブルの一部の項目が不足している) ためです。SAP に接続するためには、次の項目をファイル DEFAULT.PFL に追加してください (note 0043288 参照)。

```
abap/set_etct_env_at_new_mode =0
install/collate/active =0
rscp/TCP0B =TCP0B
```

SAP システムを再起動してください。地域特有の言語設定が期待どおりに動かないかもしれませんが、システムに接続できるはずです。地域設定を修正 (して、適切なロケールを提供) したら、これらの項目を DEFAULT.PFL から削除して SAP システムを再起動してかまいません。

11.5.12.5. ORA-12546 エラー。リスナーの適切なパーミッションによる操作

次のコマンドを実行して、Oracle リスナーをユーザ `oraids` で起動してください。

```
# umask 0; lsnrctl start
```

そうしないと、ソケットが正しいパーミッションになっていないという ORA-12546 エラーを受け取るかもしれません。SAP note 0072984 をご覧ください。

11.5.12.6. R3SETUP 中の [DIPGNTAB_IND_IND] エラー

一般的なことは SAP note 0130581 (R3SETUP の `DIPGNTAB` 段階での終了) をご覧ください。この特定のインストール段階において、なぜかインストールプロセスは、

適切な SAP システム名 "IDS" を使うかわりに、空文字列 "" を使います。ディレクトリにアクセスするのにパスを <sid> (この場合は IDS) を使って生成するので、これはちょっとした問題を起こすことがあります。

```
/usr/sap/IDS/SYS/...
/usr/sap/IDS/DVMGS00
```

にアクセスするかわりに、つぎのパスを使います。

```
/usr/sap//SYS/...
/usr/sap/D00i
```

インストールを続けるため、リンクと追加のディレクトリを作成します。

```
# pwd
/compat/linux/usr/sap
# ls -l
total 4
drwxr-xr-x 3  idsadm sapsys 512 May 5 11:20 D00
drwxr-x--x 5  idsadm sapsys 512 May 5 11:35 IDS
lrwxr-xr-x 1  root   sapsys 7 May 5 11:35 SYS -> IDS/SYS
drwxrwxr-x 2  idsadm sapsys 512 May 5 13:00 tmp
drwxrwxr-x 11 idsadm sapsys 512 May 4 14:20 trans
```

この現象を解説している SAP notes (0029227 and 0008401) があります。

11.5.12.7. R3SETUP 中の [RFCRSWBOINI_IND_IND] エラー

(ファイル CENTRDB.R3S で) 問題の段階の状態 (STATUS) を ERROR から OK に設定して R3SETUP を再起動してください。インストール後、レポート RSWBOINS をトランザクション SE38 から実行しなければなりません。RFCRSWBOINI および RFCRADDBDIF 段階の詳細については SAP note 0162266 をご覧ください。

11.5.12.8. R3SETUP 中の [RFCRADDBDIF_IND_IND] エラー

(ファイル CENTRDB.R3S で) 問題の段階の状態 (STATUS) を ERROR から OK に設定して R3SETUP を再起動してください。インストール後、レポート RADDBDIF をトランザクション SE38 から実行しなければなりません。詳しくは SAP note 0162266 をご覧ください。

11.6. 高度なトピックス

Linux バイナリ互換機能がどのような仕組みなのか興味がある人はこのセクションを読んでください。以下の文章で説明されていることのほとんどは FreeBSD chat メーリングリスト に投稿された Terry Lambert (<tlambert@primenet.com >) 氏のメール (Message ID: <199906020108.SAA07001@usr09.primenet.com>) をもとにしています。

11.6.1. どのように動くのでしょう?

FreeBSD は、“実行クラスローダ (execution class loader) ”と呼ばれる抽象的な機構を持っています。これは execve(2) システムコールへの楔という形で実装されています。

FreeBSD は、シェルインタプリタやシェルスクリプトを実行するための **#!** ローダを持った単一のプログラムローダではなく、ローダのリストを持っているのです。

歴史的には、Unix プラットフォーム上の唯一のローダーがマジックナンバー (一般的にはファイルの先頭の 4 ないし 8 バイトの部分) の検査を行ないシステムで実行できるバイナリかどうかを検査し、もしそうならバイナリローダーを呼び出すというようになっていました。

もし、そのシステム用のバイナリでない場合には、execve(2) システムコールの呼び出しは失敗の戻り値を返し、シェルがシェルコマンドとして実行しようと試みていたわけです。

この仮定は「現在利用しているシェルがどのようなものであっても」変わりません。

後に sh(1) に変更が加えられ、先頭の 2 バイトを検査した結果 **:\n** であれば代わりに csh(1) を呼び出す、というようになりました (この変更は SCO が最初に行なったと思われます)。

現在の FreeBSD は、プログラムローダリストを走査します。その際、空白文字までの文字列をインタプリタとして認識する、通常の **#!** ローダを用いるため、該当するものが存在しなければ最終的に /bin/sh がロードされます。

Linux ABI をサポートするため、FreeBSD は ELF バイナリを示すマジックナンバを確認します。(ただし、この段階では FreeBSD、Solaris、Linux、そしてその他の ELF イメージ形式を使っている OS を区別することはできません)。

ELF ローダは、特殊なマーク (brand) があるかどうか探します。このマークとは、ELF イメージのコメントセクションのことです。SVR4/Solaris の ELF バイナリには、このセクションは存在しません。

Linux バイナリを実行するためには、ELF バイナリに brandelf(1) で説明されている **Linux** のマークが付けられていなければなりません。

```
# brandelf -t Linux file
```

上のようにすることで、指定されたファイルは **Linux** のマークが付けられ、ELF ローダが認識できるようになります。

ELF ローダが **Linux** マークを確認すると、ローダは **proc** 構造体内の ある一つのポインタを置き換えます。システムコールは全て、このポインタ (伝統的な Unix システムで

はこれは構造体の配列 `sysent[]` で、システムコールが含まれています) を通してインデックスされます。さらに、そのプロセスには Linux カーネルモジュールに必要な シグナルトランポリンコード (訳注: シグナルの伝播を実現するコード) 用の特殊なトラップベクタの設定や、他の (細かな) 調整のための設定が行なわれます。

Linux システムコールベクタは、さまざまなデータに加えて `sysent[]` エントリーのリストを含んでおり、それらのアドレスはカーネルモジュール内にあります。

Linux バイナリがシステムコールを発行する際、トラップコードは `proc` 構造体を用いてシステムコール関数ポインタを 解釈します。そして FreeBSD ではなく Linux 用のシステムコールエントリポイントを得るわけです。

さらに、Linux モードは状況に応じてファイルシステム本来のルートマウントポイントを置き換えてファイルの参照を行ないます。これは、`union` オプションを指定してマウントされたファイルシステム (unionfs ではありません!)が行なっていることと同じです。ファイルを検索する際にはまず `/compat/linux/original-path` ディレクトリを、それから見つけられなかったときにのみ、`/original-path` を調べます。こうすることで、他のバイナリを要求するバイナリの実行を可能にしています (したがって、Linux 用プログラムツールは Linux ABI サポート環境下で完全に動作するわけです)。またこれは、もし対応する Linux バイナリが存在しない場合に Linux バイナリが FreeBSD バイナリをロードしたり、実行したりすることが可能であること、その Linux バイナリに自分自身が Linux 上で実行されていないことを 気付かせないようにする目的で、uname(1) コマンドを `/compat/linux` ディレクトリに置くことができる、ということを意味します。

要するに、Linux カーネルが FreeBSD カーネルの内部に存在しているわけです。カーネルによって提供されるサービス全ての実装の基礎となるさまざまな関数は FreeBSD システムコールテーブルエントリと Linux システムコールテーブルエントリの両方で共通に利用されています。 これらにはファイルシステム処理、仮想メモリ処理、シグナル伝送、System V IPC などが含まれますが、FreeBSD バイナリは FreeBSD グルー (訳注: glue; 二者の間を仲介するという意味) 関数群、そして Linux バイナリは Linux グルー関数群を用いる、という点だけが異なります (過去に存在したほとんどの OS は、自分自身のためのグルー関数群しか備えていません。前述したように、システムコールを発行する際、各々のプロセスの `proc` 構造体内にある、ローダによって動的に初期化されるポインタを参照してアドレスを得る代わりに、静的でグローバルな `sysent[]` 構造体の配列に システムコール関数のアドレスが直接格納されているのです)。

さて、どちらを本来の FreeBSD ABI (訳注: Applications Binary Interface; 同じ CPU を利用したコンピュータ間でバイナリを共有するための規約のこと) と呼ぶべきなのでしょうか? 実は、どちらが本来のものであるかということを論ずることに意味はありません。基本的に、FreeBSD グルー関数群はカーネルの中に静的にリンクされていて、Linux グルー関数群は静的にリンクすることも、カーネルモジュールを介して利用することもできるようになっている、という違いがあるだけ (ただしこれは現時点においての話であり、将来のリリースで変更される可能性がありますし、おそらく実際に変更されるでしょう) です。

あ、「でもこれは本当にエミュレーションと呼べるのか」って? 答えは「いいえ」です。これは ABI の実装であり、エミュレーションとは異なります。エミュレータが呼び出されているわけではありません (シミュレータでもないことをあらかじめ断っておきましょう)。

では、これがよく「Linux エミュレーション」と呼ばれるのは何故でしょうか? それはもちろん FreeBSD の売りにするためでもあるのですが、 実際には、次のような理由によります。 この機能が初めて実装された頃、 動作原理を説明する以外にこの機能を表現する言葉はありませんでした。しかし、コードをコンパイルしたりモジュールをロードしない場合、「FreeBSD 上で Linux バイナリを実行する」という表現は、 厳密に考えると適切ではありません。そこで、その際にロードされているもの自身を表現する言葉 — すなわち「Linux エミュレータ」が必要だったのです。

パート III. システム管理

FreeBSD ハンドブックの以下の章は、FreeBSD のシステム管理の面について書かれています。各章のはじめでは、その章で学ぶ内容や 実際に取り組む前に知っておくべきことについて説明します。

各章は、必要になった時に個別に参照できるように構成されています。 どの順番で読んでも構いませんし、FreeBSD を使うのに、すべてを読み通す必要がある、というわけでもありません。

目次

12. 設定とチューニング 403
12.1. この章では 403
12.2. 中核となる設定 403
12.3. アプリケーションの設定 404
12.4. サービスの起動 405
12.5. バーチャルホスト 406
12.6. 設定ファイル 407
12.7. sysctl によるチューニング 412
12.8. ディスクのチューニング 412
12.9. Kernel 制限のチューニング 414
13. FreeBSD の起動のプロセス 417
13.1. この章では 417
13.2. 起動時の問題 417
13.3. ブートマネージャと起動ステージ 419
13.4. カーネル起動時の応答 424
13.5. Device Hints 425
13.6. init: プロセス制御の初期化 426
13.7. シャットダウン動作 427
14. ユーザと基本的なアカウントの管理 429
14.1. この章では 429
14.2. アカウントの種類 429
14.3. アカウント情報の管理 432
14.4. ユーザへの制限 438
14.5. グループの管理 441
15. セキュリティ 445
15.1. この章では 445
15.2. はじめに 445
15.3. FreeBSDの安全性を高める 448
15.4. DES, MD5 と Crypt 458
15.5. S/Key 459
15.6. Kerberos 466
15.7. ファイアウォール 474
15.8. OpenSSL 486
15.9. IPsec 486
15.10. OpenSSH 493
15.11. ファイルシステムアクセス制御リスト 498
16. ストレージ 501
16.1. この章では 501
16.2. デバイス名 501
16.3. ディスクの追加 502
16.4. RAID 505
16.5. 光メディア (CD & DVD) の作成と使用 511
16.6. フロッピーディスクの作成と使用 518
16.7. データテープの作成と使用 521

16.8. フロッピーディスクへのバックアップ 524
16.9. バックアップの基本 526
16.10. ネットワーク、メモリ、そしてファイルベースのファイルシステム 535
16.11. ファイルシステムのスナップショット 540
16.12. ファイルシステムクォータ 542
16.13. ディスクパーティションの暗号化 546
17. 地域化 (localization) - I18N/L10N の利用と設定 553
17.1. この章では 553
17.2. 基礎知識 553
17.3. 地域化の利用 554
17.4. 国際化に対応したプログラムのコンパイル 561
17.5. FreeBSD を特定の言語に地域化する 561
18. FreeBSD のアップデートとアップグレード 565
18.1. この章では 565
18.2. FreeBSD Update 566
18.3. ドキュメントのアップデート 575
18.4. 開発ブランチを追いかける 579
18.5. ソースの同期 582
18.6. world の再構築 583
18.7. 複数のマシンで追いかける 595

第12章 設定とチューニング

寄稿: Chern Lee.
基にしたチュートリアルの執筆: Mike Smith.
基にした tuning(7) の執筆: Matt Dillon.

12.1. この章では

システムを正しく設定することは、メンテナンスや将来の更新の際の作業の量を減らします。この章では FreeBSD システムの管理上の設定の側面について記述します。

またこの章では FreeBSD システムのパフォーマンスを最適化する チューニングについても記述します。

この章を読むと、以下のことがわかります。

- `rc.conf` の設定と `/usr/local/etc/rc.d` スタートアップシステムの基礎
- ネットワークデバイスに対する、仮想ホストの設定方法
- `/etc` ディレクトリ内のさまざまな設定ファ イルの使い方
- `sysctl` 変数を使った FreeBSD のチューニング方法
- ディスク性能のチューニング方法と、カーネルの制限の変更方法

この章を読む前に、以下のことをやっておくとよいでしょう。

- Unix と FreeBSD の基本を理解する (4章UNIX の基礎知識)。
- FreeBSD のソースコードを最新に保つこと (18章FreeBSD のアップデートとアップグレード) と、カーネルコンフィグレーションおよび構築の基礎 (9章FreeBSD カーネルのコンフィグレーション) に親しんでおく。

12.2. 中核となる設定

システムの設定情報が収められている主な場所は `/etc/rc.conf` です。このファイルにはシステムの起動時にシステムの設定を行なうものをはじめ 多岐に渡る設定情報が含まれています。そのファイル名はダイレクトに、それが `rc*` ファイル群の設定情報であることを示しています。

管理者は `/etc/defaults/rc.conf` のデフォルトの設定を `rc.conf` ファイルにエントリを作ることで上書きすべきです。デフォルトのファイルをそのまま `/etc` にコピー

するのはやめるべきです。それはデフォルト値であってサンプルではないのです。システム固有のすべての変更は `rc.conf` ファイ ルの中でするべきです。

管理の手間を減らす為、クラスター化されたアプリケーションには　サイト共通の設定とシステム固有の設定を分離するさまざまな戦略が適用できます。 推奨されるアプローチは、サイト共通の設定は `/etc/rc.conf.site` のような別のファイルに置き、それをシステム固有の設定情報しか含ませない `/etc/rc.conf` からインクルードすることです。

`rc.conf` は sh(1) によって読み込まれているので、これはじつに簡単に達成できます。たとえば、

- rc.conf:

```
. rc.conf.site
hostname="node15.example.com"
network_interfaces="fxp0 lo0"
ifconfig_fxp0="inet 10.1.1.1"
```

- rc.conf.site:

```
defaultrouter="10.1.1.254"
saver="daemon"
blanktime="100"
```

`rc.conf.site` ファイルは `rsync` のようなプログラムを使うことで全システ ムに配布でき、一方 `rc.conf` ファイルはユニークなままを保つことができます。

システムを sysinstall(8) や `make world` 等で 更新した場合 `rc.conf` ファイルは上書きされません。なのでシステムの設定情報が失われることもありません。

12.3. アプリケーションの設定

基本的に、インストールされたアプリケーションには独自の文法を持つ 固有の設定ファイルがあります。 これらのファイルがベースシステムから分離されているということは重要で、このためパッケージ管理ツールによる配置と管理が容易になっています。

基本的に、それらのファイルは `/usr/local/etc` にインストールされます。設定ファイルの数が多数にのぼるアプリケーションに対しては、それら用にサブディレクトリが作られます。

通常、ports やパッケージがインストールされると 設定ファイルのサンプルが一緒にインストールされます。 大抵、識別のためにサフィックスとして「.default」がついています。アプリケーションのための設定ファイルがまだ存在していなければ、.defaults ファイルをコピーすることで作成できます。

`/usr/local/etc/apache` ディレクトリの例をご覧ください。

```
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf.default
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf.default
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic.default
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types.default
-rw-r--r--  1 root  wheel   7980 May 20  1998 srm.conf
-rw-r--r--  1 root  wheel   7933 May 20  1998 srm.conf.default
```

ファイルサイズの差から、srm.conf ファイルだけが変更されていることが分かります。後に apache を更新した時にも、この変更されたファイルは上書きされることはありません。

12.4. サービスの起動

一つのシステムでサービスをいくつも立ち上げているということは よくあることです。それらには独自の立ち上げかたがあることがあり、それぞれ有利な点があります。

Ports collection やパッケージからインストールしたソフトウェアは しばしば /usr/local/etc/rc.d にスクリプトを置き、システムが起動した時には start、システムをシャッ トダウンする時には stop を引数にして実行します。これは root で実行すべき、または root で起動することを期待されているシステム ワイドなサービスを起動する場合に推奨される方法です。これらのスクリプトはパッケージの一部としてインストール時に記録され、パッケージとともに削除されます。

/usr/local/etc/rc.d にある 一般的なスクリプトは次のようなものです。

```
#!/bin/sh
echo -n ' FooBar'

case "$1" in
start)
        /usr/local/bin/foobar
        -;;
stop)
        kill -9 `cat /var/run/foobar.pid`
        -;;
*)
        echo "Usage: `basename $0` {start|stop}" >&2
        exit 64
        -;;
esac

exit 0
```

このスクリプトはその目的を果すべく起動時に `start`、シャットダウン時に `stop` をつけて呼ばれます。

サービスの中には固有のポートに接続を受けたときに inetd(8) から起動されるものもあります。これはメールリーダサーバ (POP や IMAP 等) の場合によくあります。これらのサービスは `/etc/inetd.conf` ファイルを編集することで有効化されます。このファイルの編集に関する詳細は inetd(8) を見てください。

これらの他に `/etc/rc.conf` による有効化/無効化がカバーされていないサービスもあります。それらは伝統的に `/etc/rc.local` にコマンドを書き込むことで実行されていました。FreeBSD 3.1 にはデフォルトの `/etc/rc.local` は存在していません。もし管理者によって作られていれば、その時は一般的なやりかたとして認められるべきでしょう。`rc.local` は最後の場所と考えられているということを 知っておいてください。サービスを起動させるのにもっといい場所があるなら そこから始めてください。

注記

`/etc/rc.conf` でその他のコマンドを実行しないでください。そのかわり、デーモンの起動やブート時のコマンド実行は `/usr/local/etc/rc.d` にスクリプトを配置してください。

この他にサービスの起動に cron(8) を利用することもできます。 このアプローチには、cron(8) がそのプロセスを `crontab` の所有者権限で実行したり、サービスが 非特権ユーザによって立ち上げられ管理されるなどといった有利な点が いくつもあります。

これで cron(8) の機能の利点を得ることができます。日時の指定を `@reboot` で置き換えることでジョブは システムがブートした直後、cron(8) が起動した時に実行されます。

12.5. バーチャルホスト

FreeBSD の非常にありふれた用途の一つにバーチャルサイトの ホスティングがあります。これは一つのサーバがネットワークには複数のサーバとして現れるものです。これは一つのネットワークインタフェイスに 複数のアドレスを割当てることで実現されます。

ネットワークインタフェイスは「真の」アドレスを 一つと「別名」のアドレスを複数持ちます。これらの別 名は通常 `/etc/rc.conf` に別名のエントリを置くことで追加されます。

`fxp0` インタフェイスへの別名のエント リは以下の様なものです。

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

別名のエントリは　alias0　から始まり昇順に命名されなければなり　ません　(たとえば、_alias1, _alias2 の様になります)。設定プロセス は最初に欠けた番号のところで停まります。

別名のネットマスクの計算は重要ですが、幸いなことに非常に簡単です。　個々のインタフェイスについてそのネットワークのネットマスクを正しく　表現しているアドレスが必ず一つ必要です。 そのネットワークに所属しているそれ以外のアドレスのネットマスクは　すべて 1 でなければなりません。

例として、fxp0 インタフェイスが二つ のネットワークに接続されているものを考えてみましょう。一つはネットマスクが 255.255.255.0 である 10.1.1.0 ネットワークで、もう一つはネットマスクが 255.255.255.240 である 202.0.75.16 ネットワークです。システムは 10.1.1.0 には 10.1.1.1 として、202.0.75.20 には 202.0.75.17 として現れるようにします。

以下のエントリはネットワークインタフェイスを上述の環境に正しく 設定するものです。

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

12.6. 設定ファイル

12.6.1. /etc のレイアウト

設定のための情報が含まれているディレクトリはたくさんあります。 それぞれ以下のものを含んでいます。

/etc	システム全般の設定情報。ここにあるデータはシステム 固有のものです。
/etc/defaults	デフォルトのシステム設定ファイル。
/etc/mail	追加的な　sendmail(8)　の設定、他の MTA の設定ファイル。
/etc/ppp	ユーザモード、およびカーネルモードの ppp プログラムの設定。
/etc/namedb	named(8)　のデータのデフォルトの置場。通常　boot　ファイルはここに置かれ、　/

	`var/db` に置かれた他のデータを 参照するディレクティブを含みます。
`/usr/local/etc`	インストールされたアプリケーションの設定ファイル。 アプリケーションごとのサブディレクトリを含んでいることがあります。
`/usr/local/etc/rc.d`	インストールされたアプリケーションの起動/停止スクリプト。
`/var/db`	永続的なシステム固有のデータファイル。 たとえば named(8) のゾーンファイル、データベースファイル等。

12.6.2. ホスト名

12.6.2.1. `/etc/resolv.conf`

`/etc/resolv.conf` は FreeBSD に インターネットドメインネームシステム (DNS) にどのようにアクセスするかを指定します。

`resolv.conf` の最もよくあるエントリは

`nameserver`	リゾルバが問い合わせるべきネームサーバの IP アドレス。サーバはリストの順に 3 番目まで問い合わせられます。
`search`	ホスト名をルックアップするための検索リスト。 通常、ローカルなホスト名のドメインから決定されます。
`domain`	ローカルドメイン名。

基本的な `resolv.conf` 。

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```

注記

`search` オプションと `domain` オプションは、どちらか一方しか使ってはいけません。

DHCP を利用している場合、dhclient(8) は通常 `resolv.conf` を DHCP サーバから受け取った情報で書き換えます。

12.6.2.2. `/etc/hosts`

`/etc/hosts` は古きインターネットを 偲ばせるシンプルなテキストのデータベースです。 これはホスト名と IP アドレスをマッピングする DNS や NIS と組み合わせて使われます。LAN でつながれているローカルな計算機は、名前引きを簡単にするために named(8) サーバを立ち上げるかわりにここに書くことができます。 さらに `/etc/hosts` はインターネット名のローカルなレコードを提供し、よくアクセスされる名前を外部に問い合わせるのを減らすためにも使えます。

```
# $FreeBSD$
#
# Host Database
# This file should contain the addresses and aliases
# for local hosts that share this file.
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the ↺
resolution order.
#
#
::1                     localhost localhost.my.domain myname.my.↺
domain
127.0.0.1               localhost localhost.my.domain myname.my.↺
domain

#
# Imaginary network.
#10.0.0.2               myname.my.domain myname
#10.0.0.3               myfriend.my.domain myfriend
#
# According to RFC 1918, you can use the following IP networks for
# private nets which will never be connected to the Internet:
#
#       10.0.0.0        -   10.255.255.255
#       172.16.0.0      -   172.31.255.255
#       192.168.0.0     -   192.168.255.255
#
# In case you want to be able to connect to the Internet, you need
# real official assigned numbers.  PLEASE PLEASE PLEASE do not try
# to invent your own network numbers but instead get one from your
# network provider (if any) or from the Internet Registry (ftp to
# rs.internic.net, directory `/templates').
#
```

`/etc/hosts` は、次のようなごく簡単なフォーマットになっています。

```
[############] [########] [##1] [##2] ...
```

例:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

これ以上の情報は hosts(5) をあたってください。

12.6.3. ログファイルに関係する設定

12.6.3.1. **syslog.conf**

`syslog.conf` は syslogd(8) プログラムのための設定ファイルです。これはどのタイプの `syslog` メッセージを対応する ログファイルに記録するかを指定します。

```
# $FreeBSD$
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manual page.
*.err;kern.debug;auth.notice;mail.crit          /dev/console
*.notice;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
security.*                                      /var/log/security
mail.info                                       /var/log/maillog
lpr.info                                        /var/log/lpd-errs
cron.*                                          /var/log/cron
*.err                                           root
*.notice;news.err                               root
*.alert                                         root
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/
console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/
all.log
#*.*                                            /var/log/all.log
# uncomment this to enable logging to a remote log host named loghost
#*.*                                            @loghost
# uncomment these if you're running inn
# news.crit                                     /var/log/news/news.↺
crit
# news.err                                      /var/log/news/news.↺
err
# news.notice                                   /var/log/news/news.↺
notice
!startslip
*.*                                             /var/log/slip.log
!ppp
*.*                                             /var/log/ppp.log
```

これ以上の情報は syslog.conf(5) のマニュアルページに あたってください。

12.6.3.2. **newsyslog.conf**

`newsyslog.conf` は、通常 cron(8) によって予定を決めて実行されるプログラム newsyslog(8) のための設定ファイルです。newsyslog(8) は、ログファイルをいつ保存して再編するかを決定します。`logfile` は `logfile.0` に移され、`logfile.0` は `logfile.1` に、そして以下同様に移されます。また、ログファイルを gzip(1) 形式で保存することもできます。この場合ファイル名は `logfile.0.gz` , `logfile.1.gz` の様になります。

`newsyslog.conf` はどのログファイルが管理され、どのくらいの期間保存され、そしていつ touch されるかを指定します。ログファイルはあるサイズに到達するか、ある決められた時刻・日時で再編されあるいは保存されます。

```
# configuration file for newsyslog
# $FreeBSD$
#
# filename          [owner:group]    mode count size when [ZB] [/
pid_file] [sig_num]
/var/log/cron                           600  3     100  *     Z
/var/log/amd.log                        644  7     100  *     Z
/var/log/kerberos.log                   644  7     100  *     Z
/var/log/lpd-errs                       644  7     100  *     Z
/var/log/maillog                        644  7     *    @T00  Z
/var/log/sendmail.st                    644  10    *    168   B
/var/log/messages                       644  5     100  *     Z
/var/log/all.log                        600  7     *    @T00  Z
/var/log/slip.log                       600  3     100  *     Z
/var/log/ppp.log                        600  3     100  *     Z
/var/log/security                       600  10    100  *     Z
/var/log/wtmp                           644  3     *    @01T05 B
/var/log/daily.log                      640  7     *    @T00  Z
/var/log/weekly.log                     640  5     1    $W6D0 Z
/var/log/monthly.log                    640  12    *    $M1D0 Z
/var/log/console.log                    640  5     100  *     Z
```

これ以上の情報は newsyslog(8) のマニュアルページに あたってください。

12.6.4. `sysctl.conf`

`sysctl.conf` は `rc.conf` によく似ています。値は**##=#**のかたちでセットされます。指定された値はシステムがマルチユーザモードに移行した後でセットされます。すべての変数がこのモードで設定可能というわけではありません。

以下は `sysctl.conf` のサンプルで 致命的なシグナルを記録しないように、また Linux プログラムに それらが実際は FreeBSD 上で動いていることを知らせる様にチューニングしています。

```
kern.logsigexit=0       # Do not log fatal signal exits (e.g. sig 11)
compat.linux.osname=FreeBSD
compat.linux.osrelease=4.3-STABLE
```

12.7. sysctl によるチューニング

sysctl(8) は稼働中の FreeBSD システムに変更を加えるためのインタフェイスです。これには経験を積んだ管理者用の TCP/IP スタックや 仮想メモリシステムのパフォーマンスを劇的に改善する 先進的なオプションが含まれます。500 を越えるシステム変数を sysctl(8) で読んだり セットしたりできます。

本質的には sysctl(8) の機能は次の二つ、システムの設定を読むことと変更することです。

読み取り可能なすべての変数を表示するには以下のようにします。

```
% sysctl -a
```

個々の変数、たとえば `kern.maxproc` を読むには以下のようにします。

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

特定の変数をセットするには、直感的な文法 **##=#** を使ってください。

```
# sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

sysctl 変数の値は通常、文字列、数値、真偽値のいずれかです。(真偽値は yes の場合には `1` で no の場合には `0` です)。

12.8. ディスクのチューニング

12.8.1. sysctl 変数

12.8.1.1. **vfs.vmiodirenable**

`vfs.vmiodirenable` sysctl 変数のデフォルトは 1 (オン) で、0 (オフ) または 1 (オン) にセットすることができます。このパラメータはディレクトリがシステムによってどのように キャッシュされるかを制御します。ほとんどのディレクトリは小さく、ファイルシステムにおいては単一フラグメント (典型的には 1K) であり、バッファキャッシュではさらに小さくなっています (典型的には 512 バイト)。しかしデフォルトモードで動作している時は、 大量のメモリを搭載していても バッファキャッシュは固定数のディレクトリしかキャッシュしません。この sysctl をオンにすると、バッファキャッシュが VM ページキャッシュを、ディレクトリをキャッシュするために使うことを可能にします。 これによる利点は、全てのメモリがディレクトリを キャッシュするのに使えるようになるということです。 欠点は、キャッシュに使われる最小のメモリの大きさが 512 バイトではなく 物理ページサイズ (大抵は 4K) になることです。多数のファイルを操作するサービスを稼動しているなら、常にこの

オプションをオンにすることを推奨します。そのようなサービスには、web キャッシュや大規模なメールシステム、ニューズシステムなどが含まれます。このオプションは一般にメモリを消費しますが、 性能を削減することはありません。 ただし実験して調べてみるべきでしょう。

12.8.1.2. `hw.ata.wc`

FreeBSD 4.3 では IDE のライトキャッシュがオフになりました。これは IDE ディスクへの書き込み帯域幅を減らしてしまうことになりますが、ハードドライブベンダに起因するデータの一貫性に関する 重大な問題のために必要なことだと考えられました。基本的には、書き込み完了時期について IDE ドライブが嘘をつくという問題です。 IDE ライトキャッシュがオンであると IDE ハードドライブはデータを順番に書きこまないばかりか、ディスクの負荷が高い時にはいくつかのブロックの書き込みを 無期限に延期してしまいます。クラッシュや電源故障の場合、ファイルシステムの重大な破壊をもたらします。したがって私たちはデフォルトを安全側に変更しました。 残念ながらこれは大変な性能の低下をもたらし、 私たちはあきらめてこのリリース後にオンに戻しました。 `hw.ata.wc` sysctl 変数を見てデフォルトを チェックしてみるべきです。もし IDE ライトキャッシュがオフになっていたら、hw.ata.wc カーネル変数を 1 に戻すことでオンに戻すことができます。これはブート時にブートローダから行わなければなりません。カーネルがブートした後に行っても効果はありません。

詳しくは ata(4) を見てください。

12.8.2. ソフトアップデート

tunefs(8) プログラムはファイルシステムを細かくチュー ニングするのに使えます。このプログラムにはさまざまなオプションがありま すが、ここではソフトアップデートをオンオフすることだけを考えま す。以下の様にして切り替えます。

```
# tunefs -n enable /filesystem
# tunefs -n disable /filesystem
```

ファイルシステムはマウントされているあいだは tunefs(8) で変更することができません。 ソフトアップデートを有効にする いい機会はシングルユーザモードでどのパーティションもマウント されていない時です。

注記

FreeBSD 4.5 からは、ファイルシステム生成時に newfs(8) の `-U` オプションを使って ソフトアップデートを有効化できるようになりました。

ソフトアップデートはメタデータの性能、主にファイルの作成と削除の性能を劇的に改善します。 すべてのファイルシステムでソフトアップデートを有効にすることを推奨します。ソフトアップデートに関して、2 つの欠点を意識すべきです。 1 つめは、ソフトアップデートはクラッシュ時におけるファイルシス テムの一貫性は保証しますが、物理ディスクの更新が何秒か (1 分に達することもあります!) 遅れる可能性が高いことです。システムがクラッシュした場合、より多くの作業結果が消えてしまうかもしれません。 2 つめは、ソフトアップデート はファイルシステムブロックを解放するのを遅らせるということです。あるファイルシステム (たとえばルートファイルシステム) が満杯近くの時に それに対する大規模な更新、たとえば `make installworld` をすると、空き領域を使い果たして更新が失敗してしまうことがあります。

12.9. Kernel 制限のチューニング

12.9.1. File/Process 制限

12.9.1.1. **kern.maxfiles**

`kern.maxfiles` はあなたのシステムの要求に 応じて増減させることができます。この変数はあなたのシステムのファイル記述子の最大値を示します。 ファイル記述子テーブルが溢れるような時には、システムメッセー ジバッファに頻繁に `file: table is full` と表示されます。これは、`dmesg` コマンドで確認できます。

ファイル、ソケット、パイプ (fifo) は それぞれオープンされるとファイル記述子を一つ消費します。大規模なプロダクションサーバでは その時実行されているサービスの種類や数に応じては あっさり数千のファイル記述子が必要になります。

`kern.maxfile` のデフォルト値はカーネル コンフィグレーションファイルの `MAXUSERS` オ プションで決まります。`kern.maxfiles` は `MAXUSERS` の値に比例して増加します。カスタムカーネルをコンパイルする際は、このカーネルコンフィグ レーションオプションをシステムの利用法に合わせて設定するとよ いでしょう。カーネルは、この数値からほとんどの制限の初期値を 決定します。業務用マシンに、実際に 256 名のユーザが一度に接 続することはないかもしれませんが、大規模なウェブサーバに必要 なリソースは同程度になります。

注記

FreeBSD 4.5 からは、 カーネルコンフィグレーションファイルで `MAXUSERS` を `0` に設定すると、システムの RAM 容量に基づいて適切なデフォルト値が選択されます。

12.9.2. ネットワークの制限

カーネルコンフィグレーションオプション NMBCLUSTERS は、そのシステムで利用可能なネッ トワーク mbuf の量を決定します。通信量の多いサーバで MBUF の量 が少ないと、FreeBSD の性能が低下してしまいます。クラスター一つは およそ 2kB のメモリに対応しているので、1024 だとカーネルメモリ から約 2 MB をネットワークバッファに予約することになります。ど れだけ必要になるかを、簡単な計算で出すことができます。同時に最 大 1000 接続までゆくウェブサーバがあり、それぞれの接続によって 受信バッファ 16kB と送信バッファ 16kB が消費されるなら、ウェ ブサーバをまかなうのに 32MB 程度のネットワークバッファが必要 になります。経験的に有用な値は、それを 2 倍したものなので、32MBx2 = 64MB/2K = 32768 になります。

第13章 FreeBSD の起動のプロセス

13.1. この章では

計算機を起動しオペレーティングシステムをロードするプロセスは、「ブートストラッププロセス」、もしくは単に「起動」と呼ばれます。FreeBSD の起動プロセスを使えば、システムをスタートするときに起きることを かなり柔軟にカスタマイズすることができます。同じ計算機にインストールされた 別のオペレーティングシステムを選択することもできますし、同じオペレーティングシステムの違うバージョンを選択することも、インストールされた別のカーネルを選択することさえできます。

この章では、指定できる設定オプションと FreeBSD の起動プロセスのカスタマイズ方法について詳しく述べます。この章では FreeBSD カーネルがスタートし、デバイスを検出し、init(8) を起動するまでに起きることすべてを扱います。どの最中のことだかはっきりしていない人のために補足すると、テキストの色が明るい白から灰色に変わるまでに起きていることです。

この章を読むと、以下のことが分かります。

- どのように FreeBSD のブートストラップシステムが構成され、そしてそれらが互いにどう関係しているのか
- 起動プロセスを制御するために FreeBSD のブートストラップの各要素に付加できるオプション
- device.hints(5) の基本的な記述方法

x86 限定

この章では Intel x86 システム上で動作する FreeBSD の起動プロセスだけを扱います。

13.2. 起動時の問題

計算機の電源を入れ、オペレーティングシステムをスタートさせるのには、 おもしろいジレンマがあります。定義により、 計算機は、オペレーティングシステムが起動するまでは、

ディスクからプログラムを動かすことも含めて、　何をどうすればよいかまったく知りません。では、計算機はオペレーティングシステムなしに ディスクからプログラムを実行することができず、 オペレーティングシステムのプログラムがディスク上にあるのなら、 どうやってオペレーティングシステムを起動するのでしょうか?

この問題はほらふき男爵の冒険 という本の中に書かれている問題ととてもよく似ています。登場人物がマンホールの下に半分落っこちて、靴紐 (ブートストラップ) をつかんで自分を引っぱり、持ち上げるのです。計算機の黎明期には、ブートストラップ という用語でオペレーティングシステムをロードする 機構のことを指していたのですが、いまはこれを縮めて「ブート (起動)」と言います。

x86 ハードウェアでは、基本入出力システム (Basic Input/Output System: BIOS) にオペレーティングシステムをロードする責任があります。オペレーティングシステムをロードするために、 BIOS がハードディスク上のマスターブートレコード (Master Boot Record: MBR) を探します。MBR はハードディスク上の特定の場所になければなりません。BIOS には MBR をロードし起動するのに十分な知識があり、オペレーティングシステムをロードするために必要な作業の残りは、場合によっては BIOS の助けを得た上で MBR が実行できることを仮定しています。

MBR 内部のコードは、通常ブートマネージャと呼ばれます。とりわけユーザとの対話がある場合にそう呼ばれます。その場合は、通常もっと多くのブートマネージャのコードが、ディスクの最初のトラック または OS のファイルシステム上におかれます (ブートマネージャはブートローダ と呼ばれることもありますが、FreeBSD はこの言葉を起動のもっと後の段階に対して使います)。よく使われるブートマネージャには、boot0 (Boot Easy とも呼ばれる FreeBSD 標準のブートマネージャ), Grub, GAG や LILO 等があります (MBR 内に収まるのは boot0 だけです)。

ディスク上にインストールされているオペレーティングシステムが 1 つであれば、標準の PC MBR で十分です。 この MBR はディスク上の最初の起動可能な (アクティブな) スライスを探し、そのスライスにあるコードを起動してオペレーティングシステムの残りをロードします。デフォルトで fdisk(8) がインストールする MBR は、このような MBR です。`/boot/mbr` を基にしています。

ディスク上にオペレーティングシステムを複数インストールしているなら、　別のブートマネージャ (複数のオペレーティングシステムの一覧を表示できて、起動するオペレーティングシステムを選択できるようなもの) をインストールするとよいでしょう。このようなブートマネージャの中から 2 つを次の節で説明します。

FreeBSD ブートストラップシステムの残りは 3 段階に分かれます。第 1 ステージは MBR によって起動されるもので、MBR は計算機を特定の状態にするために必要なことだけ知っていて、第 2 ステージを起動します。第 2 ステージでは、第 3 ステージを起動する前に、もうちょっとやることができます。第 3 ステージでオペレーティングシステムのロード作業を完了します。起動作業がこれらの 3 段階に分かれているのは、PC の規格がステージ 1 とステージ 2 で実行できるプログラムのサイズに制限を課しているからです。これらの作業をつなぎ合わせることによって、FreeBSD はより柔軟なローダ (loader) を提供しているのです。

その後カーネルが起動し、デバイスの検出と初期化を開始します。 そしてカーネルの起動が終わると、制御はユーザープロセスの init(8) へ移されます。init(8) はまず ディスクが利用可能であることを確かめ、ファイルシステムのマウント、ネットワークで利用するネットワークカードのセットアップ、そして通常 FreeBSD システムで初期時に起動されるすべてのプロセスの起動、といったユーザーレベルでのリソース (資源) 設定を行ないます。

13.3. ブートマネージャと起動ステージ

13.3.1. ブートマネージャ

MBR やブートマネージャのコードは起動プロセスの第 0 ステージと呼ばれることがあります。この節では前述の 2 つのブートマネージャ、boot0 と LILO について説明します。

boot0 ブートマネージャ. FreeBSD のインストーラや boot0cfg(8) がデフォルトでインストールする MBR は、**/boot/boot0** を基にしています (**boot0** は非常に単純なプログラムです。MBR にあるプログラムは、スライステーブルと末尾の識別子 **0x55AA** があるため、446 バイトの大きさでなければならないという制限があるためです)。ハードディスクに boot0 と複数のオペレーティングシステムをインストールした場合、起動時に以下のような画面が出るでしょう。

例13.1 **boot0** のスクリーンショット

```
F1 DOS
F2 FreeBSD
F3 Linux
F4 ??
F5 Drive 1

Default: F2
```

他のオペレーティングシステム、特に Windows® は、既存の MBR を自らの MBR で上書きしてしまうことで知られています。もしそうなってしまったら、もしくは既存の MBR を FreeBSD の MBR で置き換えたいのなら、次のコマンドを使ってください。

```
# fdisk -B -b /boot/boot0 device
```

device は起動するデバイス名で、たとえば 1 番目の IDE ディスクは **ad0**、2 番目の IDE コントローラに接続されている 1 番目の IDE ディスクは **ad2**、1 番目の SCSI ディ

スクは da0 などとなります。MBR の設定をカスタマイズしたい場合は boot0cfg(8) を使ってください。

LILO ブートマネージャ.　このブートマネージャをインストールして FreeBSD を起動するようにするには、まず Linux を起動して /etc/lilo.conf コンフィグレーションファイルに以下を追加してください。

```
other=/dev/hdXY
table=/dev/hdX
loader=/boot/chain.b
label=FreeBSD
```

上記において、FreeBSD のプライマリパーティションとドライブを Linux の識別子を使って指定してください。*X* を Linux のドライブ文字に、また *Y* を Linux のプライマリパーティション番号に置き換えてください。SCSI ドライブを使っているのであれば、*/dev/hd* を */dev/sd* のように読み替えてください。同じドライブ上に両方のオペレーティングシステムが置いてあるなら、loader=/boot/chain.b 行は不要です。これで /sbin/lilo -v を実行するとシステムに新しい変更が反映されます。画面のメッセージを見て確認しておきましょう。

13.3.2. 起動ステージ 1 /boot/boot1 と起動ステージ 2 /boot/boot2

概念上、第 1 ステージと第 2 ステージは ハードディスクの同じ領域上の同一のプログラムの部分部分です。スペースの制約のため 2 つに分割されていますが、いつも一緒にインストールされます。インストーラまたは bsdlabel は、両者を 1 つにまとめたファイル /boot/boot をコピーします (下記参照)。

第 1 ステージと第 2 ステージは、ファイルシステムの外部、起動スライスの最初のトラックに置かれ、先頭が最初のセクタにきます。boot0 またはその他のブートマネージャは、起動プロセスを続けるために必要なプログラムがそこにあると想定しています。使用されるセクタの数は、/boot/boot の大きさから簡単に分かります。

boot1 は 512 バイトの大きさでなければならないという制限があるので、非常に単純なプログラムです。このプログラムは boot2 を検索し、実行するため、そのスライスの情報を保持する FreeBSD の BSD ラベル に関する最低限の情報だけを持っています。

boot2 はもう少し高機能です。これは FreeBSD のファイルシステム上でファイルを見つける能力を持ち、実行するカーネルやローダを指定するための簡単なインタフェースを提供します。

ローダ (loader) はさらに高機能なもので、使いやすく簡単な起動設定が行なえる手段を提供します。boot2 は通常それを起動します。以前の boot2 には、カーネルを直接起動する機能しかありませんでした。

例13.2 **boot2** のスクリーンショット

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

もし仮にインストールされた `boot1` と `boot2` を変更したいのであれば、bsdlabel(8) を使ってください。

```
# bsdlabel -B diskslice
```

`diskslice` は起動するディスクとスライスで、たとえば最初の IDE ディスクの 1 番目のスライスは `ad0s1` となります。

Dangerously Dedicated Mode

bsdlabel(8) を使うとき、 `ad0` のようにディスク名だけを指定すると、 スライスを持たない危険な専用ディスクを作成してしまいます。これはまず間違いなく、あなたがやりたいことではないでしょうから、必ず Return キーを押す前に bsdlabel(8) コマンドを二重にチェックしてください。

13.3.3. 起動ステージ 3 `/boot/loader`

ローダは三段階の起動プロセスの最終段階です。 ローダは通常、ファイルシステム上の `/boot/loader` として存在しています。

ローダは、よりさまざまなコマンド群をサポートした 強力なインタプリタによって提供される簡易組み込みコマンド群を利用することで、 ユーザが利用しやすい設定手段となるように設計されています。

13.3.3.1. ローダプログラムの処理の流れ

ローダは初期化の際にコンソールとディスクの検出を行ない、どのディスクから起動しているかを調べます。そして必要な変数を設定してからインタプリタを起動し、スクリプトからコマンドを送ったり手でコマンドを入力したりできます。

ローダは次に `/boot/loader.rc` を読み込み、通常、変数の標準値を定義した `/boot/defaults/loader.conf` と、そのマシンにローカルな変数を定義した `/`

boot/loader.conf を読み込みます。loader.rc はそれらの変数にもとづき、選択されたモジュールとカーネルをロードします。

ローダは最後に、標準設定で 10 秒のキー入力待ち時間を用意し、 入力がなければカーネルを起動します。 入力があった場合、簡易コマンド群が使えるプロンプトが表示され、ユーザは変数を調整したり、 すべてのモジュールをアンロードしたり、 モジュールをロードしたりすることができます。その後、最終的な起動や再起動へ移行します。

13.3.3.2. ローダの組み込みコマンド

もっともよく使われるローダのコマンドを以下に示します。利用可能なコマンドをすべて知りたい場合、loader(8) を参照してください。

autoboot *seconds*
: *seconds* で与えられた時間内に入力がなければ、カーネルの起動へと進みます。カウントダウンを表示し、標準設定では 10 秒間です。

boot [-options] [kernelname]
: すぐにカーネルの起動へ進みます。 オプション、カーネル名が指定されている場合は、それらが使われます。

boot-conf
: すべてのモジュールの設定を、 起動時と同じように変数にもとづいて自動的に行ないます。このコマンドは、まず unload を行なって、変数—普通 kernel など—を変更した場合にのみ有効に働きます。

help [topic]
: /boot/loader.help を読み込み、ヘルプメッセージを表示します。*topic* に index が指定された場合、利用可能な *topic* を表示します。

include *filename* …
: 指定されたファイル名のファイルを処理します。 ローダはファイルを読み込み、行単位で解釈します。エラーが発生した場合、include コマンドの実行はその時点で停止します。

load [-t type] *filename*
: 指定されたファイル名のカーネル、 カーネルモジュール、あるいは type に指定された種類のファイルをロードします。ファイル名以降に指定された引数はファイルへと渡されます。

ls [-l] [path]
: 指定された *path* にあるファイルを表示します。*path* が指定されていなければ、ルートディレクトリを表示します。-l が指定されていればファイルサイズも表示されます。

lsdev [-v]
: モジュールがロード可能なすべてのデバイスを表示します。もし `-v` が指定されていれば、より詳細な出力がされます。

lsmod [-v]
: ロード済みのモジュールを表示します。`-v` が指定されていれば、より詳細な内容が出力されます。

more *`filename`*
: `LINES` 行を表示するごとに停止しながら指定されたファイルを表示します。

reboot
: すぐにシステムを再起動します。

set *`variable`* , set *`variable`* =*`value`*
: ローダの環境変数を設定します。

unload
: すべてのロード済みモジュールを削除します。

13.3.3.3. ローダの使用例

次にあげるのは、ローダの実践的な使用例です。

- 普段使っているカーネルをシングルユーザモードで起動します。

```
boot -s
```

- 普段使っているカーネルとモジュールをアンロードし、古い (もしくは別の) カーネルをロードします。

```
unload
load kernel.old
```

`kernel.GENERIC` とすると、インストールディスクに入っていた generic カーネルを指定することができます。また、直前にインストールされていたカーネル (たとえば、カーネルを自分で設定したり、アップグレードしたりした場合) を指定するには `kernel.old` とします。

注記

普段のカーネルで使っているモジュールを 指定したカーネルでロードする場合は、下のようにします。

```
unload
```

```
set kernel="kernel.old"
boot-conf
```

- カーネルの設定スクリプト (通常、 カーネル起動時に設定される内容を自動化するスクリプト) をロードします。

```
load -t userconfig_script /boot/kernel.conf
```

13.4. カーネル起動時の応答

カーネルがローダ (通常は) かboot2 (ローダを迂回して) によってロードされると、起動フラグを調べます。もし起動フラグがあれば、それに応じて動作を調整します。

13.4.1. カーネル起動フラグ

良く使われる起動フラグは次のとおりです。

`-a`
: カーネル初期化中に、 ルートファイルシステムとしてマウントするデバイスを尋ねます。

`-C`
: CDROM から起動します。

`-c`
: 起動時にカーネルコンフィグレーションを行なう UserConfig を実行します。

`-s`
: シングルユーザモードで起動します。

`-v`
: カーネル起動時により詳細な情報を表示します。

注記

起動フラグはこの他にもあります。それらについては boot(8) を参照してください。

13.5. Device Hints

##: Tom Rhodes.

注記

これは FreeBSD 5.0 以降の機能です。これ以前のバージョンには存在しません。

起動プロセスの間に loader(8) は device.hints(5) を読み込みます。このファイルにはカーネル起動の環境変数が格納されており、これらの環境変数は「device hints」と呼ばれることがあります。「device hints」はデバイスを設定するためにデバイスドライバが使用します。

device hints は ステージ 3 ブートローダ でも設定できます。変数は `set` コマンドを用いて追加したり、`unset` コマンドを用いて削除できます。`show` コマンドを用いて一覧を見ることもできます。`/boot/device.hints` に設定されている変数は このときに上書きすることができます。ローダで設定した device hints の効果は一時的なものであるため、次回起動するときには無効になります。

システムが起動すると、kenv(1) コマンドでカーネル環境変数をダンプすることができます。

`/boot/device.hints` は 1 行につき一つの変数を設定でき、行頭の「#」はその行がコメントであることを示しています。書式は次の通りです。

```
hint.driver.unit.keyword="value"
```

ステージ 3 ブートローダ で設定するときの書式は次の通りです。

```
set hint.driver.unit.keyword=value
```

`driver` はデバイスドライバの名前、 `unit` はデバイスドライバのユニット番号、`keyword` はヒントキーワードです。キーワードは次の設定を指定します:

- `at`: デバイスがどのバスに接続されているか指定します。
- `port`: 使用する I/O ポートの開始アドレスを指定します。
- `irq`: 使用する IRQ を指定します。
- `drq`: 使用する DMA チャネルを指定します。
- `maddr`: 使用する物理メモリアドレスを指定します。

- `flags`: デバイスに対してさまざまなフラグを設定します。
- `disabled`: `1` が設定されていると、そのデバイスは無効になります。

デバイスドライバはこのリスト以外の変数を設定できるかもしれませんし、 このリスト以外の変数を必要とするかもしれません。 したがって、デバイスドライバのマニュアルを読むことをおすすめします。 より多くの情報を知りたければ、device.hints(5), kenv(1), loader.conf(5), loader(8) などのマニュアルを参照してください。

13.6. init: プロセス制御の初期化

カーネルの起動が完了すると、init(8) というユーザプロセスに制御が移されます。これは `/sbin/init` 、もしくは `loader` の `init_path` 変数で指定される場所にあります。

13.6.1. 自動再起動 (automatic reboot)の動作

自動再起動では、システム上で利用できるファイルシステムの一貫性を確認します。もしそれに問題があって fsck(8) がその不一致を修復できなければ、管理者に直接対処させるため init(8) はシステムをシングルユーザモードへと移行させます。

13.6.2. シングルユーザモード

このモードには、自動再起動の処理中か、ユーザが起動時に `-s` オプションを指定した場合、あるいは `loader` で `boot_single` 変数を設定することによって移行します。

また、マルチユーザモードから 再起動オプション (`-r`) や停止 (halt) オプション (`-h`) なしで shutdown(8) を呼び出すとこのモードに移行します。

`/etc/ttys` でシステムコンソール `console` が `insecure` に設定されている場合、システムはシングルユーザモードに移行する前に `root` のパスワードを入力するように求めます。

例13.3 **`/etc/ttys`** の insecure コンソール

```
# name  getty                           type    status
  comments
#
# If console is marked "insecure", then init will ask for the ↺
root password
# when going to single-user mode.
#
```

```
# #) console # "insecure" ########################
# ## init # root ###################
#
console none                                unknown off insecure
```

注記

insecure コンソールとは、 あなた自身、コンソールが物理的に安全でないと考えていて、root のパスワードを知る人だけがシングルユーザモードを 使えるようにしたいという意味であり、コンソールを安全でない状態で使いたいという意味ではありません。 そのため、安全性を求めるならば secure でなく insecure を選んでください。

13.6.3. マルチユーザモード

init(8) がファイルシステムが正常であると判断するか、 ユーザがシングルユーザモードを終了すると、システムはマルチユーザモードへ移行し、リソースの設定を始めます。

13.6.3.1. リソース設定 (rc)

リソース設定システムはデフォルト設定を /etc/defaults/rc.conf から、そのシステム独自の細かな設定を /etc/rc.conf から読み込みます。 そして /etc/fstab に記述されるシステムファイルシステムをマウントし、 ネットワークサービスの開始、 さまざまなシステムデーモンの開始、 そして最後に、ローカルにインストールされた package の起動スクリプトの実行へと進みます。

リソース設定システムに関する参考資料は、rc(8) にあります。これはスクリプトそのものを調べることと同じくらい優れたものです。

13.7. シャットダウン動作

shutdown(8) を用いてシステムを意図的にシャットダウンした場合、init(8) は /etc/rc.shutdown というスクリプトの実行を試みます。そして、すべてのプロセスへ TERM シグナルを送り、続いてうまく終了できなかったプロセスへ KILL シグナルを送ります。

電源管理機能を持ったシステムで稼働している FreeBSD では shutdown -p now コマンドによって、 直ちに電源を落とすことができます。FreeBSD を再起動するに

は、 `shutdown -r now` を実行するだけです。 shutdown(8) を実行するには、`root` であるか、`operator` グループのメンバでなければなりません。halt(8) や reboot(8) コマンドを利用することもできますが、より多くの情報を知るために、それらと shutdown(8) のマニュアルページを参照してください。

注記

電源管理機能には acpi(4) がカーネルに組み込まれているか、モジュールが読み込まれていることが必要です。

第14章 ユーザと基本的なアカウントの管理

寄稿: Neil Blakey-Milner.

14.1. この章では

FreeBSD は、複数のユーザが同時にコンピュータを使えるようにします。 スクリーンとキーボードの前に一度に座れるのはその中の一人だけですが ユーザは何人でもネットワークを通してログインできます。システムを使うためには、どのユーザもアカウントがなければなりません。

この章では、以下のことを説明します。

- FreeBSD システムにおけるさまざまな種類のユーザアカウントについて
- ユーザアカウントを追加、削除および変更する方法
- ユーザやグループが利用できるリソースの上限を制御する方法
- グループの作成、およびグループにユーザをメンバとして追加する方法

14.2. アカウントの種類

FreeBSD システムへアクセスするには、かならずアカウントが使われ、また、プロセスもすべてユーザによって実行されるので、ユーザとアカウントの管理は、重要なものです。

アカウントには大きく分けて三種類あります。システムアカウント (system accounts)、ユーザアカウント (user accounts)、そしてスーパーユーザ (superuser) です。

14.2.1. システムアカウント

システムアカウントは、DNS、メール、 ウェブサーバといった各種サービスを運用するために使われます。 この目的は、セキュリティを確保するためです。もしすべてのサービスがスーパーユーザで実行されていると、それらのサービスはどんな動作でも可能となり、適切な制限を適用することができません。

システムアカウントの具体例は、`daemon`, `operator`, `bind`, `news` および `www` といったものです。

`nobody` は通常の特権を持たないシステムアカウントです。しかし、`nobody` を利用するサービスが増えれば増えるほど、それに所属するファイルやプロセスも増え、その特権も大きくなるということを忘れないようにしてください。

14.2.2. ユーザアカウント

ユーザアカウントは、主に現実のユーザがシステムにアクセスする手段として用いられるものです。システムにアクセスするすべてのユーザは、それぞれ唯一のユーザアカウントを持つべきです。こうすることで管理者は誰が何を行なっているかがわかりますし、他の人の設定を壊してしまうようなことを避けることができます。

それぞれのユーザは快適にシステムを利用するため、 シェル、エディタ、キー設定、言語など、各自の環境をセットアップすることができます。

FreeBSD システム上のどのアカウントにも、以下のような情報がなにかしら結び付けられています。

ユーザ名
: `login:` プロンプトに対して入力するユーザの名前です。ユーザ名はそのシステムで一意でなければならず、2 名のユーザに同じユーザ名をつけることはできません。有効なユーザ名を作成するには passwd(5) に記載されているいくつもの規則があります。アプリケーションの上位互換性を保つために、8 文字以下の小文字からなるユーザ名を使うことが推奨されています。

パスワード
: それぞれのユーザアカウントにはパスワードがあります。 パスワードは空白にもできますが、行うべきではありません。

ユーザ ID (UID)
: ユーザ ID (UID) は、FreeBSD システムがユーザを一意に識別するための数値です。ユーザ名を指定できるコマンドは、ユーザ名を UID に変換してから扱っています。65535 より大きな UID は、32 ビット以上の整数に対応していないソフトウェアにおいて互換性の問題を引き起こす可能性があるので、65535 以下の UID を使用することが推奨されています。

グループ ID (GID)
: グループ ID (GID) は、ユーザが属する第一グループを一意に識別するための数値です。グループは、UID ではなく、ユーザの GID に基づいて資源へのアクセスを制御する仕組みです。これは、ある種の設定ファイルのサイズを大幅に小さくします。ユーザは、複数のグループに所属できます。65535 より大きな GID は、ソフトウェアに問題を引き起こす可能性があるので、65535 以下の GID を使うことを推奨します。

ログインクラス
: ログインクラスはグループの仕組みを拡張したもので、 別々のユーザに対してシステムを調整する時に、 さらなる柔軟性を提供します。ログインクラスの詳細については、「ユーザへの制限」で説明します。

パスワード変更時間
: デフォルトでは、FreeBSD は定期的にパスワードを変更することをユーザに強制しません。これを pw(8) を使用してユーザごとに設定し、一部またはすべてのユーザに、一定の時間がたったらパスワードを強制的に変更させることができます。

アカウント失効時間
: デフォルトでは、FreeBSD はアカウントを失効させません。たとえば学校で生徒のアカウントがある場合など、 限られた期間だけのアカウントを作成するなら、 そのアカウントがいつ失効するか pw(8) を使って指定できます。有効期間が経過したら、そのアカウントのディレクトリやファイルは残っていますが、 システムへのログインはできなくなります。

ユーザの氏名
: FreeBSD ではユーザ名でアカウントを一意に識別しますが、 必ずしもユーザの本名を反映したものではありません。 この情報をアカウントに関連付けることもできます。この情報は、コメントのように、空白、大文字、および 8 字以上で記載できます。

ホームディレクトリ
: ホームディレクトリは、システム中のディレクトリへのフルパスです。 これはユーザがログインした時に作業を開始するディレクトリです。 一般的な慣習は、すべてのユーザのホームディレクトリを `/home/username` か `/usr/home/username` の下に置くことです。各ユーザは、個人のファイルやサブディレクトリを、ユーザのホームディレクトリに保存します。

ユーザシェル
: シェルは、ユーザがシステムと対話するデフォルトの環境を提供します。いろいろな種類のシェルがあり、経験を積んだユーザはそれぞれ好みがあり、それをアカウントの設定に反映できます。

14.2.3. スーパーユーザアカウント

スーパーユーザアカウントは通常 `root` と呼ばれ、 システム管理を行なうために使われ、権限に制限がありません。 そのため、このアカウントはメールのやりとり、システムの調査、 プログラミングといった日常的な作業を行なうために使われるべきものではありません。

その理由は、スーパーユーザが通常のユーザアカウントと異なり、 操作にまったく制限を受けないことによります。 そのためスーパーユーザアカウントで操作を間違えると、 システムに重大な影響を与えてしまう恐れがあるのです。 ユーザアカウントでは、 仮に操作を間違えてもシステムを壊してしまうようなことはできないようになっています。そのため、ユーザアカウントでログインし、 高い権限が必要なコマンドを実行するときだけスーパーユーザになることが推奨されています。

スーパーユーザで実行するコマンドはいつでも、二回、三回と確認してください。なぜならスペースが多かったり、文字が欠けていたりするだけで、 取り返しのつかないデータの破壊につながる可能性があるからです。

スーパーユーザの権限を得るには、さまざまな方法があります。 root ユーザとしてログインする方法もありますが、これはまったくお勧めできません。

スーパーユーザの権限を手に入れるには、かわりに su(1) を使って下さい。 - オプションをつけて実行すると、実行したユーザに root ユーザの環境が設定されます。このコマンドは wheel グループに入ってるユーザのみが実行でき、他のユーザは実行出来ません。また、実行時には root ユーザのパスワードを必要とします。

以下の例では、make install を行うときにスーパーユーザの権限が必要なので、このコマンドを実行する時だけユーザはスーパーユーザになります。コマンドを実行したら、ユーザは exit を実行してスーパーユーザからログアウトし、通常のユーザアカウントの権限に戻ります。

例14.1 スーパーユーザ権限でプログラムをインストールする

```
% configure
% make
% su -
Password:
# make install
# exit
%
```

1 人の管理者が一台のマシン、 もしくは小規模なネットワークを管理する場合には、su(1) のフレームワークはうまく機能するでしょう。 この代わりとなるのは、 security/sudo package または port です。これはログ機能や、スーパーユーザの権限で実行できるユーザやコマンドを設定できます。

14.3. アカウント情報の管理

FreeBSD には、 ユーザアカウントを操作するのにさまざまなコマンドが用意されています。もっとも一般的なコマンドを以下に示し、それに続いて詳しい使用例を示します。

コマンド	要約
adduser(8)	コマンドラインからユーザを追加するための推奨アプリケーション

コマンド	要約
rmuser(8)	コマンドラインからユーザを削除するための推奨アプリケーション
chpass(1)	ユーザデータベースの情報を変更するための柔軟なツール
passwd(1)	ユーザのパスワードを変更する簡単なコマンドラインツール
pw(8)	ユーザアカウントのあらゆる箇所を変更する強力で柔軟なツール

14.3.1. adduser

adduser(8) は、新しいユーザを登録するためのシンプルなプログラムです。ユーザを追加すると、このプログラムは、/etc/passwd と /etc/group を自動的に更新します。また、新規ユーザのホームディレクトリを作成し、/usr/share/skel から、デフォルトで使用される設定ファイルをコピーします。また、新しく作成されたユーザに対して、ウェルカムメッセージをメールで送信することも可能です。

例14.2 FreeBSD におけるユーザの追加

```
# adduser
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
```

```
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
#
```

注記

入力したパスワードは画面に表示されませんので、ユーザアカウントを作成する際には、パスワードを間違えて入力してしまわないように注意してください。

14.3.2. rmuser

システムから完全にユーザを削除するには、rmuser(8) を使います。このコマンドは、次の手順を実行します。

1. 指定されたユーザの crontab(1) エントリが存在する場合には削除。
2. 指定されたユーザの at(1) ジョブをすべて削除。
3. 指定されたユーザが所有するすべてのプロセスを強制終了。
4. ローカルパスワードファイルから、指定されたユーザのエントリを削除。
5. 指定されたユーザのホームディレクトリを削除　(ディレクトリの所有者が指定されたユーザのものだった場合)。
6. `/var/mail` から、指定されたユーザの到着メールファイルを削除。
7. `/tmp` のような一時ファイル保存領域から、指定されたユーザの所有するファイルを削除。
8. そして最後に、`/etc/group` にある すべてのグループから、指定されたユーザを削除します。

注記

指定されたユーザと同じ名前のグループで、そのユーザが削除されると空のグループとなる場合は、 そのグループ自体が

削除されます。これは adduser(8) によってユーザごとに作成される、ユニークなグループに対応するものです。

スーパユーザアカウントの削除に rmuser(8) を利用することはできません。スーパユーザアカウントの削除はほとんどすべての場合、 大規模なシステムの破壊を意味するからです。

デフォルトでは、以下の例のような対話モードが使われます。

例14.3 **rmuser** による対話的なアカウントの削除

```
# rmuser jru
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/
zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
Updating password file, updating databases, done.
Updating group file: trusted (removing group jru -- personal ↺
group is empty) done.
Removing user's incoming mail file /var/mail/jru: done.
Removing files belonging to jru from /tmp: done.
Removing files belonging to jru from /var/tmp: done.
Removing files belonging to jru from /var/tmp/vi.recover: done.
#
```

14.3.3. chpass

chpass(1) を用いて、 パスワード、シェル、その他の個人情報といった、 ユーザデータベース情報を変更できます。

スーパユーザ権限に限り、chpass(1) を用い、他のユーザの情報やパスワードを変更できます。

ユーザ名の他にオプションを指定しないと、chpass(1) はユーザ情報を編集するエディタを表示します。ユーザがエディタを終了すると、ユーザデータベースが新しい情報に更新されます。

注記

スーパユーザでない場合は、エディタを抜けた後にパスワードを聞かれます。

例14.4 スーパユーザによる対話的な **chpass**

```
#Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

ユーザは、この情報の限られた部分のみ変更が可能です。　また、変更できるのはそのユーザ自身のアカウント情報のみです。

例14.5 通常のユーザによる対話的な **chpass**

```
#Changing user database information for jru.
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

注記

chfn(1) および chsh(1) はいずれも、chpass(1) へのリンクです。また、ypchpass(1), ypchfn(1) および ypchsh(1) も同様です。NIS のサポートは自動的に行なわれますので、コマンドの先頭に yp をつける必要はありません。NIS の設定については、で説明されています。

14.3.4. passwd

passwd(1) は、ユーザが自分のパスワードを変更する通常の方法です。スーパユーザ権限では、他のユーザのパスワードを変更するのに使われます。

注記

誤って、または不正なパスワードの変更を避けるため、 新しいパスワードを設定する前に、もとのパスワードを入力しなければなりません。 スーパーユーザの権限でユーザのパスワードを変更する際には、もとのパスワードを入力する必要はありません。

例14.6 自分のパスワードの変更

```
% passwd
Changing local password for jru.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

例14.7　スーパーユーザ権限での他のユーザのパスワード変更

```
# passwd jru
Changing local password for jru.
```

```
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

注記

chpass(1) 同様、yppasswd(1) は、passwd(1) へのリンクになっていますので、NIS はどちらのコマンドでも動作します。

14.3.5. **pw**

pw(8) は、ユーザやグループの作成、削除、 変更および表示を行なうコマンドラインのユーティリティです。 これは、システムユーザファイルやシステムグループファイルの フロントエンドとして働きます。pw(8) はとても強力な一連のコマンドラインオプションを有しており、シェルスクリプトで使うのに向いていますが、新しいユーザは、この章で紹介されている他のコマンドに比べて難しいと感じるかもしれません。

14.4. ユーザへの制限

FreeBSD は、 個々のユーザが利用できるシステム資源の量を管理者が制限できる方法をいくつも用意しています。その種の制限は、ディスククォータ (quota) とその他の資源の制限の 2 つの章で説明します。

ディスククォータは、ユーザが利用できるディスク容量を制限し、 その都度計算しなくてもディスク使用量を簡単に確認できる手段も提供しています。クォータについては、「ファイルシステムクォータ」で解説しています。

その他のリソースの制限とは、ユーザが消費できる CPU、メモリなどのリソースを制限する手段のことです。これはログインクラスを用いて定義されているもので、この後で解説しています。

ログインクラスは `/etc/login.conf` で定義します。詳細な説明は login.conf(5) に詳しく記載されています。 各ユーザアカウントにはログインクラスが割り当てられていて (デフォルトでは `default` です)、それぞれのログインクラスにはログインケーパビリティの集合が割り当てられています。ログインケーパビリティとは、 **##=#** の組のことで、**##** は周知の識別子、**#** は、**##** に応じて処理される任意の文字列です。ログインクラスとケーパビリティを設定するのはどちらかといえば簡単なことで、login.conf(5) でも説明されています。

注記

FreeBSD は通常、直接 `/etc/login.conf` から設定を読み込まず、より速く検索できる `/etc/login.conf.db` データベースから読み込みます。`/etc/login.conf` を編集する時には `/etc/login.conf.db` を次のコマンドを実行してアップデートする必要があります。

```
# cap_mkdb /etc/login.conf
```

リソースの制限は、2 つの点で標準的なログインケーパビリティと異なっています。第一に、どの制限についても、ソフト (現在の) リミットとハードリミットがあります。ソフトリミットは、ユーザやアプリケーションが調整できますが、ハードリミットを超えることはできません。ユーザはハードリミットを下げることはできますが、上げることはスーパユーザのみができます。第二に、ほとんどのリソースの制限は特定のユーザに対してプロセス毎に適用されるもので、そのユーザが利用するリソースの総量を制限するものではありません。ただし、この違いは制限を特別扱いすることで実現されるものであり、ログインケーパビリティフレームワークの実装によるものではありません。

以下が最もよく使われるリソースの制限になります。残りは、他のすべてのログインケーパビリティと並んで login.conf(5) に書かれています。

`coredumpsize`
: プログラムが生成する core ファイルのサイズにかかる制限は、`filesize` やディスククォータなどの、ほかのディスク使用に関する制限に従属します。この制限は、ディスク領域の消費を制御するあまり厳しくない手段としてよく使われています。ユーザは core ファイルを自分で生成するわけではなく、削除しないことも多いので、これを設定すれば大きなプログラムが異常終了してもディスクの空きがなくならずに済みます。

`cputime`
: そのユーザのプロセスが消費できる CPU 時間 の上限です。これを超えたプロセスは、カーネルにより終了されます。

注記

これは、消費される CPU 時間 についての制限であって、top(1) や ps(1) のフィールドで表示される CPU の割合に関するものではありません。

`filesize`
: ユーザが所有できるファイルの大きさの上限です。ディスククォータ と違い、この制限はユーザのファイルをすべてまとめた集合にではなく、 個々のファイルにかかります。

`maxproc`
: ユーザが実行できるプロセス数の上限です。 フォアグラウンドプロセスとバックグラウンドプロセスの両方を扱います。この上限は、sysctl(8) 変数 `kern.maxproc` で指定されたシステムの制限を超えることはできません。 同時に複数ログインすることや、 パイプライン実行することは便利なことが多いので、 この値をあまり小さな値に設定すると、そのユーザの生産性が悪化する可能性があります。大きなプログラムをコンパイルする場合のように、タスクによっては複数のプロセスやプリプロセッサが実行されます。

`memorylocked`
: これは、1 つのプロセスが mlock(2) によりメインメモリにロックされることを要求できるメモリの最大容量です。amd(8) のようなシステムで重要なプログラムは、メインメモリへロックして、システムがスワップする際に、ディスクのスラッシングを引き起こさないようにします。

`memoryuse`
: これは、どの時点かを問わず、あるプロセスが消費できる最大のメモリ容量です。これは、メインメモリとスワップの使用量を合わせたものです。メモリ消費を抑えるための包括的な制限ではありませんが、手始めにはよいでしょう。

`openfiles`
: これは、あるプロセスが開いておける最大のファイル数です。 FreeBSD では、ファイルはまた、ソケットや IPC チャンネルを表わすのにも使われています。 ですから、あまり低い値に設定しないよう注意してください。これに対応するシステム全体の制限は sysctl(8) `kern.maxfiles` で定義されています。

`sbsize`
: これは、あるユーザが消費できるネットワークメモリ (つまり mbuf) の上限の量です。ユーザは、ネットワーク通信を制限するのに使えます。

`stacksize`
: これは、プロセスのスタックサイズの上限です。あるプログラムが使用しうるメモリの量を制限するには、これだけでは十分ではありません。したがって、他の制限と組み合わせて使わなければなりません。

リソースの制限を設定するにあたり、 ほかにもいくつか覚えておかなければならないことがあります。以下は、一般的なこつやお勧め、さまざまなコメントになります。

- システム起動時に `/etc/rc` から起動されたプロセスは、`daemon` ログインクラスに割り当てられます。
- システムに付属していた `/etc/login.conf` はほとんどの制限について妥当な値になっていますが、 すべてのシステムにおいてふさわしいというわけではありません。制限をあまり緩くするとシステムを悪用しやすくしてしまいますし、 厳しくしすぎると生産性を悪化させてしまいます。
- Xorg のユーザには、 他のユーザより多くのリソースを与えるべきかもしれません。Xorg そのものが多くのリソースを使うだけでなく、より多くのプログラムを並行して使うことをユーザに促します。
- 多くの制限は個々のプロセスにかかるもので、 一人のユーザにまとめてかかるものではありません。例えば、`openfiles` を 50 に設定することは、ユーザが動かすそれぞれのプロセスが最大 50 個のファイルを開けるということです。あるユーザが開けるファイルの総数は、`openfiles` の値に `maxproc` をかけたものになります。同じことがメモリ消費量にもあてはまります。

リソースの制限と、ログインクラス、 ログインケーパビリティ一般についての詳しい情報は、cap_mkdb(1), getrlimit(2) および login.conf(5) をご覧ください。

14.5. グループの管理

グループとは、ユーザを羅列したものです。グループは、グループ名と GID で識別されます。FreeBSD では、あるプロセスが何かするのを許可するかどうかをカーネルが判断する際に、プロセスの UID とそのユーザが所属するグループの一覧を利用します。ほとんどの場合、ユーザもしくはプロセスの GID は一覧の最初のグループを指しています。

グループ名から GID への写像は `/etc/group` にあります。これは、コロンで区切られた 4 項目からなるテキストファイルです。1 番目の項目はグループ名、2 番目は暗号化されたパスワード、3 番目が GID、4 番目がカンマで区切られたメンバの一覧です。文法についての完全な説明は、group(5) をご覧ください。

スーパーユーザは、`/etc/group` をテキストエディタで編集できます。もしくは、pw(8) を使ってグループの追加や編集をできます。たとえば、`teamtwo` というグループを追加して、その存在を確認するには、次のように使います。

例14.8 pw(8) によるグループの追加

```
# pw groupadd teamtwo
# pw groupshow teamtwo
teamtwo:*:1100:
```

この例では、1100 という番号は、teamtwo の GID です。この時点では、teamtwo にメンバはいません。以下のコマンドは、jru を teamtwo のメンバに追加します。

例14.9 pw(8) により新しいグループにメンバを追加する

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
teamtwo:*:1100:jru
```

-M の引数は、カンマで区切られた新しい (空の) グループに追加するもしくは存在するグループのメンバを置き換えるユーザの一覧です。ユーザにとっては、このグループのメンバーシップはパスワードファイルに記載されているプライマリのグループとは異なります。pw(8) の groupshow コマンドを使った時は、そのユーザはグループの一員として表示されませんが、id(1) などのツールを使って情報を問い合わせれば、その情報を引き出せます。ユーザをグループに追加をする際に、pw(8) は /etc/group しか扱わず、/etc/passwd から追加のデータを読んだりはしません。

例14.10 pw(8) によるグループへのユーザ追加

```
# pw groupmod teamtwo -m db
# pw groupshow teamtwo
teamtwo:*:1100:jru,db
```

この例では、-m の引数は、カンマで区切られたグループに追加するユーザの一覧です。前の例と異なり、これらのユーザはグループ一覧に追加され、 グループのユーザ一覧を置き換えることはありません。

例14.11 グループに所属しているユーザを調べるための id(1) の使い方

```
% id jru
uid=1001(jru) gid=1001(jru) groups=1001(jru), 1100(teamtwo)
```

この例では、`jru` は `jru` グループと `teamtwo` グループのメンバです。

このコマンドや `/etc/group` のフォーマットの詳細については、pw(8) および group(5) をご覧ください。

第15章 セキュリティ

本章の基にした security(7) マニュアルページの執筆: Matthew Dillon.

訳: 日野 浩志 <hino@ccm.cl.nec.co.jp >、(jpman プロジェクトの成果を利用させ ていただきました)。

15.1. この章では

この章では、基本的なシステムセキュリティの考え方、 覚えておくべき一般的なルールを紹介し、FreeBSD における高度な話題について簡単に説明します ここで扱う話題の多くは、 一般的なシステムやインターネットセキュリティにもあてはまります。インターネットはもはや、誰もが親切な隣人であろうとする「友好的な」場ではありません。あなたのシステムを安全に保つことは、あなたのデータ、知的財産、時間、その他を、ハッカーやその同類から守るためには欠かせません。

FreeBSD は、 システムとネットワークの整合性と安全性を確実にする仕組みと一連のユーティリティを提供しています。

この章を読むと、以下のことがわかります。

- FreeBSD に関する基本的なシステムセキュリティの考え方
- DES や MD5 のような、FreeBSD で利用できるさまざまな暗号化手法について
- 代替認証システムである、ワンタイムパスワード認証システム S/Key の設定方法
- もう一つの代替認証システム Kerberos の設定方法
- IPFW で firewall を構築する方法
- IPsec の設定方法
- FreeBSD で使われている SSH 実装である OpenSSH の設定および使用方法
- ファイルシステムの ACL (アクセス制御リスト) とは何か、またその使用法

この章を読む前に、次のことが必要になります。

- FreeBSD およびインターネットの基本概念の理解

15.2. はじめに

セキュリティとは、システム管理者をいつも悩ませる仕事の一つです。 すべての BSD Unix マルチユーザシステムは、 従来からいくつかのセキュリティ機構を備えていますが、ユーザを疑心暗鬼に陥らせないように追加のセキュリティ機構を構築し 保守する仕

事はおそらく、システム管理者としてもっとも大きな責務の一つでしょう。マシンの安全性に反映されるのは、管理者が作業したことだけです。 またセキュリティ問題は、快適な環境に必要なものと競合します。一般に Unix システムは膨大な数のプロセスを同時に動作させることができ、そのプロセスの大部分は、サーバ = 外部から接続し、通信するものとして動作します。かつてのミニコンとメインフレームがデスクトップにとってかわり、さらにコンピュータが相互に接続されたネットワークを形成するようになった今日、セキュリティは一層大きな関心事になってきています。

セキュリティを実装するには、 タマネギのように階層化する手法 (a layered「onion」approach) が最適です。どうすれば良いのか簡単に説明すると、便利な機能と同じ数だけセキュリティの階層を作り、システムへの侵入を注意深く監視するのです。あなたはセキュリティを過度に厳重にしたり、 侵入の監視に時間をとられたいとは思わないでしょう。この侵入の発見という部分は、あらゆるセキュリティ機構において最も重要な部分の一つなのです。 たとえば、システムの各バイナリに `schg` フラグ (chflags(1) 参照) を設定するのは、大して意味がありません。フラグを設定すると一時的にバイナリが保護され、 侵入してきた攻撃者によってシステムに加えられる変更のうち、 容易に検出可能な変更は行なえなくなります。しかしその結果として、セキュリティ機構がその侵入者を検出することも まったくできなくなってしまうでしょう。

また、システムセキュリティには、 さまざまな形での攻撃に対処することとも関係しています。攻撃の中には `root` 権限を奪おう (「root 権限を破る」) とはしないけれども、クラッシュやシステムの不安定状態を引き起こそうとするものもあります。このセキュリティ問題は、いくつかに分類することが可能です。

1. サービス妨害攻撃 (denial of service attack)

2. ユーザアカウントの不正利用 (user account compromise)

3. アクセス可能なサーバを使った root 権限の不正利用

4. ユーザアカウントを経由した root 権限の不正使用

5. バックドアの設置

サービス妨害攻撃 (DoS 攻撃) とは、 マシンから必要な資源を奪う行為です。 通常、サービス妨害攻撃はそのマシンで実行されるサーバや ネットワークスタックを過負荷状態にしてマシンをクラッシュさせたり、 マシンを使えなくしたりするような力任せの方法です。サービス妨害攻撃の中には、ネットワークスタックのバグを利用して、パケット一つでマシンをクラッシュさせようとするものもあります。 後者には、カーネルにバグ修正を施すことによってのみ対応することができます。サーバプロセスに対する攻撃は、オプションを適切に指定することによって、攻撃されている状況でサーバプロセスの負荷上昇に限界を設定することで 対応できる場合が多いです。これらに比べると、ネットワークへの力任せの攻撃への対応はずっと難しくなります。 たとえば、偽造パケットによる攻撃 (spoof-packet attack) は、インターネットからシステムを切り離す以外の方法で 防ぐことはほとんど不可能です。 この攻撃によって、マシンを落としてしまうことはできないかもしれませんが、接続しているインターネット回線を飽和させてしまうことはできます。

ユーザアカウントの不正利用は、サービス妨害攻撃よりもずっとよくある問題です。このご時勢でも、自分たちのマシンで標準の telnetd, rlogind, rshd, ftpd サーバを実行させているシステ ム管理者は多いのです。これらのサーバは、デフォルトでは、暗号化されたコネクション上で動作していません。その結果、抱えているユーザ 数が標準くらいであれば、リモートログイン (そのシステムにログイン するには最も普通で便利な方法です) しているユーザのうち一人以上は、 パスワードを覗き見られてしまうでしょう。システム管理者が注意深い 人ならば、たとえログインが成功していたとしても、リモートアクセスログを解析して、疑わしい送信元アドレスを探すものです。

ひとたび攻撃者がユーザアカウントへのアクセス権を入手したら、 攻撃者は `root` 権限を破れると仮定するべきです。 しかし、セキュリティを十分維持し、手入れの行き届いたシステムにおい ては、あるユーザアカウントへのアクセスが可能となっても、 必ずしも攻撃者に `root` へのアクセス権を与えるとは限りません。この違いは重要です。というのは、一般的に `root` へのアクセス権がなければ、攻撃者は自分の侵入の痕跡を隠蔽することができませんし、 そのユーザのファイルを引っかき回したり、 マシンをクラッシュさせたりするのがせいぜいです。ユーザアカウントの不正利用はめずらしいことではありません。なぜなら一般ユーザは、システム管理者ほど注意を払わない傾向があるからです。

システム管理者は、あるマシン上で `root` 権限を奪取する方法は、潜在的に何通りもあるということを心しておかねばなりません。攻撃者は `root` のパスワードを知っているかもしれませんし、 攻撃者が `root` 権限で実行されているサーバのバグを見つけ、 ネットワーク接続を介して `root` 権限を破ることができるかもしれません。 また、攻撃者は suid-root プログラムに存在するバグを知っていて、ユーザアカウントを破れば `root` 権限を奪取できるかもしれません。攻撃者があるマシン上で `root` 権限を破る方法を知ったならば、 攻撃者は裏口を用意する必要がありません。 これまでに発見され、ふさがれた `root` の穴の多くには、攻撃者が自分のしたことの痕跡を消そうとした作業が、かなりの割合で含まれています。 そのため、ほとんどの攻撃者は裏口を作るのです。裏口は、攻撃者がたやすくシステムへの `root` アクセスを再び得られるようにしますが、 有能な管理者に侵入を検知する便利な手段を与えるものでもあります。攻撃者に裏口を作らせないようにするということは、 セキュリティにとっては実際には良くないことかもしれません。なぜなら、攻撃者が最初に見つけて侵入してきたセキュリティホールは ふさがれないからです。

セキュリティを改善する方法は、常に、タマネギの皮のように階層化する手法 (a multi-layered「onion peel」approach) で実装されるべきです。これらは次のように分類できます。

1. `root` とスタッフのアカウントの安全性を高める。

2. `root` の安全性を高める **=** `root` 権限で動作するサーバと suid/sgid バイナリ。

3. ユーザアカウントの安全性を高める。

4. パスワードファイルの安全性を高める。

5. カーネルのコア、raw デバイス、ファイルシステムの安全性を 高める。

6. システムに対して行なわれた、不適切な変更をすばやく検出す る。

7. 必要と思われる以上の対応をとる (paranoia)。

本章の次の節では、上記の各項目についてより深く掘り下げていき ます。

15.3. FreeBSDの安全性を高める

コマンド対プロトコル

この文書を通して、コマンドまたはアプリケーションを指すのには 太字 を使います。たとえばプロトコルであると同時にコマンドでもある ssh などに対して使います。

以下の節では、本章の前節 でとりあげた FreeBSD システムの安全性を高める方法について 述べます。

15.3.1. `root` アカウントとスタッフアカウントの安全性を高める

`root` のアカウントの安全性を確保しないうちから スタッフのアカウントの安全性をうんぬんしてもしかたがありません。ほとんどのシステムでは、`root` アカウントに割り当てたパスワードが 1 つあり ます。まず最初にすべきことは、このパスワードはいつで も不正利用の危険に晒されていると仮定することです。これは `root` のパスワードを消すべきだと言っているのではありません。`root` のパスワードは、マシンにコンソールからアクセスするのには、ほとんどいつでも必要なものです。ここで言いたいのは、コンソール 以外からは、そして可能なら su(1) コマンドを実行する場合も `root` のパスワードを使えないようにするべきである、ということで す。たとえば、あなたが使っている pty が、`/etc/ttys` ファイルで insecure と指定 されているか確認してください。そうすると、`telnet` や `rlogin` 経由では `root` で直接ログインできないようになります。これは、`/etc/ssh/sshd_config` を編集して `PermitRootLogin` に `NO` が設定されるようにすることで実現できます。sshd のような、別のログインサービスを使っている場合でも同様に、直接 `root` へログインすることを許し ていないかどうか確認してください。すべてのアクセス手段 = たとえば FTP のようなサービスが、良くクラックの対象となることを考えましょう。`root` への直接ログインは、システムコンソール経由でのみ可能であるべきなのです。

また当然、システム管理者として自分が `root` になれるようにしておく必要が ありますから、そのための穴をいくつか開けておきます。しかし、それらの穴を動作させるには、さらに

追加のパスワード認証が 必要であるようにしておくことが重要です。root でアクセス可能と する方法の一つとして、適切なスタッフアカウントを (/etc/group 中の) wheel グループに加えることがあります。 wheel グループに入っているスタッフメンバは su を使って root になることが許されます。 パスワードエントリにおいて、スタッフメンバを wheel グループに置くことによって直接 wheel 権限を与えてはいけません。スタッフメンバのアカウントは staff グループに所属させるべきで、その上で /etc/group ファイルを通して wheel グループに加えるべきです。実際に root アクセスの必要なスタッフメンバのみ wheel グループに置くようにすべきです。他の認証方法の場合、たとえば Kerberos を使用する場合には、root アカウントの Kerberos .k5login ファイルを使えば、誰も wheel グループに置く必要なく root に ksu(1) することを許可できます。このやり 方はよりよい解決策なのかもしれません。なぜなら、wheel のメカニズムでは、侵入者がパスワード ファイルを手に入れ、スタッフアカウントのいずれか 1 つを破るこ とができると、root を破ることがまだできてしまうからです。wheel のメカニズムを用いる方が、何もしないよりは良いのですが、必ずしも最も安全な選択肢とは限りません。

スタッフのアカウント、また究極には root アカウントの安全性 を高める間接的な方法は、別のログインアクセスの方法を用いてスタッ フのアカウントの安全性を高め、その上でそのスタッフのアカウント の暗号化パスワードを 「アスタリスク化」 するものです。vipw(8) コマンドを使えば、暗号化されたパスワードを 「*」 1 文字に置き換えられます。このコマンドは、/etc/master.passwd ファイルとユーザ/パスワードデータベースを更新して、パスワード認証によるログインができないようにします。

たとえば、次のスタッフアカウントを、

```
foobar:R9DT/Fa1/LV9U:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/
bin/tcsh
```

こう変更します。

```
foobar:*:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

暗号化されたパスワードは 「*」 と一致することがないので、 この変更によって通常のログインはできなくなります。こうした後は、スタッフメンバは認証のために kerberos(1) や 公開鍵 / 秘密鍵の組を用いる ssh(1) のような代わりとなる認 証手段を利用しなければなりません。 Kerberos のようなログイン機構を使う場合は、一般に Kerberos サーバを実行するマシンと自分のデスクトップワークステーションの安全性を確保しなければなりません。また ssh で公開鍵 / 秘密鍵の組を使う場合、 一般に、ログイン元マシン (通常は自分のワー クステーション) の安全性を確保しなければなりません。ここで、 ssh-keygen(1) で公開鍵 / 秘密鍵の組を生成する際、鍵の組 をパスワードで防御することにより、鍵の組への防御層を追加するこ ともできます。スタッフアカウントのパスワードを「アスタリスク」でつぶすことができると、管理者自身が設定 した安全性の高い方法でしかスタッフメンバがログインできないこと も保証できます。こうして、多くの侵入者が使う重大なセキュリティ の穴である、安全性の低い無関係なマシンからネットワークを覗き見る方法を塞ぐようなセッションを提供する、安全性の高い暗号 化されたコネクションを使うことを、スタッフメンバ全員に強制する ことができるのです。

より間接的なセキュリティの仕組みでは、制限の強いサーバから 制限の弱いサーバへログインすることを前提としています。たとえば、 メインマシンで、様々な種類のサーバを実行させている場合、ワーク ステーションではそれらのサーバを実行させてはなりません。ワーク ステーションを十分に安全にしておくためには、実行するサーバの数 を、一つもサーバが実行されていないというくらいにまでできる限り 減らすべきです。また、パスワードで保護されたスクリーンセーバを 走らせておくべきです。ワークステーションへの物理的アクセスが与 えられたとすると、もちろん言うまでもなく、攻撃者は管理者が設定 したいかなる種類のセキュリティをもうち破ることができるのです。 このことは、管理者として必ず考えておかねばならない問題ですが、システム破りの大多数は、ネットワーク経由でリモートから、ワーク ステーションやサーバへの物理的アクセス手段を持たない人々によっ て行われるという事実もまた、念頭に置いておく必要があります。

Kerberos のような方法を使うことで、スタッフアカウントのパ スワードの変更もしくは停止を一箇所で行なうことと、スタッフメン バがアカウントを持つすべてのマシンに即時にその効果を及ぼすこと が可能となります。スタッフメンバのアカウントが危険に晒されたときに、すべてのマシンでスタッフメンバのパスワードを即座に変更す る能力を過小評価してはいけません。パスワードが分散されている状 況では、N 台のマシンでパスワードを変更すると、てんやわんやの事 態を招く可能性があります。Kerberos を使用すると、パスワードの 再発行に制限 (re-passwording restriction) を課することもできま す。この機能を使うことにより、ある Kerberos チケットをしばらく 経つとタイムアウトにすることができるだけでなく、一定期間 (例 えば、1 ヶ月に 1 回) 経つと、ユーザに新しいパスワードを選ぶよ うに要求することもできます。

15.3.2. root 権限で実行されているサーバと SUID/SGID バイナリの安全性を高める

用心深いシステム管理者は、自分に必要なサーバプロセスだけを 過不足なく実行させるものです。サードパーティ製のサーバは、よくバグを持っ ていがちだということに注意して下さい。たとえば、古いバージョンの imapd や popper を実行させておくのは、全世界に万能の `root` の切符を与えているようなものです。自分で注意深くチェックしていない サーバは、決して実行してはいけません。`root` で実行させる必要のあるサーバはほとんどありません。たとえば、 ntalk, comsat, finger デーモンを、専用ユーザの砂場 (sandbox) で実行させることができます。 管理者が膨大な数の問題を経験していないのなら、この「砂場」は完 璧ではありませんが、セキュリティに関するタマネギ的アプローチは ここでも成り立ちます。砂場で実行されているサーバプロセスを経由 して侵入を果たすことができたとしても、攻撃者はさらに砂場から外 に脱出しなければなりません。攻撃者が通過せねばならない層の数が 増えれば増えるほど、それだけ攻撃者が侵入に成功する確率が減りま す。Root の抜け穴は歴史的に、基本システムサーバも含め、`root` 権限で実行されるほとんどすべてのサーバプロセスで発見されています。ユーザが sshd 経由でのみログインし、telnetd, rshd, rlogind 経由でログインすることが決してないマシンを稼働させているのであれば、それらのサービスを停 止させて下さい!

FreeBSD では、今では ntalkd, comsat, finger は砂場で実行させることがデフォルトになっています。次に砂場で実行させるべきプログラムの候補と して、named(8) が

あります。 /etc/defaults/rc.conf ファイルには、 named を砂場で実行するために必要な 引数がコメントアウトされた形式で含まれています。新しいシステム をインストールしているか、それとも既存のシステムをアップグレードして使っているかに依存しますが、砂場として使用する特別のユー ザアカウントがインストールされていないかもしれません。用心深い システム管理者であれば、できるだけいつでも研究を怠らず、サーバに砂場を仕込むものでしょう。

通常、砂場で実行しないサーバが他にいくつかあります。 sendmail, popper, imapd, ftpd などです。これらのうちいくつか のサーバには代わりとなるものがありますが、代わりのものをインス トールするには、あなたが思うより多くの仕事が必要になるかもしれ ません (便利さという要素がまたも勝利を収めるわけです)。これら のサーバは、root 権限で実行しなければばならないかもしれません。また、 これらのサーバ経由で生じる侵入を検出するためには、他の仕組みに 頼らなくてはならないかもしれません。

システムの root 権限の潜在的な穴で他に大きなものには、シ ステムにインストールされた suid-root/sgid バイナリがあります。これらのバイナリは、rlogin のように、/bin, /sbin, /usr/bin, /usr/sbin に存在するものがほとんどです。100% 安全なものは存在しないとは いえ、システムデフォルトの siud/sgid バイナリは比較的安全とい えます。それでもなお、root セキュリティホールがこれらのバイナリにときおり発見されています。1998 年に xterm (普通、suid 設定されています) を脆弱にしていた Xlib の root セキュリティホールが見つかりました。安全である方がよいので、用心深いシステム管理者は残念に思いながらも、スタッフのみが実行 する必要がある suid バイナリは、スタッフのみがアクセス可能な特 別なグループに含めるように制限を加え、誰も使わない suid バイナ リは (chmod 000 を実行して) 片付けてしまう でしょう。ディスプレイを持たないサーバは、一般的に xterm のバイナリを必要としません。sgid バイナリもほとんど同様の危険な存在になり得ます。侵入者が kmem に sgid されたバイナリを破ることができた場合、その侵入者 は /dev/kmem を読み出すことができるように なるでしょう。つまり、暗号化されたパスワードファイルを読み出す ことができるようになるので、パスワードを持つどのアカウントをも、潜在的な危険に晒すことになります。他にも、kmem グループを破った侵入者が pty を通して 送られたキーストロークを監視できるという危険があります。キース トロークには、安全な方法でログインするユーザが使っている pty も含まれます。 tty グループを破った侵入者は、ほぼ任意のユーザの tty へ書き込みができます。ユーザが端末プログラムやキーボードを シミュレーションする機能を持ったエミュレータを使っている場合、 侵入者は潜在的に、結局そのユーザとして実行されるコマンドをユー ザの端末にエコーさせるデータストリームを生成できる可能性があり ます。

15.3.3. ユーザアカウントの安全性を高める

ユーザアカウントは、普通、安全性を高めることが最も困難です。 スタッフに対しては、とても厳格なアクセス制限を強制しパスワードを「アスタリスク」で外すことができるでしょうが、管理者が 持ちうる一般ユーザすべてのアカウントに対して同じことはできない かもしれません。管理者が十分に統率をとることができるなら、管理 者は勝利し、ユーザのアカウントの安全を適切に確保できるかもしれ ません。それができないならば、よりいっそう気を配って一般ユーザ のアカウントを監視するよりほかありません。一般ユーザアカウン

トに対し ssh や Kerberos を利用することには、システム管理がさらに増えたりテクニカルサポートが必要に なるなどの問題があります。それでも、暗号化パスワードファイルと 比較するとはるかに良い解です。

15.3.4. パスワードファイルの安全性を高める

できるだけ多くのパスワードを * で外し、 それらのアカウントのアクセスには ssh や Kerberos を使うようにすることが、唯一の確実な方法です。 暗号化パスワードファイル (`/etc/spwd.db`) は `root` でのみ読み出し可能だけれども、 たとえ、侵入者が root の書き込み権限は得られなくとも、 読み出しアクセス権限を得ることは可能かもしれません。

セキュリティスクリプトで常にパスワードファイルの変更をチェックし、報告するようにすべきです (ファイルの完全性のチェック 節参照)。

15.3.5. カーネルのコア、raw デバイス、ファイルシステムの安全性を 高める

`root` の権限を破ると、攻撃者はほとんど何でもできますが、特に重宝さ れる特定の事柄もいくつかあります。たとえば、最近のカーネルは、組 み込みのパケット覗き見デバイス (packet sniffing device) ドライ バを備えているものがほとんどです。FreeBSD では `bpf` デバイスと呼ばれています。侵入者 は普通、侵入済みのマシンでパケット覗き見プログラムを実行させよ うと試みます。侵入者にわざわざそういう機能を提供する必要はない ので、ほとんどのシステムで `bpf` デバイスを組み込むべきではありません。

`bpf` デバイスを外しても、`/dev/mem` と `/dev/kmem` という悩みの種がまだ残っています。この問題に関しては、侵入者は raw ディスクデバイスに書き込 むこともできます。ほかにも、モジュールローダ、kldload(8) とい う、別のカーネル機能があります。やる気まんまんの侵入者は、KLD モジュールを使って自分独自の `bpf` もしくはその他覗き見デバイス を動作中のカーネルにインストールできます。この問題を 避けるため、システム管理者はカーネルをより高い安全レベル (securelevel)、少なくとも安全レベル 1 で実行させる必要がありま す。安全レベルは `sysctl` を使って `kern.securelevel` 変数を操作して設定できます。ひとたび安全レベルに 1 を設定すると、raw デバ イスに対する書き込みアクセスは拒否され、たとえば `schg` のような特別な `chflags` フラグの機能が 強制されます。システム起動に関わる重要なバイナリやディレクトリ、 スクリプトファイルなど、安全レベルが設定されるまでの間に実行さ れるすべてのものに対しても、確実に `schg` フラグを設定してください。この設定をやり過ぎても 構いませんが、より高い安全レベルで動作している場合、システムの アップグレードがはるかに困難になります。システムをより高い安全 レベルで実行させるようにするが、すべてのシステムファイルとディ レクトリに `schg` フラグを設定しないというところで妥協するという手もあります。 もう一つの可能性としては、単純に `/` および `/usr` を読み 込み専用でマウントすることです。ここで特筆すべきことは、システ ムを守ろうとして厳しくしすぎると、侵入を検出するという非常に重 要なことができなくなってしまうということです。

15.3.6. ファイルの完全性のチェック: バイナリ、設定ファイルなど

ことこの問題に至ると、システム管理者にできることは、便利さ という要素がその醜い頭を上げない程度に、コアシステムの設定と制 御ファイルを防御することだけです。たとえば、 / および /usr にある 大部分のファイルに schg ビットを設定するた めに chflags を使用するのは、おそらく逆効果 でしょう。なぜなら、そうすることでファイルは保護できますが、侵 入を検出する窓を閉ざしてしまうことにもなるからです。セキュリティ のタマネギの最後の層はおそらく最も重要なもの = 検出で す。セキュリティの残りのものは、突然の侵入を検出できなければ、まったく有用ではありません (あるいは、もっと悪ければ、安全性に 対する間違った感覚を植え付けてしまいます)。タマネギの仕事の半 分は、もう半分の検出側が攻撃者を攻撃の最中に捕えるようにするた めに、攻撃者を食い止めるのではなく侵入を遅らせることなのです。

侵入を検出する最も良い方法は、変更されていたり、消えていたり、入れた覚えがないのに入っているファイルを探すことです。変更 されたファイルを探すのに最も良い方法は、もう一つの (しばしば中 央に集められた)、アクセスが制限されたシステムから行なうもので す。さらに安全でアクセス制限されたシステム上でセキュリティ用ス クリプトを書けば、スクリプトは潜在的な攻撃者からはほぼ見えなくなります。これは重要なことです。この有効性を最大限に活用 するためには、一般的に、アクセスの制限されたマシンから実際に使っている他のマシンへのかなりのアクセスを許可する必要があります。普 通は、他のマシンからアクセス制限されたマシンへ読み込み専用の NFS エクスポートをしたり、アクセス制限されたマシンから他のマシンへ ssh 接続を行なうために、ssh 鍵のペアを作ったりすることで行います。 ネットワークのトラフィックを別にして、NFS は最も可視性のない方法です = 各クライアント上のファイルシステムを、 事実上検出されずに監視できるようになります。アクセス制限された サーバがスイッチを通してクライアントに接続されている場合、たい てい NFS がより良い選択肢です。アクセス制限されたサーバがハブ や、いくつかのルーティング層を通してクライアントに接続している場合、 NFS は (ネットワークの面で) あまりにも危険なので、 ssh の方が認証を行った跡は残りますが、良い方法でしょう。

アクセス制限されたマシンに、監視しようとするクライアントシ ステムへの少なくとも読み込みのアクセス権を与えたら、次に実際に 監視するためのスクリプトを書かなくてはいけません。NFS マウント をすれば、find(1) や md5(1) などの単純なシステムユー ティリティでスクリプトを書くことができます。少なくとも 1 日 1 回、クライアントのファイルを直接 md5 にかけ、さらにもっと頻繁 に /etc および /usr/local/etc にあるようなコントロール用 ファイルを試験するのが一番です。アクセス制限されたマシンが正しい と知っている、基となる md5 情報と比べて違いが見つかった場合、 システム管理者に調べて欲しいと悲鳴を上げるようにすべきです。優 れたセキュリティ用スクリプトは、/ および /usr などのシステムパーティション上で不適 当に suid されたバイナリや、新たに作成されたファイルや削除され たファイルがないかどうかを調べるでしょう。

NFS ではなく、ssh を使用する場合は、 セキュリティ用スクリプトを書くのはずっと難しいことで す。スクリプトを動かすためには、クライアントに対してスクリプト を scp しなくてはいけませんし、それは目に見 えてしまいます。そして、安全のためには、スクリプトが使うバイナ リ (find など) を scp する必要もあります。クライアントマシンの ssh クライア

ントはすでに攻撃されてしまっているかもしれません。結局のところ、安全でないリンク上の場合は ssh は必要かもしれませんが、ssh を扱うのはとても大変なことです。

優れたセキュリティ用スクリプトは、ユーザやスタッフメンバの アクセス設定ファイルの変更もチェックするものです。`.rhosts`, `.shosts`, `.ssh/authorized_keys` など … `MD5` チェックの範囲外になってしまうであろう ファイル群です。

ユーザ用のディスク容量が非常に大きい場合は、パーティション 上の各ファイルを見て回るのに大変な時間がかかるかもしれません。この場合は、マウントフラグを設定して、このパーティションに suid されたバイナリやデバイスを置けないようにするのが良い考えです。`nodev` および `nosuid` オプション (mount(8) 参照) が知るべきものでしょう。とにかく少なくとも週に 1 度はファイルシステムをスキャンするべきです。なぜなら、この層の目的は、侵入が成功したかどうかに関わらず、侵 入があったことの検出をすることだからです。

プロセスアカウンティング (accton(8) 参照) は、マシンへの侵入を検出するためのメカニズムとして推奨できる、比較的オーバヘッドの少ないオペレーティングシステムの機能です。侵入を受けた後でも当該ファイルが無傷である場合に、侵入者が 実際にどのようにしてシステムに侵入したかを追跡するのに特に役立ちます。

最後に、セキュリティスクリプトはログファイルを処理するよう にし、ログファイル自体もできるだけ安全性の高い方法で生成するよ うにすべきです = リモート syslog は極めて有益になり得ま す。侵入者は自分の侵入の痕跡を覆い隠そうとしますし、また、ログ ファイルはシステム管理者が最初の侵入の時刻と方法を追跡してゆく ために極めて重要です。ログファイルを永久に残しておくための 1 つの方法は、システムコンソールをシリアルポートにつないで走らせ、 コンソールを監視している安全なマシンを通して絶えず情報を集める ことです。

15.3.7. 偏執狂的方法

多少偏執狂的になっても決して悪いことにはなりません。原則的 に、システム管理者は、便利さに影響を与えない範囲でいくつでもセ キュリティ機能を追加することができます。また、いくらか考慮した結果、便利さに影響を与えるセキュリティ機能を追加することもできます。より重要なことは、セキュリティ管理者はこれを多少混ぜこぜにして使うべきだということです。= もしあなたが、本文書に書かれている勧告をそのまま使用した場合は、予想される攻撃者はやはり本文書を読んでいるわけですから、あなたの防御策を教えてしまうことになります。

15.3.8. サービス妨害攻撃

このセクションではサービス妨害攻撃 (DoS 攻撃) を扱います。サービス妨害攻撃は、普通は、パケット攻撃です。ネットワークを飽 和させる最先端の偽造パケット (spoofed packet) 攻撃に対してシス テム管理者が打てる手はそれほど多くありませんが、一般的に、その 種の攻撃によってサーバがダウンしないことを確実にすることで、被 害をある限度に食い止めることはできます。

1. サーバの fork の制限。

2. 踏み台攻撃の制限 (ICMP 応答攻撃、ping broadcast など)。

3. カーネルの経路情報のキャッシュ。

よくあるサービス妨害攻撃は、fork するサーバプロセスに対す るものです。これは、サーバにプロセス、ファイル記述子、メモリを マシンが死ぬまで食い尽くさせようとするものです。inetd (inetd(8) 参照) には、この種の攻撃を制限するオプションが いくつかあります。マシンがダウンすることを防止することは可能で すが、この種の攻撃によりサービスが中断することを防止することは 一般的に言ってできないことに注意する必要があります。inetd のマニュアルページを注意深く読んで下さい。特に、`-c`, `-C`, `-R` オプションに注意して下さい。IP 偽造攻撃 (spoofed-IP attack) は inetd の `-C` オプションの裏をかけるので、一般にオプションを組み合わせて使用するべきであることに注意して下さい。スタンドアロンサーバの中には、自分自身で fork を制限するパ ラメータを持っているものがあります。

Sendmail には、`-OMaxDaemonChildren` オプションがあります。シ ステム負荷の値変化には遅れがあるので、sendmail の負荷限界指定 オプションを使うよりも、このオプションを使う方がまともに動作す る可能性ははるかに高いです。 sendmail の実行を開始する際に、`MaxDaemonChildren` パラメータを設定するべき です。その値は、通常見込まれる負荷を扱える程度に十分高いが、そ れだけの数の sendmail を操作しよう とするとマシンが卒倒してしまうほどには高くないような値に設定す るべきです。sendmail をキュー処理モード (`-ODeliveryMode=queued`) で実行することや、sendmail デーモン (`sendmail -bd`) をキュー処 理用プロセス (`sendmail -q15m`) と別に実行す ることも、用心深いことと言えます。それでもなおリアルタイムでの 配送を望むのであれば、`-q1m` のようにすることで、キュー処理をはるかに短い時間間隔で行うことができます。いずれに しても、`MaxDaemonChildren` オプションに合理 的な値を確実に指定して、sendmail がなだれをうって失敗すること がないようにして下さい。

syslogd は直接攻撃される可能性 があるので、可能ならばいつでも `-s` オプション を用いることを強く推奨します。これができないなら、`-a` オプションを使って下さい。

tcpwrapper の逆 identd などの接 続返し (connect-back) を行うサービスについては十分注意を払うよ うにするべきです。これらは直接攻撃を受ける可能性があります。こ ういう事情があるので、tcpwrapper の 逆 ident 機能を使おうとは思わないのが一般的です。

境界ルータのところでファイアウォールを設けて、外部からのア クセスに対して内部サービスを防御するという考えは実によいもので す。この考えは、LAN の外部からの飽和攻撃を防ぐことにあり、内部 サービスをネットワークベースの `root` 権限への攻撃から防御するこ とにはあまり考慮を払っていません。ファイアウォールは常に排他的 に設定して下さい。つまり、「ポート A, B, C, D と M から Z まで以外 のすべてにファイア

ウォールを設ける」というふうにです。このようにすることで、named (ゾーンのプライマリである場合)、 ntalkd, sendmail などのインターネットからア クセスできるサービスとして特に指定するもの以外の、小さい番号の ポートすべてをファイアウォールで防御することができます。ファイ アウォールをこの他のやり方 = つまり包含的もしくは受容的 なファイアウォールとして設定しようとする場合、「close」することを忘れてしまうサービスがいくつか 出てきたり、新しい内部サービスを追加したのにファイアウォールの 更新を忘れたりする可能性がよく出てきます。ファイアウォール上の 大きい番号のポートを開けておくことにより、小さい番号のポートを 危険に晒すことなく受容的な動作を許すことができます。FreeBSD で は、`net.inet.ip.portrange` への `sysctl` (`sysctl -a | fgrep portrange`) をいろいろ使用することで、動的バインドに使用される ポート番号の範囲を制御できることを記憶にとどめておいてください。これによりファイアウォールの設定を簡略化することもできます。 たとえば、通常の first/last 範囲として 4000 から 5000 を、高位ポートの範囲として、49152 から 65535 を指定し、(いくつかのインターネットアクセス可能 なポートをブロックから除外するのはもちろんですが) 4000 より下のすべてのポートをブロックするという設定が考えられます。

また別のよくあるサービス妨害攻撃として、踏み台攻撃 (springboard attack) と呼ばれるものがあります = これは、 あるサーバを攻撃し、そこ結果として生成される応答が自分自身、ロー カルネットワーク、そして他のマシンを過負荷に追い込むようにする 攻撃です。この種の攻撃の中で最もありふれたものに、ICMP ping broadcast 攻撃があります。攻撃 者は、実際に攻撃したいマシンのアドレスを送信元アドレスに設定し た ping パケットを偽造して、対象の LAN のブロードキャストアド レスに向けてパケットを送信します。境界にあるルータがブロードキャ ストアドレスに対する ping パケットを握り潰すように設定されてい ない場合、LAN は、詐称された送信元アドレスに向けて応答パケット を生成するはめになり、犠牲となるマシンが飽和するところまで行っ てしまいます。攻撃者が同じトリックを異なるネットワーク上のいく つものブロードキャストアドレスに対して同時に使用した場合、とく にひどいことになります。これまでに、120 メガビット以上のブロー ドキャスト攻撃が観測されています。2 番目の踏み台攻撃は、ICMP エラー報告の仕掛けを狙うものです。攻撃者は ICMP エラー応答を生 成するパケットを生成し、サーバの受信ネットワークを飽和させ、そ の結果としてサーバが送信ネットワークを ICMP 応答で飽和させてし まうようにすることができます。mbuf を消費し尽くさせることによ り、この種の攻撃でサーバをクラッシュさせることも可能です。サー バが生成した ICMP 応答を十分速く送信できない場合、とくにひどい ことになります。FreeBSD カーネルには、この種の攻撃の効果を抑制する `ICMP_BANDLIM` と呼ばれる新しいカーネルコンパイルオプション があります。踏み台攻撃の 3 つめの主要なクラスに属する攻撃は、 udp echo サービスのような、特定の inetd 内部サービスに関連する ものです。攻撃者は、単に送信元アドレスがサーバ A の echo ポー トであり、送信先アドレスがサーバ B の echo ポートであるように UDP パケットを偽造します。ここでサーバ A, B はともにあなたの LAN に接続されています。この 2 つのサーバは、この一つのパケッ トを両者の間で互いに相手に対して打ち返しあいます。このようにし てパケットをほんのいくつか注入するだけで、攻撃者は両方のサーバ と LAN を過負荷状態にすることができます。同様の問題が内部 chargen ポートにも存在します。 有能なシステム管理者はこの手の inetd 内部テストサービスのすべてを無効にしておくものです。

偽造パケット攻撃は、カーネルの経路情報キャッシュに過負荷を　生じさせるために用いられることもあります。`net.inet.ip.rtexpire` , `rtminexpire` , `rtmaxcache` の `sysctl` パラメータを参照して下さい。でた らめな送信元 IP アドレスを用いた偽造パケット攻撃により、カーネ ルは、一時的なキャッシュ経路を経路情報テーブルに生成します。こ れは `netstat -rna | fgrep W3` で見ることがで きます。これらの経路は、普通は 1600 秒程度でタイムアウトになり ます。カーネルがキャッシュ経路テーブルが大きくなり過ぎたことを 検知すると、カーネルは動的に `rtexpire` を減らしますが、`rtminexpire` より小さくなるようには決して減らしません。ここに問題が 2 つあります。

1. 負荷の軽いサーバが突然攻撃された場合、カーネルが十分素　早く反応できないこと。

2. カーネルが持続的攻撃に耐えられるほど十分 `rtminexpire` が低く設定されていないこと。

自分のサーバが T3 もしくはそれより高速の回線でインターネッ トに接続されている場合、sysctl(8) を用いて `rtexpire` と `rtminexpire` とを手動で上書きしておくことが思慮深いことといえます。どちらか 一方でも 0 には決してしないで下さい (自分のマシンをクラッシュ させたくないのであれば)。両パラメータを 2 秒 に設定すれば、攻撃から経路情報テーブルを守るには十分でしょう。

15.3.9. Kerberos および SSH を用いたアクセスの問題

もしあなたが、Kerberos と ssh を使いたいのだとしたら、 両者に関して言っておかねばならない問題がいくつかあります。 Kerberos V は大変優れた認証プロトコルですが、Kerberos 化された telnet や rlogin は、バイナリストリームを扱う のに不向きになってしまうようなバグがあります。さらに、デフォル トでは、Kerberos は `-x` オプションを使わない限 りセッションを暗号化してくれません。ssh では、デフォルトですべてを暗号 化してくれます。

ssh はあらゆる場面でとても良く働いてくれます。 ただし、デフォルトで暗号鍵を転送してしまうこと を除けばです。これはつまり、暗号鍵を持った安全なワークステーショ ンがあって、この暗号鍵で残りのシステムとアクセスできるようになっ ている場合に、安全でないマシンへ ssh 接続を行なうとあなたの暗号鍵を使えてしまうということです。実際の鍵そのものが見えてしまうわけではありませんが、ssh はあなたが login している間、転送用ポートを作ります。攻撃者が安全でないマシンの `root` を破ったら、このポートを使って暗号鍵を取得し、この暗号鍵でロックが外れる他のマシンへのアクセスを得てしまいます。

スタッフのログインには、Kerberos を組み合せた ssh を使用することを勧めます。ssh は、Kerberos 対応機能と一緒 にコンパイルできます。こうすると、見えてしまうかもしれない ssh 鍵をあまりあてにしないで良いようになります。また、それと同時に、Kerberos 経由でパスワードを保護することもできます。ssh 鍵は、安全なマシンからの自動化され

たタスク (Kerberos はこの用途には不向きです) のみに使用するべきです。また、 ssh の設定で鍵転送をしないようにするか、あるいは ssh が `authorized_keys` ファイル中に書くことを許 している `from=IP/DOMAIN` オプションを使用し て、特定のマシンからログインしてきたときのみ鍵が有効であるよう にすることも勧めます。

15.4. DES, MD5 と Crypt

##: Bill Swingle.

訳: 花井 浩之 **<hanai@FreeBSD.org >**, 12 September 1996.

訳改訂: 日野 浩志 **<hino@ccm.cl.nec.co.jp >**, 12 March 2001.

Unix システムにおけるすべてのユーザは、そのアカウントに対応し た一つのパスワードを持っています。それらのパスワードはユーザ本人 と本当のオペレーティングシステムのみが知っているべきであるという ことは明らかでしょう。それらのパスワードを秘密に保っておくために、パスワードは「一方向ハッシュ」として知られる方式で暗 号化されます。一方向ハッシュとは、簡単に暗号化はできるが解読は難 しいという方法です。言葉を換えると、先ほど明らかであると書いたの は実は正しくないのです: オペレーティングシステム自身は 本当はパスワードを知らないのです。その代わりに 暗号化された形でのみパスワードを知っていま す。「素のテキスト」としてパスワードを得る唯一の方法は、 可能な限りのパスワード空間を検索するという力任せの方法です。

不幸なことに、Unix が生まれようとしているときにパスワードを 安全な形で暗号化できる方式は DES (Data Encryption Standard) に基づいたものだけでした。このことは米国に住んでいるユーザにとって は大して問題ではありませんでしたが、DES のソースコードを米国外に 輸出することはできないという問題がありました。そのために、FreeBSD は、米国の法律を守ることと、未だに DES を使っていた他の Unix 一族との互換性を保つこととを両立する方法を探し出す必要がありました。

その解決方法は、米国のユーザは DES のライブラリをインストー ルして DES を使用できるが、米国外のユーザは国外に輸出可能な他の ひとつの暗号化方式を使用することができる、というように暗号化ライ ブラリを分割することでした。これが FreeBSD がデフォルトの暗号化 方式として MD5 を使うようになったいきさつです。MD5 は DES よりも より安全であると考えられているため、DES をインストールする一番の 理由は互換性を保つためといえます。

15.4.1. 暗号化機構を理解する

FreeBSD 4.4 の前までは、`libcrypt.a` は暗号化に使われるライブラリへのシンボリックリンクでした。FreeBSD 4.4 で `libcrypt.a` は設定可能なパスワード認証ハッシュライブラリを提供するようになりました。 現在のところ、このライブラリは DES, MD5 および Blowfish ハッシュ関数に対応しています。デフォルトでは、FreeBSD はパスワードの暗号化に MD5 を利用します。

FreeBSD がどの暗号化方式を使うようにセットアップされている かを判断するのは簡単です。`/etc/master.passwd` ファイルの中の暗号化さ れたパスワードを調べてみるのが一つの方法です。MD5 ハッシュで暗 号化されたパスワードは、DES ハッシュで暗号化されたパスワードよ りも長く、`$1$` という文字で始まるという特徴を持っています。 `$2$` で始まるパスワードは、Blowfish ハッシュ関数で暗号化されています。 DES のパスワードはこ れといって識別可能な特徴は持っていませんが、MD5 のパスワードよりは短く、そして `$` という文字を含ま ない 64 文字のアルファベットを使って表現されているので、比較的 短い文字列でドル記号で始まっていないものはおそらく DES のパスワードでしょう。

新規パスワードがどちらのパスワード形式になるかは、 `/etc/login.conf` の中の 「passwd_format」 ログインケーパビリティによって制御されます。 その値としては、「des」、「md5」または「blf」を設定することができます。ログインケーパビリティに関するより詳細な情報は、login.conf(5) マニュアルページをご覧ください。

15.5. S/Key

S/Key は一方向ハッシュ関数を基にしたワンタイムパスワード方式 です。FreeBSD では、互換性のために MD4 ハッシュを用いていますが 他のシステムでは MD5 や DES-MAC を用いてます。S/Key は、バージョン1.1.5 以降のすべての FreeBSD に含まれていますし、FreeBSD 以外 の数多くのシステムの上でも利用されています。S/Key は Bell Communications Research, Inc. の登録商標です。

FreeBSD バージョン 5.0 以降では、S/Key は機能的に同等な OPIE (Onetime Passwords In Everything) で置き換えられました。OPIE はデフォルトでは MD5 ハッシュを使用します。

以下の説明では、三種類の異なる「パスワード」が使われます。 まず一つ目は、あなたが普段使っている普通の Unix スタイルの、もしくは Kerberos のパスワードです。ここではこれを 「Unix パスワード」 と呼ぶことにします。二つ目は、S/Key の `key` プログラム、または OPIE の `opiekey` プログラムによって生成され、 `keyinit` または `opiepasswd` プログラムとログインプロンプトが受け付けるパスワードです。 ここではこれを 「ワンタイムパスワード」 と呼ぶことにします。三つ目のパスワードは、 `key`/`opiekey` (と場合により `keyinit`/`opiepasswd`) プログラムに対してユーザが入力する秘密のパスワードで、ワンタイムパスワードを生成するのに使われます。ここではこれを 「秘密のパスフレーズ」 もしくは単に 「パスフレーズ」 と呼ぶことにします。 (訳注: ユーザが頭の中だけにしまっておくべきものが、この秘密のパスフレーズです。なお、原文ではこれを「password」と表記していますが、混乱を避けるために訳文ではすべて「秘密のパスフレーズ」に統一しています)。

秘密のパスフレーズは、Unix パスワードと何の関連性もありません。両者を同一に設定することは可能ですが、お奨めしません。Unix パスワードは長さが 8 文字に制限されています (訳注: FreeBSD で DES を導入していない場合はもっと長いパスワードも認識

されます)。これに対し、S/Key や OPIE では秘密のパスフレーズを好きなだけ長くすることができます (訳注: 実装上、key コマンドなどの バッファ長で制限されてしまう可能性があります。200 文字程度に押 えておいた方がよいでしょう :-)。6 語から 7 語からなるパスフレーズがふつうです。ほとんどの部分で、S/Key や OPIE システムは Unix のパスワードシステムと完全に独立して動作するようになっています。

パスフレーズに加え、S/Key や OPIE システムにとって重要な 2 種類のデータがあります。一つは「シード (seed: 種)」または「キー (key: 鍵)」と呼ばれるもので、2 つの文字と 5 つの数字で構成されます。もう一つは「シーケンス番号 (iteration count)」で、1 から 100 までの整数です。S/Key はここまで に述べたデータを利用してワンタイムパスワードを生成します。その方 法は、まずシードと秘密のパスフレーズを連結し、それに対してシーケ ンス番号の回数だけ MD4/MD5 ハッシュを繰り返し計算します。そしてその結果を 6 つの短い英単語に変換します。認証システム (一次的には PAM) は、前回最後に受け付けたワンタイムパスワードを記録しています。そして、その前回 のワンタイムパスワードと、ユーザが入力したワンタイムパスワードを 1 回ハッシュ関数にかけた結果とが一致した場合に、このユーザは認証 されます。一方向ハッシュ関数を使っているので、もし正しく認証され たワンタイムパスワードが一回盗聴されたとしても、次回以降に使われ る複数のワンタイムパスワードを生成することは不可能です。シーケ ンス番号はログインが成功するたびに一つずつ減らされて、ユーザとロ グインプログラムの間で同期が取られます。シーケンス番号が 1 まで 減ったら、S/Key や OPIE を再度初期化する必要があります。

次に、それぞれのシステムで関連する 3 つのプログラムについて説明します。 key と opiekey プログラムは、シーケンス番号と、シードと、 秘密のパスフレーズを受け付けて、ワンタイムパスワード 1 つ、または一連のワンタイムパスワードの一覧を生成します。 keyinit と opiepasswd プログラムは、それぞれ S/Key と OPIE を初期化するのに使用され、また秘密のパスフレーズ、シーケンス番号やシードを変更するためにも使用されます。 それぞれのプログラムを実行するには、秘密のパスフレーズか、 または、シーケンス番号とシードとワンタイムパスワードの 1 組かの、どちらかを与えます。 keyinfo と opieinfo プログラムは、それぞれに対応する認証ファイル (/etc/skeykeys または /etc/opiekeys) を調べて、プログラムを起動したユーザの現在のシーケンス番号とシードを表示します。

この文書では、4 種類の異なる操作について説明します。1 つ目は、keyinit または opiepasswd を信頼できる通信路上で利用して、 最初にワンタイムパスワードを設定したり、 秘密のパスフレーズやシードを変更する操作です。 2 つ目は、同じことを行うために keyinit または opiepasswd を信頼できない通信路上で利用する操作です。この場合は信頼できる通信路経由の key または opiekey を併用します。3 つ目は、key または opiekey を使い、信頼できない通信路を通じてログインする操作です。4 番目は、key または opiekey を使って複数のワンタイムパスワードを一気に生成する操作です。ここで生成した複数のワンタイムパスワードは、メモしたり印刷したりして携帯し、信頼できる通信路が一切ないところで利用することができます。(訳注: ワンタイムパスワードを記録した紙をなくさないこと! 電話番号や IP アドレス、ユーザ名を一緒にメモしていたら最悪です!!)

15.5.1. 信頼できる通信路での初期化

信頼できる通信路 (たとえばあるマシンのコンソール画面や、ssh を使っている時など) を利用しているときに、S/Key を初めて初期化 すること、S/Key の秘密のパスフレーズを変更すること、またはシー ドを変更すること、をおこなうことができます。そのためには、まず あなた自身がログインし、`keyinit` コマンドを 以下のようにパラメータなしで実行します。

```
% keyinit
Adding unfurl:
Reminder - Only use this method if you are directly connected.
If you are using telnet or rlogin exit with no password and use ↺
keyinit -s.
        ) `keyinit' ##################
        ) ## - ##########################
        ) ######## telnet # rlogin #############
        ) ############################
        ) keyinit -s ########
Enter secret password:
Again secret password:

ID unfurl s/key is 99 to17757
DEFY CLUB PRO NASH LACE SOFT
```

OPIE では `opiepasswd` が代わりに使われます。

```
% opiepasswd -c
[grimreaper] ~ $ opiepasswd -f -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you ↺
are using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
        ) `opiepasswd' ##################
        ) ################################
        ) ##############telnet, xterm ###########
        ) ##########^C ####################
        ) ########opiepasswd # -c #################
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:
ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED
```

`Enter new secret pass phrase:` または `Enter secret password:` というプロンプトに対して、あなたが考えた秘密のパスフレーズを入力します。このパスフ レーズはログインするときに使うものではなく、ログインするときに 使うワンタイムパスワードを生成するために使うものであることを覚 えておいてください。「ID」から始まる行は、1 回分のパラメータで、あなたのログイン名とシーケンス番号とシードです。(訳注: `keyinit` コマンドは 次回にログインするときに使えるパラメータを参考の

ためにここで表示します)。システムにログインするときには、システム側が自動的にこれらのパラメータを表示してくれますから、これらのパラメータを 覚えておく必要はありません。最後の行が、今述べたパラメータと入力 された秘密のパスフレーズから計算されたワンタイムパスワードです。 この例を実行した後、次にログインするときに打ち込むべきワンタイ ムパスワードがこれです。

15.5.2. 信頼できない通信路での初期化

信頼できない通信路を使って秘密のパスフレーズを初期化または変更するためには、それとは別に `key` または `opiekey` プログラムを実行するための信頼できる通信路を用意しておく必要があります。 たとえばそれは、あなたが信頼できる Macintosh のデスクアクセサリや信頼できるマシンのシェルプロンプトだったり するでしょう。(訳注: ここでの通信路とはマシンそのものになりま す。信頼できるマシンとは、信頼できる人がしっかり管理しているマ シンということです)。他に準備しておくものとして、シーケンス番 号 (100 は適切な値といえるでしょう) と、場合によっては自分で考 えた、またはランダムに生成されたシードがあります。(あなたが S/Key を初期化しようとしているマシンへの) 信頼できない通信路を 使うときには、`keyinit -s` コマンドを以下のよ うに使用します。

```
% keyinit -s
Updating unfurl:
Old key: to17758
Reminder you need the 6 English words from the key command.
 ) `keyinit' ##################
 ) ## - skey ######### 6 ############
Enter sequence count from 1 to 9999: 100
Enter new key [default to17759]:
s/key 100 to 17759
s/key access password:
s/key access password:CURE MIKE BANE HIM RACY GORE
```

OPIE では、`opiepasswd` を使います。

```
% opiepasswd

Updating unfurl:
You need the response from an OTP generator.
 ) `opiepasswd' ##################
 ) OTP #################
Old secret pass phrase:
        otp-md5 498 to4268 ext
        Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
        otp-md5 499 to4269
        Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

デフォルトのシード (keyinit プログラム は困ったことにこれを key と呼んでいるのですが、混乱しないよう注意してください) で構わなければ、Return を押してください。次に、アクセスパスワードを入れる前に、あらかじめ用意しておいた信頼できる通信路(信頼できるマシンや信頼でき る S/Key デスクアクセサリなど) へ移って、先ほどと同じパラメータ を入力します。

```
% key 100 to17759
Reminder - Do not use this program while logged in via telnet or ↺
rlogin.
Enter secret password: <#########>
CURE MIKE BANE HIM RACY GORE
```

OPIE では、

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

ここで信頼できない通信路の方に戻って、 生成されたワンタイムパスワードをコピーして対応するプログラムに入力します。

15.5.3. ワンタイムパスワードを一つ生成する

S/Key または OPIE を初期化したら、ログイン時には以下のようなプロンプトが出てくるでしょう。

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <####>
s/key 97 fw13894
Password:
```

OPIE については、

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <username>
otp-md5 498 gr4269 ext
```

```
Password:
```

ここでは表示していませんが、S/Key と OPIE のプロンプトには便利な機能が備わっています。 パスワードプロンプトに対して、何も入力せずに Return を押すとエコーモードに切り替わります。つまりタイプした文字がそのまま見えるようになるのです。これは、紙に印刷していたりするワンタイムパスワードを 手で入力しなければならない場合に特に役立つ機能です。

次に、このログインプロンプトに対して入力するワンタイムパスワードを生成しなければなりません。 これは、key または opiekey プログラムを使える信頼できるマシン上で行わなければなりません。 (これらのプログラムには DOS や Windows, MacOS 版があります)。 どちらも、コマンドラインからシーケンス番号とシードを指定しなければなりません。 ログインしようとしているマシンのログインプロンプトから直接カットアンドペーストすると楽でしょう。

信頼できるシステムで

```
% key 97 fw13894
Reminder - Do not use this program while logged in via telnet or ↺
rlogin.
Enter secret password:
WELD LIP ACTS ENDS ME HAAG
```

OPIE では

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

ここでワンタイムパスワードが得られました。ログインを続けましょう。

```
login: <username>
s/key 97 fw13894
Password: <return to enable echo>
s/key 97 fw13894
Password [echo on]: WELD LIP ACTS ENDS ME HAAG
Last login: Tue Mar 21 11:56:41 from 10.0.0.2 ...
```

15.5.4. 複数のワンタイムパスワードを生成する

都合によっては、信頼できるマシンや信頼できる通信路が一切確　保できないようなところで S/Key を使う必要があるでしょう。この ような場合には、key コマンドを使って複数の ワンタイムパスワードをあらかじめ一気に生成し、紙に印刷して携帯 していくことができます。たとえば

```
% key -n 5 30 zz99999
```

```
Reminder - Do not use this program while logged in via telnet or ↺
rlogin.
Enter secret password: <#########>
26: SODA RUDE LEA LIND BUDD SILT
27: JILT SPY DUTY GLOW COWL ROT
28: THEM OW COLA RUNT BONG SCOT
29: COT MASH BARR BRIM NAN FLAG
30: CAN KNEE CAST NAME FOLK BILK
```

`-n 5` という引数によって 5 個のワンタイム パスワードを順に生成します。ここで `30` は、最 後のシーケンス番号となるべき数字です。出力は普通に使う順番とは 逆に出力されていることに注意してください (訳注: 一番最初に使うワンタイムパスワードは一番最後に出力され たものです)。この結果をカットアンドペーストして `lpr` コマンドを使って印刷すると よいでしょう。もしあなたがセキュリティに偏執するなら、この結果を紙と鉛筆を使っ て手で書き移した方がよいかもしれません。ここで、出力の各行はシー ケンス番号とそれに対応する一回分のワンタイムパスワードです。消費済みの ワンタイムパスワードの行をペンで消していくと便利で しょう。

15.5.5. Unix パスワードの利用を制限する

設定ファイル `/etc/skey.access` を使って Unix パスワードの利用を制限することができます。この場合の判 断基準として、ログインを受け付ける際のホスト名、ユーザ名、端末 のポート、IP アドレスなどが利用できます。この設定ファイルの詳 細に関してはマニュアル skey.access(5) をご覧ください。マ ニュアルにはこの機能に関わるセキュリティについて、いくつかの警 告が記述してあります。この機能を使ってセキュリティを高めようと するのならば絶対にこのマニュアルを読んでください。

もし `/etc/skey.access` ファイルが存在 しないならば (FreeBSD のデフォルト状態ではそうです)、すべての ユーザが Unix パスワードを利用することができます。逆に、もし ファイルが存在するならば、`skey.access` ファ イルに明示的に記述されていない限り、すべてのユーザは S/Key の 利用を要求されます。どちらの場合においても、そのマシンのコンソー ルからはいつでも Unix パスワードを使ってログインすることが可能 です。

以下によく使われるであろう三種類の設定を含む設定ファイルの 例を示します。

```
permit internet 192.168.0.0 255.255.0.0
permit user fnord
permit port ttyd0
```

はじめの行 (`permit internet`) で、telnet などで接続するときの IP のソースアドレス (注意: これは偽造され るおそれがあります) が特定の値とマスクに一致している場合に、Unix パスワードの利用を許可することを指定しています。この設定自体はセキュリティを高めるための機能ではありません。そうでは なく、ログインの権利を持つ許可されたユーザに対して、現在そのユー ザが使っているネットワークが信頼できないと考えられるので S/Key を使うべきである、ということを気づかせるための機能であると考え てください。

二行目 (`permit user`) によって、ある特定のユーザ、この場合は `fnord`、に対して、いつでも Unix パスワードの利用を許可するように指定しています。一般的にはこの設定をおこなうべきではありません。`key` プログラムがどうしても使えない環境にい る人や、ダム端末しかない環境にいる人、または何度教えても聞く耳 を持たないような人をサポートする必要がある場合にのみ設定をおこ なってください。

三行目 (`permit port`) によって、ある特定 の端末ポートからログインしようとするすべてのユーザに対して Unix パスワードの利用を許可するように指定しています。この設定 はダイヤルアップ回線に対する設定として利用できるでしょう。

15.6. Kerberos

##: Mark Murray.
#########: Mark Dapoz.

訳: 有村 光晴 **<arimura@jp.FreeBSD.org >**.

Kerberosは、サーバのサービスによってユーザが安全に認証を受けられる ようにするための、ネットワークの付加システム及びプロトコルです。リモートログイン、リモートコピー、システム間での安全なファイルのコピ ーやその他のリスクの高い仕事がかなり安全に、そしてこれまでより制御 できるようになります。

以下の文章は、FreeBSD用として配布されているKerberosをセットアップ する際のガイドとして読むことができます。しかし、 完全な説明が必要な場合には、マニュアルページを読んだ方がよい でしょう。

15.6.1. Kerberos のインストール

Kerberos は選択が任意な FreeBSD のコンポーネントです。もっとも簡単なインストール方法は、FreeBSD のインストール時に sysinstall で 'krb4' または 'krb5' 配布物を選択することです。そうすると、Kerberos の 'eBones' (KerberosIV) または 'Heimdal' (Kerberos5) 実装がインストールされます。これらの実装が入っているのは、これがアメリカ合衆国およびカナダの外で開発されたものであるため、アメリカ合衆国からの暗号ソフトウェアの輸出が制限されていた時代でも アメリカ合衆国およびカナダ以外の国に住んでいるシステム所有者の手に入るものだったからです。

ほかに、MIT で実装された Kerberos が Ports Collection の security/krb5 から利用できます。

15.6.2. 初期データベースの作成

この作業はKerberosサーバだけでおこないます。まず、古いKerberosの データベースが存在しないことを確認してください。ディレクトリ`/etc/kerberosIV` に移って、次のファイルだけが 存在することをチェックします。

```
# cd /etc/kerberosIV
# ls
README          krb.conf        krb.realms
```

もし他のファイル（principal.* や master_key）が 存在する場合には、 kdb_destroy というコマンドで古い Kerberosデータベースを消してください。Kerberosが走っていなければ、単に余計なファイルを消せばよいです。

まず、krb.conf と krb.realms を編集してKerberosの 管理領域 (realm) を定義してください。 ここでは管理領域が EXAMPLE.COM で、サーバ名が grunt.example.com であるとします。krb.conf というファイルを次のように編集してください。

```
# cat krb.conf
EXAMPLE.COM
EXAMPLE.COM grunt.example.com admin server
CS.BERKELEY.EDU okeeffe.berkeley.edu
ATHENA.MIT.EDU kerberos.mit.edu
ATHENA.MIT.EDU kerberos-1.mit.edu
ATHENA.MIT.EDU kerberos-2.mit.edu
ATHENA.MIT.EDU kerberos-3.mit.edu
LCS.MIT.EDU kerberos.lcs.mit.edu
TELECOM.MIT.EDU bitsy.mit.edu
ARC.NASA.GOV trident.arc.nasa.gov
```

この例にあるような他の管理領域は、実際には必要ありません。この例は複数の管理領域を認識する方法を示したものですので、これらの行は含めなくても結構です。

1行目はこのシステムが動いている管理領域の名前です。 他の行は管理領域とホスト名のエントリです。 行の1つめの単語が管理領域で、2つめがその管理領域の中で「鍵配布センター」(Key Distribution Center) として働くホスト名です。ホスト名の次に admin server と書いてある場合には、そのホストが 管理データベースサーバ (Administrative Database Server) も提供することを意味します。これらの単語について詳しく知りたい場合には Kerberos のマニュアルページをご覧ください。

ここで、EXAMPLE.COM という管理領域に grunt.example.com およびその他の .example.com ドメインのすべてのホストを追加しなければなりません。krb.realms は次のようになります。

```
# cat krb.realms
grunt.example.com EXAMPLE.COM
.example.com EXAMPLE.COM
.berkeley.edu CS.BERKELEY.EDU
.MIT.EDU ATHENA.MIT.EDU
.mit.edu ATHENA.MIT.EDU
```

もう一度注意しますが、他の管理領域を書く必要はありません。 これらは複数の管理領域を認識できるようにマシンを設定する方法を　示した例ですので、これらの行は消して構いません。

1行目は名前をつけた管理領域に 特定の システムを含めるための ものです。 残りの行は名前をつけた管理領域にサブドメインのデフォルトの　システムを含めるためのものです。

これでデータベースを作成する準備ができました。この操作はKerberos サーバ (鍵配布センター) を起動するだけです。kdb_initコ マンドを次のように実行してください。

```
# kdb_init
Realm name [default  ATHENA.MIT.EDU -]:  EXAMPLE.COM
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.

Enter Kerberos master key:
```

ここで鍵を保存して、ローカルのマシンにあるサーバが取り出せるように します。それにはkstashコマンドを使用します。

```
# kstash

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
```

これで暗号化されたマスタパスワードが /etc/kerberosIV/master_key　に保存されました。

15.6.3. すべてが動くようにするための設定

Kerberosを導入する　それぞれの　システムのデータベースに、2つ　のprincipal　(主体名)　を追加する必要があります。その名前は　kpasswdとrcmdです。 これら2つのprincipalは、個々　のシステムにおいて、 システム名と同じ名前のインスタンスと組にして作成 されます。

これらの kpasswd と rcmd というデーモンによって、他の システムからKerberosのパスワードを変更したり、rcpや rlogin, rshといったコマンドを実行したりできるよ うになります。

それでは実際にこれらのエントリを追加しましょう。

```
# kdb_edit
Opening database...

Enter Kerberos master key:
```

```
Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name:  passwd
Instance:  grunt

<Not found>, Create [y] ?  y

Principal: passwd, Instance: grunt, kdc_key_ver: 1
New Password:                    <---- ####RANDOM##########
Verifying password

New Password:                    <---- ####RANDOM##########

Random password [y] ?  y

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 -] ?
Max ticket lifetime (*5 minutes) [ 255 -] ?
Attributes [ 0 -] ?
Edit O.K.
Principal name:  rcmd
Instance:  grunt

<Not found>, Create [y] ?

Principal: rcmd, Instance: grunt, kdc_key_ver: 1
New Password:                    <---- ####RANDOM##########
Verifying password

New Password:                    <---- ####RANDOM##########

Random password [y] ?

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 -] ?
Max ticket lifetime (*5 minutes) [ 255 -] ?
Attributes [ 0 -] ?
Edit O.K.
Principal name:                   <---- #############
```

15.6.4. サーバファイルの作成

次に、各マシンにおけるサービスを定義している、すべてのインスタンスを展開します。これには ext_srvtab というコマンドを使用します。このコマンドで作成されるファイルは、Kerberos の各クライアントの /etc/kerberosIV ディレクトリに安全な方法で コピーまたは移動する必要があります。 このファイルはそれぞれのサーバとクライアントに存在しなければならず、また Kerberos の運用において重要なものです。

```
# ext_srvtab grunt

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Generating 'grunt-new-srvtab'....
```

このコマンドは一時的なファイルを作成するだけです。ファイル名をすべ てのサーバが読めるような srvtab という名前に変更しな ければなりません。mvコマンドを用いてシステムの場所に移動 してください。

```
# mv grunt-new-srvtab srvtab
```

そのファイルがクライアントに配るためのもので、 ネットワークが安全で はないと思われる場合には、 client-new-srvtab を移動 可能なメディアにコピーして物理的に安全な方法で運んでください。クラ イアントの/etc/kerberosIV ディレクトリで、名前を srvtabに変更し、modeを600にするのを忘れないでください。

```
# mv grumble-new-srvtab srvtab
# chmod 600 srvtab
```

15.6.5. データベースへのユーザの追加

ここで、 ユーザのエントリをデータベースに追加する必要があります。 始めに、 ユーザjaneのエントリを作成してみましょう。 kdb_edit を用いて次のように作成してください。

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.
```

```
Principal name: jane
Instance:

<Not found>, Create [y] ? y

Principal: jane, Instance: , kdc_key_ver: 1
New Password:                   <---- ################
Verifying password

New Password:                   <---- ################

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 -] ?
Max ticket lifetime (*5 minutes) [ 255 -] ?
Attributes [ 0 -] ?
Edit O.K.
Principal name:                 <---- ############
```

15.6.6. すべてのテスト

まず始めにKerberosデーモンを起動する必要があります。 /etc/rc.conf ファイルを正しく編集してあれば、マシンを再 起動することでに自動的にデーモンが起動します。 これはKerberosサー バでのみ必要です。 Kerberosクライアントは/etc/kerberosIV から必要なものを自動的に入手します。

```
# kerberos &
Kerberos server starting
Sleep forever on error
Log file is /var/log/kerberos.log
Current Kerberos master key version is 1.

Master key entered.  BEWARE!

Current Kerberos master key version is 1
Local realm: EXAMPLE.COM
# kadmind -n &
KADM Server KADM0.0A initializing
Please do not use 'kill -9' to kill this job, use a
regular kill instead

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
```

さあ、これで上で作成した jane というIDのチケットを kinitコマンドで得ることができます。

```
% kinit jane
MIT Project Athena (grunt.example.com)
```

```
Kerberos Initialization for "jane"
Password:
```

klist コマンドを用いてトークンを見て、きちんとチケットを持って いるかどうか確認してください。

```
% klist
Ticket file:    /tmp/tkt245
Principal:      jane@EXAMPLE.COM

  Issued           Expires          Principal
Apr 30 11:23:22  Apr 30 19:23:22  krbtgt.EXAMPLE.COM@EXAMPLE.COM
```

passwd コマンドを用いてパスワードを変更して、kpasswd デーモンが Kerberos データベースに対して認証されるかどうかチェックして ください。

```
% passwd
realm EXAMPLE.COM
Old password for jane:
New Password for jane:
Verifying password
New Password for jane:
Password changed.
```

15.6.7. su 特権の追加

Kerberos は root 権限が必要な 各 ユーザに対し、su コマンドのパスワードをユーザ毎に 別のもの として持つことを可能にします。root に su できる権利を与えられた id を追加します。これは、principal に付いている root というインスタンスに よって制御されています。kdb_edit を用いて jane.root というエントリを Kerberosデータベースに作成します。

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name:  jane
Instance:  root

<Not found>, Create [y] ? y

Principal: jane, Instance: root, kdc_key_ver: 1
```

```
New Password:                    <---- #############
Verifying password

New Password:                    <---- ##############

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 -] ?
Max ticket lifetime (*5 minutes) [ 255 -] ?  12 <--- #######
####
Attributes [ 0 -] ?
Edit O.K.
Principal name:                   <---- #############
```

実際にトークンをもらって、ちゃんと働いているかどうか確認しましょう。

```
# kinit jane.root
MIT Project Athena (grunt.example.com)
Kerberos Initialization for "jane.root"
Password:
```

ここで root ユーザの .klogin ファイルにユーザを追加する必要があります。

```
# cat /root/.klogin
jane.root@EXAMPLE.COM
```

su してみましょう。

```
% su
Password:
```

どのトークンを持っているか見てみましょう。

```
# klist
Ticket file: /tmp/tkt_root_245
Principal:      jane.root@EXAMPLE.COM

  Issued           Expires          Principal
May  2 20:43:12  May  3 04:43:12  krbtgt.EXAMPLE.COM@EXAMPLE.COM
```

15.6.8. 他のコマンドの使用

ここまでの例では、jane という principal を root とい うインスタンス付きで作成しました。これはユーザと同じ名前をprincipalとしており、Kerberosのデフォルトの値です; <username>. root という形式の <principal>.<instance> で、必要なエントリが root のホームディレクトリの .klogin ファイルにあれば、<username> が root に su できます。

```
# cat /root/.klogin
jane.root@EXAMPLE.COM
```

同様に、ユーザのホームディレクトリの .klogin ファイルに次の ような行がある場合には

```
% cat ~/.klogin
jane@EXAMPLE.COM
jack@EXAMPLE.COM
```

jane または jack という名前で (前述のkinit によって) 認証されている EXAMPLE.COM という管理領域のユーザ なら誰でもrlogin や rsh, rcp等によってこ のシステム (grunt) のjaneのアカウントまたはファ イルにアクセスできます。

たとえば、jane が他のシステムに Kerberos を用いて login します。

```
% kinit
MIT Project Athena (grunt.example.com)
Password:
% rlogin grunt
Last login: Mon May  1 21:14:47 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
 The Regents of the University of California.   All rights reserved.

FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

次の例では、Jack が同じマシンの Jane のアカウントに login します。jane は .klogin ファイルを前述のように設定しており、Kerberos では jack という principal をインスタンスなしで設定してあります。

```
% kinit
% rlogin grunt -l jane
MIT Project Athena (grunt.example.com)
Password:
Last login: Mon May  1 21:16:55 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
 The Regents of the University of California.   All rights reserved.

FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

15.7. ファイアウォール

##: Gary Palmer # Alex Nash.

訳: 佐伯 隆司 **<saeki@jp.FreeBSD.org >**. 11 November 1996.

ファイアウォールは、インターネットに参加している人はもちろんのこと、プライベートネットワークのセキュリティ向上のための アプリケーションを 探している人にとっても、ますます興味深くなりつつある分野です。このセクションではファイアウォールとは何か、ファイ

アウォールの使用法、そしてファイアウォールを構築するために FreeBSD のカーネルで提供されているファシリティ (機能) の使用法について説明したいと思います。

注記

社内のネットワークと 「巨大かつ信頼のおけない インターネット」との間にファイアウォールを構築することで セキュリティ上のすべての問題が解決できると考える人がいます。ファイアウォールはセキュリティ上の問題を 解決する助けになる場合もありますが、充分な設定がなされていないファイアウォールは、 まったくファイアウォールを 持たない場合よりもセキュリティ上の危険を増大させてしまいます。ファイアウォールにできることは、あなたのシステムにもう一つのセキュリティ層を 追加することだけで、本気でアタックをしかけてくるクラッカーが内部ネットワークに 侵入するのを妨げることはできません。ファイアウォールを侵入不可能と過信して 内部のセキュリティをおろそかにすることは、 単にクラッカーの仕事を少し簡単にするだけでしか ありません。

15.7.1. ファイアウォールとは何か ?

現在インターネットで普通に使用されている ファイアウォールには 二つの異なるタイプがあります。一つは、厳密には パケットフィルタリングルータ と 呼ばれるタイプのものです。これはマルチホームのホストマシン (複数の ネットワークに接続されているマシン) のカーネルが、ある規則にしたがって パケットを転送したりブロックしたりするものです。もう一つは、 proxy (代理) サーバ として知られているタイプのものです。これは、おそらくはマルチホームのホストマシン上で、カーネルによるパケット転送を 禁止して、デーモンにより認証の提供とパケットの転送とを おこなうものです。

二つのタイプのファイアウォールを組み合わせて使用して、 特定のマシン (要塞ホストと呼ばれる) だけが パケットフィルタリングルータを通して内部ネットワークへ パケットを送ることができるよう設定している サイトがしばしば存在します。proxy (代理) サービスは通常の認証機構よりもセキュリティを 強化してある 要塞ホストで動作させます。

FreeBSD は (IPFW として知られる) カーネルパケットフィルタ込みで提供されています。この節の残りでは、このフィルタについて集中して説明します。サードパーティから提供されるソフトウェアを使用することにより、Proxy サーバを FreeBSD 上に構築することができます。しかし、現在入手可能な proxy サーバはたいへんバラエティに富んでいるので、この節でそれらすべてをカバーすることはできません。

15.7.1.1. パケットフィルタリングルータ

ルータとは、二つまたはそれ以上のネットワークの間で パケットの転送をおこなう マシンのことです。パケットフィルタリングルータは、そのカーネルの内部に、一つ一つのパケッ

トをルールリストと比較して 転送するかしないかを決める 特別なコードを持っています。最近の IP ルーティングソフトウェアのほとんどは、内部に パケットのフィルタリングをおこなうためのコードを持っていて、デフォルトでは すべてのパケットを転送するようになっています。 このフィルタを有効にするためには、 パケットの通過を許すべきかどうかを決める ルールを自分で定義する必要があります。

パケットを通すべきか通すべきでないかを決めるために、 パケットヘッダの内容にマッチするものが ルールリストから探されます。マッチするルールが見つかると、ルールアクションが実行されます。ルールアクションには、パケットを捨てる、パケットを転送する、またはパケットの発信元に ICMP メッセージを送り返すというものがあります。ルールの検索は先頭から順番におこなわれ、 通常は最初にマッチしたものだけが 適用されます。そのため、このルールリストは「ルールチェーン」と呼ばれることもあります。

パケットマッチングの基準は使用するソフトウェアに よって異なりますが、通常はパケットの発信元 IP アドレス、宛先 IP アドレス、発信元ポート番号、宛先ポート番号 (ポート番号はポートをサポートするプロトコルの場合のみ)、パケットタイプ (UDP, TCP, ICMP など) に基づくルールを指定することができます。

15.7.1.2. Proxy サーバ

Proxy サーバとは通常のシステムデーモン (telnetd, ftpd など) を 特別なサーバで置き換えたマシンのことです。これらのサーバは、通常は中継をおこなって特定方向への接続だけを許すため、proxy サーバ と呼ばれます。(たとえば) proxy telnet サーバをファイアウォールホストで走らせておきます。外部からユーザがファイアウォールに対して telnet を実行すると、proxy telnet サーバが応答して、 何らかの認証機構を実行します。これを通過した後で、 内部ネットワークへのアクセスがおこなえるように なるのです。(内部ネットワークからの信号は proxy サーバがかわりに受け取り、外へ向けて送り出します)。

Proxy サーバは通常、普通のサーバより堅固に構築されていて、しばしば「使い捨て」パスワードシステムなどを含む、多様な認証機構を持っています。「使い捨て」パスワードシステムとは、 どういうものなのでしょうか。仮に誰かが何らかの方法で、 あなたが使用したパスワードを手に入れたとします。しかし、一度使用したことで、そのパスワードは既に無効になっているのです。ですから、 そのパスワードをもう一度使用したとしても、あなたのシステムへ アクセスすることはできないというわけです。これらのサーバは中継をおこなうだけで、実際のところサーバホスト自身への アクセスをユーザに許してはいません。そのため、何者かがセキュリティシステムに 侵入用の裏口を取り付けることは、より困難になっています。

proxy サーバはアクセス制限の方法をいくつも持っていて、特定のホスト だけがサーバへのアクセス権を得ることができるように なっていることがあります。そして目的のマシンと通信できるユーザを制限するように 設定することもできます。もう一度言いますが、どんなファシリティ (機能) が使えるかは、どんな proxy サービスをおこなうソフトウェアを選ぶかに大きく 依存します。

15.7.2. IPFW で何ができるか

FreeBSD とともに配布されている IPFW は、カーネル内部にあってパケットのフィルタリングとアカウンティングをおこなうシステムであり、ユーザ側のコントロールユーティリティである ipfw(8) を含んでいます。ルーティングの決定をおこなう際に、これらは互いに協力して、カーネルで使用されるルールを定義したり、現在使用されているルールを問い合わせたりすることができます。

IPFW は互いに関連する二つの部分からなっています。ファイアウォールセクションは パケットフィルタリングをおこないます。また、IP アカウンティングセクションはファイアウォールセクションのものと 似たルールに基づいてルータの使用を追跡します。これにより、(たとえば) 特定のマシンからルータへのトラフィックがどのくらい 発生しているか調べたり、どれだけの WWW (World Wide Web) トラフィックが フォワードされているかを知ることができます。

IPFW は、ルータではないマシンにおいても入出力コネクションの パケットフィルタリングのために使用することができるように設計されています。これは一般的な IPFW の使用法とは異なる特別な使い方ですが、こういった状況でも同じコマンドとテクニックが使用されます。

15.7.3. FreeBSD で IPFW を有効にする

IPFW システムの中心となる部分はカーネル内部にあります。そのため、どのファシリティ (機能) を必要とするかによって、1 つまたは複数のオプションをカーネルコンフィグレーションファイルに追加し、カーネルを再コンパイルする必要があるでしょう。カーネルの再コンパイル方法の詳細については、「カーネルのコンフィグレーション」(9章FreeBSD カーネルのコンフィグレーション) をご覧ください。

現在、IPFW に関係するカーネルコンフィグレーションオプションは 三つあります。

`options IPFIREWALL`
: パケットフィルタリングのためのコードを カーネルに組み込みます。

`options IPFIREWALL_VERBOSE`
: syslogd(8) を通じて パケットのログを取るためのコードを有効にします。フィルタルールでパケットのログを取るように指定しても、このオプションが指定されていなければ、ログを取ることはできません。

`options IPFIREWALL_VERBOSE_LIMIT=10`
: syslogd(8) を通じて ログを取るパケットの数をエントリ毎に制限します。敵対的な環境においてファイアウォールの 動作のログを取りたいけれど、syslog の洪水によるサービス拒絶攻撃に対し 無防備でありたくないという場合に、このオプションを使用したいと思うことが あるかもしれません。

  チェーンエントリのログが指定された制限数に達すると、そのエントリに関するログ取りは停止されます。ログ取りを再開するには、ipfw(8) ユーティリティを使用して関連するカウンタをリセットする必要があります。

```
# ipfw zero 4500
```

4500 とは、ログ取りを続行したいチェーンエントリの番号です。

注記

以前のバージョンの FreeBSD は `IPFIREWALL_ACCT` というオプションを 持っていました。しかし、ファイアウォールコードがアカウンティングファシリティ (機能) を 自動的に含むようになったため、現在では使用されることはなくなっています。

15.7.4. IPFW の設定

IPFW ソフトウェアの設定は ipfw(8) ユーティリティを通じておこないます。このコマンドの構文は非常に複雑に見えますが、一旦その構造を理解すれば比較的単純です。

このユーティリティでは今のところ四つの異なる コマンドカテゴリが 使用されています: それは追加 / 削除、表示、フラッシュ、およびクリアです。追加 / 削除はパケットの受け入れ、拒絶、ログ取りをどのようにおこなうか というルールを構築するのに使用します。表示はルールリスト (またはチェーン) と (アカウンティング用) パケットカウンタの 内容を調べるのに使用します。フラッシュはチェーンからすべてのエントリを 取り除くのに使用します。クリアは一つまたはそれ以上のアカウンティングエントリを ゼロにするのに 使用します。

15.7.4.1. IPFW ルールの変更

この形式での使用法は:

`ipfw` [-N] コマンド [index] アクション [log] プロトコル アドレス [オプション]

この形式で使用する際に有効なフラグは一つだけです。

-N
: アドレスやサービス名を 文字列に変換して表示します。

コマンド は一意である限り短縮可能です。有効な コマンド は

add
: ファイアウォール / アカウンティングルールリストに エントリを追加します。

delete
: ファイアウォール / アカウンティングルールリストから エントリを削除します。

以前のバージョンの IPFW では、ファイアウォールエントリと パケットアカウンティングエントリが別々に利用されていました。現在のバージョンでは、それぞれのファイアウォールエントリ毎に パケットアカウンティングエントリが備えられています。

index 値が指定されていると、 エントリはチェーン中の指示された位置に置かれます。index 値が指定されて いなければ、エントリは (65535 番のデフォルトルールである パケット拒絶を別にして) 最後のチェーンエントリの index に 100 を足した 位置 (チェーンの最後) に置かれます。

カーネルが `IPFIREWALL_VERBOSE` つきでコンパイルされている場合、`log` オプションはマッチしたルールを システムコンソールに出力させます。

有効な アクション は:

reject
: パケットを捨てます。ICMP ホスト / ポート到達不能パケットを (適切な方を) 発信元へ送ります。

allow
: 通常通りパケットを通過させます。(別名: `pass` および `accept`)

deny
: パケットを捨てます。発信元は ICMP メッセージによる 通知を受けません (そのためパケットが 宛先に到達しなかったように見えます)。

count
: このルールはパケットカウンタを更新するだけで、 パケットを 通過させたり拒絶したりしません。検索は次のチェーンエントリから続けられます。

それぞれの アクション は一意な先頭部分だけでも認識されます。

指定可能な プロトコル は以下の通りです。

all
: 任意の IP パケットにマッチします。

icmp
: ICMP パケットにマッチします。

tcp
: TCP パケットにマッチします。

udp
: UDP パケットにマッチします。

アドレス の指定は:

from *`address/mask`* [*`port`*] to *`address/mask`* [*`port`*] [via *`interface`*]

port はポートをサポートする プロトコル (UDP と TCP) の 場合にだけ指定可能です。

via は必須ではなく、特定のインタフェースを通ってきたパケット だけにマッチするように、IP アドレスまたはローカル IP インタフェースの ドメイン名、またはインタフェース名 (たとえば ed0) を 指定することができます。インタフェースユニット番号はオプションで、ワイルドカードで指定することが できます。たとえば、ppp* はすべてのカーネル PPP インタフェースに マッチします。

address/mask の指定は:

address

または

address /*mask-bits*

または

address :*mask-pattern*

IP アドレスのかわりに有効なホスト名を指定することも可能です。 *mask-bits* はアドレスマスクで上位何ビットを1にするべきかを 示す十進数値です。たとえば次の指定、192.216.222.1/24 はクラス C のサブネット (この場合 192.216.222) の任意のアドレスにマッチするマスクを作成します。*mask-pattern* は与えられたアドレスと 論理 AND される IP アドレスです。キーワード any は「任意の IP アドレス」を指定するために 使用することができます。

ブロックするポート番号は以下のように指定します:

port [, *port* [, *port* […]]]
のように単独のポートまたはポートのリストを指定します。または

port- *port*
のようにポートの範囲を指定します。単独のポートとポートのリストを 組み合わせて指定することも可能ですが、その場合は常に範囲の方を 最初に指定しなければなりません。

使用可能な オプション は:

frag
　データグラムの最初の フラグメントでなければマッチします。

in
　入力途中のパケットであればマッチします。

out
　出力途中のパケットであればマッチします。

ipoptions *spec*
: IP ヘッダが *spec* に指定された カンマで区切られた オプションのリストを含んでいればマッチします。サポートされている IP オプションのリストは: `ssrr` (ストリクトソースルート)、`lsrr` (ルーズソースルート)、`rr` (レコードパケットルート)、そして `ts` (タイムスタンプ) です。特定のオプションを含まないことを指定するには `!` を先頭につけます。

established
: パケットが既に確立されている TCP コネクションの一部であれば (つまり RST または ACK ビットがセットされていれば) マッチします。established ルールをチェーンの最初の方に置くことで、ファイアウォールのパフォーマンスを向上させることが できます。

setup
: パケットが TCP コネクションを確立しようとするものであれば (SYN ビットがセットされ ACK ビットはセットされていなければ) マッチします。

tcpflags *flags*
: TCP ヘッダが *flags* に指定された カンマで区切られたフラグの リストを含んでいればマッチします。サポートされているフラグは、`fin`, `syn`, `rst`, `psh`, `ack` と `urg` です。特定のフラグを含まないことを指定するには `!` を先頭につけます。

icmptypes *types*
: ICMP タイプが *types* リストに 存在していればマッチします。リストはタイプの範囲または個々のタイプを カンマで区切った任意の組合せで指定できます。 一般的に使用されている ICMP タイプは: `0` エコーリプライ (ping リプライ)、`3` 相手先到達不可能、`5` リダイレクト、`8` エコーリクエスト (ping リクエスト)、そして `11` 時間超過 (traceroute(8) で使用されているように、TTL 満了を示すのに使用されます) です。

15.7.4.2. IPFW ルールリストの表示

この形式での使用法は:

`ipfw` [-a] [-t] [-N] l

この形式で使用する際に有効なフラグは三つあります。

-a
: リスト表示の際にカウンタの値も表示します。 このオプションは アカウンティングカウンタの 内容を見る唯一の手段です。

-t
: 各チェーンエントリが最後に マッチした時刻を表示します。この時刻表示は ipfw(8) ユーティリティで使用される入力形式と 互換性がありません。

-N
(可能であれば) アドレスやサービス名を文字列に変換して表示します。

15.7.4.3. IPFW ルールのフラッシュ

チェーンをフラッシュするには:

`ipfw` flush

カーネルに固定されているデフォルトルール (インデックス 65535 番) 以外の、ファイアウォールチェーンの中のすべてのエントリを削除します。デフォルトではすべてのパケットが拒絶されるので、一旦これを実行すると、パケットを許可するエントリがチェーンに追加されるまで、あなたのシステムがネットワークから切り放されてしまいます。そのため、ルールのフラッシュをおこなうときは注意が必要です。

15.7.4.4. IPFW パケットカウンタのクリア

一つまたはそれ以上のパケットカウンタをクリアするためには:

`ipfw` zero [*`index`*]

`index` が指定されていなければ、すべてのパケットカウンタが クリアされます。*`index`* が指定されていれば、特定のチェーンエントリだけが クリアされます。

15.7.5. ipfw についてのコマンドの例

このコマンドは、ホスト `evil.crackers.org` から ホスト `nice.people.org` の telnet ポートへの すべてのパケットを拒絶します。

```
# ipfw add deny tcp from evil.crackers.org to nice.↺
people.org 23
```

次の例は、ネットワーク `crackers.org` (クラス C) 全体から マシン `nice.people.org` (の任意のポート) への 任意の TCP トラフィックを拒絶し、ログを取ります。

```
# ipfw add deny log tcp from evil.crackers.org/24 to ↺
nice.people.org
```

あなたの内部ネットワーク (クラス C のサブネット) に対する X セッションを 張れないようにする場合、以下のコマンドで必要なフィルタリングがおこなえます。

```
# ipfw add deny tcp from any to my.org/28 6000 setup
```

アカウンティングレコードを見るには:

```
# ipfw -a list
```

または短縮形式で

```
# ipfw -a l
```

最後にチェーンエントリがマッチした 時刻を見ることもできます。

```
# ipfw -at l
```

15.7.6. パケットフィルタリングファイアウォールの構築

注記

以下の提案は、ただの提案にすぎません: 必要な処理はそれぞれのファイアウォールで異なるため、 あなた独自の要求にあったファイアウォールを構築する方法を ここで述べることはできないのです。

最初にファイアウォールをセットアップするとき、コントロールされた環境でファイアウォールホストの 設定がおこなえるような テストベンチセットアップが用意できない場合には、カーネルのログ取りを 有効にしてログ取り版のコマンドを使用することを 強くおすすめします。そうすることで、 大した混乱や中断なしに問題となる範囲の特定と処置を 素早くおこなうことができます。 初期セットアップフェーズが完了してからであっても、 アタックの可能性のあるアクセスをトレースしたり、 要求の変化に応じてファイアウォールルールを 変更したりできるので、`deny' に対するログ取りをおこなうことをおすすめします。

注記

`accept` コマンドでログを取っていると、ファイアウォールをパケットが一つ通過する毎に 1 行のログが生成されるため 大量の ログデータが発生します。そのため、大規模な FTP/HTTP 転送などをおこなうと、システムが非常に 遅くなってしまいます。また、パケットが通過するまでにカーネルにより 多くの仕事を要求するため、パケットのレイテンシ (latency) を増加させてしまいます。syslogd もログをディスクに記録するなど、より多くの CPU タイムを 使用し始め、実に容易に `/var/log` が置かれているパーティションを溢れさせてしまう可能性があります。

ファイアウォールは、`/etc/rc.conf.local` か、もしくは `/etc/rc.conf` によって有効化されるべきです。関連マニュアルページには、どのドアノブ (訳注: ポートや IP アドレスなど、ネットワークからの入口を示すもののこと) に手をつければ良いのかに つ

いての説明と、ファイアウォール設定の既定値のリストがあります。もし、設定の既定値を使わない場合には、`ipfw list` とすることで、現在のルールセットを `rc.conf` から読み込める形で ファイルに出力できます。また、`/etc/rc.conf.local` や `/etc/rc.conf` によってファイアウォールを有効化しない場合には、すべての IP インタフェースが設定されるよりも前に、 確実にファイアウォールの有効化が行なわれるようにすることが重要です。

次の問題は、ファイアウォールが実際には何を する べきかです ！ これは外部からそのネットワークへのどんなアクセスを許したいか、また内部から外界へのアクセスを どのくらい許したいかに大きく依存します。いくつか一般的なルールを挙げると:

- 1024 番以下のポートへのすべての TCP 入力アクセスをブロックします。ここは finger, SMTP (mail) そして telnet など、最もセキュリティに敏感な サービスが存在する場所だからです。

- すべての 入力 UDP トラフィックをブロックします。これは UDP を使用しているサービスで有用なものは極めて少ないうえ、 (Sun の RPC と NFS プロトコルのように) 有用なトラフィックであったとしても、通常セキュリティに対する脅威となるためです。UDP はコネクションレスプロトコルであるため、入力 UDP トラフィックを拒絶することは すなわち出力 UDP トラフィックに対する返答をも ブロックすることになるので、このことはそれなりの不利益をもたらします。たとえば外部の archie (prospero) サーバを使用している (内部の) ユーザに とって問題となる可能性があります。もし archie へのアクセスを許したければ、191 番と 1525 番のポートから 任意の UDP ポートへ来るパケットがファイアウォールを通過することを 許可しなければなりません。123 番のポートから来るパケットは ntp パケットで、これも通過の許可を考慮する必要があるもう一つのサービスです。

- 外部から 6000 番のポートへのトラフィックをブロックします。6000 番のポートは X11 サーバへのアクセスに使用されるポートで、セキュリティに対する脅威となりえます。(特に自分のワークステーションで `xhost +` をおこなう癖を持っている人がいればなおさらです)。X11 は実際に 6000 番以降のポートを使用する可能性があるため、通過許可に 上限を定めると、そのマシンで走らせることのできる X ディスプレイの 個数が制限されます。RFC 1700 (Assigned Numbers) で定義されているように、上限は 6063 です。

- 内部のサーバ (たとえば SQL サーバなど) がどのポートを使用するかを チェックします。 それらのポートは通常、上で指定した 1-1024 番の範囲から外れていますので、これらも同様にブロックしておくことは おそらく良い考えです。

これとは別のファイアウォール設定に 関するチェックリストが CERT から 入手可能です。http://www.cert.org/tech_tips/packet_filtering.html

前にも述べたように、これはただの ガイドライン にすぎません。 ファイアウォールでどのようなフィルタルールを使用するかは、 あなた自身が 決めなければなりません。 これまでのアドバイスに従ったにも関わらず、誰かがあなたのネットワークに 侵入してきたとしても、わたしたちは「いかなる」責任もとることはできません。

15.7.7. IPFW のオーバーヘッドと最適化

多くの人が IPFW がどのくらいのオーバヘッドをシステムに加えるかを知りたがっています。この答えは、使っているルールセットとプロセッサのスピードによってほぼ決まります。イーサネットを使っていてルールセットが少ないアプリケーションにとって答えは、「その影響は無視できる程度」です。実際の測定値を見ないと満足できない方は、引き続きお読みください。

次の測定は 486-66 (訳注: Intel 社製 CPU i486, 66MHz のこと) 上で 2.2.5-STABLE を使用して行なわれました (IPFW はその後の FreeBSD のリリースで多少変更されていますが、現在も同程度の速度で動きます)。IPFW には、`ip_fw_chk` ルーチン内でかかる時間を測定して、 1000 パケット毎に結果をコンソールに表示する変更が加えられています。

それぞれ 1000 ずつのルールからなる 2 つのルールセットでテストが行なわれました。1 つ目のルールセットは最悪のケースを見るために、次のルールを繰り返しています。

```
# ipfw add deny tcp from any to any 55555
```

これは、最終的にパケットが (ポート番号から) ルールにマッチしないことがわかるまでに IPFW のほとんどのパケットチェックルーチンが実行されるような、最悪の場合を示します。このルールを 999 個繰り返し並べた後に `allow ip from any to any` がきます。

2 つ目のルールセットは、なるべく早く確認が終了するように書かれたものです。

```
# ipfw add deny ip from 1.2.3.4 to 1.2.3.4
```

このルールでは、発信元の IP アドレスが一致しないので、すぐに確認が終わります。前とおなじように、1000 個目のルールは `allow ip from any to any` です。

前者のパケットあたりのオーバヘッドはおよそ 2.703 ms/packet または 1 つのルールにつき 2.7 マイクロ秒です。したがって、このルールにおけるパケット処理時間の理論的な限界は、毎秒約 370 パケットです。10 Mbps の Ethernet で 1500 バイト程度のパケットサイズを仮定すると、バンド幅の利用効率は 55.5% が限界です。

後者では、それぞれのパケットがおよそ 1.172 ms または、1 つのルールにつき 1.2 マイクロ秒で処理されていました。パケット処理時間の理論的な限界は、毎秒約 853 パケットとなりますので、10 Mbps Ethernet のバンド幅を使い切ることができます。

このテストに使われたルールの数が多過ぎることと、 その性質から、これは実際の状況を反映したものではありません。 これらは上に示したタイミング情報を出すためだけに用いられたものです。効率の良いルールセットを作るためには、次のような事を考えればよいでしょう。

- `established` ルールは TCP トラフィックの大半を処理するため、先頭の方に持ってきてください。このルールの前には `allow tcp` という記述を置かないでください。

- 良く使われるルールを、あまり良く使われないルールよりも前の方に (もちろんファイアウォールの許可設定を変えない範囲で) 持ってきてください。`ipfw -a l` でパケット数の統計を取ることで、どのルールが最もよく使われているかを調べられます。

15.8. OpenSSL

FreeBSD 4.0 では、OpenSSL ツールキットが基本構成の一部に 含まれています。OpenSSL は、Secure Sockets Layer v2/v3 (SSLv2/SSLv3) や Transport Layer Security v1 (TLSv1) ネットワークセキュリティプロトコルと同様の 多目的な暗号化ライブラリを提供します。

しかしながら、OpenSSL に含まれるアルゴリズムのひとつ (特に IDEA) は、合衆国内、その他の地域において、 特許により保護されています。そのため、 無制約な利用は許されません。IDEA は FreeBSD の OpenSSL 配布に含まれていますが、デフォルトではコンパ イルされません。もし IDEA を使いたいなら、そしてあなたがそのライ センス条項に合致するなら、`/etc/make.conf` の中の `MAKE_IDEA` スイッチを有効にして、`make world` でソースをリビルドしてください。

現在は RSA アルゴリズムはアメリカとその他の国で自由に利用で きます。以前は特許により保護されていました。

15.8.1. ソースコードのインストール

OpenSSL は `src-crypto` と `src-secure` cvsup コレクションの一部です。FreeBSD のソースコードの取得と更新の詳細は、FreeBSD の入手の項を参照して下さい。

15.9. IPsec

##: Yoshinobu Inoue.

訳: 日野 浩志 **<hino@ccm.cl.nec.co.jp >**, 14 March 2001.

終端文字

この節の、また他の節を通して、末尾に「^D」が置かれている例があることに気づかれるでしょう。 これは、Control キーを押しながら D キーを押すことを意味しています。ほかによく使われる文字は

「^C」で、Control キーを押しながらCを押すことを意味しています。

ヒント

FreeBSD の IPsec 実装について説明した HOWTO は、他に `http://www.daemonnews.org/200101/ipsec-howto.html` と `http://www.freebsddiary.org/ipsec.php` があります。

IPsec 機構は、IP 層とソケット層に対して安全な通信を提供します。この節ではその使い方を説明します。実装の詳細に関しては The Developers' Handbook を参照してください。

現在の IPsec の実装は、トランスポートモードとトンネルモードの両方に対応しています。しかし、トンネルモードにはいくつかの制限事項があります。http://www.kame.net/newsletter/ にはより総合的な例が載っています。

ここで述べる機能を利用するには、以下のオプションをカーネルコンパイル時に指定する必要があることにご注意ください。

```
options         IPSEC           #IP security
options         IPSEC_ESP       #IP security (crypto; define w/
IPSEC)
```

15.9.1. IPv4 におけるトランスポートモードの例

ホスト A (`10.2.3.4`) とホスト B (`10.6.7.8`) との間に安全なチャネルを配置するために、セキュリティアソシエーションを設定しましょう。ここでは、少し込み入った例を示します。ホスト A からホストB へは old AH のみを使います。ホスト B からホスト A へは new AH と new ESP を組み合わせます。

ここで「AH」/「new AH」/「ESP」/「new ESP」に対応するアルゴリズムを決めないといけません。アルゴリズムの名前を知るには、setkey(8) マニュアルページをご覧ください。ここでは、AH に MD5 を、new AH には new-HMAC-SHA1 を、new ESP には 8 バイト IV の new-DES-expIV を選びました。

鍵長はそれぞれのアルゴリズムに大きく依存します。たとえば、MD5 では鍵長は 16 バイトでなければなりませんし、new-HMAC-SHA1 では 20 バイトでなければなりませんし、new-DES-expIV では 8 バイトでなければなりません。ここではそれぞれ「MYSECRETMYSECRET」,「KAMEKAMEKAMEKAMEKAME」,「PASSWORD」とします。

次に、それぞれのプロトコルに対して SPI (セキュリティパラメータインデックス: Security Parameter Index) を割り当てます。3 種類のセキュリティヘッダ (ホスト A からホスト B に 1 つ、ホスト B から ホスト A に 2 つ) を生成するので、この安全なチャネルには 3 つの SPI が必要になることに注意してください。さらに、SPI は 256 以上でなければならないことにも注意してください。ここではそれぞれ 1000, 2000, 3000 を割り当てます。

```
           (1)
### A ------> ### B

(1)PROTO=AH
 ALG=MD5(RFC1826)
 KEY=MYSECRETMYSECRET
 SPI=1000

           (2.1)
### A <------ ### B
         <------
           (2.2)

(2.1)
PROTO=AH
 ALG=new-HMAC-SHA1(new AH)
 KEY=KAMEKAMEKAMEKAMEKAME
 SPI=2000

(2.2)
PROTO=ESP
 ALG=new-DES-expIV(new ESP)
  IV length = 8
 KEY=PASSWORD
 SPI=3000
```

次に、セキュリティアソシエーションを設定しましょう。ホスト　A　とホスト　B　の両方で、setkey(8) を実行します。

```
# setkey -c
    add 10.2.3.4 10.6.7.8 ah-old  1000 -m transport -A ↺
keyed-md5 "MYSECRETMYSECRET" -;
    add 10.6.7.8 10.2.3.4 ah  2000 -m transport -A hmac-
sha1 "KAMEKAMEKAMEKAMEKAME" -;
    add 10.6.7.8 10.2.3.4 esp 3000 -m transport -E des-
cbc "PASSWORD" -;
    ^D
```

実際には、セキュリティポリシのエントリが定義されるまでは IPsec による通信は行われません。この例の場合、両方のホストを設定する必要があります。

```
A #:
```

```
# setkey -c
    spdadd 10.2.3.4 10.6.7.8 any -P out ipsec
 ah/transport/10.2.3.4-10.6.7.8/require -;
    ^D

B #:

# setkey -c
    spdadd 10.6.7.8 10.2.3.4 any -P out ipsec
 esp/transport/10.6.7.8-10.2.3.4/require -;
    spdadd 10.6.7.8 10.2.3.4 any -P out ipsec
 ah/transport/10.6.7.8-10.2.3.4/require -;
    ^D


  ### A ------------------------------------> ### B
  10.2.3.4                                     10.6.7.8
         |                                    |
         ========== old AH keyed-md5 ==========>

         <========== new AH hmac-sha1 ===========
         <========== new ESP des-cbc ============
```

15.9.2. IPv6 におけるトランスポートモードの例

IPv6 を使ったもう一つの例。

ホスト-A とホスト-B 間の TCP ポート番号 110 番の通信には、ESP トランスポートモードが推奨されます。

```
             ============ ESP ============
            |                            |
         ###-A                       ###-B
         fec0::10 -------------------- fec0::11
```

暗号化アルゴリズムは blowfish-cbc で、その鍵は「kamekame」、認証アルゴリズムは hmac-sha1 で、その鍵は「this is the test key」とします。ホスト-A の設定は次のようになります。

```
# setkey -c <<EOF
    spdadd fec0::10[any] fec0::11[110] tcp -P out ipsec
 esp/transport/fec0::10-fec0::11/use -;
    spdadd fec0::11[110] fec0::10[any] tcp -P in ipsec
 esp/transport/fec0::11-fec0::10/use -;
    add fec0::10 fec0::11 esp 0x10001
 -m transport
 -E blowfish-cbc "kamekame"
```

```
 -A hmac-sha1 "this is the test key" -;
    add fec0::11 fec0::10 esp 0x10002
 -m transport
 -E blowfish-cbc "kamekame"
 -A hmac-sha1 "this is the test key" -;
    EOF
```

そしてホスト-B の設定は次のようになります。

```
# setkey -c <<EOF
    spdadd fec0::11[110] fec0::10[any] tcp -P out ipsec
 esp/transport/fec0::11-fec0::10/use -;
    spdadd fec0::10[any] fec0::11[110] tcp -P in ipsec
 esp/transport/fec0::10-fec0::11/use -;
    add fec0::10 fec0::11 esp 0x10001 -m transport
 -E blowfish-cbc "kamekame"
 -A hmac-sha1 "this is the test key" -;
    add fec0::11 fec0::10 esp 0x10002 -m transport
 -E blowfish-cbc "kamekame"
 -A hmac-sha1 "this is the test key" -;
    EOF
```

SP の方向に注意してください。

15.9.3. IPv4 におけるトンネルモードの例

2 台のセキュリティゲートウェイ間のトンネルモード

セキュリティプロトコルは old AH トンネルモード、すなわち RFC1826 で指定されるものです。認証アルゴリズムは「this is the test」を鍵とする keyed-md5 です。

```
                          ======= AH =======
                          |                |
    ######-A   ######-A    ######-B   ######-B
       10.0.1.0/24 ---- 172.16.0.1 ----- 172.16.0.2 ---- 10.0.2.0/24
```

ゲートウェイ-A における設定は次のようになります。

```
# setkey -c <<EOF
    spdadd 10.0.1.0/24 10.0.2.0/24 any -P out ipsec
 ah/tunnel/172.16.0.1-172.16.0.2/require -;
    spdadd 10.0.2.0/24 10.0.1.0/24 any -P in ipsec
 ah/tunnel/172.16.0.2-172.16.0.1/require -;
    add 172.16.0.1 172.16.0.2 ah-old 0x10003 -m any
 -A keyed-md5 "this is the test" -;
    add 172.16.0.2 172.16.0.1 ah-old 0x10004 -m any
```

```
 -A keyed-md5 "this is the test" -;

EOF
```

上記の例のように、もしポート番号フィールドを書かないと、[any] と同じ意味になります。-m は使用される SA のモードを指定します。-m any はセキュリティプロトコルのモードのワイルドカードを意味します。この SA をトンネルモードとトランスポートモードの両方で使用できます。

そしてゲートウェイ-B では次のようになります。

```
# setkey -c <<EOF
    spdadd 10.0.2.0/24 10.0.1.0/24 any -P out ipsec
 ah/tunnel/172.16.0.2-172.16.0.1/require -;
    spdadd 10.0.1.0/24 10.0.2.0/24 any -P in ipsec
 ah/tunnel/172.16.0.1-172.16.0.2/require -;
    add 172.16.0.1 172.16.0.2 ah-old 0x10003 -m any
 -A keyed-md5 "this is the test" -;
    add 172.16.0.2 172.16.0.1 ah-old 0x10004 -m any
 -A keyed-md5 "this is the test" -;

EOF
```

二台のセキュリティゲートウェイ間の SA の束の作成

ゲートウェイ-A とゲートウェイ-B の間では、AH トランスポートモードと ESP トンネルモードが要求されます。この例では、まず ESP トンネルモードが適用され、次に AH トランスポートモードが適用されます。

```
                          ========== AH =========
                          |  ======= ESP =====  |
                          |  |               |  |
    ######-A     ######-A     ######-B     ######-B
    fec0:0:0:1::/64 --- fec0:0:0:1::1 ---- fec0:0:0:2::1 --- ↺
fec0:0:0:2::/64
```

15.9.4. IPv6 におけるトンネルモードの例

暗号化アルゴリズムは 3des-cbc, ESP の認証アルゴリズムは hmac-sha1 とします。AH の認証アルゴリズムは hmac-md5 とします。ゲートウェイ-A での設定は次のようになります。

```
# setkey -c <<EOF
    spdadd fec0:0:0:1::/64 fec0:0:0:2::/64 any -P out ↺
ipsec
 esp/tunnel/fec0:0:0:1::1-fec0:0:0:2::1/require
```

```
 ah/transport/fec0:0:0:1::1-fec0:0:0:2::1/require -;
    spdadd fec0:0:0:2::/64 fec0:0:0:1::/64 any -P in ↺
ipsec
 esp/tunnel/fec0:0:0:2::1-fec0:0:0:1::1/require
 ah/transport/fec0:0:0:2::1-fec0:0:0:1::1/require -;
    add fec0:0:0:1::1 fec0:0:0:2::1 esp 0x10001 -m ↺
tunnel
 -E 3des-cbc "kamekame12341234kame1234"
 -A hmac-sha1 "this is the test key" -;
    add fec0:0:0:1::1 fec0:0:0:2::1 ah 0x10001 -m ↺
transport
 -A hmac-md5 "this is the test" -;
    add fec0:0:0:2::1 fec0:0:0:1::1 esp 0x10001 -m ↺
tunnel
 -E 3des-cbc "kamekame12341234kame1234"
 -A hmac-sha1 "this is the test key" -;
    add fec0:0:0:2::1 fec0:0:0:1::1 ah 0x10001 -m ↺
transport
 -A hmac-md5 "this is the test" -;

    EOF
```

異なる通信端での SA の作成

ホスト-A とゲートウェイ-A の間では ESP トンネルモードが要求されています。暗号化アルゴリズムは cast128-cbc で、ESP の認証アルゴリズムは hmac-sha1 です。ホスト-A とホスト-B との間では ESP トランスポートモードが推奨されています。暗号化アルゴリズムは rc5-cbc で、ESP の認証アルゴリズムは hmac-md5 です。

```
            ================== ESP =================
            |  ======= ESP =======                 |
            |  |                 |                 |
          ###-A        ######-A         ###-B
        fec0:0:0:1::1 ---- fec0:0:0:2::1 ---- fec0:0:0:2::2
```

ホスト-A での設定は次のようになります。

```
# setkey -c <<EOF
    spdadd fec0:0:0:1::1[any] fec0:0:0:2::2[80] tcp -P ↺
out ipsec
 esp/transport/fec0:0:0:1::1-fec0:0:0:2::2/use
 esp/tunnel/fec0:0:0:1::1-fec0:0:0:2::1/require -;
    spdadd fec0:0:0:2::1[80] fec0:0:0:1::1[any] tcp -P ↺
in ipsec
 esp/transport/fec0:0:0:2::2-fec0:0:0:l::1/use
 esp/tunnel/fec0:0:0:2::1-fec0:0:0:1::1/require -;
```

```
   add fec0:0:0:1::1 fec0:0:0:2::2 esp 0x10001
-m transport
-E cast128-cbc "12341234"
-A hmac-sha1 "this is the test key" -;
   add fec0:0:0:1::1 fec0:0:0:2::1 esp 0x10002
-E rc5-cbc "kamekame"
-A hmac-md5 "this is the test" -;
   add fec0:0:0:2::2 fec0:0:0:1::1 esp 0x10003
-m transport
-E cast128-cbc "12341234"
-A hmac-sha1 "this is the test key" -;
   add fec0:0:0:2::1 fec0:0:0:1::1 esp 0x10004
-E rc5-cbc "kamekame"
-A hmac-md5 "this is the test" -;

   EOF
```

15.10. OpenSSH

##: Chern Lee.

セキュアシェル (secure shell) はリモートマシンへのセキュアなアクセスに使われるネットワーク接続ツールの集合です。 これは `rlogin`, `rsh`, `rcp`, `telnet` をそのまま置き換えて使えます。また、他のあらゆる TCP/IP 接続を ssh 経由でセキュアにトンネル/フォワードすることもできます。ssh はすべてのトラフィックを暗号化し、盗聴や接続の乗っ取り等のネットワークレベルの攻撃を事実上無効化します。

OpenSSH は OpenBSD プロジェクトによって維持管理されており、SSH v1.2.12 に最新のすべてのバグ修正と更新を適用したものをベースにしています。OpenSSH クライアントは SSH プロトコル 1 と 2 の両方に互換性があります。OpenSSH は FreeBSD 4.0 以降ベースシステムに取り込まれています。

15.10.1. OpenSSH を使うことの利点

telnet(1) や rlogin(1) を使う場合、一般にデータはネットワークを平文で流れます。ネットワークをクライアントとサーバの間のどこかで盗聴することで あなたのユーザ/パスワード情報やセション中を流れるデータを盗むことが可能です。 OpenSSH はこれらを予防する為にさまざまな認証と暗号化の方法を提供します。

15.10.2. sshd を有効にする

`rc.conf` ファイルに 以下の行を追加してください。

```
sshd_enable="YES"
```

次に起動したときから ssh デーモンが起動します。もしくは単に sshd デーモンを実行しても構いません。

15.10.3. SSH クライアント

ssh(1) ユーティリティは rlogin(1) と同様に働きます。

```
# ssh user@example.com
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host 'example.com' added to the list of known hosts.
user@example.com's password: *******
```

ログインは `rlogin` や `telnet` でセッションを張った時と同様に続きます。 SSH はクライアントが接続した時、 サーバの信頼性の検証のために鍵指紋システム (key fingerprint system) を利用します。 初めての接続の際にのみ、ユーザは `yes` と入力することを要求されます。 これ以降の login では保存されていた鍵指紋を照合することで検証されます。 SSH クライアントは保存されていた鍵指紋が login しようとした際に送られてきたものと異なっていた場合には警告を表示します。指紋は `~/.ssh/known_hosts` に、また SSH v2 指紋の場合は `~/.ssh/known_hosts2` に保存されます。

デフォルトでは、OpenSSH サーバは SSH v1 と SSH v2 両方の接続を受け付けるように設定されています。クライアントはそのどちらかを選択できます。バージョン 2 は、旧バージョンよりも堅固で安全です。

`ssh` に、プロトコル v1 と v2 についてそれぞれ、引数 `-1` または `-2` を渡すことで、利用するプロトコルを強制できます。

15.10.4. Secure copy

`scp` コマンドは `rcp` と同様に働きます。 安全な方法で行っているほかは、ローカルのファイルをリモートマシンへ、 あるいはリモートマシンのファイルをローカルにコピーするのは同じです。

```
#  scp user@example.com:/COPYRIGHT COPYRIGHT
user@example.com's password: *******
COPYRIGHT            100% |*****************************|  4735
00:00
#
```

前回の例でこのホストの指紋がすでに保存されていれば この `scp` を使う時に検証が行なわれます。

`scp` に渡される引数は、`cp` のものと似ており、ファイル (1 つまたは複数) が 1 つめの引数になり、コピー先が 2 つめの引数になります。 ファイル

はネットワーク越しに SSH を通して送られるので、引数に指定するファイルには `user@host:<path_to_remote_file>` という形式をとるものがあります。

15.10.5. 設定

システム全体の設定ファイルは、OpenSSH デーモン、クライアントの両方とも `/etc/ssh` ディレクトリにあります。

`ssh_config` はクライアントの動作設定、`sshd_config` はデーモンの動作設定を行ないます。

さらに、`rc.conf` オプションの `sshd_program`（デフォルトは `/usr/sbin/sshd`）と `sshd_flags` により、詳細な設定が行えます。

15.10.6. ssh-keygen

パスワードの代わりに ssh-keygen(1) を使ってユーザの認証用の RSA 暗号鍵を作ることができます。

```
% ssh-keygen
Initializing random number generator...
Generating p:  .++ (distance 66)
Generating q:  ..............................++ (distance 498)
Computing the keys...
Key generation complete.
Enter file in which to save the key (/home/user/.ssh/identity):
Enter passphrase:
Enter the same passphrase again:
Your identification has been saved in /home/user/.ssh/identity.
...
```

ssh-keygen(1) は認証に使う為の公開鍵と秘密鍵のペアを作ります。秘密鍵は `~/.ssh/identity` に保存され、公開鍵は `~/.ssh/identity.pub` に保存されます。公開鍵はリモートマシンの `~/.ssh/authorized_keys` にも置かなければなりません。

これでパスワードの代わり RSA 認証を使ってリモートマシンに接続できるようになったはずです。

ssh-keygen(1) でパスフレーズを使っている場合は、ユーザは秘密鍵を使うために毎回パスフレーズの入力を行なう必要があります。

同じ目的で、`ssh-keygen -d`（FreeBSD FreeBSD-CURRENT では `ssh-keygen -t dsa`）コマンドを使って SSH v2 DSA 鍵を生成することもできます。これは、SSH v2 セッション専用の DSA 公開/秘密鍵を生成します。公開鍵は `~/.ssh/id_dsa.pub` に保存され、秘密鍵は `~/.ssh/id_dsa` に置かれます。

DSA 公開鍵はリモートマシンの `~/.ssh/authorized_keys2` 内におきます。

ssh-agent(1) と ssh-add(1) は 複数のパスワード化された秘密鍵の管理に使われます。

15.10.7. SSH トンネリング

OpenSSH は暗号化されたセッションの中に他のプロトコルを カプセル化するトンネルを作ることができます。

以下のコマンドは ssh(1) で telnet 用のトンネルを作成します。

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

`ssh` コマンドは、次のオプションとともに利用します。

`-2`
: `ssh` にプロトコルバージョン 2 を使うことを指示します。(古い ssh サーバを使っているときには指定しないでください)

`-N`
: はトンネルだけでコマンドはないことを示します。省略されると ssh(1) は通常のセッションを開始します。

`-f`
: `ssh` にバックグラウンド実行を強制します。

`-L`
: ローカルトンネルを *`localport:remotehost:remoteport`* という形式で指定します。

`user@foo.example.com`
: リモートの SSH サーバです。

SSH のトンネルは `localhost` の指定されたポートに listen するソケットを作ることで実現されています。SSH はローカルのホスト/ポートで受けた接続すべてを SSH 接続経由で指定されたリモートホストのポートへ転送します。

この例では、`localhost` のポート *`5023`* がリモートマシンの `localhost` のポート *`23`* に転送されるようになっています。*`23`* は telnet なのでこれは SSH トンネルを通るセキュアな telnet セッションを作ります。

このようにして SMTP や POP3, FTP 等のセキュアではない TCP プロトコルをカプセル化することができます。

例15.1 SSH を用いた SMTP 用の安全なトンネルの作成

```
% ssh -2 -N -f -L 5025:localhost:25 ↺
user@mailserver.example.com

user@mailserver.example.com's password: *****
% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mailserver.example.com ESMTP
```

ssh-keygen(1) と別のユーザアカウントを組み合わせて使うことでより透過的で悩まずに済むような SSH のトンネル環境を作ることができます。 パスワードを入力するところで暗号鍵を使い、トンネルは別のユーザ権限で実行することが可能です。

15.10.7.1. 実用的な SSH トンネルの例

15.10.7.1.1. POP3 サーバへの安全な接続

仕事で、外部からの接続を受ける SSH サーバがあるとします。同じオフィスのネットワークには、POP3 サーバが動いているメールサーバがあるとします。 ネットワークもしくはあなたの家とオフィスの間のネットワーク経路は、 完全に信頼できるものかもしれませんし、そうではないかもしれません。そのため、電子メールは安全なやり方で見るようにしなければなりません。解決策は、オフィスの SSH サーバへの SSH 接続を行い、メールサーバへのトンネルを作成することです。

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-
server.example.com
user@ssh-server.example.com's password: ******
```

トンネルが作成されて動作したら、 メールクライアントに対し `localhost` のポート 2110 に POP3 リクエストを送るように指示できます。そこへの接続は、トンネルを経由して安全に `mail.example.com` に転送されます。

15.10.7.1.2. 厳格なファイアウォールをすり抜ける

内向けの接続をフィルタするだけでなく、 外向けの接続もフィルタして、 極端に厳しいファイアウォールルールを課すネットワーク管理者もいます。リモートのマシンには、SSH 接続と web サーフィンのための 22 番および 80 番ポートにしか接続させてもらえないかもしれません。

あなたは、それ以外の (もしかすると仕事に関係ない) サービスにアクセスしたくなるかもしれません。例えば、音楽ストリーミングを行う Ogg Vorbis サーバといったものです。この Ogg Vorbis サーバが 22 番または 80 番ポート以外でストリーミングを行っていたら、あなたはそのサーバに接続できないでしょう。

それに対する解決策は、あなたが接続しているネットワークのファイアウォールの外部にあるマシンに対して SSH 接続を行い、Ogg Vorbis サーバへのトンネルに利用することです。

```
% ssh -2 -N -f -L 8888:music.example.com:8000 ↺
user@unfirewalled.myserver.com
user@unfirewalled.myserver.com's password: *******
```

ストリーミングクライアントを `localhost` の 8888 番ポートに向けると、`music.example.com` の 8000 番ポートに転送され、ファイアウォールをすり抜けられます。

15.10.8. もっと詳しく知りたい人へ

OpenSSH

ssh(1) scp(1) ssh-keygen(1) ssh-agent(1) ssh-add(1)

sshd(8) sftp-server(8)

15.11. ファイルシステムアクセス制御リスト

##: Tom Rhodes.

スナップショットのようなファイルシステムの拡張と連携して、FreeBSD 5.0 以降ではファイルシステムアクセス制御リスト (ACLs) によるセキュリティを提供しています。

アクセス制御リストは、標準的な UNIX のパーミッションモデルを、非常に互換性の高い (POSIX.1e) やり方で拡張しています。この機能は、管理者がより洗練されたセキュリティモデルを利用し、その恩恵を受けられるようにします。

UFS ファイルシステム用の ACL サポートを有効にするには、次のオプションをカーネルに組み込まなければなりません。

```
options UFS_ACL
```

もしこのオプションが組み込まれていなければ、ACLs に対応したファイルシステムをマウントしようとすると、警告が表示されます。このオプションは `GENERIC` カーネルに含まれています。ACLs は、ファイルシステムの拡張属性が有効になっていることに依存しています。拡張属性は、次世代 UNIX ファイルシステムである UFS2 でネイティブ対応されています。

注記

UFS1 に拡張属性を付すように設定するのは、UFS2 よりも高いレベルの管理オーバヘッドが必要になります。また、UFS2 における拡張属性のパフォーマンスも大きく上がっています。 その結果、アクセス制御リストを利用する上では、一般的には UFS1 よりも UFS2 の方がおすすめです。

ACLs は、マウント時の管理フラグ `acls` で有効にされます。これは `/etc/fstab` に記述できます。 マウント時のフラグは、tunefs(8) を使って、ファイルシステムヘッダのスーパブロックにある ACLs フラグを変更するという方法で、常に自動で設定されるようになります。一般的には、下記の理由からスーパブロックフラグを使う方がよいでしょう。

- マウント時に指定した ACLs フラグは再マウント (mount(8) `-u`) 時に変更できません。完全に umount(8) した上で、新たに mount(8) するしかありません。これは、起動後にルートファイルシステムで ACLs を有効にできないことを意味します。また、ファイルシステムを利用し始めた後では、その配列を変えられないことも意味しています。

- スーパブロックフラグを設定すると、`fstab` に記述されていなかったり、デバイスの順番が変わってしまっても、常に ACLs が有効な状態でマウントされます。こうすることで、ファイルシステムを ACLs を有効にしないままマウントしてしまい、ACLs が正しくないかたちで強制され、セキュリティ問題につながることを防ぎます。

注記

ACLs の動作を変更して、まったく新たに mount(8) を行わなくてもフラグを有効にできるようにすることも可能でしょう。しかし、我々は、うっかり ACLs を有効にしないでマウントしてしまうのを防ぐようにした方が望ましいと考えました。 ACLs を有効にし、その後無効にしてから、拡張属性を取り消さないでまた有効にしてしまうと、鬱陶しい状態に自分で入り込んでしまえるからです。一般的には、一度ファイルシステムで ACLs を有効にしたら、無効にすべきではありません。そうしてしまうと、 ファイル保護がシステムのユーザの意図と齟齬をきたす可能性があるばかりか、ACLs を再度有効にすると、それまでパーミッションが変更されてきたファイルに古い ACLs を割り当ててしまい、 予想しない動作につながることも考えられます。

ACLs を有効にしたファイルシステムは、パーミッション設定の表示に + (プラス) 記号がつきます。例えば、次のようになります。

```
drwx------  2 robert  robert  512 Dec 27 11:54 private
drwxrwx---+ 2 robert  robert  512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert  robert  512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert  robert  512 Dec 27 11:57 directory3
drwxr-xr-x  2 robert  robert  512 Nov 10 11:54 public_html
```

ここでは、ディレクトリ `directory1` , `directory2` および `directory3` のすべてで ACLs が働いています。ディレクトリ `public_html` は対象外です。

第16章 ストレージ

16.1. この章では

この章では、FreeBSD におけるディスクの使用方法を説明します。これにはメモリディスク、ネットワークに接続されたディスク、および標準的な SCSI/IDE 記憶デバイスが含まれます。

この章では、以下の分野について説明します。

- 物理ディスク上のデータ構成 について記述するために FreeBSD が使用する用語 (パーティションおよびスライス)
- システムにハードディスクを追加する方法
- メモリディスクのような仮想ファイルシステムを設定する方法
- 使用できるディスク容量を制限するためにクォータを設定する方法
- 攻撃者から保護するためにディスクを暗号化する方法
- FreeBSD で CD や DVD を作成する方法
- バックアップのためのさまざまな記憶メディアオプション
- FreeBSD で利用できるバックアッププログラムの使用方法
- フロッピーディスクにバックアップする方法
- スナップショットとは何か、そしてそれを効果的に使用する方法

16.2. デバイス名

以下は、FreeBSD で対応している物理記憶デバイスとそれに対応するデバイス名のリストです。

表16.1 物理ディスクへの名前付け

ドライブの種類	ドライブのデバイス名
IDE ハードドライブ	ad
IDE CD-ROM ドライブ	acd
SCSI ハードドライブおよび USB 大容量記憶デバイス	da

ドライブの種類	ドライブのデバイス名
SCSI CD-ROM ドライブ	`cd`
その他の非標準的 CD-ROM ドライブ	ミツミ CD-ROM は `mcd`, Sony CD-ROM は `scd`, 松下/パナソニック CD-ROM は `matcd`
フロッピードライブ	`fd`
SCSI テープドライブ	`sa`
IDE テープドライブ	`ast`
フラッシュドライブ	DiskOnChip® フラッシュデバイスは `fla`
RAID ドライブ	Adaptec® AdvancedRAID は `aacd`, Mylex® は `mlxd` および `mlyd`, AMI MegaRAID® は `amrd`, Compaq Smart RAID は `idad`, 3ware® RAID は`twed`

16.3. ディスクの追加

##: David O'Brien.

現在一つしかドライブがない計算機に新しく SCSI ディスクを追加したいとしましょう。まずコンピュータの電源を切り、コンピュータやコントローラ、ドライブの製造元の説明書に従ってドライブを取り付けます。このあたりの手順は非常に多岐にわたるため、詳細はこの文書の範囲外です。

`root` ユーザでログインします。ドライブの取り付け後は `/var/run/dmesg.boot` を調べて新しいディスクが見つかっていることを確認しておきます。この例では、新しく付けたドライブは `da1` で、我々はそれを `/1` にマウントしたいとしましょう (もし IDE ドライブを付けようとしているのなら、デバイス名は 4.0 以前のシステムでは `wd1`, ほとんどの 4.x システムでは `ad1` になるでしょう)。

FreeBSD は IBM-PC 互換のコンピュータで動くため、PC BIOS のパーティションを考慮に入れる必要があります。これは従来の BSD パーティションとは異なります。PC ディスクは 4 つまでの BIOS パーティションエントリを持つことができます。もしそのディスクを本当に FreeBSD 専用にしたい場合には 専用 モードで用いることもできます。そうでない場合には、FreeBSD は PC BIOS パーティションのどれか一つの中に入れることになります。FreeBSD では、従来の BSD パーティションと混乱しないように PC BIOS パーティションのことをスライスと呼びます。また、別の OS がインストールされていたコンピュータで使われていたが FreeBSD 専用にするディスク上でもスライスを用いることができます。これは、他の OS の `fdisk` ユーティリティを混乱させないためです。

スライスの場合、ドライブは `/dev/da1s1e` として加えられるでしょう。これは、SCSI ディスクでユニット番号は 1 (二つめの SCSI ディスク), スライスは 1 (PC BIOS のパーティションが 1) で BSD パーティション `e`, と読みます。専用ディスクの場合だと単純に `/dev/da1e` として加えられるでしょう。

16.3.1. sysinstall(8) の利用

1. sysinstall の操作

   `sysinstall` の使い易いメニューを利用して、新しいディスクのパーティション分けやラベル付けを行なうことができます。`root` ユーザでログインするか `su` コマンドを用いるかして root 権限を取得します。`/stand/sysinstall` を実行して `Configure` メニューに入ります。`FreeBSD Configuration Menu` の中でスクロールダウンして `Fdisk` の項目を選びます。

2. fdisk パーティションエディタ

   fdisk では、ディスク全体を FreeBSD で使うために **`A`** を入力します。「remain cooperative with any future possible operating systems」と聞かれたら `YES` と答えます。**`W`** で変更をディスクに書き込みます。ここで **`q`** と入力して FDISK エディタを抜けます。 次にマスタブートレコードについて聞かれます。 ここでは既に動いているシステムにディスクを追加しようとしているので `None` を選びます。

3. ディスクラベルエディタ

   次に sysinstall を終了し、もう一度起動する必要があります。同じ手順を踏んで今度は `Label` オプションを選択し、`Disk Label Editor` に入ります。ここでは従来の BSD パーティションを作成します。一つのディスクは `a` から `h` までのラベルがついた最大 8 つのパーティションを持つことができます。いくつかのパーティションラベルは特別な用途に用いられます。`a` パーティションはルートパーティション (`/`) です。したがって、システムディスク (つまり起動ディスク) のみに `a` パーティションがあるべきです。`b` パーティションはスワップパーティションに用いられ、複数のディスクにスワップパーティションを作ることができます。`c` は専用モードにおけるディスク全体、もしくはスライスモードにおけるスライス全体を指します。他のパーティションは汎用的に用いられます。

   sysinstall のラベルエディタ は、ルートパーティションでもスワップパーティションでもないパーティションには、`e` パーティションを採用しようとします。ラベルエディタでファイルシステムを作成するには **`C`** を入力してください。FS (ファイルシステム) かスワップかを聞かれたら `FS` を選びマウントポイント (たとえば `/mnt`) を入力します。 インストール後のモードでディスクを追加する場合、 sysinstall は `/etc/fstab` にエントリを追加しないため、ここで指定するマウントポイントはそれほど重要ではありません。

さて、ディスクに新しいラベルを書き込み、 そこにファイルシステムを作る準備が整いました。早速 W を叩いて実行しましょう。sysinstall からの、新しいパーティションをマウントできない、というエラーは無視してください。Label Editor から抜け、sysinstall を終了します。

4. 終了

最後に /etc/fstab を編集し、新しいディスクのエントリを追加します。

16.3.2. コマンドラインユーティリティの利用

16.3.2.1. スライスの利用

このセットアップ方法では、 すでにコンピュータに他のオペレーティングシステムがインストールされていても 正しく協調動作することが可能で、他のオペレーティングシステムの fdisk ユーティリティを混乱させることもありません。新しいディスクにインストールする場合は、この方法を用いることが推奨されています。後述する ##### は、そうしなければならない理由がある時にのみ、利用するようにしてください。

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# fdisk -BI da1 # ###########
# disklabel -B -w -r da1s1 auto  # ############
# disklabel -e da1s1  # ############################
# mkdir -p /1
# newfs /dev/da1s1e  # ########################
# mount /dev/da1s1e /1  # #############
# vi /etc/fstab # /etc/fstab #############
```

IDE ディスクを使う場合は da の部分を ad とします。4.X より前のシステムでは、(訳注: ad ではなく) wd としてください。

16.3.2.2. 専用モード

新しいドライブを他の OS と共有しない場合には ## モードを用いることもできます。このモードはマイクロソフトの OS を混乱させることを憶えておいてください (しかし、それらによって壊されることはありません)。一方、IBM の OS/2® はどんなパーティションでも見つけたら理解できなくても「専有」します。

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# disklabel -Brw da1 auto
# disklabel -e da1 # `e' ##########
# newfs -d0 /dev/da1e
# mkdir -p /1
# vi /etc/fstab # /dev/da1e #######
# mount /1
```

もう一つの方法は次の通り。

```
# dd if=/dev/zero of=/dev/da1 count=2
# disklabel /dev/da1 | disklabel -BrR da1 /dev/stdin
# newfs /dev/da1e
# mkdir -p /1
# vi /etc/fstab # /dev/da1e #######
# mount /1
```

注記

FreeBSD 5.1-RELEASE から、従来の disklabel(8) プログラムは bsdlabel(8) ユーティリティに置き換えられました。bsdlabel(8) では、使用されていない数多くのオプションやパラメタが削除されました。たとえば `-r` オプションは bsdlabel(8) では取り除かれました。詳細については bsdlabel(8) のマニュアルページを参照してください。

16.4. RAID

16.4.1. ソフトウェア RAID

16.4.1.1. Concatenated Disk Driver (CCD) の設定

##: Christopher Shumway.

##: Jim Brown.

大容量記録に関する解決法を選択する際にもっとも重視すべき要素は、 速度、信頼性、そして費用です。三つを同時にバランスよく実現することは稀です。通常、速くて信頼性のある大容量記録装置は高価であり、 費用を抑えようとすると速度または信頼性のどちらかが犠牲になります。

ここで例にあげるシステムの設計においては、費用が最も重要な要素として、次に速度、最後に信頼性が選択されています。 このシステムでのデータ転送速度は結局のところネットワークによって制限されます。 信頼性は大変重要です。ただし、以下で説明する CCD ドライブは、データ自体はすでに CD-R に完全にバックアップしてあるもの (したがって交換は簡単にできます) の、オンラインデータの役割をさせています。

あなた自身の要求事項を決定することは、 大容量記録に関する解決法を選択することの最初の段階です。 もしあなたの要求事項が費用より速度または信頼性を優先するなら、解決法はこのシステムとは違うものになるでしょう。

16.4.1.1.1. ハードウェアのインストール

IDE システムディスクに加えて、Western Digital 製の 30GB, 5400RPM の IDE ディスク三台を使って、以下に説明されているような約 90GB のオンラインストレージとなる CCD ディスクを作成しました。各 IDE ディスクがそれぞれの IDE コントローラとケーブルをもっていることが理想的ですが、費用を最低限にするために、IDE コントローラを追加していません。その代わり、それぞれの IDE コントローラがマスタデバイスを一つ、スレーブデバイスを一つ持つように、ディスクはジャンパを使って設定されています。

再起動の際に、システム BIOS が接続されたディスクを自動的に検出するように設定されました。より重要なことは、FreeBSD が再起動の際にそれらを検出することです。

```
ad0: 19574MB <WDC WD205BA> [39770/16/63] at ata0-master UDMA33
ad1: 29333MB <WDC WD307AA> [59598/16/63] at ata0-slave UDMA33
ad2: 29333MB <WDC WD307AA> [59598/16/63] at ata1-master UDMA33
ad3: 29333MB <WDC WD307AA> [59598/16/63] at ata1-slave UDMA33
```

注記

FreeBSD がディスクをすべて検出しないときは、ジャンパを正しく設定してあるか確認してください。多くの IDE ドライブは「ケーブルセレクト」ジャンパを持っています。 これはマスタ/スレーブの関係を設定するジャンパでは ありません。ドライブの文書を参照して、正しいジャンパ設定を見つけてください。

次に、ファイルシステムの一部分として、 それらをどのように接続するのかを考慮します。vinum(8) および ccd(4) の両方を検討すべきでしょう。この設定では、ccd(4) を選択しました。

16.4.1.1.2. CCD の設定

ccd(4) ドライバは、いくつかの同じディスクを使って、 一つの論理的ファイルシステムに連結することができます。ccd(4) を使用するためには、カーネルが ccd(4) に対応している必要があります。次の行をカーネルコンフィギュレーションファイルに追加して、カーネルを再構築し、再インストールしてください。

```
pseudo-device   ccd     4
```

5.X システムでは、上記の代わりに次の行を追加しなければなりません。

```
device   ccd
```

注記

FreeBSD 5.X では ccd(4) デバイスの数を指定する必要はありません。ccd(4) デバイスドライバは自己複製するようになりました — 新しいデバイスインスタンスは、必要に応じてその都度自動的に作成されます。

FreeBSD 3.0 以降では、カーネルモジュールを読み込んで ccd(4) に対応することもできます。

ccd(4) を設定するために、まず disklabel(8) を使用してディスクにラベルを書き込まなくてはなりません。

```
disklabel -r -w ad1 auto
disklabel -r -w ad2 auto
disklabel -r -w ad3 auto
```

このコマンドはディスク全体を示す `ad1c`, `ad2c` および `ad3c` に対するディスクラベルを作成します。

注記

FreeBSD 5.1-RELEASE から、従来の disklabel(8) プログラムは bsdlabel(8) ユーティリティに置き換えられました。bsdlabel(8) では、使用されていない数多くのオプションやパラメタが削除されました。たとえば `-r` オプションは bsdlabel(8) では取り除かれました。詳細については bsdlabel(8) のマニュアルページを参照してください。

次に、ディスクラベルのタイプを変更します。 disklabel(8) を使用してディスクラベルを編集してください。

```
disklabel -e ad1
disklabel -e ad2
disklabel -e ad3
```

このコマンドは `EDITOR` 環境変数に設定されているエディタ (一般的には vi(1)) でそれぞれのディスクの現在のディスクラベルを開きます。

変更されていないディスクラベルは以下のようになります。

```
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0     0     0   # (Cyl.    ↺
0 - 59597)
```

ccd(4) で使用する e パーティションを作成します。通常では c パーティションの行をコピーすれば良いでしょう。しかし、fstype は **4.2BSD** でなければ なりません。ディスクラベルは以下のようになるでしょう。

```
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0     0     0   # (Cyl.    ↺
0 - 59597)
  e: 60074784        0    4.2BSD        0     0     0   # (Cyl.    ↺
0 - 59597)
```

16.4.1.1.3. ファイルシステムの構築

ccd0c デバイスノードはまだ存在していないかも知れません。そのときは、次のコマンドを実行して作成してください。

```
cd /dev
sh MAKEDEV ccd0
```

注記

FreeBSD 5.0 では devfs(5) が /dev 以下のデバイスノードを自動的に管理するので、MAKEDEVを使用する必要はありません。

すべてのディスクにラベルを書き込んだので、ccd(4) を構築してください。これを行うためには、以下のようなオプションで ccdconfig(8) を使います。

```
ccdconfig ccd0❶ 32❷ 0❸ /dev/ad1e❹ /dev/ad2e /dev/ad3e
```

各オプションの使用法と意味は以下の通りです。

❶ 一番目の引数は設定するデバイスです。この例の場合は /dev/ccd0c です。/dev/ の部分はオプションです。

❷ ファイルシステムに対するインタリーブです。インタリーブは、 ディスクブロック内のストライプサイズを定義します。ディスクブロックは通常 512 バイトです。したがって 32 インタリーブは 16,384 バイトとなります。

❸ これは ccdconfig(8) に対するフラグです。ドライブミラーリングを有効にしたい場合、ここにフラグを指定します。この設定では ccd(4) に対するミラーリングは提供しませんので、0 (ゼロ) を指定しています。

❹ この ccdconfig(8) に対する最後の引数は、アレイ内に置くデバイスです。それぞれのデバイスに対する完全なパス名を使用します。

ccdconfig(8) を実行すると ccd(4) が設定されます。これでファイルシステムをインストールすることが可能です。オプションについて newfs(8) を参照するか、次のように実行してください。

```
newfs /dev/ccd0c
```

16.4.1.1.4. 自動的に設定する

一般的に、再起動するたびに ccd(4) をマウントしたいと思うでしょう。これを行うために、まず設定をしなければなりません。次のコマンドを用いて、現在の設定を `/etc/ccd.conf` に書き出します。

```
ccdconfig -g > /etc/ccd.conf
```

`/etc/ccd.conf` が存在すると、再起動の際に `/etc/rc` スクリプトが `ccdconfig -C` を実行します。これにより、ccd(4) は自動的に設定された後、マウントされます。

注記

シングルユーザモードで起動している場合には、ccd(4) を mount(8) する前に、アレイを設定するために次のコマンドを実行する必要があります。

```
ccdconfig -C
```

自動的に ccd(4) をマウントするには、`/etc/fstab` に ccd(4) のエントリ追加します。このように設定すると起動時にマウントされます。

```
/dev/ccd0c              /media      ufs     rw      2       2
```

16.4.1.2. Vinum ボリュームマネージャ

Vinum ボリュームマネージャは、仮想ディスクドライブを実装したブロックデバイスドライバです。Vinum は、ディスクハードウェアをブロックデバイスインタフェースから 分離し、データを配置します。 その結果、ディスク記憶装置を従来のスライスで扱うのと比較して、柔軟性、性能および信頼性が向上しています。vinum(8) は RAID-0, RAID-1 および RAID-5 モデル、そしてそれぞれの組合せを実装しています。

vinum(8) の詳細については Vinum ボリュームマネジャ を参照してください。

16.4.2. ハードウェア RAID

FreeBSD は、さまざまなハードウェア RAID コントローラにも対応しています。これらのデバイスはアレイを制御するための 特別なソフトウェアを FreeBSD で必要することなく、RAID サブシステムを制御します。

カード上の BIOS を使用して、カードはそれ自身でディスク操作のほとんどを制御します。以下は Promise IDE RAID コントローラを使用した設定の簡単な説明です。このカードがインストールされ、システムが起動したときには、情報の入力を促すプロンプトを表示します。指示にしたがってカードの設定画面に進んでください。接続されたドライブを組み合わせるように設定することができます。設定後、ディスクは FreeBSD に対して単一のドライブのように見えます。他の RAID レベルは適宜設定できます。

16.4.3. ATA RAID1 アレイの再構築

FreeBSD はアレイ内の障害ディスクを動作中に交換できます。ただし、再起動前にそれを検知していることが必要です。

`/var/log/messages` または dmesg(8) の出力に次のような行があるでしょう。

```
ad6 on monster1 suffered a hard error.
ad6: READ command timeout tag=0 serv=0 - resetting
ad6: trying fallback to PIO mode
ata3: resetting devices .. done
ad6: hard error reading fsbn 1116119 of 0-7 (ad6 bn 1116119; cn ↺
1107 tn 4 sn 11) status=59 error=40
ar0: WARNING - mirror lost
```

atacontrol(8) を使用して詳細を調べてください。

```
# atacontrol list
ATA channel 0:
 Master:      no device present
 Slave:   acd0 <HL-DT-ST CD-ROM GCR-8520B/1.00> ATA/ATAPI rev 0

ATA channel 1:
 Master:      no device present
 Slave:       no device present

ATA channel 2:
 Master:  ad4 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
 Slave:       no device present

ATA channel 3:
 Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
 Slave:       no device present

# atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: DEGRADED
```

1. ディスクを安全に取り外すために、まずアレイから切り離します。

```
# atacontrol detach 3
```

2. ディスクを取り外します。

3. スペアのディスクを取り付けます。

```
# atacontrol attach 3
Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
Slave:   no device present
```

4. アレイを再構築します。

```
# atacontrol rebuild ar0
```

5. 再構築コマンドは完了するまで他の操作を受け付けません。しかし、 もう一つ別のターミナルを (Alt+F***n*** を押して) 開き、次のコマンドを実行すると進行状態を確認することができます。

```
# dmesg | tail -10
[output removed]
ad6: removed from configuration
ad6: deleted from ar0 disk1
ad6: inserted into ar0 disk1 as spare

# atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: REBUILDING 0% completed
```

6. 操作が完了するまでお待ちください。

16.5. 光メディア (CD & DVD) の作成と使用

##: Mike Meyer.

16.5.1. はじめに

CD は他の一般的なディスクと異なる様々な特徴を持っています。 そもそもユーザが書き込むことができません。また遅延なしで連続的に読み出せるように、トラック間をヘッドが移動しないですむようにデザインされています。 さらにこのサイズのメディアの中ではシステムをまたぐデータの 移動が比較的簡単でもあります。

CD はトラックの概念を持っていますが、これはデータを連続的に読み出すためのものであってディスクの物理特性ではありません。FreeBSD で CD を作成するには、まず CD のトラックとなるデータファイルを用意し、そのトラックを CD に書き込みます。

ISO 9660 ファイルシステムはこの様な差異を扱うべく設計されました。その結果、ファイルシステムは一般的に使用するのに差しつかえない程度に 制限されて標準化されて

います。幸いなことに、ISO 9660 ファイルシステムには拡張機構が提供されています。適切に書かれた CD は、拡張機構に対応したシステムでは拡張を利用して、そうでないシステムでは 拡張機構を使用しない範囲で動作するようになっています。

sysutils/mkisofs プログラムは ISO 9660 ファイルシステムを含むデータファイルを作成するのに使われます。これには様々な拡張をサポートするオプションがあり、以下で説明します。このソフトウェアは、ports の sysutils/mkisofs からインストールすることができます。

CD に書き込むためのツールは、お使いの CD ライタが ATAPI 接続か否かにも依存します。ATAPI CD ライタなら、ベースシステムの一部である `burncd` プログラムを使います。SCSI や USB の CD ライタなら、ports の sysutils/cdrecord をインストールして `cdrecord` プログラムを使うべきでしょう。

`burncd` が対応しているドライブは限定されています。ドライブが対応されているかどうかを確認するには、CD-R/RW supported drives にある一覧を見てください。

注記

FreeBSD 5.X または FreeBSD 4.8-RELEASE 以降のバージョンを使用している場合、ATAPI/CAM モジュール を使用すると ATAPI ハードウェア上で SCSI ドライブ用の `cdrecord` および他のツールを使用できるようになります。

16.5.2. mkisofs

sysutils/mkisofs は UNIX® ファイルシステムの名前空間におけるディレクトリツリーのイメージとして ISO 9660 ファイルシステムを作成します。最も簡単な使い方は以下の通りです。

```
# mkisofs -o imagefile.iso /path/to/tree
```

このコマンドは *`/path/to/tree`* 以下のディレクトリツリーのコピーである ISO 9660 ファイルシステムを含んだ *`imagefile.iso`* ファイルを作成します。この過程において、ファイル名は標準的な ISO 9660 ファイルシステムの制限に適合するようなファイル名に対応づけられ、 ISO ファイルシステムでファイル名を文字化できないファイルは除外されます。

この制限を回避するために利用できるオプションはいくつもあります。特に `-R` オプションは UNIX® システムで標準的な Rock Ridge 拡張を有効にします。`-J` オプションは Microsoft のシステムで標準的な Joliet 拡張を有効にし、 `-hfs` オプションは Mac OS® で使用されている HFS ファイルシステムを作成するために使われます。

FreeBSD でしか使わないのであれば、-U オプションを使用するとあらゆるファイル名制限を無効にできます。さらに -R オプションとともに使うことで FreeBSD と同一のファイルシステムイメージを作成できますが、これは ISO 9660 標準の多くを無視しています。

一般的に使われる最後のオプションは -b オプションです。これは「El Torito」ブータブル CD を作成するのに使う起動イメージのありかを指定します。このオプションは引数として起動イメージへのパスを、CD に書き込まれるディレクトリツリーの頂点からの相対位置で取ります。したがって /tmp/myboot がブート可能な FreeBSD システムで /tmp/myboot/boot/cdboot にブートイメージがあるならば、以下のようにすることで ISO 9660 ファイルシステムのイメージを /tmp/bootable.iso に作成することができます。

```
# mkisofs -U -R -b boot/cdboot -o /tmp/bootable.iso /
tmp/myboot
```

この後、カーネルで vn (FreeBSD 4.X) または md (FreeBSD 5.X) が設定されていれば、ファイルシステムを以下のようにしてマウントすることができます。

```
# vnconfig -e vn0c /tmp/bootable.iso
# mount -t cd9660 /dev/vn0c /mnt
```

FreeBSD 4.X および FreeBSD 5.X に対しては以下の通りです。

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

/mnt と /tmp/myboot が同一かどうか確認してください。

sysutils/mkisofs には挙動を細かく制御するために他にもたくさんのオプションがあります。特に、ISO 9660 レイアウトの変更や Joliet および HFS ディスク作成などの 詳細は mkisofs(8) のマニュアルページをご覧ください。

16.5.3. burncd

あなたが持っているのが ATAPI CD ライタなら、CD に ISO イメージを書き込むために burncd コマンドが使えます。burncd はベースシステムの一部で /usr/sbin/burncd としてインストールされています。使い方はとても単純でオプションも少ししかありません。

```
# burncd -f cddevice data imagefile.iso fixate
```

以上のコマンドは *imagefile.iso* のコピーを *cddevice* に書き込みます。デフォルトのデバイスは /dev/acd0c です。書き込み速度や操作完了後に CD を自動的に取り出す方法、オーディオデータの書き込みなどのオプションについては burncd(8) を見てください。

16.5.4. cdrecord

あなたが持っている CD ライタが ATAPI ではなければ、CD を書き込むのに cdrecord を使う必要があります。cdrecord はベースシステムの一部ではなく、sysutils/cdrtools の port または 適切な package を利用してインストールしなければなりません。なお、ベースシステムを変更するとバイナリに矛盾が発生し、「コースター」を作ってしまうおそれがあります。したがって、システムをアップグレードする度にこの port も作り直すか、あるいはFreeBSD の安定版を追いかけているのならば、新しいバージョンが利用できるようになった時に ports をアップグレードする必要があります。

cdrecord にはたくさんのオプションがありますが、基本的な使い方は burncd よりもさらに簡単です。ISO 9660 イメージを書き込むには以下のようにします。

```
# cdrecord dev=device imagefile.iso
```

cdrecord のトリッキーな部分は、使用する dev を見つけるところにあります。適切な設定を見つけるためには cdrecord の -scanbus フラグを使います。たとえば、以下のような結果が出力されるでしょう。

```
# cdrecord -scanbus
Cdrecord 1.9 (i386-unknown-freebsd4.2) Copyright (C) 1995-2000 Jörg ↺
Schilling
Using libscg version 'schily-0.1'
scsibus0:
        0,0,0     0) 'SEAGATE ' 'ST39236LW       ' '0004' Disk
        0,1,0     1) 'SEAGATE ' 'ST39173W        ' '5958' Disk
        0,2,0     2) *
        0,3,0     3) 'iomega  ' 'jaz 1GB         ' 'J.86' Removable ↺
Disk
        0,4,0     4) 'NEC     ' 'CD-ROM DRIVE:466' '1.26' Removable ↺
CD-ROM
        0,5,0     5) *
        0,6,0     6) *
        0,7,0     7) *
scsibus1:
        1,0,0   100) *
        1,1,0   101) *
        1,2,0   102) *
        1,3,0   103) *
        1,4,0   104) *
        1,5,0   105) 'YAMAHA  ' 'CRW4260         ' '1.0q' Removable ↺
CD-ROM
        1,6,0   106) 'ARTEC   ' 'AM12S           ' '1.06' Scanner
        1,7,0   107) *
```

リストにあるデバイスに対する適切な dev の値がここに示されています。あなたの CD ライタをこのリストから見つけ、カンマで区切られた 3 つの数値を dev の値として使ってください。この例では CRW デバイスは 1,5,0 なので、適切な入力は dev=1,5,0 となります。値を明示するもっと簡単な方法もあります。詳細は cdrecord(1) を見てくださ

い。そこにはオーディオトラックを書き込む方法や、書き込み速度その他を操作する方法も書かれています。

16.5.5. オーディオ CD の複製

CD からオーディオデータを連続したファイルに展開し、ブランク CD にこれらのファイルを書き込むことで、オーディオ CD を複製することができます。この手順は ATAPI および SCSI ドライブの間で少し異なります。

手順16.1 SCSI ドライブ

1. `cdda2wav` を使用してオーディオを展開します。

```
% cdda2wav -v255 -D2,0 -B -Owav
```

2. `cdrecord` を使用して `.wav` ファイルに書き出します。

```
% cdrecord -v dev=2,0 -dao -useinfo  *.wav
```

「cdrecord」に説明されているように *`2.0`* が適切に指定されていることを確かめてください。

手順16.2 ATAPI ドライブ

1. ATAPI CD ドライバでは、それぞれのトラックを `/dev/acddtnn` のように利用できます。ここで *`d`* はドライブ番号であり、*`nn`* は二桁十進のトラック番号です。一桁の場合 0 を前に付加する必要があります。したがって、一番目のディスクの一番目のトラックは `/dev/acd0t01`、二番目のトラックは `/dev/acd0t02`、三番目のトラックは `/dev/acd0t03` などとなります。

   適切なデバイスファイルが `/dev` に存在することを確かめてください。存在しなければ、たとえば次のようにして作成します。

```
# cd /dev
# sh MAKEDEV acd0t99
```

注記

FreeBSD 5.0 では devfs(5) が `/dev` にエントリを自動的に作成、管理するので、`MAKEDEV` を使用する必要はありません。

2. dd(1) を使用して各トラックを展開します。ファイルを展開する際、ブロックサイズを指定しなければなりません。

```
# dd if=/dev/acd0t01 of=track1.cdr bs=2352
# dd if=/dev/acd0t02 of=track2.cdr bs=2352
...
```

3. burncd を使用して、展開したファイルをディスクに書き込みます。これらがオーディオファイルであること、そして書き込みが終了したときに burncd がディスクを固定 (fixate) することを明示しなければなりません。

```
# burncd -f /dev/acd0c audio track1.cdr track2.↺
cdr ... fixate
```

16.5.6. データ CD の複製

データ CD を、sysutils/mkisofs を用いて作成されたイメージファイルと機能的に等価なイメージファイルにコピーできます。これを使用して、すべてのデータ CD を複製することができます。ここでの例は CDROM デバイスが acd0 であるとしています。あなたの CDROM デバイスに読み替えてください。CDROM の場合には、パーティション全体またはディスク全体 を指定するために c をデバイス名の後に追加しなければなりません。

```
# dd if=/dev/acd0c of=file.iso bs=2048
```

これでディスクイメージを取り出すことができました。 すでに説明した方法を用いて CD に書き込むことができます。

16.5.7. データ CD の使用

さて、標準的なデータ CDROM を作成したので、おそらく次はそれをマウントしてデータを読み出したいと思うでしょう。デフォルトでは mount(8) は、ファイルシステムタイプを ufs としています。次のように実行しようとすると、

```
# mount /dev/cd0c /mnt
```

Incorrect super block というエラーが返されてマウントできないでしょう。 CDROM は UFS ファイルシステムではないために、このような手順でマウントしようすると失敗します。ファイルシステムのタイプが ISO9660 であると mount(8) に教えさえすれば、すべてはうまく動作します。mount(8) に -t cd9660 オプションを指定することでこれを行います。たとえば /dev/cd0c の CDROM デバイスを /mnt にマウントしたい場合は、以下のように実行します。

```
# mount -t cd9660 /dev/cd0c /mnt
```

使用している CDROM インタフェースによっては、 デバイス名 (この例では /dev/cd0c) が異なるかもしれないことに注意してください。また、-t cd9660 オプションは、単に mount_cd9660(8) を実行します。この例を以下のように短縮することもできます。

```
# mount_cd9660 /dev/cd0c /mnt
```

一般的にこの方法では、すべてのメーカの データ CDROM を使用することができます。しかしながら、特定の ISO 9660 拡張が施されたディスクでは奇妙な動作をするかもしれません。 たとえば Joliet ディスクは、 すべてのファイル名を 2 バイトの Unicode 文字で格納します。 FreeBSD カーネルは (まだ) Unicode を理解できないので、非英語文字はクエスチョンマークで表示されます (FreeBSD 4.3 以降を使用している場合、CD9660 ドライバには適切な Unicode 変換表を読み込むための急ごしらえのフックが含まれています。いくつかの共通のエンコードに対するモジュールは sysutils/cd9660_unicode port から利用可能です)。

CDROM をマウントしようとする時に、Device not configured と表示されるかもしれません。これは、ディスクがトレーにないと CDROM ドライブが判断しているか、ドライブがバス上に認識できないことを通常意味します。ディスクが挿入されたことを CDROM ドライブが認識するには数秒かかりますので、辛抱強く待ってください。

バスのリセットに返答するためのタイムアウトが短いために、時々 SCSI CDROM は認識に失敗するかもしれません。SCSI CDROM を持っている場合は、次のオプションをカーネルコンフィギュレーションファイルに追加して、カーネルを再構築してください。

```
options SCSI_DELAY=15000
```

これより、SCSI バスを起動時に 15 秒間停止させて、CDROM ドライブがバスリセットに応答する機会を与えます。

16.5.8. Raw データ CD の書き込み

ISO 9660 ファイルシステムを作成すること無く、ファイルを直接 CD に書き込むこともできます。この方法をバックアップ目的に使用している人もいます。これは、標準 CD を書き込むよりもさらに速く実行することができます。

```
# burncd -f /dev/acd1c -s 12 data archive.tar.gz fixate
```

このように CD に書き込まれたデータを取得するには、raw デバイスノードからデータを読み込まなくてはなりません。

```
# tar xzvf /dev/acd1c
```

このディスクを通常の CDROM としてマウントすることはできません。 このような CDROM は FreeBSD を除いて、他のすべてのオペレーティングシステムでは読み込むことはできません。CD をマウントしたいか、その他のオペレーティングシステムとデータを共有したい場合は、上記に説明したように sysutils/mkisofs を使用しなくてはなりません。

16.5.9. ATAPI/CAM ドライバの使用

このドライバは、ATAPI デバイス (CD-ROM, CD-RW, DVD ドライブなど) へ SCSI サブシステムを通じてアクセスすることを可能にします。これにより、sysutils/cdrdao または cdrecord(1) のようなアプリケーションが使用できるようになります。

このドライバを使用するためには、カーネルコンフィギュレーションファイルに次の行を追加する必要があります。

```
device atapicam
device scbus
device cd
device pass
```

次の行もカーネルコンフィギュレーションファイルに必要です。

```
device ata
device atapicd
```

両方がすでに存在しなければなりません。

それから再構築し、新しいカーネルをインストールし、コンピュータを再起動します。起動プロセス中にディスクライタは以下のように表示されるでしょう。

```
acd0: CD-RW <MATSHITA CD-RW/DVD-ROM UJDA740> at ata1-master PIO4
cd0 at ata1 bus 0 target 0 lun 0
cd0: <MATSHITA CDRW/DVD UJDA740 1.00> Removable CD-ROM SCSI-0 device
cd0: 16.000MB/s transfers
cd0: Attempt to query device size failed: NOT READY, Medium not ↺
present - tray closed
```

ドライブは `/dev/cd0` デバイスを通じてアクセスすることが可能となります。たとえば、次のようにして CD-ROM を `/mnt` にマウントします。

```
# mount -t cd9660 /dev/cd0c /mnt
```

`root` 権限で次のコマンドを実行して、ライタの SCSI アドレスを得ることができます。

```
# camcontrol devlist
<MATSHITA CDRW/DVD UJDA740 1.00>   at scbus1 target 0 lun 0 ↺
(pass0,cd0)
```

したがって、`1,0,0` が cdrecord(1) およびその他の SCSI アプリケーションで使用する SCSI アドレスです。

ATAPI/CAM および SCSI システムの詳細は atapicam(4) および cam(4) マニュアルページを参照してください。

16.6. フロッピーディスクの作成と使用

##: Julio Merino.

##: Martin Karlsson.

フロッピーディスクにデータを格納することはしばしば役にたちます。 たとえば、ある人が他のリムーバブル記録メディアを何も持っていないときや、 小さなデータを他のコンピュータに移動させる必要があるときです。

この節では、FreeBSD におけるフロッピーディスクの使用方法を説明します。主に 3.5 インチの DOS フロッピーのフォーマットと操作方法を扱いますが、他のフロッピーディスクの形式についても概念は似ています。

16.6.1. フロッピーのフォーマット

16.6.1.1. デバイス

他のデバイスと同様に、フロッピーディスクは `/dev` にあるエントリを通じてアクセスされます。4.X およびそれ以前のリリースにおいて raw フロッピーディスクにアクセスするには `/dev/fdN` または `/dev/fdNX` を使用します。*N* はドライブ番号を表し、大抵は 0 です。*X* は文字を表します。

5.0 およびそれ以降のリリースでは、単に `/dev/fdN` を使用します。

16.6.1.1.1. 4.X およびそれ以前のリリースでのディスクサイズ

`/dev/fdN.size` というデバイスもあります。*size* はフロッピーディスクのサイズをキロバイトで示したものです。 これらのエントリは低レベルフォーマットの際に、 ディスクサイズを決定するのに使用されます。1440kB は以下の例で使用されるサイズです。

時々 `/dev` 下のエントリは (再) 作成されなければなりません。次のコマンドでこれを行います。

```
# cd /dev && ./MAKEDEV "fd*"
```

16.6.1.1.2. 5.X およびそれ以降のリリースでのディスクサイズ

FreeBSD 5.0 では devfs(5) が `/dev` 内のエントリを自動的に管理するので、`MAKEDEV`を使用する必要はありません。

所望のディスクサイズは fdformat(1) に `-f` フラグを通して渡されます。対応しているサイズは fdcontrol(8) のマニュアルページに掲載されていますが、最良に動作するのは 1440kB だと助言しておきます。

16.6.1.2. フォーマット

フロッピーディスクは、使用前に低レベルフォーマットをする必要があります。通常、ベンダは低レベルフォーマット済みのディスクを出荷していますが、 フォーマットはメディアの

品質を確認するよい方法です。より大きな (または小さな) ディスクサイズにすることも可能ですが、ほとんどのフロッピーディスクのサイズは 1440kB で動作するように設計されています。

フロッピーディスクを低レベルフォーマットするには fdformat(1) を使用する必要があります。このユーティリティは引数としてデバイス名を指定します。

ディスクが良好かあるいは不良であるかを決定するのに役立つので、 エラーメッセージをすべてメモに取っておいてください。

16.6.1.2.1. 4.X 以前のリリースでのフォーマット

`/dev/fdN.size` デバイスを使ってフロッピーをフォーマットします。新しい 3.5 インチフロッピーディスクをドライブに挿入し、以下のコマンドを実行してください。

```
# /usr/sbin/fdformat /dev/fd0.1440
```

16.6.1.2.2. 5.0 以降のリリースでのフォーマット

`/dev/fdN` デバイスを使用してフロッピーをフォーマットします。新しい 3.5 インチフロッピーディスクをドライブに挿入し、以下のコマンドを実行してください。

```
# /usr/sbin/fdformat -f 1440 /dev/fd0
```

16.6.2. ディスクラベル

ディスクを低レベルフォーマットしたら、 次にディスクラベルを作成する必要があります。ディスクラベルは後で破棄されますが、 システムがディスクのサイズとジオメトリを決定するのに必要になります。

新しいディスクラベルはディスク全体を引き継ぎ、 フロッピーのジオメトリに関する適切な情報のすべてが含まれます。 ディスクラベルに対するジオメトリの値は `/etc/disktab` に掲載されています。

次のように disklabel(8) を実行できます。

```
# /sbin/disklabel -B -r -w /dev/fd0 fd1440
```

注記

FreeBSD 5.1-RELEASE から、従来の disklabel(8) プログラムは bsdlabel(8) ユーティリティに置き換えられました。bsdlabel(8) では、使用されていないオプションおよびパラメタの数多くが削除されました。たとえば `-r` オプションは bsdlabel(8) では取り除か

れました。詳細については bsdlabel(8) マニュアルページを参照してください。

16.6.3. ファイルシステム

これでフロッピーを高レベルフォーマットする準備ができました。これは FreeBSD がディスクを読み書きする新しいファイルシステムを作成します。 新しいファイルシステムを作成するとディスクラベルは破棄されます。したがって、ディスクを再フォーマットするときには、ディスクラベルを再作成しなくてはなりません。

フロッピーのファイルシステムには UFS または FAT を使用できます。フロッピーに対しては FAT が一般的によりよい選択です。

フロッピー上に新しいファイルシステムを作成するには次のようにします。

```
# /sbin/newfs_msdos /dev/fd0
```

これでディスクが使用できるようになりました。

16.6.4. フロッピーの使用

フロッピーを使用するために、mount_msdos(8) (4.X 以前のリリース) または mount_msdosfs(8) (5.0 以後のリリース) を用いてマウントします。Ports Collection から emulators/mtools を使用することもできます。

16.7. データテープの作成と使用

一般的なテープメディアには 4mm, 8mm, QIC, ミニカートリッジ、DLT があります。

16.7.1. 4mm (DDS: Digital Data Storage)

4mm テープはワークステーションのバックアップメディアとして QIC に取って代わりつつあります。この傾向は QIC ドライブの主要なメーカであった Archive を Conner が買収し QIC ドライブの製造を中止したことで加速しました。4mm ドライブは小型で静かですが 8mm ドライブが持っている信頼性ほど、その評判は良くありません。また、4mm カートリッジは 8mm カートリッジよりも安価で小型 (3 x 2 x 0.5 インチ、76 x 51 x 12 mm) になっています。ただし、8mm と同様に、4mm のヘッドはヘリカルスキャン方式 (訳注: VTR と同様の回転ヘッドを使う方式) を採用しているため、比較的寿命が短いです。

ドライブのデータスループットは、150 kB/s から 最大で 500 kB/s 程度です。データ容量は 1.3 GB から 2.0 GB です。ドライブのほとんどで利用可能なハードウェア圧縮を使用すると、容量が約 2 倍になります。マルチドライブテープライブラリユニットは 1 つ

の筐体に 6 つのドライブを収容可能で、自動的にテープの交換ができます。ライブラリの容量は 240 GB に達します。

現在の DDS-3 標準は 12 GB (圧縮時 24 GB) までのテープ容量に対応しています。

8mm ドライブと同様に 4mm ドライブはヘリカルスキャンを使用します。ヘリカルスキャン方式の利点および欠点はすべて 4mm および 8mm ドライブの両方に当てはまります。

テープは 2,000 回のパスあるいは 100 回フルバックアップした後には交換するべきです。

16.7.2. 8mm (Exabyte)

8mm テープは SCSI テープドライブとして最もよく使われているもので、データ交換用として最良の選択です。ほとんどのサイトには Exabyte 2 GB 8mm テープドライブがあるでしょう。8mm ドライブは信頼性が高く、使いやすく、静かです。カートリッジは安価で小型です (4.8 x 3.3 x 0.6 インチ、122 x 84 x 15 mm)。8mm テープの欠点は、テープとヘッドの相対的な速度が高速なために、 比較的ヘッドとテープの寿命が短いことです。

データスループットは 250 kB/s から 500 kB/s 程度です。データ容量は 300 MB から 7 GB です。ほとんどのドライブで利用可能なハードウェア圧縮を利用すると、容量が約 2 倍になります。これらのドライブは、単一のユニットから 6 つのドライブと 120 本のテープを一つの筐体に収容したマルチドライブテープライブラリまで利用可能です。テープはユニットによって自動的に取り換えられます。ライブラリの容量は 840 GB 以上に達します。

Exabyte の「Mammoth」モデルはテープ 1 本あたり 12 GB (圧縮時 24 GB) に対応し、従来のテープドライブと比べ費用は約 2 倍になります。

データはヘリカルスキャンを用いてテープに記録されます。 ヘッダはメディアに対してある傾き (約 6 度) に配置されます。 テープはヘッドのある円筒の周の 270 度にわたって接触します。テープが円筒面を走行する間、円筒は回転しています。この結果、高密度のデータのつまったトラックは、 狭い間隔でテープの上端と下端の間を斜めに横切ります。

16.7.3. QIC

QIC-150 テープとドライブは、おそらく最も一般的に使われているドライブとメディアでしょう。 QIC テープドライブは「現実的な」 バックアップドライブとしては最も高価でないものです。 欠点はメディアのコストです。QIC テープは 8mm や 4mm テープと比較して GB あたりのデータの保存で 5 倍ほど高価です。 しかし、あなたの必要とする量が半ダース程のテープで十分であれば、 QIC は正しい選択かもしれません。QIC は最も一般的なテープドライブです。 すべてのサイトに QIC ドライブのどれかの容量のものがあります。問題は、 QIC は同じようなテープ (まったく同じ場合もある) に多様な記

録密度があることです。QIC ドライブは静かではありません。 これらのドライブはデータ記録を開始する前に音をたててシークしますし、リード、ライト、シークの時にはっきりと聞こえる音を出します。QIC テープの大きさは (6 x 4 x 0.7 インチ、152 x 102 x 17 mm) です。1/4 インチ幅のテープも使用している ミニカートリッジ は別に議論します。テープライブラリやチェンジャはありません。

データスループットは ~1500 kB/s から ~5000 kB/s 程度です。データ容量は 400 MB から 150 GB です。ハードウェア圧縮が最近のドライブの多くで利用できます。QIC ドライブは DAT ドライブに置き換えられつつあり、あまり頻繁には使用されなくなっています。

データは複数のトラックに分かれてテープに記録されます。 トラックはテープメディアの長さ方向の一端からもう一方の端までです (訳注: 1 トラックの read/write が終わるとテープの走行方向を反転させ 次のトラックの read/write を行います)。トラックの数と、 それに対応するトラックの幅はテープの容量によって変わります。 すべてではありませんが、最近のドライブはほとんど、少なくとも読み出しについては (場合によっては書き込みも) 下位互換性があります。QIC はデータの安全性についてはよいといわれています (ヘリカルスキャンドライブに比べて機構は単純でより丈夫です)。

テープは 5000 回のバックアップで寿命となるでしょう。

16.7.4. XXX* ミニカートリッジ

16.7.5. DLT

DLT　はここに示したドライブのタイプの中で最高速のデータ転送レートを発揮します。1/2 インチ (12.5mm) テープが単リールのカートリッジ (4 x 4 x 1 インチ、100 x 100 x 25 mm) に入っています。カートリッジのひとつの側面全体がスイングゲートになっています。ドライブの機構がこのゲートを開け、テープリーダを引き出します。テープリーダには楕円形の穴があり、ドライブがテープを「引っ掛ける」のに使います。巻き取りのためのリールはドライブの中にあります。ここに挙げた他のカートリッジはすべて (9 トラックテープは唯一の例外です) 送り出しリールと巻き取りリールの両方がカートリッジの中にあります。

データスループットは約 1.5 MB/s で、4mm, 8mm, QIC テープドライブの 3 倍です。データ容量は単一のドライブで 10 GB から 20 GB の範囲です。マルチテープチェンジャ、マルチテープドライブ、5 から 900 巻のテープを 1 から 20 ドライブで扱うマルチドライブテープライブラリがあり、50 GB から 9 TB の容量が得られます。

圧縮によって、DLT Type IV フォーマットは 70 GB までの容量に対応しています。

データは (QICテープのように) テープの走行方向と平行に複数あるトラックへ記録されます。 2 つのトラックに同時書き込みを行います。 read/write ヘッドの寿命は比較的長いと言えます。テープの走行が止まればヘッドとテープの間の相対運動は無いからです。

16.7.6. AIT

AIT は、Sony が発表した新しいフォーマットで、テープ 1 本あたり 50 GB (圧縮時) まで格納できます。 テープにはメモリチップが搭載されており、 テープの内容の索引情報を保持しています。他のテープではテープ上のファイルの位置を把握するのに数分必要とするのですが、このテープドライブでは索引情報を読んで直ちに決定することができます。SAMS:Alexandria のようなソフトウェアは、40 を超える ATI テープライブラリを操作できるのはもちろんのこと、 テープのメモリチップと直接通信して、スクリーンに内容を表示し、 どのファイルがどのテープにバックアップされたかを調べて、 正しいテープを見つけ、読み込み、テープからデータを復元することができます。

このようなライブラリは大体 $20,000 くらいするので、愛好家が購入できる価格帯からは外れてしまいます。

16.7.7. 新品のテープを初めて使う場合

全く新品の空テープを読もうとしたり書き込もうとすると、処理は失敗するでしょう。次のようなメッセージがコンソールに出力されるでしょう。

```
sa0(ncr1:4:0): NOT READY asc:4,1
sa0(ncr1:4:0):  Logical unit is in process of becoming ready
```

テープに識別ブロック (Identifier Block:block number 0) がありません。QIC-525 標準を採用したすべての QIC テープドライブは識別ブロックをテープに書き込みます。2 つの解決方法があります。

- `mt fsf 1` によりテープドライブはテープに識別ブロックを書き込みます。
- フロントパネルのボタンを押してテープを取り出します。

  再びテープを挿入し、データをテープに `dump` します。

  `dump` は DUMP: End of tape detected と報告し、コンソールには HARDWARE FAILURE info:280 asc:80,96 と表示されるでしょう。

  `mt rewind` を使ってテープを巻戻します。

  次からはテープの操作はうまくいくでしょう。

16.8. フロッピーディスクへのバックアップ

16.8.1. データをバックアップするのにフロッピーは使えますか?

フロッピーディスクは以下の理由によって、 実際にバックアップをつくるための適切なメディアではありません。

- メディアの信頼性が (特に長期間の場合) 低い。
- バックアップとリストアがとても遅い。
- 容量が非常に小さい (1 ダースかそこらのフロッピーディスクに ハードディスク全体をバックアップしていた時代は、はるか遠くに過ぎ去りました)。

しかしながら、データをバックアップする他の手段がないのなら、 バックアップを取らないよりもフロッピーディスクを使う方がましでしょう。

フロッピーディスクを使用せざるを得ないときは、 品質のよいディスクを使用してください。事務所のその辺に数年転がっていたフロッピーは使わない方が良いでしょう。評判のよいメーカの新しいディスクを使用することが理想です。

16.8.2. それではどうやってデータをフロッピーにバックアップするのですか?

フロッピーにバックアップする最もよい方法は、`-M` (マルチボリューム) オプション付きで tar(1) コマンドを使用することです。これで、複数のフロッピーにわたってバックアップすることが可能になります。

カレントディレクトリとサブディレクトリ内のすべてのファイルをバックアップするには、 以下のコマンドを (`root` 権限で) 使用します。

```
# tar Mcvf /dev/fd0 *
```

1 枚目のフロッピーが一杯になると、tar(1) は次のボリュームを挿入するように要求します (tar(1) はさまざまなメディアを扱えるので、ボリュームと表示します。この文脈ではフロッピーディスクのことです)。

```
Prepare volume #2 for /dev/fd0 and hit return:
```

指定したファイルがすべて保存されるまで (ボリューム番号を増やしながら) これが繰り返されます。

16.8.3. バックアップを圧縮することはできませんか?

残念なことに tar(1) はマルチボリュームアーカイブに対して、`-z` オプションを使うことができません。もちろん、すべてのファイルを gzip(1) で圧縮し、それらを tar(1) を用いてフロッピーに保存して、それから再び gunzip(1) することはできます。

16.8.4. どのようにしてバックアップをリストアしたらいいのでしょうか?

すべてのアーカイブをリストアするには以下のようにします。

```
# tar Mxvf /dev/fd0
```

特定のファイルだけをリストアするには 2 つの方法があります。1 つ目は、1 枚目のフロッピーを用いて以下のようにするものです。

```
# tar Mxvf /dev/fd0 filename
```

tar(1) ユーティリティは、必要なファイルを見つけるまで次のディスクを挿入するように要求します。

もう 1 つは、必要なファイルがどのフロッピーに保存されているか分かっている場合、そのフロッピーを挿入して上記と同じコマンドを使用するだけでもよいです。あるフロッピー上にある 1 番目のファイルが、その前のフロッピーから続いている場合は、そのファイルのリストアを要求していなくても tar(1) はそれをリストアできないと警告することに注意してください!

16.9. バックアップの基本

主なバックアッププログラムは dump(8), tar(1), cpio(1) の三つです。

16.9.1. ダンプとリストア

伝統的な UNIX® のバックアッププログラムは `dump` と `restore` です。これらはファイルシステムによって作成されるファイル、リンク、 ディレクトリといった抽象の下位にある、ディスクブロックの集合としてドライブを操作します。`dump` はデバイス上のファイルシステム全体をバックアップします。ファイルシステムの一部分だけ、または二つ以上のファイルシステムにわたるディレクトリツリーをバックアップすることはできません。 `dump` はファイルおよびディレクトリをテープに書き込まずに、 ファイルおよびディレクトリを含んだ raw データブロックを書き込みます。

注記

ルートディレクトリで `dump` を使った場合、 `/home`, `/usr` など、他の多くのディレクトリはバックアップされません。これらのディレクトリは通常、 他のファイルシステムへのマウントポイントであったり、シンボリックリンクとなっているためです。

`dump` には AT&T UNIX のバージョン 6 (およそ 1975 年) の初期から残っている癖があります。デフォルトのパラメタは、現在利用可能な高密度メディア (最大 62,182 ftpi) ではなく、9 トラックテープ (6250 bpi) に最適な値となっています。現在のテープドライブの容量を利用するために、これらのデフォルト値をコマンドラインで上書きしなければなりません。

`rdump` と `rrestore` を用いて他のコンピュータに接続されているテープドライブにネットワーク経由でデータをバックアップすることも可能です。 どちらのプログラムもリモートのテープドライブにアクセスするために `rcmd` および `ruserok` に依存して

います。したがって、バックアップを実行するユーザがリモートコンピュータの `.rhosts` ファイルに書かれていなければなりません。 `rdump` および `rrestore` の引数はリモートコンピュータに適切なものを用いなければなりません。FreeBSD コンピュータから `komodo` と呼ばれる Sun に接続されている Exabyte テープへ `rdump` するには以下のようにします。

```
# /sbin/rdump 0dsbfu 54000 13000 126 komodo:/dev/nsa8 /
dev/da0a 2>&1
```

注意: `.rhosts` 認証を許可することには、セキュリティに関する暗黙の仮定があります。あなたの置かれている状況を注意深く調べてください。

`ssh` 越しに `dump` と `restore` をより安全な形で使うこともできます。

例16.1 ssh 越しの **dump** の利用

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh1 -c ↺
blowfish \
          targetuser@targetmachine.example.com dd ↺
of=/mybigfiles/dump-usr-l0.gz
```

または、環境変数 `RSH` を設定して、`dump` の組み込み機能を利用する。

例16.2 **RSH** を設定した ssh 越しの **dump** を利用

```
# RSH=/usr/bin/ssh /sbin/dump -0uan -f ↺
targetuser@targetmachine.example.com:/dev/sa0
```

16.9.2. tar

tar(1) は AT&T UNIX の バージョン 6 (1975 年ごろ) にまで遡ることができます。`tar` はファイルシステムと協調して動作し、ファイルとディレクトリをテープに書き込みます。`tar` は cpio(1) で使用可能なフルレンジのオプションには対応していませんが、`tar` には `cpio` が使用するような奇妙なコマンドパイプラインは必要ありません。

`tar` の多くの版はネットワーク経由のバックアップには対応していません。FreeBSD が使用している GNU 版の `tar` は、`rdump` と同じ構文でリモートデバイスに対応してい

ます。komodo と呼ばれる Sun に接続された Exabyte テープドライブに対して tar を実行するには以下のようにします。

```
# /usr/bin/tar cf komodo:/dev/nsa8 . 2>&1
```

リモートデバイスに対応していない版に対しては、パイプラインと rsh を使用してリモートテープドライブにデータを送ることができます。

```
# tar cf - . | rsh hostname dd of=tape-device obs=20b
```

ネットワークを越えたバックアップのセキュリティを懸念しているなら、 rsh の代わりに ssh を使うべきです。

16.9.3. cpio

cpio(1) は本来 UNIX® ファイルを磁気メディアで交換するためのプログラムです。cpio はバイトスワッピング、多くの異なるアーカイブフォーマットの書き込みオプションがあり (それ以外にも多数のオプションがあります)、パイプで他のプログラムにデータを渡すこともできます。この最後にあげた特徴が、cpio をインストールメディアとしては優れた選択肢にしています。cpio はディレクトリツリーの探索の機能はなく、ファイルリストは stdin からの入力でなくてはなりません。

cpio はネットワーク経由のバックアップには対応していません。以下のようにパイプラインと rsh を用いてリモートテープドライブにデータを送ることができます。

```
# for f in directory_list; do
find $f >> backup.list
done
# cpio -v -o --format=newc < backup.list | ssh user@host
 "cat > backup_device"
```

directory_list はバックアップしたいディレクトリのリストで、*user@host* はバックアップを実行したいユーザとホスト名の組であり、*backup_device* はバックアップを書き込みたいデバイスです (たとえば /dev/nsa0)。

16.9.4. pax

pax(1) は tar と cpio に対する IEEE/POSIX® の回答です。長年の間、さまざまな版の tar と cpio は互いにわずかに非互換になってきていました。それらをしらみ潰しに標準化する代わりに、POSIX® は新しいアーカイブユーティリティを作りました。pax は、いくつもの cpio や tar のフォーマットの読み書きに対応しようと試みているほか、専用に新しいフォーマットを開発しました。コマンド群は tar よりも cpio の方にいくぶん似ています。

16.9.5. Amanda

Amanda (Advanced Maryland Network Disk Archiver) は単一のプログラムではなく、クライアント/サーバ型のバックアップシステムです。 Amanda サーバは、Amanda クライアントを有する ネットワークに接続されたコンピュータからデータを受け取り、 備え付けられたテープドライブにバックアップします。 いくつもの大容量ディスクを備えたサイトでの共通の問題は、データディレクトリをテープにバックアップするのに時間がかかりすぎることです。Amanda はこの問題を解決します。Amanda は「ホールディングディスク」を使用して、同時に複数のファイルシステムのバックアップを行うことができます。 Amanda の設定ファイルにかかれたすべてのファイルシステムのフルバックアップを特定の間隔でとるために「アーカイブセット」と呼ばれるテープグループを作成します。「アーカイブセット」には 夜間に作成されるすべてのファイルシステムの増分 (または差分) のバックアップも含まれます。 障害が起きたファイルシステムのリストアには、最も新しいフルバックアップと増分のバックアップが必要です。

設定ファイルでは、バックアップの制御と Amanda によるネットワークトラフィック量を設定します。 Amanda は上記のバックアッププログラムのいずれかを使ってデータをテープに書き込みます。Amanda は port または package として利用可能です。デフォルトではインストールされていません。

16.9.6. 何もしない

「何もしない」というのはコンピュータのプログラムではありませんが、 バックアップの戦略として最も広く採用されています。 これには初期投資が必要ありません。 従わなければならないバックアップスケジュールもありません。 ただ何もしないだけです。データに何か起きたら苦笑いして耐えてください!

あなたにとって時間やデータの価値が少ないか、あるいはまったくないのであれば「何もしない」 のはあなたのコンピュータに最も適したバックアッププログラムでしょう。しかし注意してください。UNIX® は便利なツールです。 6 ヶ月も使用していれば、 あなたにとって価値のあるファイルの山が出来上がっているでしょう。

「何もしない」 ことはコンピュータが同じものをもう一度作り直すことのできる `/usr/obj` やその他のディレクトリツリーについては適切なバックアップ方法です。一例として、このハンドブックの HTML 版 または PostScript® 版を構成するファイルがあります。 これらの文書形式は SGML ファイルから作成されたものです。 HTML または PostScript® ファイルのバックアップは必要ありません。 SGML ファイルは定期的にバックアップされています。

16.9.7. どのバックアッププログラムが最適ですか?

dump(8) です。以上。 Elizabeth D. Zwicky はここで検討したプログラムすべてについて拷問的なテストを行いました。 すべてのデータと UNIX® ファイルシステムの状態すべてを保存するのに最適なのは、明らかに `dump` です。Elizabeth は多種多様の

特異な状態 (いくつかはあまり珍しくないものもあります) を含むファイルシステムを作成し、それらのファイルシステムのバックアップとリストアを行って、それぞれのプログラムのテストを行いました。特異な状態とは、ホールがあるファイル、ホールとヌルブロックがあるファイル、奇妙な文字をファイル名に持つファイル、読み取り不可、書き込み不可のファイル、デバイスファイル、バックアップ中のファイルのサイズ変更、バックアップ中のファイルの作成および削除、などです。彼女は 1991 年 10 月の LISA V で結果を発表しています。torture-testing Backup and Archive Programs を参照してください。

16.9.8. 緊急時のリストア手順

16.9.8.1. 惨事が起きる前に

発生する可能性があるどのような惨事に対しても、備えるのに必要な手順は以下の 4 ステップだけです。

最初に、各ディスクのディスクラベルとファイルシステムテーブル (`/etc/fstab`)、ブートメッセージ全体をそれぞれ 2 枚ずつ印刷します (たとえば `disklabel da0 | lpr`)。

2 番目に、ブートフロッピーと fix-it フロッピー (`boot.flp` および `fixit.flp`) にそのシステムのデバイスがすべて含まれているか確認します。最も簡単に確認する方法は、フロッピーをドライブに入れてマシンをリブートしてブートメッセージを確認することです。あなたのシステムのデバイスのすべてが含まれ、機能していれば 3 番目の手順に進んでください。

さもなければ、そのシステムのすべてのディスクをマウントでき、テープドライブにもアクセスできるカーネルを備えたカスタムブートフロッピーを 2 枚作成する必要があります。これらのフロッピーディスクには `fdisk`, `disklabel`, `newfs`, `mount` と、利用するバックアッププログラムが入っていなければなりません。これらのプログラムはスタティックリンクされていなければなりません。`dump` を使用するのなら、このフロッピーには `restore` も含まれていなければなりません。

3 番目に、定期的にバックアップテープを作成します。最後のバックアップの後で行われた変更は、回復できずに失われます。バックアップテープにライトプロテクトを施してください。

4 番目に、フロッピーディスク (`boot.flp` と `fixit.flp`、か、第 2 段階で作成した 2 枚のカスタムブートフロッピーディスクのどちらか) およびバックアップテープのテストをします。手順のメモを作りましょう。このメモはブートフロッピー、印刷した紙、バックアップテープと一緒に保存しておきます。リストアを行うときには、このメモがバックアップテープを壊すのを防ぐくらい取り乱しているかもしれません (どのように? `tar xvf /dev/sa0` の代わりに、うっかり `tar cvf /dev/sa0` と入力してバックアップテープを上書きしてしまうかもしれません)。

訳注

上書きはライトプロテクトをしておけば防げますが、 何らかの原因でプロテクトがはずれているかもしれません。ちなみに訳者の経験から言えば、上のようなミスタイプは結構起きます。

安全性を増すために、毎回、ブートフロッピーを作成し、2 巻のバックアップテープを取ります。一方を離れた場所に保管します。離れた場所は同じ事務所の建物の地下室ではいけません。世界貿易センタービルにあった数多くの会社は、苦い経験によりこの教訓を得ました。離れた場所とは、コンピュータやディスクドライブから十分な距離を取って 物理的に分離されていなければなりません。

例16.3 ブートフロッピーを作成するスクリプト

```
#!/bin/sh
#
# create a restore floppy
#
# format the floppy
#
PATH=/bin:/sbin:/usr/sbin:/usr/bin

fdformat -q fd0
if [ $? -ne 0 -]
then
  echo "Bad floppy, please use a new one"
  exit 1
fi

# place boot blocks on the floppy
#
disklabel -w -B /dev/fd0c fd1440

#
# newfs the one and only partition
#
newfs -t 2 -u 18 -l 1 -c 40 -i 5120 -m 5 -o space /dev/fd0a

#
# mount the new floppy
#
mount /dev/fd0a /mnt

#
# create required directories
```

```
#
mkdir /mnt/dev
mkdir /mnt/bin
mkdir /mnt/sbin
mkdir /mnt/etc
mkdir /mnt/root
mkdir /mnt/mnt   # for the root partition
mkdir /mnt/tmp
mkdir /mnt/var

#
# populate the directories
#
if [ ! -x /sys/compile/MINI/kernel -]
then
  cat << EOM
The MINI kernel does not exist, please create one.
Here is an example config file:
#
# MINI - A kernel to get FreeBSD onto a disk.
#
machine         "i386"
cpu             "I486_CPU"
ident           MINI
maxusers        5

options         INET                    # needed for _tcp ↺
_icmpstat _ipstat
                                        #            _udpstat ↺
_tcpstat _udb
options         FFS                     #Berkeley Fast File ↺
System
options         FAT_CURSOR              #block cursor in ↺
syscons or pccons
options         SCSI_DELAY=15           #Be pessimistic about ↺
Joe SCSI device
options         NCONS=2                 #1 virtual consoles
options         USERCONFIG              #Allow user ↺
configuration with -c XXX

config          kernel root on da0 swap on da0 and da1 dumps ↺
on da0

device          isa0
device          pci0

device          fdc0 at isa? port "IO_FD1" bio irq 6 drq 2 ↺
vector fdintr
device          fd0 at fdc0 drive 0

device          ncr0

device          scbus0
```

```
device          sc0 at isa? port "IO_KBD" tty irq 1 vector ↺
scintr
device          npx0 at isa? port "IO_NPX" irq 13 vector ↺
npxintr

device          da0
device          da1
device          da2

device          sa0

pseudo-device   loop            # required by INET
pseudo-device   gzip            # Exec gzipped a.out's
EOM
  exit 1
fi

cp -f /sys/compile/MINI/kernel /mnt

gzip -c -best /sbin/init > /mnt/sbin/init
gzip -c -best /sbin/fsck > /mnt/sbin/fsck
gzip -c -best /sbin/mount > /mnt/sbin/mount
gzip -c -best /sbin/halt > /mnt/sbin/halt
gzip -c -best /sbin/restore > /mnt/sbin/restore

gzip -c -best /bin/sh > /mnt/bin/sh
gzip -c -best /bin/sync > /mnt/bin/sync

cp /root/.profile /mnt/root

cp -f /dev/MAKEDEV /mnt/dev
chmod 755 /mnt/dev/MAKEDEV

chmod 500 /mnt/sbin/init
chmod 555 /mnt/sbin/fsck /mnt/sbin/mount /mnt/sbin/halt
chmod 555 /mnt/bin/sh /mnt/bin/sync
chmod 6555 /mnt/sbin/restore

#
# create the devices nodes
#
cd /mnt/dev
./MAKEDEV std
./MAKEDEV da0
./MAKEDEV da1
./MAKEDEV da2
./MAKEDEV sa0
./MAKEDEV pty0
cd /

#
# create minimum file system table
```

```
#
cat &gt; /mnt/etc/fstab &lt;&lt;EOM
/dev/fd0a    /    ufs    rw  1  1
EOM

#
# create minimum passwd file
#
cat > /mnt/etc/passwd <<EOM
root:*:0:0:Charlie &:/root:/bin/sh
EOM

cat > /mnt/etc/master.passwd <<EOM
root::0:0::0:0:Charlie &:/root:/bin/sh
EOM

chmod 600 /mnt/etc/master.passwd
chmod 644 /mnt/etc/passwd
/usr/sbin/pwd_mkdb -d/mnt/etc /mnt/etc/master.passwd

#
# umount the floppy and inform the user
#
/sbin/umount /mnt
echo "The floppy has been unmounted and is now ready."
```

16.9.8.2. 惨事の後は

重要な問題は、ハードウェアが生き残ったかどうかです。 定期的にバックアップを取っていれば、ソフトウェアについて心配する必要はありません。

ハードウェアに障害があれば、 コンピュータを使用する前にその部品を交換してください。

ハードウェアに問題が無ければ、フロッピーを確認してください。 カスタムブートフロッピーディスクを使用しているのであれば、シングルユーザモードでブートして (`boot:` プロンプトで `-s` を入力します)、次の段落は飛ばしてください。

`boot.flp` と `fixit.flp` を使用しているのであればこのまま読み進めてください。 `boot.flp` フロッピーをフロッピードライブに入れて、コンピュータを起動してください。 本来のインストールメニューが画面に表示されます。`Fixit--Repair mode with CDROM or floppy.` オプションを選択します。指示された通り `fixit.flp` をいれてください。`restore` とその他必要となるプログラムは `/mnt2/stand` にあります。

そして、ファイルシステムを一つずつ回復します。

最初のディスクのルートパーティションを `mount` してみてください (たとえば `mount /dev/da0a /mnt`)。 ディスクラベルが破壊されている場合

は、`disklabel` を用いてあらかじめ印刷して保存しておいた通りにパーティションを作り直し、ディスクラベルを作成してください。`newfs` を使用してファイルシステムを作り直します。 ルートパーティションを読み書き可能にマウントし直します (`mount -u -o rw /mnt`)。バックアッププログラムとバックアップテープを使用して、このファイルシステムのデータを回復します (たとえば `restore vrf /dev/sa0`)。ファイルシステムをアンマウントします (たとえば `umount /mnt`)。障害を受けたファイルシステムそれぞれについて繰り返してください。

システムが動き出したら、 新しいテープにデータをバックアップしてください。 どのような理由で再び事故が起きたり、データが失われるかわかりません。 これに数時間を費すことで、後々の災難から救われます。

16.10. ネットワーク、メモリ、そしてファイルベースのファイルシステム

########: Marc Fonvieille.

FreeBSD にはフロッピーや CD, ハードディスクなどの手元の計算機に取り付けたディスクの他に、別の形態のディスク、仮想ディスク、もあります。

これには、Network File System のようなネットワークファイルシステムや Coda, メモリベースのファイルシステムおよびファイルベースのファイルシステムがあります。

稼働させている FreeBSD のバージョンによって、ファイルベースおよびメモリベースのファイルシステムを作成したり操作するために、異なるツールを使用しなければならないでしょう。

注記

FreeBSD 4.X の使用者は必要なデバイスを作成するために MAKEDEV(8) を使用しなければならないでしょう。FreeBSD 5.0 以降では、devfs(5) がデバイスノードを自動的に割り当ててくれるので、使用者が意識する必要はありません。

16.10.1. FreeBSD 4.X でファイル中に構築されるファイルシステム

vnconfig(8) ユーティリティを使えば擬似ディスクデバイスを設定し、 有効にすることができます。 vnode とはファイルの内部的な表現方法であり、 ファイルに関する操作の中心となるものです。つまり、vnconfig(8) はファイルシステムを生成したり操作したりするためにファイルを用いるのです。一つ例を挙げると、ファイルに収められたフロッピーや CD-ROM のイメージをマウントするために用いることができます。

vnconfig(8) を使用するためには、カーネルが vn(4) デバイスに対応している必要があります。そうでなければ、カーネルコンフィギュレーションファイルに 次の行を追加してカーネルを再構築し、システムを再起動してください。

```
pseudo-device vn
```

既にあるファイルシステムイメージのマウント

例16.4 FreeBSD 4.X での vnconfig を用いた既存のファイルシステムイメージのマウント

```
# vnconfig vn0 diskimage
# mount /dev/vn0c /mnt
```

vnconfig(8) を用いたファイルシステムイメージの新規作成

例16.5 **vnconfig** を用いたファイルベースディスクの新規作成

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# vnconfig -s labels -c vn0 newimage
# disklabel -r -w vn0 auto
# newfs vn0c
Warning: 2048 sector(s) in last cylinder unallocated
/dev/vn0c:     10240 sectors in 3 cylinders of 1 tracks, 4096 ↺
sectors
        5.0MB in 1 cyl groups (16 c/g, 32.00MB/g, 1280 i/g)
super-block backups (for fsck -b #) at:
 32
# mount /dev/vn0c /mnt
# df /mnt
Filesystem  1K-blocks     Used    Avail Capacity  Mounted on
/dev/vn0c        4927        1     4532     0%    /mnt
```

16.10.2. FreeBSD 5.X でファイル中に構築されるファイルシステム

mdconfig(8) ユーティリティは FreeBSD 5.X において メモリディスク (md(4)) を設定し、有効にするために使用されます。mdconfig(8) を使用するためには md(4) モジュー

ルを読み込むか、カーネルコンフィギュレーションファイルに md(4) デバイスを追加してカーネルを再構築し、システムを再起動してください。

```
device md
```

mdconfig(8) コマンドは、三つのタイプのメモリベース仮想ディスクに対応しています。malloc(9) を用いて割り当てられたメモリディスク、ファイルをベースにしたメモリディスク、およびスワップ領域をベースにしたメモリディスクです。想定される使用法は、ファイル内に保持されたフロッピーイメージまたは CD イメージをマウントすることです。

既にあるファイルシステムイメージのマウント

例16.6 FreeBSD 5.X での **mdconfig** を用いた既存のファイルシステムイメージのマウント

```
# mdconfig -a -t vnode -f diskimage -u 0
# mount /dev/md0c /mnt
```

mdconfig(8) を用いたファイルシステムイメージの新規作成

例16.7 **mdconfig** を用いたファイルシステムイメージの新規作成

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdconfig -a -t vnode -f newimage -u 0
# disklabel -r -w md0 auto
# newfs md0c
/dev/md0c: 5.0MB (10240 sectors) block size 16384, fragment ↺
size 2048
 using 4 cylinder groups of 1.27MB, 81 blks, 256 inodes.
super-block backups (for fsck -b #) at:
 32, 2624, 5216, 7808
# mount /dev/md0c /mnt
# df /mnt
Filesystem  1K-blocks     Used    Avail Capacity  Mounted on
/dev/md0c        4846        2     4458     0%    /mnt
```

`-u` オプションを用いて ユニット番号を指定しない場合、mdconfig(8) は未使用のデバイスを自動的に選択するために md(4) デバイスの auto-unit 機能を使用します。割り当てられたユニットの名前は `md4` のように標準出力に出力されます。mdconfig(8) の詳細についてはマニュアルページを参照してください。

注記

FreeBSD 5.1-RELEASE から、従来の disklabel(8) プログラムは bsdlabel(8) ユーティリティに置き換えられました。bsdlabel(8) では、使用されていないオプションおよびパラメタの数多くが削除されました。たとえば `-r` オプションは bsdlabel(8) では取り除かれました。詳細については bsdlabel(8) マニュアルページを参照してください。

mdconfig(8) ユーティリティは大変役に立ちますが、 ファイルベースのファイルシステムを作成するために、 多くのコマンドの入力が必要となります。FreeBSD 5.0 では mdmfs(8) と呼ばれるツールも用意されています。このプログラムは mdconfig(8) を用いて md(4) ディスクを設定し、newfs(8) を用いて UFS ファイルシステムを作成し、mount(8) を用いてマウントします。たとえば、上記と同じファイルシステムを作成し、マウントしたい場合は、下記のように入力するだけです。

例16.8 `mdmfs` を用いたファイルベースディスクの設定とマウント

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdmfs -F newimage -s 5m md0 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0        4846    2  4458     0%    /mnt
```

ユニット番号を指定せずに `md` オプションを使用した場合、mdmfs(8) は未使用のデバイスを自動的に選択するために md(4) デバイスの auto-unit 機能を使用します。mdmfs(8) についての詳細はマニュアルページを参照してください。

16.10.3. FreeBSD 4.X でのメモリベースのファイルシステム

md(4) ドライバは FreeBSD 4.X においてメモリファイルシステムを作成するために単純で効果的な手段です。メモリを割り当てるために malloc(9) 関数が使用されます。

vnconfig(8) を用いて作成したファイルシステムを例に取ると、以下のようにします。

例16.9 FreeBSD 4.X での md メモリディスク

```
# dd if=newimage of=/dev/md0
5120+0 records in
5120+0 records out
# mount /dev/md0c /mnt
# df /mnt
Filesystem  1K-blocks     Used    Avail Capacity  Mounted on
/dev/md0c        4927        1     4532     0%    /mnt
```

詳細については md(4) マニュアルページを参照してください。

16.10.4. FreeBSD 5.X でのメモリベースのファイルシステム

メモリベースおよびファイルベースのファイルシステムに対しても 同じツール (mdconfig(8) または mdmfs(8)) を使用できます。メモリベースのファイルシステムに対する記憶領域は malloc(9) 関数を用いて割り当てられます。

例16.10 **mdconfig** を用いたメモリベースディスクの新規作成

```
# mdconfig -a -t malloc -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment ↺
size 2048
 using 4 cylinder groups of 1.27MB, 81 blks, 256 inodes.
 with soft updates
super-block backups (for fsck -b #) at:
 32, 2624, 5216, 7808
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md1        4846    2  4458     0%    /mnt
```

例16.11 `mdmfs` を用いたメモリベースディスクの新規作成

```
# mdmfs -M -s 5m md2 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md2        4846    2  4458     0%    /mnt
```

mdconfig(8) のコマンドラインの `malloc` を `swap` に置き換えることで、malloc(9) 関数によるファイルシステムを使用する代わりに スワップ領域を使用することが可能です。デフォルトでは mdmfs(8) ユーティリティはスワップベースのディスクを作成します (`-M` なし)。詳細は mdconfig(8) および mdmfs(8) マニュアルページを参照してください。

16.10.5. システムからメモリディスクを切り離す

メモリベースまたはファイルベースのファイルシステムが使用されていない場合、 すべてのリソースをシステムに開放するべきです。 はじめにファイルシステムをアンマウントします。次にシステムからディスクを切り離し、リソースを開放するために mdconfig(8) を使用します。

たとえば `/dev/md4` によって使用されたすべてのリソースを切り離し、開放するには以下のようにします。

```
# mdconfig -d -u 4
```

`mdconfig -l` コマンドを使用することによって、設定された md(4) デバイスについての情報を表示することが可能です。

FreeBSD 4.X では vnconfig(8) はデバイスを切り離すのに使用されます。たとえば `/dev/vn4` によって使用されたすべてのリソースを切り離し、開放するには以下のようにします。

```
# vnconfig -u vn4
```

16.11. ファイルシステムのスナップショット

##: Tom Rhodes.

FreeBSD 5.0 は Soft Updates と協調するファイルシステムスナップショットという新しい機能を提供します。

スナップショットは指定したファイルシステムのイメージを作成し、また、ファイルとして扱うことができるようになります。 スナップショットファイルはアクションが実行されるファイルシステム内で作成されなければなりません。 また、ユーザは一つのファイルシステムあたり 20 までスナップショットを作成することができます。 有効なスナップショットはスーパーブロック内に記録されるので、 リブートしてから永続的にアンマウントおよびリマウントを記録します。 スナップショットが必要無くなったときは、 標準の rm(1) コマンドを用いて削除することができます。 スナップショットはどんな順番で削除してもよいのですが、 その他のスナップショットが開放されたブロックのうちいくらかをおそらく必要とするので、使用されていたすべてのスペースを得られるとは限りません。

初めてスナップショットを作成すると、root でさえも書き込めないように schg フラグ (chflags(1) のマニュアルページを参照) が設定されます。unlink(1) コマンドは、スナップショットに schg フラグが設定されていてもそれらを削除することのできる例外です。したがって、スナップショットファイルを削除する前に、schg フラグをクリアする必要はありません。

スナップショットは mount(8) コマンドを用いて作成されます。/var のスナップショットを /var/snapshot/snap に作成したいときは、以下のコマンドを使用します。

```
# mount -u -o snapshot /var/snapshot/snap /var
```

また、スナップショットを作成するのに mksnap_ffs(8) も使えます。

```
# mksnap_ffs /var /var/snapshot/snap
```

スナップショットにはいくつかの利用法があります。

- スナップショットをバックアップ目的に使用する管理者もいます。 なぜならスナップショットは CD やテープに転送できるからです。

- ファイルの完全性を検証するために、 fsck(8) をスナップショットに実行してもよいでしょう。スナップショットをマウントしたときにそのファイルシステムがクリーンであったとすると、そのスナップショットをマウントするときはいつでもクリーンな (そして変更のない) 結果を得るでしょう。これは本質的には バックグラウンド fsck(8) が行うことです。

- スナップショット上で dump(8) ユーティリティを実行すると、 スナップショットのファイルシステムとタイムスタンプが一致するダンプが返されるでしょう。dump(8) は -L オプションを使用することで、 一つのコマンドでスナップショットをとり、ダンプイメージを作成して、スナップショットを削除することが可能です。

- ファイルシステムの「凍結された」イメージとしてスナップショットを mount(8) します。/var/snapshot/snap のスナップショットを mount(8) するには以下のようにします。

```
# mdconfig -a -t vnode -f /var/snapshot/snap -u 4
# mount -r /dev/md4 /mnt
```

これで /mnt にマウントした 凍結状態の /var ファイルシステム構造を探索できます。すべてがスナップショットが作成された時と同じ状態になるはずです。ただし、 以前に作成されたスナップショットがサイズ 0 のファイルとして現れることが唯一の例外です。スナップショットの使用を終えた場合、以下のようにアンマウントできます。

```
# umount /mnt
# mdconfig -d -u 4
```

softupdates およびファイルシステムスナップショットに関する詳細については、http://www.mckusick.com/ にある Marshall Kirk McKusick のウェブサイトを参照してください。ここには技術的な論文もあります。

16.12. ファイルシステムクォータ

クォータは OS の持っているオプショナルな機能であり、ファイルシステム毎にユーザやグループのメンバが使用するディスク容量やファイルの数を制限することができます。この機能は、あるユーザやグループに割り当てられるリソースの量を制限することが望ましいようなタイムシェアリングシステムにおいてよく用いられます。この機能を用いることによって使用可能なディスク容量の全てを一人のユーザやユーザのグループが使ってしまうことを防ぐことができます。

16.12.1. ディスククォータを使うためのシステム設定

ディスククォータの設定を始める前に、 まずはカーネルにクォータが組み込まれていることを確認しましょう。カーネルのコンフィグレーションファイルに次の行を入れます。

```
options QUOTA
```

標準の GENERIC カーネルでは、 この機能は有効になっていませんので、 ディスククォータを利用するためには上記を設定後カーネルを構築しなおし、作成されたカスタムカーネルをインストールしなければいけません。 カーネルのコンフィグレーションに関しては 9章FreeBSD カーネルのコンフィグレーション をご覧ください。

次に /etc/rc.conf でディスククォータを有効にする必要があります。次の行を加えましょう。

```
enable_quotas="YES"
```

起動時の動作をさらに細かくコントロールするためにもう一つ設定用の変数があります。通常、起動時には quotacheck(8) によりそれぞれのファイルシステムのクォータの整合性がチェックされます。quotacheck(8) の役割は、クォータデータベースのデータが正しくファイルシステム上のデータを反映しているか確認することです。 これはかなり時間を食う処理であり、起動にかかる時間に大きな影響を及ぼします。このステップをとばしたい人のために /etc/rc.conf に次の変数が用意されています。

```
check_quotas="NO"
```

もし 3.2-RELEASE よりも前の FreeBSD を使っているならば設定はもっと単純で、一つの変数のみです。次の行を /etc/rc.conf で設定してください。

```
check_quotas="YES"
```

最後に、ファイルシステム毎にディスククォータを有効にするために /etc/fstab を編集する必要があります。ここでユーザもしくはグループ、あるいはその両方にクォータを設定することができるのです。

あるファイルシステム上にユーザ毎のクォータを有効にする場合には、 /etc/fstab 中でクォータを有効にしたいファイルシステムエントリのオプション部に userquota を加えます。例えば次のようになります。

```
/dev/da1s2g   /home    ufs rw,userquota 1 2
```

同様に、グループクォータを有効にするには userquota キーワードの代わりに groupquota を用います。ユーザとグループの両方のクォータを有効にするには次のようにします。

```
/dev/da1s2g    /home    ufs rw,userquota,groupquota 1 2
```

デフォルトでは、クォータファイルはそのファイルシステムのルートディレクトリに ユーザ用、グループ用それぞれ quota.user , quota.group という名前で置かれます。さらに詳しい情報は fstab(5) をご覧ください。fstab(5) マニュアルには別の場所を指定することができると書いてはありますが、 あまり勧められません。なぜなら、 様々なクォータ関係のユーティリティがそれにうまく対処できるようにないためです。

この時点で、一度システムを再起動して新しいカーネルで立ち上げましょう。/etc/rc が自動的に適当なコマンドを実行し、 /etc/fstab で有効にした全てのクォータ用に初期ファイルを作ってくれます。従って、空のクォータファイルを手で作る必要は一切ありません。

通常の運用では quotacheck(8) や quotaon(8), quotaoff(8) といったコマンドを手で動かす必要はないのですが、慣れるためにもこれらのマニュアルは読んでおきましょう。

16.12.2. クォータリミットの設定

一旦クォータを有効にしたら本当に有効になっているのか確認しておきましょう。簡単な方法は次のコマンドを実行することです。

```
# quota -v
```

ディスクの使用状況と、クォータが有効になっているファイルシステムのクォータリミットが一行にまとめて出力されるでしょう。

さあ、edquota(8) でクォータリミットを設定する準備ができました。

ユーザやグループが使用できるディスク容量や作成できるファイルの数に制限をかけるにはいくつかのオプションがあります。割り当てディスク容量を制限 (ブロッククォータ) することもファイル数を制限 (inode クォータ) することも、両者を組み合わせることもできるのです。これらの制限はそれぞれさらに二つのカテゴリ、ハードリミットとソフトリミット、に分けることができます。

ハードリミットを越えることはできません。　あるユーザが一旦ハードリミットにたっした場合、そのファイルシステムではそれ以上の割り当ては望めません。例えばあるファイルシステム上に 500 ブロックのハードリミットが設定されており現在 490 ブロックを使用している場合、さらに 10 ブロックしか使えないのです。11 ブロックを使おうとすると失敗します。

一方、 ソフトリミットはある限られた時間内であれば越えることができます。 この時間は猶予期間として知られており、デフォルトでは 1 週間です。あるユーザが自分のソフトリミットを猶予期間よりも長い間越えているとソフトリミットはハードリミットに変わり、それ以上使用することはできなくなります。 ユーザがソフトリミットよりも減らせば猶予期間はリセットされます。

以下は edquota(8) コマンドを実行した時に見ることになるであろう例です。edquota(8) コマンドが起動されると環境変数 EDITOR で指定されるエディタに入ります。EDITOR が設定されていない場合には vi が起動されます。ここでクォータリミットを編集します。

```
# edquota -u test
```

```
Quotas for user test:
/usr: blocks in use: 65, limits (soft = 50, hard = 75)
        inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: blocks in use: 0, limits (soft = 50, hard = 75)
        inodes in use: 0, limits (soft = 50, hard = 60)
```

通常、クォータが有効になっているファイルシステム毎に 2 行あります。一つはブロックリミット用でもう一つは inode リミット用です。クォータリミットを変更したいところを書き変えるだけでかまいません。たとえばこのユーザのブロックリミットを、ソフトリミットは 50 から 500 へ、ハードリミットは 75 から 600 に変更する場合、

```
/usr: blocks in use: 65, limits (soft = 50, hard = 75)
```

から

```
 /usr: blocks in use: 65, limits (soft = 500, hard = 600)
```

へ書き換えます。新しいクォータリミットはエディタを終了すれば設定されます。

ある範囲の UID に対してクォータリミットを設定したい場合がありますが、このような時には edquota(8) コマンドの -p オプションを使うといいでしょう。まず、あるユーザに割り当てたいクォータリミットを設定し、次に edquota -p protouser startuid-enduid を実行するのです。例えばユーザ test にお望みのクォータリミットが付いて

いるとしましょう。次のコマンドにより 10,000 から 19,999 の間の UID に対して同じクォータリミットを付けることができるのです。

```
# edquota -p test 10000-19999
```

さらに詳しいことは edquota(8) のマニュアルページをご覧ください。

16.12.3. クォータリミットとディスク使用状況のチェック

quota(1) または repquota(8) といったコマンドを使ってクォータリミットやディスクの利用状況を確認することができます。 quota(1) コマンドは個々のユーザやグループのクォータやディスク利用状況を確認するのに使えます。 ユーザは自身のクォータ、そして所属するグループのグループのみ確認することができます。スーパーユーザのみが他のユーザや所属していないグループのクォータと利用状況を見ることができます。
repquota(8) コマンドを使うと、クォータが有効になっているファイルシステム用の全てのクォータやディスク容量のサマリを得ることができます。

以下は二つのファイルシステムにクォータ制限がかけられているユーザに対する`quota -v` コマンドの出力例です。

```
Disk quotas for user test (uid 1002):
     Filesystem  blocks   quota   limit   grace   files   quota   ↺
limit   grace
          /usr      65*     50      75   5days       7      50
  60
       /usr/var       0      50      75               0      50
  60
```

上の例で、`/usr` ファイルシステム上ではこのユーザは現在 50 ブロックというソフトリミットを 15 ブロックオーバーし 5 日間の猶予期間が残っています。アスタリスク * はクォータリミットを越えているユーザを示していることに注意してください。

通常、そのユーザが全く使っていないファイルシステムは、 クォータリミットが付けられているとしても quota(1) コマンドの出力には現われません。`-v` オプションを用いればそのようなファイルシステム、上の例では `/usr/var` 、を表示することができます。

16.12.4. NFS 上の クォータ

クォータは NFS サーバ上のクォータサブシステムにより実行されます。rpc.rquotad(8) デーモンにより、NFS クライアント上の quota(1) コマンドは情報を得ることができ、クライアントマシン上のユーザが自分のクォータの統計を見ることができます。

`/etc/inetd.conf` において以下のように `rpc.rquotad` を有効にしましょう。

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.↺
rquotad
```

そして以下のように inetd を再起動します。

```
# kill -HUP `cat /var/run/inetd.pid`
```

16.13. ディスクパーティションの暗号化

##: Lucky Green.

FreeBSD は無許可のデータアクセスに対する優れたオンライン保護機能を提供します。 ファイルのパーミッションおよび強制的アクセスコントロール (MAC: Mandatory Access Control) (Mandatory Access Control (MAC) を参照) は、コンピュータが動作中で、OS が実行中であるときに、 無許可の第三者がデータにアクセスするのを防ぐことに役立ちます。 しかしながら、攻撃者がコンピュータに物理的にアクセスし、機密データをコピーし分析するためにコンピュータのハードドライブを別のシステムに移動させることができれば、OS によって強化された許可属性は意味をなさなくなります。

攻撃者が電源の落ちたコンピュータや ハードドライブを手にいれる手段にかかわらず、GEOM ベースのディスク暗号化 (gbde: GEOM Based Disk Encryption) は、著しい資源を持ち本気で攻撃を仕掛けるつもりでやってきた攻撃者からさえもコンピュータのファイルシステム上にあるデータを保護することができます。 個々のファイルだけを暗号化する煩わしい方法と異なり、gbde は全ファイルシステムを透過的に暗号化します。平文テキストは決してハードドライブのプラッタに関係しません。

16.13.1. カーネルで gbde を有効にする

1. **root** になる

   gbde の設定をするにはスーパユーザの権限が必要になります。 以下のコマンドを実行して、root になってください。

   ```
   % su -
   Password:
   ```

2. オペレーティングシステムのバージョンを確かめる

   gbde(4) が動作するには FreeBSD 5.0 以降が必要です。以下のコマンドを実行して、オペレーティングシステムのバージョンを確認してください。

   ```
   # uname -r
   5.0-RELEASE
   ```

3. カーネルコンフィギュレーションファイルに gbde(4) 対応を追加する

   お好みのテキストエディタを使用して、 以下の行をカーネルコンフィギュレーションファイルに加えます。

```
options GEOM_BDE
```

FreeBSD カーネルを設定、再コンパイル、インストールします。 この手順は 9章FreeBSD カーネルのコンフィグレーション で説明されています。

新しいカーネルで再起動します。

16.13.2. 暗号化されたハードドライブの準備

以下の例では、システムに新しいハードディスクを追加しようとしています。このシステムは単一の暗号化されたパーティションを保持することになります。 このパーティションは `/private` としてマウントされます。gbde は `/home` および `/var/mail` を暗号化するのにも使用できますが、 より複雑な指示を必要となるのでこの解説の範疇を越えています。

1. 新しいハードドライブを追加する

   「ディスクの追加」で説明されている通りに新しいドライブをシステムに設置します。この例では、新しいハードドライブは `/dev/ad4s1c` パーティションに 加えられたものとします。`/dev/ad0s1*` デバイスは、この例のシステム上に存在する標準的な FreeBSD パーティションを表します。

   ```
   # ls /dev/ad*
   /dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
   /dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
   /dev/ad0s1a     /dev/ad0s1d     /dev/ad4
   ```

2. gbde ロックファイルを保持するディレクトリを作成する

   ```
   # mkdir /etc/gbde
   ```

   gbde ロックファイルには、 暗号化されたパーティションにアクセスするのに必要となる情報が格納されています。ロックファイルにアクセスしない場合、gbde は 膨大な手動による介在なしには (ソフトウェアは対応していません)、暗号化されたパーティションに含まれるデータを解読することはできないでしょう。 それぞれの暗号化されたパーティションは別々のロックファイルを使用します。

3. gbde パーティションを初期化する

   gbde パーティションは使用する前に初期化されなければなりません。 この初期化は一度だけ実行される必要があります。

   ```
   # gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c
   ```

   エディタが開くので、テンプレートをもとにさまざまなオプションを設定してください。UFS1 または UFS2 で使用するには、sector_size を 2048 に設定してください。

```
$FreeBSD: src/sbin/gbde/template.txt,v 1.1 2002/10/20 11:16:13 ↺
phk Exp $
#
# Sector size is the smallest unit of data which can be read or ↺
written.
# Making it too small decreases performance and decreases ↺
available space.
# Making it too large may prevent filesystems from working.  512 ↺
is the
# minimum and always safe.  For UFS, use the fragment size
#
sector_size     =       2048
[...-]
```

gbde(8) はデータを保護するのに使用するパスフレーズを二度尋ます。パスフレーズはそれぞれ同じでなければなりません。データを保護する gbde の能力は、あなたが選択したパスフレーズの品質に完全に依存します。[1]

`gbde init` コマンドは gbde パーティションに対するロックファイルを作成します。この例では `/etc/gbde/ad4s1c` に格納されます。

注意

gbde ロックファイルは、 すべての暗号化されたパーティションの内容とともにバックアップされなければ なりません。 ロックファイルだけを削除している間、 ロックファイルなしでは信念の固い攻撃者が gbde パーティションを解読することを防ぐことができない一方で、 正当な所有者は、gbde(8) およびこの設計者にまったく支持されない膨大な量の作業なしには、 暗号化されたパーティション上のデータにアクセスすることができないでしょう。

4. カーネルに暗号化されたパーティションを接続する

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
```

暗号化されたパーティションを初期化する際に選択したパスフレーズを入力するように求められます。 新しい暗号化デバイスは `/dev` に `/dev/device_name.bde` として現れます。

[1]記憶するのが簡単で、安全なパスフレーズを選択する方法については、 Diceware Passphrase ウェブサイトを参照してください。

```
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

5. 暗号化デバイス上にファイルシステムを作成する

   カーネルに暗号化デバイスが接続されると、 デバイス上にファイルシステムを作成できます。暗号化デバイス上にファイルシステムを作成するには newfs(8) を使用します。従来の UFS1 ファイルシステムで初期化するより、新しい UFS2 ファイルシステムで初期化した方が高速なので、`-O2` オプションとともに newfs(8) を使用することが推奨されています。

> **注記**
>
> FreeBSD 5.1-RELEASE 以降では、`-O2` オプションはデフォルトです。

```
# newfs -U -O2 /dev/ad4s1c.bde
```

> **注記**
>
> newfs(8) は、デバイス名に `*.bde` 拡張子によって認識される、接続された gbde パーティションに対して実行されなければなりません。

6. 暗号化パーティションをマウントする

   暗号化ファイルシステムに対するマウントポイントを作成します。

```
# mkdir /private
```

   暗号化ファイルシステムをマウントします。

```
# mount /dev/ad4s1c.bde /private
```

7. 暗号化ファイルシステムが利用可能か確かめる

   これで暗号化ファイルシステムは df(1) で見ることができ、利用する準備ができました。

```
% df -H
Filesystem          Size    Used   Avail Capacity  Mounted on
/dev/ad0s1a        1037M     72M    883M     8%    /
/devfs              1.0K    1.0K      0B   100%    /dev
/dev/ad0s1f         8.1G     55K    7.5G     0%    /home
/dev/ad0s1e        1037M    1.1M    953M     0%    /tmp
/dev/ad0s1d         6.1G    1.9G    3.7G    35%    /usr
/dev/ad4s1c.bde     150G    4.1K    138G     0%    /private
```

16.13.3. 存在する暗号化ファイルシステムをマウントする

システムを起動する度に、すべての暗号化ファイルシステムは 使用前にカーネルに接続し、 エラーの有無をチェックし、マウントする必要があります。 必要なコマンドは `root` ユーザとして実行されなければなりません。

1. カーネルに gbde パーティションを接続する

   ```
   # gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
   ```

   パーティションの暗号化を初期化する際に選択したパスフレーズを入力するように求められるでしょう。

2. ファイルシステムのエラーをチェックする

   暗号化ファイルシステムを自動的にマウントするために `/etc/fstab` に設定を掲載することはまだできないため、マウントする前に fsck(8) を実行して、ファイルシステムのエラーをチェックしなければなりません。

   ```
   # fsck -p -t ffs /dev/ad4s1c.bde
   ```

3. 暗号化ファイルをマウントする

   ```
   # mount /dev/ad4s1c.bde /private
   ```

   これで暗号化ファイルシステムが利用できるようになりました。

16.13.3.1. 暗号化パーティションを自動的にマウントする

スクリプトを作成して、暗号化パーティションを自動的に接続、 チェック、マウントすることは可能です。しかしながら、安全上の理由によりスクリプトに gbde(8) パスワードを含めるべきではありません。その代わりに、コンソールまたは ssh(1) による接続からパスワードを入力するようなスクリプトが手動で実行されることが推奨されます。

16.13.4. gbde が採用した暗号の保護

gbde(8) は 128bit AES の CBC モードを使用してセクタペイロードを暗号化します。ディスク上のそれぞれのセクタは異なる AES 鍵で暗号化されます。 セクタ鍵がユーザ

が入力したパスフレーズからどのように導き出されるかを含め、 gbde の暗号手法の設計についての詳細は、gbde(4) を参照してください。

16.13.5. 互換性に関する問題

sysinstall(8) は gbde 暗号化デバイスと互換性がありません。sysinstall(8) を実行する前に `*.bde` デバイスはすべてカーネルから切断されなければなりません。そうしないと、sysinstall(8) が初めにデバイスを走査する際にクラッシュしてしまうでしょう。暗号化デバイスを切断するには、以下のコマンドを使用します。

```
# gbde detach /dev/ad4s1c
```

第17章 地域化 (localization) - I18N/L10N の利用と設定

寄稿: Andrey Chernov.
改訂: Michael C. Wu.

17.1. この章では

FreeBSD は、ユーザーおよび貢献者が世界中にいる、非常に分散した プロジェクトです。この章では、英語以外の言語を使うユーザーの実用に 耐えられるようにする FreeBSD の国際化 (internationalization) と 地域化 (localization) 機能について解説します。システムレベルでもアプリケーションレベルでも、国際化の実装には 様々な側面があるので、必要に応じて読者に対してより専門的な文書情報を 示すことにします。

この章では、以下の分野について説明します。

- 近代的なオペレーティングシステムで、異なる言語および ロケールがどのように符号化されているか。
- ログインシェルでロケールを設定するには どうするか。
- コンソールを英語以外の言語用に設定するには どうするか。
- 様々な言語で効率的に X Window System を使うには どうすればよいか。
- 国際化されたアプリケーションを書くための 情報はどこにあるか。

この章を読む前に、以下のことを理解しておく必要があります。

- サードパーティ製アプリケーションのインストール方法 (5章アプリケーションのインストール - packages と ports)。

17.2. 基礎知識

17.2.1. I18N/L10N とは?

開発者たちはしばしば、internationalization を縮めて I18N と表記します。18 は internationalization の最初と最後の間の 文字数です。L10N も同じ命名法を用いて「localization」を縮めたものです。これらを合わせて、I18N/L10N された (すなわち国際化/地域化された) 手法、プロトコル、アプリケーションは、自分達の好みの言語を 使うことを可能にしてくれます。

国際化されたアプリケーションはライブラリとして国際化キット を用いてプログラミングされています。これは開発者が単純なファイルを書いて、表示されるメニューやテキストを各国語に翻訳できるようにしてくれます。 プログラマのみなさんには、 これらの方法を利用することを強く推奨します。

17.2.2. なぜ I18N/L10N を使わなければいけないの?

英語以外の言語を見たり、入力したり、処理したりする場合はいつでも、I18N/L10N が使われます。

17.2.3. I18N で使えるようになる言語には何があるの?

I18N や L10N は FreeBSD に限った話題ではありません。現在では、世界中の主要な言語のほとんどから選ぶことができます。 たとえば中国語、ドイツ語、日本語、韓国語、フランス語、ロシア語、 ベトナム語などがありますが、これらに限定されるわけではありません。

17.3. 地域化の利用

I18N は FreeBSD に特有のものではなく、一つの考え方です。以下の慣習にしたがって FreeBSD を利用するようにしてください。

地域化の設定は言語コード、 国コード、エンコーディングという三つの用語を基本とします。ロケール名はこれらから以下のように構成されます。

```
#####_####.########
```

17.3.1. 言語コードと国コード

FreeBSD (やその他の国際化をサポートした UNIX®-like なシステム) を特定の言語に地域化するには、国と言語を特定するためのコードを知る必要があります (国コードはアプリケーションに指定された言語のどの変種 (variation) を用いれば良いかを教えてくれます)。加えて、ウェブブラウザ、SMTP/POP サーバ、ウェブサーバなどもこれらを元に様々な選択を行います。以下は言語/国コードの例です。

言語/国コード	説明
en_US	英語 (合衆国)
ru_RU	ロシア語 (ロシア)
zh_TW	繁体字中国語 (台湾)

17.3.2. エンコーディング

いくつかの言語では、8-bit やワイド文字、 多バイト文字など ASCII とは異なったエンコード法を用います (multibyte(3) 参照)。 古いアプリケーションはこれらを認識せず、 誤ってコントロール文字として認識してしまいます。 最近のアプリケーションは、大抵 8-bit 文字を認識します。 実装方法にも依りますが、アプリケーションのコンパイル時もしくは configure 時に、ワイド/多バイト文字のサポートを指定する必要があるかも知れません。 ワイド/多バイト文字を入力したり処理したりすることを可能にするために、FreeBSD Ports Collection では各言語向けに異なったプログラムを提供しています。各 FreeBSD Port の国際化文書を参照してください。

特に、正しく configure したり、configure/Makefile/ コンパイラに適切な値を渡すために、アプリケーションの 文書を良く読む必要があります。

次のことを心に留めておいてください。

- 言語固有の、C 言語の char で表現できる シングルバイトの文字セット (multibyte(3) を参照)、たとえば ISO8859-1, ISO8859-15, KOI8-R, CP437。
- ワイド、多バイトのエンコーディング、たとえば EUC, Big5。

現在有効な文字セットのリストに関しては IANA Registry をチェックしてください。

注記

FreeBSD では、X11 互換のロケール符号を用いています。

17.3.3. 国際化アプリケーション

FreeBSD の ports/packages システムでは、 それとひと目でわかるように国際化アプリケーションには名前に `I18N` という文字が含まれています。ただし、それらのアプリケーションが常にあなたの望む言語を サポートしているとは限りません。

17.3.4. ロケールの設定

通常は、ログインシェルで環境変数 `LANG` に ロケール名を設定し export すれば十分です。これは、ユーザーの `~/.login_conf` ファイル、またはユーザーの シェルの初期設定ファイル (`~/.profile` , `~/.bashrc` , `~/.cshrc`) でできます。`LC_CTYPE` や `LC_CTIME` のような ロケールのサブセットを設定する必要はありません。詳細に関しては、各言語向けの FreeBSD 文書を参照してください。

以下の二つの環境変数を設定ファイルで指定する必要があります。

- POSIX® setlocale(3) 関連の関数のための `LANG`

- アプリケーション用の MIME 文字セットのための MM_CHARSET

これにはユーザのシェルの設定、アプリケーション固有の設定、X11 の設定などが含まれます。

17.3.4.1. ロケールを設定する方法

ロケールを設定するには以下で説明するように、二つの方法があります。 一つは推奨される方法で、ログインクラス (login class) において環境変数に割り当てる方法。もう一つはシステムのシェル 初期化ファイル において環境変数の指定を追加する方法です。

17.3.4.1.1. ログインクラスを用いる方法

この方法では、各シェルの初期化ファイルに特定のシェル設定を追加する代わりに、すべてのシェルにおいて一度に必要なロケール名と MIME 文字セットを環境変数に割り当てることができます。ユーザの設定はユーザ自身で行なえますが、管理者の設定にはスーパユーザの権限が必要となります。

17.3.4.1.1.1. ユーザの設定

ユーザのホームディレクトリの .login_conf ファイルを用いて、 両方の変数に Latin-1 エンコーディングを設定する 簡単な例は次の通りです。

```
me:\
 :charset=ISO-8859-1:\
 :lang=de_DE.ISO8859-1:
```

これは、BIG-5 エンコーディングされた繁体字中国語用の環境変数を設定する .login_conf の一例です。 中国語、日本語、韓国語用のロケール変数を 正しく認識しないソフトウェアに対処するため、よりたくさんの変数を設定していることに注目してください。

```
#Users who do not wish to use monetary units or time formats
#of Taiwan can manually change each variable
me:\
 :lang=zh_TW.Big5:\
 :lc_all=zh_TW.Big:\
 :lc_collate=zh_TW.Big5:\
 :lc_ctype=zh_TW.Big5:\
 :lc_messages=zh_TW.Big5:\
 :lc_monetary=zh_TW.Big5:\
 :lc_numeric=zh_TW.Big5:\
 :lc_time=zh_TW.Big5:\
 :charset=big5:\
 :xmodifiers="@im=xcin": #Setting the XIM Input Server
```

詳細に関しては 管理者の設定 と login.conf(5) を参照してください。

17.3.4.1.1.2. 管理者の設定

`/etc/login.conf` において、正しい言語がユーザのクラスに指定されていることを確認してください。`/etc/login.conf` は、このようになります。

```
language_name :accounts_title :\
 :charset=MIME_charset :\
 :lang=locale_name :\
 :tc=default:
```

先ほどの例のように Latin-1 での設定はこのようになります。

```
german:German Users Accounts:\
 :charset=ISO-8859-1:\
 :lang=de_DE.ISO8859-1:\
 :tc=default:
```

ユーザのログインクラスを変更する前に次のコマンドを実行して、`/etc/login.conf` の新たな設定がシステムに見えるようにしてください。

```
# cap_mkdb /etc/login.conf
```

ログインクラスを vipw(8) で変更する

新しいユーザを追加するために `vipw` を用います。そして以下のようなエントリを作成します。

```
user:password:1111:11:language :0:0:User Name:/home/user:/bin/sh
```

ログインクラスを adduser(8) で変更する

新しいユーザを追加するために `adduser` を用います。そして以下の手順を踏みます。

- `/etc/adduser.conf` で `defaultclass = language` と設定します。この場合、他の言語のユーザには `default` クラスを指定することを 忘れないでください。
- もうひとつの方法は、adduser(8) が

  ```
  Enter login class: default []:
  ```

  と聞いてきたときに、毎回言語を指定するやり方です。
- さらに別の方法は、異なる言語を利用するユーザを 追加する際に、以下のようにするやり方です。

  ```
  # adduser -class language
  ```

ログインクラスを pw(8) で変更する

新しいユーザを追加するために pw(8) を 用いる場合、以下の形式で実行します。

```
# pw useradd user_name -L language
```

17.3.4.1.2. シェルの初期化ファイルによる方法

注記

シェルごとに異なった設定が必要なため、 この方法は推奨されません。代わりに ログインクラスを用いる方法を使ってください。

ロケール名と MIME 文字コードを追加するには、 /etc/profile や /etc/csh.login などのシェル初期化ファイル に以下の二つの環境変数を設定します。以下に示す例は、ドイツ語の設定です。

/etc/profile では 次のように設定します。

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

また /etc/csh.login では 次のように設定します。

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

もしくは、上記のやり方を /usr/share/skel/dot.profile (/etc/profile と同形式) や /usr/share/skel/dot.login (/etc/csh.login と同形式) に追加することもできます。

X11 では、$HOME/.xinitrc に 使用しているシェルに合った形式で

```
LANG=de_DE.ISO8859-1; export LANG
```

もしくは、

```
setenv LANG de_DE.ISO8859-1
```

と指定します。

17.3.5. コンソールの設定

C 言語の char で表現できるシングルバイトの文字セット用に、/etc/rc.conf でその言語に対応した適切なコンソールフォントを指定してください。

```
font8x16=#####
font8x14=#####
font8x8=#####
```

ここで ##### は /usr/share/syscons/fonts ディレクトリにあるフォントファイルから .fnt という拡張子を除いたものです。

また、sysinstall (FreeBSD バージョンが 5.2 よりも古い場合は /stand/sysinstall) を使って、C 言語の char で表現できるシングルバイトの文字セット用の正しい キーマップとスクリーンマップを指定するようにしてください。 sysinstall では、Configure を選択し、Console を選択します。もしくは、/etc/rc.conf に以下の行を加えてください。

```
scrnmap=#########
keymap=######
keychange="############### "
```

ここで ######### は /usr/share/syscons/scrnmaps ディレクトリにあるマップファイルから .scm という拡張子を除いたものです。VGA アダプタが疑似グラフィクス領域のフォント文字マトリクスで bit 8 を bit 9 に拡張することに対処するために (例えばスクリーンフォントが bit 8 列を使っている時に文字をその領域から外に移動する場合)、フォントに適切にマップされたスクリーンマップが必要となります。

もし、/etc/rc.conf を以下のように設定して、moused デーモンを有効化している場合は、次の段落に書かれているマウスカーソルに関する情報を確認してください。

```
moused_enable="YES"
```

設定省略時には、syscons(4) ドライバのマウスカーソルは キャラクタセット中の 0xd0-0xd3 の範囲を占めています。そのため、利用している言語がこの範囲のキャラクタセットを使っている場合、 カーソルの占める範囲を重ならないように移動させなければなりません。FreeBSD でこれを回避するには、次の行を /etc/rc.conf に追加してください。

```
mousechar_start=3
```

は /usr/share/syscons/keymaps ディレクトリにあるキーマップファイルから .kbd という拡張子を除いたものです。どのキーマップを使うかよくわからないなら、kbdmap(1) で再起動せずにキーマップを試すことができます。

ファンクションキーの並びはキーマップにより定義されてはいないため、 端末タイプに合わせたファンクションキーを設定するために keychange が必要となります。

また、/etc/ttys の中のすべての ttyv* において、正しいコンソール端末タイプを設定するようにしてください。現在の定義済の値は以下の通りです。

文字セット	端末タイプ
ISO8859-1 もしくは ISO8859-15	`cons25l1`
ISO8859-2	`cons25l2`
ISO8859-7	`cons25l7`
KOI8-R	`cons25r`
KOI8-U	`cons25u`
CP437 (VGA のデフォルト)	`cons25`
US-ASCII	`cons25w`

ワイド/多バイト文字の言語については、 `/usr/ports/language` 内の適切な FreeBSD port を利用してください。いくつかの ports はシステムからシリアルの vtty のように見えるようにコンソールとして振る舞います。したがって、X11 と疑似シリアルコンソール用に充分な vtty を確保しておかなければなりません。コンソールで他の言語を使うためのアプリケーションのリストの 一部です。

言語	ports の位置
繁体字中国語 (BIG-5)	chinese/big5con
日本語	japanese/kon2-16dot または japanese/mule-freewnn
韓国語	korean/han

17.3.6. X11 の設定

X11 は FreeBSD プロジェクトの一部ではありませんが、FreeBSD ユーザのための情報を記しておきます。詳細に関しては、Xorg ウェブサイトや、あなたの使っている X11 サーバのサイトを参照してください。

`~/.Xresources` を使うことで、 アプリケーション固有の国際化の設定 (フォント、メニューなど) を追加することができます。

17.3.6.1. フォントの表示

Xorg サーバ (x11-servers/xorg-server) か XFree86™ サーバ (x11-servers/XFree86-4-Server) をインストールし、言語の TrueType® フォントをインストールします。ロケールを正しく設定すれば、選んだ言語がメニューなどに表示されるはずです。

17.3.6.2. 非英語の文字を入力する

X11 Input Method (XIM) プロトコルはすべての X11 クライアント用の新たな標準規格です。すべての X11 アプリケーションは XIM 入力サーバからの入力を受け取る XIM クライアントとして書かれるべきです。様々な言語用の XIM サーバが存在します。

17.3.7. プリンタの設定

プリンタにはいくつかの C 言語の char で表現できる シングルバイトの文字セットがハードウェアに組み込まれています。 ワイド/多バイトの文字セットでは特殊な設定が必要であり、 apsfilter を使うことをお勧めします。 言語固有のコンバータを用いて、PostScript® か PDF フォーマット に文書をコンバートする場合もあるでしょう。

17.3.8. カーネルとファイルシステム

FreeBSD の高速ファイルシステム (FFS) は 8-bit 透過であり、C 言語の char で表現できるいかなる文字セットも使うことが できます (multibyte(3) を参照)。しかし、ファイルシステム中には文字セットの名前は記録されていません。したがって、これは単なる 8-bit であり、エンコーディングに関しては何の情報もないのです。公式には、FFS はまだいかなるワイド/マルチバイトの文字セットもサポートしていません。しかし FFS でそのようなサポートを行うためのパッチが、多くのワイド/マルチバイトの文字セットに存在します。 それらは単に一時的で汎用性のない解決策であり、 わたしたちはそれらをソースツリーに含めないことを決めています。これらのパッチに関しては、各言語のウェブサイトを参照してください。

FreeBSD の MS-DOS® ファイルシステムでは、 MS-DOS®, Unicode 文字セット、FreeBSD ファイルシステムの 文字セットの間で変換を行うことが可能です。詳細は mount_msdos(8) を参照してください。

17.4. 国際化に対応したプログラムのコンパイル

FreeBSD ports の多くはすでに国際化されています。いくつかには port の名前に -I18N と付いています。これらはもちろんのこと、他のプログラムも国際化への対応を組み込んだものがあり、コンパイルに際して特別な注意を払う必要はありません。

しかし、MySQL のようなアプリケーションでは、 特定の文字セットを使うように `Makefile` を設定する必要があります。これは大抵 `Makefile` の中で 対処されているか、ソース中の configure に値を渡すことで対応しています。

17.5. FreeBSD を特定の言語に地域化する

17.5.1. ロシア語 (KOI8-R エンコーディング)

##: Andrey Chernov.

KOI8-R エンコーディングの詳細については、 KOI8-R References (Russian Net Character Set) を参照してください。

17.5.1.1. ロケールの設定

以下の行を `~/.login_conf` に追加してください。

```
me:My Account:\
 :charset=KOI8-R:\
 :lang=ru_RU.KOI8-R:
```

ロケール を 設定する際の例については、この章の前の方を参照してください。

17.5.1.2. コンソールの設定

- `/etc/rc.conf` ファイルに次の行を追加してください。

```
mousechar_start=3
```

- また、`/etc/rc.conf` で以下の設定を使ってください。

```
keymap="ru.koi8-r"
scrnmap="koi8-r2cp866"
font8x16="cp866b-8x16"
font8x14="cp866-8x14"
font8x8="cp866-8x8"
```

- `/etc/ttys` の各 `ttyv*` エントリにおいて、端末タイプとして `cons25r` を指定してください。

コンソールを設定する際の例については、この章の前の方を参照してください。

17.5.1.3. プリンタの設定

ロシア語用の文字を搭載したプリンタはほとんど ハードウェアコードページ CP866 を使っているため、KOI8-R を CP866 に変換する専用の出力フィルタが必要となります。このフィルタはデフォルトで `/usr/libexec/lpr/ru/koi2alt` に インストールされています。ロシア語用のプリンタの `/etc/printcap` エントリは以下のようになります。

```
lp|Russian local line printer:\
 :sh:of=/usr/libexec/lpr/ru/koi2alt:\
 :lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

記述の詳細については printcap(5) を参照してください。

17.5.1.4. MS-DOS® ファイルシステムとロシア語ファイル名

以下の fstab(5) エントリの例は、マウントされた MS-DOS® ファイルシステムにおいてロシア語ファイル名を 使えるようにします。

```
/dev/ad0s2      /dos/c  msdos   rw,-Wkoi2dos,-Lru_RU.KOI8-R 0 0
```

`-L` オプションは利用するロケール名を選択し、`-W` オプションは文字変換表を設定します。`-W` オプションを使う時は、変換表が `/usr/libdata/msdosfs` にあるので、`/`

usr を MS-DOS® パーティションより前に マウントするようにしてください。詳しくは、mount_msdos(8) のマニュアルを参照してください。

17.5.1.5. X11 の設定

1. まず X 以外のロケールの設定を行ってください。

2. Xorg を使っているなら、x11-fonts/xorg-fonts-cyrillic パッケージをインストールしてください。

   /etc/X11/xorg.conf ファイルの "Files" セクションをチェックしてください。既存の FontPath エントリの前に以下の行を追加しなければなりません。

   ```
   FontPath   "/usr/X11R6/lib/X11/fonts/cyrillic/misc"
   FontPath   "/usr/X11R6/lib/X11/fonts/cyrillic/75dpi"
   FontPath   "/usr/X11R6/lib/X11/fonts/cyrillic/100dpi"
   ```

   もし高解像度のビデオモードを使っている場合には、75 dpi と 100 dpi の行を入れ替えてください。

3. ロシア語のキーボードを使えるようにするには、以下の行を xorg.conf ファイルの "Keyboard" セクションに追加します。

   ```
   Option "XkbLayout"   "us,ru"
   Option "XkbOptions"  "grp:toggle"
   ```

   また、XkbDisable が無効 (コメントアウト) になっていることを確認してください。

   grp:caps_toggle については、ロシア語/ラテン文字の切り替えは CapsLock で行います。 従来の CapsLock の機能は Shift+CapsLock で使うことができます (ラテン文字モードの時のみ)。 grp:toggle については、ロシア語/ラテン文字の切り替えは Right Alt で行います。 Xorg では、理由は不明ですが grp:caps_toggle は動作しません。

   キーボードに「Windows®」キーがあり、ロシア語モードでそのキーにいくつかの非英字キーが 割り当てられているようなら、xorg.conf ファイルに以下の行を追加してください。

   ```
   Option "XkbVariant" ",winkeys"
   ```

注記

ロシア語の XKB キーボードは、地域化されていないアプリケーションではうまく動かないかも知れません。

注記

地域化がされたアプリケーションは少なくともプログラムの最初の方で `XtSetLanguageProc (NULL, NULL, NULL);` を呼び出すべきです。X11 アプリケーションを地域化する方法については、 KOI8-R for X Window を参照してください。

17.5.2. 台湾向けの繁体字中国語への地域化

FreeBSD-Taiwan プロジェクトは、多くの 中国語 ports を利用した、 FreeBSD を中国語化するための手引き `http://netlab.cse.yzu.edu.tw/~statue/freebsd/zh-tut/` を提供しています。`FreeBSD Chinese HOWTO` の現在の編集者は Shen Chuan-Hsing `<statue@freebsd.sinica.edu.tw >` です。

Chuan-Hsing Shen `<statue@freebsd.sinica.edu.tw >` は FreeBSD-Taiwan の `zh-L10N-tut` を使って Chinese FreeBSD Collection (CFC) を作成しました。パッケージとスクリプトは `ftp://freebsd.csie.nctu.edu.tw/pub/taiwan/CFC/` から入手できます。

17.5.3. ドイツ語への地域化 (すべての ISO 8859-1 言語向け)

Slaven Rezic `<eserte@cs.tu-berlin.de >` は FreeBSD マシン上でウムラウトを使うためのチュートリアルを書きました。 チュートリアルはドイツ語で書かれており、`http://www.de.FreeBSD.org/de/umlaute/` から入手できます。

17.5.4. 日本語/韓国語への地域化

日本語に関しては `http://www.jp.FreeBSD.org/` を、韓国語に関しては `http://www.kr.FreeBSD.org/` を参照してください。

17.5.5. 英語以外の FreeBSD ドキュメント

FreeBSD の一部を他の言語に翻訳してくれている人たちがいます。これらは メインサイトのリンクを辿るか `/usr/share/doc` から入手できます。

第18章 FreeBSD のアップデートとアップグレード

再構成、再編成および改訂: Jim Mock.
原作: Jordan Hubbard, Poul-Henning Kamp, John Polstra 、Nik Clayton.

18.1. この章では

あるリリースから次のリリースまでの期間にも、 FreeBSD の開発は休みなく続けられています。 最新の開発ツリーと同期することを好む人もいますし、 公式のリリース版を好んで使う方もいます。 しかしながら、公式のリリースといえども、 セキュリティや他の重要な修正のため、時にはアップデートを行う必要があります。 使用しているバージョンに関わらず、FreeBSD は、 手元のシステムを最新の開発ツリーと同期するために必要なツールをすべて用意しています。 そして、これらのツールは、FreeBSD のバージョンをアップグレードするために使えます。この章では、開発ブランチを追いかける方法、および、FreeBSD システムをアップデートする基本的なツールについて解説しています。

この章を読んで分かるのは:

- freebsd-update, Subversion もしくは CTM を使った FreeBSD システムの更新方法
- インストールされているシステムと、変更が行われていない状態との比較方法。
- Subversion またはドキュメント用の ports を使って、インストールされているドキュメントを最新のものにアップデートする方法。
- 2 つの開発ブランチ、FreeBSD-STABLE と FreeBSD-CURRENT の違い
- ベースシステム全体を再構築しインストールする方法

この章を読む前に、以下の準備をしましょう。

- ネットワーク接続の適切な設定 (22章高度なネットワーク)
- サードパーティ製のソフトウェアのインストール方法の習得 (5章アプリケーションのインストール - packages と ports)

注記

この章を通じて、 FreeBSD のソースコードをダウンロードしたりアップデートするのに svn が用いられます。このコマンドを使うた

めには、devel/subversion port または package をインストールしておく必要があります。

18.2. FreeBSD Update

##: Tom Rhodes.
############: Colin Percival.

システム管理における重要な側面に、すみやかにセキュリティパッチを適用し、オペレーティングシステムを新しいリリースにアップグレードすることがあります。FreeBSD には、これらの処理を行うために `freebsd-update` と呼ばれるユーティリティが用意されています。

このユーティリティを用いると、FreeBSD のセキュリティおよび eratta アップデートをバイナリによって行うことができます。 手動でパッチもしくは新しいカーネルをコンパイルし、 インストールする必要はありません。 バイナリアップデートは、 セキュリティチームがサポートしているすべてのアーキテクチャとリリースで利用できます。`http://www.FreeBSD.org/ja/security/` には、サポートが行われているリリースや保守終了予定日の一覧があります。

このユーティリティは、マイナーリリースであったり、 他のリリースブランチへのアップグレードにも対応しています。 新しいリリースにアップデートする前に、 アップデートしようとしているリリースのアナウンスに目を通し、重要な情報がないかどうかを確認してください。リリースのアナウンスは `http://www.FreeBSD.org/ja/releases/` で確認できます。

注記

もし `crontab` の中に freebsd-update(8) の機能が含まれていたら、オペレーティングシステムのアップグレード作業を終えるまでは無効にしてください。

この節では、`freebsd-update` で使われる設定ファイルの説明、セキュリティパッチの適応方法のデモンストレーション、オペレーティングシステムをアップグレードする際に考慮すべきことについて説明します。

18.2.1. 設定ファイル

`freebsd-update` のデフォルトの設定ファイルは、そのままでも用いることができます。`/etc/freebsd-update.conf` の設定をデフォルトからきめ細かく調整して、

アップデートプロセスを制御するユーザもいます。このファイルのコメントにおいて利用可能なオプションが説明されていますが、以下の項目については補足が必要でしょう。

```
# Components of the base system which should be kept updated.
Components world kernel
```

このパラメータは、FreeBSD のどの部分を最新に維持するかを設定します。デフォルトでは、ベースシステム全体、そしてカーネルをアップデートします。`src/base` や `src/sys` のように、個々の項目を指定することもできます。 この部分についてはデフォルトのままにしておき、アップデートする項目をユーザがリストに加える形にするのがベストでしょう。ソースコードとバイナリが同期していないと、 長い年月の間に悲惨な結果がもたらされる可能性があります。

```
# Paths which start with anything matching an entry in an IgnorePaths
# statement will be ignored.
IgnorePaths /boot/kernel/linker.hints
```

`/bin` や `/sbin` 等の特定のディレクトリをアップデートで変更しないように、これらのパスを追加してください。このオプションは、ローカルの変更点を `freebsd-update` が上書きすることを防ぐ目的にも利用できます。

```
# Paths which start with anything matching an entry in an ↺
UpdateIfUnmodified
# statement will only be updated if the contents of the file have ↺
not been
# modified by the user (unless changes are merged; see below).
UpdateIfUnmodified /etc/ /var/ /root/ /.cshrc /.profile
```

このオプションは、指定したディレクトリにある設定ファイルを、 ローカルで変更されていない場合のみアップデートします。ユーザがこれらのファイルを変更していると、これらのファイルの自動アップデートは妨げられます。 他に、`KeepModifiedMetadata` という別のオプションが存在します。 このオプションは、`freebsd-update` がマージ中に変更点を保存するようにします。

```
# When upgrading to a new FreeBSD release, files which match ↺
MergeChanges
# will have any local changes merged into the version from the new ↺
release.
MergeChanges /etc/ /var/named/etc/ /boot/device.hints
```

`freebsd-update` がマージすべきファイルが存在するディレクトリの一覧です。 ファイルのマージのプロセスは、mergemaster(8) と同様 diff(1) パッチの連続ですが、選択肢は少なく、マージを承認するか、エディタを起動するか、 `freebsd-update` を中断するかどうかを選んでください。もし、心配な点があれば、`/etc` をバックアップしてからマージを承認してください。 `mergemaster` の詳細な情報については、「設定ファイルの同期」で確認してください。

```
# Directory in which to store downloaded updates and temporary
# files used by FreeBSD Update.
```

```
# WorkDir /var/db/freebsd-update
```

ここではすべてのパッチや一次ファイルを置くディレクトリを指定しています。 バージョンをアップグレードするのであれば、 この場所には少なくともギガバイトの空き容量が必要です。

```
# When upgrading between releases, should the list of Components be
# read strictly (StrictComponents yes) or merely as a list of ↺
components
# which *might* be installed of which FreeBSD Update should figure ↺
out
# which actually are installed and upgrade those (StrictComponents ↺
no)?
# StrictComponents no
```

このオプションを yes に設定すると、 freebsd-update は Components のリストが完全に正しいと判断し、 このリスト以外の変更点については取り扱いません。 freebsd-update は、効率的に Components リストに属するファイルをアップデートします。

18.2.2. セキュリティパッチの適用

FreeBSD のセキュリティパッチを適用する過程は簡単になり、 管理者は freebsd-update を使って、システムを完全にパッチがあたった状態に保つ事ができるようになりました。FreeBSD セキュリティ勧告の詳細については、FreeBSD セキュリティ勧告 の節で説明されています。

以下のコマンドを実行すると、FreeBSD のセキュリティパッチがダウンロードされ、インストールされます。最初のコマンドは、未対応のパッチがあるかどうかを調べます。もし未対応のパッチがある場合には、パッチが当てられた際に変更されるファイルのリストが作成されます。2 番目のコマンドはパッチを適用します。

```
# freebsd-update fetch
# freebsd-update install
```

アップデートによってカーネルにパッチが当たった場合には、 パッチが当たったカーネルで起動するように、システムを再起動する必要があります。もし、実行中のバイナリにパッチが当てられた場合には、 パッチの当てられたバージョンのバイナリが使われるように、影響のあるアプリケーションを再起動する必要があります。

毎日一度アップデートがないかどうかを自動的に確認するように設定するには、 以下のエントリを /etc/crobntab に追加してください。

```
@daily                                          root    freebsd-update cron
```

パッチが存在すると、 自動的にダウンロードされますが、適用はされません。 root宛てにメールで、ダウンロードされたパッチを確認し、freebsd-update install とともに手動でインストールする必要のあることが通知されます。

うまく行かなかった場合には、freebsd-update を以下のように実行すると、最後の変更までロールバックできます。

```
# freebsd-update rollback
Uninstalling updates... done.
```

カーネルまたはカーネルモジュールがアップデートされた場合には、 完了後にもう一度システムを再起動して、影響のあったバイナリを再起動してください。

freebsd-update ユーティリティが自動的にアップデートするカーネルは GENERIC のみです。カスタムカーネルがインストールされている場合には、 freebsd-update がインストールした後、 カーネルを再構築し、もう一度インストールする必要があります。しかしながら、GENERIC カーネルが /boot/GENERIC に存在する場合には、現在のシステムで実行されているカーネルでなくとも、freebsd-update によりアップデートされます。

注記

GENERIC カーネルを、常に /boot/GENERIC に置いておいてください。さまざまな問題を解決する際や、バージョンをアップグレードする際に助けとなります。GENERIC カーネルを用意する方法については、「FreeBSD 9.X 以降のシステムにおけるカスタムカーネル」または「FreeBSD 8.X におけるカスタムカーネル」を参照してください。

/etc/freebsd-update.conf のデフォルトの設定を変更しない限り、freebsd-update は、他の更新と共にカーネルソースをアップデートします。新しいカスタムカーネルの再構築と再インストールは、通常通り行うことができます。

freebsd-update は、常にカーネルをアップデートするとは限りません。freebsd-update install によってカーネルソースが変更されなかった場合には、 カスタムカーネルを再構築する必要はありません。しかしながら freebsd-update は、/usr/src/sys/conf/newvers.sh を常にアップデートします。これは、現在のシステムのパッチレベルを uname -r が -p で表示する時にこのファイルが参照されます。そのため、何も変更されていない場合でも、カスタムカーネルを再構築することにより、 uname がシステムの正確なパッチレベルを報告するようになります。 各システムにインストールされているアップデートをすばやく把握できるようになるので、特に複数のシステムを管理するときに助けとなります。

18.2.3. メジャーおよびマイナーバージョンのアップグレードを行う

FreeBSD のマイナーバージョン間のアップグレード、 たとえば、FreeBSD 9.0 から FreeBSD 9.1 へのアップグレードは、マイナーバージョン アップグレードと呼ばれます。メジャーバージョン アップグレードは、FreeBSD 9.X から FreeBSD 10.X へのアップグレードといった、 FreeBSD のメジャーバージョンが変わるようなアップグレードのことです。 どちらのアップグレードもリリース番号のターゲットを指定する事で、 `freebsd-update` によって行う事ができます。

注記

カスタムカーネルを使っているシステムでは、アップグレードを行う前に `GENERIC` カーネルが、`/boot/GENERIC` に置かれている事を確認してください。`GENERIC` カーネルを用意する方法については、「FreeBSD 9.X 以降のシステムにおけるカスタムカーネル」または「FreeBSD 8.X におけるカスタムカーネル」を参照してください。

以下のコマンドを実行すると、FreeBSD 9.0 のシステムを FreeBSD 9.1 にアップグレードします。

```
# freebsd-update -r 9.1-RELEASE upgrade
```

コマンドを実行すると、`freebsd-update` は設定ファイルと現在のシステムを評価し、アップデートするために必要な情報を収集します。 画面には、どのコンポーネントが認識され、どのコンポーネントが認識されていないといったリストが表示されます。たとえば以下のように表示されます。

```
Looking up update.FreeBSD.org mirrors... 1 mirrors found.
Fetching metadata signature for 9.0-RELEASE from update1.FreeBSD.↺
org... done.
Fetching metadata index... done.
Inspecting system... done.

The following components of FreeBSD seem to be installed:
kernel/smp src/base src/bin src/contrib src/crypto src/etc src/games
src/gnu src/include src/krb5 src/lib src/libexec src/release src/
rescue
src/sbin src/secure src/share src/sys src/tools src/ubin src/usbin
world/base world/info world/lib32 world/manpages

The following components of FreeBSD do not seem to be installed:
kernel/generic world/catpages world/dict world/doc world/games
world/proflibs
```

```
Does this look reasonable (y/n)? y
```

ここで、freebsd-update はアップグレードに必要なすべてのファイルをダウンロードします。何をインストールし、どのように進むかといった質問をされることもあります。

カスタムカーネルを使っていると、上記のステップで以下のような警告が表示されます。

```
WARNING: This system is running a "MYKERNEL" kernel, which is not a
kernel configuration distributed as part of FreeBSD 9.0-RELEASE.
This kernel will not be updated: you MUST update the kernel manually
before running "/usr/sbin/freebsd-update install"
```

この時点ではこの警告を無視してもかまいません。アップデートされた GENERIC カーネルは、アップグレードプロセスの途中で利用されます。

すべてのパッチがローカルシステムへダウンロードされたら、次にパッチが適用されます。このプロセスには時間がかかります。 この時間はコンピュータの性能とワークロードに依存します。 その後、設定ファイルがマージされます。 このプロセスでは、ユーザはファイルをマージするか、 画面上にエディタを立ち上げて手動でマージするかを尋ねられます。 プロセスが進むごとに、成功したマージのすべての結果の情報がユーザに示されます。マージに失敗したり、無視した場合には、プロセスが中断します。ユーザによっては /etc のバックアップを取り、master.passwd や group のような重要なファイルを後で手動でマージする方もいます。

注記

すべてのパッチは別のディレクトリでマージされており、 まだ、システムには反映されていません。 すべてのパッチが正しく適用され、すべての設定ファイルがマージされてプロセスがスムーズに進んだら、ユーザは以下のコマンドを用いて、変更点をディスクに反映してください。

```
# freebsd-update install
```

パッチは最初にカーネルとカーネルモジュールに対して当てられます。システムがカスタムカーネルを実行している場合には、nextboot(8) を使って次回の再起動時のカーネルを、アップデートされた /boot/GENERIC に設定してください。

```
# nextboot -k GENERIC
```

警告

GENERIC カーネルで再起動する前に、カーネルにシステムが適切に起動するために必要なすべてのドライバが含まれていること、もしアップデートしているコンピュータがリモートでアクセスしているのであれば、 ネットワーク接続に必要なすべてのドライバも含まれていることを確認してください。特に、これまで実行しているカスタムカーネルが、 カーネルモジュールとして提供されているビルドインの機能を含んでいるのであれば、 これらのモジュールを一時的に /boot/loader.conf の機能を用いて、GENERIC に読み込んでください。 アップグレードプロセスが終わるまでは、重要ではないサービスを無効にするとともに、必要のないディスクやネットワークのマウントなども避けることが推奨されています。

アップデートされたカーネルでコンピュータを再起動してください。

```
# shutdown -r now
```

システムがオンラインに戻ったら、以下のコマンドを使って freebsd-update を再び実行してください。 アップデートプロセスの状態は保存されているので、 freebsd-update を実行すると、最初からではなく、次のステップに進み、古い共有ライブラリとオブジェクトファイルを削除します。

```
# freebsd-update install
```

注記

使用しているライブラリのバージョン番号の付けられ方によって、3 つのインストールフェーズが 2 つになる場合もあります。

アップグレードはこれで終了です。 もしメジャーアップグレードを行った場合には、 「メジャーバージョンアップグレード後の ports の再構築」で説明されているようにすべての ports および package を再構築してください。

18.2.3.1. FreeBSD 9.X 以降のシステムにおけるカスタムカーネル

freebsd-update を使う前に、GENERIC カーネルが /boot/GENERIC に置かれていることを確認してください。ただ一度だけカスタムカーネルを構築したのであれば、/

boot/kernel.old は GENERIC カーネルそのものです。このディレクトリの名前を /boot/kernel へと変更してください。

もし、2 回以上カスタムカーネルを構築した後であったり、カスタムカーネルを構築した回数がわからなければ、現在のオペレーティングシステムのバージョンの GENERIC カーネルを入手してください。 コンピュータへの物理的なアクセスが可能であれば、 インストールメディアから GENERIC カーネルをインストールできます。

```
# mount /cdrom
# cd /cdrom/usr/freebsd-dist
# tar -C/ -xvf kernel.txz boot/kernel/kernel
```

別な方法としては、GENERIC カーネルをソースから再構築して、インストールしてください。

```
# cd /usr/src
# make kernel __MAKE_CONF=/dev/null SRCCONF=/dev/null
```

freebsd-update がこのカーネルを GENERIC カーネルとして認識するために、GENERIC コンフィグレーションファイルは、とにかく変更してはいけません。また、特別なオプションを指定しないで構築してください。

freebsd-update は、 /boot/GENERIC が存在する事だけを必要とするので、GENERIC カーネルで再起動する必要はありません。

18.2.3.2. FreeBSD 8.X におけるカスタムカーネル

FreeBSD 8.X システムでは、GENERIC カーネルを入手する方法や構築の方法が少し異なります。

コンピュータへの物理的なアクセスが可能であれば、 以下のコマンドを実行することで、インストールメディアから GENERIC カーネルをインストールできます。

```
# mount /cdrom
# cd /cdrom/X.Y-RELEASE /kernels
# ./install.sh GENERIC
```

ここで *X.Y-RELEASE* をリリース番号に置き換えてください。GENERIC カーネルは、デフォルトで /boot/GENERIC にインストールされます。

または、GENERIC カーネルをソースから再構築してください。

```
# cd /usr/src
# env DESTDIR=/boot/GENERIC make kernel __MAKE_CONF=/
dev/null SRCCONF=/dev/null
# mv /boot/GENERIC/boot/kernel/* /boot/GENERIC
# rm -rf /boot/GENERIC/boot
```

`freebsd-update` が、このカーネルを `GENERIC` カーネルとして扱うように、`GENERIC` コンフィグレーションファイルは、とにかく変更してはいけません。また、特別なオプションを指定しないで構築してください。

`GENERIC` カーネルで再起動する必要はありません。

18.2.3.3. メジャーバージョンアップグレード後の ports の再構築

一般的に、マイナーバージョンアップグレードの後では、 インストールされているアプリケーションは、問題なく動作するでしょう。メジャーバージョンが異なるとアプリケーションバイナリーインタフェース (ABI) が異なるため、サードパーティ製のアプリケーションの多くは動作しなくなるでしょう。メジャーバージョンアップグレードを行った後では、インストールされているすべての packages, ports を ports-mgmt/portmaster のようなユーティリティを使ってアップグレードする必要があります。 以下のコマンドを用いて、 インストールされているすべてのアプリケーションを再構築できます。

```
# portmaster -af
```

このコマンドを実行すると、 設定を変更するオプションを持つアプリケーションは、 設定変更のスクリーンを表示し、ユーザからの指示待ちの状態で停止します。この振る舞いをやめ、デフォルトのオプションを使用するには、上記のコマンドに `-G` を含めてください。

ソフトウェアのアップグレードが終わったら、最後にもう一度 `freebsd-update` を実行して、すべてのアップグレードプロセスのやり残し作業を行い、アップグレードのプロセスを完了してください。

```
# freebsd-update install
```

`GENERIC` カーネルを一時的に読み込んでいたのであれば、9章FreeBSD カーネルのコンフィグレーション に書かれている手順に従って、新しいカスタムを構築し、インストールしてください。

コンピュータを再起動し、新しい FreeBSD を立ち上げてください。これでアップグレードのプロセスは完了です。

18.2.4. システムの状態の比較

`freebsd-update` を用いて、インストールされている FreeBSD の状態と、正しく動作することが分かっている状態とを比較できます。 このコマンドは、現在のシステムのユーティリティ、ライブラリ、設定ファイルを評価するので、組み込みの侵入検知システム (IDS) として使うことができます。

警告

このコマンドは、security/snort のような本当の IDS の置き換えになるものではありません。 `freebsd-update` はデータを

> ディスクに保存するので、 不正な変更が行われる可能性があります。`kern.securelevel` と、`freebsd-update` のデータを使用しないときに、 読み取りのみの許可属性に設定されているファイルシステムに置くことで、 不正な変更の可能性を低くできますが、 よりよい解決方法は、 DVD または安全に保存されている外部 USB ディスクのような安全なディスクとシステムを比較することです。組み込まれているユーティリティを用いた、別の方法による IDS 機能については、FreeBSD バイナリによる検出 の節をご覧ください。

比較を行うには、結果の出力先のファイル名を指定してください。

```
# freebsd-update IDS >> outfile.ids
```

システムは検査され、リリースファイルの SHA256 ハッシュ値と現在インストールされているファイルのハッシュ値がファイルの一覧と共に、指定した出力先のファイルに送られます。

これらの行は極めて長いのですが、出力形式は簡単にすぐに解析できます。 たとえば、これらのリリースで異なっているすべてのファイルを知りたいのであれば、以下のコマンドを実行してください。

```
# cat outfile.ids | awk '{ print $1 }' | more
/etc/master.passwd
/etc/motd
/etc/passwd
/etc/pf.conf
```

上の表示例では出力は切り捨てられており、 実際にはもっと多くのファイルが存在します。 これらのファイルには、運用中に変更されるファイルがあります。 たとえば、`/etc/passwd` はユーザがシステムに追加されると変更されます。 また、カーネルモジュールは、`freebsd-update` によりアップデートされるため、変更されます。このような特別なファイルやディレクトリを除外するには、それらを `/etc/freebsd-update.conf` の `IDSIgnorePaths` オプションに追加してください。

18.3. ドキュメントのアップデート

ドキュメントは、FreeBSD オペレーティングシステムの必須要素です。FreeBSD ドキュメントの最新バージョンは、FreeBSD ウェブサイト (http://www.freebsd.org/doc/) から入手できますが、FreeBSD ウェブサイト、ハンドブック、FAQ および文書の最新版をローカルに用意しておくと便利です。

この章では、ソースまたは Ports Collection を使って、ローカルの FreeBSD ドキュメントを最新に保つ方法を説明します。

ドキュメントを編集したり、ドキュメントの誤りを報告する方法については、新しい貢献者のための FreeBSD ドキュメンテーションプロジェクト入門 (http://www.freebsd.org/doc/en_US.ISO8859-1/books/fdp-primer/) をご覧ください。

18.3.1. ソースから FreeBSD ドキュメントをインストールする

ソースから FreeBSD ドキュメントを構築するのに必要なツールは、FreeBSD のベースシステムには含まれてはいません。 svn などの必要なツールは、FreeBSD ドキュメンテーションプロジェクトが開発している textproc/docproj package または port からインストールできます。

インストールしたら、svn を使って、ドキュメントのソースをダウンロードしてください。以下の例で、*https://svn0.us-west.FreeBSD.org* を「Subversion ミラーサイト」の中からもっとも近いミラーのアドレスに変更してください。

```
# svn checkout https://svn0.us-west.FreeBSD.org /doc/
head /usr/doc
```

最初にドキュメントのソースをダウンロードするには少し時間がかかります。 ダウンロードが終わるまでお待ちください。

ダウンロードしたドキュメントのソースをアップデートするには、以下のコマンドを実行してください。

```
# svn update /usr/doc
```

最新のドキュメントのソースのスナップショットを /usr/doc に用意できたら、インストールされているドキュメントをアップデートする準備はすべて整いました。

利用可能なすべての言語のドキュメントをアップデートするには、 以下のように入力してください。

```
# cd /usr/doc
# make install clean
```

もし、ある特定の言語のみをアップデートしたいのであれば、/usr/doc の下にある各言語のサブディレクトリで make を実行してください。

```
# cd /usr/doc/en_US.ISO8859-1
# make install clean
```

ドキュメントをアップデートする別の方法は、/usr/doc または各言語のサブディレクトリで以下のコマンドを実行してください。

```
# make update
```

FORMATS を設定して、以下のようにインストールする出力形式を指定できます。

```
# cd /usr/doc
# make FORMATS='html html-split' install clean
```

ドキュメンテーションの一部のアップデートを簡単にするオプションや、 特定の翻訳のビルドを行うためのオプションが用意されています。 これらのオプションは、システム全般のオプションである /etc/make.conf や、make に与えるコマンドラインオプションで設定できます。

オプションには以下のようなものがあります。

DOC_LANG
: ビルドおよびインストールの言語およびエンコーディングの一覧。 たとえば、英語のドキュメントを指定するには en_US.ISO8859-1 を設定します。

FORMATS
: ビルドを行うフォーマット、または出力フォーマットの一覧。 現在は html, html-split, txt, ps そして pdf に対応しています。

DOCDIR
: ドキュメントをインストールする場所。デフォルトは /usr/share/doc です。

FreeBSD のシステム全般のオプションに関連するもっと多くの make 変数については、make.conf(5) をご覧ください。

18.3.2. ports を用いたドキュメンテーションのアップデート

#########:Marc Fonvieille.

これまでのセクションでは、ソースコードを用いた FreeBSD ドキュメントのアップデート方法について説明してきました。 この節では、インストールされている FreeBSD のドキュメントをアップデートするもう一つの方法である、Ports Collection を用いた方法について説明し、以下について説明します。

- 構築済のドキュメントの packages をインストールする方法。ローカルでの構築作業やドキュメンテーションツールチェインをインストールする必要はありません。
- ports フレームワークを使ったドキュメントのソースの構築方法。 チェックアウトおよび構築作業が簡単になります。

FreeBSD のドキュメントをアップデートするこれらの方法は、ドキュメンテーションエンジニアリングチーム <doceng@FreeBSD.org > が毎月アップデートしている ドキュメンテーション ports および packages によりサポートされています。 これらの ports は、FreeBSD Ports Collection の docs カテゴリ (http://www.freshports.org/docs/) にまとめられています。

ドキュメンテーション ports の構成は以下の通りです。

- misc/freebsd-doc-en package または portは、すべての英語文書をインストールします。
- misc/freebsd-doc-all メタ package もしくは port は、すべての利用可能な言語のすべてのドキュメントを構築します。
- 各言語のために package または port が用意されています。たとえば、misc/freebsd-doc-hu はハンガリー語のドキュメンテーション port です。

バイナリ package を使うと、インストールする言語に用意されているすべての形式の FreeBSD ドキュメントがインストールされます。たとえば、以下のコマンドを実行すると、ハンガリー語のドキュメントの最新 package がインストールされます。

```
# pkg install hu-freebsd-doc
```

注記

ドキュメントの package は、対応する port 名とは異なり、*lang*-freebsd-doc の形式で名前がつけられています。ここで、*lang* は言語コードの短縮形です。ハンガリー語の場合は hu、簡体字の場合には zh_cn です。

ドキュメントのフォーマットを指定する場合には、package ではなく port から構築をしてください。たとえば、英語のドキュメントを構築してインストールするには以下のようにして下さい。

```
# cd /usr/ports/misc/freebsd-doc-en
# make install clean
```

この port には、構築およびインストールするフォーマットを設定するメニューがあります。デフォルトでは、http://www.FreeBSD.org と同じ形式である分割版の HTML 形式、PDF が選択されています。

以下のように、ドキュメンテーション ports を構築する際の make オプションが用意されています。

WITH_HTML
: HTML 形式を構築します。各ドキュメントに対し、単一版の HTML ファイルが構築されます。整形されたドキュメントは、article.html や book.html といった名前でインストールされます。

WITH_PDF
: 整形されたドキュメントは、article.pdf や book.pdf といった名前でインストールされます。

DOCBASE
: ドキュメントのインストール先を設定します。デフォルトのインストール先は `/usr/local/share/doc/freebsd` です。

以下は、上記の変数を用いてハンガリー語のドキュメントを PDF 形式でインストールする方法です。

```
# cd /usr/ports/misc/freebsd-doc-hu
# make -DWITH_PDF DOCBASE=share/doc/freebsd/hu install clean
```

5章アプリケーションのインストール - packages と ports に書かれている手順を使って、ドキュメンテーション package または port をアップデートできます。たとえば、以下のコマンドを実行すると、ports-mgmt/portupgrade から、package だけを使ってインストールされているハンガリー語のドキュメントをアップデートします。

```
# portmaster -PP hu-freebsd-doc
```

18.4. 開発ブランチを追いかける

FreeBSD には二つの開発ブランチがあります。それは FreeBSD-CURRENT と FreeBSD-STABLE です。

この節ではそれぞれのブランチと対象としている読者についての説明と、どのようにしてシステムの対応するブランチを最新の状態に保つかについて説明します。

訳: 花井 浩之 **<hanai@FreeBSD.org >**、1996 年 11 月 6 日

18.4.1. FreeBSD-CURRENT を使う

FreeBSD-CURRENT とは FreeBSD の開発の「最前線」なので、FreeBSD-CURRENT のユーザは高い技術力を持つことが要求されます。そこまでの技術力を持っていないが、開発ブランチを追いかけたいと考えているユーザは、かわりに FreeBSD-STABLE を追いかけると良いでしょう。

FreeBSD-CURRENT は FreeBSD の最新のソースコードであり、中には現在開発途上のソフトウェア、実験的な変更、あるいは過渡的な機能などが含まれています。また、この中に入っている機能がすべて、次の公式リリースに入るとは限りません。FreeBSD-CURRENT をソースからほぼ毎日コンパイルしている人はたくさんいますが、短い期間ではコンパイルさえできない状態になっている時期もあります。これらの問題は可能な限り迅速に解決されますが、FreeBSD-CURRENT が不幸をもたらすか、それとも新しい機能をもたらすかは、まさにソースコードを同期した瞬間によるのです!

FreeBSD-CURRENT は、次の 3 つの重要なグループを対象としています。

1. ソースツリーのある部分に関して活発に作業している FreeBSD コミュニティのメンバ。

2. 活発にテストしている FreeBSD コミュニティのメンバ。彼らは、種々の問題を解決するのに時間を惜しまない人々であり、さまざまな変更に関する提案や FreeBSD の大まかな方向付けを行ないたいと思っている人々でもあり、パッチも提出します。

3. さまざまな事に目を向け、参考のために最新のソースを使いたいと思っていたり、時々コメントやコードを寄稿したいと考えているユーザ。

FreeBSD-CURRENT は、次のリリースの前に、最も早く新しい機能を入手する手段として、 期待してはいけません。 リリース前の機能は十分にテストされていないため、 バグを含んでいる可能性が大いにあるためです。 また、バグを修正するための素早い方法でもありません。いかなるコミットは、元からあるバグを修正するのと同じく、新しいバグを生み出すおそれがあります。FreeBSD-CURRENT には「公式のサポート」はありません。

FreeBSD-CURRENT を追いかけるには

1. freebsd-current と svn-src-head メーリングリストに加わってください。 さまざまな人がシステムの現在の状態について述べているコメントを見たり、FreeBSD-CURRENT の現在の状態に関する重要な情報を見逃さないために、必須の ことです。

   svn-src-head メーリングリストでは、それぞれの変更についての commit ログが記録されています。また、それに関して起こり得る副作用の情報を得ることができますので、参加する価値のあるメーリングリストです。

   これらのメーリングリストに入るには、 http://lists.FreeBSD.org/mailman/listinfo をたどって参加したいメーリングリストをクリックし、 手順の説明にしたがってください。FreeBSD-CURRENT だけでなく、ソースツリー全体の変更点を追いかけるのであれば、svn-src-all メーリングリストを購読してください。

2. FreeBSD-CURRENT のソースを同期してください。 特に svn を使って「Subversion ミラーサイト」 の一覧にある Subversion ミラーサイトのひとつの `head` ブランチから -CURRENT コードをチェックアウトしてください。

3. インターネットの接続がとても遅かったり、制限がある場合には、「CTM を使う」で説明されている CTM を利用すると良いでしょう。ただし、svn ほどには信頼はできないので、svn を利用されることを推奨します。

4. リポジトリのサイズが大きいため、興味のある部分や、パッチを当てる部分のソースのみを同期するユーザもいます。しかしながら、ソースからオペレーティングシステムをコンパイルしようと思っているユーザは、一部分だけではなく、FreeBSD-CURRENT の すべて をダウンロードする必要があります。

   FreeBSD-CURRENT をコンパイル する前に `/usr/src/Makefile` を注意深く読み、「world の再構築」 に書かれている手順に従ってください。 FreeBSD-CURRENT メーリングリスト と `/usr/src/UPDATING` を読めば、次のリリースへ

向けて移ってゆくに当たって、ときどき必要となる既存システムからの新システムの構築手順についての最新情報が得られるでしょう。

5. アクティブになってください! FreeBSD-CURRENT のユーザには、拡張やバグ潰しに関して提案することが勧められています。コードを伴う提案はいつでも歓迎されます!

18.4.2. FreeBSD-STABLE を使う

訳: 岩崎 満 **<iwasaki@jp.FreeBSD.org >**

FreeBSD-STABLE とは定期的に公開されるリリースを作成するための開発ブランチです。このブランチに加えられる変更は FreeBSD-CURRENT よりゆっくりで、原則として、事前に FreeBSD-CURRENT で試験ずみであるという特徴があります。ただそうであっても、これは開発用ブランチの一つであり、ある時点における FreeBSD-STABLE のソースがどんな場合にも使えるものであるとは限りません。このブランチはもう一つの開発の流れというだけであって、エンドユーザ向けのものではありません。もし試験をする資源的な余裕がない場合は、代わりに最新の FreeBSD リリースを使ってください。

FreeBSD の開発プロセスに興味があったり、それに対する貢献を考えていて、特にそれが次回の FreeBSD のリリースに関係するものであるなら FreeBSD-STABLE を追うことを考えると良いでしょう。

FreeBSD-STABLE ブランチはいつもコンパイルができ、安定に動作すべきですが、それが保証されているというわけではありません。FreeBSD-STABLE のユーザは FreeBSD-CURRENT よりも多いため、FreeBSD-CURRENT で発見されなかったバグが FreeBSD-STABLE で発見され、ときどきそれが問題となることがあるのは避けることができません。このような理由から、盲目的に FreeBSD-STABLE を追いかけるべきではありません。特に、開発環境もしくはテスト環境でコードを十分に試験せずに、プロダクション品質が要求されるサーバを FreeBSD-STABLE にアップグレードしてはいけません。

FreeBSD-STABLE を追いかけるには

1. FreeBSD-STABLE の構築に関連する事柄や、その他の注意すべき点 に関する情報を得るために、freebsd-stable メーリングリストに加わってください。また開発者は議論の余地がある修正や変更を考えている場合に、このメーリングリストで公表し、提案された変更に関して問題が生じるかどうかを返答する機会をユーザに与えます。

   追いかけているブランチに関連する svn メーリングリストに参加してください。たとえば、9-STABLE ブランチを追いかけているユーザは svn-src-stable-9 メーリングリストに参加してください。このリストでは、変更がなされるごとに作成される commit log やそれに伴う起こりうる副作用についての情報が記録されています。

   これらのメーリングリストに入るには、http://lists.FreeBSD.org/mailman/listinfo をたどって参加したいメーリングリストをクリックし、手順の説明にしたがってください。

ソースツリー全体の変更点を追いかけるには、svn-src-all メーリングリストを購読してください。

2. 新しい FreeBSD-STABLE システムをインストールするには、ミラーサイト から最近の FreeBSD-STABLE リリースをインストールするか、毎月公開されている FreeBSD-STABLE からビルドされたスナップショットを使ってください。スナップショットの詳細については、www.freebsd.org/ja/snapshots をご覧ください。

   既に FreeBSD が動いているシステムを FreeBSD-STABLE にアップグレードするには、svn を使って、希望する開発ブランチのソースをチェックアウトしてください。`stable/9` といったブランチ名は、www.freebsd.org/releng で説明されています。インターネットへの接続に信頼できる回線を利用できないのであれば、CTM (「CTM を使う」) を使ってください。

3. FreeBSD-STABLE をコンパイルしたり FreeBSD-STABLE へとアップグレードする前に、`/usr/src/Makefile` を注意深く読み、「world の再構築」に書かれている手順に従ってください。FreeBSD-STABLE メーリングリスト と `/usr/src/UPDATING` を読んで、次のリリースへ向けて移ってゆくに当たって、ときどき必要となる既存システムからの新システムの構築手順についての最新情報を得てください。

18.5. ソースの同期

訳: 岩崎 満 **<iwasaki@jp.FreeBSD.org >**、1997 年 9 月 13 日

FreeBSD のソースの最新を追いかける方法は色々あります。この節では、基本的なサービスである Subversion および CTM について説明します。

警告

ソースツリーの一部を最新のものに更新することは可能です。ただし、サポートされているアップデート手順は、ソースツリー全体を最新のものに更新し、`/bin`, `/sbin` といったユーザ空間で動作するもの、およびカーネルソースを再構築することのみです。ソースツリーの一部だけであったり、カーネルだけ、もしくはユーザランドのプログラムだけを更新した場合は、問題が生じることがよくあります。この時に発生する問題はコンパイル時のエラーからカーネルパニック、データの破壊とさまざまです。

Subversion は pull 同期モデルを採用しています。ユーザ (または `cron` スクリプト) が `svn` プログラムを起動し、ローカルにあるソースを最新状態にします。更新情報はその時点の最新のものであり、いつダウンロードするかはユーザがコントロー

ルするので、Subversion はローカルのソースツリーをアップデートする好ましい方法です。特定のファイルやディレクトリに限定して更新することも簡単にできます。更新情報はサーバによって素早く生成されます。Subversion によるソースの同期方法については、「Subversion を使う」で説明されています。

一方、CTM はあなたが持っているソースとマスタアーカイブ上に あるそれとの対話的な比較をおこないませんし、あるいは向こう側から変更点を pull したりもしません。そのかわりに、前回の実行時からの変更を認識するスクリプトが マスタ CTM マシン上で一日に数回実行され、すべての変更を compress して通し番号を振り、さらに電子メールで、印字可能な ASCII キャラクタのみで転送できるようにエンコードします。ダウンロードした後は、これらの「デルタ」は、自動的にデコード、検査してユーザのソースのコピーに変更を適用する `ctm.rmail` によって処理可能です。この処理は Subversion よりずっと効率的であり、pull モデルというよりむしろ push モデルであるため、サーバ資源の負荷は軽くなります。CTM を用いたソースの同期方法については、「CTM を使う」をご覧ください。

Subversion であれば、うっかりローカルのアーカイブの一部を消してしまっても、壊れた部分を検出して再構築してくれます。CTM はこれをやってくれません。もしソースツリーの一部を消してしまい、そしてバックアップを取っていないのであれば、最新の ベースデルタ を用いて、一からやり直し、CTM を使ってすべてを再構築しなければなりません。

18.6. world の再構築

FreeBSD-STABLE、FreeBSD-CURRENT などの FreeBSD のどれか特定のバージョンについて、ローカルのソースツリーを同期させたら、そのソースツリーを使ってシステムを再構築できます。このプロセスは world の再構築と呼ばれます。

world を再構築する前に、以下を行ってください。

手順18.1 world の構築前に行う作業

1. 重要なデータを他のシステムやリムーバブルメディアにバックアップし、きちんとバックアップが作成されていることを確認したら、 起動可能なインストールメディアを用意してください。システムを再構築する前に、バックアップを作成することの重要性は、いくら強調してもし過ぎると言うことはありません。システム全体の再構築は難しい作業ではありませんが、 どんなに注意していたとしても、 ソースツリーそのものに手違いがあった時には、 システムが起動しなくなってしまう状態になることがあるのです。多分、それを使うことはないと思いますが、あとで後悔することのないよう、念のため用意しておきましょう!

2. 追いかけているブランチに応じて、freebsd-stable もしくは freebsd-current の最近のエントリを調べて、既知の問題や影響を受けるシステムを確認してください。既

知の問題が同期しているバージョンのコードに影響する場合は、その問題が解決されたことを報告する「問題解決 (all clear)」のアナウンスが投稿されるまで待ってから、ソースを同期して、ローカルのソースに必要な修正を入れてください。

3. `/usr/src/UPDATING` を読み、同期しているソースのバージョンで必要となるステップがないかどうかを調べて下さい。このファイルには潜在的な問題や特定のコマンドを実行する順などの重要な情報が含まれています。大きなアップグレードでは、world をインストールする前に特定のファイルの名前を変更したり、削除するといった、特別なステップが追加で必要となることがあります。ファイルの最後には、現在推奨されているアップグレードの手順が詳しく正確に説明されています。もし、`UPDATING` に書かれている手順が、この節に書かれているものと矛盾していたら、`UPDATING` の手順を採用してください。

`make world` は使わないこと

古いドキュメントの中には、`make world` を使うことを薦めているものがあります。これは、重要な手順をいくつか抜かしてしまうので、エキスパートでなければ使うべきではありません。ほぼあらゆる場合において、`make world` を実行するのは間違っており、ここで説明されている手順を用いるべきです。

18.6.1. システム更新の概要

world の構築では、「ソースの同期」に書かれている手順で入手したソースを用いて、古い FreeBSD のバージョンをアップデートしてください。

FreeBSD では、「world」は、カーネル、コアシステムのバイナリ、ライブラリ、プログラミングファイル、組み込みのコンパイラを意味します。これらのコンポーネントの構築およびインストールの順番は重要です。

例えば、古いコンパイラは、バグを含み、新しいカーネルをコンパイルできない可能性があります。そのため、新しいカーネルの構築には、新しいコンパイラを使う必要があり、新しいコンパイラを構築しなくてはいけませんが、必ずしも、新しいコンパイラがインストールされている必要があるわけではありません。

新しい world は、新しいカーネルの機能に依存している可能性があるため、新しい world をインストールする前に、新しいカーネルがインストールされていなければなりません。古い world は、新しいカーネルでは正しく動かないかも知れません。そのため、新しいカーネルをインストールしたら、直ちに新しい world をインストールしてください。

設定の中には、新しい world をインストールする前に変更すべきものがありますが、古い world を壊す可能性があります。そのため、一般的に設定のアップデートは、2 つの

手順が使われます。多くの場合、アップデートのプロセスは、ファイルを置き換えたり、追加のみを行い、古いファイルを削除しません。このことが問題を引き起こす可能性があるため、/usr/src/UPDATING には、手動で削除すべきファイルをどのステップで削除すべきかが書かれています。

これらを配慮し、アップグレードの推奨手順が作られました。

注記

make を実行したときの出力は、 ファイルに保存すると良いでしょう。 何か障害が発生した場合には、エラーメッセージのコピーを FreeBSD メーリングリストに投稿してください。

ファイルに保存する最も簡単な方法は、script コマンドを使い、引数として出力の保存先のファイル名を指定することです。/tmp に出力を保存しないようにしてください。 このディレクトリは、次の再起動で削除されてしまう可能性があります。出力の保存には、/var/tmp が適しています。次のコマンドを world の構築の直前に行ない、再構築が終了したら **exit** と入力してください。

```
# script /var/tmp/mw.out
Script started, output file is /var/tmp/mw.out
```

手順18.2 world の構築プロセスの概要

world の構築プロセスで用いられるコマンドは、ここで示されている順番で実行してください。この節では各コマンドの機能についてまとめます。

1. システム上で world の構築が一度でも行われたのであれば、前回の構築の際のコピーが /usr/obj に存在するはずです。このディレクトリが存在しているのであれば、このディレクトリを削除して、make buildworld の行程にかかる時間を短縮し、依存問題に悩まされるようなトラブルを回避することができます。

```
# cd /usr/obj
# chflags -R noschg *
# rm -rf *
```

2. 新しいコンパイラと関連ツールを最初にコンパイルし、その後、新しいコンパイラで、新しい world の残りの部分をコンパイルします。コンパイルされたものは、/usr/obj に格納されます。

```
# cd /usr/src
# make buildworld
```

3. コンパイラとカーネルのミスマッチを防ぐため、`/usr/obj` にある新しいコンパイラを用いて新しいカーネルを構築します。 再構築は、ある種のメモリ構造体が変更されたような場合には必須で、`ps` や `top` のようなプログラムは、カーネルとソースコードのバージョンが一致しないと正常に動作しないでしょう。

```
# make buildkernel
```

4. 新しくアップデートされたカーネルで起動できるように、 新しいカーネルとカーネルモジュールをディスク上に配置します。 `kern.securelevel` を 1 より大きくしていて、 かつ カーネルのバイナリファイルに `noschg` のようなフラグを設定している場合は、 まず、シングルユーザモードに移行してください。 それ以外の場合は、マルチユーザモードでこれらのコマンドを問題なく動かせるはずです。 `kern.securelevel` について詳しくは init(8) を、ファイルに対するさまざまなフラグについて詳しくは chflags(1) をご覧ください。

```
# make installkernel
```

5. すでに実行されているソフトウェアをアップデートする際の問題を最小限にするため、シングルユーザモードに移行します。また、新しいカーネル上で古い world が実行される際の問題も最小限にします。

```
# shutdown now
```

シングルユーザモードに移行したら、UFS でフォーマットされているシステムでは、以下のコマンドを実行してください。

```
# mount -u /
# mount -a -t ufs
# swapon -a
```

もしシステムが ZFS でフォーマットされている場合には、以下の 2 つのコマンドを実行してください。この例では、zpool の名前は `zroot` であると仮定します。

```
# zfs set readonly=off zroot
# zfs mount -a
```

6. オプション: もし、デフォルトの US English 以外のキーボードマップが必要であれば、kbdmap(1) で変更してください。

```
# kbdmap
```

7. その後、どちらのファイルシステムでも、 CMOS クロックが地域時間に設定されていて GMT ではない場合 (date(1) が正しい時間と地域を表示しないなら当てはまります) には、次のコマンドを実行してください。

```
# adjkerntz -i
```

8. システムの再構築では、/etc, /var や /usr といったディレクトリに新規に導入されたファイルや、変更された設定ファイルに対する更新は行なわれません。次に、新しい world に対する /etc の最初の設定ファイルのアップデートを行います。以下のコマンドは installworld に成功するために本質的なファイルのみを比較します。たとえば、このステップでは、最後のアップデート後に FreeBSD に追加された、新しいグループや新しいシステムアカウント、もしくはスタートアップスクリプトがシステムに追加されることがあります。次のコマンドに関する詳細な情報については、「設定ファイルの同期」を参照してください。

```
# mergemaster -iF
```

9. /usr/obj にある新しい world およびシステムのバイナリをインストールします。

```
# cd /usr/src
# make installworld
```

10. 残りの設定ファイルをアップデートします。

```
# mergemaster -p
```

11. 使われなくなったファイルを削除します。もし使われなくなったファイルがディスクに残っていると、問題が起きる可能性があるため重要な作業です。

```
# make delete-old
```

12. 再起動を行い、新しいカーネル、world そして設定ファイルをロードします。

```
# reboot
```

13. 古いライブラリを削除する前に、「ports のアップグレード」に書かれている手順にしたがって、すべての ports を再構築する必要があります。再構築が終わったら、新しいライブラリと競合することを避けるため、 使われなくなったライブラリを削除します。 この過程に関する詳細については、「使われなくなったファイル、ライブラリの削除」を参照して下さい。

```
# make delete-old-libs
```

もしシステムがダウンタイムを持つことができるのであれば、システムのコンパイルをマルチユーザモードでおこない、インストールのためにシングルユーザモードに移行するという方法ではなく、コンパイルをシングルユーザモードで行うことを考えてください。システムの再インストールでは、たくさんの重要なシステムファイル、 すべての標準的なシステムバイナリ、ライブラリ、 インクルードファイルが変更されるので、 実際に動作しているシステムにおいて、特にアクティブなユーザは、トラブルに見舞われる可能性があります。

18.6.2. 設定ファイル

world の構築プロセスでは、いくつかの設定ファイルが使われます。

/usr/src に置かれている、Makefile には、FreeBSD を構成するプログラムの構築方法や、どういう順番でそれらを構築すべきかといった指示が記述されています。

make で利用可能なオプションの説明は make.conf(5) や、共通の例が /usr/share/examples/etc/make.conf にあります。 これらの設定を /etc/make.conf に追加すると、make の実行やプログラムの構築方法を設定できます。これらのオプションは、make が使われる際には常に有効となるため、Ports Collection からアプリケーションをコンパイルする時、ユーザが書いた C プログラムや FreeBSD オペレーティングシステムを構築する際に影響を及ぼします。ある設定を変更したことにより、影響が広い範囲におよび、驚くべき結果をもたらす可能性があります。両方のファイルに書かれているコメントを読むことと、デフォルトの設定は、パフォーマンスと安全性の観点から選ばれていることを覚えておいてください。

/etc/src.conf は、ソースコードを用いたオペレーティングシステムの構築をコントロールします。/etc/make.conf とは異なり、/etc/src.conf に書かれた設定は、FreeBSD オペレーティングシステムそのものを構築するときにのみ影響します。このファイルで設定可能な多くのオプションについては、src.conf(5) に記述されています。 一見したところ無効にされている、 使われていないカーネルモジュールやビルドオプションに注意してください。 ときどき予期しなかったり、わずかな影響を与えることがあります。

18.6.3. 変数とターゲット

make の使用における一般的な書式は、次のとおりです。

```
# make -x -DVARIABLE target
```

この例では、*-x* が make に渡されるオプションになります。どのようなオプションが利用できるかについては、make(1) を参照してください。

変数を渡すには、変数の名前を *-DVARIABLE* のように指定してください。この変数は Makefile の動作をコントロールします。変数の指定は、/etc/make.conf で設定するか、make の実行時に指定するかのどちらかで行います。たとえば、以下の変数は、プロファイル版のライブラリを構築しないことを指定します。

```
# make -DNO_PROFILE target
```

これは、/etc/make.conf の中で以下のように設定することに対応します。

```
NO_PROFILE=    true     #    Avoid compiling profiled libraries
```

target は、make にどのように動作するのかを指示するためのものです。Makefile は利用可能なターゲットを定義しています。ターゲットには、システムの再構築に必要な

段階を、多くのさらに細かい段階に分割するため、構築の過程で利用されるものがあります。

選択肢が分けられていることは、二つの理由から有用です。まず第一に、構築作業は稼働中のシステムにまったく影響を与えません。そのため、マルチユーザモードで稼働中のシステムでも、安全に `buildworld` を実行できます。ただし、`installworld` はシングルユーザモードで行なうことをおすすめします。

第二に、NFS マウントを利用することで、「複数のマシンで追いかける」で説明されているように、ネットワーク上の複数のマシンをアップグレードすることが可能な点があげられます。

`make` に `-j` をつけると、同時に複数のプロセスを生成できます。構築過程の大部分では CPU 性能の限界より I/O 性能の限界の方が問題となるため、シングル CPU とマルチ CPU マシンの両方に効果があります。

普通のシングル CPU マシンで以下のコマンドを実行すると、最大 4 個までのプロセスを同時に実行します。メーリングリストに投稿された経験的な報告によると、4 個という指定が最も良いパフォーマンスを示すようです。

```
# make -j4 buildworld
```

マルチ CPU マシンでは、`6` から `10` の間の値を設定し、速度がどれくらい向上するか確認してみてください。

注記

`make buildworld` に変数を指定した場合は、同じ指定を `make installworld` にも指定しなければなりません。ただし `installworld` では、`-j` を 絶対に使ってはいけません。

たとえば、以下のコマンドを実行したなら、

```
# make -DNO_PROFILE buildworld
```

以下のようにしてインストールしなければなりません。

```
# make -DNO_PROFILE installworld
```

もしそうしなかった場合、2 番目のコマンドは、`make buildworld` の段階で構築されていないプロファイル版ライブラリをインストールしようとしてしまうでしょう。

18.6.4. 設定ファイルの同期

##: Tom Rhodes.

FreeBSD の mergemaster(8) Bourne シェルスクリプトは、`/etc` にある設定ファイルと、`/usr/src/etc` にある設定ファイルの違いを確認するためのものです。これを使うのが、ソースツリーにある設定ファイルにシステムの設定ファイルを 更新するために推奨される方法です。

`mergemaster` を使う前に、既存の `/etc` をどこか安全な場所にコピーしておきましょう。再帰的なコピーを行なう `-R` と、ファイルの更新時間や所有者などを保存する `-p` と共に実行してください。

```
# cp -Rp /etc /etc.old
```

`mergemaster` を実行すると、`/` を起点とした一時的なルート環境を構築し、さまざまなシステム設定ファイルを (訳注: デフォルトでは `/var/tmp/temproot` に) 置いていきます。これらのファイルは現在システムにインストールされているファイルと比較されます。異なるファイルは diff(1) 形式で示され、`+` の記号は追加または変更された行を表し、`-` は完全に削除されたか新しく置き換えられた行を表します。ファイルの違いの表示方法についてのより詳しい情報は、diff(1) を参照してください。

次に mergemaster(8) は違いのあるファイルをそれぞれ示し、選択可能なオプションを表示します。 ここでは、新しいファイルを削除するか、 一時ファイルをそのままインストールするか、一時ファイルと現在インストールされているファイルを統合するか、もしくは結果をもう一度見るかを選択できます。

一時ファイルの削除を選ぶと、`mergemaster` に現在のファイルを変更しないで新しいバージョンを削除せよと伝えます。この選択は、お勧めできません。`mergemaster` のプロンプトで ? とタイプすれば、いつでもヘルプが見られます。ファイルのスキップを選ぶと、他のすべてのファイルを終えたあと、もう一度そのファイルが提示されます。

一時ファイルをそのままインストールすることを選ぶと、 現在のファイルを新しいファイルで置き換えます。ほとんどの手を加えていないファイルは、これが一番よい選択です。

ファイルの統合を選んだ場合、テキストエディタが起動され、両方のファイルの中身が提示されます。 画面上に並ぶ両方のファイルを見て新しいファイルを作成するために両方から必要な部分を選択し、2 つのファイルを統合することができます。並んでいるファイルを比較するとき、l で左側の中身を選択し、r で右側の中身を選択します。最終出力は左右両方の部分でできたファイルになるでしょう。 このファイルをインストールすることができます。たいてい、このオプションはユーザが設定を変更したファイルに使われます。

結果をもう一度見る、を選択すると、ファイルの相異点をもう一度見ることができます。

`mergemaster` がシステムファイルの比較を終えたあと、他のオプションについてもプロンプトが表示されます。たとえば、パスワードファイルを再構築するかどうかを尋ねることがあります。最後に残った一時ファイルを削除するかどうかを尋ねて終了します。

18.6.5. 使われなくなったファイル、ライブラリの削除

############: Anton Shterenlikht.

FreeBSD の開発サイクルにおいて、ファイルやシステムの一部が使われなくなることがあります。 それらの機能が別の場所で実装されたり、 ライブラリのバージョン番号が変わったり、 システムから完全に削除されることがあるためです。 システムのアップデート時に削除が必要になるのは、 古いファイル、ライブラリそしてディレクトリです。 これらのファイルを削除することで、 記憶媒体やバックアップ媒体において不必要な容量を占めている古いファイルが、 システム上に散乱することがなくなります。 また、古いライブラリのセキュリティや安定性に問題があると、ライブラリを新しくしてシステムを安定な状態にし、 古いライブラリによりシステムがクラッシュすることを防がなければなりません。 使われなくなったファイル、ディレクトリ、ライブラリは `/usr/src/ObsoleteFiles.inc` にまとめられています。以下の手順により、 アップグレードの過程でこれらのファイルを削除できます。

`make installworld` と、その後の `mergemaster` が無事に終わったら、使われなくなったファイルやライブラリを確認してください。

```
# cd /usr/src
# make check-old
```

見つかった古いファイルは、以下のコマンドで削除できます。

```
# make delete-old
```

使われなくなったファイルを削除する際、ファイルごとに確認が求められます。確認を省略し、自動的にファイルを削除するには、 以下のように `BATCH_DELETE_OLD_FILES` を設定してください。

```
# make -DBATCH_DELETE_OLD_FILES delete-old
```

`yes` をコマンドへパイプでつなげても省略できます。

```
# yes|make delete-old
```

Warning

使われなくなったファイルを削除すると、 削除したファイルに依存していたアプリケーションは壊れてしまいます。 特に、古いライブラリを削除する場合に起こり得ます。通常、`make delete-old-libs` を実行する前に、 これらの古いライブラリを使っているプログラム、ports、ライブラリを再構築する必要があります。

共有ライブラリをチェックするユーティリティとして、 sysutils/libchk や sysutils/bsdadminscripts を利用できます。

使われなくなった共有ライブラリは、 新しいライブラリと競合し、以下のようなメッセージを表示することがあります。

```
/usr/bin/ld: warning: libz.so.4, needed by /usr/local/lib/libtiff.s
o, may conflict with libz.so.5
/usr/bin/ld: warning: librpcsvc.so.4, needed by /usr/local/lib/
libXext.so, may conflict with librpcsvc.so.5
```

この問題を解決するには、まずライブラリがどの port によってインストールされたかを調べて下さい。

```
# pkg which /usr/local/lib/libtiff.so
  /usr/local/lib/libtiff.so was installed by package tiff-3.9.4
# pkg which /usr/local/lib/libXext.so
  /usr/local/lib/libXext.so was installed by package libXext-1.1.1,1
```

見つかった port をアンインストールし、再構築、再インストールしてください。この過程は ports-mgmt/portmaster で自動化できます。すべての ports が再構築され、古いライブラリがどこにも使われていないことを確認したら、以下のコマンドで古いライブラリを削除してください。

```
# make delete-old-libs
```

もしちょっとした問題があった場合でも、システムの一部を再構築するのは簡単です。たとえば、アップグレードや `/etc` のマージの途中で誤って `/etc/magic` を削除してしまい、その結果 `file` が動作しなくなってしまったような場合には、次のコマンドを実行して修復してください。

```
# cd /usr/src/usr.bin/file
# make all install
```

18.6.6. よくある質問

変更が行なわれたら、その度にシステムの再構築が必要になるのでしょうか?

それは変更の内容によります。たとえば、svn を実行したとき、次にあげるようなファイルが更新されていたとします。

```
src/games/cribbage/instr.c
src/games/sail/pl_main.c
src/release/sysinstall/config.c
src/release/sysinstall/media.c
src/share/mk/bsd.port.mk
```

このときには、改めてシステム全体を再構築する必要はないでしょう。 そのかわり、適切なサブディレクトリに移って `make all install` を行ってください。しかし、

たとえば src/lib/libc/stdlib のような大きな変更が行なわれた場合には、システム全体を再構築することを検討してください。

2 週間ごとにシステムを再構築して、その 2 週間分の変更を取り込むユーザもいますし、変更のあった部分だけ再構築し、すべての依存関係を確かめたいと考えるユーザもいます。それらはどのくらいの頻度でアップグレードしたいか、そして FreeBSD-STABLE か FreeBSD-CURRENT のどちらを追いかけているのかにもよります。

どうして signal 11 (もしくは他のシグナル番号) のエラーがたくさん出て コンパイルが失敗するのでしょうか?

これは通常、ハードウェアに問題があることを示しています。world の再構築は、ハードウェア (特にメモリ) に対する負荷耐久試験を行なうための有効な手段です。本当にこの問題によるものかどうかは、make をもう一度実行し、異なる段階で異常終了が発生するか、ということから確認できます。

このエラーに対応するには、RAM を始めとして、マシンの部品をメモリから交換して、どの部分が悪いのかを調べてみてください。

終了したら /usr/obj を削除してもかまいませんか?

このディレクトリには、コンパイルの段階で生成された すべてのオブジェクトファイルが含まれています。通常 make buildworld の最初の段階では、このディレクトリを削除して新しくつくり直すようになっています。構築終了後も /usr/obj を保存しておいても、あまり意味はありません。削除すれば、だいたい 2GB のディスクスペースを解放することができます。

構築を中断した場合、その構築を途中から再開することはできますか?

それは、問題が起こるまでに、どれだけの作業を終えているかによります。一般的に make buildworld は、基本的なツールや、システムライブラリの新しいコピーを作成します。その後、これらのツールやライブラリがインストールされてから、自分自身の再構築に使われ、もう一度、インストールされます。システムの残りの部分がその新しいシステムファイルを用いて作り直されます。

再構築の最終段階では、まったく安全に以下のコマンドを実行することができます。これは、前回の make buildworld の作業をやり直しません。

```
# cd /usr/src
# make -DNO_CLEAN all
```

次のメッセージ

```
--------------------------------------------------------------
Building everything..
--------------------------------------------------------------
```

が make buildworld の出力にある場合には、上のようにしてもほとんど悪影響が現れることはありません。

もしこのメッセージがない場合には、 安全を確保し、後悔するようなことがないよう、システムの再構築を最初からやり直しましょう。

make world を高速化できますか?

いくつかの方法で build world のプロセスを高速化できます。 たとえば、全体のプロセスは、シングルユーザモードで動かすことで高速になります。しかしながら、この方法では、プロセスが完了するまで、ユーザがシステムにアクセスすることはできません。

ファイルシステムを注意深く設計したり、 ZFS データセットを使うことでも変わります。`/usr/src` と `/usr/obj` を、異なるディスク上の別のファイルシステムに置くことを検討してください。また可能ならば、異なるディスクコントローラに接続された異なるディスクにファイルシステムを置いてください。`/usr/src` をマウントする時には、最後にアクセスされた時刻の書き込みを抑制するように、`noatime` オプションを付けてマウントしてください。もし、`/usr/src` が、独立したファイルシステムではないときには、`noatime` オプションで、`/usr` を再マウントしてください。

`/usr/obj` のあるファイルシステムを、`async` オプションをつけてマウントもしくは再マウントしてください。 これによって、ディスクへの書き込みが非同期になります。つまり、書き込み命令はすぐに完了するのに対し、実際にデータがディスクに書き込まれるのは、その数秒後になります。これによって、書き込み処理の一括化が可能になるため、劇的なパフォーマンスの向上が期待できます。

警告

このオプションを指定すると、ファイルシステムは 壊れやすくなってしまうことに注意してください。 このオプションを付けていて、突然電源が落ちた場合には、 再起動後にファイルシステムが復旧不能になる可能性が 非常に高くなります。

もし、`>/usr/obj` が、ファイルシステムにある唯一のディレクトリであれば、 これは問題になりません。しかし、同じファイルシステムに、他の貴重なデータを置いているときには、このオプションを有効にする前に、バックアップをきちんと取っておきましょう。

`/etc/make.conf` に「NO_PROFILE=true」をセットして、プロファイル版の作成を無効化してください。

make(1) に `-jn` を指定して、複数のプロセスを並列に実行させてください。 これは、単一のプロセッサでも複数のプロセッサでも、 同様に恩恵を得ることができます。

なにか悪いことがあったらどうすればいいですか?

まず、自分の環境に前のビルドの余計なゴミが残っていないことをはっきりと確認してください。

```
# chflags -R noschg /usr/obj/usr
# rm -rf /usr/obj/usr
# cd /usr/src
# make cleandir
# make cleandir
```

ええ、`make cleandir` は本当に 2 回実行するのです。

そして、`make buildworld` を行い、全プロセスを最初からやり直してください。

まだ問題があれば、エラーと `uname -a` の出力を FreeBSD general questions メーリングリスト に送ってください。設定についてさらに質問されても答えられるよう用意してください!

18.7. 複数のマシンで追いかける

##: Mike Meyer.

複数のコンピュータで同じソースツリーを追いかけていて、 全部のマシンにソースをダウンロードして全部を再構築するのは、 ディスクスペース、ネットワーク帯域、 そして CPU サイクルの無駄使いです。解決策は 1 つのマシンに仕事のほとんどをさせ、残りのマシンは NFS 経由でそれをマウントする、というものです。このセクションではそのやり方を概観します。NFS の使い方の詳細については、「NFS」をご覧下さい。

まず初めに、同じバイナリで動かそうとするマシンたちを決めます。 このマシンたちのことをビルドセットと呼びます。それぞれのマシンはカスタムカーネルを持っているかもしれませんが、 同じユーザランドバイナリを動かそうというのです。 このビルドセットから、 ビルドマシンとなるマシンを 1 台選びます。 ベースシステムとカーネルを構築するのはこのマシンになります。 理想的には、このマシンは `make buildworld` と `make buildkernel` を実行するのに十分な CPU を持った速いマシンであるべきです。

テストマシン となるべきマシンも選んでください。更新されたソフトウェアを使う前にそのマシンでテストするのです。テストマシンはかなり長い時間落ちていても だいじょうぶなマシンであったほうがいいでしょう。ビルドマシンでもかまいませんが、ビルドマシンである必要はありません。

このビルドセットのマシンはすべて `/usr/obj` と `/usr/src` をビルドマシンから FTP 経由でマウントする必要があります。 ビルドセット自体が複数ある場合は、 `/usr/src` はひとつのビルドマシン上にあるべきです。他のマシンからはそれを NFS マウントするようにしましょう。

ビルドセットのすべてのマシン上の `/etc/make.conf` と `/etc/src.conf` がビルドマシンと一致していることを確認してください。つまり、ビルドマシンはビルドセットのどのマシンもインストールしようとしている ベースシステムを全部ビルドしなければならないということです。また、各ビルドマシンは `/etc/make.conf` にそれぞれのビルドマシンのカーネル名を `KERNCONF` で指定し、ビルドマシンは自分自身のカーネルから順に全部のカーネル名を `KERNCONF` にリストアップしてください。ビルドマシンは各マシンのカーネル設定ファイルを `/usr/src/sys/`*arch*`/conf` に持っていなければなりません。

ビルドマシンにて、「world の再構築」に書いてあるようにカーネルとベースシステムを構築してください。 でも、まだビルドマシンにはインストールしないでください。 そのかわり、ビルドしたカーネルをテストマシンにインストールしてください。FTP 経由で `/usr/src` および `/usr/obj` をテストマシンにマウントしてください。 その後、`shutdown now` を実行してシングルユーザモードに移行し、新しいカーネルとベースシステムをインストールし、いつもするように `mergemaster` を実行してください。終わったら、再起動して通常のマルチユーザ動作に戻します。

テストマシンにあるものすべてがちゃんと動いている確信が得られたら、同じ手順でビルドセットの他のマシンにも新しいソフトウェアをインストールします。

ports ツリーにも同じ方法が使えます。 最初のステップは、 ビルドセットのすべてのマシンが NFS 経由で `/usr/ports` をマウントすることです。 そして、distfiles を共有するように `/etc/make.conf` を設定します。NFS マウントによってマップされる `root` ユーザが何であれ、`DISTDIR` はそのユーザが書き込める共通の共有ディレクトリに設定する必要があります。ports をローカルでビルドする場合には、各マシンは `WRKDIRPREFIX` を自分のマシンのビルドディレクトリに設定しなければなりません。また、ビルドシステムが packages をビルドしてビルドセットのコンピュータに配布するのであれば、`DISTDIR` と同じようにビルドシステム上の `PACKAGES` ディレクトリも設定してください。

パート IV. ネットワーク通信

FreeBSD は、高性能なネットワークサーバとして最も広く使用されているオペレーティングシステムの 1 つです。各章の内容は以下の通りです。

- シリアル通信
- PPP と PPP オーバイーサネット (PPPoE)
- 電子メール
- ネットワークサーバの運用
- ファイアウォール
- その他の高度なネットワークに関する話題

各章は、必要になった時に個別に参照できるように構成されています。 どの順番で読んでも構いませんし、ネットワーク環境で FreeBSD を使うのに、すべてを読み通す必要がある、というわけでもありません。

目次

19. シリアル通信 601
19.1. この章では 601
19.2. はじめに 601
19.3. シリアル端末 607
19.4. ダイアルインサービス 612
19.5. ダイアルアウトサービス 622
19.6. シリアルコンソールの設定 626
20. PPP と SLIP 637
20.1. この章では 637
20.2. ユーザ ppp の利用 637
20.3. カーネル PPP の利用 653
20.4. PPP オーバイーサネット (PPPoE) の利用 662
20.5. SLIP の利用 663
21. 電子メール 677
21.1. この章では 677
21.2. 電子メールを使う 678
21.3. sendmail の設定 679
21.4. MTA の変更 682
21.5. トラブルシュート 686
21.6. 先進的なトピックス 689
21.7. UUCP とともに SMTP を使う 692
21.8. ダイアルアップ接続でメールを使う 694
21.9. SMTP 認証 696
22. 高度なネットワーク 699
22.1. この章では 699
22.2. ゲートウェイと経路 699
22.3. 無線ネットワーク 707
22.4. Bluetooth 714
22.5. ブリッジ 724
22.6. NFS 727
22.7. ディスクレス稼働 734
22.8. ISDN 742
22.9. NIS/YP 747
22.10. DHCP 767
22.11. DNS 773
22.12. NTP 788
22.13. ネットワークアドレス変換 (NAT) 791
22.14. inetd「スーパサーバ」 795
22.15. パラレルライン IP (PLIP) 800
22.16. IPv6 803

第19章 シリアル通信

19.1. この章では

Unix は現在に至るまで、常にシリアル通信機能をサポートしていました。 実際、本当に初期の Unix マシンは、ユーザとの入出力にシリアル通信を使っていました。 10 文字毎秒のシリアルプリンタ、 キーボードから構成された「端末(terminal)」 が広く使われていた当時とは、 何もかもがすっかり変わっています。この章では、FreeBSD でシリアル通信を行なういくつかの方法について説明しています。

この章を読むと、以下のことがわかります。

- FreeBSD システムへの端末の接続方法
- リモートホストへダイヤルするためのモデムの使い方
- リモートのユーザがモデムでシステムにログインできるようにする方法
- シリアルコンソールからのシステム起動方法

この章を読む前に、以下のことを行っておくべきです。

- 新しいカーネルを構成してインストールする方法を覚える (9章FreeBSD カーネルのコンフィグレーション)。
- Unix のパーミッションとプロセスについて理解する (4章UNIX の基礎知識)。
- FreeBSD で使おうとしているシリアルハードウェア (モデムまたはマルチポートカード) のテクニカルマニュアルを読めるようにする。

19.2. はじめに

19.2.1. 用語解説

bps

Bits per Second の略で、データの転送速度を表す単位。

DTE

Data Terminal Equipment の略。たとえばコンピュータ本体のこと

DCE

Data Communications Equipment の略で、具体的にはモデムのこと。

RS-232

EIA (米電気産業協会) のハードウェアシリアル通信の標準規格

通信におけるデータ転送速度に関して、 このセクションでは 「ボー」 (baud) という用語は使いません。 ボーというのは一定時間に生じうる電気的状態の変化の数を表すにすぎず、「bps」 (bits per second) という単位の方が正しいからです (少なくとも、こういう表現をしておけば、意地の悪い人に怒られることもないのではないかと思います)。

19.2.2. ケーブルとポート

モデムまたはシリアル端末を FreeBSD システムに接続するためには、コンピュータ上のシリアルポートと、シリアルデバイスに接続する適切なケーブルが必要です。ハードウェアとそれが必要とするケーブルについてよく理解しているなら、この節は飛ばしても問題ありません。

19.2.2.1. ケーブル

シリアルケーブルにはさまざまな種類があります。 我々の目的にあうもっとも一般的な 2 種類は、ヌルモデムケーブル[1] と、スタンダード (ストレート) RS-232 ケーブルです ハードウェアの説明文書に必要なケーブルの種類が記載されているはずです。

19.2.2.1.1. ヌルモデムケーブル

ヌルモデムケーブル (またはリバースケーブルあるいはクロ スケーブル) は、たとえば「signal ground」 信号のように、いくつかの信 号はそのまま通しますが、 他の信号は途中で入れ替えて通します。たとえば、「send data」 信号のピンは、反対側のコネクタの「receive data」 信号の ピンと繋がっています。

自分で使うケーブルは自分で作りたいということであれば、 端末で使うヌルモデムケーブルを作成できます。この表では、RS-232C の信号線の名前と、DB-25 コネクタ上のピンの番 号を示しています。

Signal	Pin #		Pin #	Signal
TxD	2	connects to	3	RxD
RxD	3	connects to	2	TxD
DTR	20	connects to	6	DSR
DSR	6	connects to	20	DTR
SG	7	connects to	7	SG
DCD	8	connects to	4	RTS
RTS	4		5	CTS
CTS	5	connects to	8	DCD

[1] 訳注: リバースケーブル、もしくはクロスケーブルと呼ばれることもあります。

注記

DCD と RST では、コネクタ内部でピン4を5に接続し、そして逆側のコネクタのピン8と接続します。

19.2.2.1.2. スタンダード RS-232C ケーブル

スタンダードシリアルケーブル (またはストレートケーブル) の場合は、すべての RS-232C 信号をそのまま通します。つまり、片方の「send data」信号のピンは、逆側の「send data」信号のピンと繋がっています。モデムを FreeBSD に接続するときや、一部の端末を接続するときにこのタイプの ケーブルを使用します。

19.2.2.2. ポート

シリアルポートは、FreeBSDが動作しているホスト コンピュータと端 末の間でデータのやりとりを行うために用いるデバイスです。 ここでは、現在存在するポートの種類と FreeBSD でのポートのアクセス方法について解 説します。

19.2.2.2.1. ポートの種類

シリアルポートには何種類かのものがあります。ケーブルを購 入したり自作したりする前に、そのケーブルのコネクタの形状が端末および FreeBSD システムのポートの形状と一致していることを 確認してください。

ほとんどの端末は DB25 ポートを搭載しています。FreeBSDが動作しているも のを含めて、PCは DB25 または DB9 ポートを搭載しています。マルチポート のシリアルカードの場合は、RJ-12 や RJ-45 のポートを搭載しているかもし れません。

利用されているポートの種類に関しては、ハードウェアについてきたドキュメントを参照してください。また、多くの場合、ポートの形状から判断することもできるでしょう。

19.2.2.2.2. ポートの名前

FreeBSDでは、`/dev` ディレクトリ内のエントリを介 してシリアルポートへのアクセスがおこなわれます。2種類の異なったエン トリがあります。

- 着信用のポートの名前は、`/dev/ttydN` (*N* は 0から始まるポート番号) となっています。一般に端末の接続には 着信用ポートを用います。着信用のポートでは、 シリアルラインのデータ キャリア検出 (DCD) 信号がオンになっている必要があります。
- 発信用のポートの名前は、`/dev/cuaaN` となっています。発信用のポートは普通モデムの接続に用い、端末の接続には 利用しません。ただ、ケーブルまたは端末がキャリア検出信号を使えない タイプのものの場合は、 発信用のポートを使うとよいでしょう。

たとえば、端末を一つ目のシリアルポート (MS-DOS でいうところの COM1) に接 続したとすると、/dev/ttyd0 がこの端末を指すことになります。また、二つ目のシリアルポート (COM2) ならば /dev/ttyd1 となり、以下この形式のデバイスエントリを使います。

19.2.3. カーネルの設定

デフォルトでは、FreeBSD は 4 つのシリアルポートに対応しています。MS-DOS の世界では、COM1, COM2, COM3 および COM4 と呼ばれています。FreeBSD では、現在のところ BocaBoard の 1008 や 2016 などの、「単純な」マルチポートシリアルインタフェースや、Digiboard や Stallion Technologies が製造しているよりインテリジェントなマルチポートカードにも対応しています。しかしながら、デフォルトのカーネルは、標準の COM ポートしか見ません。

搭載されているシリアルポートのいずれかを、カーネルが認識しているかどうか確認したい場合は、カーネルの起動時のメッセージを注意深く見るか、あるいは /sbin/dmesg コマンドを使って、起動時の出力メッセージを確認してください。特に、sio で始まるメッセージをよく見てください。

ヒント

以下のコマンドで sio という文字列を含むメッセージだけを表示できます。

```
# /sbin/dmesg | grep 'sio'
```

たとえば、シリアルポートを四つ持つシステムの場合は、 以下のようなシリアルポートに関するメッセージがカーネルによって表示されます。

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
sio2 at 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
sio3 at 0x2e8-0x2ef irq 9 on isa
sio3: type 16550A
```

もし、カーネルに正常に認識されないポートがある場合は、 おそらくカスタマイズした FreeBSD カーネルを構築する必要があるでしょう。 カーネルコンフィグレーションの詳細については 9章FreeBSD カーネルのコンフィグレーション をご覧ください。

カーネルコンフィグレーションの該当するデバイス行は、次のようになります。

```
device  sio0 at isa? port "IO_COM1" tty irq 4 vector siointr
device  sio1 at isa? port "IO_COM2" tty irq 3 vector siointr
```

```
device  sio2 at isa? port "IO_COM3" tty irq 5 vector siointr
device  sio3 at isa? port "IO_COM4" tty irq 9 vector siointr
```

システムに搭載されていないデバイスに関する記述は、 コメントアウトまたは削除してしまってかまいません。 sio(4) のマニュアルを見て、 マルチポートのボードのためのコンフィグレーションファイルの記述の仕方を確認してください。 デバイスのフラグの指定方法がバージョンによって異なりますので、別のバージョンの FreeBSD で利用していたコンフィグレーションファイルを流用する場合には十分注意してください。

注記

なお、`port "IO_COM1"`, `IO_COM2`, `IO_COM3` および `IO_COM4` は、 それぞれのポートの一般的なアドレスである `0x3f8`, `0x2f8`, `0x3e8` および `0x2e8` を表します。また、割り込み番号 4, 3, 5 と 9 は、それぞれ `COM1:` から `COM4:` のポートで一般的に使用される IRQ です。また、ISA バスのコンピュータの場合、一般的なシリアルポートは複数のポートで一つの IRQ を共有することが できませんので注意が必要です (マルチポートのシリアルボードの場合は、複数の 16550A ベースのポートで一つまたは二つの IRQ を共有するための機構を備えています)。

19.2.4. デバイススペシャルファイル

カーネルに組み込まれているほとんどのデバイスは、 `/dev` ディレクトリにある、「デバイススペシャルファイル」を介してアクセスされます。 `sio` デバイスの場合は、着信用の `/dev/ttydN` および、発信用の `/dev/cuaaN` が利用されます。さらに、FreeBSD は、初期化デバイス (`/dev/ttyidN` と `/dev/cuai0N`) およびロッキングデバイス (`/dev/ttyldN` と `/dev/cual0N`) も用意しています。 初期化デバイスは、通信ポートがオープンされる度に、そのポートの初期設定を行うために使われます。たとえば、 `RTS/CTS` によるフロー制御を行うモデムが接続されている場合の `crtscts` などのパラメータの初期化が行われます。 ロッキングデバイスは、ポートの設定をロックし、 他のユーザやプログラムにこれらを変更されることのないようにするために利用されます。通信ポートの設定、デバイスのロックと初期化および設定の変更に関しては、それぞれ termios(4), sio(4) と stty(1) のマニュアルをご覧ください。

19.2.4.1. デバイススペシャルファイルの作成

注記

FreeBSD 5.0 には、必要に応じてデバイスノードを自動的に作成する `devfs` ファイルシステムがあります。`devfs` が有効になって

いるバージョンの FreeBSD を動かしているなら、この節は飛ばしてかまいません。

デバイススペシャルファイルの管理は、ディレクトリ /dev にあるシェルスクリプト MAKEDEV で行います。MAKEDEV を使って、COM1 (ポート 0) をダイアルアップのポートとして利用するための デバイススペシャルファイルを作るには、/dev に cd してから、MAKEDEV ttyd0 と実行してください。同様に、MAKEDEV ttyd1 とすることで、COM2 (ポート 1) 用のデバイススペシャルファイルを作成できます。

MAKEDEV は、/dev/ttydN のデバイススペシャルファイルだけでなく、/dev/cuaaN, /dev/cuaiaN , /dev/cualaN , /dev/ttyldN および /dev/ttyidN ノードも作成します。

デバイススペシャルファイルの作成後、 これらのファイルの許可属性が適切に設定されていて、 これらのデバイスを利用してもよいユーザのみが読み書きできるようになっていることを確認してください (特に /dev/cua* の許可属性には注意を払ってください)。この確認を怠ると、 一般のユーザがあなたのモデムを使うことができるようなことになりかねません。デフォルトの /dev/cua* の許可属性は、以下のようになっていて、たいていの場合適切なものだと思います。

```
crw-rw----    1 uucp     dialer    28, 129 Feb 15 14:38 /dev/cuaa1
crw-rw----    1 uucp     dialer    28, 161 Feb 15 14:38 /dev/cuaia1
crw-rw----    1 uucp     dialer    28, 193 Feb 15 14:38 /dev/cuala1
```

上の設定では、ユーザ uucp と、グループ dialer に属するユーザが発信用のデバイスを利用できます。

19.2.5. シリアルポートの設定

ttydN (または cuaaN) デバイスは、アプリケーション上でシリアルポートをオープンする時に使用する、標準的なデバイスです。プロセスがデバイスをオープンする際、端末 I/O 設定のデフォルトセットが適用されます。これらの設定内容は、次のコマンドで確認することができます。

```
# stty -a -f /dev/ttyd1
```

このデバイスの設定を変更した場合、 その設定はデバイスがクローズされるまで有効です。 デバイスが再びオープンされる時、デフォルトの設定値に戻ります。 デフォルトの設定を変更するためには、「初期状態」 を設定したいデバイスをオープンして調節できます。たとえば、ttyd5 というデバイスに対して、デフォルトで CLOCAL モード, 8 bits, XON/XOFF フロー制御を設定したい場合は、次のように入力してください。

```
# stty -f /dev/ttyid5 clocal cs8 ixon ixoff
```

システム全体のシリアルデバイス初期化は `/etc/rc.serial` で制御されています。このファイルは、シリアルデバイスのデフォルトの設定を決めます。

また、「ロック状態」のデバイスに調節を加えることで、アプリケーションがある種の設定を変更してしまうことを防げます。たとえば、`ttyd5` の速度を 57600 bps に固定したい場合には、次のように入力してください。

```
# stty -f /dev/ttyld5 57600
```

これで、`ttyd5` をオープンして、シリアルポートの転送スピードを変更しようとするアプリケーションは 57600 bps で頭打ちになります。

本来、初期状態やロックされているデバイスに書き込めるのは `root` アカウントだけにすべきです。

19.3. シリアル端末

##: Sean Kelly.

訳: 中根 雅文 **<max@FreeBSD.org >**

シリアル端末を利用することで、コンピュータのコンソールのそばにいないときや、手近にネットワーク接続されているコンピュータがないときでも、FreeBSD の機能を便利に、かつ安価に利用することができます。ここでは、FreeBSD にシリアル端末を接続する方法を解説します。

19.3.1. 端末の種類と利用方法

もともと Unix システムにはコンソールがありませんでした。ユーザはコンピュータのシリアルポートに接続された端末からログインしてプログラムを利用していました。ちょうどモデムと通信ソフトを使ってリモートのコンピュータにログインし、テキストベースのプログラムを利用するのとよく似ています。

最近の PC は、高品質の画像を表示できるコンソールを搭載していますが、ほとんどすべての Unix 系 OS には未だにシリアルポートを使ってログインするための機能があり、FreeBSD でもこの機能がサポートされています。現在使用されていないシリアルポートに端末を接続することでシステムにログインし、通常はコンソールや X ウィンドウシステムの `xterm` のウィンドウ上で起動しているテキストベースのプログラムであれば何でも利用できます。

職場での利用ということで考えるならば、FreeBSD が動作しているコンピュータに接続された何台ものシリアル端末を各社員の机に配置するというようなことが可能です。 また、家庭での利用方法としては、余っている古い IBM PC や Macintosh を FreeBSD が動いているパワフルなコンピュータの端末として利用できます。普通ならシングルユーザのコンピュータを、パワフルなマルチユーザのシステムに変えることができるのです。

FreeBSD では、以下に挙げる 3 種類の端末が利用できます。

- ダム (dumb) 端末
- PCを利用した端末
- X 端末

以下は、それぞれについての解説です。

19.3.1.1. ダム端末

ダム端末は、 シリアルライン経由でのコンピュータとの接続専用のハードウェアです。 ダム端末は、テキストの送受信および表示ができる程度の計算能力しかもっていないので、「dumb」(間抜け) というように呼ばれています。この端末上でプログラムを実行することはできません。 テキストエディタ、コンパイラ、E-mail、 ゲームなどなどのプログラムを実行するのは、ダム端末を接続しているコンピュータの方です。

Digital Equipment 社の VT-100 や、Wyse 社の WY-75 を初めとして、多くのメーカが何百種類ものダム端末を作っています。 ほとんどどんな種類のダム端末でも FreeBSD に接続して使用できます。さらに、 高性能の端末の中には画像を取り扱えるものもありますが、限られた数のソフトウェアパッケージしかこういった機能には対応していません。

ダム端末は、 X ウィンドウシステムで提供されるようなグラフィックアプリケーションを必要としない職場で広く用いられています。

19.3.1.2. PC を端末として利用する

ダム端末 がテキストの表示および送受信の機能をそなえただけのものならば、 言うまでもなく、どんな PC もダム端末になり得ます。必要なものは適切なケーブルと、その PC の上で動作する端末エミュレーション を行うソフトウェアのみです。

このような環境は、家庭においてよく利用されます。 たとえば、あなたの同居人が FreeBSD のコンソールを専有している時などに、 あまりパワーのないコンピュータを FreeBSD システムにシリアル端末として接続し、その端末上でテキストだけを用いる作業をおこなうことができます。

19.3.1.3. X 端末

X 端末は、 既存のものの中で最も洗練された種類の端末といえます。 X 端末は、たいていの場合シリアルポートではなく、 イーサネットのようなネットワークを利用した接続をおこないます。また、アプリケーションの利用においても、 テキストベースのものだけでなく、X アプリケーションの利用が可能です。

ここでは、参考までに端末について紹介しただけで、X 端末の設定や利用についての解説は おこないません。

19.3.2. 設定

ここでは、端末からのログインを可能にするために必要な FreeBSD 側の設定について解説します。既に端末を接続するポートが利用できるように kernel の設定をおこない、端末が接続されているものと考えて、解説を進め ます。

13章FreeBSD の起動のプロセス で述べたように `init` プロセスは、システム起動時にすべてのプロセス管理や初期化をおこなっています。`init` が行っている仕事の一つは、`/etc/ttys` ファイルを読んで、利用可能な端末上で `getty` プロセスを起動することです。`getty` プロセスは、ログイン名を読み込み `login` プログラムを起動します。

したがって、FreeBSD の端末を設定するには、`root` で次の手順を踏まなければなりません。

1. 端末を接続するポートの `/dev` のエントリが含ま れている行がまだ存在しなければ、これを `/etc/ttys` に追加してく ださい。

2. `/usr/libexec/getty` が対象となるポートに対して 実行されるように指定してください。また、`/etc/gettytab` ファイ ル内の適切な *`getty`* タイプのエントリを指定してください。

3. デフォルトのターミナルタイプを指定してください。

4. 対象となるポートを「on」に設定してください。

5. そのポートが「secure」であるかどうかを指定してください。

6. `init` に `/etc/ttys` を読み込みなおさせてく ださい。

また、必要に応じて `/etc/gettytab` を変更し、上の 2で使用する *`getty`* のエントリを追加してください。 この章ではこの方法については特に解説しませんので、gettytab(5) および getty(8) のマニュアルをご覧ください。

19.3.2.1. `/etc/ttys` へのエントリの追加

`/etc/ttys` には、FreeBSDシステム上のログインを許可するすべての ポートを記述します。たとえば、一つ目の仮想コンソール `ttyv0` のエン トリもこのファイルにあります。このエントリのおかげで、コンソールからの ログインが可能になっています。このファイルには、他の仮想コンソール、シ リアルポートおよび仮想端末のエントリも含まれています。端末を接続する場合は、そのポートの `/dev` のエントリを、`/dev` の部分を省略して記述します (たとえば `/dev/ttyv0` については、`ttyv0` として記述します)。

FreeBSD のデフォルトのインストール状態では、`ttyd0` から `ttyd3` までの、初めの 4 つのシリアルポートに対応した `/etc/ttys` ファイルが置かれています。これらのポートのいずれかに端末を接続する場合は、新たにエントリを追加する必要はありません。

例19.1 端末の項目を /etc/ttys に追加する

システムに 2 台の端末、Wyse-50 と、VT-100 端末をエミュレートしている Procomm 端末ソフトウェアを動かしている古い 286 IBM PC をシステムに接続しようとしていると考えてください。Wyse は 2 番目のシリアルポートに、286 は 6 番目のシリアルポート (マルチポートシリアルカード上のポート) に接続します。/etc/ttys 内の対応する項目は次のようになります。

```
ttyd1❶  "/usr/libexec/getty std.38400"❷  wy50❸  on❹  ↺
insecure❺
ttyd5   "/usr/libexec/getty std.19200"  vt100  on  insecure
```

❶ 最初のフィールドには、通常 /dev にある端末のスペシャルファイル名を指定します。

❷ 2 番目のフィールドは、 この回線に対して実行するコマンドで、通常は getty(8) です。getty は、回線を初期化して開き、速度を設定して、ユーザ名を入力するプロンプトを出して login(1) プログラムを実行します。

getty プログラムは、コマンドラインから (省略可能な) パラメータ *getty* タイプを受け取ります。*getty* タイプは、bps レートやパリティのような端末回線の特性を示します。getty プログラムは、これらの特性を /etc/gettytab ファイルから読み込みます。

/etc/gettytab ファイルには、端末回線について新旧多くの項目があります。ほとんどの場合、std で始まる項目は、ケーブルで接続された端末に働きます。これらの項目はパリティを無視します。110 から 115200 までの間の bps レートそれぞれに対して一つ std 項目があります。もちろん、このファイルに独自の項目を加えてもかまいません。gettytab(5) のマニュアルに詳しい情報が載っています。

/etc/ttys ファイルに *getty* タイプを設定する時は、端末の通信設定が対応していることを確かめましょう。

たとえば、Wyse-50 はパリティなしで、38400 bps で接続します。286 PC はパリティなしで、19200 bps で接続します。

❸ 第 3 フィールドは、その tty 回線に通常つながる端末の種別です。ダイアルアップポートでは、実際、 ユーザがどんな種類の端末やソフトウェアで接続してくることもありうるので、このフィールドには unknown または dialup がよく使われています。ケーブルで配線された端末については、端末種別は変わりませんので、termcap(5) データベースファイルから実際の端末種別を、このフィールドに記入できます。

我々の例では、Wyse-50 には実際の端末種別を使っていますが、Procomm を動かしている 286 PC は、VT-100 をエミュレートするように設定します。

❹ 4 番目のフィールドは、ポートを有効にすべきかどうかを指定します。ここに `on` と記入すると、`init` プロセスが 2 番目のフィールドに記載されているプログラム、`getty` を起動します。このフィールドを `off` にすると、`getty` は動かず、そのポートからはログインできません。

❺ 最後のフィールドは、そのポートが安全かどうか指定します。あるポートが安全だということは、そのポートから `root` (またはその他の UID が 0 の) アカウントのログインを許可してよいと信頼しているということです。安全でないポートからは、`root` のログインは許可されません。安全でないポートでは、ユーザは特権を持たないアカウントでログインした後に、su(1) や類似の仕組みを使ってスーパユーザ特権を獲得します。

鍵のかかる部屋にある端末であっても、「insecure」 にしておくことが強く推奨されます。スーパユーザ特権が必要なら、ログインしてから `su` を使うのは十分簡単です。

19.3.2.2. `init` にファイル `/etc/ttys` の再読み込みをさせる

必要な変更を `/etc/ttys` ファイルに加えたら、SIGHUP (ハングアップ) シグナルを `init` プロセスに送って設定ファイルを強制的に再読み込みさせます。たとえば

```
# kill -HUP 1
```

注記

`init` は、システムで最初に起動するプロセスなので、PID は常に 1 です。

すべての設定が正しくおこなわれ、すべてのケーブルがただしく接続されていて、かつ端末の電源が入っていれば、この時点で各端末で `getty` プロセスが動いていて、ログインプロンプトが表示されているはずです。

19.3.3. 接続のトラブルシューティング

細心の注意を払って設定をおこなっても、ときには端末の接続がう まくいかない場合があるでしょう。以下に、よく見られる問題とその解決方法 を示します。

問: ログインプロンプトが表示されない

答: 端末の電源が接続され、 スイッチが入っていることを確認してください。もし、PCを端末として利用している場合は、通信ソフトが適切なシリアルポー トを利用する設定になっているかどうか確かめてください。

ケーブルがしっかりと端末と FreeBSDが動作しているコンピュータの両方に接続され ていることを確認してください。また、 正しい種類のケーブルを利用しているか確かめてください。

端末と FreeBSD の間の通信速度とパリティの設定が一致していることを確認 してください。出力をモニタに表示するタイプの端末の場合は、モニタのコントラストと明るさの設定を確認してください。また、出力が印刷 されるタイプの端末の場合は、紙とインクが十分にあるかどうかを確かめてく ださい。

`getty` が動いていて、端末を認識していることを確認してください。たとえば、動作中の `getty` プロセスの一覧を `ps` で取得するには、以下のように入力してください。

```
# ps -axww|grep getty
```

その端末に対応する項目が表示されるはずです。 たとえば、以下の表示例は、`getty` は 2 番目のシリアルポート (`ttyd1`) に対して `/etc/gettytab` 中の `std.38400` エントリを使って動作しているということを示しています。

```
22189  d1  Is+    0:00.03 /usr/libexec/getty std.38400 ttyd1
```

もし、`getty` プロセスが一つも動いていないようであれば、`/etc/ttys` の中で、そのポートを利用可能にする設定をしたかどうか確かめてください。 また、`ttys` ファイルを変更したら、`kill -HUP 1` を実行するのを忘れないでください。

問: ログインプロンプトの代わりにゴミが表示される

答: 端末と FreeBSD の間の通信速度およびパリティの設定が一致していることを確かめてください。また、`getty` プロセスの情報を調べて、適切な *`getty`* のタイプが使用されていることを確認してください。間違った getty タイプが使用されている場合は、`/etc/ttys` を修正し てから、`kill -HUP 1` を実行してください。

問: 文字が重複して表示される、入力したパスワードが表示される

答: 端末または通信ソフトの設定で、「半二重 (half duplex)」あるいは「ローカ ルエコー」となっているところを、「全二重 (full duplex)」に変更してください。

19.4. ダイアルインサービス

##: Guy Helmer.
##: Sean Kelly.

訳: 中根 雅文 <max@FreeBSD.org >. 6 September 1996.

FreeBSD システムをダイアルインサービス用に設定することは、端末の代わりにモデムを扱うこと以外は、端末の接続によく似ています。

19.4.1. 外づけモデムと内蔵モデムについて

ダイアルアップのサービスに関していえば、外づけのモデムの方が適している ようです。これは、多くの外づけのモデムは設定を不揮発ラムに書き込んで半 永久的に保存することができますし、また RS-232 に関する重要な情報を知る ための点滅するライトによるインディケータが 搭載されているからです。点滅 するライトは、 システムを見に来た訪問者に強い印象を与えるという効果だけ でなく、モデムが適切に動作しているかどうかを知るためにも 有効です。

一方、たいていの内蔵型のモデムには 不揮発性ラムが搭載されていないため、ディップスイッチの変更以外に設定を保存する方法がありません。また、も しインディケータがついていても、おそらくコンピュータのケース カバーが 外されていなければその状態を確認するのは 難しいでしょう。

19.4.1.1. モデムとケーブル

外付けモデムを使用しているなら、それにあったケーブルが必要です。通常の信号が全て接続されている限り、標準的な RS-232C ケーブルで十分でしょう。

- Transmitted Data (SD)
- Received Data (RD)
- Request to Send (RTS)
- Clear to Send (CTS)
- Data Set Ready (DSR)
- Data Terminal Ready (DTR)
- Carrier Detect (CD)
- Signal Ground (SG)

FreeBSD で 2400bps 以上の転送速度を利用する場合には、フロー制御のため に RTS 信号と CTS 信号が必要です。また、接続の確立と回線の切 断を検出するために CD 信号を利用します。さらに、DTR 信号を使っ て回線切断後のモデムのリセットを行います。ケーブルの中には、総ての必要 な信号線が接続されていないものもありますの

で、たとえば、回線切断後でも ログイン セッションが残ってしまうといった問題が発生した場合などには、ケーブルに問題がある可能性もあります。

FreeBSD も他の Unix 系 OS と同様、回線の接続およ び切断の検出や回線の切断および回線切断後の モデムの初期化にハードウェア シグナルを利用します。FreeBSD は、モデムに対するコマンドの送信やモデ ムの状態の監視を行いません。パソコンで運用されている BBS への接続に慣 れている方にとっては、ちょっとめんどうかもしれませんね。

19.4.2. シリアル インタフェースについて

FreeBSD では、NS8250-、NS16450-、NS16550- および NS16550A- に基づ いた EIA RS-232C (CCITT V.24) 規格のシリアル インタフェースをサポート しています。8250 および 16450 ベースのディバイスには1文字のキャラクタ バッファが搭載されています。また、16550 系のディバイスには、16文字分 のバッファが搭載されていて、はるかによいパフォーマンスを得られます (ただし、無印の 16550 では、バグがあって 16 文字バッファが利用できませ んので、可能であれば 16550A 系のディバイスを利用してください)。1文字 のバッファの物は、16550 系のものと比べて OS にかける負荷が大きいので、16550A 系ディバイスの利用を強く推奨します。多数のシリアル ポートを利 用する場合や、負荷の高いシステムにおいては、16550A 系ディバイスを使う ことで、エラー発生率を低く押さえることができます。

19.4.3. 概要

端末に関しては、 ダイアルイン接続に割り当てられたそれぞれのシリアルポートに対して、`init` が `getty` を起動します。たとえば、モデムが `/dev/ttyd0` に割り当てられていたら、`ps ax` コマンドを実行すると、以下のような出力が得られるはずです。

```
 4850 ??  I      0:00.09 /usr/libexec/getty V19200 ttyd0
```

ユーザがモデムに電話をかけ、モデム同士が接続されると、 モデムの CD (Carrier Detect) が検出されます。その結果、 kernel がキャリア信号を検出して、`getty` によるポートのオープンの処理が終了します。`getty` は、`login:` プロンプトを指定されている初期回線速度で送信します。`getty` は、正常に文字列を受信できるかどうか監視し、通常の設定では、もし異常な文字列を検出した場合 (理由としては、`getty` の速度とモデ ムの接続速度が異なっているような場合が考えられます)、正常に文字列が 受信できるまで、`getty` は速度を変え続けます。

ユーザがログイン名を入力すると、`getty` は `/usr/bin/login` を起動して、パスワードの入力を要求し、その 後ユーザのシェルを起動します。

19.4.4. 設定ファイル

FreeBSD のシステムへのダイアルアップによるアクセスを実現するために編集が必要と思われる設定ファイルが、`/etc` ディレクトリに 3 つあります。まず、`/etc/gettytab`

には、/usr/libexec/getty デーモンの設定を記述します。つぎに、/etc/ttys に保存されている情報から、/sbin/init はどの tty デバイスに対して getty のプロセスを実行するべきか判断します。最後に、/etc/rc.serial スクリプトに、シリアルポートの初期化のためのコマンドを記述することができます。

Unix にダイアルアップモデムを接続する方法には、二つの考え方があります。一つの方法は、ダイアルインしてくるユーザの接続速度に関係なく、常にモデムとローカルのコンピュータの RS-232 インタフェースの接続速度を一定に保つように設定する方法です。この設定の長所は、ユーザがダイアルインして接続されると、即座にシステムからのログインプロンプトが送信されるということです。 短所は、システムが実際のモデム間の速度を知ることができないために、Emacs のようなフルスクリーンのプログラムが、端末との接続速度が遅い場合でも、そのような場合に効果的な方法で画面出力を行わない点です。

もう一つは、モデムの RS-232 インタフェースとコンピュータの接続速度を、モデム間の接続速度に応じて変化させるような設定です。たとえば、 モデム間 の接続が V.32bis (14.4 Kbps) ならば、 モデムとコンピュータの間の接続を 19.2 Kbps とし、 モデム間の接続が 2400 bps の時には、モデムとコンピュータ間も 2400 bps で接続するような設定をします。この場合、 getty は、モデムが返すリザルトコードからモデムとコンピュータの接続速度を認識することができませんので、getty は、まず初期速度で login: という文字列を送信して、それに対する応答の文字列を監視します。 ここで、ユーザ側の端末に無意味な文字列が表示された場合、ユーザは意味のある文字列を受信するまで Enter キーを繰り返し押さなければならないということを知っていると仮定しています。もし接続速度が間違っている場合、getty は、ユーザから送られた文字を無意味な文字列として扱い、 次の速度を試します。そして、ここで再度 login: プロンプトを送信します。 この一連の動作が異常な回数繰り返されることも考えられますが、普通は 1 度か 2 度のキー入力があれば、 ユーザはまともなプロンプトを受信できます。このログインの動作が前者の固定速度による方法に比べて美しくないのは明らかですが、この方法では、低速度で接続しているユーザに対するフルスクリーンのプログラムからのレスポンスが改善されます。

このセクションでは、両方の設定方法について解説しますが、 どちらかというとモデム間の速度に応じて RS-232 インタフェースの速度が変化するような 設定の方に偏った説明になってしまうと思います。

19.4.4.1. /etc/gettytab

/etc/gettytab は、getty(8) の設定ファイルで、termcap(5) と同様の形式で記述されます。ファイルのフォーマットや定 義できる機能についての詳細については、gettytab(5) のマニュアルを ご覧ください。

19.4.4.1.1. 固定速度の設定

モデムとコンピュータ間の通信速度を固定して使う場合、 おそらく /etc/gettytab に特に変更を加える必要はないはずです。

19.4.4.1.2. 可変速度の設定

getty が利用するモデムとコンピュータの接続速度に関する情報を /etc/gettytab に記述する必要があります。もし、2400 bps のモ デムをお使いになるのであれば、既存の D2400 のエントリがそのまま利 用できるでしょう。

```
#
# Fast dialup terminals, 2400/1200/300 rotary (can start either way)
#
D2400|d2400|Fast-Dial-2400:\
        :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200:\
        :nx=D300:tc=1200-baud:
5|D300|Fast-Dial-300:\
        :nx=D2400:tc=300-baud:
```

高速モデムをお使いの場合は、おそらく /etc/gettytab に新たなエントリを追加する必要があります。 以下の例は、14.4 Kbps のモデムを、 最大インタフェース速度を 19.2 Kbps として利用するためのエントリです。

```
#
# Additions for a V.32bis Modem
#
um|V300|High Speed Modem at 300,8-bit:\
        :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
        :nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:\
        :nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:\
        :nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
        :nx=V9600:tc=std.19200:
```

上記の例を利用した場合、パリティなし、8ビットの接続が行われます。

上記の例では、まず 19.2 Kbps (V.32bis) によるモデムとコンピュータ間の接続を試み、続いて 9600 bps (V.32)、2400 bps、1200 bps、300 bpsと順に試み、再び 19.2 Kbps による接続を試みるという循環に入ります。 この接続速度の循環は、nx=(「next table」) の機能で実現されています。また、各行はそれぞれ tc=(「table continuation」) の機能を使って、その他の接続速度に依存した「標準的な」設定を取り込んでいます。

もし、お使いのモデムが 28.8 Kbps であったり、14.4 Kbps の圧縮転送の機能を有効に利用したい場合は、19.2 Kbps よりも速い速度を利用するように設定する必要があります。以下に 57.6 Kbps から接続を試みる gettytab の設定例を示しておきます。

```
#
# Additions for a V.32bis or V.34 Modem
# Starting at 57.6 Kbps
```

```
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
        :nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
        :nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
        :nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
        :nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
        :nx=VH9600:tc=std.57600:
```

もし、お使いの CPU が低速のものであったり、CPU に対する負荷が高い場合で、16550A 系のシリアルポートをお使いでない場合、57.6 Kbps の接続において、sio の「silo」エラーが発生するかもしれません。

19.4.4.2. /etc/ttys

/etc/ttys ファイルの設定は、例19.1「端末の項目を /etc/ttys に追加する」で扱われています。モデムの設定も似たようなものですが、getty に異なる引数を渡して、異なる端末種別を指定しなければなりません。 固定速度および可変速度両方に共通する形式は次のようになります。

```
ttyd0   "/usr/libexec/getty XXX"   dialup on
```

1 番目の項目は、このエントリで対象とするデバイススペシャルファイルです。上の例では ttyd0 として、/dev/ttyd0 を getty に監視させることを表しています。2 番目の項目 "/usr/libexec/getty xxx" (*xxx* は初期段階で使われる gettytab のエントリに置き換えてください) が、init がこのディバイスに対して起動するプロセスです。3 番目の dialup は、デフォルトのターミナルタイプです。4 番目の on は、この行が有効であることを init に対して示しています。5 番目の項目に secure を指定することもできますが、これは、たとえばシステムのコンソールのように、物理的に安全な端末に対してのみ指定するようにしてください。

デフォルトのターミナルタイプ (上記の例では dialup) は、ローカルのユーザの好みによって異なってきます。 ユーザがログインスクリプトをカスタマイズして、ターミナルタイプが dialup の時には自動的に他のターミナルタイプを設定できるように、 ダイアルアップのポートのデフォルトのターミナルタイプには dialup が伝統的に用いられています。しかし、筆者のサイトでは、ほとんどのユーザが VT102 エミュレーションを使っているので、ダイアルアップのポートのデフォルトターミナルタイプとして vt102 を指定しています。

/etc/ttys の修正がすんだら、以下のようなコマンドを使って init プロセスに HUP シグナルを送り、/etc/ttys を読み込み直させてください。

```
# kill -HUP 1
```

ただ、もし初めてシステムを設定しているのであれば、モデムが適切に設定されて接続されるまでは、init に対してシグナルを送らない方がいいかもしれません。

19.4.4.2.1. 固定速度の設定

速度を固定する設定では、/etc/ttys の中で、getty に対し て固定速度のエントリを指定する必要があります。たとえば、以下の例はポートのスピードが 19.2 Kbps に固定されたモデムのための ttys のエントリです。

```
ttyd0   "/usr/libexec/getty std.19200"   dialup on
```

モデムが異なる速度で固定されている場合は、std.19200 のかわりに std.speed を適切な値に置き換えたものにしてください。 /etc/gettytab に挙がっている適切な種類を使うようにしてください。

19.4.4.2.2. 可変速度の設定

可変速度の設定では、ttys のエントリが、/etc/gettytab の中の適切な「自動速度調整」の初期設定のエントリを参照していなければな りません。たとえば、もし前述の 19.2 Kbps から接続を試みる可変速度の設定例 (V19200 の gettytab エントリ)をそのまま ttys に追加したのであれば、ttys エントリは以下のようになります。

```
ttyd0   "/usr/libexec/getty V19200"   dialup on
```

19.4.4.3. /etc/rc.serial

V.32、V.32bis または V.34 モデムのような高速モデムを利用する場合、ハードウェア (RTS/CTS) フロー制御を行う必要があります。FreeBSD kernel のモデムポートにハードウェアフロー制御のフラグを設定するための stty コマンドを、/etc/rc.serial に記述できます。

たとえば、シリアルポート 1 番 (COM2) のダイヤルインおよびダイヤルアウト初期化デバイスに termios フラグ crtscts を設定するには、次の行を /etc/rc.serial に追加するとよいでしょう。

```
# Serial port initial configuration
stty -f /dev/ttyid1 crtscts
stty -f /dev/cuai01 crtscts
```

19.4.5. モデムの設定

もし、あなたのモデムがパラメータを不揮発ラムに 保存できるタイプならば、MS-DOS 上の Telix や FreeBSD 上の tip などのような通信プログラム を使って、パラメータを設定してください。getty が利用する初期速度でモデムに接続して、以下の条件を満たすよ うに不揮発ラムの設定を変更してください。

- 接続時に CD 信号がオンになる

- 接続時に DTR がオンになり、DTR オフで回線を切断しモ デムをリセットする。
- 送信時フロー制御には CTS を利用。
- XON/XOFF によるフロー制御を行わない。
- 受信時のフロー制御は RTS を使用。
- Quiet mode (リザルト コードを返さない)
- コマンド エコーを返さない。

これらを実現するためのコマンドやディップ スイッチの設定に関しては、モ デムのマニュアルを参照してください。

以下に、USRobotics Sportster の 14,400 bps の外づけモデムの設定例を示 しておきます。

```
ATZ
AT&C1&D2&H1&I0&R2&W
```

ことのついでに、たとえば、V42.bis や MNP5 のデータ圧縮を使用するかど うかなどのモデムの他の設定について確認、調整しておくのもよいかもしれま せん。

さらに、USRobotics Sportster の 14,400 bps の外づけモデムでは、以下の ようなディップ スイッチの設定も必要です。他のモデムをお使いの方も、以 下の例を設定の参考にしてください。

- スイッチ 1: UP — DTR 標準
- スイッチ 2: N/A (リザルトコードを単語形式にするか数値形式にするか)
- スイッチ 3: UP — リザルトコードを返さない
- スイッチ 4: DOWN — コマンドエコーを返さない
- スイッチ 5: UP — 自動着信
- スイッチ 6: UP — CD 標準
- スイッチ 7: UP — 不揮発ラムからデフォルト値をロードする
- スイッチ 8: N/A (Smart Mode/Dumb Mode)

リザルト コードを返さないように設定しておかないと、`getty` が誤っ て `login:` プロンプトをコマンド モードのモデムに送信してしまった場 合に、モデムがこの入力をエコーしたり、この入力に対するリザルト コード を返してしまったりすることになります。この結果として、モデムと `getty` の間で延々と無意味なやりとりが続いてしまう可能性があります。

19.4.5.1. 固定速度の設定

固定速度の設定では、 モデムとコンピュータ間の通信速度をモデムとモデム間 の接続速度に関係なく、常に一定に保つように、 モデムを設定する必要があり ます。USRobotics Sportster の 14,400 bps 外づけモデムの場合、以下のコ マンドで、モデムとコンピュータ間の速度が、コマンド送信時の速度に固定さ れます。

```
ATZ
AT&B1&W
```

19.4.5.2. 可変速度の設定

可変速度の設定では、シリアル ポートの速度が、 着信速度に応じて変化する ように設定しなければいけません。USRobotics Sporster の 14,400 bps 外 づけモデムの場合、以下のコマンドで、エラー訂正機能を利用した通信の場合 は、コマンドを送信した時の通信速度にシリアル ポートの速度を固定し、エ ラー訂正機能を利用しない接続では、シリアル ポートの速度が変化するよう に設定されます。

```
ATZ
AT&B2&W
```

19.4.5.3. モデムの設定の確認

ほとんどの高速モデムには、 現在の設定をある程度人間にも理解できる形式に して表示させるコマンドがあります。USRobotics Sporster の 14,400 bps 外づけモデムの場合は、`ATI5` コマンドで、現在の不揮発ラムの設定を 表示することができます。さらに、ディップ スイッチの設定も含めた現在の 設定を確認するためには、`ATZ` コマンドを送信してから、`ATI4` コマンドを送信してください。

他のメーカーのモデムをお使いの場合は、 モデムのマニュアルで設定値の確認 方法を確認してください。

19.4.6. トラブルシューティング

以下の手順でダイアル アップ モデムの動作を確認することができます。

19.4.6.1. FreeBSD システムの動作確認

モデムを FreeBSD システムに接続し、 システムをブートします。あなたのモ デムにモデムの状態を確認するためのインジケータがあれば、 DTR のイ ンジケータの状態に注目してください。もし、 システムのコンソールに `login:` プロンプトが表示された時に、DTR のインジケータが点灯 すれば、FreeBSD が適切なポートに対して `getty` を起動し、モデムへ の着信を待っている状態であることを意味しています。

もし DTR のインジケータが点灯しない場合は、システムのコンソールから FreeBSD にログインして、`ps ax` を実行し、FreeBSD が適切なポートに対して`getty` プロセスを

起動しようとしているのかどうか確認してください。プロセスに関する情報の中に、以下のような行が表示されるはずです。

```
  114 ??  I      0:00.10 /usr/libexec/getty V19200 ttyd0
  115 ??  I      0:00.10 /usr/libexec/getty V19200 ttyd1
```

モデムにまだ着信がない状態の時に、 以下のように上とは異なる出力があった 場合、getty は既にモデム ポートのオープンを終了したということに なります。

```
  114 d0  I      0:00.10 /usr/libexec/getty V19200 ttyd0
```

getty は、CD (carrier detect) 信号がオンの状態になるまで、ポートのオープンを完了することはできませんので、この場合は接続に問題が あるか、あるいはモデムの設定に問題があることが考えられます。

もし、期待した ttydN ポートをオープンしようとしている getty が見あたらない場合は、再度 /etc/ttys の内容を確認し、書式などに誤りがないか 調べてみてください。また、ログ ファイル /var/log/messages に init および getty から何か出力がないかどうかも確認してみてく ださい。もし何かメッセージが記録されていたら、再度 /etc/ttys、/etc/gettytab の二つの設定ファイルと、ディバイス スペシャルファイル /dev/ttydN を確認し、記述に誤りがないか、足りないエントリがないか、足りないディバイス スペシャルファイルがないかといった 点について調べてみてください。

19.4.6.2. モデムで接続してみる

実際にモデムを使って別のコンピュータから 接続してみてください。この時、8 ビット、パリティなし、 1 ストップビットで接続するようにしてください。 接続後すぐにプロンプトが返ってこない場合や、 無意味な文字列が表示される 場合は、1秒に1回くらいの割合で Enter キーを押してみてください。しばらくたって、なおも login: プロンプトが現れない場合 は、BREAK 信号を送信してみてください。この時、端末側で使って いるモデムが高速モデムならば、 このモデムのインタフェースの接続速度を固 定してから、再度ダイアル インしてみてください。(たとえば、USRobotics Sportster の場合は、AT&B1)

それでもまだ login: プロンプトが表示されない場合は、/etc/gettytab の以下の点について再度確認してみてください。

- /etc/ttys の対応する行の 2番目の項目で、/etc/gettytab の中で定義されているエントリが指定されているか
- 各 nx= で /etc/gettytab の中で定義されているもの が指定されているか
- 各 tc= で /etc/gettytab の中で定義されているもの が指定されているか

もしダイアル インしても、FreeBSD システム側のモデムが応答しない場合は、FreeBSD 側のモデムが DTR がオンになった時に電話にでるように設定さ れているかを確認して

ください。もしモデムの設定に問題がなさそうならば、モデムのインジケータ (がもしあれば) で、DTR がオンになっているか を確認してください。

この確認のステップを数回繰り返しても うまくいかない場合は、一度休憩して、 しばらくたってから挑戦してみましょう。それでもだめなら、 おそらく FreeBSD general questions メーリングリスト にあなたのモデムについての情報と問題を書いたメールを送れ ば、メーリング リストのメンバーが問題の解決を助けるべく努力してくれる でしょう。

19.5. ダイアルアウトサービス

訳: 丸山 剛司 **<tmaruya@nnc.or.jp >**. 31 December 1996.

以下はモデムを利用して他のコンピュータと 接続する方法を説明しています。 これはリモートホストとターミナル接続を確立するための 適切な方法です。

これは BBS に接続するときによく使います。

この種の接続は PPP 接続に問題がある場合、Internet 上にあるファイルを 転送するのに非常に役に立ちます。FTP で何らかのファイルを転送したいのに PPP 接続を確立できない場合は、ファイルを FTP 転送するためにターミナルセッション を利用します。そしてZMODEM を利用してファイルを転送します。

19.5.1. 私の Hayes モデムはサポートされていません、どうすればよいでしょう?

実際、`tip` の マニュアルページは古くなっています。既に Hayes ダイアラが組み込まれています。`/etc/remote` ファイル中で `at=hayes` を使ってください。

Hayes ドライバは、最近のモデムの新しい機能である `BUSY`、`NO DIALTONE`、`CONNECT 115200` などのメッセージを 認識できるほど賢くはなく、単に混乱を起こすだけです。`tip`を使う場合には、(`ATX0&W` とするなどして) これらの メッセージを表示させないようにしなくてはいけません。

また、`tip` のダイアルのタイムアウトは 60秒です。モデムの タイムアウト設定はそれより短くすべきであり、 そうしないと `tip` は通信に問題があると判断するでしょう。`ATS7=45&W` を実行してください。

注記

デフォルトの `tip` は、Hayes モデムに完全に対応しているわけではありません。解決方法は `/usr/src/usr.bin/tip/tip` の下の `tipconf.h` を変更することです。もちろんこれにはソース配布ファイルが必要です。

> `#define HAYES 0` と記述されている行を `#define HAYES 1` と変更し、そして `make`, `make install` を実行します。これでうまく動作するでしょう。

19.5.2. これらの AT コマンドを入力するには?

`/etc/remote` ファイルの中で「direct」エントリを作ります。たとえばモデムが 1番目のシリアルポートである `/dev/cuaa0` に接続されている場合、次のようにします:

```
cuaa0:dv=/dev/cuaa0:br#19200:pa=none
```

モデムがサポートする最大の bps レートを br フィールドに使います。そして `tip cuaa0` を実行すると、モデムが利用できるようになります。

`/dev/cuaa0` がシステムに存在しない場合は、次のようにします:

```
# cd /dev
# sh MAKEDEV cuaa0
```

または `root` になって以下のように `cu` コマンドを実行します:

```
# cu -lline -sspeed
```

`line` にはシリアルポートを指定します (例えば `/dev/cuaa0`)。そして *`speed`* には接続する速度を指定します (例えば `57600`)。その後 AT コマンドを実行したら、~. と入力すれば終了します。

19.5.3. pn 機能の @ 記号が使えません!

電話番号 (pn) 機能の中での @ 記号は、tip に `/etc/phone` にある電話番号を参照するように伝えます。しかし @ の文字は `/etc/remote` のような 設定ファイルの中では特殊文字となります。バックスラッシュを使ってエスケープをおこないます:

```
pn=\@
```

19.5.4. コマンドラインから電話番号を指定するには?

「generic」エントリと呼ばれるものを `/etc/remote` に追加します。例えば次のようにします:

```
tip115200|Dial any phone number at 115200 bps:\
     :dv=/dev/cuaa0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600bps:\
     :dv=/dev/cuaa0:br#57600:at=hayes:pa=none:du:
```

そして

```
# tip -115200 5551234
```

のように利用できます。tip より cu を使いたい場合、cu の generic エントリを使います。

```
cu115200|Use cu to dial any number at 115200bps:\
 :dv=/dev/cuaa1:br#57600:at=hayes:pa=none:du:
```

そして

```
# cu 5551234 -s 115200
```

と実行します。

19.5.5. 毎回 bps レートを入力しなければいけませんか?

tip1200 や cu1200 用のエントリを記述し、適切な通信速度を br フィールドに設定します。tip は 1200 bps が正しいデフォルト値であるとみなすので、tip1200 エントリを参照します。もちろん 1200 bps を使わなければならないわけではありません。

19.5.6. ターミナルサーバを経由して複数のホストへアクセスしたいです

毎回接続されるのを待って CONNECT <host> と入力する かわりに、tip の cm 機能を使います。例えば、/etc/remote に次のようなエントリを追加します:

```
pain|pain.deep13.com|Forrester's machine:\
 :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank's machine:\
 :cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminal server:\
 :dv=/dev/cuaa2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

これで、tip pain や tip muffin と実行すると pain や muffin のホストに接続することができ、tip deep13 を実行するとターミナルサーバに接続します。

19.5.7. tip を使ってそれぞれのサイトの 複数の回線に接続できますか?

これは大学に電話回線がいくつかあって 数千人の学生が接続しようとする 場合によくある問題です。

あなたの大学のエントリを /etc/remote ファイルに作成して、pn のフィールドには @ を使います:

```
big-university:\
 :pn=\@:tc=dialout
dialout:\
 :dv=/dev/cuaa3:br#9600:at=courier:du:pa=none:
```

そして /etc/phone ファイルに大学の電話番号の一覧を書きます:

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

`tip` は一連の電話番号を試みて、最終的に接続できなければあきらめます。リトライを続けさせたい場合は、`tip` を while ループに入れて 実行します。

19.5.8. Ctrl+P を 1 回送るために Ctrl+P を 2 度押す必要があるのはなぜ?

Ctrl+P はデフォルトの「force (強制)」文字であり、`tip` に次の文字が リテラルデータであることを伝えます。force 文字は「変数の設定」を意味する `~s` エスケープによって他の文字にすることができます。

`~sforce=single-char` と入力して改行します。*`single-char`* は、任意の 1 バイト文字です。 *`single-char`* を省略すると NUL 文字になり、これは Ctrl+2 や Ctrl+Space を押しても入力できます。また、*`single-char`* に Shift+Ctrl+6 を割り当てる方法を使っているターミナルサーバもあります。

`$HOME/.tiprc` に次のように定義することで、任意の文字を force 文字として利用できます:

```
force=<single-char>
```

19.5.9. 打ち込んだ文字が突然すべて大文字になりました??

Ctrl+A を押してしまい、caps-lock キーが壊れている場合のために設計された `tip` の「raise character」 モードに入ったのでしょう。 既に述べたように `~s` を使って、`raisechar` をより適切な値に 変更してください。もしこれら両方の機能を使用しないのであれば、force 文字と同じ設定にすることもできます。

以下は Ctrl+2 や Ctrl+A などを頻繁に使う必要のある Emacs ユーザにうってつけの .tiprc ファイルのサンプルです。

```
force=^^
raisechar=^^
```

^^ は Shift+Ctrl+6 です。

19.5.10. `tip` でファイルを転送するには?

もし他の Unix のシステムと接続しているなら、`~p`(put) や `~t`(take) でファイルの送受信ができます。これらのコマンドは 相手のシステムの上で `cat` や `echo` を実行することで 送受信をします。書式は以下のようになります:

`~p` ローカルのファイル名 [リモートのファイル名]

~t リモートのファイル名 [ローカルのファイル名]

この方法ではエラーチェックをおこないませんので、zmodem などの他のプロトコルを使った方がよいでしょう。

19.5.11. tip から zmodem を実行するには?

ファイルを受信するには、リモート側で送信プログラムを起動します。そして ~C rz と入力すると、ローカル側へのファイルの受信が 始まります。

ファイルを送信するには、 リモート側で受信プログラムを起動します。そして ~C sz files と入力すると、リモート側への ファイルの送信が始まります。

19.6. シリアルコンソールの設定

##: Kazutaka YOKOTA.
##: Bill Paul.

19.6.1. 導入

FreeBSD は、 コンソールとしてシリアルポート上のダム端末しか持たないシステムでも起動します。 この様な構成はきっと次のような二種類の人達に便利でしょう。それは、キーボードやモニタのないマシンに FreeBSD をインストールしたいシステム管理者と、カーネルやデバイスドライバをデバッグしたい開発者です。

13章FreeBSD の起動のプロセス で説明されているように、FreeBSD は 3 ステージ構成のブートストラップを用いています。最初の 2 つのステージは、ブートディスクにある FreeBSD スライスの最初に格納されている、ブートブロックのコードが行います。それからブートブロックは、第 3 ステージのコードとしてブートローダ (/boot/loader) を読み込み、実行します。

シリアルコンソールを設定するためには、ブートブロックコード、 ブートローダコード、カーネルを設定する必要があります。

19.6.2. シリアルコンソールの設定

1. シリアルケーブルを用意してください。

   ヌルモデムケーブル、もしくは標準シリアルケーブルとヌルモデムアダプタが必要となります。シリアルケーブルについては「ケーブルとポート」をご覧ください。

2. キーボードをはずして下さい。

   たいていの PC システムは Power-On Self-Test (POST) の間にキーボードを検出し、もし見つからなければエラーと なります。また、キーボードがないことを大

きな音で知らせ、キーボードが接続されるまでは起動を中断するようなマシンもあります。

コンピュータがエラーを表示していても、とにかく起動するなら特別な対応は必要ありません (Phoenix BIOS を搭載しているマシンには、Keyboard failed と表示されても、正常に起動するものがあります)。

あなたのコンピュータがキーボードを接続していない状態で 起動しないようなら、(もし可能ならば) エラーを無視するように BIOS を設定する必要があります。設定方法の詳細については、マザーボードのマニュアルを調べてください。

ヒント

BIOS の設定でキーボードを「Not installed」にするということは、キーボードを使えないということを 意味しているわけではありません。これは、BIOS がキーボードがなくても文句を言わないように、電源投入時には キーボードを探すな、と指示するだけです。このフラグを 「Not installed」 にしていてもキーボードを 接続したままにできますし、ちゃんと動作します。

注記

あなたのシステムが PS/2 マウスを使っているなら、 おそらくマウスもキーボード同様にはずす必要があるでしょう。 というのは、PS/2 マウスは部分的にキーボードとハードウェアを 共有しており、マウスを接続したままにしていると、 キーボードも存在する、と誤って検出してしまう可能性があるからです。AMI BIOS を持つ Gateway 2000 Pentium 90MHz システム はこれに該当すると言われています。一般的にこれは問題ではありません。なぜなら、どっちにしても マウスはキーボードなしではたいして役に立たないからです。

3. `COM1` (`sio0`) にダム端末を接続してください。

ダム端末がなければ、かわりに古い PC/XT でモデム プログラムを走らせて使ったり、シリアルポートに他の Unix マシンを繋いだりできます。もしも `COM1` (`sio0`) がなければ、作成してください。今のところ、`COM1` 以外のポートを 選択するためにはブートブロックの再コンパイルが必要です。すでに `COM1` を他の装置に 使っていた場合は、一時的にその装置をはずして いったん FreeBSD がうまく動作してから、

新しいブートブロックとカーネルをインストールしてください。 (上記はとにかくファイル/演算/端末サーバの COM1 が利用可能であると仮定して います。あなたが本当に何かのために COM1 が必要 (で、なおかつその何かを COM2 (sio1) に付け替えることができない) ならば、多分、そもそも 悩んでる場合ではありません。)

4. カーネルコンフィグファイルの COM1 (sio0) に適切なフラグを 設定していることを確認してください。

   関連するフラグ:

   0x10
   : このポートのコンソールサポートを有効にします。このフラグが設定されない場合、他のフラグは無視されます。 現在のところ、一つのポートしかコンソールサポートを有効に できません。(config ファイルに書かれた順番で) 最初にこのフラグを 指定されたポートが選択されます。なお、このオプションを指定するだけでシリアルポートが コンソールとして使えるわけではありません。 このフラグと一緒に、以下のフラグも指定するかもしくは -h オプションも使ってください。

   0x20
   : 後述される -h オプション を無視して、(他に優先度の高いコンソールがない限り) このポートをコンソールとして指定します。このフラグは FreeBSD バージョン 2.*X* の COMCONSOLE オプションに対応するものです。フラグ 0x20 は必ず フラグ 0x10 と一緒に指定されなければなりません。

   0x40
   : (0x10 と組み合わせることで) このポートを予約し、通常のアクセスができないようにします。 このフラグは、シリアルコンソールとして使いたいポートに 指定すべきではありません。 唯一の使い道は、ユニットがカーネルのリモートデバッグ用 であることを指定することです。リモートデバッグの詳細については The Developer's Handbook を参照してください。

   > **注記**
   >
   > FreeBSD 4.0 以降では、フラグ 0x40 の意味が若干異なり、シリアルポートにリモートデバッグを指定するためには、別のフラグを使います。

   例:

   ```
   device sio0 at isa? port "IO_COM1" tty flags 0x10 irq 4
   ```

   詳細については sio(4) のマニュアルを参照してください。

もしこれらのフラグがセットされていなければ、(別のコンソールで) UserConfig を実行するか、カーネルを再コンパイルする必要があります。

5. ブートドライブの a パーティションの ルートディレクトリに boot.config を作成してください。

   このファイルは、ブートブロックコードに対してどのように システムを起動したいかを教えます。シリアルコンソールを活かすためには、以下のオプションを幾つか — 複数の場合も一行で、設定する必要があります:

   -h
   : 内蔵コンソールとシリアルコンソールの切替えを行います。これを使用してコンソールデバイスを変更できます。例えば、内蔵 (ビデオ) コンソールからブートした場合、カーネルとブートローダがコンソールデバイスとして シリアルポートを使用するようにするため、-h を使って指示できます。反対に、シリアルポートからブートした場合、ブートローダとカーネルがコンソールとして代わりに ビデオディスプレイを使用するようにするため、-h を使用できます。

   -D
   : シングルとデュアルのコンソール設定を切り替えます。 シングル設定では、上記の -h オプションの状態によって、コンソールは内蔵コンソール (ビデオディスプレイ)かシリアルポートのいずれかになります。 デュアルコンソール設定では、ビデオディスプレイと シリアルポートの両方が、-h オプションの状態によらず、同時にコンソールになります。 しかし、デュアルコンソール設定は、ブートブロックが 実行されている間でしか効果を持ちません。一旦ブートローダに制御が移ると、-h オプションによって指定されたコンソールが 唯一のコンソールになります。

   -P
   : ブートブロックがキーボードを検出するようにします。 キーボードが発見できなかった場合には、-D と -h オプションが自動的にセットされます。

   注記

   現バージョンのブートブロックでは容量の制限により、 -P オプションは拡張キーボードしか 検出できません。キーが 101 個より少ない (そして F11 と F12 がない) キーボードは検出されない可能性があります。この制限から、いくつかのラップトップコンピュータの キーボードは正しく検出されないでしょう。もし、あなたのシステムがこのようなキーボードを使っているのであれば、-P オプションを

外してください。 残念ながら、この問題の回避策はありません。

`-P` オプションを使ってコンソールを 自動的に選ぶか、`-h` オプションを使って シリアルコンソールを有効にしてください。

さらに boot(8) で説明されている他のオプションも使う ことができます。

`-P` 以外のオプションはブートローダ (`/boot/loader`) に渡されます。ブートローダは、`-h` オプションだけの状態を 調べることで内蔵ビデオとシリアルポートのどちらがコンソールに なるのか決めます。つまり、`/boot.config` の中で `-D` オプションを指定して `-h` オプションを指定しなかった場合、ブートブロック実行中でのみシリアルポートをコンソールとして 使うことができます。ブートローダは内蔵ビデオディスプレイを コンソールとして使います。

6. マシンを起動する。

FreeBSD を起動したとき、ブートブロックは `/boot.config` の内容をコンソールに表示 します。例えば、

```
/boot.config: -P
Keyboard: no
```

行の二番目は、`/boot.config` にオプション `-P` が指定してあるときだけ表示され、キーボードが存在するかどうかを表します。これらのメッセージは、シリアルか内蔵のいずれか、あるいはその両方のコンソールに表示されます。どちらに表示されるかは、`/boot.config` の設定によって変わります。

オプション指定	メッセージの表示される場所
なし	内蔵
`-h`	シリアル
`-D`	シリアルと内蔵の両方
`-Dh`	シリアルと内蔵の両方
`-P`、キーボードが存在する場合	内蔵
`-P`、キーボードが存在しない場合	シリアル

このメッセージが表示された後、ブートブロックがブートローダのロードを再開し、他の全てのメッセージがコンソールに表示されるまで、 若干時間がかかります。通常の環境では、ブートブロックに 割り込みをかける必要はありませんが、ちゃんとセットアップされているかどうか確かめるために、割り込みをかけることができるようになっています。

ブートプロセスに割り込みをかけるには、 コンソールの (Enter 以外の) キーをたたいて下さい。 ブートブロックはその時、操作を指定するためのプロンプトを表示します。こんな風に表示されるでしょう。

```
>> FreeBSD/i386 BOOT
Default: 0:wd(0,a)/boot/loader
boot:
```

上に示したメッセージが、シリアルか内蔵、 あるいはその両方といった、`/boot.config` で指定したとおりのコンソールに表示されることを確認して下さい。メッセージが正しいコンソールに表示されたら、Enter キーを押してブートプロセスを継続してください。

もし、シリアルコンソールを利用するように設定しているのに シリアル端末にプロンプトが出てこない場合は、設定のどこかに間違いがあります。ブートブロック(とブートローダ、カーネル)に対して シリアルポートをコンソールに使うことを伝えるため、割り込みをかけた時に `-h` を入力し、(可能ならば) Enter/Return キーを押して下さい。そして、一度システムを起動させてから、どこが悪いのかをチェックして下さい。

ブートローダがロードされ、ブートプロセスの第三ステージに いる時には、まだ内蔵コンソールとシリアルコンソールを切り替えることができます。 それにはブートローダの環境変数を適切に設定すれは良いのですが、詳細については「ブートローダからコンソールを変更するには」を参照してください。

19.6.3. まとめ

このセクションで扱ったさまざまな設定と、 最終的に選択されるコンソールに関するまとめです。

19.6.3.1. Case 1: **sio0** の flags に 0x10 をセットした場合

```
device sio0 at isa? port "IO_COM1" tty flags 0x10 irq 4
```

/boot.config 内のオプション	ブートブロック実行中のコンソール	ブートローダ実行中のコンソール	カーネルのコンソール
なし	内蔵	内蔵	内蔵
`-h`	シリアル	シリアル	シリアル
`-D`	内蔵、シリアルの両方	内蔵	内蔵
`-Dh`	内蔵、シリアルの両方	シリアル	シリアル
`-P`、キーボードが存在する場合	内蔵	内蔵	内蔵

/boot.config 内のオプション	ブートブロック実行中のコンソール	ブートローダ実行中のコンソール	カーネルのコンソール
-P、キーボードが存在しない場合	内蔵、シリアルの両方	シリアル	シリアル

19.6.3.2. Case 2: sio0 の flags に 0x30 をセットした場合

```
device sio0 at isa? port "IO_COM1" tty flags 0x30 irq 4
```

/boot.config 内のオプション	ブートブロック実行中のコンソール	ブートローダ実行中のコンソール	カーネルのコンソール
なし	内蔵	内蔵	シリアル
-h	シリアル	シリアル	シリアル
-D	内蔵、シリアルの両方	内蔵	シリアル
-Dh	内蔵、シリアルの両方	シリアル	シリアル
-P、キーボードが存在する場合	内蔵	内蔵	シリアル
-P、キーボードが存在しない場合	内蔵、シリアルの両方	シリアル	シリアル

19.6.4. シリアルコンソールを利用する上で役に立つ情報

19.6.4.1. シリアルポートの通信速度をもっと速いものに設定するには

デフォルトのシリアルポート通信速度は、9600 ボー、8 ビット、パリティなし、ストップビット 1 です。通信速度を変更したい場合には、少なくとも ブートブロックの再コンパイルが必要になります。`/etc/make.conf` に次のような行を追加して、新しくブートブロックをコンパイルして下さい。

```
BOOT_COMCONSOLE_SPEED=19200
```

もし、シリアルコンソールがブート時の `-h` オプション以外の方法で設定されていたり、カーネルが利用するシリアルコンソールが ブートブロック実行中のものと異なる場合には、 カーネルコンフィグレーションファイルに次のオプションを追加して、 新しくカーネルをコンパイルしなければなりません。

```
options CONSPEED=19200
```

19.6.4.2. sio0 以外のシリアルポートを コンソールとして使うには

sio0 以外のポートをコンソールとして使うには、再コンパイルが必要です。それがどんな理由であれ、他のポートを使用する場合には ブートブロック、ブートローダ、カーネルを次のようにして再コンパイルして下さい。

1. カーネルソースを取得する (18章FreeBSD のアップデートとアップグレード をご覧ください)。

2. /etc/make.conf を編集し、BOOT_COMCONSOLE_PORT に 使用したいポートのアドレス(0x3F8、0x2F8、0x3E8 or 0x2E8)を 設定してください。使用可能なのは sio0 から sio3 (COM1 から COM4) までで、マルチポートシリアルカードは使えません。また、ここで割り込みの設定をする必要はありません。

3. 設定を変更するために新たなカーネルコンフィグレーションファイルを作成し、 使いたいシリアルポートのフラグを適切に設定します。例えば、sio1 (COM2) をコンソールにしたければ、

   ```
   device sio1 at isa? port "IO_COM2" tty flags 0x10 irq 3
   ```

   または、

   ```
   device sio1 at isa? port "IO_COM2" tty flags 0x30 irq 3
   ```

   とします。その際、他のシリアルポートにコンソールフラグをつけてはいけません。

4. ブートブロックを再コンパイルし、インストールする。

   ```
   # cd /sys/boot/i386/boot2
   # make
   # make install
   ```

5. ブートローダを再コンパイルし、インストールする。

   ```
   # cd /sys/boot/i386/loader
   # make
   # make install
   ```

6. カーネルを再構築し、インストールする。

7. disklabel(8) を使ってブートブロックをブートディスクに書き込み、 新しいカーネルから起動する。

19.6.4.3. シリアルポートから DDB デバッガを起動するには

シリアルコンソールからカーネルデバッガを起動したい(これは リモートで診断する際に便利ですが、もしおかしな BREAK 信号がシリアルポートに送られるような場合には危険です!) 場合には、次のオプションを使ってカーネルをコンパイルして下さい。

```
options BREAK_TO_DEBUGGER
options DDB
```

19.6.4.4. シリアルコンソールにログインプロンプトを表示させるには

シリアルコンソールからブートメッセージを確認したり、　シリアルコンソールを経由してカーネルデバッグセッションに入ることが　できるので、これは必要がないかもしれませんが、login プロンプトをシリアルポートに 出力するように設定することもできます。これには、次のようにします。

エディタで `/etc/ttys` というファイルを開き、次に示す行に移動して下さい。

```
ttyd0 "/usr/libexec/getty std.9600" unknown off secure
ttyd1 "/usr/libexec/getty std.9600" unknown off secure
ttyd2 "/usr/libexec/getty std.9600" unknown off secure
ttyd3 "/usr/libexec/getty std.9600" unknown off secure
```

`ttyd0` から `ttyd3` は、`COM1` から `COM4` に対応しています。設定したいポートの `off` を `on` に変更して下さい。また、もしシリアルポートの通信速度を変更しているなら、`std.9600` が実際の通信速度になるように、例えば `std.19200` のように変更して下さい。

さらに、実際のシリアル端末に合わせて、端末タイプを `unknown` から変更することも可能です。

ファイルの編集が終了したら、変更を有効化するために `kill -HUP 1` を実行しなければなりません。

19.6.5. ブートローダからコンソールを変更するには

前セクションは、ブートブロックの設定を変更することでシリアルコンソールを　セットアップする方法について解説していました。このセクションでは、ブートローダへのコマンド入力と環境変数設定で コンソールの指定を行なう方法を紹介します。ブートローダがブートブロックの後、 ブートプロセスの第三ステージで呼び出されたとき、 ブートローダの設定には、ブートブロックの設定がそのまま使われます。

19.6.5.1. シリアルコンソールをセットアップする

ブートローダとカーネルに対して シリアルコンソールを使用するように設定するには、 単に `/boot/loader.rc` のファイルに、次のような一行を書くだけで実現できます。

```
set console=comconsole
```

これは、前セクションで扱ったブートブロックの設定に 全く関係なく機能します。

上に示した行は、`/boot/loader.rc` の最初の行に書き込まなくてはいけません。これはできるだけ早く、ブートメッセージをシリアルコンソールに 出力させるために必要なことです。

同様にして、次のように内蔵コンソールを指定することもできます。

```
set console=vidconsole
```

もし、ブートローダの環境変数 `console` が設定されていない場合、ブートローダ、そしてその次に起動するカーネルは ブートブロックで指定された `-h` オプションに 示されたコンソールを使用します。

3.2 以降のバージョンにおいては `/boot/loader.rc` ではなく、 `/boot/loader.conf.local` や `/boot/loader.conf` にコンソール指定を書き込みます。その場合、`/boot/loader.rc` は次のようになっていなければなりません。

```
include /boot/loader.4th
start
```

それから、`/boot/loader.conf.local` を作成して、次の行をそこに追加して下さい。

```
console=comconsole
```

か、もしくは

```
console=vidconsole
```

です。詳細については、loader.conf(5) を参照して下さい。

注記

その際、ブートローダはオプション指定なし (ブートブロックに `-P` オプションが指定されたのと等価)になり、 キーボードの存在を調べて 内蔵コンソールとシリアルコンソールを自動的に選択する機能は働きません。

19.6.5.2. `sio0` 以外のシリアルポートを コンソールとして使うには

`sio0` 以外のシリアルポートを コンソールとして使うには、ブートローダを再コンパイルする必要があります。 それには、「`sio0` 以外のシリアルポートを コンソールとして使うには」に書かれている説明にしたがって下さい。

19.6.6. 注意

シリアルコンソールというアイデアは、 グラフィック出力用のハードウェアやキーボードが接続されていない 専用サーバのセットアップを可能にするためのものです。 ほとんどのシステムはキーボードなしで起動できますが、 不幸にも、グラフィックアダプタなしでは起

動できないシステムはたくさんあります。AMI BIOS を採用しているマシンでは、CMOS 設定の `graphics adapter' を `Not Installed' にするだけで、グラフィックアダプタがなくとも起動できるように設定することができます。

しかしながら、多くのマシンはこのようなオプションを持っていませんし、ディスプレイハードウェアがシステムに存在しないと起動しないように　なっています。そのようなマシンでは、モニタを接続する必要がなかったとしても、適当なグラフィックカード(モノクロのジャンク品でも構いません)を 挿入したままにしておく必要があるでしょう。また、AMI BIOS をインストールする、という手もあります。

第20章 PPP と SLIP

改訂: Jim Mock, 2000 年 3 月 1 日.

20.1. この章では

もしあなたがモデムを使ってインターネットに接続したり，他の人々に FreeBSD による インターネットへのダイヤルアップ接続を 提供しようとしているのでしたら，PPP または SLIP 接続を選択することができます.

この節では 3 種類の PPP について説明しています. それは ユーザ, カーネル, そして PPPoE (PPP オーバイーサネット) です. また SLIP のクライアントとサーバの設定についても記述しています.

最初に説明するのは, ユーザ PPP です. ユーザ PPP は FreeBSD に 2.0.5-RELEASE の時に, 既に存在していたカーネル実装の PPP に加えて導入されました.

ユーザ PPP とカーネル PPP の主な違いは何かと疑問に思われるかも 知れませんが, その答えは簡単です. ユーザ PPP はデーモンとしては実行されず 必要に応じて実行されるのです. PPP インタフェイスを組み込んだカーネルは 必要ではなく, ユーザプロセスとして実行されカーネルとのデータの やり取りにはトンネルデバイスドライバ (`tun`) を 使用します.

この節ではこれ以降ユーザ PPP のことは, `pppd` のような他の PPP ソフトウエアと特に区別する必要がある場合を除いて, 単に `ppp` と記述します. またこの節に記述されているコマンドは すべて root で実行されなければなりません.

20.2. ユーザ ppp の利用

原作: Brian Somers, 協力: Nik Clayton, Dirk-Willem van Gulik **<Dirk.vanGulik@jrc.it >**, Peter Childs **<pjchilds@imforei.apana.org.au >**.

20.2.1. ユーザ PPP

20.2.1.1. 前提条件

以下の情報を手に入れておく必要があるでしょう:

- PPP で接続するインターネットサービスプロバイダ (ISP) のアカウント. さらに, 接続済みのモデム (またはその他のデバイス) があり, プロバイダとの接続が可能なように正しく設定されている.
- プロバイダの電話番号.

- ログイン名とパスワード. これは通常の unix 形式のログイン名と パスワードの組という場合もありますし, PPP PAP または CHAP の ログイン名とパスワードの組という場合もあります.

- 一つ以上のネームサーバの IP アドレス. 通常, プロバイダから IP アドレスを二つ指示されている はずです. 一つすら提供されていないならば, `ppp.conf` ファイル中で `enable dns` コマンドを使って ppp にネームサーバを設定するよう 指示できます.

プロバイダからは以下の情報が提供されているはずですが, どうしても必要というわけではありません:

- プロバイダのゲートウェイの IP アドレス. ゲートウェイとは, あなたがそこに接続をおこなって, デフォルトルート として設定することになるマシンです. プロバイダがこのアドレスを明示していなくても, 最初は 適当に設定しておいて, 接続時にプロバイダの PPP サーバから 正しいアドレスを教えてもらうことができます.

  このアドレスは, ppp から `HISADDR` として参照されます.

- プロバイダのネットマスク設定. プロバイダが明示していないとしても, ネットマスクとして `255.255.255.0` を使用しておけば問題ありません.

- もしプロバイダから固定の IP アドレスとホスト名の割り当てを 受けていれば, その情報を指定しておくこともできます. 割り当てを受けていなければ, 接続先から適切な IP アドレスを指定してもらいます.

もし, 必要な情報が不足していれば, プロバイダに連絡を取って 確認しておいてください.

20.2.1.2. ppp 対応カーネルの構築

説明でも述べているように, `ppp` はカーネルの `tun` デバイスを使います. 使っているカーネルがどれであっても, `tun` デバイスを設定しなければなりません. FreeBSDに付属しているデフォルトの `GENERIC` カーネルに合うように `tun` デバイスは前もって設定されています. しかしながら, 自分で修正したカーネルをインストールするのであれば, pppが正しく動くよう, カーネルが設定されているか確認しなくてはいけません.

これを確認するには, カーネルコンパイルディレクトリ (`/sys/i386/conf` または `/sys/pc98/conf`) に移動して, カーネルコンフィグレーションファイルを調べます. 以下の行がどこかに含まれている必要があります.

```
pseudo-device tun 1
```

この行がカーネルコンフィグレーションファイルに 含まれていない場合, この行を追加して カーネルの再コンパイルとインストールをおこなう必要があります. 元々の `GENERIC` カーネルは 標準でこれを含んでいますので, カスタムカーネルをインストールしているのではなかったり, `/sys` ディレクトリが存在しないのであれば, 何も変更する必要はありません. カーネルコンフィグレーションの詳細については, FreeBSD カーネルのコンフィグレーション を参照してください.

以下のコマンドを実行することで，現在のカーネルにトンネルデバイスが いくつ組み込まれているかを調べることができます:

```
# ifconfig -a
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
 inet 200.10.100.1 --> 203.10.100.24 netmask 0xffffffff
tun1: flags=8050<POINTOPOINT,RUNNING,MULTICAST> mtu 576
tun2: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
 inet 203.10.100.1 --> 203.10.100.20 netmask 0xffffffff
tun3: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
```

注記

FreeBSD 4.0やより最近のリリースでは， すでに使われている tun デバイスしか見つけることが できないでしょう. これは，全く tun デバイスを見つけることが できないかもしれないということです. しかし，もしこうなって しまっても，心配することはありません. そのデバイスは ppp が使おうとする時に動的に作られるはず だからです.

この例ではトンネルデバイスが四つ存在し，そのうち二つに 設定がおこなわれ，使用中であることがわかります. 上の例で RUNNING フラグがオンになっている ものがありますが，これは そのインタフェースが何かに使用されていることを示している だけであるということに注意してください. つまり，RUNNING になっていない インタフェースがあったとしても，それはエラーではありません.

トンネルデバイスがカーネルに組み込まれておらず，何らかの理由で カーネルの再構築ができない場合でも，方法がないわけではありません. 動的にデバイスをロードすることができるはずです. 詳細については modload(8) や lkm(4) など，適切なマニュアルを参照してください.

20.2.1.3. tun デバイスの確認

ほとんどのユーザは tun デバイス (/dev/tun0) が一つあれば充分でしょう. より多くのデバイスを使う場合 (すなわち，カーネルコンフィグレーション ファイルで pseudo-device tun の行に 1 以外の数値を指定している場合)，以下で tun0 と書かれている部分をすべて，あなたが使うデバイスの番号に あわせて読みかえてください.

tun0 デバイスが正しく作成されていることを確認する最も簡単な方法は，それを作り直すことです. そのためには，以下のコマンドを実行します:

```
# cd /dev
# ./MAKEDEV tun0
```

カーネルに 16 個のトンネルデバイスを組み込んだのであれば, tun0 だけでなく他の tun デバイスも作成しておく必要があるでしょう:

```
# cd /dev
# ./MAKEDEV tun15
```

また, カーネルが正しく設定されているかどうかを調べるために 以下のコマンドを実行して, このような出力が得られることを確認します:

```
# ifconfig tun0
tun0: flags=8050<POINTOPOINT,RUNNING,MULTICAST> mtu 1500
```

まだ RUNNING フラグがセットされていない場合もあります. その時は以下のような出力が得られるでしょう:

```
# ifconfig tun0
tun0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
```

前述したように, FreeBSD 4.0 以降のリリースでは tun デバイスは要求に応じて 作られるので, もしそのデバイスがまだ使われていなければ, 見つけられないかもしれないということを思い出してください.

20.2.1.4. 名前の解決に関する設定

リゾルバ (resolver) はシステムの一部分で, IP アドレスとホスト名との 変換をおこないます. IP アドレスとホスト名を対応させるためのマップを, 二つの場所のうちの一つから探すように設定できます. 一つめは /etc/hosts (man 5 hosts) と呼ばれるファイルです. 二つめはインターネット ドメインネームサービス (DNS) と呼ばれる 分散データベースですが, これに関する議論は このドキュメントで扱う範囲を 越えていますので, これについての説明はおこないません.

リゾルバは名前のマッピングを おこなうシステムコールの集合体です. ただし どこからマッピング情報を見つけるのかは, 最初に指示しておく必要があります. これは まず /etc/host.conf ファイルを編集することでおこないます. 混乱の元になりますので, このファイルを /etc/hosts.conf と 呼んだりしてはいけません (余分な s がついていますね).

20.2.1.4.1. /etc/host.conf ファイルの編集

このファイルには 以下の 2 行が (この順番で) 書かれているはずです:

```
hosts
bind
```

これは, 最初に /etc/hosts ファイルを調べ, そこで目的の名前が 見つけられなかった場合に DNS を引きにいくようリゾルバに指示します.

20.2.1.4.2. /etc/hosts(5) ファイルの編集

このファイルはローカルネットワーク上に存在するマシンの IP アドレスと ホスト名を含んでいるはずです. 最低でも ppp を動作させるマシンのエントリが 含まれている必要があります. そのマシンのホスト名が `foo.bar.com` で, IP アドレスが `10.0.0.1` であると仮定すると, `/etc/hosts` は 以下の行を含んでいなければいけません:

```
127.0.0.1 localhost.bar.com localhost
127.0.0.1 localhost.bar.com.
10.0.0.1 foo.bar.com  foo
10.0.0.1 foo.bar.com.
```

一つめの行は `localhost` を現在のマシンの別名として定義しています. マシン固有の IP アドレスが何であっても, この行の IP アドレスは 常に `127.0.0.1` でなければいけません. 二つめの行はホスト名 `foo.bar.com` (と, その省略形 `foo`) を IP アドレス `10.0.0.1` にマップします.

もしプロバイダから固定の IP アドレスとホスト名を割り当てられて いるのであれば, それを `10.0.0.1` エントリのかわりに使ってください.

20.2.1.4.3. `/etc/resolv.conf` ファイルの編集

`/etc/resolv.conf` はリゾルバの振舞いを指定します. もし自前の DNS サーバを走らせているのなら, このファイルは空のままに しておくこともできます. 通常は, 以下のように書いておく必要があるでしょう:

```
domain bar.com
nameserver x.x.x.x
nameserver y.y.y.y
```

x.x.x.x と *y.y.y.y* はプロバイダから指示されたアドレスで, 接続するプロバイダが提供しているネームサーバを すべて書いてください. `domain` に指定するのは このマシンのデフォルトのドメイン名で, おそらく 書かなくても問題は無いでしょう. このファイルの各エントリの詳細については, `resolv.conf` のマニュアルページを参照してください.

バージョン 2 以降の ppp を使用している場合には, `enable dns` コマンドを使用してネームサーバのアドレスを プロバイダに問い合わせるように指示することができます. 上の指定とは異なるアドレスをプロバイダが指定してきた場合 (または `/etc/resolv.conf` でネームサーバが指定されていない場合), ppp はプロバイダが指定したアドレスで `resolv.conf` を書きかえます.

20.2.1.5. ppp の設定

ユーザ ppp と `pppd` (カーネルレベルの PPP 実装) は どちらも `/usr/share/examples/ppp` ディレクトリに置かれた設定ファイルを使います. ここには設定ファイルのサンプルが用意されていて, ユーザ ppp の設定を おこなう際に大変参考になりますので, 削除したりしないでください.

ppp の設定をするためには, 必要に応じていくつかのファイルを編集する必要が あります. 書き込む内容は, プロバイダが静的に IP アドレスを割り当てる (つまり, 固定の IP アドレスを一つ与えられて, 常にそれを使う) か, または動的に IP アドレスを割り当てる (つまり, PPP セッションごとに IP アドレスが変化する可能性がある) かということに ある程度依存します.

20.2.1.5.1. 静的 IP アドレスによる PPP 接続

まず /etc/ppp/ppp.conf という設定ファイルを作成する必要があります. これは以下の例とほとんど同じようなものになるでしょう.

注記

: で終る行は 1 カラム目から始め, その他の行はスペースまたはタブで以下の例のように 段をつける (インデントする) 必要があります.

```
1     default:
2       set device /dev/cuaa0
3       set speed 115200
4       set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \"\" ↺
ATE1Q0 OK-AT-OK \\dATDT\\TTIMEOUT 40 CONNECT"
5     provider:
6       set phone "(123) 456 7890"
7       set login "TIMEOUT 10 \"\" \"\" gin:--gin: foo word: bar ↺
col: ppp"
8       set timeout 300
9       set ifaddr x.x.x.x y.y.y.y 255.255.255.0 0.0.0.0
10      add default HISADDR
11      enable dns
```

ファイルでは行番号を取り除いておいてください. これは解説の際に参照する行を示すためにつけたものです.

Line 1:
デフォルトエントリを指定します. このエントリ中のコマンドは ppp が起動された際に自動的に実行されます.

Line 2:
モデムが接続されているデバイスを指定します. COM1: は /dev/cuaa0 に, COM2: は /dev/cuaa1 になります.

Line 3:
通信速度 (DTE 速度) を指定します. もし 115200 が使えない (最近のモデムなら大抵使えるはずですが) 場合には, かわりに 38400 を指定してみてください.

Line 4:
ダイアルスクリプトを指定します. ユーザ PPP は chat(8) 言語に似た, 受信待ち文字列と 送信文字列の対からなるスクリプトを使用します. この言語の機能に関しては, マニュアルページを参照してください.

Line 5:
接続するプロバイダの名前「provider」を エントリ名として指定します.

Line 6:
このプロバイダの電話番号を指定します. 複数の電話番号を `:` や `|` で区切って指定することができます. これら区切り文字の違いについては, ppp(8) に 詳しく書かれています. 要約すると, 毎回違う番号に かけたいのであれば `:` を使います. 常にまず先頭の番号にかけてみて, つながらない時にだけ 2 番目以降の番号に かけたいのであれば `|` を使います. 例に示されているように, 常に電話番号全体を引用符で くくって (クォートして) おきます.

Line 7:
ダイアルスクリプトと同様に, ログインスクリプトも chat 言語風の記述をおこないます. この例は, 以下のようなログインセッションを使用する プロバイダのためのものです:

```
J. Random Provider
login: foo
password: bar
protocol: ppp
```

このスクリプトは必要に応じて 書きかえなければならないでしょう. 初めてスクリプトを書く時には, 予想した通りに 処理が進んだかどうかを確認するため,「chat」ログを とるようにしておいた方が良いでしょう.

PAP や CHAP を使用する場合には, ここでログインすることは ありませんから, ログイン文字列は空白のままにしておくべきです. 詳細については PAP および CHAP による認証を参照してください.

Line 8:
デフォルトの接続タイムアウト時間を (秒数で) 指定します. この例では, 300 秒間通信がおこなわれなければ 自動的に接続を切るように指定しています. タイムアウトさせたくない場合には, この値を 0 に設定します.

Line 9:
インタフェースのアドレスを指定します. 文字列 *x.x.x.x* は プロバイダに割り当てられた IP アドレスで置きかえてください. 文字列 *y.y.y.y* はプロバイダから指示されたゲートウェイ (接続先となるマシン) の IP アドレスで置きかえてください. プロバイダがゲートウェイのアドレスを 指示していない場合は, `10.0.0.2/0` を使用しておいてください. もし「仮の」アドレスを使用する必要がある場合には, 動的 IP アドレスによる PPP 接続に関する指示に従って, `/etc/ppp/ppp.linkup` にエン

トリを作成していることを 確認してください. この行が省略されている場合, ppp を -auto モードで動作させることはできません.

Line 10:
プロバイダのゲートウェイへの経路を デフォルトルートとして 追加します. 特殊文字列 HISADDR は, 9 行目で指定された ゲートウェイのアドレスで置きかえられます. HISADDR は 9 行目までは初期化されていませんので, その行よりも後でしか使えないことに 注意してください.

Line 11:
ネームサーバのアドレスが正しいか どうかを確認するため, プロバイダに問い合わせをおこなうよう ppp に指示します. プロバイダがこの機能をサポートしていれば, ppp は /etc/resolv.conf のネームサーバエントリを 正しいアドレスに更新することができます.

静的な IP アドレスを持っていて, 接続が完了する前にルーティングテーブルの エントリが正しく設定されているのであれば, ppp.linkup に エントリを追加する必要はありません. しかし, この場合でもエントリを追加して, 接続が完了した時点で プログラムを呼び出したいことがあるかもしれません. これについては後ほど sendmail を例として説明します.

これらの設定ファイルのサンプルが /usr/share/examples/ppp ディレクトリに置かれています.

20.2.1.5.2. 動的 IP アドレスによる PPP 接続

プロバイダが静的な IP アドレスの割り当てをおこなっていない場合, ppp が相手側のホスト (ゲートウェイ) と交渉して, こちら側と相手側のアドレスを 決めるように設定することができます. これは, 起動時には「仮の」アドレスを使っておいて, 接続後に IP コンフィグレーション プロトコル (IPCP) を使用して ppp が IP アドレスを正しく設定できるようにすることで実現されます. 静的 IP アドレスによる PPP 接続に 以下の変更を加える以外は, ppp.conf の設定は同じです:

```
9       set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.0
```

繰り返しますが, 行番号は取り除いておいてください. これは解説の際に参照する行を示すためにつけたものです. なお, 少なくともスペース 1 個分の段づけ (インデント) が必要です.

Line 9:
/ 文字の後ろの数字は, アドレス交渉の際に固定しておきたい ビットの数です. 場合によっては, もっと適切な IP アドレスを 指定しておきたいこともあるかもしれませんが, ほとんどの場合には 上の例の通りで問題ありません.

最後の引数 (0.0.0.0) は, アドレスの交渉の際に 10.0.0.1 ではなく 0.0.0.0 を使用するよう ppp に指示するためのものです. set ifaddr コマンド

の最初の引数として 0.0.0.0 を指定してはいけません. さもないと, -auto モードで動作させる際に 初期経路を設定することができなくなります.

バージョン 1.X の ppp を使用する場合, /etc/ppp/ppp.linkup にもエントリを作成しておく必要があります. ppp.linkup は接続が確立された後に使用されます. この時点では, ppp は実際にどの IP アドレスを使うべきなのか わかっているはずです. 以下のエントリは存在する仮の経路を削除し, 正しい経路を作成します:

```
1     provider:
2       delete ALL
3       add default HISADDR
```

Line 1:
接続を確立する際に, ppp は以下のルールに従って ppp.linkup のエントリを検索します: まず ppp.conf で使用されたのと同じラベルを探します. もし見つからなければ, ゲートウェイの IP アドレスのエントリを 探します. このエントリは 4 オクテットの IP アドレス形式の ラベルです. それでも まだエントリが見つからなければ, MYADDR エントリを探します.

Line 2:
この行は, 使用する tun インタフェースに関する既存の経路を (ダイレクトルートのエントリを除き) すべて削除するよう ppp に指示します.

Line 3:
この行は HISADDR への経路をデフォルトルートとして 追加するように ppp に指示します. HISADDR は IPCP で 決定されたゲートウェイの IP アドレスで置きかえられます.

詳細なサンプルについては, /usr/share/examples/ppp/ppp.conf.sample ファイル中のpmdemand エントリと /usr/share/examples/ppp/ppp.linkup.sample を参照してください.

バージョン 2 の ppp から「sticky routes」が導入されました. MYADDR や HISADDR を含む add コマンドと delete コマンドを記憶して, MYADDR や HISADDR のアドレスが変化した際には経路の再設定をおこないます. したがって, これらのコマンドを ppp.linkup に 繰り返し記述する必要は無くなりました.

20.2.1.5.3. かかってきた電話を ppp で受けるには

かかってきた電話を ppp が受けるように設定する際に, そのマシンが LAN に接続されているのであれば, パケットを LAN に転送するかどうかを決定する必要があります. 転送をおこなう場合には, その LAN のサブネットから IP アドレスを ppp クライアントに割り当て, 以下のコマンドを指定するのが良いでしょう.

```
gateway_enable=YES
```

20.2.1.5.3.1. どの getty を使いますか?

getty でダイアルアップサービスをおこなう場合の優れた解説が FreeBSD でダイアルアップサービスをおこなうための設定 にあります.

`getty` に代わるものとしては, mgetty があります. これは `getty` をより柔軟にしたもので, ダイアルアップ回線での使用を意図して 設計されています.

`mgetty` を使う場合の利点は, `mgetty` が積極的にモデムと通信する ということです. つまり, もし `/etc/ttys` でポートを閉じている場合, モデムは電話をとらなくなります.

最近のバージョンの `mgetty` (0.99beta 以降) では, PPP ストリームの 自動検出もサポートされています. これにより, クライアント側で スクリプトを準備しなくてもサーバに アクセスすることができます.

`mgetty` に関する, より詳細な情報については Mgetty と AutoPPP を参照してください.

20.2.1.5.3.2. ppp の実行許可

`ppp` は通常, ID 0 のユーザ (root) として動作しなければいけませんが, 以下で説明するように, `ppp` を通常のユーザとしてサーバモードで実行させたい 場合には, そのユーザを `/etc/group` の `network` グループに 追加して, ppp を実行する許可を与えておかなければいけません.

また, そのユーザが設定ファイル内の目的のエントリに アクセスできるように, 以下のように `allow` コマンドで許可を与えておく必要があります:

```
allow users fred mary
```

このコマンドがデフォルトエントリに 書かれている場合には, 指定されたユーザは すべてのエントリをアクセスできるようになります.

20.2.1.5.3.3. 動的 IP ユーザのための ppp シェルの設定

`/etc/ppp/ppp-shell` という名前で, 以下のような内容のファイルを 作成します:

```
#!/bin/sh
IDENT=`echo $0 | sed -e 's/^.*-\(.*\)$/\1/'`
CALLEDAS="$IDENT"
TTY=`tty`

if [ x$IDENT = xdialup -]; then
 IDENT=`basename $TTY`
fi

echo "PPP for $CALLEDAS on $TTY"
echo "Starting PPP for $IDENT"

exec /usr/sbin/ppp -direct $IDENT
```

このスクリプトには実行可能属性をつけておきます. 次に, 以下のコマンドを実行し, ppp-dialup という名前で このスクリプトへのリンクを作成します:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

すべてのダイアルアップ ppp ユーザのログインシェルとして このスクリプトを使用します. 以下は pchilds というユーザ名の ダイアルアップユーザを /etc/password へ登録した場合の例です. (パスワードファイルを直接エディタで編集したりせず, vipw を使ってください)

```
pchilds:*:1011:300:Peter Childs PPP:/home/ppp:/etc/ppp/ppp-dialup
```

任意のユーザが読むことのできる, /home/ppp ディレクトリを 作成します. /etc/motd が表示されないようにするため, このディレクトリには以下のように大きさが 0 バイトのファイルを 作成しておきます.

```
-r--r--r--   1 root     wheel           0 May 27 02:23 .hushlogin
-r--r--r--   1 root     wheel           0 May 27 02:22 .rhosts
```

20.2.1.5.3.4. 静的 IP ユーザのための PPP シェルの設定

上記と同じように ppp-shell ファイルを作成し, 静的な IP アドレスを割り当てるアカウントそれぞれについて ppp-shell へのシンボリックリンクを作成します.

例えば, クラス C ネットワークの経路制御を必要とする, 三人のダイアルアップユーザ fred, sam, mary がいるとすると, 以下のコマンドを実行することになります:

```
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-fred
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-sam
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-mary
```

これらのユーザのダイアルアップアカウントでは, 上で作成した それぞれのシンボリックリンクを ログインシェルとして設定しておきます. (つまり, ユーザ mary のログインシェルは /etc/ppp/ppp-mary に なります).

20.2.1.5.3.5. 動的 IP ユーザのための ppp.conf の設定

/etc/ppp/ppp.conf ファイルは, 大体以下のような内容になるでしょう:

```
default:
  set debug phase lcp chat
  set timeout 0

ttyd0:
  set ifaddr 203.14.100.1 203.14.100.20 255.255.255.255
  enable proxy

ttyd1:
  set ifaddr 203.14.100.1 203.14.100.21 255.255.255.255
```

```
  enable proxy
```

注記

上の例のように段をつける (インデントする) 必要があることに注意してください.

`default:` エントリはセッションごとにロードされます. `/etc/ttys` で有効にしてある各ダイアルアップ回線ごとに一つ, 上記の `ttyd0:` のようなエントリを作成します. 各行の相手側アドレスとして, それぞれ別の IP アドレスを 動的 IP ユーザのための IP アドレスのプールから割り当てておく必要があります.

20.2.1.5.3.6. 静的 IP ユーザのための **ppp.conf** の設定

上のサンプルの `/usr/share/examples/ppp/ppp.conf` の内容に加えて, 静的に IP を割り当てられたダイアルアップユーザ それぞれのためのエントリを追加する必要があります. ここでも `fred`, `sam`, `mary` の例を使うことにしましょう.

```
fred:
  set ifaddr 203.14.100.1 203.14.101.1 255.255.255.255

sam:
  set ifaddr 203.14.100.1 203.14.102.1 255.255.255.255

mary:
  set ifaddr 203.14.100.1 203.14.103.1 255.255.255.255
```

必要であれば, それぞれの静的 IP ユーザに対する経路制御情報も `/etc/ppp/ppp.linkup` ファイルに書いておくべきでしょう. 以下の例ではクライアントの PPP リンクを経由する, クラス C の `203.14.101.0` ネットワークへの経路を追加しています.

```
fred:
  add 203.14.101.0 netmask 255.255.255.0 HISADDR

sam:
  add 203.14.102.0 netmask 255.255.255.0 HISADDR

mary:
  add 203.14.103.0 netmask 255.255.255.0 HISADDR
```

20.2.1.5.4. **mgetty**, AutoPPP, マイクロソフト拡張の詳細

20.2.1.5.4.1. **mgetty** と AutoPPP

`AUTO_PPP` オプションつきでコンパイルした `mgetty` を使えば, `mgetty` が PPP 接続の LCP フェーズを検出して, 自動的に PPP シェルを起動するように 設定することがで

きます. しかし この場合, デフォルトの login/password シーケンスは発生しないので, ユーザの認証は PAP または CHAP を使っておこなう必要があります.

このセクションでは, ユーザ (あなた) が問題なく AUTO_PPP オプションつきの mgetty (v0.99beta またはそれ以降) の設定, コンパイル, インストールができているものと仮定しています.

/usr/local/etc/mgetty+sendfax/login.config ファイルが 以下の行を含んでいることを確認してください:

```
/AutoPPP/ -      -        /etc/ppp/ppp-pap-dialup
```

これにより, PPP 接続を検出したら mgetty が ppp-pap-dialup スクリプトを実行するようになります.

/etc/ppp/ppp-pap-dialup という名前で, 以下のような内容のファイルを 作成します (このファイルには実行可能属性を つけておく必要があります):

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap
```

さらに, かかってきた電話すべてを自分で扱うエントリを /etc/ppp/ppp.conf に作成します.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

この方法でログインする それぞれのユーザは, PAP によるユーザ認証を おこなうために /etc/ppp/ppp.secret ファイルにユーザ名とパスワードを 書いておくか, または /etc/password ファイルを使うように,

```
enable passwdauth
```

ユーザに静的な IP アドレスを割り当てる場合には, そのアドレスを /etc/ppp/ppp.secret の第三引数として指定することができます. サンプルについては, /usr/share/examples/ppp/ppp.secret.sample を参照してください.

20.2.1.5.4.2. マイクロソフト拡張

クライアントからの要求に応じて, ppp が DNS や NetBIOS ネームサーバの アドレスを通知するように 設定をおこなうこともできます.

バージョン 1.X の ppp で これらの拡張機能を有効にするには, 以下の行を /etc/ppp/ppp.conf の適切なセクションに追加する必要があるでしょう.

```
enable msext
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

バージョン 2 以降の ppp では, 以下のようになります:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

これにより, クライアントはプライマリと セカンダリのネームサーバアドレス および NetBIOS ネームサーバホストを知ることができます.

バージョン 2 以降の ppp では, `set dns` の行を省略した場合には `/etc/resolv.conf` に書かれているネームサーバのアドレスを使用します.

20.2.1.5.5. PAP および CHAP による認証

いくつかのプロバイダでは, PAP または CHAP のいずれかの認証メカニズムを 使用して接続時の認証をおこなうように システムを設定しています. この場合, プロバイダは接続の際に `login:` プロンプトを送信せず, 最初から PPP で通信を始めようとするでしょう.

PAP ではパスワードがそのまま送られてしまうため, CHAP に比べると安全性が 低くなりますが, このパスワードはシリアル回線のみを通して送られます. そのため, クラッカーが「盗み聞き」する余地は多くないので, 通常ここの セキュリティは問題にはなりません.

静的 IP アドレスによる PPP 接続または 動的 IP アドレスによる PPP 接続の セクションに戻って, 以下の変更をおこないます:

```
7       set login
…
12      set authname MyUserName
13      set authkey MyPassword
```

これまでと同様に, 行番号は取り除いておいてください. これは解説の際に参照する行を示すためにつけたものです. なお, 少なくともスペース 1 個分の段づけ (インデント) が必要です.

Line 7:
PAP または CHAP を使用する場合, 通常 プロバイダはサーバへの ログインを必要としません. そのため,「set login」文字列を 無効にしておかなければいけません.

Line 12:
この行は PAP/CHAP ユーザ名を指定します. *MyUserName* に 正しい値を入れておく必要があります.

Line 13:
この行は PAP/CHAP パスワードを指定します. *MyPassword* に 正しい値を入れておく必要があります. PAP と CHAP はデフォルトで両方とも 受け付けられるようになって いますが, PAP や CHAP を使用するという 意思を明示するために,

```
15      accept PAP
```

または

```
15      accept CHAP
```

という行を追加しておくのも良いでしょう.

20.2.1.5.6. 動作中の ppp の設定変更

適切な診断ポートが設定されている場合には, バックグラウンドで動作中の ppp プログラムと通信することができます. この設定をおこなうためには, 以下の行を設定ファイルに追加しておきます:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

これにより, ppp は指定された unix ドメインの ソケットをモニタして, クライアントから正しいパスワードを受け取った後に アクセスを許可します. このソケット名に含まれる %d は, この ppp が使用している tun デバイスの デバイス番号で置きかえられます.

一旦ソケットの設定が終了したら, スクリプト中で pppctl(8) を 使用して, 動作中の ppp を操作することができるでしょう.

20.2.1.6. システムの最終設定

これで ppp の設定は終りました. しかし ppp を動かす前に, まだ少し必要なことがあります. それらの設定は, すべて /etc/rc.conf ファイルを 編集することでおこないます. (このファイルは以前には /etc/sysconfig と呼ばれていました)

このファイルを上から順に設定していきます. まずは hostname= の行が設定されていることを確認します. 例えば以下のように:

```
hostname="foo.bar.com"
```

もしプロバイダが静的な IP アドレスとホスト名を割り当てているのなら, ホスト名としてそれを使うのが おそらくベストでしょう.

次に network_interfaces 変数を調べます. 必要に応じて (on demand) プロバイダにダイアルするようにシステムを設定したい場合には, tun0 デバイスがこのリストに追加されていることを確認しておきます. それ以外の場合には, tun0 デバイスをリストから削除しておきます.

```
network_interfaces="lo0 tun0" ifconfig_tun0=
```

注記

ifconfig_tun0 変数が空で, /etc/start_if.tun0 という名前の ファイルが作成されていなければなりません. このファイルの内容は以下のようになります.

```
ppp -auto mysystem
```

このスクリプトはネットワークの設定時に実行され, ppp デーモンを自動モードで立ち上げます. このマシンがもし LAN のゲートウェイであれば, `-alias` スイッチも使用したいと思うかもしれません. 詳細に関しては, マニュアルページを参照してください.

以下のようにルータプログラムを `NO` に設定します.

```
router_enable="NO"
```

`routed` は, `ppp` が作成したデフォルトのルーティングテーブル エントリを削除してしまう場合がありますので, (初期設定では起動されるようになっている) `routed` デーモンが 起動されないようにしておくことが重要です.

`sendmail_flags` 行が `-q` オプションを含まないように 設定しておいた方がよいでしょう. さもないと, `sendmail` が アドレスを調べようとして発信をおこなってしまう場合があります. 以下のような設定で良いでしょう:

```
sendmail_flags="-bd"
```

この結果, PPP リンクを立ち上げた時には いつでも以下のコマンドを実行して, キューにたまっているメールを `sendmail` に送信させる作業が必要になるでしょう.

```
# /usr/sbin/sendmail -q
```

`ppp.linkup` 中で `!bg` コマンドを使用することで, これを自動的に おこなうこともできます:

```
1     provider:
2       delete ALL
3       add 0 0 HISADDR
4       !bg sendmail -bd -q30m
```

こうするのが嫌であれば, SMTP トラフィックをブロックするように「dfilter」を設定しておくこともできます. 詳細についてはサンプルファイルを参照してください.

後はマシンをリブートするだけです.

リブートが終ったら,

```
# ppp
```

コマンドを実行し, 続いて PPP セッションを開始させるために `dial provider` と入力することもできますし, (`start_if.tun0` スクリプトを作成していない場合に), 外部

へのトラフィックが発生した時に, ppp が自動的に セッションを確立してくれるようにしたいのであれば, 以下のコマンドを実行することもできます.

```
# ppp -auto provider
```

20.2.1.7. まとめ

要約すると, 初めて ppp を設定する際には, 以下のステップが不可欠です:

クライアント側:

1. カーネルに tun デバイスが組み込まれていることを確認.
2. /dev ディレクトリに tunX デバイスファイルが 存在することを確認.
3. /etc/ppp/ppp.conf にエントリを作成. ほとんどのプロバイダでは, pmdemand の例で充分でしょう.
4. 動的 IP アドレスを使用するなら, /etc/ppp/ppp.linkup に エントリを作成.
5. /etc/rc.conf (または sysconfig) ファイルを更新.
6. 必要に応じてダイヤル (demand dialing) したいのであれば, start_if.tun0 スクリプトを作成.

サーバ側:

1. カーネルに tun デバイスが組み込まれていることを確認.
2. /dev ディレクトリに tunX デバイスファイルが 存在することを確認.
3. (vipw(8) コマンドを使って) /etc/passwd にエントリを作成.
4. このユーザのホームディレクトリに ppp -direct direct-server か何かを実行するプロファイルを作成.
5. /etc/ppp/ppp.conf にエントリを作成. direct-server の例で充分でしょう.
6. /etc/ppp/ppp.linkup にエントリを作成.
7. /etc/rc.conf ファイルを更新.

20.3. カーネル PPP の利用

原作: Gennady B. Sorokopud **<gena@NetVision.net.il >**, Robert Huff **<rhuff@cybercom.net >**.

訳: 石墨 紀孝 **<graphite@jp.FreeBSD.org >**. 1996 年 9 月 6 日.

20.3.1. カーネル PPP の設定

PPP の設定を始める前に, pppd が /usr/sbin にあり, また /etc/ppp というディレクトリが存在することを確認してください.

pppd はふたつのモードで動作します.

1. 「クライアント」モード. シリアル接続やモデムを利用して, そのマシンを 外部のネットワークに PPP 接続したい場合に用います.

2. 「サーバ」モード. そのマシンがネットワーク上にあるときに, PPP を使って ほかのコンピュータを接続する際に用います.

どちらの場合でも, オプションファイルを設定する必要があります (/etc/ppp/options または, そのマシン上で PPP を使用する人が 複数いる場合には ~/.ppprc).

また, ダイヤルとリモートホストへの接続をおこなうために, シリアル接続やモデムを 操作する, なんらかのソフトウェアが必要です (kermit が適しているでしょう).

20.3.2. PPP クライアントとしての動作

わたしは, CISCO ターミナルサーバの PPP 回線に接続するために, 下記のような /etc/ppp/options を使用しています.

```
crtscts         # enable hardware flow control
modem           # modem control line
noipdefault     # remote PPP server must supply your IP address.
                # if the remote host doesn't send your IP during IPCP
                # negotiation , remove this option
passive         # wait for LCP packets
domain ppp.foo.com      # put your domain name here

:<remote_ip>    # put the IP of remote PPP host here
                # it will be used to route packets via PPP link
                # if you didn't specified the noipdefault option
                # change this line to <local_ip>:<remote_ip>

defaultroute    # put this if you want that PPP server will be your
                # default router
```

接続方法:

1. kermit (またはその他のモデム操作プログラム) を使ってリモートホストに ダイヤルし, 接続してください. そして, あなたのユーザ名とパスワード (必要 であれば, その他にもリモートホストで PPP を有効にするための操作) を入力 します.

2. kermit を抜けてください. (回線を切断せずに)

3. 下記のように入力します:

```
# /usr/src/usr.sbin/pppd.new/pppd /dev/tty01 19200
```

(通信速度とデバイス名には, あなたの環境に適したものを入れてください)

これでこのコンピュータは PPP で接続されました. もし, なんらかの理由で 接続に失敗したならば, `/etc/ppp/options` ファイルに `debug` オプションを追加して, 問題点を突き止めるために, コンソールに表示される メッセージを調べてください.

下記の `/etc/ppp/pppup` スクリプトは, 上記の作業を すべて自動的におこないます:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" -] -; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" -] -; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi

ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.dial
pppd /dev/tty01 19200
```

`/etc/ppp/kermit.dial` は kermit 用のスクリプトで, ダイヤルして, リモートホストでの認証に必要なすべての処理をおこないます. (そのようなスクリプトの例は この文書の終わりに添付してあります)

PPP 接続を切断するには, 下記のような `/etc/ppp/pppdown` スクリプトを 使用します:

```
#!/bin/sh
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" -] -; then
        echo 'killing pppd, PID=' ${pid}
        kill -TERM ${pid}
fi

ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" -] -; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
```

```
fi

/sbin/ifconfig ppp0 down
/sbin/ifconfig ppp0 delete
kermit -y /etc/ppp/kermit.hup
/etc/ppp/ppptest
```

PPP が動作中かどうかを調べます (`/usr/etc/ppp/ppptest`):

```
#!/bin/sh
pid=`ps ax| grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" -] -; then
        echo 'pppd running: PID=' ${pid-NONE}
else
        echo 'No pppd running.'
fi
set -x
netstat -n -I ppp0
ifconfig ppp0
```

モデム回線を切断します (`/etc/ppp/kermit.hup`):

```
set line /dev/tty01 ; put your modem device here
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
echo \13
exit
```

次は `kermit` の代わりに `chat` を使う方法です.

原作: Robert Huff **<rhuff@cybercom.net >**.

pppd 接続を確立するためには, 次の二つのファイルの設定だけで十分です.

`/etc/ppp/options` :

```
/dev/cuaa1 115200

crtscts  # enable hardware flow control
```

```
modem  # modem control line
connect "/usr/bin/chat -f /etc/ppp/login.chat.script"
noipdefault # remote PPP serve must supply your IP address.
         # if the remote host doesn't send your IP during
                # IPCP negotiation, remove this option
passive         # wait for LCP packets
domain <your.domain> # put your domain name here

:  # put the IP of remote PPP host here
         # it will be used to route packets via PPP link
                # if you didn't specified the noipdefault option
                # change this line to <local_ip>:<remote_ip>

defaultroute # put this if you want that PPP server will be
         # your default router
```

`/etc/ppp/login.chat.script` :

注記

(実際には一行になります.)

```
ABORT BUSY ABORT 'NO CARRIER' "" AT OK ATDT<phone.number>
 CONNECT "" TIMEOUT 10 ogin:-\\r-ogin: <login-id>
 TIMEOUT 5 sword: <password>
```

正しくインストールし編集した後は, 必要な事はこれだけです

```
# pppd
```

このサンプルは主に Trev Roydhouse <Trev.Roydhouse@f401.n711.z3.fidonet.org> から寄せられた情報に基づいており, 承諾を得て使用しています.

20.3.3. PPP サーバとしての動作

`/etc/ppp/options` :

```
crtscts                         # Hardware flow control
netmask 255.255.255.0           # netmask ( not required )
192.114.208.20:192.114.208.165  # ip's of local and remote hosts
                                # local ip must be different from one
                                # you assigned to the ethernet ( or ↺
other )
                                # interface on your machine.
                                # remote IP is ip address that will ↺
be
```

```
                                # assigned to the remote machine
domain ppp.foo.com              # your domain
passive                         # wait for LCP
modem                           # modem line
```

下記のような /etc/ppp/pppserv スクリプトで, そのマシンを PPP サーバにすることができます.

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" -] -; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" -] -; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi

# reset ppp interface
ifconfig ppp0 down
ifconfig ppp0 delete

# enable autoanswer mode
kermit -y /etc/ppp/kermit.ans

# run ppp
pppd /dev/tty01 19200
```

PPP サーバを終了するには, この /etc/ppp/pppservdown スクリプト を使用します:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" -] -; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" -] -; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi
ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.noans
```

下記の kermit スクリプトは，モデムの自動応答機能を有効，または無効にします（/etc/ppp/kermit.ans ）:

```
set line /dev/tty01
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
inp 5 OK
echo \13
out ATS0=1\13   -; change this to out ATS0=0\13 if you want to ↺
disable
                -; autoanswer mod
inp 5 OK
echo \13
exit
```

この /etc/ppp/kermit.dial スクリプトは，リモートホストに ダイヤルし，認証手続きをするのに使用します．あなたは必要に応じて，これを 変更しないといけないでしょう．あなたのユーザ名とパスワードをこの スクリプトに書かなければいけませんし，モデムやリモートホストからの 応答によっては，入力待ちの文を変更する必要もあります．

```
;
; put the com line attached to the modem here:
;
set line /dev/tty01
;
; put the modem speed here:
;
set speed 19200
set file type binary            -; full 8 bit file xfer
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none
set modem hayes
set dial hangup off
```

```
set carrier auto                -; Then SET CARRIER if necessary,
set dial display on             -; Then SET DIAL if necessary,
set input echo on
set input timeout proceed
set input case ignore
def \%x 0                       -; login prompt counter
goto slhup

:slcmd                          -; put the modem in command mode
echo Put the modem in command mode.
clear                           -; Clear unread characters from ↺
input buffer
pause 1
output +++                      -; hayes escape sequence
input 1 OK\13\10                -; wait for OK
if success goto slhup
output \13
pause 1
output at\13
input 1 OK\13\10
if fail goto slcmd              -; if modem doesn't answer OK, try ↺
again

:slhup                          -; hang up the phone
clear                           -; Clear unread characters from ↺
input buffer
pause 1
echo Hanging up the phone.
output ath0\13                  -; hayes command for on hook
input 2 OK\13\10
if fail goto slcmd              -; if no OK answer, put modem in ↺
command mode

:sldial                         -; dial the number
pause 1
echo Dialing.
output atdt9,550311\13\10               -; put phone number here
assign \%x 0                    -; zero the time counter

:look
clear                           -; Clear unread characters from ↺
input buffer
increment \%x                   -; Count the seconds
input 1 {CONNECT }
if success goto sllogin
reinput 1 {NO CARRIER\13\10}
if success goto sldial
reinput 1 {NO DIALTONE\13\10}
if success goto slnodial
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
```

```
if < \%x 60 goto look
else goto slhup

:sllogin                        -; login
assign \%x 0                    -; zero the time counter
pause 1
echo Looking for login prompt.

:slloop
increment \%x                   -; Count the seconds
clear                           -; Clear unread characters from ↺
input buffer
output \13
;
; put your expected login prompt here:
;
input 1 {Username: }
if success goto sluid
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 10 goto slloop         -; try 10 times to get a login prompt
else goto slhup                 -; hang up and start again if 10 ↺
failures

:sluid
;
; put your userid here:
;
output ppp-login\13
input 1 {Password: }
;
; put your password here:
;
output ppp-password\13
input 1 {Entering SLIP mode.-}
echo
quit

:slnodial
echo \7No dialtone.  Check the telephone line!\7
exit 1

; local variables:
; mode: csh
; comment-start: "; "
; comment-start-skip: "; "
; end:
```

20.4. PPP オーバイーサネット (PPPoE) の利用

原作: Jim Mock (node.to より) 10 Jan 2000.

以下の解説は, PPPoE として知られる, PPP オーバイーサネットの設定法です.

20.4.1. 必要なもの

あなたのシステムで PPPoE を適切に機能させるためには, 以下のものが必要です.

- FreeBSD 3.4やそれより新しいバージョンのカーネルソース
- FreeBSD 3.4やそれより新しいバージョンのppp

20.4.2. カーネルコンフィギュレーション

以下に示すオプションをカーネルコンフィギュレーションファイルに 追加して, その後 新しいカーネルを コンパイルする必要があります.

- options NETGRAPH

以下は任意

- options NETGRAPH_PPPOE
- options NETGRAPH_SOCKET

この機能は実行時には有効ではありませんが, 要求に応じて ppp は関係のあるモジュールを 読み込みます.

20.4.3. `ppp.conf` の設定

これは動作している `ppp.conf` の 例です:

```
default: # or name_of_service_provider
  set device PPPoE:xl1 # replace xl1 with your ethernet device
  set mru 1492
  set mtu 1492
  set authname YOURLOGINNAME
  set authkey YOURPASSWORD
  set log Phase tun command # you can add more detailed logging if ↺
you wish
  set dial
  set login
  set ifaddr 10.0.0.1/0 10.0.0.2/0
  add default HISADDR
  nat enable yes # if you want to enable nat for your local net

papchap:
  set authname YOURLOGINNAME
  set authkey YOURPASSWORD
```

-natオプションを付けてPPPoE を起動する際には注意するべきです.

20.4.4. PPP の起動

以下を root 権限において実行することで, 起動させることができます:

```
# ppp -ddial name_of_service_provider
```

20.4.5. システム起動時に PPP を立ち上げる

/etc/rc.conf ファイルに以下の行を追加 してください:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES"
ppp_profile="default" # or your provider
```

20.5. SLIP の利用

原作: 浅見 賢 <asami@FreeBSD.org >,Guy Helmer, 協力: Wilko Bulte, Piero Serini.

訳: 花井 浩之 <hanai@FreeBSD.org > 1996 年 8 月 8 日.

20.5.1. SLIPクライアントのセットアップ

ここには FreeBSD マシンを静的アドレスのネットワークにつなげる場合の SLIPのセットアップの一つの方法を書いてあります. ホスト名を動的に割り当てる(つまり, ダイヤルアップするたびにアドレスが かわる)ためには, おそらくもっと凝ったことが必要です.

まず, モデムがどのシリアルポートにつながっているか決めましょう. 私は /dev/cuaa1 から /dev/modem へというシンボリックリンクを張り, コンフィグレーションではその名前だけを使っています. /etc や .kermrc など, システム全体に散らばっているファイルを修正する 必要がでるとまったく煩わしいのです!

注記

ここで, /dev/cuaa0 は COM1であり, cuaa1はCOM2です.

カーネルのコンフィグレーションファイルに

```
pseudo-device   sl      1
```

という記述があるのを確認してください. これは GENERIC カーネルに含まれている ので削除していない限り大丈夫でしょう.

20.5.1.1. 最初の設定

1. /etc/hosts ファイルにあなたのマシンのゲートウェイとネームサーバ を加えてください. 私のは以下のようになっています.

```
127.0.0.1               localhost loghost
136.152.64.181          silvia.HIP.Berkeley.EDU silvia.HIP silvia
136.152.64.1            inr-3.Berkeley.EDU inr-3 slip-gateway
128.32.136.9            ns1.Berkeley.edu ns1
128.32.136.12           ns2.Berkeley.edu ns2
```

2. /etc/host.conf ファイル中で hostsがbind よりも前にあること を確認してください. さもないとヘンなことが起こるかもしれません.

3. /etc/rc.conf ファイルを編集してください. なお, お使いの FreeBSD が 2.2.2 よりも前のバージョンのものの場合は, /etc/sysconfig を編集してください.

   1. 行

   ```
   #hostname=myname.my.domain#
   ```

   を編集してホスト名をセットしてください. 完全なInternetホスト名を与えるべきです.

   2. 行

   ```
   #network_interfaces="lo0"#
   ```

   を

   ```
   #network_interfaces="lo0 sl0"#
   ```

   へ変更することにより ネットワークインタフェースのリストに sl0 を加えてください.

   3. 行

   ```
   ifconfig_sl0="inet ${hostname} slip-gateway netmask ↺
   0xffffff00 up"
   ```

   を加えて sl0 のスタートアップフラグをセットしてください.

   4. 行

   ```
   #defaultrouter=NO#
   ```

   を

```
#defaultrouter=slip-gateway#
```

へ変更してデフォルトのルータを 指定してください.

4. 次の

```
domain HIP.Berkeley.EDU
nameserver 128.32.136.9
nameserver 128.32.136.12
```

という内容を含むファイル `/etc/resolv.conf` を作ってください. 見ればわかるように, これらはネームサーバホストを設定しています. もちろん, 実際のドメイン名やアドレスは あなたの環境に依存します.

5. root と toor (及びパスワードを持っていない他のアカウントすべて) のパスワードを設定してください. passwdコマンドを使いましょう. `/etc/passwd` や `/etc/master.passwd` といったファイルを編集してはいけません!

6. マシンを再起動して正しいホスト名で 立ち上がることを確認してください.

20.5.1.2. SLIP接続をおこなう

1. モデムを起動, つながったらプロンプトで `slip`とタイプし, マシン名と パスワードを入力してください. 入力する必要があるものは環境に よって異なります. 私は次のようなスクリプトでkermitを使っています.

```
# kermit setup
set modem hayes
set line /dev/modem
set speed 115200
set parity none
set flow rts/cts
set terminal bytesize 8
set file type binary
# The next macro will dial up and login
define slip dial 643-9600, input 10 =>, if failure stop, -
output slip\x0d, input 10 Username:, if failure stop, -
output silvia\x0d, input 10 Password:, if failure stop, -
output ***\x0d, echo \x0aCONNECTED\x0a
```

(もちろん, ホスト名とパスワードは変える必要があります). 接続するためには kermit のプロンプトで `slip`とタイプするだけです.

注記

ファイルシステムのどんなところにもプレインテキスト　にパスワードを書いておくのは一般的にはよくありません．覚悟の上で やってください．私は単に不精なだけです．

2. ここでkermitから抜け出し (zでkermitをサスペンドできます), root で

```
# slattach -h -c -s 115200 /dev/modem
```

と入力しましょう．もしルータの向う側のホストへ ping できるなら接続成功です! もしうまく いかなければslattachへの引数として -c の代わりに-aとやってみてください．

20.5.1.3. 接続の切り方

slattachを殺すためにrootで

```
# kill -INT `cat /var/run/slattach.modem.pid`
```

とタイプしてください．そして kermit に戻り (もしkermitをサスペンドしていたなら fg), kermitから抜けてください (q).

slattachのマニュアルページにはインタフェースを落すために　ifconfig　sl0 downをしなければいけないと書いていますが,　私には差がないように見えます．(ifconfig sl0 とやっても同じ結果が得られる.)

時にはモデムがキャリアを落すのを　拒絶するかもしれません(私のは　よくそうなります)．その時は単にkermitをスタートしてまた終了 してください．普通は2回目で落ちます．

20.5.1.4. トラブルシューティング

もし動かなければ自由に私に質問してください．今までいろんな人がつまずいた のは次のようなことです．

- slattach で -c や -a を使わなかった(私はなぜこれが致命的になり得るのか わかりませんが, このフラグを付けることで少なくとも一人の 問題は解決しました.)
- sl0 の代わりに s10 を使った(いくつかのフォントでは見分けるのは難しい かもしれません).
- インタフェースの状態を見るために ifconfig sl0 をやってみてください．私は,

```
# ifconfig sl0
```

```
sl0: flags=10<POINTOPOINT>
        inet 136.152.64.181 --> 136.152.64.1 netmask ffffff00
```

となります.

- また, pingが「no route to host」というメッセージを返す時には `netstat -r` でルーティングテーブルを確認しましょう. 私のは,

```
# netstat -r
Routing tables
Destination      Gateway            Flags     Refs     Use  IfaceMTU
    Rtt
Netmasks:
(root node)
(root node)

Route Tree for Protocol Family inet:
(root node) =>
default          inr-3.Berkeley.EDU UG          8   224515  sl0 -
    -
localhost.Berkel localhost.Berkeley UH          5    42127  lo0 -
     0.438
inr-3.Berkeley.E silvia.HIP.Berkele UH          1        0  sl0 -
    -
silvia.HIP.Berke localhost.Berkeley UGH        34 47641234  lo0 -
     0.438
(root node)
```

となります. (これはたくさんのファイルを転送した後でのもので, あなたの見る数字はもっと小さいかも しれません).

20.5.2. SLIPサーバのセットアップ方法

訳: 冨田 重成 **<ts@icu.ac.jp >**. 1996 年 9 月 6 日.

この文書の目的は, SLIPサーバ機能を FreeBSDシステムのもとで設定するため の助言を提供することです. SLIPサーバ機能を設定するということは, リモートの SLIPクライアントがログインできるようにするために, 自動的に接続処 理をおこなうようにすることです. この文書は著者の経験に基づいておりますが, 実際のシステム構成や要望は異なりますから, すべての疑問にこの文書が答え ることはできません. なお, ここでの助言を試みた結果, あなたのシステムへ の悪影響やデータの損失が生じたとしても, 著者が責任を持つことはできませ んのでご了解をお願いします.

20.5.2.1. 前提

この文書の内容はテクニカルなものなので, 前提知識が必要です. すなわち, TCP/IPネットワークプロトコルについての知識, 特に, ネットワークとノード のアドレス指定をはじめ, ネットワークアドレスマスク, サブネット化, ルー ティング, および RIPなどのルーティングプロトコルなどに関する知識を前提 としています. ダイヤルアップサーバで SLIP機

能を設定するためには, これ らの概念についての知識が必要ですから, もし不案内であると思われる方は, O'Reilly & Associates, Inc.から出版されている Craig Hunt氏の TCP/IP Network Administration (ISBN 0-937175-82-X)か, または Douglas Comer氏の TCP/IPプロトコルに関する一連の書籍をお読みください.

前提知識に加え, さらに, モデムの設定が完了しており, そのモデムを経由し てログインできるように, システムファイル群が適切に記述できているものと 仮定しています. もしモデムの準備ができていないときには, あらかじめダイヤ ルアップ機能の設定についてのチュートリアルをお読みください. Webブラ ウザが使えるのであれば http://www.FreeBSD.org/ におけるチュー トリアルの一覧を調べてください. あるいは, この文書を見つけた場所を調べ て, `dialup.txt` やそれに類似した名前の文書をお読みください. 関連す るマニュアルページとしては, シリアルポート向けデバイスドライバについて の sio(4) をはじめ, モデムからのログインを 受理できるようにシステ ムを設定するための ttys(5), gettytab(5), getty(8), init(8) など, さらには, シリアルポート関連パラメータ (たと えば直接接続シリアルインタフェースの `clocal`) についての stty(1) なども助けになるかもしれません.

20.5.2.2. 概要

一般的な設定内容で FreeBSDを SLIPサーバとして利用すると, その動作は次 のようになります. まず, SLIPユーザが FreeBSD による SLIPサーバへ電話し て, SLIP専用IDでログインします. なお, このIDを持ったユーザはシェルとし て `/usr/sbin/sliplogin` を使います. この `sliplogin` は, ファイル `/etc/sliphome/slip.hosts` の中から, ログインIDと一致する 記述行を探します. もし一致する行があれば, ログインしたシリアル回線を, 利用可能な SLIPインタフェースへ接続し, その後にシェルスクリプト `/etc/sliphome/slip.login` で SLIPインタフェースを設定します.

20.5.2.2.1. SLIPサーバへのログイン例

仮に SLIPユーザIDが `Shelmerg` とします. すると, `/etc/master.passwd` における `Shelmerg` のエントリは次のよ うなものになります (実際には一つの行に続いている) .

```
Shelmerg:password:1964:89::0:0:Guy Helmer - SLIP:/usr/users/
Shelmerg:/usr/sbin/sliplogin
```

`Shelmerg` がログインすると, `sliplogin` は, ファイル `/etc/sliphome/slip.hosts` からユーザIDと一致する行を探しま す. いま仮に, `/etc/sliphome/slip.hosts` に次のような記述がなされていたとします.

```
Shelmerg        dc-slip sl-helmer       0xfffffc00      autocomp
```

`sliplogin` が上記のエントリを見つけると, Shelmerg が使用して いるシリアル回線を, 利用可能な SLIPインタフェースのなかの最初のものへ 接続し, 次の内容の `/etc/sliphome/slip.login` を実行します.

```
/etc/sliphome/slip.login 0 19200 Shelmerg dc-slip sl-helmer ↺
0xfffffc00 autocomp
```

もし上記の手順が正常に処理されると, `/etc/sliphome/slip.login` は, `sliplogin` が割り当てた SLIPインタフェース (この例では `slip.login` で与えられたパラメータのうちで最初の値である SLIP インタフェース0である) に対して `ifconfig` を実行し, ローカル IPアドレス (`dc-slip`)をはじめ, リモート IPアドレス (`sl-helmer`), SLIPインタフェースへのネットワークマスク (`0xfffffc00`), およびその他のフラグ (`autocomp`)を設定 します. 逆に, さきほどの手順が正常に終了しなかった場合, 通常は `sliplogin` は十分な情報を syslog の `daemon` 機能経由で `/var/log/messages` へ記録します (syslogd(8) や syslog.conf(5) のマニュアルページを参照のうえ, さらに `/etc/syslog.conf` を調べて `syslogd` がどのファイルへ記録するかを確認のこと) .

例はこのくらいにして, さっそくシステムのセットアップを始めてみましょう.

20.5.2.3. カーネルのコンフィグレーション

FreeBSD のデフォルトのカーネルには, 通常, 二つの SLIPインタフェースが 準備されています (`sl0` と `sl1`) . これらのインタフェー スが使用中のカーネルに準備されているかどうかを調べるには, `netstat -i` を実行してください.

`netstat -i` の出力例

```
Name  Mtu   Network     Address              Ipkts Ierrs    Opkts Oerrs
  Coll
ed0   1500  <Link>0.0.c0.2c.5f.4a         291311     0   174209     ↺
0   133
ed0   1500  138.247.224 ivory            291311     0   174209     ↺
0   133
lo0   65535 <Link>                          79     0       79     ↺
0     0
lo0   65535 loop        localhost           79     0       79     ↺
0     0
sl0*  296   <Link>                         0     0        0     ↺
0     0
sl1*  296   <Link>                         0     0        0     ↺
0     0
```

`netstat -i` の出力に `sl0` と `sl1` のインタフェー スが含まれているということから, カーネルには二つの SLIPインタフェー スが組み込まれているということを示しています. (`sl0` と `sl1` に付いたアスタリスクは, `netstat -i` の実行時点で はインタフェースが「ダウン」していることを表しています.)

なお, パケットのフォワード機能は FreeBSD のデフォルトのカーネルでは設定 されていません (すなわちルータとしては動作しない) . もしインターネット 接続ホストについての RFC要件 (RFC 1009 [Requirements for Internet Gateways] と 1122

[Requirements for Internet Hosts — Communication Layers], おそらく 1127 [A Perspective on the Host Requirements RFCs] も) に準拠して, FreeBSDによる SLIPサー バをルータとして動作させたいときには, /etc/rc.conf (バージョン 2.2.2 より前の FreeBSD では /etc/sysconfig) ファイル の gateway_enable 変数を YES としてください. もし古いシステ ムで /etc/sysconfig ファイルすらないときには, 次のコマン ドを /etc/rc.local へ追加してください.

```
sysctl -w net.inet.ip.forwarding = 1
```

この新しい設定を有効とするには, リブートする必要があります.

デフォルトのカーネルコンフィグレーションファイル (/sys/i386/conf/GENERIC) の最後の部分に, 次のような行がありま す.

```
pseudo-device sl 2
```

この行によって, 使用可能な SLIPデバイスの総数が決まります. すなわち, 行 末の数値が, 同時に動作可能な SLIP接続の最大数となります.

カーネルの再構築については, FreeBSDカー ネルのコンフィグレーション を参照ください.

20.5.2.4. Sliploginのコンフィグレーション

すでにご説明したように, /usr/sbin/sliplogin のコンフィグレー ションのために, 3種類のファイルが/etc/sliphome ディレクトリに あります (sliplogin についての実際のマニュアルページとしては sliplogin(8) を参照のこと) . ファイル slip.hosts は SLIPユーザおよびその IPアドレスを決めます. 通常, ファイル slip.login は, SLIPインタフェースを設定することだけに使 用します. slip.logout はオプションのファイルで, slip.login で設定した内容を, シリアル接続が終了した時点で解除 するときに使用します.

20.5.2.4.1. slip.hosts のコンフィグレーション

/etc/sliphome/slip.hosts には, 少なくとも 4 つの項目をホワイ トスペース (スペースやタブ) で区切って指定します.

- SLIPユーザのログインID
- SLIPリンクのローカル (SLIPサーバ側) アドレス
- SLIPリンクのリモートアドレス
- ネットワークマスク

ホスト名をローカルおよびリモートのアドレスとして 記述できます (IPアドレ スの決定は, /etc/host.conf の指定内容に応じて, /etc/hosts か DNSのいずれかによっ

て決定される) . また, ネット ワークマスクも /etc/networks ファイルに記述された名前を参照す ることで, 指定することもできると思います. これまでの例としてあげたシス テムでの /etc/sliphome/slip.hosts は次のようになります.

```
#
# login local-addr      remote-addr     mask            opt1    opt2
#                                                 ↺
(normal,compress,noicmp)
#
Shelmerg  dc-slip       sl-helmerg      0xfffffc00      autocomp
```

それぞれの行の最後には, 次に示すオプションを一つ以上指定できます.

- normal — ヘッダを圧縮しない
- compress — ヘッダを圧縮する
- autocomp — リモートの設定に応じて, ヘッダを圧縮する
- noicmp — ICMPパケットを禁止する (「ping」パケットは送出されず, バンド幅を占有しない)

なお, FreeBSDバージョン2の初期リリースの sliplogin は, 旧 FreeBSD 1.x では有効であった上記のオプションを無視していましたので, normal, compress, autocomp, そして noicmp などのオプションは FreeBSD 2.2でサポートされるまでは効果がありませんでした (た だしこれらのフラグを使うためには slip.login スクリプトへ記述する 必要がある) .

SLIPリンクでのローカルとリモート向けのアドレスの 選び方は, TCP/IPサブネッ トを専用に割り当てるか, または「プロキシ ARP」を SLIPサーバへ用いるかによって違います (「プロキシ ARP」という用語のここでの使い方は本来のものではないが, 説明のためにこの用語を使う) . もし, どちらの方式を選ぶべきか判らなかったり, IPアドレスの割り当て方が不明のときには, 上述の 前提 の節で紹介した TCP/IP関連書籍を参考になさるか, またはあなたの IPネットワークを管理している方に相談なさると よいでしょう.

独立したサブネットを SLIPクライアントへ適用するときには, すでに割り当てられている IPネットワーク番号の範囲からサブネット番号を割り当て, 同 時にそのサブネットの範囲内で有効な IPアドレスを SLIPクライアントの IP 番号として割り当てる必要があります. さらに, この SLIPサブネットから SLIPサーバを経由して最も近い IPルータへの経路を静的に設定するか, または gated を FreeBSDによる SLIPサーバへインストールして, 適当 なルーティングプロトコルを使って, SLIPサーバ経由のサブネットへの経路情 報をルータ群へ通知できるように設定するか, のいずれかをおこなう必要があります.

「プロキシ ARP」方式を採用するときには, SLIPクライアント向けの IPアドレス として, SLIPサーバのサブネットの範囲から 選んで割り当てるとともに, arp(8) コマンドを使うために /etc/sliphome/slip.login と/etc/sliphome/slip.logout のス

クリプトを修正して, SLIPサー バにおける ARPテーブル内のプロキシ ARPエントリへ 反映させる必要がありま す.

20.5.2.4.2. slip.login のコンフィグレーション

ファイル /etc/sliphome/slip.login の一般的な内容は次にようになります.

```
#!/bin/sh -
#
# @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
```

この slip.login ファイルの役目は単に, SLIPインタフェースにつ いてのローカルとリモートのアドレス, およびそのネットワークマスクを ifconfig コマンドで設定することです.

もし「プロキシ ARP」方式を採用する (SLIPクライアントへ独立したサブネットを使わない) ときには, ファイル /etc/sliphome/slip.login は次のような内容になります.

```
#!/bin/sh -
#
# @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
# Answer ARP requests for the SLIP client with our Ethernet addr
/usr/sbin/arp -s $5 00:11:22:33:44:55 pub
```

この slip.login で追加された行 arp -s $5 00:11:22:33:44:55 pub は, SLIPサーバにおける ARPテーブルへ新たなエントリを作ります. SLIPサーバ は, この ARPエントリが作られると, SLIPクライアントの IPアドレスと話し たい他の IPノードが要求してきたときにはいつも, SLIPサーバ の Ethernet MACアドレスを返すようになります.

上記の例を実際に流用なさるときには, 例にある Ethernet MACアドレス (00:11:22:33:44:55) を, あなたのシステムの実際のEthernetカー ドの MAC

アドレスと置き換えなければ 「プロキシ ARP」 はうまく動作しません! SLIPサーバの Ethernet MACアドレスを調べるには `netstat -i` コマ ンドを利用してください. 実行結果の第2行は次のようなものになるはずです.

```
ed0   1500  <Link>0.2.c1.28.5f.4a         191923     0   129457     ↺
0   116
```

この例での Ethernet MACアドレスは `00:02:c1:28:5f:4a` であると 読みます. なお arp(8) における MAC アドレスの指定に際しては, コマンド `netstat -i` が付けた Ethernet MACアドレスのピリオド記 号をコロン記号と置き換え, かつ単一桁の 16 進数にはゼロを先頭に加える必 要があります. この指定についての正確な情報は arp(8) を参照く ださい.

注記

`/etc/sliphome/slip.login` と `/etc/sliphome/slip.logout` を作成したならば, ファイル属性の「実行」ビット (すなわち `chmod 755 /etc/sliphome/slip.login /etc/sliphome/slip.logout`) を 設定しなければなりません. さもなければ `sliplogin` が うまく実行されません.

20.5.2.4.3. `slip.logout` のコンフィグレーション

ファイル `/etc/sliphome/slip.logout` は必ずしも必要なものではあ りません (ただし「プロキシ ARP」を利用する場合を除く) . もしこのファイルを 作成するときには, 次に示す標準的な `slip.logout` スクリプト例を 参考にしてください.

```
#!/bin/sh -
#
# slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
```

「プロキシ ARP」を利用する場合, この `/etc/sliphome/slip.logout` を 使って, 特定の SLIPクライアント向けの ARPエントリを削除したくなるようなときがあります.

```
#!/bin/sh -
#
#       @(#)slip.logout
```

```
#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
# Quit answering ARP requests for the SLIP client
/usr/sbin/arp -d $5
```

コマンド `arp -d $5` は, SLIPクライアントがログインした 際に,「プロキシ ARP」を使った `slip.login` によって追加され た ARPエントリを削除します.

これによって, 繰り返して利用することができるわけです. 必ず, `/etc/sliphome/slip.logout` を作成した後に, 実行ビットを設定し てください (`chmod 755 /etc/sliphome/slip.logout`) .

20.5.2.5. ルーティングについての考慮点

「プロキシ ARP」方式を利用せずに SLIPクライアントとその他のネットワーク (Internetも含む) の構成要素との間でパケットをルーティングするときには, SLIPサーバ経由でSLIPクライアントが属するサブネットまでの経路を, 最も 近いデフォルトのルータ群へ静的な経路情報として 追加しなければならないか, または `gated` を FreeBSDによるSLIPサーバへインストールして, SLIP サブネットについての経路情報を, 適当なルーティングプロトコルでルー タ群へ通知できるように設定するか, のどちらかをおこなわなければなりません.

20.5.2.5.1. 静的な経路

静的な経路を最も近いデフォルトの ルータ群へ追加することが困難なことがあ ります (経路情報を追加できる権限がなければそもそも不可能となる). もし あなたの組織に複数のルータで構成された ネットワークがあるならば, ある種 のルータ (たとえば CiscoやProteonなど) は, 静的な経路を SLIPサブネットへ使うようにルータを設定しなければならないだけでなく, その静的経路を 他のどのルータへ知らせるのかもあらかじめ 指定しておく必要がありますから, 静的経路に基づくルーティングを軌道に乗せるには それなりの専門的技術やト ラブルシューティングやコツが必要だと思います.

20.5.2.5.2. `gated` の稼働

静的経路についての頭痛への代替手段は, `gated` を FreeBSDによる SLIPサー バへインストールして, 適切なルーティングプロトコル (RIP/OSPF/BGP/EGP) を使って SLIPサブネットについての経路情報を他のルータへ知らせるように 設定することです. ports コレクションから `gated` を用いることもできますし, GateD 匿名 FTP サイト から探して自分自身で構築することもで きます. この文章を執筆時点の最新バージョンは `gated-R3_5Alpha_8.tar.Z` であり, このファイル「だけで」FreeBSDで 動作させることができます. `gated` についてのすべての情報と文書 は Merit GateD コン

ソーシアム からはじまる Web 上で入手できます. gated のコンパイルとインストールを行ったならば, 独自の 設定のために /etc/gated.conf ファイルを記述してください. 次の 例は, 筆者が FreeBSDによる SLIP サーバで使っている内容と類似のものです.

```
#
# gated configuration file for dc.dsu.edu; for gated version ↺
3.5alpha5
# Only broadcast RIP information for xxx.xxx.yy out the ed Ethernet ↺
interface
#
#
# tracing options
#
traceoptions "/var/tmp/gated.output" replace size 100k files 2 ↺
general -;

rip yes {
 interface sl noripout noripin -;
 interface ed ripin ripout version 1 -;
 traceoptions route -;
} -;

#
# Turn on a bunch of tracing info for the interface to the kernel:
kernel {
 traceoptions remnants request routes info interface -;
} -;

#
# Propagate the route to xxx.xxx.yy out the Ethernet interface via ↺
RIP
#

export proto rip interface ed {
        proto direct {
                xxx.xxx.yy  mask 255.255.252.0 metric 1; # SLIP ↺
connections
        } -;
} -;

#
# Accept routes from RIP via ed Ethernet interfaces

import proto rip interface ed {
        all -;
} -;
```

この gated.conf ファイルの例では, SLIPのサブネット *xxx.xxx.yy* についての経路情報を RIPを使って Ethernetへブロー ドキャストしています. もし ed ドライバ以外の Ethernetドライバを使うのであれば, ed インタフェースの記述を適切なものに置き換えてくだ さい. またこの例では, gatedの動作をデバッグするために, /var/tmp/

`gated.output` へトレース情報を出力するように指示して います. `gated` が希望通りに動作したならば, このトレースオプショ ンを止めることができます. なお, 例における *`xxx.xxx.yy`* を, あ なた自身の SLIPサブネットのネットワークアドレスに換えてください (また `proto direct` 部分のネットワークマスクも換えることを忘れないこ と) .

`gated` のコンパイルとインストールが終了し, コンフィグレーショ ンファイルの作成も完了したら, FreeBSDシステムではデフォルトの `routed` に代わって `gated` を起動してください. そのため には, `/etc/netstart` の `routed/gated` 起動パラメータを 適切な値に設定してください. `gated` のコマンドラインパラメータにつ いての情報は, `gated` のマニュアルページを参照してください.

第21章 電子メール

原作: Bill Lloyd.
改訂: Jim Mock.

訳: 田中 美穂子 <Mihoko_Tanaka@yokogawa.co.jp >. 14 January 1997.

21.1. この章では

「電子メール」、email としてのほうが知られているでしょう、は現代で最も広く利用されているコミュニケーション手段の一つです。この章では FreeBSD 上でメールサーバを実行するための基本的な導入を説明します。しかし、この文書は完璧な参考文献ではなく、実際のところ考慮すべき重要な点の多くが省略されています。この件について、より網羅したものについては 付録B 参考図書 に掲載されている多くの優れた書籍を参照してください。

この章では、以下の分野について説明します。

- 電子メールの送受信に関係しているソフトウェアの構成要素
- FreeBSD における sendmail の基本的な設定ファイルのある場所
- スパマーがあなたのメールサーバを踏台として不正に使用することを防ぐ方法
- あなたのシステムに sendmail の置き換えとなる代替の MTA をインストールして設定する方法
- メールサーバにまつわる共通の問題の解決法
- UUCP とともに SMTP を使う方法
- ダイアルアップ接続でメールを使う方法
- セキュリティを向上するために SMTP 認証を設定する方法

この章を読む前に、以下のことを理解しておく必要があります。

- ネットワーク接続の適切な設定方法 (22章高度なネットワーク)
- あなたのメールホストに対する DNS 情報の適切な設定方法 (22章高度なネットワーク)
- サードパーティ製ソフトウェアのインストール方法 (5章アプリケーションのインストール - packages と ports)

21.2. 電子メールを使う

email の交換には 5 つの主要な部分があります。それらは ユーザープログラム、サーバーデーモン、DNS、POP もしくは IMAP のデーモン、そしてもちろん メールホストです。

21.2.1. ユーザープログラム

いくつか名前を挙げれば、mutt, pine, elm そして mail といったコマンドラインプログラムや balsa, xfmail のような GUI プログラム、WWW ブラウザーのようにさらに「洗練された」ものまであります。これらのプログラムは、email の処理を server daemons を呼び出したり TCP 経由で渡したり、といった手段でローカルの「メールホスト」に任せるだけです。

21.2.2. メールホストサーバデーモン

通常、これは sendmail (FreeBSD のデフォルト) や qmail, postfix もしくは exim といった他のメールサーバーデーモンの一つです。 他にもあるのですが、以上のものが広く使われています。

サーバーデーモンは通常 2 つの機能 — やってくるメールを受け取るのと出ていくメールを配送する、を持っています。メールを読むために POP や IMAP で接続する、ということはできません。そのためにはもう一つデーモンが必要なのです。

いくつかの古いバージョンの sendmail には深刻なセキュリティ問題がありますが、現在のバージョンを使っていれば特に問題ないことに注意してください。例のごとく、どんなソフトウェアを利用する時にも最新の状態にしておくのが大事なのです。

21.2.3. Email と DNS

Domain Name System (DNS) とそのデーモンである `named` は email の配送において大変重要な役割を担ってます。 あなたのサイトからもう一つのサイトへメールを配送するためには、サーバーデーモンは DNS からそのサイトを探し、メールの受け取り先のホストを決定します。

メールがあなたに送られた場合にも同じような仕組みになっています。 DNS にはホスト名と IP アドレス、ホスト名とメールホストをマッピングするデータベースがあります。IP アドレスは A レコードで指定されます。MX (Mail eXchanger) レコードはあなた宛のメールを受け取るホストを指定します。あなたのホスト名に対する MX レコードがない場合には、メールは直接あなたのホストに配送されます。

21.2.4. メールの受け取り

メールはメールホストが受け取ります。 このホストは送られてきたメールを集め、 (ユーザーが) 読んだりピックアップしたりするために保存します。保存されているメールをピッ

クアップするにはメールホストに接続する必要があります。これは POP や IMAP を用いて行なわれます。メールホスト上で直接メールを読みたい時は POP や IMAP のサーバーは必要ありません。

POP や IMAP のサーバーを走らせるためには 2 つのことをやらなければいけません。

1. POP や IMAP のデーモンを ports コレクション からインストールします。
2. `/etc/inetd.conf` を修正して POP や IMAP のサーバーが起動されるように設定します。

21.2.5. メールホスト

メールホストとは責任をもってメールを配送したり、 あなたのホストや、もしかするネットワークも、に宛てたメールを受け取ったりするホストに与えられる名前です。

21.3. sendmail の設定

##: Christopher Shumway.

sendmail(8) は FreeBSD のデフォルトの メールトランスファエージェント (MTA) です。 sendmail の仕事はメールユーザエージェント (MUA) からのメールを受け取り、それを設定ファイルで定義された適当なメーラに届けることです。sendmail はネットワーク接続を受け入れて、ローカルのメールボックスにメールを届けたり 別のプログラムにメールを渡したりもできます。

sendmail は次の設定ファイルを使用します。

ファイル名	機能
`/etc/mail/access`	sendmail アクセスデータベースファイル
`/etc/mail/aliases`	メールボックスエイリアス
`/etc/mail/local-host-names`	sendmail が受け付ける配送先ホストのリスト
`/etc/mail/mailer.conf`	メーラプログラムの設定
`/etc/mail/mailertable`	メーラ配送表
`/etc/mail/sendmail.cf`	sendmail の主設定ファイル
`/etc/mail/virtusertable`	仮想ユーザおよび仮想ドメイン表

21.3.1. `/etc/mail/access`

アクセスデータベースは、どのホストまたは IP アドレスがローカルメールサーバに接続できるか、 そして接続の種類は何か、ということを定義します。 ホストは `OK`, `REJECT`,

RELAY として指定できます。または、メーラエラーを指定することで、単に sendmail のエラー処理ルーチンに渡されます。OK として指定されたホスト (これはデフォルトです) は、メールの最終宛先がローカルマシンである限り、このホストへメールを送ることを認められます。REJECT として指定されたホストは、すべてのメール接続を拒絶されます。ホスト名に対して RELAY オプションを指定されたホストは、このメールサーバを通過して任意の宛先へメールを送ることを認められます。

例21.1 sendmail アクセスデータベースの設定

```
cyberspammer.com                550 We don't accept mail from ↺
spammers
FREE.STEALTH.MAILER@            550 We don't accept mail from ↺
spammers
another.source.of.spam          REJECT
okay.cyberspammer.com           OK
128.32                          RELAY
```

この例では五つのエントリがあります。 表の左側に当てはまるメール送信者は、表の右側の動作に支配されます。 はじめの二つの例は、エラーコードを sendmail のエラー処理ルーチンに渡します。 メールが表の左側に当てはまると、リモートホストにそのメッセージが表示されます。 次のエントリは another.source.of.spam というインターネット上の特定のホストからのメールを拒絶します。 次のエントリは okay.cyberspammer.com からのメール接続を受け入れます。このエントリは上にある cyberspammer.com という行よりもさらに厳密です (厳密に一致すればするほど、そうでないものより優先されます)。最後のエントリは 128.32 から始まる IP アドレスのホストからの電子メールのリレーを認めます。 これらのホストは他のメールサーバに到達できるこのメールサーバを使ってメールを送ることができるでしょう。

このファイルを変更したら、データベースを更新するために /etc/mail/ ディレクトリで make コマンドを実行する必要があります。

21.3.2. /etc/mail/aliases

エイリアスデータベースには、 他のユーザ、ファイル、プログラムまたは他のエイリアスに展開される 仮想的なメールボックスの一覧が記載されています。 /etc/mail/aliases において使用できる例をいくつかあげます。

例21.2 メールエイリアス

```
root: localuser
```

```
ftp-bugs: joe,eric,paul
bit.bucket:  /dev/null
procmail: "|/usr/local/bin/procmail"
```

ファイル形式はシンプルです。 コロンの左側にあるメールボックス名は、右側のターゲットに展開されます。はじめの例は単純に `root` のメールボックスを `localuser` のメールボックスに展開し、それからエイリアスデータベースをもう一度調べます。 一致するエントリがなければメッセージはローカルユーザである `localuser` に配送されます。次の例はメールリストです。 `ftp-bugs` のメールボックスへのメールは `joe`, `eric` および `paul` の三つのローカルメールボックスに展開されます。リモートメールボックスは `user@example.com` のように指定できることに注意してください。次の例はメールをファイル、この場合 `/dev/null` に書き込みます。最後の例はメールをプログラムに送ります。この場合メールのメッセージは UNIX® パイプを通じて `/usr/local/bin/procmail` の標準入力に書き込まれます。

このファイルを変更したら、データベースを更新するために`/etc/mail/` ディレクトリで `make` コマンドを実行する必要があります。

21.3.3. `/etc/mail/local-host-names`

これは sendmail(8) がローカルホスト名として認めるホスト名のリストです。sendmail がメールを受け取るすべてのドメインやホストにこのファイルを置いてください。 たとえば、このメールサーバは `example.com` というドメインおよび `mail.example.com` というホストへのメールを受け取るとすると、`local-host-names` ファイルの内容は次のようになるでしょう。

```
example.com
mail.example.com
```

このファイルを更新したら、変更を読み込むために sendmail(8) を再起動する必要があります。

21.3.4. `/etc/mail/sendmail.cf`

sendmail の主設定ファイルである `sendmail.cf` は、電子メールアドレスの書き換えから、リモートメールサーバへ拒絶メッセージを送ることまで sendmail の全般的な動作をすべて制御します。 当然、そのようなさまざまな役割によりこの設定ファイルは大変複雑で、その詳細についてはこの節の少し範囲外です。好運なことに、標準的な構成のメールサーバではこのファイルをめったに変更する必要はありません。

sendmail の主設定ファイルは sendmail の機能と動作を決定する m4(1) マクロから構築できます。詳細については `/usr/src/contrib/sendmail/cf/README` を参照してください。

このファイルを更新したら、その変更を反映するために sendmail を再起動する必要があります。

21.3.5. /etc/mail/virtusertable

virtusertable は仮想ドメインおよび仮想メールボックスに対するアドレスを実際のメールボックスと対応づけます。これらのメールボックスにはローカル、リモート、/etc/mail/aliases に定義されたエイリアス、またはファイルを使用できます。

例21.3 仮想ドメインメール対応表の例

```
root@example.com                root
postmaster@example.com          postmaster@noc.example.net
@example.com                    joe
```

上の例では example.com ドメインへの対応づけをしています。このファイルはファイルの下までファーストマッチ (訳注: 一致するルールが複数ある場合、一番最初に一致したルールが適用されること) で処理されます。はじめの行では root@example.com を ローカルの root メールボックスに対応づけています。 次のエントリでは postmaster@example.com を noc.example.net ホスト上の postmaster メールボックスに対応づけています。最後に、今までのところでは example.com に関して何も一致しない場合、最後のエントリと一致するでしょう。 これは example.com の誰かに送ったすべてのメールが一致します。これは joe のローカルメールボックスに対応づけられています。

21.4. MTA の変更

##: Andrew Boothman.
Information taken from e-mails written by Gregory Neil Shapiro.

すでに述べたように、FreeBSD には MTA (Mail Transfer Agent) として、sendmail がすでにインストールされています。したがって、デフォルトではこれがメールの送受信を担当しています。

しかしながら、さまざまな理由によって、システムの MTA を変更しようと考えるシステム管理者もいるかもしれません。その理由は、単に他の MTA を試してみたいというものから 他のメーラに依存する特定の機能やパッケージが必要だといったものまで、多岐にわたることでしょう。幸い、理由がどんなものであれ、FreeBSD では簡単に変更できます。

21.4.1. 新しい MTA のインストール

さまざまな MTA が利用できます。FreeBSD Ports Collection から探しはじめるのがよいでしょう。もちろん、どんな場所からでも、あなたが利用したい MTA が FreeBSD で動作する限りすべて自由に使えます。

新しい MTA をインストールすることからはじめましょう。新しい MTA をインストールすると、あなたの要求が実際に実現したかどうか決める機会が与えられます。さらに、サービスを sendmail から引き継ぐ前に 新しいソフトウェアを設定する機会が与えられます。これを行う場合、新しいソフトウェアが `/usr/bin/sendmail` のようなシステムバイナリを上書きしようとしないことを確認してください。 そうしないとあなたが設定する前に新しいメールソフトウェアが本格的に動作しはじめてしまいます。

あなたが選択したソフトウェアを設定する方法についての情報は、その MTA の文書を参照してください。

21.4.2. sendmail を無効にする

sendmail を起動するために使用されていた手続きは、 4.5-RELEASE と 4.6-RELEASE の間で著しく変更されました。したがって、それを無効にするための手続きは微妙に違います。

21.4.2.1. 2002 年 4 月 4 日より前の FreeBSD 4.5-STABLE (4.5-RELEASE とそれ以前のバージョンが該当)

`/etc/rc.conf` に次の行を加えてください。

```
sendmail_enable="NO"
```

これは sendmail のメール受信機能を無効にします。 しかし `/etc/mail/mailer.conf` (下記参照) が変更されていなければ、sendmail はメールの送信にまだ使われるでしょう。

21.4.2.2. 2002 年 4 月 4 日以降の FreeBSD 4.5-STABLE (4.6-RELEASE とそれ以降のバージョンが該当)

sendmail を完全に無効にするためには `/etc/rc.conf` に次の行を加えなくてはいけません。

```
sendmail_enable="NONE"
```

警告

もしこの方法で sendmail のメール送信機能を無効にしたのなら、 完全に動作する代替メール配送システムと置き換えることが重要です。 さもなければ、periodic(8) などのシステム機能は、それらの結果を通常想定しているようにメールで配送すること

> ができなくなるでしょう。 システムの多くの部分が sendmail 互換のシステムがあることを想定しているかもしれません。 もしそれらを無効にした後に、アプリケーションがメールを送ろうとするために sendmail のバイナリを使用し続ければ、メールは使われていない sendmail のキューに入り、そして決して配送されないでしょう。

もし sendmail のメール受信機能だけを無効にしたいのなら `/etc/rc.conf` に以下の行を追加してください。

```
sendmail_enable="NO"
```

sendmailの起動オプションに関する詳細は rc.sendmail(8) マニュアルをご覧ください。

21.4.3. 起動時に新しい MTA を起動する

起動時に新しい MTA を起動するには二つの選択肢があります。ここでも、あなたが稼働させている FreeBSD のバージョンに依存します

21.4.3.1. 2002 年 4 月 11 日より前の FreeBSD 4.5-STABLE (4.5-RELEASE とそれ以前のバージョンが該当)

`/usr/local/etc/rc.d/` ディレクトリに、ファイル名が `.sh` でおわり、`root` によって実行可能なスクリプトを追加します。このスクリプトは `start` および `stop` パラメータを引数として受け付けるようにします。 起動時にシステムスクリプトは次のコマンドを実行するでしょう。

```
/usr/local/etc/rc.d/supermailer.sh start
```

これは手動でサーバを起動するためにも使用できます。 システム終了時にはシステムスクリプトは `stop` オプションを使用して、次のコマンドを実行するでしょう。

```
/usr/local/etc/rc.d/supermailer.sh stop
```

これはシステムが稼働している間に手動でサーバを停止するためにも使えます。

21.4.3.2. 2002 年 4 月 11 日以降の FreeBSD 4.5-STABLE (4.6-RELEASE とそれ以降のバージョンが該当)

より新しいバージョンの FreeBSD では、上記の方法または次の行を `/etc/rc.conf` に設定できます。

```
mta_start_script="filename"
```

`filename` は、あなたが MTA を立ち上げるために起動時に実行するスクリプト名です。

21.4.4. システムのデフォルトメーラとして sendmail を置き換える

sendmail プログラムは UNIX® システム上の標準ソフトウェアとして本当にどこでも利用できるので、これがすでにインストールおよび設定されているとみなしている ソフトウェアもあるかもしれません。この理由により、代替となる MTA の多くは sendmail コマンドラインインタフェースと 互換性のある実装を提供しています。これを「差し込む」ことによって、sendmail の置き換えとして代替 MTA を使用することが容易になります。

したがって、あなたが互換メーラを使用しているときには、`/usr/bin/sendmail` のような標準 sendmail バイナリを実行しようとするソフトウェアが、実際にはその代わりにあなたの選択したメーラを実行しているということを 確かめる必要があるでしょう。好運なことに、FreeBSD はこの仕事をする mailwrapper(8) と呼ばれるシステムを提供しています。

インストールされたまま sendmail が稼働しているときには `/etc/mail/mailer.conf` には以下のような記述があるでしょう。

```
sendmail    /usr/libexec/sendmail/sendmail
send-mail   /usr/libexec/sendmail/sendmail
mailq       /usr/libexec/sendmail/sendmail
newaliases  /usr/libexec/sendmail/sendmail
hoststat    /usr/libexec/sendmail/sendmail
purgestat   /usr/libexec/sendmail/sendmail
```

このことは、これらのうちどの共通コマンド (`sendmail` 自身のような) が実行されても、システムは `mailer.conf` を確認して、代わりに `/usr/libexec/sendmail/sendmail` を実行する `sendmail` という名前の mailwapper のコピーを呼び出すことを意味します。このようなシステムでは、デフォルトの `sendmail` が呼び出されたときに、どのバイナリが実際に実行されるかを変更するのが簡単になります。

したがって、sendmail の代わりに `/usr/local/supermailer/bin/sendmail-compat` を実行させたいのなら、次のように `/etc/mail/mailer.conf` を変更してください。

```
sendmail    /usr/local/supermailer/bin/sendmail-compat
send-mail   /usr/local/supermailer/bin/sendmail-compat
mailq       /usr/local/supermailer/bin/mailq-compat
newaliases  /usr/local/supermailer/bin/newaliases-compat
hoststat    /usr/local/supermailer/bin/hoststat-compat
purgestat   /usr/local/supermailer/bin/purgestat-compat
```

21.4.5. 完了

あなたのやりたいようにすべてを設定しおえたら、もはや必要のない sendmail のプロセスを終了して新しいソフトウェアに関するプロセスを起動するか、 単に再起動してください。再起動することによって、新しい MTA が起動時に正しく立ち上がるように システムが設定されているかどうか確認することもできるでしょう。

21.5. トラブルシュート

問: どうして自分のサイトのホストなのに FQDN を使わなければいけないのですか?

答: 恐らく、そのホストは実際には別のドメインにあるのでしょう。 例えば `foo.bar.edu` ドメインにいて、 `bar.edu` というドメイン内の `mumble` というホストにアクセスしたいとします。 この時は単に `mumble` ではなく `mumble.bar.edu` と FQDN で参照しなければなりません。

そもそも、BSD BIND のリゾルバー (resolver) ではこのようなことが可能でしたが、 FreeBSD に入っている最新版の BIND では自分のドメイン以外に対する FQDN でない省略形は許されません。従ってホストを `mumble` と曖昧に指定した場合は `mumble.foo.bar.edu` という名前があればそれになり、そうでなければ root ドメインから検索されます。

これは、`mumble.bar.edu` と `mumble.edu` ということなったドメイン名に対してホスト名のサーチがおこなわれていた以前の振る舞いとは異なったものです。このような事が悪い例もしくはセキュリティホールとみなされる理由については RFC 1535 を見てください。

`/etc/resolv.conf` で

```
domain foo.bar.edu
```

と書いてある行を

```
search foo.bar.edu bar.edu
```

と書き換えることで上のようなことができます。しかし、RFC 1535 にあるように検索順序が「内部 (local) と外部 (public) の管理の境界」をまたがないようにしてください。

問: sendmail が mail loops back to myself というメッセージを出すのですが。

答: sendmail FAQ に次のように書いてあります。

```
#Local configuration error# #################

553 relay.domain.net config error: mail loops back to myself
554 <user@domain.net>... Local configuration error

######################?

####### domain.net ###############
MX ###########(####
relay.domain.net) ##########
####### domain.net
```

```
############################
#####
FEATURE(use_cw_file) ###########
/etc/mail/local-host-names ### domain.net ##########
#####/etc/mail/sendmail.cf ### #Cw domain.net#
##########
```

sendmail FAQ は http://www.sendmail.org/faq にありますので、メールの設定に「おかしなこと」があれば常に読んでください。

問: ダイアルアップ PPP ホストでメールサーバを実行するにはどうしたらいいの?

答: LAN 上にある FreeBSD マシンを、インターネットに接続したいとします。FreeBSD マシンは、その LAN でのメールゲートウェイになります。FreeBSD マシンは専用線接続ではありません (訳注: ダイアルアップ接続など)。

これには、少なくとも二つの方法があります。一つは UUCP を使うことです。

もう一つの方法は、あなたのドメインに対するセカンダリ MX サービスを提供する常時稼働のインターネットサーバを用意することです。たとえば、あなたの会社のドメインが example.com で、ISP があなたのドメインに セカンダリ MX サービスを提供するために example.net ドメインを 用意するとしたら次のようにします。

```
example.com. MX 10 example.com.
MX 20 example.net.
```

最終的なメール受信先としては、一つのホストだけが定義されるべきです (example.com 上の /etc/mail/sendmail.cf ファイルに、Cw example.com を追加します)。

送信側の sendmail が、メールを配送しようとしている時、モデムの接続を介してあなたのところ (example.com) に接続しようとします。大抵の場合、あなたのマシンがオンラインでないために、接続はタイムアウトしてしまうでしょう。sendmail プログラムは自動的に、たとえばあなたのインターネットプロバイダなどのセカンダリの MX サイト (example.net) にメールを配送するでしょう。セカンダリ MX サイトは定期的にあなたのホストに接続し、プライマリ MX ホスト (example.com) にメールを配送しようとするでしょう。

ログインスクリプトとして、このようなものを使うとよいでしょう。

```
#!/bin/sh
# Put me in /usr/local/bin/pppmyisp
( sleep 60 -; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

ユーザごとにログインスクリプトを作りたい場合には、上記のスクリプトの代わりに、sendmail -qRexample.com を使用することもできます。このようにする

と、キューの中の example.com に対するすべてのメールは、すぐに強制的に処理されます。

さらに、次のような改良もできます。

以下は、FreeBSD Internet service provider's メーリングリスト から抜粋してきたメッセージです。

```
> ################ MX ########
> ################################## MX
> ####### (######################
> ##################)#
> #### sendmail ##30 ########################
> ###################### MX ##################
> ### 30 ## ################
>
> ############### sendmail #################?
> ###################### (root) ###########

sendmail.cf # #privacy flags# #######
Opgoaway,restrictqrun #########

root ########################
restrictqrun #############MX ###############
######################### MX ####
###########

# If we are the best MX for a host, try directly instead of ↺
generating
# local config error.
OwTrue

########################
##########################################
################################
############################
#host.customer.com# ######customer.com# #############
DNS ####customer.com# #### A ##############
```

問: なぜ他のホストにメールを送ろうとすると、いつも Relaying Denied と怒られてしまうの？

答: FreeBSD がインストールされたデフォルトの状態では、sendmail は動作しているホストからのメールだけを送るように設定されています。たとえば POP3 サーバがインストールされているとすると、ユーザは学校や職場など他のリモートの場所からメールを確認することが できます。しかし、彼らは外部からそのホスト以外へのメールを 送ることはやはりできません。通常、メールを送ろうとしてから少しすると、5.7 Relaying Denied というエラーメッセージの書かれたメールが MAILER-DAEMON から送られてくるでしょう。

これを解決する方法はいくつかあります。 一番の正攻法は /etc/mail/relay-domains リレードメインファイルにあなたの ISP のアドレスを書くことです。これをするのに簡単な方法は次のとおりです。

```
# echo "your.isp.example.com" > /etc/mail/relay-domains
```

このファイルを作成または編集したら、sendmail を再起動してください。もしあなたがサーバ管理者でメールをローカルに送りたくないか、 ポイントを使用して他のマシン (や、さらに他の ISP) の クライアントまたはシステムへ送りたい時は、とても効果があります。さらに、あなたが一つあるいは二つだけのメールアカウントを 設定している場合でもこれは非常に有用です。追加すべきアドレスがたくさんある場合には、単にこのファイルをあなたの好きなテキストエディタで開いて、そして一行に一つずつドメインを追加してください。

```
your.isp.example.com
other.isp.example.net
users-isp.example.org
www.example.org
```

これで、リストに掲載されているすべてのホスト (ユーザがあなたのシステムにアカウントを持っていると規定する) からあなたのシステムを通るすべてのメールは送信に成功するでしょう。これはあなたのシステムから SPAM を送ることを認めることなく、リモートであなたのシステムからメールを送ることをユーザに 認めるためのとてもよい方法です。

21.6. 先進的なトピックス

これからのセクションでは、メールの設定やドメイン全体のためのメールの設定といったさらに突込んだ話題について触れます。

21.6.1. 基本事項

あなたのマシンに FreeBSD を普通にインストールして、/etc/resolv.conf ファイルを設定するか、またはネームサーバを走らせれば、他のホストへ電子メールを送ることができるようになります。あなたのホスト宛のメールをあなた自身の FreeBSD ホスト上の MTA (たとえば sendmail) に配送するようにしたい場合には、次の二つの方法があります。

- 自身でネームサーバーを実行し、自分のドメインを持つ。例えば FreeBSD.org 。
- あなたのホストへ直接メールが配送されるようにする。これはメールがあなたのマシンの現在の DNS 名に直接配送されるようにすることにより実現できます。 たとえば example.FreeBSD.org 。

上のどちらを選ぶ場合でも、自分のホストに直接メールが配送されるようにするには恒久的で 静的 な IP アドレス (ほとんどの PPP ダイアルアップ設定で用いられる動的なアドレスではなく) を持っていなければなりません。もしファイアウォールの中にいるならば、SMTP トラフィックが通過してくれないといけません。もし自分のホストでメールを直接受け取りたいならば、次の二つのうちのどちらかができていることを確認してください。

- 自分のドメインでの (一番値の小さい) MX レコードが自分のホストの IP アドレスを差していることを確認する。
- 自分のドメインの中に自分のホスト用の MX エントリがないことを確認する。

上のどちらかが設定されていれば、 自分のホストでメールを受け取ることができるでしょう。

次のコマンドを実行してみてください。

```
# hostname
example.FreeBSD.org
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
```

もしあなたのマシンが上記のメッセージだけを出力したならば、
<yourlogin@example.FreeBSD.org > へのメールは問題なく配送されるでしょう (sendmail が example.FreeBSD.org 上で正しく動作していると仮定します)。

上記のメッセージの代わりに、

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by hub.FreeBSD.org
```

というメッセージが出力された場合は、 あなたのホスト (example.FreeBSD.org) に宛てたメールは全て直接配送されずに hub 上の同じユーザー名に配送されます。

上の情報は DNS サーバーが扱います。メールルーティング情報をもつ DNS レコードは、Mail eXchange エントリーです。MX エントリが存在しない場合には、IP アドレスにしたがって、直接宛先ホストに配送されます。

freefall.FreeBSD.org の現時点での MX エントリは、次のようになっています。

```
freefall  MX 30 mail.crl.net
freefall  MX 40 agora.rdrop.com
freefall  MX 10 freefall.FreeBSD.org
freefall  MX 20 who.cdrom.com
```

freefall は多くの MX エントリを持っています。一番 MX の値の小さいホストが利用可能な場合は直接メールを受け取ります。 もしなにかの理由でアクセスができない時

には、他のホスト (ときどき「バックアップ MX」と呼ばれます) が一時的にメールを受け取ります。そして、より値の小さいホストが利用可能になったときにメールを渡し、最終的に一番値の小さいホストに渡ります。

使い勝手をよくするためには、代替の MX サイトは、それぞれ 別の経路でインターネットへ接続しているとよいでしょう。インターネットプロバイダまたは他の関連サイトが、このサービスを 提供することができます。

21.6.2. あなたのドメインに対するメール設定

「メールホスト」 (メールサーバーとしても知られています) をセットアップするためには、いろいろなワークステーションに宛てた全てのメールを受ける必要があります。 基本的には、あなたのドメイン内 (この場合だと `*.FreeBSD.org`) のすべてのホスト名宛てのすべてのメールを「受け取って」、そのメールをあなたのメールサーバーに配送し、ユーザーがマスタメールサーバ上でメールをチェックできるようにします。

話を簡単にするために、あるユーザーのアカウントはどのマシンでも同じユーザー名にすべきです。そのためには adduser(8) を使ってください。

使用する予定のメールホストは、各ワークステーションごとにメール交換が できるように設定されていなければなりません。これは DNS の設定で次のように行なうことができます。

```
example.FreeBSD.org A 204.216.27.XX  ; #########
MX 10 hub.FreeBSD.org ; ######
```

これは、ワークステーションの A レコードがどこを指していようとも そのワークステーション宛てのメールをメールホストに転送する、というものです。

自前で DNS サーバを運用しているのでなければ、この作業は自分では行えません。自分で DNS サーバを運用しないとかできないという場合は、あなたの DNS を提供しているインターネットプロバイダなどに依頼して 作業を行ってもらってください。

もしバーチャル電子メールホストを運用するなら次の情報が役に立つでしょう。例として、あなたには自分のドメイン、ここでは `customer1.org` 、を持っている顧客がいるとしましょう。あなたは `customer1.org` 宛ての全てのメールを `mail.myhost.com` というメールホストに集めたいとします。DNS エントリーは次のようになるでしょう。

```
customer1.org  MX 10 mail.myhost.com
```

`customer1.org` に対して電子メールを送りたいだけなら、A レコードは必要ありません。

注記

customer1.org に対して ping を実行しても、A レコードが存在しない限りうまくいかないことに留意しておいてください。

やらなければいけない最後のことは、メールホスト上の sendmail に対してどんなドメインやホスト宛のメールを受け取るのか、を教えることです。いくつかの方法がありますが次のどちらかでいいでしょう。

- FEATURE(use_cw_file) を使っているなら、/etc/mail/local-host-names ファイルにホストを加えます。もし sendmail のバージョンが 8.10 より前であれば該当ファイルは /etc/sendmail.cw です。
- /etc/sendmail.cf もしくは sendmail 8.10 以降なら /etc/mail/sendmail.cf といったファイルに Cwyour.host.com という行を加えます。

21.7. UUCP とともに SMTP を使う

FreeBSD とともに出荷されている sendmail の設定は、サイトがインターネットに直接接続しているものとして設計されています。UUCP 経由でメールを交換したいサイトは、他にも 設定ファイルをインストールしなければいけません。

/etc/mail/sendmail.cf を手動で調整することは先進的なトピックです。sendmail のバージョン 8 は設定ファイルを m4(1) プリプロセッサから生成します。これにより、高度に抽象化された設定を行うことができます。m4(1) による設定ファイルは /usr/src/usr.sbin/sendmail/cf 以下にあります。

もしシステムをすべてのソースとともにインストールしていなければ、sendmail の設定材料は分割された個別のソース tarball を取得してください。FreeBSD のソースコードが入った CDROM をマウントしているのなら、

```
# cd /cdrom/src
# cat scontrib.?? | tar xzf - -C /usr/src/contrib/
sendmail
```

と展開してください (展開してもたった数百 KB 程度です)。cf ディレクトリの README ファイルは m4 による設定の基本的な手引として役に立つでしょう。

UUCP 配送に対応するための一番よい方法は mailertable 機能を使用することです。これは経路を決定するために sendmail が使用できるデータベースを作成します。

まずはじめに .mc ファイルを作成しなければいけません。/usr/src/usr.sbin/sendmail/cf/cf にいくつか例があります。foo.mc という名前のファイルをあなたが作成したとすると、有効な sendmail.cf ファイルへ変換するには次のようにするだけです。

```
# cd /usr/src/usr.sbin/sendmail/cf/cf
# make foo.cf
# cp foo.cf /etc/mail/sendmail.cf
```

典型的な .mc ファイルは次のようになるでしょう。

```
VERSIONID(`Your version number ') OSTYPE(bsd4.4)

FEATURE(accept_unresolvable_domains)
FEATURE(nocanonify)
FEATURE(mailertable, `hash -o /etc/mail/mailertable')

define(`UUCP_RELAY', your.uucp.relay )
define(`UUCP_MAX_SIZE', 200000)
define(`confDONT_PROBE_INTERFACES')

MAILER(local)
MAILER(smtp)
MAILER(uucp)

Cw    your.alias.host.name
Cw    youruucpnodename.UUCP
```

accept_unresolvable_domains , nocanonify および confDONT_PROBE_INTERFACES 機能を含んでいる行は、メール配送時にまったくDNSを使用しません。UUCP_RELAY の記述はUUCP配送に対応するのに必要です。そこにインターネットホスト名を単に書くだけで .UUCP pseudo ドメインアドレスを扱うことができるようになります。大抵の場合、あなたのISPのメールリレーをそこに入力するでしょう。

次に、/etc/mail/mailertable が必要になります。メールを配送するリンクが外界との間に一つだけの場合は、次のようにファイルを記述するだけで十分でしょう。

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
.                              uucp-dom:your.uucp.relay
```

次はさらに複雑な例です。

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#
```

```
horus.interface-business.de   uucp-dom:horus
.interface-business.de        uucp-dom:if-bus
interface-business.de         uucp-dom:if-bus
.heep.sax.de                  smtp8:%1
horus.UUCP                    uucp-dom:horus
if-bus.UUCP                   uucp-dom:if-bus
.                             uucp-dom:
```

はじめの三行はドメインで宛先を指定されたメールが、配送路を「近道」するために、デフォルトルートではなく代わりにいくつかの UUCP 隣接ホストへ送られる特別な場合を扱います。次の行はメールを SMTP で配送可能なローカルイーサネットドメインへ送ります。最後に `uucp-neighbor !recipient` がデフォルトルートを上書きすることを許可するための UUCP 隣接ホストは .UUCP 仮想ドメイン記法で言及されます。最後の行は常に他のすべてが当てはまるシングルドットです。これは UUCP 隣接ホストへの UUCP 配送をすることで、世界に向けたあなたの普遍的メールゲートウェイとして役に立ちます。`uucp-dom:` キーワードの後ろにあるノード名はすべて、`uuname` コマンドを使用することで確かめられる正しい UUCP 隣接ホストである必要があります。

このファイルは、実際に使用する前に DBM データベース形式に変換する必要があることに注意してください。これを実行するコマンドラインは mailertable ファイルの先頭にコメントとして書かれています。mailertable を変更するたびにいつもこのコマンドを実行する必要があります。

最後のアドバイス: もし、いくつかのメールルーティングがうまく動いているかどうか分からないときは sendmail に `-bt` オプションをつけることを覚えておいてください。これは sendmail を アドレステストモード で起動します。あなたがテストしたいメールルーティングのアドレスを後につけて、単純に `3,0` と入力してください。最後の行は、内部で使われたメールエージェント、このエージェントが呼び出された目的地ホスト、および (もしかしたら変換された) アドレスを表示します。このモードを終了するには Ctrl+D を入力します。

```
% sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 foo@example.com
canonify           input: foo @ example . com
...
parse            returns: $# uucp-dom $@ your.uucp.relay  $: foo ↺
< @ example . com . >
> ^D
```

21.8. ダイアルアップ接続でメールを使う

あなたが固定 IP アドレスを持っているのなら、デフォルトから何も変更する必要はありません。割りあてられたインターネット名をホスト名に設定すれば、sendmail が残りをやってくれます。

あなたが動的に割り当てられた IP アドレスを持っていて、インターネットに接続するのにダイアルアップ PPP を使用しているのなら、おそらく ISP のメールサーバにメールボックスがあるでしょう。ここでは、あなたの ISP のドメインが example.net , あなたのユーザ名が user, あなたのマシンは bsd.home と呼ばれているものとします。また、ISP から、メールリレーとして relay.example.net を使用してよいと通知されているとします。

(訳注: ISP 上の) メールボックスからメールを取得するためには、取得アプリケーションをインストールしないといけません。fetchmail ユーティリティは、さまざまなプロトコルの多くに対応しているのでよい選択肢です。通常、あなたの ISP は POP3 を提供しています。あなたが user-PPP を使用しているなら、次のエントリを /etc/ppp/ppp.linkup に追加することで、インターネット接続が確立したときに自動的にメールを取得することができます。

```
MYADDR:
 !bg su user -c fetchmail
```

あなたがローカルではないアカウントへのメールを配送するために (下記のような) sendmail を使用しているなら、インターネット接続が確立するとすぐに、sendmail があなたのメールキューを処理して欲しいとおそらく考えるでしょう。これを行うには、/etc/ppp/ppp.linkup ファイルの fetchmail コマンドの後に次のコマンドを追加してください。

```
 !bg su user -c "sendmail -q"
```

bsd.home 上に user というアカウントを所有しているとします。bsd.home 上の user のホームディレクトリに .fetchmailrc ファイルを作成します。

```
poll example.net protocol pop3 fetchall pass MySecret
```

このファイルはパスワード MySecret を含んでいるので、user を除く他の誰にも読めるようになっていてはいけません。

正しい from: ヘッダでメールを送るためには、sendmail が user@bsd.home ではなく user@example.net を使用するようにしなくてはいけません。また、素早くメール送信をするために sendmail にすべてのメールを relay.example.net 経由で送るようにもしたいかもしれません。

次の .mc ファイルで十分でしょう。

```
VERSIONID(`bsd.home.mc version 1.0')
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
```

```
MASQUERADE_AS(`example.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define(`SMART_HOST', `relay.example.net')
Dmbsd.home
define(`confDOMAIN_NAME',`bsd.home')dnl
define(`confDELIVERY_MODE',`deferred')dnl
```

`.mc` ファイルを `sendmail.cf` ファイルに変換する方法の詳細については前の節を参照してください。また、`sendmail.cf` ファイルを変更した後は、sendmail を再起動し忘れないでください。

21.9. SMTP 認証

メールサーバ上で SMTP 認証を行うと、多くの利益があります。SMTP 認証は sendmail にもう一つのセキュリティ層を追加することができます。さらに、ホストを切りかえるモバイルユーザにとっては、その都度メールクライアントの設定を変更せずとも 同じメールサーバを利用できるようになります。

1. ports から security/cyrus-sasl をインストールします。この port は security/cyrus-sasl にあります。security/cyrus-sasl にはここで使用する方法に対する多くのコンパイルオプションがあり、確実に `pwcheck` オプションを選択してください。

2. security/cyrus-sasl をインストールした後に `/usr/local/lib/sasl/Sendmail.conf` を編集して (もし無ければ作成して) 次の行を追加してください。

   ```
   pwcheck_method: passwd
   ```

   この方法は sendmail があなたの FreeBSD の `passwd` データベースに対して認証することを可能にします。この方法は SMTP 認証に必要となる、それぞれのユーザに対する一組の新しいユーザ名とパスワードを 作成する際のトラブルを減らし、ログインパスワードとメールパスワードを同じままにします。

3. ここで `/etc/make.conf` 編集し、次の行を加えます。

   ```
   SENDMAIL_CFLAGS=-I/usr/local/include/sasl1 -DSASL
   SENDMAIL_LDFLAGS=-L/usr/local/lib
   SENDMAIL_LDADD=-lsasl
   ```

   これらの行は sendmail に対して、コンパイルするときに cyrus-sasl とリンクするための適切な設定オプションを与えるものです。sendmail を再コンパイルする前に cyrus-sasl がインストールされていることを確かめてください。

4. 次のコマンドを入力して sendmail を再コンパイルしてください。

```
# cd /usr/src/usr.sbin/sendmail
# make cleandir
# make obj
# make
# make install
```

sendmail のコンパイルは `/usr/src` が大幅に変更されていなくて、必要な共有ライブラリが利用可能であれば何の問題も起こらないでしょう。

5. sendmail をコンパイルして再インストールした後は、`/etc/mail/freebsd.mc` ファイル (またはあなたが `.mc` ファイルとして使用しているファイル。多くの管理者は唯一の名前を用いるために hostname(1) の出力を `.mc` として使用することを選んでいます) を編集してください。次の行を加えてください。

```
dnl set SASL options
TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
define(`confAUTH_MECHANISMS', `GSSAPI DIGEST-MD5 CRAM-MD5 ↺
LOGIN')dnl
define(`confDEF_AUTH_INFO', `/etc/mail/auth-info')dnl
```

これらのオプションは、ユーザを認証するために sendmail が利用可能な異なる方法を設定します。もし pwcheck 以外の方法を使用したいのならドキュメントを参照してください。

6. 最後に `/etc/mail` で make(1) を実行してください。これにより、新しい `.mc` ファイルから `freebsd.cf` という名前 (またはあなたの `.mc` に使用している名前) の `.cf` ファイルが作成されます。それから `make install restart` コマンドを実行してください。新しい `.cf` ファイルが `sendmail.cf` にコピーされ、sendmail が適切に再起動されるでしょう。この手続きについての詳細は `/etc/mail/Makefile` を参照してください。

すべてがうまくいけば、ログイン情報をメールクライアントに入力し、テストメッセージを送ることができるでしょう。より詳細に調べるには sendmail の `LogLevel` を 13 に設定し、すべてのエラーについて `/var/log/maillog` を見てください。

このサービスがシステムを起動した後にいつでも利用可能となるように、`/etc/rc.conf` に次の行を追加しておくとよいでしょう。

```
sasl_pwcheck_enable="YES"
sasl_pwcheck_program="/usr/local/sbin/pwcheck"
```

これにより、システムの起動時に SMTP_AUTH が確実に初期化されるでしょう。

詳細については SMTP 認証 に関する sendmail の文書を参照してください。

第22章 高度なネットワーク

22.1. この章では

この章では UNIX® システム上で良く利用されるネットワークサービスについて説明します。FreeBSD が利用するすべてのネットワークサービスをどのように定義し、設定し、テストし、そして保守するのかを扱います。さらに、本章を通してあなたの役に立つ設定例が載っています。

この章を読めば以下のことが分かります。

- ゲートウェイと経路の基本
- FreeBSD をブリッジとして動作させる方法
- ネットワークファイルシステム (NFS) の設定方法
- ディスクレスマシンのネットワークブートの設定方法
- ユーザアカウントを共有するためのネットワークインフォメーションサーバ (NIS) の設定方法
- DHCP を用いて自動的にネットワーク設定を行う方法
- ドメインネームサーバ (DNS) の設定方法
- NTP プロトコルを用いて日時を同期してタイムサーバを設定する方法
- ネットワークアドレス変換 (NAT) の設定方法
- `inetd` デーモンの管理方法
- PLIP 経由で二台のコンピュータを接続する方法
- FreeBSD で IPv6 を設定する方法

この章を読む前に、以下のことを行っておくべきです。

- `/etc/rc` スクリプトの基本を理解していること
- 基礎的なネットワーク用語に精通していること

22.2. ゲートウェイと経路

##: Coranth Gryphon.

あるマシンがネットワーク上で他のマシンをみつけることができるようにするには、 あるマシンから他のマシンへどのようにたどり着くかを記述する適切な仕組みが必要です。 この仕組みをルーティングと呼びます。「経路」(route) は「送信先」(destination) と「ゲートウェイ」の 2 つのアドレスの組で定義します。この組合せは、この 送信先 へたどり着こうとする場合は、その ゲートウェイ を通じて通信することを示しています。 送信先には個々のホスト、サブネット、「デフォルト」の 3 つの型があります。「デフォルトルート」は他のどの経路も適用できない場合に使われます。デフォルトルートについてはのちほどもう少し詳しく述べます。また、ゲートウェイには、個々のホスト、インタフェース (「リンク」とも呼ばれます)、イーサネットハードウェアアドレス (MAC アドレス) の 3 つの型があります。

22.2.1. 例

以下に示す `netstat` の例を使って、ルーティングのさまざまな状態を説明します。

```
% netstat -r
Routing tables

Destination      Gateway            Flags     Refs     Use     ↺
Netif Expire

default          outside-gw         UGSc       37      418      ppp0
localhost        localhost          UH          0      181       lo0
test0            0:e0:b5:36:cf:4f   UHLW        5    63288       ed0
    77
10.20.30.255     link#1             UHLW        1     2421
example.com      link#1             UC          0        0
host1            0:e0:a8:37:8:1e    UHLW        3     4601       lo0
host2            0:e0:a8:37:8:1e    UHLW        0        5       ↺
lo0 =>
host2.example.com link#1             UC          0        0
224              link#1             UC          0        0
```

最初の 2 行はデフォルトルート (次節で扱います) と、`localhost` への経路を示しています。

`localhost` に割り当てるインタフェース (`Netif` 欄) としてこのルーティングテーブルが指定しているのは `lo0` で、これはループバックデバイスともいいます。これは結局のところ出たところに戻るだけなので、 この送信先あてのトラフィックは、LAN に送られずに、すべて内部的に処理されます。

次の行では `0:e0:` から始まるアドレスに注目しましょう。これはイーサネットハードウェアアドレスで、MAC アドレスともいいます。 FreeBSD はローカルなイーサネット上の任意のホスト (この例では `test0`) を自動的に認識し、 イーサネットインタフェース `ed0` にそのホストへの直接の経路をつけ加えます。この種の経路には、タイムアウト時間 (`Expire` 欄) も結びつけられており、指定された時間内にホストからの応答がないこと

を判断するのに用いられます。その場合、そのホストへの経路情報は自動的に削除されます。これらのホストは RIP (Routing Information Protocol) という、最短パス判定に基づいてローカルなホストへの経路を決定する仕組みを利用して認識されます。

さらに FreeBSD ではローカルサブネットへの経路情報も加えることができます (`10.20.30.255` は `10.20.30` というサブネットに対するブロードキャストアドレスで、`example.com` はこのサブネットに結びつけられているドメイン名)。`link#1` という名称は、このマシンの一つ目のイーサネットカードのことをさします。これらについては、何も追加インタフェースが指定されていないことがわかります。

これら 2 つのグループ (ローカルネットワークホストとローカルサブネット) は、両方とも routed というデーモンによって自動的に経路が設定されます。routed を動かさなければ、静的に定義した (つまり明示的に設定した) 経路のみが存在することになります。

`host1` の行は私たちのホストのことで、イーサネットアドレスで示されています。送信側のホストの場合、FreeBSDはイーサネットインタフェースへ送るのではなく、ループバックインタフェース (`lo0`) を使います。

2 つある `host2` の行は、ifconfig(8) のエイリアスを使ったときにどのようになるかを示す例です (このようなことをする理由については Ethernet の節を参照してください)。`lo0` の後にある `=>` は、インタフェースが (このアドレスがローカルなホストを参照しているので) ループバックを使っているというだけでなく、エイリアスになっていることも示しています。このような経路はエイリアスに対応しているホストにのみ現れます。ローカルネットワーク上の他のすべてのホストでは、それぞれの経路に対して単に`link#1` となります。

最後の行 (送信先サブネット `224`) はマルチキャストで扱うものですが、これは他の節で説明します。

最後に `Flags` (フラグ) 欄にそれぞれの経路のさまざまな属性が表示されます。以下にフラグの一部と、それが何を意味しているかを示します。

U	Up: この経路はアクティブです。
H	Host: 経路の送信先が単一のホストです。
G	Gateway: この送信先へ送られると、どこへ送ればよいかを明らかにして、そのリモートシステムへ送られます。
S	Static: この経路はシステムによって自動的に生成されたのではなく、手動で作成されました。
C	Clone: マシンに接続したときにこの経路に基づく新しい経路が作られます。この型

	の経路は通常はローカルネットワークで使われます。
W	WasCloned: ローカルエリアネットワーク (LAN) の (Clone) 経路に基づいて自動的に生成された経路であることを示します。
L	Link: イーサネットハードウェアへの参照を含む経路です。

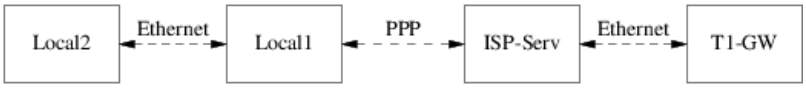
22.2.2. デフォルトルート

ローカルシステムからリモートホストにコネクションを張る必要がある場合、 既知の経路が存在するかどうかを確認するためにルーティングテーブルをチェックします。 到達するための経路を知っているサブネットの内部にリモートホストがある場合 (Cloned routes)、システムはそのインタフェースから接続できるかどうか確認します。

知っているパスがすべて駄目だった場合でも、 システムには最後の手段として 「デフォルト」 ルートがあります。このルートはゲートウェイルート (普通はシステムに 1 つしかありません) の特別なものです。そして、フラグ欄には必ず c が表示されています。このゲートウェイは、LAN 内のホストにとって、どのマシンでも外部へ (PPP リンク、DSL、ケーブルモデム、T1、 またはその他のネットワークインタフェースのいずれかを経由して) 直接接続するために設定されるものです。

外部に対するゲートウェイとして機能するマシンでデフォルトルートを設定する場合、 デフォルトルートはインターネットサービスプロバイダ (ISP) のサイトのゲートウェイマシンになるでしょう。

それではデフォルトルートの一例を見てみましょう。一般的な構成を示します。



ホスト `Local1` とホスト `Local2` はあなたのサイト内にあります。`Local1` はダイアルアップ PPP 接続経由で ISP に接続されています。この PPP サーバコンピュータは、その ISP のインターネットへの接続点に向けた外部インタフェースを備えた他のゲートウェイコンピュータへ LAN を通じて接続しています。

あなたのマシンのデフォルトルートはそれぞれ次のようになります。

ホスト	デフォルトゲートウェイ	インタフェース
Local2	Local1	Ethernet
Local1	T1-GW	PPP

「なぜ (あるいは、どうやって) デフォルトゲートウェイを、Local1 が接続されている ISP のサーバではなく、T1-GW に設定するのか」という質問がよくあります。

PPP 接続で、あなたのサイト側の PPP インタフェースは、ISP のローカルネットワーク上のアドレスを用いているため、ISP のローカルネットワーク上のすべてのマシンへの経路は 自動的に生成されています。つまりあなたのマシンは、どのようにして T1-GW に到達するかという経路を既に知っていることになりますから、ISP サーバにトラフィックを送るのに、中間的な段階を踏む必要はありません。

一般的にローカルネットワークでは X.X.X.1 というアドレスをゲートウェイアドレスとして使います。ですから (同じ例を用います)、あなたの class-C のアドレス空間が 10.20.30 で ISP が 10.9.9 を用いている場合、デフォルトルートは次のようになります。

ホスト	デフォルトルート
Local2 (10.20.30.2)	Local1 (10.20.30.1)
Local1 (10.20.30.1, 10.9.9.30)	T1-GW (10.9.9.1)

デフォルトルートは /etc/rc.conf ファイルで簡単に定義できます。この例では、Local2 マシンで /etc/rc.conf に次の行を追加しています。

```
defaultrouter="10.20.30.1"
```

route(8) コマンドを使ってコマンドラインから直接実行することもできます。

```
# route add default 10.20.30.1
```

経路情報を手動で操作する方法について詳しいことは route(8) のマニュアルページをご覧ください。

22.2.3. デュアルホームホスト

ここで扱うべき種類の設定がもう一つあります。それは 2 つの異なるネットワークにまたがるホストです。技術的にはゲートウェイとして機能するマシン (上の例では PPP コネクションを用いています) はすべてデュアルホームホストです。しかし実際にはこの言葉は、2 つの LAN 上のサイトであるマシンを指す言葉としてのみ使われます。

2 枚のイーサネットカードを持つマシンが、 別のサブネット上にそれぞれアドレスを持っている場合があります。 あるいは、イーサネットカードが 1 枚しかないマシンで、ifconfig(8) のエイリアスを使っているかもしれません。物理的に分かれている 2 つのイーサネットのネットワークが使われているならば前者が用いられます。 後者は、物理的には 1 つのネットワークセグメントで、論理的には 2 つのサブネットに分かれている場合に用いられます。

どちらにしても、このマシンがお互いのサブネットへのゲートウェイ (inbound route) として定義されていることが分かるように、 おのおののサブネットでルーティングテーブルを設定します。このマシンが 2 つのサブネットの間のルータとして動作するという構成は、パケットのフィルタリングを実装する必要がある場合や、一方向または双方向のファイアウォールを利用したセキュリティを構築する場合によく用いられます。

このマシンが二つのインタフェース間で実際にパケットを受け渡すようにしたい場合は、FreeBSD でこの機能を有効にしないといけません。くわしい手順については次の節をご覧ください。

22.2.4. ルータの構築

ネットワークルータは単にあるインタフェースから別のインタフェースへパケットを転送するシステムです。 インターネット標準およびすぐれた技術的な慣習から、 FreeBSD プロジェクトは FreeBSD においてこの機能をデフォルトでは有効にしていません。rc.conf(5) 内で次の変数を YES に変更することでこの機能を有効にできます。

```
gateway_enable=YES          # Set to YES if this host will be a ↺
gateway
```

このオプションは sysctl(8) 変数の net.inet.ip.forwarding を 1 に設定します。一時的にルーティングを停止する必要があるときには、この変数を一時的に 0 に設定しなおせます。

次に、トラフィックの宛先を決めるために、 そのルータには経路情報が必要になります。ネットワークが十分簡素なら、静的経路が利用できます。また、FreeBSD は BSD の標準ルーティングデーモンである routed(8) を備えています。これは RIP (バージョン 1 および 2) および IRDP を扱えます。BGP バージョン 4、OSPF バージョン2、その他洗練されたルーティングプロトコルは net/zebra package を用いれば対応できます。また、より複雑なネットワークルーティングソリューションには、GateD® のような商用製品も利用可能です。

このように FreeBSD を設定したとしても、ルータに対するインターネット標準要求を完全に満たすわけではありません。しかし、通常利用に関しては十分といえます。

22.2.5. 静的な経路の設定

##: Al Hoang.

22.2.5.1. 手動による経路の設定

以下のようなネットワークが存在すると仮定します。

```
    INTERNET
      | (10.0.0.1/24) Default Router to Internet
```

```
     |
     |Interface xl0
     |10.0.0.10/24
   +------+
   |      | RouterA
   |      | (FreeBSD gateway)
   +------+
     | Interface xl1
     | 192.168.1.1/24
     |
  +--------------------------------+
   Internal Net 1      | 192.168.1.2/24
                       |
                  +------+
                  |      | RouterB
                  |      |
                  +------+
                      | 192.168.2.1/24
                      |
                    Internal Net 2
```

このシナリオでは、FreeBSD マシンの RouterA がインターネットに向けられたルータとして動作します。ルータは外側のネットワークへ接続できるように 10.0.0.1 へ向けたデフォルトルートを保持しています。RouterB はすでに適切に設定されており、どこへ向かう必要があるか、 行き着く方法を知っていると仮定します (この例では、図のように簡単です。192.168.1.1 をゲートウェイとして RouterB にデフォルトルートを追加するだけです)。

RouterA のルーティングテーブルを確認すると、以下のような出力を得ます。

```
% netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif
  Expire
default            10.0.0.1           UGS         0    49378    xl0
127.0.0.1          127.0.0.1          UH          0        6    lo0
10.0.0/24          link#1             UC          0        0    xl0
192.168.1/24       link#2             UC          0        0    xl1
```

現在のルーティングテーブルでは、RouterA はまだ Internal Net 2 には到達できないでしょう。192.168.2.0/24 の経路を保持していないからです。解決するための一つの方法は、経路を手動で追加することです。以下のコマンドで RouterA のルーティングテーブルに 192.168.1.2 を送り先として、Internal Net 2 ネットワークを追加します。

```
# route add -net 192.168.2.0/24 192.168.1.2
```

これにより、RouterA は、192.168.2.0/24 ネットワーク上のホストに到達出来ます。

22.2.5.2. 永続的な設定

上記の例は、起動しているシステム上に静的な経路を設定する方法としては完全です。しかしながら、FreeBSD マシンを再起動した際にルーティング情報が残らないという問題が一つあります。静的な経路を追加するには、/etc/rc.conf ファイルにルートを追加してください。

```
# Add Internal Net 2 as a static route
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

static_routes の設定変数は、スペースによって分離される文字列のリストです。それぞれの文字列は経路名として参照されます。上記の例では static_routes は一つの文字列のみを持ちます。その文字列は *internalnet2* です。その後、route_internalnet2 という設定変数を追加し、route(8) コマンドに与えるすべての設定パラメータを指定しています。前節の例では、以下のコマンド

```
# route add -net 192.168.2.0/24 192.168.1.2
```

を用いたので、"-net 192.168.2.0/24 192.168.1.2" が必要になります。

上記のように static_routes は一つ以上の文字列を持つことが出来るので、多数の静的な経路を作ることができます。以下の行は 192.168.0.0/24 および 192.168.1.0/24 ネットワークを、仮想ルータ上に静的な経路として追加する例です。

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

22.2.6. ルーティングの伝搬

外部との経路をどのように定義したらよいかはすでに説明しました。しかし外部から私たちのマシンをどのようにして見つけるのかについては説明していません。

ある特定のアドレス空間 (この例では class-C のサブネット) におけるすべてのトラフィックが、到着したパケットを内部で転送するネットワーク上の特定のホストに送られるようにルーティングテーブルを設定することができるのは分かっています。

あなたのサイトにアドレス空間を割り当てる場合、 あなたのサブネットへのすべてのトラフィックがすべて PPP リンクを通じてサイトに送ってくるようにサービスプロバイダはルーティングテーブルを設定します。しかし、国境の向こう側のサイトはどのようにしてあなたの ISP へ送ることを知るのでしょうか?

割り当てられているすべてのアドレス空間の経路を維持する (分散している DNS 情報とよく似た) システムがあり、そのインターネットバックボーンへの接続点を定義しています。「バックボーン」とは国を越え、世界中のインターネットのトラフィックを運ぶ主要な信用できる幹線のことです。どのバックボーンマシンも、あるネットワークから特定のバックボーンのマシンへ向かうトラフィックと、そのバックボーンのマシンからあなたのネットワークに届くサービスプロバイダまでのチェーンのマスタテーブルのコピーを持っています。

あなたのサイトが接続 (プロバイダからみて内側にあることになります) したということを、プロバイダからバックボーンサイトへ通知することはプロバイダの仕事です。これが経路の伝搬です。

22.2.7. トラブルシューティング

経路の伝搬に問題が生じて、いくつかのサイトが接続をおこなうことができなくなることがあります。ルーティングがどこでおかしくなっているかを明らかにするのに最も有効なコマンドはおそらく traceroute(8) コマンドでしょう。このコマンドは、あなたがリモートマシンに対して接続をおこなうことができない (たとえば ping(8) に失敗するような) 場合も、同じように有効です。

traceroute(8) コマンドは、接続を試みているリモートホストを引数にして実行します。試みている経路が経由するゲートウェイホストを表示し、最終的には目的のホストにたどり着くか、コネクションの欠如によって終ってしまうかのどちらかになります。

より詳しい情報は、traceroute(8) のマニュアルページをみてください。

22.2.8. マルチキャストルーティング

FreeBSD はマルチキャストアプリケーションとマルチキャストルーティングの両方にネイティブ対応しています。マルチキャストアプリケーションを動かすのに FreeBSD で特別な設定をする必要は一切ありません。アプリケーションは普通はそのままで動くでしょう。マルチキャストルーティングに対応するには、下のオプションを追加してカーネルをコンパイルする必要があります。

```
options MROUTING
```

さらに、`/etc/mrouted.conf` を編集してルーティングデーモン mrouted(8) を設定し、トンネルと DVMRP を設置する必要があります。マルチキャスト設定についての詳細は mrouted(8) のマニュアルページを参照してください。

22.3. 無線ネットワーク

##: Eric Anderson.

22.3.1. はじめに

常にネットワークケーブルをつないでいるという面倒なことをせずに、コンピュータを使用できることは、とても有用でしょう。FreeBSD は無線のクライアントとして、さらに「アクセスポイント」としても使えます。

22.3.2. 無線の動作モード

802.11 無線デバイスの設定には、BSS と IBSS の二つの方法があります。

22.3.2.1. BSS モード

BSS モードは一般的に使われているモードです。BSS モードはインフラストラクチャモードとも呼ばれています。このモードでは、多くの無線アクセスポイントが 1 つの有線ネットワークに接続されます。それぞれのワイヤレスネットワークは固有の名称を持っています。その名称はネットワークの SSID と呼ばれます。

無線クライアントはこれらの無線アクセスポイントに接続します。 IEEE 802.11 標準は無線ネットワークが接続するのに使用するプロトコルを規定しています。SSID が設定されているときは、 無線クライアントを特定のネットワークに結びつけることができます。SSID を明示的に指定しないことにより、無線クライアントを任意のネットワークに接続することもできます。

22.3.2.2. IBSS モード

アドホックモードとも呼ばれる IBSS モードは、 一対一通信のために設計された通信方式です。実際には二種類のアドホックモードがあります。一つは IBSS モードで、アドホックモード、または IEEE アドホックモードとも呼ばれます。このモードは IEEE 802.11 標準に規定されています。もう一つはデモアドホックモードもしくは Lucent アドホックモード (そして時々、紛らわしいことに、アドホックモード) と呼ばれるモードです。このモードは古く、802.11 が標準化する以前のアドホックモードで、これは古い設備でのみ使用されるべきでしょう。ここでは、どちらのアドホックモードについてもこれ以上言及しません。

22.3.3. インフラストラクチャーモード

22.3.3.1. アクセスポイント

アクセスポイントは一つ以上の無線クライアントが、 そのデバイスをセントラルハブとして利用できるようにする無線ネットワークデバイスです。 アクセスポイントを使用している間、すべてのクライアントはアクセスポイントを介して通信します。家屋や職場、または公園などの空間を無線ネットワークで完全にカバーするために、 複数のアクセスポイントがよく使われます。

アクセスポイントは一般的に複数のネットワーク接続 (無線カードと、その他のネットワークに接続するための一つ以上の有線イーサネットアダプタ) を持っています。

アクセスポイントは、出来合いのものを購入することもできますし、FreeBSD と対応している無線カードを組み合わせて、 自分で構築することもできます。 いくつものメーカが、さまざまな機能をもった無線アクセスポイントおよび無線カードを製造しています。

22.3.3.2. FreeBSD のアクセスポイントの構築

22.3.3.2.1. 要件

FreeBSD で無線アクセスポイントを設定するためには、 互換性のある無線カードが必要です。現状では Prism チップセットのカードのみに対応しています。また FreeBSD に対応している有線ネットワークカードも必要になるでしょう (これを見つけるのは難しくないでしょう。FreeBSD は多くの異なるデバイスに対応しているからです) 。この手引きでは、無線デバイスと有線ネットワークカードに接続しているネットワーク間のトラフィックを bridge(4) したいと仮定します。

FreeBSD がアクセスポイントを実装するのに使用する hostap 機能はファームウェアの特定のバージョンで一番よく性能を発揮します。Prism 2 カードは、1.3.4 以降のバージョンのファームウェアで使用すべきです。 Prism 2.5 および Prism 3 カードでは、バージョン 1.4.9 のバージョンのファームウェアで使用すべきです。 それより古いバージョンのファームウェアは、正常に動くかもしれませんし、動かないかもしれません。現時点では、カードのファームウェアを更新する唯一の方法は、カードの製造元から入手できる Windows® 用ファームウェアアップデートユーティリティを使うものです。

22.3.3.2.2. 設定

はじめにシステムが無線カードを認識していることを確認してください。

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
        inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
        ether 00:09:2d:2d:c9:50
        media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
        status: no carrier
        ssid ""
        stationname "FreeBSD Wireless node"
        channel 10 authmode OPEN powersavemode OFF powersavesleep 100
        wepmode OFF weptxkey 1
```

細かいことは気にせず、 無線カードがインストールされていることを示す何かが表示されていることを確かめてください。PC カードを使用していて、無線インタフェースを認識できない場合、詳しい情報を得るために pccardc(8) と pccardd(8) のマニュアルページを調べてみてください。

次に、アクセスポイント用に FreeBSD のブリッジ機能を担う部分を有効にするために、モジュールを読み込む必要があるでしょう。bridge(4) モジュールを読み込むには、次のコマンドをそのまま実行します。

```
# kldload bridge
```

モジュールを読み込む時には、何もエラーはでないはずです。　もしもエラーがでたら、カーネルに bridge(4) のコードを入れてコンパイルする必要があるかもしれません。ハンドブックのブリッジの節が、この課題を成し遂げる手助けをになるかもしれません。

ブリッジ部分が準備できたので、どのインタフェース間をブリッジするのかを FreeBSD カーネルに指定する必要があります。これは、sysctl(8) を使って行います。

```
# sysctl net.link.ether.bridge=1
# sysctl net.link.ether.bridge_cfg="wi0,xl0"
# sysctl net.inet.ip.forwarding=1
```

FreeBSD 5.2-RELEASE 以降では、次のように指定しなければなりません。

```
# sysctl net.link.ether.bridge.enable=1
# sysctl net.link.ether.bridge.config="wi0,xl0"
# sysctl net.inet.ip.forwarding=1
```

さて、無線カードを設定するときです。　次のコマンドはカードをアクセスポイントとして設定します。

```
# ifconfig wi0 ssid my_net channel 11 media DS/11Mbps ↺
mediaopt hostap up stationname "FreeBSD AP"
```

この ifconfig(8) コマンド行は `wi0` インタフェースを up 状態にし、SSID を *`my_net`* に設定し、ステーション名を *`FreeBSD AP`* に設定します。`media DS/11Mbps` オプションはカードを 11Mbps モードに設定し、また `mediaopt` を実際に有効にするのに必要です。`mediaopt hostap` オプションはインタフェースをアクセスポイントモードにします。`channel 11` オプションは使用するチャネルを 802.11b に設定します。各規制地域 (regulatory domain) で有効なチャネル番号は wicontrol(8) マニュアルページに載っています。

さて、これで完全に機能するアクセスポイントが立ち上がり、動作しています。より詳しい情報については、wicontrol(8), ifconfig(8) および wi(4) のマニュアルを読むとよいでしょう。

また、下記の暗号化に関する節を読むこともおすすめします。

22.3.3.2.3. ステータス情報

一度アクセスポイントが設定されて稼働すると、管理者はアクセスポイントを利用しているクライアントを見たいと思うでしょう。　いつでも管理者は以下のコマンドを実行できます。

```
# wicontrol -l
1 station:
00:09:b7:7b:9d:16  asid=04c0, flags=3<ASSOC,AUTH>, caps=1<ESS>, ↺
rates=f<1M,2M,5.5M,11M>, sig=38/15
```

これは一つの局が、表示されているパラメータで接続していることを示します。表示された信号は、相対的な強さを表示しているだけのものとして扱われるべきです。dBm やその他の単位への変換結果は、異なるファームウェアバージョン間で異なります。

22.3.3.3. クライアント

無線クライアントはアクセスポイント、 または他のクライアントに直接アクセスするシステムです。

典型的には、 無線クライアントが有しているネットワークデバイスは、 無線ネットワークカード 1 枚だけです。

無線クライアントを設定するにはいくつか方法があります。それぞれは異なる無線モードに依存していますが、 一般的には BSS (アクセスポイントを必要とするインフラストラクチャーモード) か、IBSS (アドホック、またはピアツーピアモード) のどちらかです。ここでは、アクセスポイントと通信をするのに、両者のうちで最も広まっている BSS モードを使用します。

22.3.3.3.1. 要件

FreeBSD を無線クライアントとして設定するのに、本当に必要なものはたった 1 つだけです。FreeBSD が対応している無線カードが必要です。

22.3.3.3.2. 無線 FreeBSD クライアントの設定

設定をはじめる前に、 あなたが接続しようとする無線ネットワークについていくつか知っておかなければなりません。 この例では、*my_net* という名前で暗号化は無効になっているネットワークに接続しようとしています。

注記

この例では暗号化を行っていないのですが、 これは危険な状況です。次の節で、暗号化を有効にする方法と、 なぜそれが重要で、 暗号技術によっては完全にはあなたを保護することができないのはなぜか、ということを学ぶでしょう。

カードが FreeBSD に認識されていることを確認してください。

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
```

```
        inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
        inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
        ether 00:09:2d:2d:c9:50
        media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
        status: no carrier
        ssid ""
        stationname "FreeBSD Wireless node"
        channel 10 authmode OPEN powersavemode OFF powersavesleep 100
        wepmode OFF weptxkey 1
```

それでは、このカードをネットワークに合わせて設定しましょう。

```
# ifconfig wi0 inet 192.168.0.20 netmask 255.255.255.0 ↺
ssid my_net
```

`192.168.0.20` と `255.255.255.0` を有線ネットワークで有効な IP アドレスとネットマスクに置き換えてください。アクセスポイントは無線ネットワークと有線ネットワークの間でデータをブリッジしているため、 ネットワーク上の他のデバイスには、このデバイスが、他と同様に、有線ネットワーク上にあるかのように見えることに注意してください。

これを終えると、あなたは標準的な有線接続を使用しているかのように、有線ネットワーク上のホストに ping を送ることができるでしょう。

無線接続に関する問題がある場合は、 アクセスポイントに接続されていることを確認してください。

```
# ifconfig wi0
```

いくらか情報が表示されるはずです。その中に以下の表示があるはずです。

```
status: associated
```

もし `associated` と表示されなければ、 アクセスポイントの範囲外かもしれないし、暗号化が有効になっているかもしれないし、 または設定の問題を抱えているのかもしれません。

22.3.3.4. 暗号化

無線ネットワークを暗号化することが重要なのは、 十分保護された領域にネットワークを留める能力がもはやないからです。無線データはその周辺全体にわたって放送されるので、それを読みたいと思う人はだれでも読むことができます。そこで暗号化が役に立ちます。 電波に載せて送られるデータを暗号化することによって、 興味を抱いた者が空中からデータを取得することをずっと難しくします。

クライアントとアクセスポイント間のデータを暗号化するもっとも一般的な方法には、WEP と ipsec(4) の二種類があります。

22.3.3.4.1. WEP

WEP は Wired Equivalency Protocol (訳注: 直訳すると、有線等価プロトコル) の略語です。WEP は無線ネットワークを有線ネットワークと同程度に安全で確実なものにしようとする試みです。 残念ながら、これはすでに破られており、 破るのはそれほど苦労しません。 これは、機密データを暗号化するという場合に、 これに頼るものではないということも意味します。

なにも無いよりはましなので、次のコマンドを使って、あなたの新しい FreeBSD アクセスポイント上で WEP を有効にしてください。

```
# ifconfig wi0 inet up ssid my_net wepmode on wepkey ↺
0x1234567890 media DS/11Mbps mediaopt hostap
```

クライアントについては次のコマンドで WEP を有効にできます。

```
# ifconfig wi0 inet 192.168.0.20 netmask 255.255.255.0 ↺
ssid my_net wepmode on wepkey 0x1234567890
```

0x1234567890 をより特異なキーに変更すべきであることに注意してください。

22.3.3.4.2. IPsec

ipsec(4) はネットワーク上で交わされるデータを暗号化するための、はるかに頑健で強力なツールです。 これは無線ネットワーク上のデータを暗号化する明らかに好ましい方法です。ハンドブック内の IPsec 節で ipsec(4) セキュリティ、およびその実装方法の詳細を読むことができます。

22.3.3.5. ツール

無線ネットワークをデバッグしたり設定するのに使うツールがわずかばかりあります。ここでその一部と、それらが何をしているか説明します。

22.3.3.5.1. bsd-airtools パッケージ

bsd-airtools パッケージは、WEP キークラッキング、アクセスポイント検知などの無線通信を監査するツールを含む完備されたツール集です。

bsd-airtools ユーティリティは net/bsd-airtools port からインストールできます。ports のインストールに関する情報はこのハンドブックの 5章アプリケーションのインストール - packages と ports を参照してください。

`dstumbler` プログラムは、アクセスポイントの発見および S/N 比のグラフ化をできるようにするパッケージツールです。 アクセスポイントを立ち上げて動かすのに苦労しているなら、`dstumbler` はうまく行く手助けになるかもしれません。

無線ネットワークの安全性をテストするのに、「dweputils」(`dwepcrack`, `dwepdump` および `dwepkeygen`) を使用することで、WEP があなたの無線安全性への要求に対する正しい解決策かどうか判断するのを助けられるかもしれません。

22.3.3.5.2. `wicontrol`, `ancontrol` および `raycontrol` ユーティリティ

これらは、無線ネットワーク上で無線カードがどのように動作するかを制御するツールです。上記の例では、無線カードが `wi0` インタフェースであるので、wicontrol(8) を使用することに決めました。もし Cisco の無線デバイスを持っている場合は、それは `an0` として動作するでしょうから、ancontrol(8) を使うことになるでしょう。

22.3.3.5.3. `ifconfig` コマンド

ifconfig(8) は wicontrol(8) と同じオプションの多くを処理できますが、いくつかのオプションを欠いています。コマンドライン引数とオプションについて ifconfig(8) を参照してください。

22.3.3.6. 対応しているカード

22.3.3.6.1. アクセスポイント

現在のところ (アクセスポイントとして) BSS モードに対応した唯一のカードは Prism 2, 2.5 または 3 チップセットを利用したデバイスです。wi(4) に完全な一覧があります。

22.3.3.6.2. クライアント

現在、FreeBSD では、ほとんどすべての 802.11b 無線カードに対応しています。Prism, Spectrum24, Hermes, Aironet または Raylink のチップセットを利用したほとんどのカードは、(アドホック、ピアツーピア、そして BSS の) IBSS モードで無線ネットワークカードとして動作するでしょう。

22.4. Bluetooth

##: Pav Lucistnik.

22.4.1. はじめに

Bluetooth は免許のいらない 2.4 GHz の帯域を利用して、10 m 程度のパーソナルネットワークを作る無線技術です。ネットワークはたいていの場合、その場その場で、携帯電話や PDA やノートパソコンなどの携帯デバイスから形成されます。Wi-Fi などの他の有名な無線技術とは違い、Bluetooth はより高いレベルのサービスを提供します。たとえば、FTP のようなファイルサーバ、ファイルのプッシュ、音声伝送、シリアル線のエミュレーションなどのサービスです。

FreeBSD 内での Bluetooth スタックは Netgraph フレームワーク (netgraph(4) 参照) を使って実現されています。ng_ubt(4) ドライバは、多種多様な Bluetooth USB ドングルに対応しています。Broadcom BCM2033 チップを搭載した Bluetooth デバイスは ubtbcmfw(4) および ng_ubt(4) ドライバによって対応されています。3Com Bluetooth PC カード 3CRWB60-A は ng_bt3c(4) ドライバによって対応されていま

す。シリアルおよび UART を搭載した Bluetooth デバイスは sio(4), ng_h4(4) および hcseriald(8) ドライバによって対応されています。この節では USB Bluetooth ドングルの使用法について説明します。Bluetooth に対応しているのは FreeBSD 5.0 以降のシステムです。

訳注

5.0, 5.1 Release ではカーネルモジュールは利用可能ですが、 種々のユーティリティとマニュアルは標準でコンパイルされていません。

22.4.2. デバイスの挿入

デフォルトでは Bluetooth デバイスドライバはカーネルモジュールとして利用できます。デバイスを接続する前に、カーネルにドライバを読み込む必要があります。

```
# kldload ng_ubt
```

Bluetooth デバイスがシステム起動時に存在している場合、/boot/loader.conf からモジュールを読み込んでください。

```
ng_ubt_load="YES"
```

USB ドングルを挿してください。コンソールに (または syslog に) 下記のような表示が現れるでしょう。

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-
out=0x2
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-
out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

/usr/share/examples/netgraph/bluetooth/rc.bluetooth を /etc/rc.bluetooth のようなどこか便利な場所にコピーしてください。このスクリプトは Bluetooth スタックを開始および終了させるのに使われます。デバイスを抜く前にスタックを終了するのはよい考えですが、(たいていの場合) しなくても致命的ではありません。スタックを開始するときに、下記のような出力がされます。

```
# /etc/rc.bluetooth start ubt0
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
```

```
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
<Paging scheme> <Power control> <Transparent SCO data>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

22.4.3. ホストコントローラインタフェース (HCI)

ホストコントローラインタフェース (HCI) は、ベースバンドコントローラおよびリンクマネージャへのコマンドインタフェースを提供し、ハードウェアステータスおよびコントロールレジスタへアクセスします。このインタフェースは Bluetooth ベースバンド機能へアクセスする画一的な方法を提供します。ホストの HCI 層は Bluetooth ハードウェア上の HCI ファームウェアと、データとコマンドをやり取りします。ホストコントローラトランスポート層 (つまり物理的なバス) のドライバは、両方の HCI 層に相互に情報を交換する能力を与えます。

一つの Bluetooth デバイスにつき、hci タイプの Netgraph ノードが一つ作成されます。HCI ノードは通常 Bluetooth デバイスドライバノード (下流) と L2CAP ノード (上流) に接続されます。すべての HCI 動作はデバイスドライバノード上ではなく、HCI ノード上で行われなくてはいけません。HCI ノードのデフォルト名は「devicehci」です。詳細については ng_hci(4) マニュアルを参照してください。

最も一般的なタスクの一つに、無線通信的に近傍にある Bluetooth デバイスの発見があります。この動作は inquiry (問い合わせ) と呼ばれています。Inquiry や他の HCI に関連した動作は hccontrol(8) ユーティリティによってなされます。下記の例は、どの Bluetooth デバイスが通信圏内にあるかを知る方法を示しています。デバイスのリストが表示されるには数秒かかります。リモートデバイスは discoverable (発見可能な) モードにある場合にのみ inquiry に返答するということに注意してください。

```
% hccontrol -n ubt0hci inquiry
Inquiry result, num_responses=1
Inquiry result #0
       BD_ADDR: 00:80:37:29:19:a4
       Page Scan Rep. Mode: 0x1
       Page Scan Period Mode: 00
       Page Scan Mode: 00
       Class: 52:02:04
       Clock offset: 0x78ef
Inquiry complete. Status: No error [00]
```

BD_ADDR は Bluetooth デバイスに固有のアドレスです。これはネットワークカードの MAC アドレスに似ています。このアドレスはデバイスとの通信を続けるのに必要となります。BD_ADDR に人間が判読しやすい名前を割り当てることもできます。**/etc/bluetooth/hosts** ファイルには、既知の Bluetooth ホストに関する情報が含まれています。次の例はリモートデバイスに割り当てられている、人間が判読しやすい名前を得る方法を示しています。

```
% hccontrol -n ubt0hci remote_name_request ↺
00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav's T39
```

リモートの Bluetooth デバイス上で inquiry を実行すると、あなたのコンピュータは「your.host.name (ubt0)」と認識されます。ローカルデバイスに割り当てられた名前はいつでも変更できます。

Bluetooth システムは一対一接続 (二つの Bluetooth ユニットだけが関係します) または一対多接続を提供します。一対多接続では、接続はいくつかの Bluetooth デバイス間で共有されます。次の例は、ローカルデバイスに対するアクティブなベースバンド接続のリストを得る方法を示しています。

```
% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4     41  ACL    0 MAST    NONE       0     0 OPEN
```

connection handle はベースバンド接続の終了が必要とされるときに便利です。もっとも、通常はこれを手動で行う必要はありません。Bluetooth スタックはアクティブでないベースバンド接続を自動的に終了します。

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]
```

利用可能な HCI コマンドの完全な一覧を得るには、`hccontrol help` を参照してください。HCI コマンドのほとんどはスーパユーザ権限を必要としません。

22.4.4. ロジカルリンクコントロールおよびアダプテーションプロトコル (L2CAP)

ロジカルリンクコントロールおよびアダプテーションプロトコル (L2CAP) は、プロトコル多重化ケーパビリティおよび分割・再編成動作を備えた、 上位層プロトコルへのコネクション指向およびコネクションレスデータサービスを提供します。 L2CAP は上位層プロトコルおよびアプリケーションが 64 KB までの長さの L2CAP データパケットを送受信することを可能にします。

L2CAP は チャネル の概念に基づいています。 チャネルはベースバンド接続の上位に位置する論理的な接続です。 それぞれのチャネルは多対一の方法で一つのプロトコルに結びつけられます。 複数のチャネルを同じプロトコルに結びつけることは可能ですが、一つのチャネルを複数のプロトコルに結びつけることはできません。チャネル上で受け取られたそれぞれの L2CAP パケットは、適切なより上位のプロトコルに渡されます。複数のチャネルは同じベースバンド接続を共有できます。

一つの Bluetooth デバイスに対して、l2cap タイプの Netgraph ノードが一つ作成されます。L2CAP ノードは通常 Bluetooth HCI ノード (下流) と Bluetooth ソケットノード

(上流) に接続されます。L2CAP ノードのデフォルト名は「devicel2cap」です。詳細については ng_l2cap(4) マニュアルを参照してください。

便利なコマンドに、他のデバイスに ping を送ることができる l2ping(8) があります。Bluetooth 実装によっては、送られたデータすべては返さないことがあります。したがって次の例で 0 バイト は正常です。

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

l2control(8) ユーティリティは L2CAP ノード上でさまざまな操作を行うのに使われます。この例は、ローカルデバイスに対する論理的な接続 (チャネル) およびベースバンド接続の一覧を得る方法を示しています。

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
Remote BD_ADDR     SCID/ DCID   PSM  IMTU/ OMTU State
00:07:e0:00:0b:ca    66/   64     3   132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
L2CAP connections:
Remote BD_ADDR    Handle Flags Pending State
00:07:e0:00:0b:ca     41 O           0 OPEN
```

別の診断ツールが btsockstat(1) です。これは netstat(1) と同様の作業を、Bluetooth ネットワークに関するデータ構造についての行います。下記の例は上の l2control(8) と同じ論理的な接続を示します。

```
% btsockstat
Active L2CAP sockets
PCB      Recv-Q Send-Q Local address/PSM       Foreign address   CID
  State
c2afe900      0      0 00:02:72:00:d4:1a/3     00:07:e0:00:0b:ca 66
  OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127   0     Yes  OPEN
Active RFCOMM sockets
PCB      Recv-Q Send-Q Local address     Foreign address   Chan ↺
DLCI State
c2e8bc80      0    250 00:02:72:00:d4:1a 00:07:e0:00:0b:ca 3    6
  OPEN
```

22.4.5. RFCOMM プロトコル

RFCOMM プロトコルは L2CAP プロトコルを介してシリアルポートのエミュレーションを提供します。このプロトコルは ETSI (訳注: 欧州電気通信標準化機構) 標準 TS

07.10 に基づいています。RFCOMM プロトコルは、単純な伝送プロトコルに RS-232 (EIATIA-232-E) シリアルポートの 9 本の結線をエミュレートする項目を加えたものです。RFCOMM プロトコルは、二つの Bluetooth デバイス間で、最大 60 までの同時接続 (RFCOMM チャネル) に対応しています。

RFCOMM の目的から、完全な通信経路は、異なるデバイス上 (通信の端点) で動作している二つのアプリケーションと、その間の通信セグメントを含んでいます。RFCOMM は、それが動いているデバイスのシリアルポートを利用するアプリケーションをカバーするためのものです。通信セグメントはあるデバイスから他のデバイスへの Bluetooth リンクです (直接接続)。

RFCOMM は直接接続している場合のデバイス間の接続、またはネットワークの場合のデバイスとモデムの間の接続にだけ関係があります。RFCOMM は、一方が Bluetooth 無線技術で通信し、もう一方で有線インタフェースを提供するモジュールのような、他の構成にも対応できます。

FreeBSD では RFCOMM プロトコルは Bluetooth ソケット層に実装されています。

22.4.6. デバイスのペアリング

デフォルトでは Bluetooth 通信は認証されておらず、すべてのデバイスが他のすべてのデバイスと通信できます。Bluetooth デバイス (たとえば携帯電話) は特定のサービス (たとえばダイアルアップサービス) を提供するために、認証を要求することも選択できます。Bluetooth 認証は通常 PIN コード で行われます。PIN コードは最長 16 文字のアスキー文字列です。ユーザは両デバイスで同じ PIN コードを入力することを要求されます。一度 PIN コードを入力すると、両デバイスは リンクキー を作成します。その後、リンクキーはそのデバイス自身または、不揮発性記憶デバイス内に格納できます。次の機会には、両デバイスは前に作成されたリンクキーを使用するでしょう。このような手続きをペアリング (pairing) と呼びます。いずれかのデバイス上でリンクキーが失われたときには、ペアリングをやり直さなければならないことに注意してください。

hcsecd(8) デーモンが Bluetooth 認証要求のすべてを扱う責任を負っています。デフォルトの設定ファイルは `/etc/bluetooth/hcsecd.conf` です。PIN コードが「1234」に設定された携帯電話に関する例は以下の通りです。

```
device {
        bdaddr  00:80:37:29:19:a4;
        name    "Pav's T39";
        key     nokey;
        pin     "1234";
      }
```

PIN コードには (長さを除いて) 制限はありません。いくつかのデバイス (たとえば Bluetooth ヘッドフォン) には固定的な PIN コードが組み込まれているかもしれません。`-d` オプションは hcsecd(8) デーモンがフォアグラウンドで動作するように強制するため、何が起きているのか確認しやすくなります。リモートデバイスがペアリングを受け取るように設定して、リモートデバイスへの Bluetooth 接続を開始してください。リ

モートデバイスはペアリングが受け入れらた、と応答して PIN コードを要求するでしょう。`hcsecd.conf` 内にあるのと同じ PIN コードを入力してください。これであなたの PC とリモートデバイスがペアとなりました。 また、リモートデバイスからペアリングを開始することもできます。以下は `hcsecd` の出力例です。

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote ↺
bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr ↺
0:80:37:29:19:a4, name 'Pav's T39', link key doesn't exist
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for ↺
remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote ↺
bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr ↺
0:80:37:29:19:a4, name 'Pav's T39', PIN code exists
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote ↺
bdaddr 0:80:37:29:19:a4
```

22.4.7. サービスディスカバリプロトコル (SDP)

サービスディスカバリプロトコル (SDP) は、クライアントアプリケーションが、サーバアプリケーションが提供するサービスの存在とその属性を発見する手段を提供します。 サービスの属性には提示されているサービスのタイプまたはクラス、および、サービスを利用するのに必要な仕組みまたはプロトコルの情報が含まれます。

SDP には SDP サーバと SDP クライアント間の通信が含まれます。SDP サーバは、サーバに関連づけられたサービスの特性について記述しているサービスレコードの一覧を維持しています。各サービスレコードにはそれぞれ 1 つのサービスの情報が書かれています。クライアントは SDP リクエストを出すことによって、SDP サーバが維持しているサービスレコードから情報を検索できます。クライアントまたはクライアントに関連づけられたアプリケーションがサービスを利用することにしたら、サービスを利用するためには、サービスプロバイダへの接続を別途開かなければなりません。SDP はサービスとそれらの属性を発見するための仕組みを提供しますが、 そのサービスを利用するための仕組みは提供しません。

通常 SDP クライアントは希望するサービスの特性に基づいてサービスを検索します。しかしながら、サービスに関する事前の情報なしに、どのタイプのサービスが SDP サーバのサービスレコードに記述されているか知ることが望ましいことがあります。この、提供されている任意のサービスを閲覧する手順を、ブラウジング (browsing) と呼びます。

現在のところ Bluetooth SDP サーバおよびクライアントは、ここ からダウンロードできる第三者パッケージ sdp-1.5 で実装されています。sdptool はコマンドラインの SDP クライアントです。次の例は SDP ブラウズの問い合わせ方法を示しています。

```
# sdptool browse 00:80:37:29:19:a4
Browsing 00:80:37:29:19:A4 ...
Service Name: Dial-up Networking
Protocol Descriptor List:
```

```
 "L2CAP" (0x0100)
 "RFCOMM" (0x0003)
   Channel: 1

Service Name: Fax
Protocol Descriptor List:
 "L2CAP" (0x0100)
 "RFCOMM" (0x0003)
   Channel: 2

Service Name: Voice gateway
Service Class ID List:
 "Headset Audio Gateway" (0x1112)
 "Generic Audio" (0x1203)
Protocol Descriptor List:
 "L2CAP" (0x0100)
 "RFCOMM" (0x0003)
   Channel: 3
```

... 等々。それぞれのサービスは属性の一覧 (たとえば RFCOMM チャネル) を持っていることに注意してください。サービスによっては、 属性のリストの一部についてメモをとっておく必要があるかもしれません。Bluetooth 実装のいくつかは、サービスブラウジングに対応しておらず、空の一覧を返してくるかもしれません。この場合、特定のサービスを検索をすることは可能です。下記の例は OBEX オブジェクトプッシュ (OPUSH) サービスを検索する方法です。

```
# sdptool search --bdaddr 00:07:e0:00:0b:ca OPUSH
```

FreeBSD 上における Bluetooth クライアントへのサービス提供は sdpd サーバが行います。

```
# sdpd
```

sdptool は、ローカル SDP サーバにサービスを登録するのにも用いられます。下記の例は PPP (LAN) サービスを備えたネットワークアクセスを登録する方法を示しています。一部のサービスでは属性 (たとえば RFCOMM チャネル) を要求することに注意してください。

```
# sdptool add --channel=7 LAN
```

ローカル SDP サーバに登録されたサービスの一覧は SDP ブラウザの問い合わせを「特別な」BD_ADDR に送ることで得られます。

```
# sdptool browse ff:ff:ff:00:00:00
```

22.4.8. ダイアルアップネットワーク (DUN) および PPP (LAN) を用いたネットワークアクセスプロファイル

ダイアルアップネットワーク (DUN) プロファイルはほとんどの場合、モデムや携帯電話とともに使用されます。このプロファイルが対象とする場面は以下のものです。

- コンピュータから携帯電話またはモデムを、ダイアルアップインターネットアクセスサーバへの接続、 または他のダイアルアップサービスを利用するための無線モデムとして使うこと

- データ呼び出しを受けるための、コンピュータによる携帯電話またはモデムの使用

PPP (LAN) によるネットワークアクセスプロファイルは、次の状況で利用できます。

- 単一の Bluetooth デバイスへの LAN アクセス

- マルチ Bluetooth デバイスへの LAN アクセス

- (シリアルケーブルエミュレーション上の PPP ネットワーク接続を使用した) PC から PC への接続

FreeBSD ではどちらのプロファイルも ppp(8) と rfcomm_pppd(8) (RFCOMM Bluetooth 接続を PPP が制御可能なように変換するラッパ) で実装されています。いずれかのプロファイルが使用可能となる前に、`/etc/ppp/ppp.conf` 内に新しい PPP ラベルが作成されていなければなりません。例については、rfcomm_pppd(8) のマニュアルページを参照してください。

次の例では、DUN RFCOMM チャネル上で BD_ADDR が 00:80:37:29:19:a4 のリモートデバイスへの RFCOMM 接続を開くのに rfcomm_pppd(8) が使われます。実際の RFCOMM チャネル番号は SDP を介してリモートデバイスから得ます。手動で RFCOMM チャネルを指定することもでき、その場合 rfcomm_pppd(8) は SDP 問い合わせを実行しません。リモートデバイス上の RFCOMM チャネルを見つけるには、sdptool を使ってください。

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-
dialup
```

PPP (LAN) サービスでネットワークアクセスを提供するためには、sdpd サーバが動いていなければなりません。これはローカル SDP サーバに LAN サービスを登録するのにも必要です。LAN サービスは RFCOMM チャネル属性を必要とすることに注意してください。`/etc/ppp/ppp.conf` ファイル内に LAN クライアントの新しいエントリを作成しなければなりません。例については rfcomm_pppd(8) のマニュアルページを参照してください。最後に、RFCOMM PPP サーバが実行され、ローカル SDP サーバに登録されているのと同じ RFCOMM チャネルで待ち受けていなければなりません。次の例は RFCOMM PPP サーバを起動する方法を示しています。

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

22.4.9. OBEX プッシュ (OPUSH) プロファイル

OBEX はモバイルデバイス間で広く使われている単純なファイル転送プロトコルです。これは主に赤外線通信で利用されており、ノートパソコンや PDA 間の汎用的なファイル

転送、および PIM アプリケーションを搭載した携帯電話その他のデバイス間で名刺やカレンダーエントリを転送するのに用いられます。

OBEX サーバおよびクライアントは、ここ からダウンロードできる obexapp-1.0 という第三者のパッケージとして実装されています。このパッケージは openobex ライブラリ (上記の obexapp に含まれます) および devel/glib12 port を必要とします。なお、obexapp はルート権限を必要としません。

OBEX クライアントは OBEX サーバとの間でオブジェクトを渡したり (プッシュ) および受け取ったり (プル) するのに使用されます。オブジェクトは、たとえば名刺や予定などになります。OBEX クライアントは RFCOMM チャネル番号を SDP によってリモートデバイスから得ることができます。これは RFCOMM チャネル番号の代わりにサービス名を指定することによって行うことができます。対応しているサービス名は IrMC, FTRN および OPUSH です。RFCOMM チャネルを番号で指定することもできます。下記は、デバイス情報オブジェクトを携帯電話から受け取り、新しいオブジェクト (名刺) が携帯電話に渡される場合の OBEX セッションの例です。

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get
get: remote file> telecom/devinfo.txt
get: local file> devinfo-t39.txt
Success, response: OK, Success (0x20)
obex> put
put: local file> new.vcf
put: remote file> new.vcf
Success, response: OK, Success (0x20)
obex> di
Success, response: OK, Success (0x20)
```

OBEX プッシュサービスを提供するためには、sdpd サーバが実行されていなければなりません。また OPUSH サービスをローカル SDP サーバに登録することも必要です。なお、OPUSH サービスには RFCOMM チャネル属性が必要です。渡されるオブジェクトをすべて格納するルートフォルダを作成しなければいけません。 ルートフォルダのデフォルトパスは `/var/spool/obex` です。最後に OBEX サーバが実行され、ローカル SDP サーバに登録されているのと同じ RFCOMM チャネルで待ち受けていなければなりません。下記の例は OBEX サーバの起動方法を示します。

```
# obexapp -s -C 10
```

22.4.10. シリアルポート (SP) プロファイル

シリアルポート (SP) プロファイルは Bluetooth デバイスが RS232 (または同様の) シリアルケーブルエミュレーションを行えるようにします。このプロファイルが対象とする場面は、レガシーアプリケーションが、仮想シリアルポート抽象を介して Bluetooth をケーブルの代替品として使うところです。

rfcomm_sppd(1) ユーティリティはシリアルポートプロファイルを実装します。 Pseudo tty が仮想シリアルポート抽象概念として用いられます。下記の例はリモートデバイスの

シリアルポートサービスへ接続する方法を示します。なお、RFCOMM チャネルを指定する必要はありません。— rfcomm_sppd(1) は SDP を介してリモートデバイスからその情報を得ることができます。これを上書きしたい場合にはコマンドラインで RFCOMM チャネルを指定してください。

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t /dev/ttyp6
rfcomm_sppd[94692]: Starting on /dev/ttyp6...
```

接続された pseudo tty はシリアルポートとして利用することができます。

```
# cu -l ttyp6
```

22.4.11. トラブルシューティング

22.4.11.1. リモートデバイスが接続できません

古い Bluetooth デバイスのなかにはロールスイッチング (role switching) に対応していないものがあります。 デフォルトでは FreeBSD が新しい接続を受け付けるときに、ロールスイッチを実行してマスタになろうとします。これに対応していないデバイスは接続できないでしょう。 なお、ロールスイッチングは新しい接続が確立されるときに実行されるので、 ロールスイッチングに対応しているかどうかリモートデバイスに問い合わせることはできません。ローカル側でロールスイッチングを無効にする HCI オプションがあります。

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

22.4.11.2. 何かがうまくいっていないみたいです。何が実際に起こっているか確認できますか?

できます。ここ からダウンロードできる第三者パッケージ hcidump-1.5 を使ってください。 hcidump ユーティリティは tcpdump(1) と似ています。 これはターミナル上の Bluetooth パケットの内容の表示および Bluetooth パケットをファイルにダンプするのに使えます。

22.5. ブリッジ

##: Steve Peterson.

22.5.1. はじめに

IP サブネットを作成して、 それらのセグメントをルータを使って接続することなしに、(Ethernet セグメントのような) 一つの物理ネットワークを二つのネットワークセグメントに分割することはとても有効な場合があります。 この方法で二つのネットワークを繋ぐデバイスは 「ブリッジ」 と呼ばれます。 二つのネットワークインタフェースカードを持つ FreeBSD システムは、ブリッジとして動作することができます。

ブリッジは、各ネットワークインタフェイスに繋がるデバイスの MAC 層のアドレス (Ethernet アドレス) を記憶することにより動作します。ブリッジはトラフィックの送信元と受信先が異なったネットワーク上にある場合にのみトラフィックを転送します。

多くの点で、ブリッジはポート数の少ない Ethernet スイッチのようなものといえます。

22.5.2. ブリッジがふさわしい状況

今日ブリッジが活躍する場面は大きく分けて二つあります。

22.5.2.1. トラフィックの激しいセグメント

ひとつは、 物理ネットワークセグメントがトラフィック過剰になっているが、 なんらかの理由によりネットワークをサブネットに分け、ルータで接続することができない場合です。

編集部門と製作部門がおなじサブネットに同居している新聞社を例に考えてみましょう。編集部門のユーザはファイルサーバとして全員サーバ A を利用し、製作部門のユーザはサーバ B を利用します。すべてのユーザを接続するのには Ethernet が使われており、高負荷となったネットワークは遅くなってしまいます。

もし編集部門のユーザを一つのネットワークセグメントに分離することができ、 製作部門のユーザも同様にできるのなら、 二つのネットワークセグメントをブリッジで繋ぐことができます。ブリッジの「反対」側へ向かうネットワークトラフィックだけが転送され、各ネットワークセグメントの混雑は緩和されます。

22.5.2.2. パケットフィルタ/帯域制御用ファイアウォール

もうひとつはネットワークアドレス変換 (NAT) を使わずにファイアウォール機能を利用したい場合です。

ここでは DSL もしくは ISDN で ISP に接続している小さな会社を例にとってみましょう。この会社は ISP からグローバル IP アドレスを 13 個割り当てられており、ネットワーク上には 10 台の PC が存在します。このような状況では、サブネット化にまつわる問題から、ルータを用いたファイアウォールを利用することは困難です。

ブリッジを用いたファイアウォールなら、 IP アドレスの問題を気にすること無く、 DSL/ISDN ルータの下流側に置くように設定できます。

22.5.3. ブリッジを設定する

22.5.3.1. ネットワークインタフェースカードの選択

ブリッジを利用するには少なくとも 2 枚のネットワークカードが必要です。残念なことに FreeBSD 4.0 ではすべてのネットワークインタフェースカードがブリッジ機能に対応しているわけではありません。カードに対応しているかどうかについては bridge(4) を参照してください。

以下に進む前に、二枚のネットワークカードをインストールしてテストしてください。

22.5.3.2. カーネルコンフィグレーションの変更

カーネルでブリッジ機能を有効にするには

```
options BRIDGE
```

という行をカーネルコンフィグレーションファイルに追加して カーネルを再構築してください。

22.5.3.3. ファイアウォール対応

ファイアウォールとしてブリッジを利用しようとしている場合には IPFIREWALL オプションも指定する必要があります。 ブリッジをファイアウォールとして設定する際の一般的な情報に関しては、「ファイアウォール」を参照してください。

IP 以外のパケット (ARP など) がブリッジを通過するようにするためには、 ファイアウォール用オプションを設定しなければなりません。 このオプションは IPFIREWALL_DEFAULT_TO_ACCEPT です。この変更により、デフォルトではファイアウォールがすべてのパケットを受け入れるようになることに注意してください。この設定を行う前に、この変更が自分のルールセットにどのような影響をおよぼすかを把握しておかなければなりません。

22.5.3.4. 帯域制御機能

ブリッジで帯域制御機能を利用したい場合、 カーネルコンフィグレーションで DUMMYNET オプションを加える必要があります。詳しい情報に関しては dummynet(4) を参照してください。

22.5.4. ブリッジを有効にする

ブリッジを有効にするには、/etc/sysctl.conf に以下の行を加えてください。

```
net.link.ether.bridge=1
```

指定したインタフェースでブリッジを可能にするには以下を加えてください。

```
net.link.ether.bridge_cfg=if1,if2
```

(*if1* および *if2* は二つのネットワークインタフェースの名前に置き換えてください)。ブリッジを経由したパケットを ipfw(8) でフィルタしたい場合には、以下の行も付け加える必要があります

```
net.link.ether.bridge_ipfw=1
```

FreeBSD 5.2-RELEASE 以降では、かわりに以下の行を使用してください。

```
net.link.ether.bridge.enable=1
 net.link.ether.bridge.config=if1,if2
```

```
net.link.ether.bridge.ipfw=1
```

22.5.5. その他の情報

ネットワークからブリッジに telnet(1) したい場合、ネットワークカードの一つに IP アドレスを割り当てるのが正しいです。一般的に、両方のカードに IP アドレスを割り当てるのはよい考えではないとされています。

ネットワーク内に複数のブリッジを設置する場合、任意のワークステーション間で一つ以上の経路を持つことはできません。技術的には、これはスパニングツリーのリンク制御はサポートされていない、ということを意味します。

ブリッジは、ping(8) にかかる時間を遅らせることがあります。特に、一方のセグメントからもう一方へのトラフィックでそうなります。

22.6. NFS

######: Tom Rhodes.
##: Bill Swingle.

FreeBSD がサポートしている多くのファイルシステムの中には、NFS とも呼ばれているネットワークファイルシステムがあります。NFS はあるマシンから他のマシンへと、ネットワークを通じてディレクトリとファイルを共有することを可能にします。NFS を使うことで、ユーザやプログラムはリモートシステムのファイルを、それがローカルファイルであるかのようにアクセスすることができます。

NFS が提供可能な最も特筆すべき利点いくつかは以下のものです。

- 一般的に使われるデータを単一のマシンに納めることができ、ユーザはネットワークを通じてデータにアクセスできるため、 ローカルワークステーションが使用するディスク容量が減ります。
- ネットワーク上のすべてのマシンに、 ユーザが別々にホームディレクトリを持つ必要がありません。NFS サーバ上にホームディレクトリが設定されれば、ネットワークのどこからでもアクセス可能です。
- フロッピーディスクや CDROM ドライブ、ZIP ドライブなどのストレージデバイスを、ネットワーク上の他のマシンで利用することができます。ネットワーク全体のリムーバブルドライブの数を減らせるかもしれません。

22.6.1. NFS はどのように動作するのか

NFS は最低二つの主要な部分、サーバと一つ以上のクライアントからなります。クライアントはサーバマシン上に格納されたデータにリモートからアクセスします。 これが適切に機能するには、いくつかのプロセスが設定されて実行されていなければなりません。

注記

FreeBSD 5.X では `portmap` ユーティリティは rpcbind ユーティリティに置き換わりました。したがって FreeBSD 5.X では、ユーザは下記の例で、portmap の例のすべてを `rpcbind` に置き換える必要があります。

サーバは以下のデーモンを動作させなければなりません。

デーモン	説明
nfsd	NFS クライアントからのリクエストを処理する NFS デーモン
mountd	nfsd(8) から渡されたリクエストを実際に実行する NFS マウントデーモン
portmap	NFS サーバの利用しているポートを NFS クライアントから取得できるようにするためのポートマッパデーモン

クライアント側では nfsiod というデーモンも実行できます。nfsiod デーモンは NFS サーバからのリクエストを処理します。これは任意であり、性能を改善しますが、通常の正しい動作には必要としません。詳細については nfsiod(8) マニュアルページを参照してください。

22.6.2. NFS の設定

NFS の設定は比較的素直な工程です。動かさなければならないプロセスは `/etc/rc.conf` ファイルを少し変更すれば起動時に実行させられます。

NFS サーバでは `/etc/rc.conf` ファイルの中で、以下のオプションが設定されていることを確かめてください。

```
portmap_enable="YES"
nfs_server_enable="YES"
mountd_flags="-r"
```

`mountd` は NFS サーバが有効になっていれば、自動的に実行されます。

クライアント側では `/etc/rc.conf` 内に以下の設定があることを確認してください。

```
nfs_client_enable="YES"
```

/etc/exports ファイルは NFS サーバがどのファイルシステムをエクスポート (ときどき「共有」と呼ばれます) するのかを指定します。/etc/exports ファイル中の各行は、エクスポートするファイルシステム、およびそのファイルシステムにアクセスできるマシンを指定します。 ファイルシステムにアクセスできるマシンとともに、 アクセスオプションも指定できます。このファイルで指定できるオプションはたくさんありますが、ここではほんの少しだけ言及します。exports(5) マニュアルページを読めば、 他のオプションは簡単にみつけられるでしょう。

いくつか /etc/exports の設定例を示します。

以下の例はファイルシステムのエクスポートの考え方を示しますが、あなたの環境とネットワーク設定に応じて設定は少し変わるでしょう。 たとえば次の行は /cdrom ディレクトリを、サーバと同じドメイン名か (そのため、いずれもドメイン名がありません)、 /etc/hosts に記述されている三つの例となるマシンに対してエクスポートします。 -ro フラグは共有されるファイルシステムを読み込み専用にします。このフラグにより、リモートシステムは共有されたファイルシステムに対して何の変更も行えなくなります。

```
/cdrom -ro host1 host2 host3
```

以下の設定は IP アドレスで指定した 3 つのホストに対して /home をエクスポートします。この設定はプライベートネットワークで DNS が設定されていない場合に便利でしょう。 内部のホスト名に対して /etc/hosts を設定するという手段もあります。 詳細については hosts(5) を参照してください。 -alldirs フラグはサブディレクトリがマウントポイントとなることを認めます。言い替えると、これはサブディレクトリをマウントしませんが、クライアントが要求するか、または必要とするディレクトリだけをマウントできるようにします。

```
/home  -alldirs  10.0.0.2 10.0.0.3 10.0.0.4
```

以下の設定は、サーバとは異なるドメイン名の 2 台のクライアントがアクセスできるように /a をエクスポートします。 -maproot=root フラグは、リモートシステムの root ユーザが、エクスポートされたファイルシステムに root として書き込むことを許可します。 -maproot=root フラグが無ければ、リモートマシンの root 権限を持っていても、共有されたファイルシステム上のファイルを変更することはできないでしょう。

```
/a  -maproot=root  host.example.com box.example.org
```

クライアントがエクスポートされたファイルシステムにアクセスするためには、 そうする権限が与えられていなければなりません。/etc/exports ファイルに クライアントが含まれているかどうか確認してください。

/etc/exports ファイルでは、それぞれの行が一つのファイルシステムを一つのホストにエクスポートすることを表します。 リモートホストはファイルシステム毎に一度だけ指定することができ、それに加えて一つのデフォルトエントリを置けます。たとえば /usr が単一のファイルシステムであると仮定します。次の /etc/exports は無効です。

```
/usr/src   client
/usr/ports client
```

単一のファイルシステムである /usr は、2 行に渡って、同じホスト client へエクスポートされています。この場合、正しい書式は次のとおりです。

```
/usr/src /usr/ports  client
```

あるホストにエクスポートされるある 1 つのファイルシステムのプロパティは、1 行ですべて指定しなければなりません。 クライアントの指定のない行は、単一のホストとして扱われます。 これはファイルシステムをエクスポートできる方法を制限しますが、 多くの場合これは問題になりません。

下記は、/usr および /exports がローカルファイルシステムである場合の、有効なエクスポートリストの例です。

```
# Export src and ports to client01 and client02, but only
# client01 has root privileges on it
/usr/src /usr/ports -maproot=root    client01
/usr/src /usr/ports               client02
# The client machines have root and can mount anywhere
# on /exports. Anyone in the world can mount /exports/obj read-only
/exports -alldirs -maproot=root      client01 client02
/exports/obj -ro
```

変更が有効となるように、/etc/exports が変更されたら mountd を再起動しなければなりません。これは mountd プロセスに HUP シグナルを送ることで実行できます。

```
# kill -HUP `cat /var/run/mountd.pid`
```

他には、再起動すれば、FreeBSD はすべてを適切に設定します。しかしながら、再起動は必須ではありません。 root 権限で以下のコマンドを実行すれば、すべてが起動するでしょう。

NFS サーバでは

```
# portmap
# nfsd -u -t -n 4
# mountd -r
```

NFS クライアントでは

```
# nfsiod -n 4
```

これでリモートのファイルシステムを実際にマウントする準備がすべてできました。この例では、サーバの名前は server で、クライアントの名前は client とします。リモートファイルシステムを一時的にマウントするだけ、もしくは設定をテストするだけなら、クライアント上で root 権限で以下のコマンドを実行するだけです。

```
# mount server:/home /mnt
```

これで、サーバの /home ディレクトリが、クライアントの /mnt にマウントされます。もしすべてが正しく設定されていれば、クライアントの /mnt に入り、サーバにあるファイルすべてを見れるはずです。

リモートファイルシステムを起動のたびに自動的にマウントしたいなら、 ファイルシステムを /etc/fstab ファイルに追加してください。例としてはこのようになります。

```
server:/home /mnt nfs rw 0 0
```

fstab(5) マニュアルページに利用可能なオプションがすべて掲載されています。

22.6.3. 実用的な使い方

NFS には実用的な使用法がいくつもあります。ここで典型的な使用法をいくつか紹介しましょう。

- 何台ものマシンで CDROM などのメディアを共有するように設定します。これは安上がりで、たいていは、 複数のマシンにソフトウェアをインストールするのにより便利な方法です。

- 大規模なネットワークでは、 すべてのユーザのホームディレクトリを格納するメイン NFS サーバを構築すると、ずっと便利でしょう。どのワークステーションにログインしても、ユーザがいつでも同じホームディレクトリを利用できるように、これらのホームディレクトリはネットワークに向けてエクスポートされます。

- 何台ものマシンで /usr/ports/distfiles ディレクトリを共有できます。こうすると、何台ものマシン上に port をインストールする必要がある時に、それぞれのマシンでソースコードをダウンロードすることなく、直ちにソースにアクセスできます。

22.6.4. amd による自動マウント

##: Wylie Stilwell.
###: Chern Lee.

amd(8) (自動マウントデーモン) は、ファイルシステム内のファイルまたはディレクトリがアクセスされると、自動的にリモートファイルシステムをマウントします。また、一定の間アクセスされないファイルシステムは amd によって自動的にアンマウントされます。amd を使用することは、通常 /etc/fstab 内に記述する恒久的なマウントに対する、単純な代替案となります。

amd はそれ自身を NFS サーバとして /host および /net ディレクトリに結びつけることによって動作します。 このディレクトリ内のどこかでファイルがアクセスされると、 amd は対応するリモートマウントを調べて、自動的にそれをマウントします。/net が、エクスポートされたファイルシステムを IP アドレスで指定してマウントするのに利用される一方

で、/host は、エクスポートされたファイルシステムをリモートホスト名で指定してマウントするのに利用されます。

/host/foobar/usr 内のファイルにアクセスすると、amd はホスト foobar からエクスポートされた /usr をマウントします。

例22.1 amd によるエクスポートされたファイルシステムのマウント

showmount コマンドを用いて、リモートホストのマウントで利用できるものが見られます。たとえば、foobar と名付けられたホストのマウントを見るために次のように利用できます。

```
% showmount -e foobar
Exports list on foobar:
/usr                               10.10.10.0
/a                                 10.10.10.0
% cd /host/foobar/usr
```

例のように showmount はエクスポートとして /usr を表示します。/host/foobar/usr にディレクトリを変更すると、amd はホスト名 foobar を解決し、お望みのエクスポートをマウントしようと試みます。

amd は /etc/rc.conf 内に次の行を記述すれば、起動スクリプトによって起動されます。

```
amd_enable="YES"
```

さらに amd_flags オプションによって amd にフラグをカスタマイズして渡せます。デフォルトでは amd_flags は次のように設定されています。

```
amd_flags="-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.↺
map"
```

/etc/amd.map ファイルは、エクスポートがマウントされるデフォルトオプションを決定します。/etc/amd.conf ファイルは、amd のより高度な機能の一部を設定します。

詳細については amd(8) および amd.conf(5) マニュアルページを参照してください。

22.6.5. 他のシステムとの統合についての問題

##: John Lind.

訳: 渡辺 智雄 **<tomo@jp.FreeBSD.org >**. 6 September 1996.

ISA バス用のイーサネットアダプタの中には性能が悪いため、 ネットワーク、特に NFS で深刻な問題がおきるものがあります。これは FreeBSD に限ったことではありませんが FreeBSD でも起こり得ます。

この問題は (FreeBSD を使用した) PC がシリコングラフィックス社やサン・マイクロシステムズ社などの高性能なワークステーションにネットワーク接続されている場合に頻繁に起こります。NFS マウントはうまく動作するでしょう。また、いくつかの操作もうまく動作するかもしれませんが、他のシステムに対する要求や応答は続いていても、突然サーバがクライアントの要求に対して応答しなくなります。これは、クライアントが FreeBSD か上記のワークステーションであるときにクライアント側に起きる現象です。多くのシステムでは、いったんこの問題が現われると、行儀良くクライアントを終了する手段はありません。NFS がこの状態に陥ってしまうと正常に戻すことはできないため、多くの場合クライアントをリセットすることが唯一の解決法となります。

「正しい」 解決法は、より高性能のイーサネットアダプタを FreeBSD システムにインストールすることですが、 満足に動作させる簡単な方法があります。 FreeBSD システムが サーバ になるのなら、クライアントからのマウント時に `-w=1024` オプションをつけて下さい。FreeBSD システムが クライアント になるのなら、NFS ファイルシステムを `-r=1024` オプションつきでマウントして下さい。これらのオプションは自動的にマウントをおこなう場合には クライアントの `fstab` エントリの 4 番目のフィールドに指定してもよいですし、手動マウントの場合は mount コマンドの `-o` パラメータで指定してもよいでしょう。

NFS サーバとクライアントが別々のネットワーク上にあるような場合、 これと間違えやすい他の問題が起きることに注意して下さい。そのような場合は、ルータが必要な UDP 情報をきちんとルーティングしているかを確かめて下さい。していなければ、たとえあなたが何をしようと解決できないでしょう。

次の例では `fastws` は高性能ワークステーションのホスト (インタフェース) 名で、`freebox` は低性能のイーサネットアダプタを備えた FreeBSD システムのホスト (インタフェース) 名です。また `/sharedfs` はエクスポートされる NFS ファイルシステムであり (exports(5) を参照) 、`/project` はエクスポートされたファイルシステムの、クライアント上のマウントポイントとなります。すべての場合において、アプリケーションによっては `hard` や `soft`, `bg` といった追加オプションがふさわしいかもしれないことに注意して下さい。

クライアント側 FreeBSD システム (`freebox`) の `/etc/fstab` の例は以下のとおりです。

```
fastws:/sharedfs /project nfs rw,-r=1024 0 0
```

`freebox` 上で手動で mount コマンドを実行する場合は次のようにして下さい。

```
# mount -t nfs -o -r=1024 fastws:/sharedfs /project
```

サーバ側 FreeBSD システム (fastws) の /etc/fstab の例は以下のとおりです。

```
freebox:/sharedfs /project nfs rw,-w=1024 0 0
```

fastws 上で手動で mount コマンドで実行する場合は次のようにして下さい。

```
# mount -t nfs -o -w=1024 freebox:/sharedfs /project
```

近いうちにどのような 16 ビットのイーサネットアダプタでも、上記の読み出し、書き込みサイズの制限なしで操作できるようになるでしょう。

失敗が発生したとき何が起きているか関心のある人に、 なぜ回復不可能なのかも含めて説明します。NFS は通常 (より小さいサイズへ分割されるかもしれませんが) 8 K の「ブロック」サイズで動作します。イーサネットのパケットサイズは最大 1500 バイト程度なので、上位階層のコードにとっては 1 つのユニットであって、NFS「ブロック」は複数のイーサネットパケットに分割されるものの、上位階層のコードにとっては 1 つのユニットであって、ユニットとして受信され、組み立て直され、肯定応答 (ACK) されなければなりません。高性能のワークステーションは次々に NFS ユニットを構成するパケットを、標準の許す限り間隔を詰めて次々に送り出すことができます。 小さく、容量の低いカードでは、 同じユニットの前のパケットがホストに転送される前に、 後のパケットがそれを踏みつぶしてしまいます。このため全体としてのユニットは、再構成も肯定応答もできません。その結果、ワークステーションはタイムアウトして再送を試みますが、8 K のユニット全体を再送しようとするので、このプロセスは際限無く繰り返されてしまいます。

ユニットサイズをイーサネットのパケットサイズの 制限以下に抑えることにより、受信した完全なイーサネットパケットについて個々に肯定応答を返せることが保証されるので、デッドロック状態を避けられるようになります。

それでも、高性能なワークステーションが力任せに次々と PC システムにデータを送ったときには踏みつぶしが起きるかもしれません。 しかし、高性能のカードを使っていれば、NFS「ユニット」で必ずそのような踏みつぶしが起きるとは限りません。踏みつぶしが起きたら、影響を受けたユニットは再送されて、 受信され、組み立てられ、肯定応答される十分な見込みがあります。

22.7. ディスクレス稼働

##: Jean-François Dockès.

訳: 鈴木 康修 **<yasu@hike.te.chiba-u.ac.jp >**

FreeBSD マシンはネットワークを通じて起動でき、 そして NFS サーバからマウントしたファイルシステムを使用して、ローカルディスクなしで動作することができます。標準の

設定ファイルを変更する以上の、システムの修正は必要ありません。 必要な要素のすべてが用意されているので、このようなシステムを設定するのは簡単です。

- ネットワークを通じてカーネルを読み込む方法は、少なくとも二つあります。

  - PXE: Intel® の Preboot Execution Environment システムは、一部のネットワークカードまたはマザーボードに組み込まれた、スマートなブート ROM の一形態です。詳細については pxeboot(8) を参照してください。

  - port の etherboot (net/etherboot) は、ネットワークを通じてカーネルを起動する ROM 化可能なコードを提供します。コードはネットワークカード上のブート PROM に焼き付けるか、あるいはローカルフロッピー (ハード) ディスクドライブ、または動作している MS-DOS® システムから読み込むことができます。 多くのネットワークカードに対応しています。

- サンプルスクリプト (`/usr/share/examples/diskless/clone_root`) はサーバ上で、 ワークステーションのルートファイルシステムの作成と維持をやり易くします。このスクリプトは少し書き換えないといけないでしょうが、早く取り掛かれるようにします。

- ディスクレスシステム起動を検知しサポートする標準のシステム起動ファイルが `/etc` 内にあります。

- 必要なら、NFS ファイルまたはローカルディスクのどちらかにスワップできます。

ディスクレスワークステーションを設定する方法はいろいろあります。 多くの要素が関わっており、その多くはローカルの状況に合わせてカスタマイズできます。下記は、単純さと標準の FreeBSD 起動スクリプトとの互換性を強調した完全なシステムの設定を説明します。記述されているシステムの特徴は次のとおりです。

- ディスクレスワークステーションは、共有された読み取り専用の **###**ファイルシステムと、共有された読み取り専用の `/usr` を使用します。

  ### ファイルシステムは、 標準的な FreeBSD (典型的にはサーバの) のルートのコピーで、一部の設定ファイルが、ディスクレス稼働、また場合によってはそのワークステーションに特有のもので上書きされています。

  書き込み可能でなければならない **###** の部分は mfs(8) ファイルシステムで覆われます。システムが再起動するときにはすべての変更が失われるでしょう。

- カーネルは DHCP (または BOOTP) および TFTP を用いて etherboot によって読み込まれます。

注意

記述されているとおり、このシステムは安全ではありません。ネットワークの保護された範囲で使用されるべきであり、他のホストから信頼されてはいけません。

22.7.1. セットアップの手順

22.7.1.1. DHCP/BOOTP の設定

ネットワークを通じて設定を取得し、 ワークステーションを起動するために一般的に使用されるプロトコルには、BOOTP と DHCP の 2 つがあります。それらはワークステーションのブートストラップ時に何ヵ所かで使用されます。

- etherboot はカーネルを見つけるために DHCP (デフォルト) または BOOTP (設定オプションが必要) を使用します (PXE は DHCP を使用します) 。
- NFS ルートの場所を定めるためにカーネルは BOOTP を使用します。

BOOTP だけを使用するようにシステムを設定することもできます。bootpd(8) サーバプログラムは FreeBSD のベースシステムに含まれています。

しかしながら、DHCP には BOOTP に勝る点が多々あります。 (よりよい設定ファイル、PXE が使えること、そしてディスクレス稼働には直接関係しない多くの長所) ここでは BOOTP だけ利用する場合と、BOOTP と DHCP を組み合わせた設定を扱います。特に ISC DHCP ソフトウェアパッケージを利用する後者の方法に重点をおきます。

22.7.1.1.1. ISC DHCP を使用する設定

isc-dhcp サーバは、BOOTP および DHCP リクエストの両方に答えることができます。

4.4-RELEASE の時点で isc-dhcp 3.0 はベースシステムの一部では無くなりました。まずはじめに net/isc-dhcp3-server port または対応する package をインストールする必要があるでしょう。ports および package に関する一般的な情報については 5章アプリケーションのインストール - packages と ports を参照してください。

isc-dhcp がインストールされると、動作するために設定ファイルを必要とします (通常 `/usr/local/etc/dhcpd.conf` が指定されます) 。下記にコメントを含めた例を示します。

```
        default-lease-time 600;
        max-lease-time 7200;
        authoritative;
```

```
option domain-name "example.com";
option domain-name-servers 192.168.4.1;
option routers 192.168.4.1;

subnet 192.168.4.0 netmask 255.255.255.0 {
  use-host-decl-names on; ❶
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.4.255;

  host margaux {
    hardware ethernet 01:23:45:67:89:ab;
    fixed-address margaux.example.com;
    next-server 192.168.4.4;❷
    filename "/tftpboot/kernel.diskless";❸
    option root-path "192.168.4.4:/data/misc/diskless";❹
  }
}
```

❶ このオプションは `host` 宣言の値を、ディスクレスホストへのホスト名として送るように `dhcpd` に指示します。別の方法として、ホスト宣言内に `option host-name margaux` を加えるものがあります。

❷ TFTP サーバを `next-server` ディレクティブに指定します (デフォルトは DHCP サーバと同じホストを使います)。

❸ カーネルとして etherboot が読み込むファイルを `filename` ディレクティブに指定します。

注記

PXE は相対的なファイル名を好むようです。また、カーネルではなく `pxeboot` を読み込みます (`option filename "pxeboot"`)。

❹ ルートファイルシステムへのパスを、通常の NFS 書式で `root-path` オプションに指定します。

22.7.1.1.2. BOOTP を使用する設定

続けて、`bootpd` で同等のことをする設定です。これは `/etc/bootptab` におきます。

BOOTP を使用するために、デフォルトではない `NO_DHCP_SUPPORT` オプション付きで etherboot をコンパイルしなければならないことと、PXE は DHCP を 必要 とすることに注意してください。bootpd の唯一明白な利点は、これがベースシステムに存在するということです。

```
.def100:\
  :hn:ht=1:sa=192.168.4.4:vm=rfc1048:\
  :sm=255.255.255.0:\
  :ds=192.168.4.1:\
  :gw=192.168.4.1:\
  :hd="/tftpboot":\
  :bf="/kernel.diskless":\
  :rp="192.168.4.4:/data/misc/diskless":

margaux:ha=0123456789ab:tc=.def100
```

22.7.1.2. Etherboot を用いるブートプログラムの準備

Etherboot のウェブサイト には主に Linux システムについて述べた 広範囲の文書 が含まれています。しかし、それにもかかわらず有用な情報を含んでいます。下記は FreeBSD システム上での etherboot の使用法についての概観を示します。

まずはじめに net/etherboot の package または port をインストールしなければなりません。etherboot port は通常 `/usr/ports/net/etherboot` にあります。ports ツリーがシステムにインストールされている場合、このディレクトリ内で `make` を実行すれば、よきに計らってくれます。ports および packages に関する情報は 5章アプリケーションのインストール - packages と ports を参照してください。

ここで説明している方法では、ブートフロッピーを使用します。他の方法 (PROM または DOS プログラム) については etherboot の文書を参照してください。

ブートフロッピーを作成するためには、etherboot をインストールしたマシンのドライブにフロッピーディスクを挿入します。それからカレントディレクトリを etherboot ツリー内の `src` ディレクトリにして次のように入力します。

```
# gmake bin32/devicetype.fd0
```

`devicetype` は ディスクレスワークステーションのイーサネットカードタイプに依存します。正しい *`devicetype`* を決定するために、同じディレクトリ内の `NIC` ファイルを参照してください。

22.7.1.3. TFTP および NFS サーバの設定

TFTP サーバ上で `tftpd` を有効にする必要があります。

1. `tftpd` が提供するファイルを置くディレクトリ (たとえば `/tftpboot`) を作成してください。

2. `/etc/inetd.conf` ファイルに以下の行を追加してください。

```
tftp    dgram   udp     wait    root  /usr/libexec/tftpd    tftpd
 -s /tftpboot
```

注記

少なくとも PXE のいくつかのバージョンが TCP 版の TFTP を要求するようです。その場合 `dgram udp` を `stream tcp` に置き換えた 2 番目の行を追加してください。

3. `inetd` に設定ファイルを再読み込みさせてください。

```
# kill -HUP `cat /var/run/inetd.pid`
```

`tftpboot` ディレクトリはサーバ上のどこにでも置けます。その場所が `inetd.conf` および `dhcpd.conf` の両方に設定されていることを確かめてください。

さらに NFS を有効にして NFS サーバの適切なファイルシステムをエクスポートする必要があります。

1. この行を `/etc/rc.conf` に追加してください。

```
nfs_server_enable="YES"
```

2. 下記を `/etc/exports` に加えることで、ディスクレスマシンのルートディレクトリが位置するファイルシステムをエクスポートしてください (ボリュームのマウントポイントを適当に調節し、*`margaux`* をディスクレスワークステーションの名前に置き換えてください)。

```
/data/misc -alldirs -ro margaux
```

3. `mountd` に設定ファイルを再読み込みさせてください。 `/etc/rc.conf` 内で NFS をはじめて有効にする必要があったのなら、代わりに再起動した方がよいかもしれません。

```
# kill -HUP `cat /var/run/mountd.pid`
```

22.7.1.4. ディスクレス用のカーネル構築

次のオプションを (通常のものに) 追加した、ディスクレスクライアント用のカーネルコンフィグレーションファイルを作成してください。

```
```

```
          options     BOOTP          # Use BOOTP to obtain IP ↺
address/hostname
          options     BOOTP_NFSROOT  # NFS mount root filesystem ↺
using BOOTP info
          options     BOOTP_COMPAT   # Workaround for broken bootp ↺
daemons.
```

BOOTP_NFSV3 および BOOTP_WIRED_TO を利用してもよいかもしれません (LINT を参照してください)。

カーネルを構築して (9章FreeBSD カーネルのコンフィグレーション を参照)、dhcpd.conf に記述した名称で tftp ディレクトリにコピーしてください。

22.7.1.5. ルートファイルシステムの準備

dhcpd.conf に root-path として記載された ディスクレスワークステーションのためのルートファイルシステムを作成する必要があります。

これを行う最も簡単な方法は /usr/share/examples/diskless/clone_root シェルスクリプトを使用することです。このスクリプトは、少なくともファイルシステムが作成される場所 (DEST 変数) を調節するために変更する必要があります。

説明についてはスクリプトの一番上にあるコメントを参照してください。ベースシステムをどのように構築するか、 またファイルがどのようにディスクレス稼働、サブネット、 または個々のワークステーションに固有のバージョンによって、 選択的にオーバライドできるかを説明します。 また、ディスクレスな場合の /etc/fstab ファイルおよび /etc/rc.conf ファイルの例を示します。

/usr/share/examples/diskless 内の README ファイルには、多くの興味深い背景情報が書かれています。しかし diskless ディレクトリ内の他の例と同じく、clone_root と /etc/rc.diskless[12] で実際に使われているものとは異なる設定方法が説明されています。ここに書かれている方法は rc スクリプトの変更が必要になりますが、こちらの方が気に入ったというのでなければ、参照にとどめてください。

22.7.1.6. スワップの設定

必要なら、サーバに置かれたスワップファイルに NFS 経由でアクセスできます。bootptab または dhcpd.conf の正確なオプションは、現時点では明確には文書化されていません。下記の設定例は isc-dhcp 3.0rc11 を使用して動作したと報告されているものです。

1. dhcpd.conf に下記の行を追加してください。

```
# Global section
option swap-path code 128 = string;
option swap-size code 129 = integer 32;
```

```
            host margaux {
              ... # Standard lines, see above
              option swap-path "192.168.4.4:/
netswapvolume/netswap" ;
              option swap-size 64000;
            }
```

これは、少なくとも FreeBSD クライアントにおいては、DHCP/BOOTP オプションコードの 128 は NFS スワップファイルへのパスで、オプションコード 129 は KB 単位のスワップサイズだということです。もっと古いバージョンの `dhcpd` では `option option-128 "...` という書式が受け付けられましたが、もはや対応していません。

代わりに、`/etc/bootptab` では次の書式を使います。

```
T128="192.168.4.4:/netswapvolume/
netswap":T129=0000fa00
```

注記

`/etc/bootptab` では、スワップの大きさは 16 進数で表さなければなりません。

2. NFS スワップファイルサーバ側でスワップファイルを作成します。

```
            # mkdir /netswapvolume/netswap
            # cd /netswapvolume/netswap
            # dd if=/dev/zero bs=1024 count=64000 ↺
of=swap.192.168.4.6
            # chmod 0600 swap.192.168.4.6
```

192.168.4.6 はディスクレスクライアントの IP アドレスです。

3. NFS スワップファイルサーバ上で `/etc/exports` に下記の行を追加してください。

```
      /netswapvolume   -maproot=0:10 -alldirs margaux
```

それから、上述したように mountd にエクスポートファイルを再読み込みさせてください。

22.7.1.7. 雑多な問題

22.7.1.7.1. 読み取り専用の /usr で動作させる

ディスクレスワークステーションが X を起動するように設定されている場合、xdm 設定ファイルを調整しなければならないでしょう。これはデフォルトでエラーファイルを /usr に置きます。

22.7.1.7.2. FreeBSD ではないサーバを使用する

ルートファイルシステムを提供するサーバが FreeBSD で動作していない場合、FreeBSD マシン上でルートファイルシステムを作成し、tar または cpio を利用して置きたい場所にコピーしなければならないでしょう。

この状況では、major/minor 整数サイズが異なっていることにより /dev 内のスペシャルファイルに関する問題が時々おこります。この問題を解決するには、非 FreeBSD サーバからディレクトリをエクスポートして、そのディレクトリを FreeBSD マシンでマウントし、FreeBSD マシン上で MAKEDEV を実行して正しいデバイスエントリを作成します (FreeBSD 5.0 およびそれ以降では、devfs(5) を使用してユーザに意識させずにデバイスノードを割り当てるので、これらのバージョンでは MAKEDEV は必要ありません)。

22.8. ISDN

訳: はらだ きろう <kiroh@jp.FreeBSD.org >. 11 December 1996.

ISDN 技術とハードウェアに関しては、Dan Kegel's ISDN Page がよい参考になるでしょう。

手軽な ISDN の導入手順は以下のようになります。

- ヨーロッパ在住の方は ISDN カードの節に進んでください。
- ダイヤルアップ専用でない回線上で、インターネットプロバイダをつかってインターネットに接続するために ISDN を使用することを第一に考えている場合は、ターミナルアダプタの使用を考えてみてください。この方法はもっとも柔軟性があり、プロバイダを変更した場合の問題も少ないでしょう。
- 2 つの LAN を接続する場合や、ISDN 専用線を使用する場合には、スタンドアロンなルータまたはブリッジの使用を勧めます。

費用はどの解決法を選ぶかを決める重要な要因です。以下に、最も安価な方法から、高価な方法まで順に説明していきます。

22.8.1. ISDN カード

##: Hellmuth Michaelis.

FreeBSD の ISDN 実装は、パッシブカードを使用した DSS1/Q.931 (または Euro-ISDN) 標準だけに対応しています。FreeBSD 4.4 からは、ファームウェアが他の信号プロトコルにも対応している 一部のアクティブカードにも対応しました。その中には、はじめて対応された一次群速度インタフェース (PRI) ISDN カードもあります。

isdn4bsd は IP over raw HDLC または同期 PPP を利用して他の ISDN ルータに接続できるようにします。PPP では、カーネル PPP を sppp(4) ドライバを修正した `isppp` ドライバとともに利用するか、または ユーザプロセス ppp(8) を利用するかのどちらかになります。ユーザ ppp(8) を利用すると、二つ以上の ISDN B チャネルを併せて利用できます。ソフトウェア 300 ボーモデムのような多くのユーティリティとともに、留守番電話アプリケーションも利用可能です。

FreeBSD が対応している PC ISDN カードの数は増加しており、ヨーロッパ全域や世界のその他多くの地域でうまく使えることが報告されています。

対応しているパッシブ ISDN カードのほとんどは Infineon (前身は Siemens) の ISAC/HSCX/IPAC ISDN チップセットを備えたカードですが、Cologne Chip から供給されたチップを備えた ISDN カード (ISA バスのみ)、Winbond W6692 チップを備えた PCI カード、Tiger300/320/ISAC チップセットを組み合わたカードの一部、および AVM Fritz!Card PCI V.1.0 や AVM Fritz!Card PnP のようなベンダ独自のチップセットに基づいたカードもあります。

現在のところ、対応しているアクティブカードは AVM B1 (ISA および PCI) BRI カードと AVM T1 PCI PRI カードです。

isdn4bsd についての文書は FreeBSD システム内の `/usr/share/examples/isdn/` ディレクトリまたは isdn4bsd のウェブサイトを参照してください。そこにはヒントや正誤表や isdn4bsd ハンドブックのような、さらに多くの文書に対するポインタがあります。

異なる ISDN プロトコルや、現在対応されていない ISDN PC カードに対応することや、その他 isdn4bsd を拡張することに興味があるなら、Hellmuth Michaelis に連絡してください。

isdn4bsd のインストール、設定、 そしてトラブルシューティングに関して質問があれば freebsd-isdn メーリングリストが利用可能です。

22.8.2. ISDN ターミナルアダプタ

ターミナルアダプタ (TA) は ISDN で、通常の電話線におけるモデムに相当するものです。

ほとんどの TA は、標準のヘイズ AT コマンドセットを使用しているので、単にモデムと置き換えて使うことができます。

TA は、基本的にはモデムと同じように動作しますが、接続方法は異なり、通信速度も古いモデムよりはるかに速くなります。PPP の設定を、モデムの場合と同じように行ってください。特にシリアル速度を使用できる最高速度に設定するのを忘れないでください。

プロバイダへの接続に TA を使用する最大のメリットは、動的 PPP を行えることです。最近 IP アドレス空間がますます不足してきているため、ほとんどのプロバイダは、固定 IP アドレスを割り当てないようになっています。ほとんどのスタンドアローンルータは、動的 IP アドレス割り当てに対応していません。

訳注

最近の ISDN ルータでは IP アドレスの動的割り当てに対応しているものも多いようです。ただし制限がある場合もありますので、詳しくはメーカに問い合わせてください。

TA を使用した場合の機能や接続の安定性は、使用している PPP デーモンに完全に依存します。そのため、FreeBSD で PPP の設定が完了していれば、使用している既存のモデムを ISDN の TA に簡単にアップグレードすることができます。ただし、それまでの PPP のプログラムに問題があった場合、その問題は TA に置き換えてもそのまま残ります。

最高の安定性を求めるのであれば、ユーザランド PPP ではなく、カーネル PPPを使用してください。

以下の TA は、FreeBSD で動作確認ずみです。

- Motorola BitSurfer および Bitsurfer Pro
- Adtran

他の TA もほとんどの場合うまく動作するでしょう。TA のメーカーでは、TA がほとんどの標準モデム AT コマンドセットを受け付けるようにするよう努力しているようです。

外部 TA を使う際の最大の問題点は、モデムの場合と同じく良いシリアルカードが必要であるということです。

シリアルデバイスの詳細と、非同期シリアルポートと同期シリアルポートの差を理解するには、FreeBSD シリアルハードウェアチュートリアルを参照してください。

標準の PC シリアルポート (非同期) に接続された TA は 128 Kbs の接続を行っていても、最大通信速度が 115.2 Kbs に制限されてしまいます。128 Kbs の ISDN の性能を最大限に生かすためには TA を同期シリアルカードに接続しなければなりません。

内蔵 TA を購入すれば、同期/非同期問題を回避できるとは思わないでください。内蔵 TA には、単に標準 PC シリアルポートのチップが内蔵されているだけです。内蔵 TA の

利点といえば、シリアルケーブルを買わなくていいということと、電源コンセントが一つ少なくて済むということくらいでしょう。

同期カードと TA の組合せでも、スタンドアロンのルータと同程度の速度は確保できます。さらに、386 の FreeBSD マシンと組合せると、より柔軟な設定が可能です。

同期カード/TA を選ぶか、スタンドアロンルータを選ぶかは、多分に宗教的な問題です。メーリングリストでもいくつか議論がありました。議論の全容については、アーカイブ を検索してください。

22.8.3. スタンドアロン ISDN ブリッジ/ルータ

ISDN ブリッジあるいはルータは、FreeBSD あるいは他の OS に特有のものでは皆目ありません。ルーティングやブリッジング技術に関する詳細は、ネットワークの参考書をご覧ください。

この節では、ルータとブリッジのどちらでもあてはまるように記述します。

ローエンド ISDN ルータ/ブリッジ製品は、価格が下がってきていることもあり、より広く選択されるようになるでしょう。ISDN ルータは、ローカルイーサネットネットワークに直接接続し、自身で他のブリッジ/ルータとの接続を制御する小さな箱です。PPP や他の広く使用されているプロトコルをつかって通信するためのソフトウェアが組み込まれています。

ルータは、完全な同期 ISDN 接続を使用するため、通常の TA と比較してスループットが大幅に向上します。

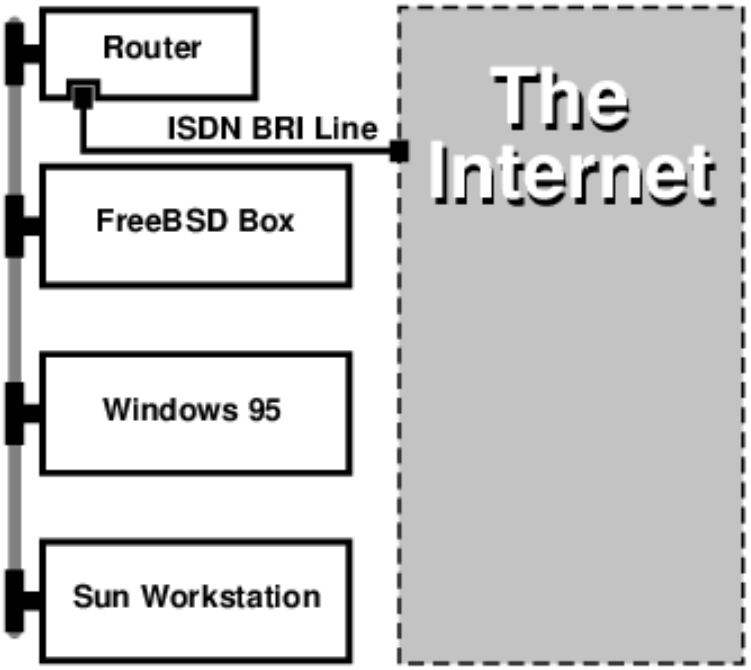
ISDN ルータ/ブリッジを使用する場合の最大の問題点は、各メーカーの製品間に相性の問題がまだ存在することです。インターネットプロバイダとの接続を考えている場合には、プロバイダと相談することをお勧めします。

事務所の LAN と家庭の LAN の間など、二つの LAN セグメントの間を接続しようとしている場合は、これはもっともメンテナンスが簡単で、安くあがる解決方法です。接続の両側の機材を購入するので、リンクがうまくいくであろうことを保証できます。

たとえば、家庭のコンピュータや支店のネットワークを本社のネットワークに接続するためには、以下のような設定が使用できます。

例22.2 支店または家庭のネットワーク

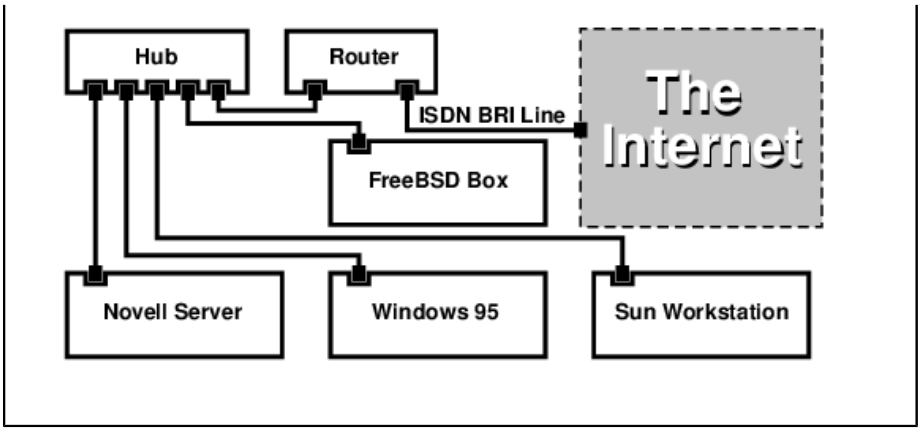
ネットワークは 10 Base 2 イーサネット (「thinnet」) のバス型トポロジを用いています。ルータとネットワークの間は、必要に応じて AUI/10BT トランシーバを使って接続してください。



家庭/支店で一台しかコンピュータを使用しないのであれば、クロスのツイストペアケーブルを使用して、直接スタンドアロンルータに接続することも可能です。

例22.3 本社 LAN や他の LAN

ネットワークは 10 base T イーサネット (「Twisted Pair」) のスター型トポロジを用いています。



ほとんどのルータ/ブリッジの大きな利点は、 別々の二つのサイトに対して、同時 にそれぞれ独立した二つの PPP 接続が可能であることです。 これは、シリアルポートを 2 つもった特定の (通常は高価な) モデルを除いて、通常の TA では対応していません。チャネルボンディングや MPP などと混同しないでください。

たとえば、事務所で専用線 ISDN 接続を使用していて、別の ISDN 回線を購入したくないときには大変便利な機能です。この場合、 事務所のルータは、インターネットに接続するための一つの専用線 B チャネル接続 (64 Kbs) を管理し、別の B チャネルを他のデータ接続に使用できます。 2 つ目の B チャネルは他の場所とのダイアルイン、ダイアルアウトに使用したり、バンド幅を増やすために、 1 つ目の B チャネルと動的に結合すること (MPPなど) ができます。

またイーサネットブリッジは、IP パケット以外も中継できます。 IPX/SPX など、使用するすべてのプロトコルを送ることが可能です。

22.9. NIS/YP

##: Bill Swingle.
##: Eric Ogren # Udo Erdelhoff.

22.9.1. NIS/YP とは?

NIS とは Network Information Services の略で Sun Microsystems によって UNIX® の (もともとは SunOS™ の) 集中管理のために開発されました。現在では事実上の業界標準になっており、 主要な UNIX® ライクシステム (Solaris™, HP-UX, AIX®, Linux, NetBSD, OpenBSD, FreeBSD、等々) はすべてこれをサポートしています。

NIS は元々、イエローページといっていましたが、商標問題から Sun はその名前を変えました。古い用語 (および yp) はまだよく見られ、使用されています。

NIS は RPC を使ったクライアント/サーバシステムです。これを使うと NIS ドメイン内のマシン間で、共通の設定ファイルを共有することができます。また NIS を使うことでシステム管理者は最小限の設定データで NIS クライアントを立ち上げることができ、1 ヶ所から設定データの追加、削除、変更が可能です。

NIS は Windows NT® のドメインシステムに似ています。内部の実装は似ても似つかないものですが、基本的な機能を対比することはできます。

22.9.2. 知っておくべき用語 / プロセス

NIS サーバの立ち上げや NIS クライアントの設定など、NIS を FreeBSD に導入するにあたって、目にするであろう用語や重要なユーザプロセスがいくつかあります。

用語	説明
NIS ドメイン名	NIS マスタサーバとそのクライアントすべて (スレーブサーバを含む) には NIS ドメイン名がついています。Windows NT® ドメイン名と同様に、NIS ドメイン名は DNS とは何の関係もありません。
portmap	RPC (Remote Procedure Call, NIS で使用されるネットワークプロトコル) を利用するために実行しておかなければなりません。 `portmap` が動作していなければ、NIS サーバを起動することも、NIS クライアントとして動作させることもできません。
ypbind	NIS クライアントを NIS サーバに「結びつけ」ます。これは NIS ドメイン名をシステムから取得し RPC を用いてサーバに接続します。`ypbind` は NIS 環境におけるクライアントとサーバ間の通信の中枢です。クライアントマシンの `ypbind` が停止した場合は、NIS サーバへアクセスすることができなくなります。
ypserv	は NIS サーバでのみ実行されるべきもので、NIS サーバプロセスそのものです。ypserv(8) が停止した場合、サーバはもはや NIS リクエストに応答することができなくなるでしょう (できれば、後を引き継ぐスレーブサーバがあるとよいでしょう)。今まで使っていたサーバが機能を停止したと

用語	説明
	き、 別のサーバに再接続しに行かない NIS の実装もいくつかあります (FreeBSD のものは違います)。 そのような場合に復帰するための唯一の方法は、 サーバプロセス (あるいはサーバ全体)、もしくはクライアントの `ypbind` プロセスを再スタートすることです。
rpc.yppasswdd	NIS マスターサーバで動かすべき、もう一つのプロセスです。これは NIS クライアントが NIS パスワードを変更することを可能にするデーモンです。 このデーモンが動作していないときは、 ユーザは NIS マスタサーバにログインし、そこでパスワードを変更しなければなりません。

22.9.3. 動作のしくみ

NIS 環境にあるホストは、マスターサーバ、スレーブサーバ、クライアントの 3 種類に分類されます。サーバは、ホストの設定情報の中心的な情報格納庫の役割をします。マスターサーバは元となる信頼できる情報を保持し、 スレーブサーバは冗長性を確保するためこの情報をミラーします。そしてクライアントは、サーバから情報の提供を受けて動作します。

この方法を用いることで、数多くのファイルにある情報が共有できます。よく NIS で共有されるのは、`master.passwd` や `group`, `hosts` といったファイルです。クライアント上のプロセスが、通常ならローカルのファイルにある情報を必要とするときは、クライアントは代わりに接続している NIS サーバに問い合わせを行います。

22.9.3.1. マシンの分類

- NIS マスターサーバ。このサーバは Windows NT® で言うところのプライマリドメインコントローラにあたります。すべての NIS クライアントで利用されるファイルを保守します。`passwd` や `group`、その他 NIS クライアントが参照するファイルは、マスターサーバにあります。

注記

一つのマシンが一つ以上の NIS ドメインのマスターサーバになることは可能です。しかし、ここでは比較的小規模の NIS 環境を対象としているため、そのような場合については扱いません。

- NIS スレーブサーバ。Windows NT® のバックアップドメインコントローラに似たもので、NIS スレーブサーバは NIS マスターサーバのデータファイルのコピーを保持します。NIS スレーブサーバは重要な環境で必要とされる冗長性を提供し、マスターサーバの負荷のバランスをとります。NIS クライアントは常に最初にレスポンスを返したサーバを NIS サーバとして接続しますが、これにはスレーブサーバも含まれます。
- NIS クライアント。NIS クライアントは大部分の Windows NT® ワークステーションのように、ログオンに際して NIS サーバ (Windows NT® ワークステーションの場合は Windows NT® ドメインコントローラ) に接続して認証します。

22.9.4. NIS/YP を使う

この節では NIS 環境の立ち上げ例を取り上げます。

注記

この節ではあなたが FreeBSD 3.3 以降を使っているものとします。ここで与えられる指示は おそらく FreeBSD の 3.0 以降のどのバージョンでも機能するでしょうが、 それを保証するものではありません。

22.9.4.1. 計画を立てる

あなたが大学の小さな研究室の管理人であるとしましょう。 この研究室は 15 台の FreeBSD マシンからなっていて、現在はまだ集中管理されていません。すなわち、各マシンは `/etc/passwd` と `/etc/master.passwd` を各々が持っています。これらのファイルは手動でお互いに同期させています。つまり現時点では、新しいユーザをあなたが追加するとき、`adduser` を 15 ヶ所すべてで実行しなければなりません。これは明らかに変える必要があるため、あなたはこのうち 2 台をサーバにして NIS を導入することを決めました。

その結果、研究室の設定はこのようなものになります。

マシンの名前	IP アドレス	役割
`ellington`	`10.0.0.2`	NIS マスタ
`coltrane`	`10.0.0.3`	NIS スレーブ
`basie`	`10.0.0.4`	教員用のワークステーション
`bird`	`10.0.0.5`	クライアントマシン
`cli[1-11]`	`10.0.0.[6-17]`	その他のクライアントマシン

もし NIS によるシステム管理の設定を行なうのが初めてなら、どのようにしたいのか、ひととおり最後まで考えてみることをお勧めします。 ネットワークの規模によらず、 いくつか決めるべきことがあるからです。

22.9.4.1.1. NIS ドメイン名を決める

ここでいうドメイン名は、今まであなたが使っていた、いわゆる「ドメイン名」と呼んでいたものとは違います。正確には「NIS ドメイン名」と呼ばれます。クライアントがサーバに情報を要求するとき、 その要求には自分が属する NIS ドメインの名前が含まれています。これは 1 つのネットワークに複数のサーバがある場合に、どのサーバが要求を処理すれば良いかを決めるために使われます。NIS ドメイン名とは、関連のあるホストをグループ化するための名前である、と考えると良いでしょう。

組織によってはインターネットのドメイン名を NIS ドメイン名に使っているところがあります。 これはネットワークのトラブルをデバッグするときに混乱の原因となるため、 お勧めできません。 NIS ドメイン名はネットワーク内で一意なければいけません。そして、 ドメイン名がドメインに含まれるマシンを表すようなものであれば分かり易いです。 たとえば Acme 社のアート (Art) 部門であれば NIS ドメイン名を「acme-art」とすれば良いでしょう。この例では NIS ドメイン名として test-domain を使用します。

しかしながらオペレーティングシステムによっては (特に SunOS™)、NIS ドメイン名をネットワークドメイン名として使うものもあります。 あなたのネットワークにそのような制限のあるマシンが 1 台でもあるときは、NIS のドメイン名としてインターネットのネットワークドメイン名を使わなければ いけません。

22.9.4.1.2. サーバマシンの物理的必要条件

NIS サーバとして使うマシンを選ぶ際には、いくつか注意すべき点があります。NIS に関する困ったことの一つに、クライアントのサーバへの依存度があります。クライアントが自分の NIS ドメインのサーバに接続できないと、マシンが使用不能になることがあまりに多いのです。もし、ユーザやグループに関する情報が得られなければ、ほとんどのシステムは一時的に停止してしまいます。こういったことを念頭に置いて、頻繁にリブートされるマシンや、開発に使われそうなマシンを選ばないようにしなければなりません。理想的には NIS サーバはスタンドアロンで NIS サーバ専用のマシンにするべきです。 ネットワークの負荷が重くなければ、他のサービスを走らせているマシンを NIS サーバにしてもかまいません。ただし NIS サーバが使えなくなると、すべての クライアントに影響をおよぼす、という点には注意しなければなりません。

22.9.4.2. NIS サーバ

元となるすべての NIS 情報は、NIS マスターサーバと呼ばれる 1 台のマシンに格納されます。この情報が格納されるデータベースを NIS マップと呼びます。FreeBSD では、このマップは `/var/yp/[domainname]` に置かれます。`[domainname]` は、サーバがサービスする NIS ドメインです。1 台の NIS サーバが複数のドメインをサポートすることも可能です。つまり、このディレクトリを各々のドメインごとに作ることができます。それぞれのドメインは、独立したマップの集合を持つことになります。

NIS のマスターサーバとスレーブサーバ上では、 ypserv デーモンがすべての NIS 要求を処理します。 ypserv は NIS クライアントからの要求を受け付け、ドメイン名とマップ名を対応するデータベースファイルへのパスに変換し、データをクライアントに返送します。

22.9.4.2.1. NIS マスターサーバの設定

やりたいことにもよりますが NIS マスターサーバの設定は比較的単純です。FreeBSD は初期状態で NIS に対応しています。必要なのは以下の行を /etc/rc.conf に追加することだけで、あとは FreeBSD がやってくれます。

1. ```
   nisdomainname="test-domain"
   ```

   この行はネットワークの設定後に (たとえば再起動後に) NIS のドメイン名を test-domain に設定します。

2. ```
   nis_server_enable="YES"
   ```

   これは FreeBSD に次にネットワークが立ち上がったとき NIS のサーバプロセスを起動させます。

3. ```
   nis_yppasswdd_enable="YES"
   ```

   これは rpc.yppasswdd デーモンを有効にします。上述したようにこれはユーザが NIS のパスワードをクライアントのマシンから変更することを可能にします。

注記

NIS の設定によっては、さらに他のエントリを付け加える必要があるかもしれません。詳細については、下記の NIS クライアントとしても動作している NIS サーバ 節を参照してください。

さて、あとはスーパユーザ権限で /etc/netstart コマンドを実行するだけです。これにより /etc/rc.conf で定義された値を使ってすべての設定が行なわれます。

22.9.4.2.2. NIS マップの初期化

NIS マップ とは /var/yp ディレクトリにあるデータベースファイルです。 これらは NIS マスタの /etc ディレクトリの設定ファイルから作られます。唯一の例外は /etc/master.passwd ファイルです。これは root や他の管理用アカウントのパスワードまでその NIS ドメインのすべてのサーバに伝えたくないという、もっともな理由によるものです。このため NIS マップの初期化の前に以下を行う必要があります。

```
# cp /etc/master.passwd /var/yp/master.passwd
```
```

```
# cd /var/yp
# vi master.passwd
```

システムに関するアカウント (bin, tty, kmem, games など) や、NIS クライアントに伝えたくないアカウント (たとえば root や他の UID が 0 (スーパユーザ) のアカウント) をすべて NIS マップから取り除かなければなりません。

注記

/var/yp/master.passwd が グループまたは誰もが読めるようになっていないようにしてください (モード 600)! 必要なら chmod コマンドを使ってください。

すべてが終わったら NIS マップを初期化します! FreeBSD には、これを行うために ypinit という名のスクリプトが含まれています (詳細はそのマニュアルページをご覧ください)。このスクリプトはほとんどの UNIX® OS に存在しますが、すべてとは限らないことを覚えておいてください。Digital Unix/Compaq Tru64 UNIX では ypsetup と呼ばれています。NIS マスタのためのマップを作るためには -m オプションを ypinit に与えます。上述のステップを完了しているなら、以下を実行して NIS マップを生成します。

```
ellington# ypinit -m test-domain
Server Type: MASTER Domain: test-domain
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
At this point, we have to construct a list of this domains YP ↺
servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server   :  ellington
next host to add:  coltrane
next host to add:  ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct?  [y/n: y] y

[..output from map generation..-]

NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

`ypinit` は `/var/yp/Makefile` を `/var/yp/Makefile.dist` から作成します。作成された時点では、そのファイルはあなたが FreeBSD マシンだけからなるサーバが 1 台だけの NIS 環境を扱っていると仮定しています。test-domain はスレーブサーバを一つ持っていますので `/var/yp/Makefile` を編集しなければなりません。

```
ellington# vi /var/yp/Makefile
```

以下の行を (もし既にコメントアウトされていないならば) コメントアウトしなければなりません。

```
NOPUSH = "True"
```

22.9.4.2.3. NIS スレーブサーバの設定

NIS スレーブサーバの設定はマスターサーバの設定以上に簡単です。スレーブサーバにログオンし `/etc/rc.conf` ファイルを前回と同様に編集します。唯一の違うところは `ypinit` の実行に `-s` オプションを使わなければいけないことです。`-s` オプションは NIS マスターサーバの名前を要求し、コマンドラインは以下のようになります。

```
coltrane# ypinit -s ellington test-domain

Server Type: SLAVE Domain: test-domain Master: ellington

Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n]  n

Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the databases from ellington.
Transferring netgroup...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byuser...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byhost...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring group.bygid...
ypxfr: Exiting: Map successfully transferred
Transferring group.byname...
ypxfr: Exiting: Map successfully transferred
Transferring services.byname...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.bynumber...
```

```
ypxfr: Exiting: Map successfully transferred
Transferring rpc.byname...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.byname...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...
ypxfr: Exiting: Map successfully transferred

coltrane has been setup as an YP slave server without any errors.
Don't forget to update map ypservers on ellington.
```

この例の場合 `/var/yp/test-domain` というディレクトリが必要になります。 NISマスターサーバのマップファイルのコピーは、このディレクトリに置いてください。これらを確実に最新のものに維持する必要があります。次のエントリをスレーブサーバの `/etc/crontab` に追加することで、最新のものに保つことができます。

```
20      *       *       *       *       root    /usr/libexec/ypxfr ↺
passwd.byname
21      *       *       *       *       root    /usr/libexec/ypxfr ↺
passwd.byuid
```

この二行はスレーブサーバにあるマップファイルを、 マスターサーバのマップファイルと同期させるものです。 このエントリは必須というわけではありませんが、マスターサーバは NIS マップに対する変更をスレーブサーバに伝えようとしますし、 サーバが管理するシステムにとってパスワード情報はとても重要なので、 強制的に更新してしまうことはよい考えです。特に、マップファイルの更新がきちんと行なわれるかどうかわからないくらい混雑するネットワークでは、重要になります。

スレーブサーバ上でも `/etc/netstart` コマンドを実行して、NIS サーバを再起動してください。

22.9.4.3. NIS クライアント

NIS クライアントは `ypbind` デーモンを使って、特定の NIS サーバとの間に結合 (binding) と呼ばれる関係を成立させます。 `ypbind` はシステムのデフォルトのドメイン (`domainname` コマンドで設定されます) を確認し、RPC 要求をローカルネット

ワークにブロードキャストします。この RPC 要求により ypbind が結合を成立させようとしているドメイン名が指定されます。要求されているドメイン名に対してサービスするよう設定されたサーバが ブロードキャストを受信すると、サーバは ypbind に応答しypbind は応答のあったサーバのアドレスを記録します。複数のサーバ (たとえば一つのマスターサーバと、複数のスレーブサーバ) が利用可能な場合、ypbind は、最初に応答したサーバのアドレスを使用します。これ以降、クライアントのシステムは、すべての NIS の要求をそのサーバに向けて送信します。ypbind は、サーバが順調に動作していることを確認するため、時々「ping」をサーバに送ります。反応が戻ってくるべき時間内に ping に対する応答が来なければ、ypbind は、そのドメインを結合不能 (unbound) として記録し、別のサーバを見つけるべく、再びブロードキャストパケットの送信を行います。

22.9.4.3.1. NIS クライアントの設定

FreeBSD マシンを NIS クライアントにする設定は非常に単純です。

1. ネットワークの起動時に NIS ドメイン名を設定して ypbind を起動させるために /etc/rc.conf ファイルを編集して以下の行を追加します。

```
nisdomainname="test-domain"
nis_client_enable="YES"
```

2. NIS サーバから、利用可能なパスワードエントリをすべて取り込むため、/etc/master.passwd からすべてのユーザアカウントを取り除いて、vipw コマンドで以下の行を /etc/master.passwd の最後に追加します。

```
+:::::::::
```

注記

この行によって NIS サーバのパスワードマップにアカウントがある人全員にアカウントが与えられます。この行を変更すると、さまざまな NIS クライアントの設定を行なうことが可能です。詳細は ネットグループ を、さらに詳しい情報については、O'Reilly の Managing NFS and NIS を参照してください。

注記

/etc/master.passwd 内に少なくとも一つのローカルアカウント (つまり NIS 経由でインポートされていないアカウン

ト) を置くべきです。また、このアカウントは `wheel` グループのメンバーであるべきです。NIS がどこか調子悪いときには、リモートからこのアカウントでログインし、root になって修復するのに利用できます。

3. NIS サーバにあるすべてのグループエントリを取り込むため、以下の行を `/etc/group` に追加します。

```
+:*::
```

上記の手順がすべて完了すれば、`ypcat passwd` によって NIS サーバの passwd マップが参照できるようになっているはずです。

22.9.5. NIS セキュリティ

一般にドメイン名さえ知っていれば、どこにいるリモートユーザでも ypserv(8) に RPC を発行して NIS マップの内容を引き出すことができます。こういった不正なやりとりを防ぐため、ypserv(8) には securenets と呼ばれる機能があります。これは、アクセスを決められたホストだけに制限するのに使える機能です。ypserv(8) は起動時に `/var/yp/securenets` ファイルから securenets に関する情報を読み込みます。

注記

上記のパス名は `-p` オプションで指定されたパス名によって変わります。このファイルは、 空白で区切られたネットワーク指定とネットマスクのエントリからなっていて、「#」 で始まる行はコメントとみなされます。簡単な securenets ファイルの例を以下に示します。

```
# allow connections from local host -- mandatory
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
10.0.0.0      255.255.240.0
```

ypserv(8) が上記のルールの一つと合致するアドレスからの要求を受け取った場合、処理は正常に行なわれます。もしアドレスがルールに合致しなければ、その要求は無視されて警告メッセージがログに記録されます。また `/var/yp/securenets` が存在しない場合、`ypserv` はすべてのホストからの接続を受け入れます。

ypserv は Wietse Venema 氏による tcpwrapper パッケージもサポートしています。そのため /var/yp/securenets の代わりに tcpwrapper の設定ファイルを使ってアクセス制御を行なうことも可能です。

注記

これらのアクセス制御機能は一定のセキュリティを提供しますが、どちらも特権ポートのテストのような「IP spoofing」攻撃に対して脆弱です。すべての NIS 関連のトラフィックはファイアウォールでブロックされるべきです。

/var/yp/securenets を使っているサーバは、古い TCP/IP 実装を持つ正当なクライアントへのサービスに失敗することがあります。 これらの実装の中にはブロードキャストのホストビットをすべて 0 でセットしてしまったり、ブロードキャストアドレスの計算でサブネットマスクを見落としてしまったりするものがあります。これらの問題にはクライアントの設定を正しく行なえば解決できるものもありますが、 問題となっているクライアントシステムを引退させるか、 /var/yp/securenets を使わないようにしなければならないものもあります。

このような古風な TCP/IP の実装を持つサーバで /var/yp/securenets を使うことは実に悪い考えであり、あなたのネットワークの大部分において NIS の機能喪失を招きます。

tcpwrapper パッケージを使うとあなたの NIS サーバのレイテンシ (遅延) が増加します。特に混雑したネットワークや遅い NIS サーバでは、遅延の増加によって、クライアントプログラムのタイムアウトが起こるかもしれません。一つ以上のクライアントシステムがこれらの兆候を示したなら、あなたは問題となっているクライアントシステムを NIS スレーブサーバにして自分自身に結び付くように強制すべきです。

22.9.6. 何人かのユーザのログオンを遮断する

わたしたちの研究室には basie という、教員専用のマシンがあります。わたしたちはこのマシンを NIS ドメインの外に出したくないのですが、マスタ NIS サーバの passwd ファイルには教員と学生の両方が載っています。どうしたらいいでしょう?

当該人物が NIS のデータベースに載っていても、そのユーザがマシンにログオンできないようにする方法があります。そうするには -***username*** をクライアントマシンの /etc/master.passwd ファイルの末尾に付け足します。*username* はあなたがログインさ

せたくないと思っているユーザのユーザ名です。これは vipw で行うべきです。vipw は /etc/master.passwd への変更をチェックし、編集終了後パスワードデータベースを再構築します。たとえば、ユーザ bill が basie にログオンするのを防ぎたいなら、以下のようにします。

```
basie# vipw
[add -bill to the end, exit]
vipw: rebuilding the database...
vipw: done

basie# cat /etc/master.passwd

root:[password]:0:0::0:0:The super-user:/root:/bin/csh
toor:[password]:0:0::0:0:The other super-user:/root:/bin/sh
daemon:*:1:1::0:0:Owner of many system processes:/root:/sbin/nologin
operator:*:2:5::0:0:System &:/:/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,,:/:/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/share/man:/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/sbin/nologin
uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/
libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/sbin/
nologin
+:::::::::
-bill

basie#
```

22.9.7. ネットグループの利用

##: Udo Erdelhoff.

前節までに見てきた手法は、 極めて少ないユーザ/マシン向けに個別のルールを必要としている場合にはうまく機能します。しかし大きなネットワークでは、ユーザに触られたくないマシンへログオンを防ぐのを 忘れるでしょう し、そうでなくとも各マシンを個別に設定して回らなければならず、集中管理という NIS の恩恵を失ってしまいます。

NIS の開発者はこの問題を ネットグループ と呼ばれる方法で解決しました。その目的と意味合いは UNIX® のファイルシステムで使われている一般的なグループと比較できます。主たる相違は数値 ID が存在しないことと、ユーザアカウントと別のネットグループを含めたネットグループを定義できることです。

ネットグループは百人/台以上のユーザとマシンを含む、大きく複雑なネットワークを扱うために開発されました。あなたがこのような状況を扱わなければならないなら便利なものなのですが、一方で、この複雑さは単純な例でネットグループの説明をすることをほとんど不可能にしています。この節の残りで使われている例は、この問題を実演しています。

あなたの行なった、研究室への NIS の導入の成功が上司の目に止ったとしましょう。あなたの次の仕事は、あなたの NIS ドメインをキャンパスの他のいくつものマシンを覆うものへ拡張することです。二つの表は新しいユーザと新しいマシンの名前とその説明を含んでいます。

ユーザの名前	説明
alpha, beta	IT 学科の通常の職員
charlie, delta	IT 学科の新しい見習い
echo, foxtrott, golf, ...	一般の職員
able, baker, ...	まだインターン

マシンの名前	説明
war, death, famine, pollution	最も重要なサーバ。IT 職員だけがログオンを許されます。
pride, greed, envy, wrath, lust, sloth	あまり重要でないサーバ。 IT 学科の全員がログオンを許されます。
one, two, three, four, ...	通常のワークステーション。本当の 職員だけがログオンを許されます。
trashcan	重要なデータの入っていないひどく古いマシン。 インターンでもこのマシンの使用を許されます。

もしあなたがこの手の制限を各ユーザを個別にブロックする形で実装するなら、あなたはそのシステムにログオンすることが許されていない各ユーザについて *-user* という 1 行を、各システムの `passwd` に追加しなければならなくなるでしょう。もしあなたが 1 エントリでも忘れればトラブルに巻き込まれてしまいます。最初のセットアップの時にこれを正しく行えるのはありえることかも知れませんが、 遂には連日の業務の間に例の行を追加し忘れてしまうでしょう。結局マーフィーは楽観主義者だったのです。

この状況をネットグループで扱うといくつかの有利な点があります。各ユーザを別個に扱う必要はなく、 ユーザを一つ以上のネットグループに割り当て、 ネットグループの全メンバのログインを許可したり禁止したりすることができます。新しいマシンを追加するときはネットグループへログインの制限を定義するだけ、新しいユーザを追加するときはそのユーザを一つ以上のネットグループへ追加するだけで、 それぞれ行なうことができます。これらの変更は互いに独立なので、「ユーザとマシンの組合わせをどうするか」 は存在しなくなります。あなたの NIS のセットアップが注意深く計画されていれば、マシンへのアクセスを認めるにも拒否するにも中心の設定をたった一カ所変更するだけです。

最初のステップは NIS マップネットグループの初期化です。FreeBSD の ypinit(8) はこのマップをデフォルトで作りませんが、その NIS の実装はそれが作られさえすればそれをサポートするものです。空のマップを作るには、単に

```
ellington# vi /var/yp/netgroup
```

とタイプして内容を追加していきます。わたしたちの例では、すくなくとも IT 職員、IT 見習い、一般職員、インターンの 4 つのネットグループが必要です。

```
IT_EMP  (,alpha,test-domain)    (,beta,test-domain)
IT_APP  (,charlie,test-domain)  (,delta,test-domain)
USERS   (,echo,test-domain)     (,foxtrott,test-domain) \
 (,golf,test-domain)
INTERNS (,able,test-domain)     (,baker,test-domain)
```

IT_EMP, IT_APP 等はネットグループの名前です。それぞれの括弧で囲まれたグループが一人以上のユーザアカウントをそれに登録しています。グループの 3 つのフィールドは

1. その記述が有効なホスト (群) の名称。ホスト名を特記しなければそのエントリはすべてのホストで有効です。もしあなたがホスト名を特記するなら、あなたは闇と恐怖と全き混乱の領域に入り込んでしまうでしょう。

2. このネットグループに所属するアカウントの名称。

3. そのアカウントの NIS ドメイン。もしあなたが一つ以上の NIS ドメインの不幸な仲間なら、あなたは他の NIS ドメインからあなたのネットグループにアカウントを導入できます。

各フィールドには、ワイルドカードが使えます。詳細は netgroup(5) をご覧ください。

注記

8 文字以上のネットグループ名は、特にあなたの NIS ドメインで他のオペレーティングシステムを走らせているときは使うべきではありません。 名前には大文字小文字の区別があります。 そのためネットグループ名に大文字を使う事は、 ユーザやマシン名とネットグループ名を区別する簡単な方法です。

(FreeBSD 以外の) NIS クライアントの中には 多数のエントリを扱えないものもあります。 たとえば SunOS™ の古い版では 15 以上の エントリ を含むネットグループはトラブルを起こします。この制限は 15 ユーザ以下のサブネットグループをいくつも作り、本当のネットグループはこのサブネットグループからなるようにすることで回避できます。

```
BIGGRP1  (,joe1,domain)  (,joe2,domain)  ↺
(,joe3,domain) [...-]
BIGGRP2  (,joe16,domain)  (,joe17,domain) [...-]
BIGGRP3  (,joe31,domain)  (,joe32,domain)
BIGGROUP  BIGGRP1 BIGGRP2 BIGGRP3
```

単一のネットグループに 225 人以上のユーザをいれたいときは、このやり方を繰り返すことができます。

新しい NIS マップの有効化と配布は簡単です。

```
ellington# cd /var/yp
ellington# make
```

これで新しい 3 つの NIS マップ `netgroup`, `netgroup.byhost`, `netgroup.byuser` ができるはずです。新しい NIS マップが利用できるか確かめるには ypcat(1) を使います。

```
ellington% ypcat -k netgroup
ellington% ypcat -k netgroup.byhost
ellington% ypcat -k netgroup.byuser
```

最初のコマンドの出力は `/var/yp/netgroup` の内容に似ているはずです。2 番目のコマンドはホスト別のネットグループを作っていなければ出力されません。3 番目のコマンドはユーザに対するネットグループのリストを得るのに使えます。

クライアント側の設定は非常に簡単です。サーバ *war* を設定するには、vipw(8) を実行して以下の行

```
+:::::::::
```

を

```
+@IT_EMP:::::::::
```

に入れ替えるだけです。

今、ネットグループ *IT_EMP* で定義されたユーザのデータだけが *war* のパスワードデータベースに読み込まれ、そのユーザだけがログインを許されています。

残念ながらこの制限はシェルの ~ の機能や、ユーザ名や数値の ユーザ ID の変換ルーチンにも影響します。つまり、`cd ~user` はうまく動かず、`ls -l` はユーザ名のかわりに数値の ID を表示し `find . -user joe -print` は「No such user」で失敗します。これを避けるためには、すべてのユーザのエントリを サーバにログインすることを許さずに 読み込まなければなりません。

これはもう一行を /etc/master.passwd に追加することで実現できます。その行は以下の

+::::::::::/sbin/nologin を含んでおり、これは「すべてのエントリを読み込むが、読み込まれたエントリのシェルは /sbin/nologin で置き換えられる」ということを意味します。passwd エントリの他のフィールドを /etc/master.passwd の既定値から置き換えることも可能です。

警告

+::::::::::/sbin/nologin の行が +@IT_EMP::::::::: の行より後ろに位置することに注意してください。さもないと NIS から読み込まれた全ユーザが /sbin/nologin をログインシェルとして持つことになります。

この変更の後では、新しい職員が IT 学科に参加しても NIS マップを一つ書き換えるだけで済みます。 同様にして、あまり重要でないサーバのローカルの /etc/master.passwd のかつての +::::::::: 行を以下のように置き換えます。

```
+@IT_EMP:::::::::
+@IT_APP:::::::::
+::::::::::/sbin/nologin
```

この行は、一般のワークステーションでは以下のようになります。

```
+@IT_EMP:::::::::
+@USERS:::::::::
+::::::::::/sbin/nologin
```

これでしばらく順調に運用していましたが、数週間後、ポリシに変更がありました。IT 学科はインターンを雇い始め、IT インターンは一般のワークステーションと余り重要ではないサーバを使うことが許され、IT 見習いはメインサーバへのログインが許されました。あなたは新たなネットグループ IT_INTERN を追加して新しい IT インターンたちをそのグループに登録し、すべてのマシンの設定を変えて回ることにしました。古い諺にこうあります。「集中管理における過ちは、大規模な混乱を導く」。

いくつかのネットグループから新たなネットグループを作るという NIS の機能は、このような状況に対処するために利用できます。 その方法の一つは、役割別のネットグループを作ることです。 たとえば、重要なサーバへのログイン制限を定義するために *BIGSRV* というネットグループを作り あまり重要ではないサーバへは *SMALLSRV* というネットグループを、そして一般のワークステーション用に *USERBOX* という第 3 のネットグループを 作ることができます。これらのネットグループの各々は、 各マシンにログインすることを許されたネットグループを含みます。あなたの NIS マップネットグループの新しいエントリは、以下のようになるはずです。

```
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP  ITINTERN
USERBOX   IT_EMP  ITINTERN USERS
```

このログイン制限の定義法は、同一の制限を持つマシンのグループを定義できるときには便利なものです。残念ながらこのようなケースは例外的なものです。ほとんどの場合、各マシンに基づくログイン制限の定義機能が必要となるでしょう。

マシンごとのネットグループの定義は、上述したようなポリシの変更を扱うことができるもうひとつの方法です。このシナリオでは、各マシンの `/etc/master.passwd` は「+」で始まる 2 つの行からなります。最初のものはそのマシンへのログインを許されたアカウントを追加するもので、2 番目はその他のアカウントを `/sbin/nologin` をシェルとして追加するものです。マシン名をすべて大文字で記述したものをネットグループの名前として使うのは良い考えです。言い換えれば、件の行は次のようになるはずです。

```
+@BOXNAME :::::::::
+::::::::::/sbin/nologin
```

一度、各マシンに対してこの作業を済ませてしまえば、二度とローカルの `/etc/master.passwd` を編集する必要がなくなります。以降のすべての変更は NIS マップの編集で扱うことができます。以下はこのシナリオに対応するネットグループマップに、いくつかの便利な定義を追加した例です。

```
# Define groups of users first
IT_EMP    (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)  (,delta,test-domain)
DEPT1     (,echo,test-domain)     (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)     (,hotel,test-domain)
DEPT3     (,india,test-domain)    (,juliet,test-domain)
ITINTERN  (,kilo,test-domain)     (,lima,test-domain)
D_INTERNS (,able,test-domain)     (,baker,test-domain)
#
# Now, define some groups based on roles
USERS     DEPT1   DEPT2     DEPT3
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP    ITINTERN
USERBOX   IT_EMP  ITINTERN  USERS
#
# And a groups for a special tasks
# Allow echo and golf to access our anti-virus-machine
SECURITY  IT_EMP  (,echo,test-domain)  (,golf,test-domain)
#
# machine-based netgroups
# Our main servers
WAR       BIGSRV
FAMINE    BIGSRV
# User india needs access to this server
POLLUTION  BIGSRV  (,india,test-domain)
#
# This one is really important and needs more access restrictions
```

```
DEATH     IT_EMP
#
# The anti-virus-machine mentioned above
ONE       SECURITY
#
# Restrict a machine to a single user
TWO       (,hotel,test-domain)
# [...more groups to follow]
```

もしユーザアカウントを管理するのにデータベースの類を使っているなら、 データベースのレポートツールからマップの最初の部分を作れるようにするべきです。 そうすれば、新しいユーザは自動的にマシンにアクセスできるでしょう。

最後に使用上の注意を: マシン別のネットグループを使うことが常に賢明というわけではありません。あなたが数ダースから数百の同一の環境のマシンを学生の研究室に配置しているのならば、NIS マップのサイズを手頃な範囲に押さえるために、マシン別のネットグループのかわりに役割別のネットグループを使うべきです。

22.9.8. 忘れてはいけないこと

NIS 環境にある今、今までとは違ったやり方が必要なことがいくつかあります。

- 研究室にユーザを追加するときは、それをマスター NIS サーバに だけ 追加しなければならず、さらに NIS マップを再構築することを忘れてはいけません。これを忘れると新しいユーザは NIS マスタ以外のどこにもログインできなくなります。たとえば、新しくユーザ「jsmith」をラボに登録したいときは以下のようにします。

  ```
  # pw useradd jsmith
  # cd /var/yp
  # make test-domain
  ```

  `pw useradd jsmith` のかわりに `adduser jsmith` を使うこともできます。

- 管理用アカウントを NIS マップから削除してください。管理用アカウントやパスワードを、それらのアカウントへアクセスさせてはいけないユーザが居るかも知れないマシンにまで伝えて回りたいとは思わないでしょう。

- NIS のマスタとスレーブをセキュアに、そして機能停止時間を最短に保ってください。もし誰かがこれらのマシンをクラックしたり、あるいは単に電源を落としたりすると、彼らは実質的に多くの人を研究室へログインできなくしてしまえます。

  これはどの集中管理システムにとってももっとも大きな弱点でしょう。 あなたの NIS サーバを守らなければ怒れるユーザと対面することになるでしょう!

22.9.9. NIS v1 との互換性

FreeBSD の ypserv は、 NIS v1 クライアントを部分的にサポートしています。FreeBSD の NIS 実装は NIS v2 プロトコルのみを使用していますが、ほかの実装で

は、古いシステムとの下位互換性を持たせるため v1 プロトコルをサポートしているものもあります。そのようなシステムに付いている ypbind デーモンは、必要がないにもかかわらず NIS v1 のサーバとの結合を成立させようとします (しかも v2 サーバからの応答を受信した後でも、ブロードキャストをし続けるかも知れません)。FreeBSD の ypserv は、クライアントからの通常のリクエストはサポートしていますが、v1 のマップ転送リクエストはサポートしていないことに注意してください。つまり FreeBSD の ypserv を、v1 だけをサポートするような古い NIS サーバと組み合わせて マスターやスレーブサーバとして使うことはできません。幸いなことに、現在、そのようなサーバが使われていることは ほとんどないでしょう。

22.9.10. NIS クライアントとしても動作している NIS サーバ

複数のサーバが存在し、サーバ自身が NIS クライアントでもあるようなドメインで ypserv が実行される場合には注意が必要です。 一般的に良いとされているのは、 他のサーバと結合をつくるようにブロードキャストさせるのではなく、サーバをそれ自身に結合させることです。もし、サーバ同士が依存関係を持っていて、一つのサーバが停止すると、 奇妙なサービス不能状態に陥ることがあります。 その結果、すべてのクライアントはタイムアウトを起こして 他のサーバに結合しようと試みますが、これにかかる時間はかなり大きく、 サーバ同士がまた互いに結合してしまったりすると、 サービス不能状態はさらに継続することになります。

`ypbind` に `-S` オプションフラグを指定して実行することで、ホストを特定のサーバに結合することが可能です。NIS サーバを再起動するたびに、これを手動で行いたくないなら、次の行を `/etc/rc.conf` に追加すればよいでしょう。

```
nis_client_enable="YES" # run client stuff as well
nis_client_flags="-S NIS domain,server"
```

詳細については ypbind(8) を参照してください。

22.9.11. パスワード形式

NIS を実装しようする人の誰もがぶつかる問題の一つに、パスワード形式の互換性があります。NIS サーバが DES 暗号化パスワード使っている場合には、同様に DES を使用しているクライアントしか対応できません。たとえば Solaris™; の NIS クライアントがネットワーク内にある場合、ほぼ確実に DES 暗号化パスワードを使用しなければならないでしょう。

サーバとクライアントがどのライブラリを使用しているかは、`/etc/login.conf` を確認してください。 ホストが DES 暗号パスワードを使用するように設定されている場合、`default` クラスには以下のようなエントリが含まれます。

```
default:\
    :passwd_format=des:\
    :copyright=/etc/COPYRIGHT:\
```

```
    [Further entries elided]
```

`passwd_format` 特性について他に利用可能な値は `blf` および `md5` (それぞれ Blowfish および MD5 暗号化パスワード) です。

`/etc/login.conf` を変更したときは、ログイン特性データベースも再構築しなければなりません。これは `root` 権限で下記のようにコマンドを実行すればできます。

```
# cap_mkdb /etc/login.conf
```

注記

すでに `/etc/master.passwd` 内に記録されているパスワード形式は、ログイン特性データベースが再構築された後、ユーザが彼らのパスワードをはじめて変更するまで変更されないでしょう。

次に、 パスワードが選択した形式で暗号化されることを確実にするために、 さらに `/etc/auth.conf` 内の `crypt_default` において、 選択したパスワード形式に高い優先順位がついていることも確認してください。 そうするためには、選択した形式をリストの先頭に置いてください。 たとえば DES 暗号化されたパスワードを使用するときは、エントリは次のようになります。

```
crypt_default = des blf md5
```

FreeBSD 上の各 NIS サーバおよびクライアントにおいて上記の手順に従えば、ネットワーク内でどのパスワード形式が使用されるかが それらのマシン間で整合されているということを確信できます。NIS クライアント上で問題があれば、ここから問題となりそうな部分を探すと良いでしょう。覚えておいてください: 異種混在ネットワークに NIS サーバを配置したいときには、DES が最大公約数的な標準となるでしょうから、すべてのシステムで DES を使用しなければならないかもしれません。

22.10. DHCP

##: Greg Sutter.

22.10.1. DHCP とは何でしょうか?

DHCP (Dynamic Host Configuration Protocol) は、システムをネットワークに接続するだけで、ネットワークでの通信に必要な情報を入手することができる仕組みです。FreeBSD では ISC (Internet Software Consortium) による DHCP の実装を使用しています。したがって、ここでの説明のうち実装によって異なる部分は ISC のもの用になっています。

22.10.2. この節で説明していること

この節は ISC DHCP システムのクライアント側およびサーバ側の構成要素の両方について説明します。クライアント側のプログラムである `dhclient` は FreeBSD のベースシステム内に含まれています。そして、サーバ側の要素は net/isc-dhcp3-server port から利用可能です。下記の説明の他に、dhclient(8), dhcp-options(5) および dhclient.conf(5) マニュアルページが役にたつ情報源です。

22.10.3. DHCP の動作

クライアントとなるマシン上で、DHCP のクライアントである `dhclient` を実行すると、まず設定情報の要求をブロードキャストします。デフォルトでは、このリクエストには UDP のポート 68 を使用します。サーバは UDP のポート 67 で応答し、クライアントの IP アドレスと、ネットマスクやルータ、DNS サーバなどの関連する情報を提供します。これらの情報のすべては DHCP の「リース」の形で送られ、DHCP サーバ管理者によって決められたある一定の時間内でのみ有効になります。これによって、ネットワークに存在しなくなったホストの IP アドレスは自動的に回収されることになります。

DHCP クライアントはサーバから非常に多くの情報を取得することができます。dhcp-options(5) に非常に大きなリストが載っています。

22.10.4. FreeBSD への組み込み

FreeBSD は ISC の DHCP クライアントである `dhclient` を完全に組み込んでいます。DHCP クライアントはインストーラと基本システムの両方で提供されています。ですから DHCP サーバを走らせているネットワーク上ではネットワーク関係の設定についての詳細な知識は必要になりません。`dhclient` は、3.2 以降のすべての FreeBSD の配布物に含まれています。

DHCP は sysinstall で対応されており、sysinstall でのネットワークインタフェイス設定の際は、「このインタフェイスの設定として DHCP を試してみますか? (Do you want to try DHCP configuration of this interface?)」という質問が最初になされます。これに同意することで `dhclient` が実行され、それが成功すればネットワークの設定情報は自動的に取得されます。

システム起動時に DHCP を使ってネットワーク情報を取得するように するには、次の二つを行なう必要があります。

- `bpf` デバイスがカーネルに組み込まれていることを確認します。これを組み込むには、カーネルコンフィグレーションファイルに `pseudo-device bpf` という行を追加し、カーネルを再構築します。カーネルの構築に関する詳細は、9章FreeBSD カーネルのコンフィグレーション を参照してください。

  `bpf` デバイスは、FreeBSD にはじめから用意されている `GENERIC` カーネルに組み込まれていますので、自分で設定を変えたカスタムカーネルを使っているのでなければ、DHCP を動作させるためにカーネルを再構築する必要はありません。

注記

セキュリティに関心のある方向けに注意しておきます。bpf デバイスは、パケットスニファ (盗聴プログラム) を動作させることができる (ただし root 権限が必要) デバイスです。bpf は DHCP を動作させるために かならず必要ですが、セキュリティが非常に重要な場面では DHCP をいつか使うかもしれないというだけで bpf デバイスをカーネルに追加すべきではないでしょう。

- /etc/rc.conf を編集して、次の行を追加してください。

```
ifconfig_fxp0="DHCP"
```

注記

で説明されているように fxp0 の部分を、動的に設定したいインタフェースの名前で置き換えることを忘れないようにしてください。

もし、使っている dhclient の場所を変更していたり、dhclient にフラグを渡したい場合は、同様に下のように書き加えてください。

```
dhcp_program="/sbin/dhclient"
dhcp_flags=""
```

DHCP サーバ dhcpd は、Ports Collection に net/isc-dhcp3-server の一部として収録されています。この port には ISC DHCP サーバと文書が含まれています。

22.10.5. 関連ファイル

- /etc/dhclient.conf

  dhclient は設定ファイル /etc/dhclient.conf を必要とします。大抵の場合、このファイルはコメントだけであり、デフォルトが通常使いやすい設定になっています。この設定ファイルは dhclient.conf(5) マニュアルページで説明しています。

- /sbin/dhclient

`dhclient` は静的にリンクされており、`/sbin` に置かれています。dhclient(8) マニュアルページで `dhclient` コマンドについてより詳しく説明しています。

- `/sbin/dhclient-script`

  `dhclient-script` は FreeBSD 特有の、DHCP クライアント設定スクリプトです。これについては dhclient-script(8) マニュアルページで説明されていますが、これを編集する必要はほとんど発生しないでしょう。

- `/var/db/dhclient.leases`

  DHCP クライアントはこのファイルに有効なリースのデータベースをログとして記録します。dhclient.leases(5) にもうすこし詳しい解説があります。

22.10.6. 参考になる文献

DHCP のプロトコルは RFC 2131 に完全に記述されています。また dhcp.org にも有用な情報源が用意されています。

22.10.7. DHCP サーバのインストールと設定

22.10.7.1. この節で説明していること

この節は DHCP の ISC (Internet Software Consortium) 実装を用いて FreeBSD システムを DHCP サーバとして動作させる方法の情報を提供します。

DHCP のサーバ部分は FreeBSD の一部として提供されません。したがって、このサービスを提供するために net/isc-dhcp3-server port をインストールする必要があるでしょう。Ports Collection を使用する情報についての詳細は 5章アプリケーションのインストール - packages と ports を参照してください。

22.10.7.2. DHCP サーバのインストール

FreeBSD システムを DHCP サーバとして設定するために、bpf(4) デバイスがカーネルに組み込まれていることを保証する必要があります。そうするためには、カーネルコンフィギュレーションファイルに `pseudo-device bpf` を追加して、カーネルを再構築してください。カーネルの構築に関する詳細は 9章FreeBSD カーネルのコンフィグレーション を参照してください。

`bpf` デバイスは、FreeBSD にはじめから用意されている `GENERIC` カーネルの一部なので、DHCP を動作させるためにカスタムカーネルを作成する必要はありません。

注記

セキュリティを特に意識する人は、bpf　bpf　はパケットスニファ(盗聴プログラム) が正常に (このようなプログラムはさらに特権アクセスを必要としますが)　動作することを可能にするデバイスでもあることに注意してください。 bpf　は　DHCP　を使用するために必要 です。しかし、セキュリティをとても気にしているなら、DHCP をいつか使うかもしれないというだけで bpf デバイスをカーネルに含めるべきではないでしょう。

次に行わねばならないのは、 net/isc-dhcp3-server　port　によってインストールされた　dhcpd.conf　のサンプルを編集することです。 デフォルトでは、これは　/usr/local/etc/dhcpd.conf.sample　で、 編集する前にこれを　/usr/local/etc/dhcpd.conf　にコピーするべきでしょう。

22.10.7.3. DHCP サーバの設定

dhcpd.conf　はサブネットおよびホストに関する宣言で構成されます。 例を使って説明するのが最も簡単でしょう。

```
option domain-name "example.com";❶
option domain-name-servers 192.168.4.100;❷
option subnet-mask 255.255.255.0;❸

default-lease-time 3600;❹
max-lease-time 86400;❺
ddns-update-style none;❻

subnet 192.168.4.0 netmask 255.255.255.0 {
  range 192.168.4.129 192.168.4.254;❼
  option routers 192.168.4.1;❽
}

host mailhost {
  hardware ethernet 02:03:04:05:06:07;❾
  fixed-address mailhost.example.com;❿
}
```

❶ このオプションは、デフォルト探索ドメインとしてクライアントに渡されるドメインを指定します。これが意味するところの詳細については resolv.conf(5) を参照してください。

❷ このオプションはクライアントが使用する、コンマで区切られた DNS サーバのリストを指定します。

❸ クライアントに渡されるネットマスクです。
❹ クライアントは特定のリース期限を要求することもできます。 それ以外の場合は、サーバはこのリース期限値 (秒) でリースを割り当てるでしょう。
❺ これはサーバがリースする時間の最大値です。クライアントがこれより長いリースを要求しても、`max-lease-time` 秒だけしか有効にならないでしょう。
❻ このオプションは、リースが受理、またはリリースされたときに DHCP サーバが DNS を更新しようとするかどうかを指定します。ISC 実装では、このオプションは必須 です。
❼ これはどの範囲の IP アドレスが、クライアントに割り当てるために予約されたプールに使用されるかを示します。この範囲に含まれている IP アドレスはクライアントに渡されます。
❽ クライアントに供給されるデフォルトゲートウェイを宣言します。
❾ (リクエストが生じた時に DHCP サーバがホストを認識できるように) ホストのハードウェア MAC アドレスを指定します。
❿ ホストに常に同じ IP アドレスを付与することを指定します。DHCP サーバはリース情報を返す前にホスト名の名前解決をするので、ここにホスト名を書いても構いません。

`dhcpd.conf` を書き終えたら以下のコマンドでサーバを起動できます。

```
# /usr/local/etc/rc.d/isc-dhcpd.sh start
```

今後サーバの設定に変更を加える必要が生じた時には、`SIGHUP` シグナルを dhcpd に送っても、多くのデーモンがそうであるようには、設定ファイルが再読み込み されないことに注意してください。`SIGTERM` シグナルを送ってプロセスを停止し、それから上記のコマンドを用いて再起動させる必要があります。

22.10.7.4. ファイル

- `/usr/local/sbin/dhcpd`

  dhcpd は静的にリンクされ `/usr/local/sbin` に置かれます。dhcpd に関するそれ以上の情報は port とともにインストールされる dhcpd(8) マニュアルページにあります。

- `/usr/local/etc/dhcpd.conf`

  dhcpd はクライアントへのサービス提供をはじめる前に設定ファイル `/usr/local/etc/dhcpd.conf` を必要とします。このファイルは、サーバの稼働に関する情報に加えて、サービスされているクライアントに提供される情報のすべてを含む必要があります。この設定ファイルについての詳細は、port によってインストールされる dhcpd.conf(5) マニュアルページを参照してください。

- `/var/db/dhcpd.leases`

DHCP サーバは発行したリースのデータベースをこのファイルにログとして保持します。port によってインストールされる dhcpd.leases(5) にはもう少し詳しい説明があります。

- `/usr/local/sbin/dhcrelay`

  dhcrelay は、DHCP サーバがクライアントからのリクエストを、 別のネットワーク上にある DHCP サーバに転送する高度な環境下で使用されます。この機能が必要なら、net/isc-dhcp3-server port をインストールしてください。port とともに提供される dhcrelay(8) マニュアルページにはより詳細な情報が含まれます。

22.11. DNS

##: Chern Lee.

22.11.1. 概観

FreeBSD はデフォルトでは DNS プロトコルの最も一般的な実装である BIND (Berkeley Internet Name Domain) を使用します。DNS はホスト名を IP アドレスに、そして IP アドレスをホスト名に関連づけるプロトコルです。 たとえば `www.FreeBSD.org` に対する問い合わせは The FreeBSD Project の ウェブサーバの IP アドレスを受け取るでしょう。その一方で `ftp.FreeBSD.org` に対する問い合わせは、対応する FTP マシンの IP アドレスを返すでしょう。同様に、その逆のことも可能です。IP アドレスに対する問い合わせを行うことで、そのホスト名を解決することができます。DNS 検索を実行するために、システム上でネームサーバを動作させる必要はありません。

DNS は、個々のドメイン情報を格納およびキャッシュした、権威のあるルートサーバおよび他の小規模なネームサーバによる多少複雑なシステムによって、インターネット全体にわたって協調して動作します。

この文書は FreeBSD で安定版として利用されている BIND 8.x について説明します。FreeBSD では BIND 9.x を net/bind9 port からインストールできます。

RFC1034 および RFC1035 は DNS プロトコルを定義しています。

現在のところ BIND は Internet Software Consortium (www.isc.org) によって保守されています。

22.11.2. 用語

この文書を理解するには DNS 関連の用語をいくつか理解しなければいけません。

用語	定義
正引き DNS	ホスト名から IP アドレスへの対応です。
オリジン (origine)	特定のゾーンファイルによってカバーされるドメインへの参照です。
named, BIND, ネームサーバ	FreeBSD 内の BIND ネームサーバパッケージの一般名称です。
リゾルバ (resolver)	マシンがゾーン情報についてネームサーバに問い合わせるシステムプロセスです。
逆引き DNS	正引き DNS の逆です。つまり IP アドレスからホスト名への対応です。
ルートゾーン	インターネットゾーン階層の起点です。 すべてのゾーンはルートゾーンの下に属します。 これはファイルシステムのすべてのファイルがルートディレクトリの下に属することと似ています。
ゾーン	同じ権威によって管理される個々の DNS ドメイン、 DNS サブドメイン、あるいは DNS の一部分です。

ゾーンの例:

- `.` はルートゾーンです。
- `org.` はルートゾーンの下のゾーンです。
- `example.org` は `org.` ゾーンの下のゾーンです。
- `foo.example.org.` はサブドメインで、`example.org.` の下のゾーンです。
- `1.2.3.in-addr.arpa` は 3.2.1.* の IP 空間に含まれるすべての IP アドレスを参照するゾーンです。

見て分かるように、ホスト名のより詳細な部分はその左側に現れます。 たとえば `example.org.` は `org.` より限定的です。同様に `org.` はルートゾーンより限定的です。ホスト名の各部分のレイアウトはファイルシステムに非常に似ています。たとえば `/dev` はルートの下であることなどです。

22.11.3. ネームサーバを実行する理由

ネームサーバは通常二つの形式があります: 権威のあるネームサーバとキャッシュネームサーバです。

権威のあるネームサーバは以下の場合に必要です。

- 問い合わせに対して信頼できる返答をすることで、ある人が DNS 情報を世界に向けて発信したいとき。
- example.org といったドメインが登録されており、その下にあるホスト名に IP アドレスを割り当てる必要があるとき。
- IP アドレスブロックが (IP からホスト名への) 逆引き DNS エントリを必要とするとき。
- プライマリサーバがダウンしているかまたはアクセスできない場合に、代わりに問い合わせに対してスレーブと呼ばれるバックアップネームサーバが返答しなければならないとき。

キャッシュネームサーバは以下の場合に必要です。

- ローカルのネームサーバが、外部のネームサーバに問い合わせするよりも、キャッシュしてより速く返答できるとき。
- ネットワークトラフィックの総量を減らしたいとき (DNS のトラフィックはインターネットトラフィック全体の 5% 以上を占めることが測定されています)

www.FreeBSD.org に対する問い合わせを発したとき、リゾルバは大体の場合上流の ISP のネームサーバに問い合わせをして返答を得ます。ローカルのキャッシュ DNS サーバがあれば、問い合わせはキャッシュ DNS サーバによって外部に対して一度だけ発せられます。情報がローカルに蓄えられるので、追加の問い合わせはいずれもローカルネットワークの外側にまで確認しなくてもよくなります。

22.11.4. 動作のしくみ

FreeBSD では BIND デーモンは自明な理由から named と呼ばれます。

ファイル	説明
named	BIND デーモン
ndc	ネームデーモンコントロールプログラム
/etc/namedb	BIND のゾーン情報が置かれるディレクトリ
/etc/namedb/named.conf	デーモンの設定ファイル

ゾーンファイルは通常 /etc/namedb ディレクトリ内に含まれており、ネームサーバによって処理される DNS ゾーン情報を含んでいます。

22.11.5. BIND の起動

BIND はデフォルトでインストールされているので、すべてを設定することは比較的単純です。

named デーモンが起動時に開始されることを保証するには、`/etc/rc.conf` に以下の変更をいれてください。

```
named_enable="YES"
```

デーモンを手動で起動するためには (設定をした後で)

```
# ndc start
```

22.11.6. 設定ファイル

22.11.6.1. `make-localhost` の利用

次のコマンドが

```
# cd /etc/namedb
# sh make-localhost
```

ローカル逆引き DNS ゾーンファイルを `/etc/namedb/localhost.rev` に適切に作成することを確認してください。

22.11.6.2. `/etc/namedb/named.conf`

```
// $FreeBSD$
//
// ####### named(8) ##########################
// ###########DNS ########################
// #####################################
// ##########################

options {
        directory "/etc/namedb";

// "forwarders" ###############################
// #####################################
// #####:
//
//      forward only;

// ############### DNS ############## IP ####
// #####################################
// ###################### DNS ##############
/*
        forwarders {
                127.0.0.1;
        };
```

```
*/
```

コメントが言っている通り、上流のキャッシュの恩恵を受けるために `forwarders` をここで有効にすることができます。 通常の状況では、ネームサーバはインターネットの特定のネームサーバを調べて、 探している返答を見つけるまで再帰的に問い合わせを行います。これが有効になっていれば、まず上流のネームサーバ (または 与えられたネームサーバ) に問い合わせて、そのキャッシュを利用するでしょう。問い合わせをする上流のネームサーバが極度に通信量が多く、 高速であった場合、これを有効にする価値があるかもしれません。

警告

ここに `127.0.0.1` を指定しても動作 しません。 上流のネームサーバの IP アドレスに変更してください。

```
        /*
         * ################################
         * ### quiery-source #################
         * ### BIND ######### 53 ################
         * BIND 8.1 ####################
         */
        // query-source address * port 53;

        /*
         * ################################
         * #################
         */
        // dump-file "s/named_dump.db";
};

// ##: ####################
/*
host { any; } {
        topology {
                127.0.0.0/8;
        };
};
*/

// ####################################
// ######
//
// ###################################
// /etc/resolv.conf # 127.0.0.1 ######################
// /etc/rc.conf ##################

zone "." {
```

```
        type hint;
        file "named.root";
};

zone "0.0.127.IN-ADDR.ARPA" {
        type master;
        file "localhost.rev";
};

zone
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.↺
INT" {
        type master;
        file "localhost.rev";
};

// ##: ### IP ##############################
// #############
//
// ####################################
// ################################ IP ####
// ##################
//
// ######## (IN-ADDR.ARPA) ##################!
// (### ".IN-ADDR.ARPA" ############# IP ###########
// ######)
//
// ################## DNS ### BIND ###########
// ####################################
// ##############
//
// ##: ######################:-) ##########
// ##############
//
// ##!!! FreeBSD # bind ############ (rc.conf ## named_flags
// #########)#####################bind ####
// ############################:
//
//      mkdir /etc/namedb/s
//      chown bind:bind /etc/namedb/s
//      chmod 750 /etc/namedb/s
```

BIND を砂場 (sandbox) で (訳注: chroot をもちいて) 動作させるための詳細は 砂場で named を実行する を参照してください。

```
/*
zone "example.com" {
        type slave;
        file "s/example.com.bak";
        masters {
                192.168.1.1;
        };
};
```

```
zone "0.168.192.in-addr.arpa" {
        type slave;
        file "s/0.168.192.in-addr.arpa.bak";
        masters {
                192.168.1.1;
        };
};
*/
```

named.conf の中で、 上記は転送と逆引きゾーンのためのスレーブエントリの例です。

新しくサービスするそれぞれのゾーンについて、新規のエントリを named.conf に加えなければいけません。

たとえば example.org に対する最もシンプルなゾーンエントリは以下のようになります。

```
zone "example.org" {
 type master;
 file "example.org";
};
```

このゾーンは type 命令で示されているようにマスタで、ゾーン情報を file 命令で指示された /etc/namedb/example.org ファイルに保持しています。

```
zone "example.org" {
 type slave;
 file "example.org";
};
```

スレーブの場合、ゾーン情報は特定のゾーンのマスタネームサーバから転送され、指定されたファイルに保存されます。 マスタサーバが停止するか到達できない場合には、 スレーブサーバが転送されたゾーン情報を保持していて、サービスできるでしょう。

22.11.6.3. ゾーンファイル

example.org に対するマスタゾーンファイル (/etc/namedb/example.org に保持されます) の例は以下のようになります。

```
$TTL 3600

example.org. IN SOA ns1.example.org. admin.example.org. (
                        5               -; Serial
                        10800           -; Refresh
                        3600            -; Retry
                        604800          -; Expire
                        86400 )         -; Minimum TTL
```

```
; DNS Servers
@       IN NS           ns1.example.org.
@       IN NS           ns2.example.org.

; Machine Names
localhost       IN A    127.0.0.1
ns1             IN A    3.2.1.2
ns2             IN A    3.2.1.3
mail            IN A    3.2.1.10
@               IN A    3.2.1.30

; Aliases
www             IN CNAME        @

; MX Record
@               IN MX   10      mail.example.org.
```

「.」 が最後についているすべてのホスト名は正確なホスト名であり、 一方で 「.」 で終了しないすべての行はオリジンが参照されることに注意してください。 たとえば `www` は `www + ####` に展開されます。この架空のゾーンファイルでは、 オリジンは `example.org.` なので `www` は `www.example.org.` に展開されます。

ゾーンファイルの書式は次のとおりです。

```
recordname      IN recordtype   value
```

DNS レコードに使われる最も一般的なものは以下のとおりです。

SOA
: ゾーン権威の起点

NS
: 権威のあるネームサーバ

A
: ホストのアドレス

CNAME
: 別名としての正規の名称

MX
: メールエクスチェンジャ

PTR
: ドメインネームポインタ (逆引き DNS で使用されます)

```
example.org. IN SOA ns1.example.org. admin.example.org. (
                        5               -; Serial
```

```
                        10800           -; Refresh after 3 hours
                        3600            -; Retry after 1 hour
                        604800          -; Expire after 1 week
                        86400 )         -; Minimum TTL of 1 day
```

example.org.
: このゾーンのオリジンでもあるドメイン名

ns1.example.org.
: このゾーンに対して権威のあるプライマリネームサーバ

admin.example.org.
: このゾーンの責任者。@ を置き換えた電子メールアドレスを指定します。(<admin@example.org > は admin.example.org になります)

5
: ファイルのシリアル番号です。 これはファイルが変更されるたびに増加させる必要があります。 現在では多くの管理者は yyyymmddrr という形式をシリアル番号として使用することを好みます。 2001041002 は最後に修正されたのが 2001/04/10 で、後ろの 02 はその日で二回目に修正されたものであるということを意味するでしょう。シリアル番号は、それが更新されたときにスレーブネームサーバに対してゾーンを通知するので重要です。

```
@       IN NS           ns1.example.org.
```

これは NS エントリです。このゾーンに対して権威のある返答を返すネームサーバはすべて、このエントリを一つ有していなければなりません。ここにある @ は example.org. を意味します。@ はオリジンに展開されます。

```
localhost       IN A    127.0.0.1
ns1             IN A    3.2.1.2
ns2             IN A    3.2.1.3
mail            IN A    3.2.1.10
@               IN A    3.2.1.30
```

A レコードはマシン名を示します。上記のように ns1.example.org は 3.2.1.2 に結びつけられるでしょう。ふたたびオリジンを示す @ がここに使用されていますが、これは example.org が 3.2.1.30 に結び付けられることを意味しています。

```
www             IN CNAME        @
```

CNAME レコードは通常マシンに別名を与えるときに使用されます。例では www はオリジン、すなわち example.org (3.2.1.30) のアドレスをふられたマシンへの別名を与えます。CNAME はホスト名の別名、または複数のマシン間で一つのホスト名をラウンド

ロビン (訳注: 問い合わせがあるたびに別の IP アドレスを返すことで、一台にアクセスが集中することを防ぐ手法) するときに用いられます。

```
@               IN MX   10      mail.example.org.
```

MX レコードは、ゾーンに対してどのメールサーバがやってきたメールを扱うことに責任を持っているかを示します。mail.example.org はメールサーバのホスト名で、10 はメールサーバの優先度を示します。

優先度が 3,2 または 1 などのメールサーバをいくつも置くことができます。example.org へ送ろうとしているメールサーバははじめに一番優先度の高いメールサーバに接続しようとします。 そして接続できない場合、二番目に優先度の高いサーバに接続しようとし、以下、メールが適切に配送されるまで同様に繰り返します。

in-addr.arpa ゾーンファイル (逆引き DNS) に対しても A または CNAME の代わりに PTR エントリが用いられることを除けば、同じ書式が使われます。

```
$TTL 3600

1.2.3.in-addr.arpa. IN SOA ns1.example.org. admin.example.org. (
                        5               -; Serial
                        10800           -; Refresh
                        3600            -; Retry
                        604800          -; Expire
                        3600 )          -; Minimum

@       IN NS   ns1.example.org.
@       IN NS   ns2.example.org.

2       IN PTR  ns1.example.org.
3       IN PTR  ns2.example.org.
10      IN PTR  mail.example.org.
30      IN PTR  example.org.
```

このファイルは上記の架空のドメインの IP アドレスからホスト名への対応を与えます。

22.11.7. キャッシュネームサーバ

キャッシュネームサーバはどのゾーンに対しても権威をもたないネームサーバです。
キャッシュネームサーバは単に自分で問い合わせをし、 後で使えるように問い合わせの結果を覚えておきます。 これを設定するには、ゾーンを何も含まずに、 通常通りネームサーバを設定してください。

22.11.8. 砂場で named を実行する

セキュリティを強めるために named(8) を非特権ユーザで実行し、砂場のディレクトリ内に chroot(8) して実行したいと思うかもしれません。こうすると named デーモンは砂

場の外にはまったく手を出すことができません。named が乗っ取られたとしても、これによって起こりうる損害が小さくなるでしょう。FreeBSD にはデフォルトで、そのための bind というユーザとグループがあります。

注記

多くの人々は named を chroot するように設定する代わりに、jail(8) 環境内で named を実行することを奨めるでしょう。この節ではそれは扱いません。

named は砂場の外 (共有ライブラリ、ログソケットなど) にアクセスできないので、named を正しく動作させるためにいくつもの段階を経る必要があります。下記のチェックリストにおいては、砂場のパスは /etc/namedb で、このディレクトリの内容には何も手を加えていないと仮定します。root 権限で次のステップを実行してください。

- named が存在することを期待しているディレクトリをすべて作成します。

```
# cd /etc/namedb
# mkdir -p bin dev etc var/tmp var/run master slave
# chown bind:bind slave var/* ❶
```

❶ これらのディレクトリに対して named が必要なのは書き込み権限だけなので、それだけを与えます。

- 基本ゾーンファイルと設定ファイルの編集と作成を行います。

```
# cp /etc/localtime etc ❶
# mv named.conf etc && ln -sf etc/named.conf
# mv named.root master

# sh make-localhost && mv localhost.rev localhost-v6.↺
rev master
# cat > master/named.localhost
$ORIGIN localhost.
$TTL 6h
@ IN SOA localhost. postmaster.localhost. (
   1 ; serial
   3600 ; refresh
   1800 ; retry
   604800 ; expiration
   3600 ) ; minimum
 IN NS localhost.
 IN A  127.0.0.1
```

```
^D
```

❶ これは named が syslogd(8) に正しい時刻でログを書き込むことを可能にします。

- 4.9-RELEASE より前のバージョンの FreeBSD を使用している場合、静的リンクされた named-xfer を構築し、砂場にコピーしてください。

```
# cd /usr/src/lib/libisc
# make cleandir && make cleandir && make depend && ↺
make all
# cd /usr/src/lib/libbind
# make cleandir && make cleandir && make depend && ↺
make all
# cd /usr/src/libexec/named-xfer
# make cleandir && make cleandir && make depend && ↺
make NOSHARED=yes all
# cp named-xfer /etc/namedb/bin && chmod 555 /etc/
namedb/bin/named-xfer ❶
```

静的リンクされた named-xfer をインストールしたら、ソースツリーの中にライブラリまたはプログラムの古くなったコピーを残さないように、掃除する必要があります。

```
# cd /usr/src/lib/libisc
# make cleandir
# cd /usr/src/lib/libbind
# make cleandir
# cd /usr/src/libexec/named-xfer
# make cleandir
```

❶ このステップは時々失敗することが報告されています。 もし失敗した場合、次のコマンドを実行してください。

```
# cd /usr/src && make cleandir && make cleandir
```

そして /usr/obj ツリーを削除します。

```
# rm -fr /usr/obj && mkdir /usr/obj
```

これはソースツリーからすべての「がらくた」を一掃します。もう一度上記の手順を行うと、今度はうまく動作するでしょう。

バージョン 4.9-RELEASE 以降の FreeBSD を使用している場合 /usr/libexec にある named-xfer のコピーはデフォルトで静的リンクされています。砂場にコピーするために単純に cp(1) が使えます。

- named が見ることができ、書き込むことのできる `dev/null` を作成します。

```
# cd /etc/namedb/dev && mknod null c 2 2
# chmod 666 null
```

- `/etc/namedb/var/run/ndc` から `/var/run/ndc` へのシンボリックリンクを作成します。

```
# ln -sf /etc/namedb/var/run/ndc /var/run/ndc
```

注記

これは単に ndc(8) を実行するたびに `-c` オプションを指定しなくてもよいようにするだけです。/var/run の中身は起動時に削除されるため、これが有用だと思うなら、このコマンドをルートの crontab に `@reboot` オプションを指定して追加してください。詳細については crontab(5) を参照してください。

- named が書き込める追加の `log` ソケットを作成するように syslogd(8) を設定します。これを行うためには、`/etc/rc.conf` 内の `syslogd_flags` 変数に `-l /etc/namedb/dev/log` を加えてください。

- 次の行を `/etc/rc.conf` に加えて named が起動し、自身を砂場内に `chroot` するように調整します

```
named_enable="YES"
named_flags="-u bind -g bind -t /etc/namedb /etc/named.conf"
```

注記

設定ファイル *`/etc/named.conf`* は 砂場のディレクトリに対して相対的な フルパスで表されることに注意してください。つまり、上記の行で示されたファイルは実際には `/etc/namedb/etc/named.conf` です。

次のステップは named がどのゾーンを読み込むか、そしてディスク上のどこにゾーンファイルがあるのかを知るために `/etc/namedb/etc/named.conf` を編集するこ

とです。下記に例をコメントを加えて示します (ここで特にコメントされていない内容については、砂場の中で動作させない DNS サーバの設定と同じです)。

```
options {
        directory "/";❶
        named-xfer "/bin/named-xfer";❷
        version "";  // Don't reveal BIND version
        query-source address * port 53;
};
// ndc control socket
controls {
        unix "/var/run/ndc" perm 0600 owner 0 group 0;
};
// Zones follow:
zone "localhost" IN {
        type master;
        file "master/named.localhost";❸
        allow-transfer { localhost; };
        notify no;
};
zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "master/localhost.rev";
        allow-transfer { localhost; };
        notify no;
};
zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.↺
ip6.int" {
 type master;
 file "master/localhost-v6.rev";
 allow-transfer { localhost; };
 notify no;
};
zone "." IN {
        type hint;
        file "master/named.root";
};
zone "private.example.net" in {
        type master;
        file "master/private.example.net.db";
 allow-transfer { 192.168.10.0/24; };
};
zone "10.168.192.in-addr.arpa" in {
        type slave;
        masters { 192.168.10.2; };
        file "slave/192.168.10.db";❹
};
```

❶ `directory` は `/` を指定します。named が必要とするファイルはすべてこのディレクトリにあります。(この指定は「通常の」(訳注: 砂場内で動作させない) ユーザにとっての `/etc/namedb` と等価です)。

❷ `named-xfer` バイナリへの (named にとっての) フルパスを指定します。named はデフォルトで `named-xfer` を `/usr/libexec` から探すようにコンパイルされているので、これが必要です

❸ このゾーンに対するゾーンファイルを named が見つけられるようにファイル名を (上記と同様に `directory` からの相対パスで) 指定します。

❹ このゾーンに対するゾーン情報がマスタサーバからが転送されたあとに、named がゾーンファイルのコピーを書き込むファイル名を (上記と同様に `directory` からの相対パスで) 指定します。これが、上記のように設定段階で `slave` ディレクトリの所有者を `bind` に変更する理由です。

上記のステップを完了したら、サーバを再起動するか syslogd(8) を再起動し、named(8) を起動してください。その際、`syslogd_flags` および `named_flags` に新たに指定したオプションが有効になっていることを確かめてください。これで named を砂場のなかで動作させることができているはずです!

22.11.9. セキュリティ

BIND は DNS の最も一般的な実装ではありますが、常にセキュリティ問題を抱えています。問題になり得る、また悪用可能なセキュリティホールが時々みつかります。

現在のインターネットおよび FreeBSD のセキュリティ問題について常に最新の情報を得るために CERT および freebsd-security-notifications を購読するとよいでしょう。

ヒント

問題が生じたとしても、最新のソースからビルドした named を用意しておけば、問題にならないかもしれません。

22.11.10. さらなる情報源

BIND/named のマニュアルページ: ndc(8) named(8) named.conf(5)

- ISC Bind 公式ページ
- BIND FAQ
- O'Reilly DNS and BIND 4th Edition
- RFC1034 - Domain Names - Concepts and Facilities (ドメイン名、その概念と基盤)
- RFC1035 - Domain Names - Implementation and Specification (ドメイン名、その実装と仕様)

22.12. NTP

##: Tom Hukins.

22.12.1. 概説

時間の経過とともに、コンピュータの時計はずれてしまいがちです。 時間が経つと、コンピュータの時計は正確でなくなってゆきます。NTP (Network Time Protocol) は時計が正確であることを保証する方法の一つです。

インターネットサービスの多くは、 コンピュータの時計が正確であることに依存しているか、 あるいは多くを負っています。 たとえば web サーバは、 あるファイルがある時刻以降に修正されていたらそのファイルを送ってほしいという要求を受け取るかもしれません。cron(8) のようなサービスは所定の時間にコマンドを実行します。時計が正確でない場合、これらのコマンドは期待したとおりには実行されないかもしれません。

FreeBSD は ntpd(8) NTP サーバを搭載しています。これは、マシンの時計を合わせるために他の NTP サーバに問い合わせをしたり、他のマシンに対して時刻を報じるために使用できます。

22.12.2. 適切な NTP サーバの選択

時刻を同期するために利用する NTP サーバを、一つ以上見つける必要があります。ネットワーク管理者、または ISP はこの目的のために NTP サーバを設定しているかもしれません — 本当にそうなのか確かめるためにドキュメントを確認してください。あなたの近くの NTP サーバを探せる 公にアクセス可能な NTP サーバのリスト があります。どのサーバを選択するとしても、そのサーバの運営ポリシを理解し、要求されているなら利用許可を求めることを忘れないでください。

使用しているサーバのうちのどれかが到達不能になるか、 その時計の信頼性が低い場合、無関係の NTP サーバをいくつか選択するとよいでしょう。ntpd(8) は他のサーバから受け取った応答を賢く利用します — 信頼できないサーバより信頼できるサーバを重視します。

22.12.3. マシンの設定

22.12.3.1. 基本設定

マシンが起動するときだけ時計を同期させたい場合は ntpdate(8) が使えます。頻繁に再起動され、たまに同期すれば十分なデスクトップマシンには適切かもしれません。しかしほとんどのマシンでは ntpd(8) を実行するべきです。

ntpd(8) を動かしているマシンでも、起動時に ntpdate(8) を使用するのはよい考えです。ntpd(8) プログラムは時計を徐々に変更します。しかし ntpdate(8) は正しい時刻と現在設定されているマシンの時刻がどんなに離れていようとも時計を設定します。

起動時に ntpdate(8) を有効にするためには、 ntpdate_enable="YES" を /etc/rc.conf に追加してください。 さらに、同期したいすべてのサーバおよび、ntpdate(8) に渡すあらゆるフラグを ntpdate_flags に指定する必要があるでしょう。

22.12.3.2. 一般設定

NTP は ntp.conf(5) に記述された書式の /etc/ntp.conf ファイルによって設定されます。簡単な例を以下に示します。

```
server ntplocal.example.com prefer
server timeserver.example.org
server ntp2a.example.net

driftfile /var/db/ntp.drift
```

server オプションは、使用するサーバを一行に一つずつ指定します。サーバが上記の ntplocal.example.com のように prefer 引数とともに指定された場合、 このサーバは他のサーバより優先されます。 優先されたサーバからの応答は、 他のサーバの応答と著しく異なる場合は破棄されますが、 そうでなければ他の応答を考慮することなく使用されます。prefer 引数は、通常、特別な時間モニタハードウェアを備えているような非常に正確であるとされている NTP サーバに対して使用されます。

driftfile オプションはシステム時計の周波数オフセットを格納するために使用するファイルを指定します。ntpd(8) プログラムは、時計の自然変動を自動的に補正するためにこれを用います。これにより、一定時間外部の時刻ソースから切り離されたとしても、十分正確な時刻を維持することを可能にします。

driftfile オプションは、使用している NTP サーバから過去に受け取った応答に関する情報を格納するために、どのファイルが使用されるか指定します。このファイルは NTP に関する内部情報を含んでいます。これは他のプロセスによって修正されてはいけません。

22.12.3.3. サーバへのアクセス制御

デフォルトでは NTP サーバはインターネット上のすべてのホストからアクセスが可能です。/etc/ntp.conf 内で restrict オプションを指定することによって、どのマシンがサーバにアクセスできるかを制御できるようにします。

NTP サーバにアクセスするマシンのすべてを拒否したいのなら、 以下の行を /etc/ntp.conf に追加してください。

```
restrict default ignore
```

あなたのネットワーク内のマシンにだけサーバに接続して時計を同期することを認めたいが、それらからサーバに対して設定を行うのを許さず、同期する端末としても利用されないようにしたいのなら、以下を加えてください。

```
restrict 192.168.1.0 mask 255.255.255.0 notrust nomodify notrap
```

192.168.1.0 をあなたのネットワークの IP アドレスに 255.255.255.0 をあなたのネットワークのネットマスクに置き換えてください。

/etc/ntp.conf には複数の restrict オプションを置けます。詳細に付いては ntp.conf(5) の Access Control Support サブセクションを参照してください。

22.12.4. NTP サーバの実行

NTP サーバが起動時に実行されることを保証するために、xntpd_enable="YES" を /etc/rc.conf に加えてください。ntpd(8) にフラグを追加したい場合は /etc/rc.conf 内の xntpd_flags パラメータを編集してください。

マシンを再起動することなくサーバを実行したいときは、/etc/rc.conf 内の xntpd_flags で追加されたパラメータをすべて指定して ntpd を実行してください。以下に例を示します。

```
# ntpd -p /var/run/ntpd.pid
```

注記

FreeBSD 5.X では /etc/rc.conf 内のさまざまなオプションの名前が変わりました。したがって、上記の xntpd に関するオプションは ntpd に置き換えてください。

22.12.5. 一時的なインターネット接続で ntpd を使用する

ntpd(8) プログラムは正しく機能するために、インターネットへの常時接続を必要としません。しかしながら、オンデマンドでダイアルアップされるように設定された一時的な接続の場合、NTP トラフィックがダイアルを引き起こしたり、接続を維持し続けるようなことを避けるようにした方がよいでしょう。ユーザ PPP を使用している場合、以下の例のように /etc/ppp/ppp.conf 内で filter ディレクティブが使用できます。

```
set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
```

詳細については ppp(8) 内の PACKET FILTERING セクション、および /usr/share/examples/ppp/ 内の例を参照してください。

注記

小さい番号のポートをブロックするインターネットアクセスプロバイダでは、応答があなたのマシンに到達しないので NTP がきちんと動作しない場合もあります。

22.12.6. さらなる情報源

NTP サーバに関する文書は HTML 形式で /usr/share/doc/ntp/ にあります。

22.13. ネットワークアドレス変換 (NAT)

##: Chern Lee.

22.13.1. 概要

一般に natd(8) として知られている FreeBSD ネットワークアドレス変換デーモンは、raw IP パケットを受信して、ソースアドレスをローカルマシンに変更し、そのパケットを外向きの IP パケットの流れに再注入するデーモンです。natd(8) は、データが戻ってきたときに、データの本来の場所を判別し、 もともと要求した相手へデータを返すことができるようにソース IP アドレスとポートを変更します。

NAT の最も一般的な使用法は、 一般的にはインターネット接続共有として知られているものを実行することです。

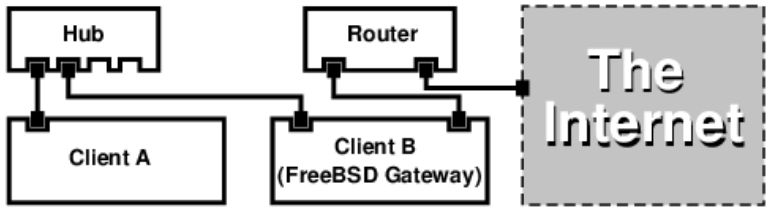
22.13.2. 設定

IPv4 の IP 空間が足りなくなりつつあること、および、ケーブルや DSL のような高速の加入者回線利用者の増加によって、 人々はますますインターネット接続を共有する手段を必要としています。一つの接続および IP アドレスを通していくつものコンピュータを回線に接続する能力がある natd(8) が合理的な選択になります。

もっともよくあるのは、ユーザが 1 つの IP アドレスでケーブルまたは DSL 回線に接続されたマシンを持っており、インターネットへのアクセスを LAN 経由でいくつかのコンピュータに提供するのに、この接続されたコンピュータを使用したいという場合です。

そのためには、インターネットに接続されている FreeBSD マシンはゲートウェイとして動作しなければなりません。このゲートウェイマシンは 2 つの NIC が必要です (1 つはイ

ンターネットルータへ接続するためで、もう 1 つは LAN に接続するためです)。LAN 上のすべてのマシンはハブまたはスイッチを通して接続されます。



インターネット接続を共有するために、このような設定がよく使用されています。LAN 内のマシンの 1 台がインターネットに接続しています。残りのマシンはその「ゲートウェイ」マシンを通してインターネットにアクセスします。

22.13.3. 設定

次のオプションがカーネルコンフィギュレーションファイルに必要です。

```
options IPFIREWALL
options IPDIVERT
```

さらに、次のオプションを入れてもよいでしょう。

```
options IPFIREWALL_DEFAULT_TO_ACCEPT
options IPFIREWALL_VERBOSE
```

下記の設定を `/etc/rc.conf` で行わなければなりません。

```
gateway_enable="YES"
firewall_enable="YES"
firewall_type="OPEN"
natd_enable="YES"
natd_interface="fxp0"
natd_flags=""
```

gateway_enable="YES"	マシンがゲートウェイとして動作するように設定します。`sysctl net.inet.ip.forwarding=1` コマンドを実行しても同じ効果がえられます。
firewall_enable="YES"	`/etc/rc.firewall` にあるファイアウォールルールを起動時に有効にします。
firewall_type="OPEN"	これはあらかじめ定義されている、 すべてのパケットを通すファイアウォールルールセットを指定します。他のタイプについては `/etc/rc.firewall` を参照してください。

natd_interface="fxp0"	パケットを転送するインタフェースを指定します (インターネットに接続されたインタフェース)。
natd_flags=""	起動時に natd(8) に渡される追加の引数

`/etc/rc.conf` に前述したオプションを定義すると、起動時に `natd -interface fxp0` が実行されます。これは手動でも実行できます。

注記

オプションの定義に natd(8) のコンフィグレーションファイルを使うこともできます。この場合には、`/etc/rc.conf` に以下の行を追加し、コンフィグレーションファイルを定義してください。

```
natd_flags="-f /etc/natd.conf"
```

`/etc/natd.conf` ファイルでは、一行ごとにオプションを設定します。たとえば、 次節の例では以下のような行を含むファイルを用意してください。

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_port tcp 192.168.0.3:80 80
```

コンフィグレーションファイルに関する、より詳細な情報については、natd(8) マニュアルページの `-f` オプションを調べてください。

LAN にぶら下がっているマシンおよびインタフェースのそれぞれには RFC 1918 で定義されているプライベートネットワーク空間の IP アドレス番号を割り当て、デフォルトゲートウェイアドレスを natd マシンの内側の IP アドレスにすべきです。

たとえば LAN 側のクライアント `A` および `B` は IP アドレス `192.168.0.2` および `192.168.0.3` を割り当てられており、natd マシンの LAN インタフェースは IP アドレス `192.168.0.1` を割り当てられています。クライアント `A` および `B` のデフォルトゲートウェイは natd マシンの `192.168.0.1` に設定されなければなりません。natd マシンの外部、またはインターネットインタフェースは natd(8) の動作に際して特別の修正を必要としません。

22.13.4. ポート転送

natd(8) の短所は、インターネットから LAN 内のクライアントにアクセスできないということです。 LAN 内のクライアントは外部に向けて接続を行うことはできますが、 入って来るものを受け取ることができません。これは、LAN クライアントのどれかでインター

ネットサービスを動かそうとした場合に、 問題になります。これを何とかする単純な方法は natd マシンから LAN クライアントへ、選択したインターネットポートを転送することです。

たとえばクライアント A で実行されている IRC サーバがあり、クライアント B 上で実行されている web サーバがあるとします。これが正しく動作するには、ポート 6667 (IRC) および 80 (web) への接続を対応するマシンに転送しなければなりません。

`-redirect_port` に適切なオプションを加えて natd(8) に渡さなければなりません。書式は以下のとおりです。

```
     -redirect_port proto targetIP:targetPORT[-targetPORT]
                    [aliasIP:]aliasPORT[-aliasPORT]
                    [remoteIP[:remotePORT[-remotePORT]]]
```

上記の例では、引数は以下のようにします。

```
    -redirect_port tcp 192.168.0.2:6667 6667
    -redirect_port tcp 192.168.0.3:80 80
```

これで適切な tcp ポートが LAN クライアントマシンに転送されます。

`-redirect_port` 引数は個々のポートを対応させるポート範囲を示すのに使えます。たとえば *`tcp 192.168.0.2:2000-3000 2000-3000`* は 2000 番から 3000 番ポートに受け取られたすべての接続を、クライアント A 上の 2000 番から 3000 番に転送します。

これらのオプションは natd(8) を直接実行するか、 `/etc/rc.conf` 内の `natd_flags=""` オプションで設定するか、もしくはコンフィグレーションファイルから渡してください。

設定オプションの詳細については natd(8) をご覧ください。

22.13.5. アドレス転送

複数の IP アドレスが利用可能ですが、それらが 1 台のマシン上になければならないときには、アドレス転送が便利です。これを用いれば natd(8) は LAN クライアントのそれぞれに外部 IP アドレスを割り当てることができます。natd(8) は LAN クライアントから外部へ出て行くパケットを適切な外部の IP アドレスで書き直し、そして特定の IP アドレスに対してやって来るトラフィックのすべてを、指定された LAN クライアントに転送します。これは静的 NAT としても知られています。たとえば `128.1.1.1`, `128.1.1.2` および `128.1.1.3` の IP アドレスが、natd ゲートウェイマシンに属しているとします。`128.1.1.2` および `128.1.1.3` は LAN クライアントの A および B に転送される一方で、`128.1.1.1` は natd ゲートウェイマシンの外部 IP アドレスとして使用することができます。

`-redirect_address` の書式は以下のとおりです。

```
-redirect_address localIP publicIP
```

localIP	LAN クライアントの内部 IP アドレス
publicIP	LAN クライアントに対応する外部 IP アドレス

上記の例では引数は以下のようになります。

```
-redirect_address 192.168.0.2 128.1.1.2
-redirect_address 192.168.0.3 128.1.1.3
```

`-redirect_port` と同様に、これらの引数は `/etc/rc.conf` 内の `natd_flags=""` オプションで設定するか、コンフィグレーションファイルから渡すことで指定できます。アドレス転送では、特定の IP アドレスで受け取られたデータはすべて転送されるので、port 転送は必要ありません。

natd マシン上の外部 IP アドレスは、アクティブで外部インタフェースにエイリアスされていなければなりません。やりかたは rc.conf(5) を参照してください。

22.14. inetd「スーパサーバ」

##: Chern Lee.

22.14.1. 概観

inetd(8) は複数のデーモンに対する接続を制御するので、「インターネットスーパサーバ」と呼ばれます。ネットワークサービスを提供するプログラムは、一般的にデーモン呼ばれます。inetd は他のデーモンを管理するサーバを努めます。接続が inetd によって受け付けられると、inetd は接続がどのデーモンに対するものか判断して、そのデーモンを起動し、ソケットを渡します。inetd を 1 つ実行することにより、それぞれのデーモンをスタンドアロンモードで実行することに比べ、全体としてのシステム負荷を減らします。

基本的に、inetd は他のデーモンを起動するために使用されます。しかし、 chargen, auth および daytime のようなささいなプロトコルは直接扱われます。

この節ではコマンドラインオプションおよび設定ファイル `/etc/inetd.conf` による inetd の設定の基本を説明します。

22.14.2. 設定

inetd は `/etc/rc.conf` の仕組によって初期化されます。 デフォルトでは `inetd_enable` オプションは 「NO」 に設定されています。 しかし多くの場合、sysinstall でセキュリティプロファイルを medium に設定することにより、有効化されます。

```
inetd_enable="YES"
```

または

```
inetd_enable="NO"
```

を /etc/rc.conf に置くことで、起動時に inetd を有効または無効にできます。

さらに inetd_flags オプションによって、 いろいろなコマンドラインオプションを inetd に渡すことができます。

22.14.3. コマンドラインオプション

inetd 書式

```
inetd [-d] [-l] [-w] [-W] [-c maximum] [-C rate] [-a address
| hostname] [-p filename] [-R rate] [configuration file]
```

-d
: デバッグモードにします。

-l
: 成功した接続のログをとります。

-w
: 外部サービスに対して TCP Wrapper を有効にします (デフォルト)。

-W
: inetd 組み込みの内部サービスに対して TCP Wrapper を有効にします (デフォルト)。

-c maximum
: サービス毎に同時に起動可能な最大値のデフォルトを指定します。 デフォルトでは無制限です。サービスごとに指定する max-child パラメータで上書きできます。

-C rate
: 1 分間にひとつの IP アドレスから起動されるサービスの、 最大値のデフォルトを指定します。デフォルトは無制限です。 サービスごとに指定する max-connections-per-ip-per-minute パラメータで上書きできます。

-R rate
: あるサービスを 1 分間に起動できる最大の数を指定します。デフォルトは 256 です。rate に 0 を指定すると、起動可能な数は無制限になります。

-a
: バインドする IP アドレスを一つ指定します。 代わりにホスト名も指定できます。この場合、ホスト名に対応する IPv4 または IPv6 アドレスが使用されます。通常 inetd

が jail(8) 内で起動される時点で、ホスト名が指定されます。この場合、ホスト名は jail(8) 環境に対応するものです。

ホスト名指定が使用され、 IPv4 および IPv6 両方にバインドしたい場合、 `/etc/inetd.conf` の各サービスに対して、各バインドに対する適切なプロトコルのエントリが必要です。たとえば TCP ベースのサービスは、ひとつはプロトコルに「tcp4」を使用し、もう一つは「tcp6」を使用する、2 つのエントリが必要です。

-p
デフォルトとは異なる PID を保持するファイルを指定します。

`/etc/rc.conf` 内の `inetd_flags` オプションを用いて、これらのオプションを inetd に渡すことができます。デフォルトでは `inetd_flags` は「-wW」に設定されており、これは inetd の内部および外部サービスに対して TCP wrapper を有効にします。初心者ユーザはこれらのパラメータを変更する必要は通常ありませんし、`/etc/rc.conf` に入力する必要もありません。

注記

外部サービスは、接続を受け取ったときに起動される inetd の外部にあるデーモンで、それに対して、内部サービスは inetd 自身が提供する内部のデーモンです。

22.14.4. `inetd.conf`

inetd の設定は `/etc/inetd.conf` ファイルによって制御されます。

`/etc/inetd.conf` が変更されたときは、以下のように inetd プロセスに HangUP シグナルを送ることにより、inetd に設定ファイルを再読み込みさせられます。

例22.4 inetd への HangUP シグナル送付

```
# kill -HUP `cat /var/run/inetd.pid`
```

設定ファイルのそれぞれの行は、個々のデーモンについての指示になります。ファイル内のコメントは「#」が先頭につきます。`/etc/inetd.conf` の書式は以下のとおりです。

```
service-name
```

```
socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute]]
user[:group][/login-class]
server-program
server-program-arguments
```

IPv4 を利用する ftpd デーモンのエントリの例です。

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd
 -l
```

service-name
: これは特定のデーモンのサービス名です。これは `/etc/services` 内のサービスリストに対応していなければなりません。これは inetd がどのポートで受け付けなければならないかを決定します。新しいサービスが作成された場合、まずはじめに `/etc/services` 内に記載しなければなりません。

socket-type
: `stream`, `dgram`, `raw` または `seqpacket` のどれかを指定します。`stream` はコネクションに基づいた TCP デーモンに使用しなければならず、一方で `dgram` は UDP 転送プロトコルを利用したデーモンに対して使用されます。

protocol
: 次のうちのどれか 1 つを指定します。

プロトコル	説明
tcp, tcp4	TCP IPv4
udp, udp4	UDP IPv4
tcp6	TCP IPv6
udp6	UDP IPv6
tcp46	TCP IPv4 および v6 の両方
udp46	UDP IPv4 および v6 の両方

{wait|nowait}[/max-child[/max-connections-per-ip-per-minute]]
: `wait|nowait` は inetd から起動したデーモンが、自分のソケットを管理できるかどうかを示します。通常マルチスレッド化されている stream ソケットデーモンは `nowait` を使用するべきである一方、`dgram` ソケットタイプは wait オプションを使用しなければなりません。`nowait` は新しいソケット毎に子のデーモンを起動する一方で、`wait` は通常複数のソケットを 1 つのデーモンに渡します。

  inetd が起動できる子のデーモンの最大数は `max-child` オプションで設定できます。特定のデーモンに対して、起動する数が 10 までという制限が必要な場合、`nowait` の後に `/10` を置きます。

max-child に加えて、他にある 1 つの場所から特定のデーモンへの最大接続数を制限するオプションが利用できます。 max-connections-per-ip-per-minute がそれです。ここに 10 を指定すると、特定の IP アドレスからの特定のサービスへの接続を 1 分間につき 10 回に制限します。これは故意または故意でない資源の浪費および、マシンへのサービス不能 (DoS) 攻撃を防ぐのに有用です。

wait または nowait はこの欄に必ず必要です。max-child および max-connections-per-ip-per-minute は任意です。

max-child または max-connections-per-ip-per-minute 制限をかけない stream タイプのマルチスレッドデーモンの設定は nowait になります。

作成できる子プロセスの上限が 10 である同じデーモンの設定は nowait/10 になります。

さらに、1 分間に IP アドレスあたりの接続制限が 20、子プロセスの上限が 10 である同じデーモンの設定は nowait/10/20 になります。

以下のように、これらのオプションはすべて fingerd デーモンのデフォルト設定に使われています。

```
finger stream  tcp     nowait/3/10 nobody /usr/libexec/fingerd ↺
fingerd -s
```

user
: user はあるデーモンが実行するときのユーザ名を指定します。 一般的にデーモンは root ユーザとして実行します。セキュリティを考慮して、 いくつかのサーバは daemon ユーザ、または最低の権限が与えられている nobody ユーザとして実行することも多く見られます。

server-program
: 接続を受け取ったときに実行するデーモンのフルパスです。 デーモンが inetd によって内部的に提供されるサービスの場合 internal を使用します。

server-program-arguments
: ここには、起動するときにデーモンに渡される、argv[0] から始まる引数を指定して、server-program と協調して動作します。mydaemon -d がコマンドラインの場合、server program arguments の値に mydaemon -d を指定します。また、デーモンが内部サービスの場合、ここに internal を指定します。

22.14.5. セキュリティ

インストールの時に選択したセキュリティプロファイルによっては、 多くの inetd のデーモンがデフォルトで有効になっているかもしれません。あるデーモンが特に必要でない場合には、それを無効にしてください! 問題となっているデーモンが記述されている行の先

頭に「#」をおいて inetd にハングアップシグナルを送ってください。fingerd のようないくつかのデーモンは、動かそうとすべきではないかもしれません。なぜなら、それらは攻撃者に対してあまりにも多くの情報を与えるからです。

セキュリティをあまり考慮せず、接続試行に対してタイムアウトまでの時間が長いか、タイムアウトしないデーモンもあります。これは、特定のデーモンに攻撃者がゆっくり接続要求を送ることによって、利用可能なリソースを飽和させることを可能にします。ある種のデーモンに `ip-per-minute` および `max-child` 制限を設けることはよい考えかもしれません。

TCP wrapper はデフォルトで有効です。inetd から起動されるさまざまなデーモンに対して TCP 制限を設けることの詳細については hosts_access(5) マニュアルページを参照してください。

22.14.6. その他

daytime, time, echo, discard, chargen および auth はすべて inetd が内部的に提供するサービスです。

auth サービスは identity (ident, identd) ネットワークサービスを提供し、ある程度設定可能です。

詳細については inetd(8) マニュアルを参照してください。

22.15. パラレルライン IP (PLIP)

PLIP はパラレルポート間で TCP/IP 通信を可能にします。これはネットワークカードの無いマシンやノートパソコンにインストールするときに役に立ちます。この節では以下について説明します。

- パラレル (ラップリンク または パラレルクロス) ケーブルの作成。
- 2 台のコンピュータの PLIP による接続。

22.15.1. パラレル (クロス) ケーブルの作成

コンピュータ用品店のほとんどでパラレル (クロス) ケーブルを購入することができます。購入することができないか、単にケーブルがどのような構造であるか知りたい場合は、次の表に通常のパラレルプリンタケーブルをもとに作成する方法が示されています。

表22.1 ネットワーク向けのパラレル (クロス) ケーブル結線

A-名称	A-端	B-端	説明	Post/Bit
DATA0	2	15	Data	0/0x01

A-名称	A-端	B-端	説明	Post/Bit
-ERROR	15	2		1/0x08
DATA1 +SLCT	3 13	13 3	Data	0/0x02 1/0x10
DATA2 +PE	4 12	12 4	Data	0/0x04 1/0x20
DATA3 -ACK	5 10	10 5	Strobe	0/0x08 1/0x40
DATA4 BUSY	6 11	11 6	Data	0/0x10 1/0x80
GND	18-25	18-25	GND	-

22.15.2. PLIP の設定

はじめに、ラップリンクケーブルを入手しなければなりません。 次に、両方のコンピュータのカーネルが lpt(4) ドライバ対応であることを確認してください。

```
# grep lp /var/run/dmesg.boot
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
```

パラレルポートは割り込み駆動ポートでなければなりません。FreeBSD 4.X では、以下のような行がカーネルコンフィギュレーションファイル内になければならないでしょう。

```
device ppc0 at isa? irq 7
```

FreeBSD 5.X では `/boot/device.hints` ファイルに以下の行がなければならないでしょう。

```
hint.ppc.0.at="isa"
hint.ppc.0.irq="7"
```

それからカーネルコンフィギュレーションファイルに `device plip` という行があるか、または `plip.ko` カーネルモジュールが読み込まれていることを確認してください。どちらの場合でも ifconfig(8) コマンドを直接実行したときに、パラレルネットワークインタフェースが現れるはずです。FreeBSD 4.X ではこのようになります。

```
# ifconfig lp0
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

FreeBSD 5.X ではこのようになります。

```
# ifconfig plip0
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

注記

パラレルインタフェースに対して用いられるデバイス名は FreeBSD 4.X (`lpX`) と FreeBSD 5.X (`plipX`) 間で異なります。

両方のコンピュータのパラレルインタフェースにラップリンクケーブルを接続します。

両方のネットワークインタフェースパラメータを `root` で設定します。 たとえば、FreeBSD 4.X を動作させている `host1` と FreeBSD 5.X を動作させている `host2` の両ホストを接続したい場合は次のようにします。

```
                host1 <-----> host2
IP Address    10.0.0.1      10.0.0.2
```

次のコマンドで `host1` 上のインタフェースを設定します。

```
# ifconfig lp0 10.0.0.1 10.0.0.2
```

次のコマンドで `host2` 上のインタフェースを設定します。

```
# ifconfig plip0 10.0.0.2 10.0.0.1
```

さて、これで接続が確立したはずです。詳細については lp(4) および lpt(4) マニュアルページをご覧ください。

さらに`/etc/hosts` に両ホストを加えるとよいでしょう。

```
127.0.0.1               localhost.my.domain localhost
10.0.0.1                host1.my.domain host1
10.0.0.2                host2.my.domain
```

接続がうまくいっているか確かめるために、両方のホスト上で互いを ping してください。たとえば `host1` で以下を実行します。

```
# ifconfig lp0
lp0: flags=8851<UP,POINTOPOINT,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 10.0.0.1 --> 10.0.0.2 netmask 0xff000000
# netstat -r
Routing tables

Internet:
Destination        Gateway           Flags     Refs     Use      ↺
Netif Expire
host2              host1               UH          0       0       lp0
```

```
# ping -c 4 host2
PING host2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=255 time=2.774 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=255 time=2.530 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=255 time=2.556 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=255 time=2.714 ms

--- host2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.530/2.643/2.774/0.103 ms
```

22.16. IPv6

##: Aaron Kaplan.
######: Tom Rhodes.

IPv6 (IPng「IP next generation」とも呼ばれます) は、著名な IP プロトコル (IPv4 とも呼ばれます) の新しいバージョンです。他の最新の *BSD システムと同様に FreeBSD は KAME IPv6 リファレンス実装を含んでいます。したがって、あなたの FreeBSD システムには IPv6を試すために必要なものすべてが備わっています。この節では IPv6 の設定と実行に関して説明します。

1990 年代のはじめには、人々は IPv4 アドレス空間が急速に縮小していることに気づくようになりました。インターネットの成長率が増大するにしたがって、2 つの心配ごとがでてきました。

- アドレスの枯渇。今日では、プライベートアドレス空間 (`10.0.0.0/8`, `192.168.0.0/24` など) およびネットワークアドレス変換 (NAT) が使用されているので、それほど心配されていません。

- ルーティングテーブルのエントリが大きくなりすぎていました。これは今でも心配な事柄です。

IPv6 は以下の、そしてその他多くの問題を扱います。

- 128 bit アドレス空間。言い換えると、理論上 340,282,366,920,938,463,463,374,607,431,768,211,456 個のアドレスが利用可能です。これは地球上の一平方メータあたり、およそ 6.67 * 10^27 個の IPv6 アドレスがあることを意味します。

- ルータは、ルーティングテーブル内にネットワーク集約アドレスだけを格納することで、ルーティングテーブルの平均を 8192 項目程度に減らします。

他にも以下のように IPv6 の便利な機能がたくさんあります。

- アドレス自動設定 (RFC2462)

- エニーキャスト (anycast) アドレス (「one-out-of many」訳注: 複数の異なるノードが応答する 1 つのアドレス。RFC2526 を参照してください)。
- 強制マルチキャストアドレス
- IPsec (IP セキュリティ)
- シンプルなヘッダ構造
- モバイル IP
- IPv4 から IPv6 への移行手段

詳細については下記を参照してください。

- Sun.com の IPv6 概観
- IPv6.org
- KAME.net
- 6bone.net

22.16.1. IPv6 アドレスの背景

いくつか違うタイプの IPv6 アドレスがあります。ユニキャスト (Unicast)、エニーキャスト (Anycast) およびマルチキャスト (Multicast) です。

ユニキャストアドレスは周知のアドレスです。ユニキャストアドレスへ送られたパケットは、まさにそのアドレスに属するインターフェースに到着します。

エニーキャストアドレスはユニキャストアドレスと構文上判別不可能ですが、インタフェース群に宛てられています。エニーキャストアドレスに送られたパケットは (ルータメトリック的に) 最も近いインタフェースに到着します。エニーキャストアドレスはルータでしか使ってはいけません。

マルチキャストアドレスはインタフェース群を識別します。マルチキャストアドレスに送られたパケットは、マルチキャスト群に属するすべてのインタフェースに到着します。

注記

IPv4 のブロードキャストアドレス (通常 `xxx.xxx.xxx.255`) は、IPv6 ではマルチキャストアドレスで表現されます。

表22.2 予約された IPv6 アドレス

IPv6 アドレス	プレフィックス長 (ビット)	説明	備考
`::`	128 ビット	不特定	IPv4 の `0.0.0.0` 参照
`::1`	128 ビット	ループバックアドレス	IPv4 の `127.0.0.1` 参照
`::00:xx:xx:xx:xx`	96 ビット	IPv4 埋め込みアドレス	下位の 32 ビットは IPv4 アドレスです。「IPv4 互換 IPv6 アドレス」とも呼ばれます。
`::ff:xx:xx:xx:xx`	96 ビット	IPv4 射影 IPv6 アドレス	下位の 32 ビットは IPv4 アドレスです。IPv6 に対応していないホストに対するアドレスです。
`fe80::` - `feb::`	10 ビット	リンクローカル	IPv4 のループバックアドレス参照
`fec0::` - `fef::`	10 ビット	サイトローカル	
`ff::`	8 ビット	マルチキャスト	
`001` (基数 2)	3 ビット	グローバルユニキャスト	すべてのグローバルユニキャストアドレスはこのプールから割り当てられます。 はじめの 3 ビットは「001」です。

22.16.2. IPv6 アドレスを読む

正規の書式では `x:x:x:x:x:x:x:x` と表されます。それぞれの「x」は 16 ビットの 16 進数です。たとえば `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982` となります。

すべてゼロの長い部分文字列がアドレス内によく現れます。 そのため、そのような部分文字列は 「::」 に短縮することができます。 たとえば、`fe80::1` は正規形の `fe80:0000:0000:0000:0000:0000:0000:0001` に対応します。

3 番目の形式は、最後の 32 ビットの部分を「.」を分割文字として使う、なじみ深い IPv4 (10 進) 形式で書くことです。たとえば `2002::10.0.0.1` は (16 進) 正

規形の `2002:0000:0000:0000:0000:0000:0a00:0001` に対応し、同時に `2002::a00:1` と書くこととも等価です。

ここまで来れば、下記を理解することができるでしょう。

```
# ifconfig
```

```
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu ↺
1500
        inet 10.0.0.10 netmask 0xffffff00 broadcast 10.0.0.255
        inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
        ether 00:00:21:03:08:e1
        media: Ethernet autoselect (100baseTX )
        status: active
```

`fe80::200:21ff:fe03:8e1%rl0` は自動的に設定されたリンクローカルアドレスです。これは自動設定の一環として、イーサネット MAC アドレスを変換したものを含んでいます。

IPv6 アドレス構造についての詳細は RFC3513 をご覧ください。

22.16.3. 接続

現在、他の IPv6 ホストおよびネットワークに接続するためには 4 つの方法があります。

- 6bone 実験ネットワークに参加する。
- 上流のプロバイダから IPv6 ネットワークの割り当てを受ける。 手順については、インターネットプロバイダに問い合わせてください。
- IPv6 over IPv4 によるトンネル。
- ダイアルアップ接続の場合 freenet6 port を使用する。

ここでは、現在もっともよく使われている方法と思われる 6bone へ接続する方法を説明します。

はじめに 6bone サイトをみて、あなたに最も近い 6bone 接続先を見つけてください。責任者に連絡すると、少しばかり運がよければ、 接続を設定する方法についての指示を受けられるでしょう。多くのばあい、これには GRE (gif) トンネルの設定が含まれます。

訳注

6bone は `3ffe::` (16 ビット) という IPv6 アドレスを割り振られた実験目的のネットワークでしたが、2006 年 6 月に運用を停

止することになっています。他の商用や試験的な IPv6 接続サービスを探してください。

ここに gif(4) トンネルを設定する典型的な例を示します。

```
# ifconfig gif0 create
# ifconfig gif0
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
# ifconfig gif0 tunnel MY_IPv4_ADDR  HIS_IPv4_ADDR
# ifconfig gif0 inet6 alias ↺
MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR
```

大文字になっている単語を、上流の 6bone ノードから受け取った情報に置き換えてください。

これでトンネルが確立されます。ping6(8) を `ff02::1%gif0` に送ることによって、トンネルが動作しているか確かめてください。ping の応答を 2 つ受け取るはずです。

注記

`ff02:1%gif0` というアドレスに興味をそそられている場合のために説明すると、これはマルチキャストアドレスです。`%gif0` は、ネットワークインタフェース `gif0` 上のマルチキャストアドレスが使用されるということを示しています。 マルチキャストアドレスに対して `ping` を送ったので、トンネルのもう一方の端も応答します。

ここまで来ると 6bone アップリンクに経路設定することは比較的簡単でしょう。

```
# route add -inet6 default -interface gif0
# ping6 -n MY_UPLINK
```

```
# traceroute6 www.jp.FreeBSD.org
(3ffe:505:2008:1:2a0:24ff:fe57:e561) from 3ffe:8060:100::40:2, 30 ↺
hops max, 12 byte packets
     1  atnet-meta6  14.147 ms  15.499 ms  24.319 ms
     2  6bone-gw2-ATNET-NT.ipv6.tilab.com  103.408 ms  95.072 ms *
     3  3ffe:1831:0:ffff::4  138.645 ms  134.437 ms  144.257 ms
     4  3ffe:1810:0:6:290:27ff:fe79:7677  282.975 ms  278.666 ms  ↺
292.811 ms
     5  3ffe:1800:0:ff00::4  400.131 ms  396.324 ms  394.769 ms
     6  3ffe:1800:0:3:290:27ff:fe14:cdee  394.712 ms  397.19 ms  ↺
394.102 ms
```

この出力はマシンによって異なります。これで、あなたが www/mozilla のような IPv6 が利用可能なブラウザを持っていれば、IPv6 サイト www.kame.net にいって踊るカメを見ることができるでしょう。

22.16.4. IPv6 世界の DNS

IPv6 のための新しい DNS レコードが 2 種類あります。

- AAAA レコード
- A6 レコード

AAAA レコードは簡単に使えます。

```
MYHOSTNAME              AAAA     MYIPv6ADDR
```

上記をプライマリゾーン DNS ファイルに加えて、もらったばかりの IPv6 アドレスにホスト名を割り当ててください。あなた自身で DNS ゾーンを管理していない場合は、DNS プロバイダに頼んでください。bind の最新バージョン (バージョン 8.3 および 9) は AAAA レコードに対応しています。

パート V. 付録

目次

A. FreeBSD の入手方法 813
A.1. CD および DVD セット 813
A.2. FTP サイト 814
A.3. CTM を使う 823
A.4. Subversion を使う 826
A.5. rsync を使用する 830
B. 参考図書 833
B.1. FreeBSD 専門の書籍 833
B.2. 利用者向けのガイド 834
B.3. 管理者向けのガイド 835
B.4. プログラマ向けのガイド 835
B.5. オペレーティングシステム内部 836
B.6. セキュリティの参考資料 837
B.7. ハードウェアの参考資料 837
B.8. UNIX® の歴史 838
B.9. 定期刊行物、雑誌およびジャーナル 839
C. インターネット上のリソース 841
C.1. メーリングリスト 841
C.2. Usenet ニュースグループ 857
C.3. World Wide Web サイト 858
C.4. Email アドレス 861
C.5. シェルアカウント 861
D. PGP 公開鍵 863
D.1. オフィサ 863

付録A FreeBSD の入手方法

A.1. CD および DVD セット

FreeBSD の CD および DVD のセットは以下のオンライン業者から入手できます。

- FreeBSD Mall, Inc.
  2420 Sand Creek Rd C-1 #347
  Brentwood, CA
  94513
  USA
  電話: +1 925 240-6652
  Fax: +1 925 674-0821
  Email: <info@freebsdmall.com >
  WWW: http://www.freebsdmall.com/

- Getlinux
  78 Rue de la Croix Rochopt
  Épinay-sous-Sénart
  91860
  France
  Email: <contact@getlinux.fr >
  WWW: http://www.getlinux.fr/

- Dr. Hinner EDV
  Kochelseestr. 11
  D-81371 München
  Germany
  電話: (0177) 428 419 0
  Email: <infow@hinner.de >
  WWW: http://www.hinner.de/linux/freebsd.html

- Linux Center
  Galernaya Street, 55
  Saint-Petersburg
  190000
  Russia
  電話: +7-812-309-06-86
  Email: <info@linuxcenter.ru >

WWW: http://linuxcenter.ru/shop/freebsd

A.2. FTP サイト

FreeBSD の公式な情報は anonymous FTP によって世界中のミラーサイトより入手できます。`ftp://ftp.FreeBSD.org/pub/FreeBSD/` サイトは接続が良く、大規模な接続を受け付けていますが、「よりネットワーク的に近い」ミラーサイトを探した方が良いでしょう (特にミラーサイトのようなものを構築しようとした場合はこれに該当します)。

さらに、FreeBSD は以下のミラーサイトから anonymous FTP によって入手できます。もし FreeBSD を anonymous FTP によって手にいれる場合は、 近くのサイトを利用するようにしてください。「一次ミラーサイト」 としてあげられているサイトには、FreeBSD の各アーキテクチャで利用可能なすべてのバージョンのアーカイブ一式が用意されています。あなたが住んでいる国や地域には、より高速にダウンロードできるサイトがおそらくあるでしょう。各国のミラーサイトには、人気のあるアーキテクチャの最新のバージョンが置いてありますが、FreeBSD のアーカイブ全体はもしかするとないかもしれません。すべてのサイトは anonymous FTP 別の方法によるアクセスを提供しているサイトもあります。各サイトで提供しているアクセス方法は、ホスト名に続く括弧の中に記載されています。

中央サーバ, 一次ミラーサイト, アイルランド, アメリカ合衆国, アルメニア, イギリス, ウクライナ, エストニア, オーストラリア, オーストリア, オランダ, カナダ, ギリシア, サウジアラビア, スイス, スウェーデン, スペイン, スロベニア, チェコ共和国, デンマーク, ドイツ, ニュージーランド, ノルウェー, フィンランド, フランス, ブラジル, ポーランド, ラトビア, リトアニア, ロシア, 韓国, 香港, 台湾, 中国, 南アフリカ, 日本.

(UTC 現在)

中央サーバ

- ftp://ftp.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.FreeBSD.org/pub/FreeBSD/ / http://ftp.FreeBSD.org/pub/FreeBSD/)

一次ミラーサイト
問題がある場合は、このドメインのホストマスタ
<mirror-admin@FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp1.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp4.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.FreeBSD.org/pub/FreeBSD/ / http://ftp4.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp10.FreeBSD.org/pub/FreeBSD/ / http://ftp10.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp11.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.FreeBSD.org/pub/FreeBSD/)

アイルランド
問題がある場合は、このドメインのホストマスタ <hostmaster@ie.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp3.ie.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

アメリカ合衆国
問題がある場合は、このドメインのホストマスタ <hostmaster@us.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp1.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.us.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.us.FreeBSD.org/pub/FreeBSD/ / http://ftp4.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.us.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp11.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp13.us.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp14.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp15.us.FreeBSD.org/pub/FreeBSD/ (ftp)

アルメニア

問題がある場合は、このドメインのホストマスタ <hostmaster@am.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp1.am.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.am.FreeBSD.org/pub/FreeBSD/ / rsync)

イギリス

問題がある場合は、このドメインのホストマスタ <hostmaster@uk.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.uk.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.uk.FreeBSD.org/ftp.freebsd.org/pub/FreeBSD/)
- ftp://ftp3.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.uk.FreeBSD.org/pub/FreeBSD/ (ftp)

ウクライナ

- ftp://ftp.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ua.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.ua.FreeBSD.org/pub/FreeBSD / rsync://ftp6.ua.FreeBSD.org/FreeBSD/)
- ftp://ftp7.ua.FreeBSD.org/pub/FreeBSD/ (ftp)

エストニア

問題がある場合は、このドメインのホストマスタ <hostmaster@ee.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.ee.FreeBSD.org/pub/FreeBSD/ (ftp)

オーストラリア
問題がある場合は、このドメインのホストマスタ <hostmaster@au.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.au.FreeBSD.org/pub/FreeBSD/ (ftp)

オーストリア
問題がある場合は、このドメインのホストマスタ <hostmaster@at.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.at.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.at.FreeBSD.org/pub/FreeBSD/ / http://ftp.at.FreeBSD.org/pub/FreeBSD/)

オランダ
問題がある場合は、このドメインのホストマスタ <hostmaster@nl.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.nl.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nl.FreeBSD.org/os/FreeBSD/ / rsync)
- ftp://ftp2.nl.FreeBSD.org/pub/FreeBSD/ (ftp)

カナダ
問題がある場合は、このドメインのホストマスタ <hostmaster@ca.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.ca.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.ca.FreeBSD.org/pub/FreeBSD/ (ftp)

ギリシア
問題がある場合は、このドメインのホストマスタ <hostmaster@gr.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.gr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.gr.FreeBSD.org/pub/FreeBSD/ (ftp)

サウジアラビア
問題がある場合は、このドメインのホストマスタ <ftpadmin@isu.net.sa > 宛にお問い合わせください。

- ftp://ftp.isu.net.sa/pub/ftp.freebsd.org/ (ftp)

スイス
問題がある場合は、このドメインのホストマスタ
<hostmaster@ch.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ch.FreeBSD.org/pub/FreeBSD/)

スウェーデン
問題がある場合は、このドメインのホストマスタ
<hostmaster@se.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.se.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.se.FreeBSD.org/)
- ftp://ftp3.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.se.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.se.FreeBSD.org/pub/FreeBSD/)

スペイン
問題がある場合は、このドメインのホストマスタ
<hostmaster@es.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.es.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.es.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp3.es.FreeBSD.org/pub/FreeBSD/ (ftp)

スロベニア
問題がある場合は、このドメインのホストマスタ
<hostmaster@si.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.si.FreeBSD.org/pub/FreeBSD/ (ftp)

チェコ共和国
問題がある場合は、このドメインのホストマスタ
<hostmaster@cz.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/

FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)

- ftp://ftp2.cz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.cz.FreeBSD.org/pub/FreeBSD/)

デンマーク

問題がある場合は、このドメインのホストマスタ <hostmaster@dk.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.dk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.dk.FreeBSD.org/pub/FreeBSD/ / http://ftp.dk.FreeBSD.org/pub/FreeBSD/)

ドイツ

問題がある場合は、このドメインのホストマスタ <de-bsd-hubs@de.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.de.FreeBSD.org/freebsd/ (ftp / http://www1.de.FreeBSD.org/freebsd/ / rsync://rsync3.de.FreeBSD.org/freebsd/)
- ftp://ftp2.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.de.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp4.de.FreeBSD.org/FreeBSD/ (ftp / http://ftp4.de.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp7.de.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp8.de.FreeBSD.org/pub/FreeBSD/ (ftp)

ニュージーランド

- ftp://ftp.nz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nz.FreeBSD.org/pub/FreeBSD/)

ノルウェー

問題がある場合は、このドメインのホストマスタ <hostmaster@no.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.no.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

フィンランド
問題がある場合は、このドメインのホストマスタ
<hostmaster@fi.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.fi.FreeBSD.org/pub/FreeBSD/ (ftp)

フランス
問題がある場合は、このドメインのホストマスタ
<hostmaster@fr.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.fr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.fr.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp3.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp7.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.fr.FreeBSD.org/pub/FreeBSD/ (ftp)

ブラジル
問題がある場合は、このドメインのホストマスタ
<hostmaster@br.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp2.br.FreeBSD.org/FreeBSD/ (ftp / http://ftp2.br.FreeBSD.org/)
- ftp://ftp3.br.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp4.br.FreeBSD.org/pub/FreeBSD/ (ftp)

ポーランド
問題がある場合は、このドメインのホストマスタ
<hostmaster@pl.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp2.pl.FreeBSD.org

ラトビア
問題がある場合は、このドメインのホストマスタ
<hostmaster@lv.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.lv.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lv.FreeBSD.org/pub/FreeBSD/)

リトアニア

問題がある場合は、このドメインのホストマスタ <hostmaster@lt.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.lt.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lt.FreeBSD.org/pub/FreeBSD/)

ロシア

問題がある場合は、このドメインのホストマスタ <hostmaster@ru.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ru.FreeBSD.org/FreeBSD/ / rsync)
- ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp4.ru.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp5.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp6.ru.FreeBSD.org/pub/FreeBSD/ (ftp)

韓国

問題がある場合は、このドメインのホストマスタ <hostmaster@kr.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.kr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp2.kr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.kr.FreeBSD.org/pub/FreeBSD/)

香港

- ftp://ftp.hk.FreeBSD.org/pub/FreeBSD/ (ftp)

台湾

問題がある場合は、このドメインのホストマスタ <hostmaster@tw.FreeBSD.org> 宛にお問い合わせください。

- ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)

- ftp://ftp4.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.tw.FreeBSD.org/ / rsync)
- ftp://ftp7.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp11.tw.FreeBSD.org/FreeBSD/)
- ftp://ftp12.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp15.tw.FreeBSD.org/pub/FreeBSD/ (ftp)

中国

問題がある場合は、このドメインのホストマスタ <hostmaster@cn.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.cn.FreeBSD.org/pub/FreeBSD/ (ftp)

南アフリカ

問題がある場合は、このドメインのホストマスタ <hostmaster@za.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.za.FreeBSD.org/pub/FreeBSD/ (ftp)

日本

問題がある場合は、このドメインのホストマスタ <hostmaster@jp.FreeBSD.org > 宛にお問い合わせください。

- ftp://ftp.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp6.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

A.3. CTM を使う

訳: 花井 浩之 **<hanai@FreeBSD.org >**、1997 年 9 月 13 日

CTM は、 リモートのディレクトリツリーを中央のツリーに同期させるための手段です。FreeBSD システムに組み込まれており、FreeBSD のソースリポジトリとの同期に使うことができます。 リポジトリ全体の同期や特定のブランチセットの同期に用いることができます。

CTM は、 特にお粗末な TCP/IP 接続しか持っていなかったり、 TCP/IP 接続が行なえない環境での利用を想定して設計されています。変更点を自動的にメールで送られてくるように設定することができます。アクティブなブランチでは 1 日に最大三つまでのデルタを受け取る必要があります。デルタのサイズは常にできるだけ小さく保たれています。大抵の場合 5KB よりも小さく、たまに (10 回に 1 回程度) 10-50KB になり、ときおり 100KB 以上のサイズになります。

CTM を使って FreeBSD の開発を追いかける時には、あらかじめパッケージ化されたリリースとは異なり、開発ソースの作業ディレクトリに関連した注意点に気をつけてください。これに関しては、開発ブランチを追いかける の章を参照してください。

デルタを作り出す過程や CTM を他の目的に使うためのドキュメントは現在ほとんどありません。CTM に関連した質問については、ctm-users メーリングリストに相談してください。

A.3.1. デルタを入手する

CTM が用いる「デルタ」は、anonymous FTP もしくはメールで入手できます。

FTP のデルタは以下のミラーサイトから入手できます。もし anonymous FTP を使って CTM のデルタを手にいれる場合は、 地理的に近くにあるミラーサイトを利用するようにしてください。何か問題がある場合は、ctm-users メーリングリストに相談してください。

カリフォルニア、サンフランシスコ近辺、公式なソース

- `ftp://ftp.FreeBSD.org/pub/FreeBSD/development/CTM/`
- `ftp://ftp.FreeBSD.org/pub/FreeBSD/CTM/`

南アフリカ、古い差分ファイルのバックアップサーバ

- `ftp://ftp.za.FreeBSD.org/pub/FreeBSD/CTM/`

台湾/中華民国

- `ftp://ctm.tw.FreeBSD.org/pub/FreeBSD/development/CTM/`
- `ftp://ctm2.tw.FreeBSD.org/pub/FreeBSD/development/CTM/`
- `ftp://ctm3.tw.FreeBSD.org/pub/FreeBSD/development/CTM/`

メールでデルタを得たいという場合は、`http://lists.freebsd.org/mailman/listinfo` から、`ctm-src` 配布メーリングリストのいずれかに参加してください。たとえば ctm-src-cur は head 開発ブランチに対応しています。ctm-src-9 は 9.X リリースのブランチに対応したものです

メールで CTM の更新ファイルを受け取り始めたら、 中身を取り出して使用するために `ctm_rmail` を使ってください。 このプロセスを自動で行ないたいのであれば、 このコマンドを `/etc/aliases` から直接実行することもできます。 さらに詳しいことは ctm_rmail(1) をご覧ください。

注記

どの方法を使ってデルタを入手していたとしても、CTM のユーザは ctm-announce メーリングリストを購読するようにしてください。このメーリングリストは、CTM に関するアナウンスが投稿される唯一の場所です。

A.3.2. CTM の使い方

初めて CTM デルタを使う前に、出発点を作成しておく必要があります。

その方法の一つは、空のディレクトリに対して、「スタータ」 デルタを使う方法です。 スタータデルタは `src-cur.3210XEmpty.gz` のように、`XEmpty` というファイル名になります。X の後ろは最初の開始ポイントに対応し、`Empty` は空のディレクトリを表します。ルールとして `Empty` からの移行デルタは 100 デルタごとに作られます。スタータデルタである `XEmpty` のデルタは、70 から 80MB の `gzip` で圧縮された大きなデータということに注意してください。

もう一つの方法は、RELEASE　のメディアからソースをコピーしたり展開する方法です。この方法では、インターネット上で大きなデータの転送を行う必要がありません。

一度ベースとなるデルタを作成したら、 それに続く大きな数字を持つ多数のすべてのデルタを適用してください。デルタを適用するためには、以下のように実行してください。

```
# cd /directory/to/store/the/stuff
# ctm -v -v /directory/which/stores/the/deltas/src-xxx.*
```

一度に適用されるデルタは一つで、 一度適用されたデルタは無視されるので、 複数のデルタを一つのコマンドで適用することができます。CTM は `gzip` で圧縮されているデルタを理解するので、ディスクの節約にもなります。

デルタを適用せずに確認だけを行うには、コマンドラインに `-c` フラグを加えてください。CTM はローカルのツリーに対して実際には何も行ないません。 デルタの完全性を確認し、 現在のツリーに問題なく使用できるかを確認します。 オプションに関するより多くの情報や、デルタを適用する時に CTM が用いるプロセスの全体については、ctm(1) をご覧ください。

ローカルのソースツリーを最新に保つには、　　新しいデルタが入手可能になるたびに、CTM に適用してください。

デルタを適用後、もしデルタを再ダウンロードするのが骨の折れる作業であれば、デルタを消さないでおいてください。 このようにすると、将来のディザスタリカバリで必要になった場合に、ローカルにあるコピーを利用できます。

A.3.3. ローカルの変更を保存する

開発者はソースツリー中のファイルを使って、実験したり変更したくなるものです。CTM はローカルの変更を制限つきでサポートします。 ファイルの存在をチェックする前に、同じ名前で `.ctm` という拡張子のついたファイルを参照しにいきます。このファイルが存在する場合、CTM はオリジナルファイルの代りにこのファイルを処理します。

この動作はローカルの変更を保持する簡単な手段を提供します。 ファイルを変更する前に、`.ctm` という拡張子のファイル名にコピーしてください。 あとは自由にオリジナルの名前を持つファイルを変更してください。 CTM は、`.ctm` の拡張子を持つファイルにのみに適用されます。

A.3.4. CTM のその他のオプション

更新で変更されるファイルを正確に知る
: CTM がローカルのソースリポジトリに対して、どのような変更を行うかを知るには、`-l` オプションを使ってください。このオプションは、変更のログを作成したり、変更されたファイルに対して、なんらかの方法で 前・後処理したい場合には、役に立つでしょう。

更新前にバックアップを取る
: CTM の更新によって変更されるファイルすべてのバックアップをとるには、`-B` *`backup-file`* オプションを指定してください。このオプションを使うと、CTM は、デルタで変更されるファイルすべてを `backup-file` としてバックアップするようになります。

更新で変更されるファイルを制限する
: CTM の更新の範囲を制限したり一連のデルタのから ほんの数ファイルを抽出するには、使用するファイルを指定する `-e` と どのファイルを無視するかを指定する `-x` オプションを使って、正規表現で指定できます。

  例えば、`lib/libc/Makefile` の最新のコピーを保存してある CTM デルタのコレクションから抽出するには、以下のコマンドを実行してください。

```
# cd /directory/to/extract/to/
# ctm -e '^lib/libc/Makefile' /directory/which/
stores/the/deltas/src-xxx.*
```

  CTM デルタで指定されたファイルごとに、`-e` そして `-x` オプションがコマンドラインで指定された順序で適用されます。すべての `-e` そして `-x` オプションが適用された後に更新対象と選択された場合に限り、CTM はそのファイルを処理します。

A.4. Subversion を使う

A.4.1. はじめに

2012 年 7 月から、FreeBSD はすべてのソースコード、ドキュメント、Ports Collection を管理するメインのバージョン管理システムに Subversion を使っています。

注記

一般的には Subversion は開発者向けのツールです。 大部分のユーザは、FreeBSD のベースシステムのアップデートに `freebsd-update` (「FreeBSD Update」)、Ports Collection のアップデートには `portsnap` (「Ports Collection の利用」) を使うべきでしょう。

この章では、FreeBSD システムへの Subversion のインストール方法および、FreeBSD リポジトリをローカルに準備する方法について説明します。また、利用可能な FreeBSD Subversion ミラーの一覧および、Subversion を利用するための情報についても紹介します。

A.4.2. インストール

リポジトリの中身をチェックアウトできるよう、事前に Subversion をインストールしておく必要があります。 もし、すでに ports ツリーが用意してあれば、 以下のようにして Subversion をインストールできます。

```
# cd /usr/ports/devel/subversion
# make install clean
```

もし ports ツリーを利用できなければ、package から Subversion をインストールできます。

```
# pkg install devel/subversion
```

A.4.3. Subversion の実行

svn コマンドを使って、必要なリポジトリのソースコードを以下のようにしてダウンロードします。このディレクトリにあるファイルを ローカル作業コピー と呼びます。

警告

checkout を使う前に、ローカルディレクトリを移動するか削除してください。

svn 以外の方法で用意されたディレクトリでチェックアウトすると、すでに存在するファイルと、 リポジトリから持ってきたファイルとの間で衝突が起きてしまいます。

Subversion では、リポジトリの指定に *protocol://hostname/path* 形式の URL を用います。以下に記載されているように、ミラーサイトは異なる複数のプロトコルに対応しています。アクセスする FreeBSD リポジトリは、パス (path) の最初で指定します。リポジトリは 3 つあります。base は FreeBSD ベースシステムのソースコード、ports は Ports Collection、そして doc はドキュメントのリポジトリです。たとえば、svn://svn0.us-east.FreeBSD.org/ports/head/ という URL は、svn プロトコルによる svn0.us-east.FreeBSD.org ミラー上の ports リポジトリのメインブランチを示しています。

以下のように入力して、リポジトリからチェックアウトしてください。

```
# svn checkout svn-mirror/repository/branch lwcdir
```

ここで、*repository* , *branch* および *root* は以下のとおりです。

・ *svn-mirror* は、Subversion ミラーサイト のひとつの URL です。

- *repository* には、プロジェクトのリポジトリ、すなわち base, ports, または doc のどれかひとつを指定します。
- *branch* は、使うリポジトリによります。ports および doc では、ほとんどの変更が head ブランチで行われます。base リポジトリでは、head ブランチで -CURRENT の最新バージョンを管理しています。-STABLE ブランチの最新バージョンは、8.*x* は stable/8, 9.*x* は stable/9, そして 10.*x* は stable/10 で管理しています。
- *lwcdir* は、指定したブランチの中身が置かれるターゲットのディレクトリです。通常 ports は /usr/ports、base は /usr/src、そして doc では /usr/doc と指定します。

以下の例では、Ports Collection を western US リポジトリから HTTPS プロトコルを使って、チェックアウトします。そしてそれらは、/usr/ports のローカル作業コピーに置かれます。もし /usr/ports がすでに存在して、それが svn によって生成されたものでなければ、チェックアウトする前に、名前を変更するか削除してください。

```
# svn checkout https://svn0.us-west.FreeBSD.org /ports/
head /usr/ports
```

初めてチェックアウトする際には、リモートリポジトリのすべてのブランチをダウンロードするので時間がかかります。どうぞ我慢してください。

初めてのチェックアウト後は、 以下を実行することでローカル作業コピーをアップデートできます。

```
# svn update lwcdir
```

この例で作成された /usr/ports をアップデートするには、 以下のようにしてください。

```
# svn update /usr/ports
```

アップデートはチェックアウトにくらべ、 変更点のあるファイルのみが転送されるので高速です。

チェックアウト後、ローカル作業コピーをアップデートするもうひとつの方法は、 /usr/ports, /usr/src または /usr/doc ディレクトリの Makefile で提供されています。SVN_UPDATE を設定して update ターゲットを使ってください。たとえば、/usr/src をアップデートするには、以下のようにしてください。

```
# cd /usr/src
# make update SVN_UPDATE=yes
```

A.4.4. Subversion ミラーサイト

すべてのミラーはすべてのリポジトリを持っています。

FreeBSD Subversion サーバのマスタである `svn.FreeBSD.org` は、公には読み出し専用でアクセスできますが、将来的には変更される予定ですので、オフィシャルミラーを使うことが推奨されます。ブラウザを用いて FreeBSD の Subversion リポジトリを参照するには、http://svnweb.FreeBSD.org/ を利用してください。

注記

FreeBSD の Subversion ミラーのネットワークは、まだ初期の段階にあるので、今後変更されることがあります。 以下のミラー一覧を不変なものとは考えないでください。特に、サーバの SSL 証明書は、いずれかの時点で変更になるでしょう。

名前	プロトコル	位置	SSL フィンガープリント
svn0.us-west.FreeBSD.org	svn, http, https	USA, カリフォルニア	SHA1 1C:BD:85:95:11:9F:EB:75:A5:4B:C8:A3:FE:08:E4:02:73
svn0.us-east.FreeBSD.org	svn, http, https, rsync	USA, ニュージャージ	SHA1 1C:BD:85:95:11:9F:EB:75:A5:4B:C8:A3:FE:08:E4:02:73
svn0.eu.FreeBSD.org	svn, http, https, rsync	Europe, UK	SHA1 39:B0:53:35:CE:60:C7:BB:00:54:96:96:71:10:94:BB:CE
svn0.ru.FreeBSD.org	svn, http, https, rsync	ロシア、モスクワ	SHA1 F6:44:AA:B9:03:89:0E:3E:8C:4D:4D:14:F0:27:E6:C7:C1

HTTPS は推奨されているプロトコルです。他のコンピュータが FreeBSD ミラーを装う (一般的には「マン・イン・ザ・ミドル」攻撃として知られています) ことや、もしくは、エンドユーザに対し好ましくない内容を送りつけようということに対し保護を行います。

HTTPS ミラーへの最初の接続の際には、サーバの フィンガープリント の確認を求められます。

```
Error validating server certificate for 'https://svn0.us-west.↺
freebsd.org:443':
 - The certificate is not issued by a trusted authority. Use the
   fingerprint to validate the certificate manually!
 - The certificate hostname does not match.
Certificate information:
 - Hostname: svnmir.ysv.FreeBSD.org
 - Valid: from Jul 29 22:01:21 2013 GMT until Dec 13 22:01:21 2040 ↺
GMT
 - Issuer: clusteradm, FreeBSD.org, (null), CA, US ↺
(clusteradm@FreeBSD.org)
 - Fingerprint: ↺
1C:BD:85:95:11:9F:EB:75:A5:4B:C8:A3:FE:08:E4:02:73:06:1E:61
(R)eject, accept (t)emporarily or accept (p)ermanently?
```

フィンガープリントを上の表の一覧のものと照合してください。フィンガープリントが一致したら、 サーバのセキュリティ証明書を一時的 (permanently) もしくは恒久的 (temporarily) に受け入れてください。一時的な認証であれば、サーバとの一回のセッションで有効期限が切れるため、次回の接続時にはもう一度検証が行われます。恒常的な認証を選んだ場合には、認証のための証明書が `~/.subversion/auth/` に保存され、有効期限が切れるまでは、フィンガープリントの確認は求められません。

ファイアウォールまたは他の問題のため、`https` を使えなければ、転送速度がより少し早い `svn` を使ってください。両方を使えない場合には、`http` を使ってください。

A.4.5. より詳しい情報

Subversion の利用に関する他の情報は、Version Control with Subversion や Subversion Documentation といった「Subversion Book」をご覧ください。

A.5. rsync を使用する

次のサイトは、FreeBSD を rsync プロトコルで提供しています。rsync ユーティリティは rcp(1) コマンドとほぼ同じ機能を実現するもので、こちらの方が豊富なオプションを備え、送り側と受け側の差分だけを 転送するという rsync リモート更新プロトコルを使用するという点が異なります。 rsync を使うと、ネットワーク経由での同期を非常に高速に行なうことが可能です。特に、FreeBSD FTP サーバや CVS リポジトリのミラーサイトを作成する時に便利でしょう。rsync は、多くのオペレーティングシステムで 利用することができます。FreeBSD 版は、net/rsync の port か、package を使ってください。

チェコ共和国
: rsync://ftp.cz.FreeBSD.org/

  提供しているコレクション:

  - ftp: FreeBSD FTP サーバの部分ミラー

・ FreeBSD: FreeBSD FTP サーバの全体ミラー

オランダ
rsync://ftp.nl.FreeBSD.org/

提供しているコレクション:

・ FreeBSD: FreeBSD FTP サーバの全体ミラー

ロシア
rsync://ftp.mtu.ru/

提供しているコレクション:

・ FreeBSD: FreeBSD FTP サーバの全体ミラー

・ FreeBSD-Archive: FreeBSD アーカイブ FTP サーバのミラー

スウェーデン
rsync://ftp4.se.freebsd.org/

提供しているコレクション:

・ FreeBSD: FreeBSD FTP サーバの全体ミラー

台湾
rsync://ftp.tw.FreeBSD.org/

rsync://ftp2.tw.FreeBSD.org/

rsync://ftp6.tw.FreeBSD.org/

提供しているコレクション:

・ FreeBSD: FreeBSD FTP サーバの全体ミラー

イギリス
rsync://rsync.mirrorservice.org/

提供しているコレクション:

・ ftp.freebsd.org: FreeBSD FTP サーバの全体ミラー

アメリカ合衆国
rsync://ftp-master.FreeBSD.org/

このサーバは、FreeBSD の一次ミラーサイトとしてのみ使われています。

提供しているコレクション:

- FreeBSD: FreeBSD FTP サーバのマスタアーカイブ
- acl: The FreeBSD マスタ ACL リスト

rsync://ftp13.FreeBSD.org/

提供しているコレクション:

- FreeBSD: FreeBSD FTP サーバの全体ミラー

付録B 参考図書

訳: 中井 幸博 <nakai@mlab.t.u-tokyo.ac.jp >, 1996 年 10 月 12 日。

FreeBSD オペレーティングシステムの個々の部分については マニュアルページで定義のような説明がなされていますが, どうやってその部分どうしをつなぎあわせて オペレーティングシステム全体を円滑に動作させるかについては, ほとんど説明されていません。それを補うためにはのよい本や, UNIX® システム管理についての 利用者向けのマニュアルが欠かせません。

B.1. FreeBSD 専門の書籍

非英語文化圏の書籍:

- FreeBSD 入門與應用 (繁体字中国語)。 Drmaster 発行, 1997. ISBN 9-578-39435-7.
- FreeBSD Unleashed (簡体字中国語訳), China Machine Press 発行. ISBN 7-111-10201-0.
- FreeBSD From Scratch Second Edition (簡体字中国語), China Machine Press 発行. ISBN 7-111-10286-X.
- FreeBSD ハンドブック第 2 版 (簡体字中国語訳), Posts & Telecom Press 発行. ISBN 7-115-10541-3.
- FreeBSD & Windows (簡体字中国語), China Railway Publishing House 発行. ISBN 7-113-03845-X
- FreeBSD Internet Services HOWTO (簡体字中国語), China Railway Publishing House 発行. ISBN 7-113-03423-3
- FreeBSD入門キット AT互換機版 第二版。 宮嵜忠臣 著。 秀和システム。 ISBN 4-87966-535-5 C3055 2900 円。
- ここまでできる FreeBSD パワーガイド。霜山 滋, 仲道 嘉夫, 山中 右次 著。秀和システム。ISBN 4-87966-637-8 2600円。
- FreeBSD徹底入門。あさだ たくや / 天川 修平 / 衛藤 敏寿 / 浜田 直樹 / 細川 達己 / 三田 吉郎 著。翔泳社。ISBN 4-88135-473-6 3600 円。
- パーソナル UNIX スターターキット FreeBSD。民田 雅人 / 古場 正行 / 増田 佳泰 / 天池 健 / 宮川 晋 共著。アスキー。ISBN 4-7561-1733-3 3000 円。
- FreeBSD ハンドブック (日本語版)。アスキー。ISBN 4-7561-1580-2 3800 円。

- FreeBSD mit Methode (ドイツ語版)。Computer und Literatur Verlag/Vertrieb Hanser 発行。1998。ISBN 3-932311-31-0
- FreeBSD de Luxe (ドイツ語), Verlag Modere Industrie 発行, 2003 年。ISBN 3-8266-1343-0.
- FreeBSD インストール & 活用マニュアル, 毎日コミュニケーションズ発行。1998 年. ISBN 4-8399-0112-0.
- Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, Widjil Widodo 著 FreeBSD を使ったインターネットサーバの構築 (Building Internet Server with FreeBSD) (インドネシア語), Elex Media Komputindo 発行。
- Absolute BSD: The Ultimate Guide to FreeBSD (繁体字中国語訳) GrandTech Press 発行 (2003 年)。ISBN 986-7944-92-5.
- The FreeBSD 6.0 Book (繁体字中国語), Drmaster 発行 (2006 年)。ISBN 9-575-27878-X.

英語の書籍:

- Absolute FreeBSD, 2nd Edition: The Complete Guide to FreeBSD, No Starch Press 刊、2007 年。ISBN: 978-1-59327-151-0
- The Complete FreeBSD, O'Reilly、2003 年。ISBN: 0596005164
- The FreeBSD Corporate Networker's Guide, Addison-Wesley 刊、2000 年。ISBN: 0201704811
- FreeBSD: An Open-Source Operating System for Your Personal Computer、The Bit Tree Press 刊、2001 年。ISBN: 0971204500
- Teach Yourself FreeBSD in 24 Hours, Sams 刊、2002 年。ISBN: 0672324245
- FreeBSD 6 unleashed, Sams 刊、2006 年。ISBN: 0672328755
- FreeBSD: The Complete Reference, McGrawHill 刊、2003 年。ISBN: 0072224096

B.2. 利用者向けのガイド

- オハイオ州立大学による UNIX Introductory Course。オンラインで HTML 版と PostScript 版が利用可能。

  FreeBSD イタリア語ドキュメンテーションプロジェクトの一環 として、このドキュメントの イタリア語訳 が用意されています。

- FreeBSD 友の会 jpman プロジェクト。FreeBSD User's Reference Manual (日本語訳)。毎日コミュニケーションズ, 1998。ISBN4-8399-0088-4 P3800E。

- Edinburgh University による UNIX 環境の初心者向け オンラインガイド。

B.3. 管理者向けのガイド

- FreeBSD 友の会 jpman プロジェクト。FreeBSD System Administrator's Manual (日本語訳)。毎日コミュニケーションズ, 1998. ISBN4-8399-0109-0 P3300E.

- Dreyfus, Emmanuel. Cahiers de l'Admin: BSD 第 2 版。(フランス語、「管理者ノート」)、Eyrolles, 2004. ISBN 2-212-11463-X

B.4. プログラマ向けのガイド

- Computer Systems Research Group, UC Berkeley. 4.4BSD Programmer's Reference Manual. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-078-3

- Computer Systems Research Group, UC Berkeley. 4.4BSD Programmer's Supplementary Documents. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-079-1

- Harbison, Samuel P. and Steele, Guy L. Jr. C: A Reference Manual. 4th Ed. Prentice Hall, 1995. ISBN 0-13-326224-3 (訳注: 邦訳は以下のものが出版されています。斎藤 信男監訳。新・詳説 C 言語リファレンス [H&Sリファレンス]。ソフトバンク, 1994。ISBN 4-89052-506-8 原本は第 4 版だが, 訳出は第 3 版のみ。)

- Kernighan, Brian and Dennis M. Ritchie. The C Programming Language. 2nd Ed. PTR Prentice Hall, 1988. ISBN 0-13-110362-8 (訳注: 邦訳は以下のものが出版されています。石田 晴久 訳。プログラミング言語 C 第 2 版(訳書訂正版) 共立出版, 1989。ISBN 4-320-02692-6)

- Lehey, Greg. Porting UNIX Software. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-126-7

- Plauger, P. J. The Standard C Library. Prentice Hall, 1992. ISBN 0-13-131509-9 (訳注: 邦訳は以下のものが出版されています。福富 寛 / 門倉 明彦 / 清水 恵介 訳。標準 C ライブラリ ANSI/ISO/JIS C規格. トッパン, 1995。ISBN 4-8101-8541-9)

- Spinellis, Diomidis. Code Reading: The Open Source Perspective. Addison-Wesley, 2003. ISBN 0-201-79940-5

- Spinellis, Diomidis. Code Quality: The Open Source Perspective. Addison-Wesley, 2006. ISBN 0-321-16607-8
- Stevens, W. Richard and Stephen A. Rago. Advanced Programming in the UNIX Environment. 2nd Ed. Reading, Mass. : Addison-Wesley, 2005. ISBN 0-201-43307-9 (訳注: 第 1 版の邦訳は以下のものが出版されています。大木 敦雄 訳。詳解 UNIX プログラミング。トッパン, 1994。ISBN 4-89052-524-6)
- Stevens, W. Richard. UNIX Network Programming. 2nd Ed. PTR Prentice Hall, 1998. ISBN 0-13-949876-1 (訳注: 第 1 版の邦訳は以下のものが出版されています. 篠田 陽一 訳. UNIX ネットワークプログラミング。トッパン, 1992. ISBN 4-8101-8509-5) 第 2 版の邦訳は以下のものが出版されています。篠田 陽一 訳. UNIX ネットワークプログラミング 第 2 版 Vol.1。トッパン, 1999。ISBN 4-8101-8612-1)

B.5. オペレーティングシステム内部

- Andleigh, Prabhat K. UNIX System Architecture. Prentice-Hall, Inc., 1990. ISBN 0-13-949843-5
- Jolitz, William.「Porting UNIX to the 386」. Dr. Dobb's Journal. January 1991-July 1992.
- Leffler, Samuel J., Marshall Kirk McKusick, Michael J Karels and John Quarterman The Design and Implementation of the 4.3BSD UNIX Operating System. Reading, Mass. : Addison-Wesley, 1989. ISBN 0-201-06196-1 (訳注: 邦訳は以下のものが出版されています。中村 明 / 相田 仁 / 計 宇生 / 小池 汎平 訳。UNIX 4.3BSDの設計と実装。丸善, 1991。ISBN 4-621-03607-6)
- Leffler, Samuel J., Marshall Kirk McKusick, The Design and Implementation of the 4.3BSD UNIX Operating System: Answer Book. Reading, Mass. : Addison-Wesley, 1991. ISBN 0-201-54629-9 (訳注: 邦訳は以下のものが出版されています。相田 仁 / 計 宇生 / 小池 汎平 訳。UNIX 4.3BSDの設計と実装。アンサーブック, トッパン, 1991。ISBN 4-8101-8039-5)
- McKusick, Marshall Kirk, Keith Bostic, Michael J Karels, and John Quarterman. The Design and Implementation of the 4.4BSD Operating System. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-54979-4

  (この本の第二章が FreeBSD ドキュメンテーションプロジェクト の一部として オンライン で公開されています。)
- Marshall Kirk McKusick, George V. Neville-Neil The Design and Implementation of the FreeBSD Operating System. Boston, Mass. : Addison-Wesley, 2004. ISBN 0-201-70245-2

- Marshall Kirk McKusick, George V. Neville-Neil, Robert N. M. Watson The Design and Implementation of the FreeBSD Operating System, 2nd Ed.. Westford, Mass. : Pearson Education, Inc., 2014. ISBN 0-321-96897-2

- Stevens, W. Richard. TCP/IP Illustrated, Volume 1: The Protocols. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63346-9

- Schimmel, Curt. Unix Systems for Modern Architectures. Reading, Mass. : Addison-Wesley, 1994. ISBN 0-201-63338-8

- Stevens, W. Richard. TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63495-3 (訳注: 邦訳は以下のものが出版されています。中本 幸一 / 石川 裕次 / 田中 伸佳 訳。詳解 TCP/IP Vol.3: トランザクション TCP, HTTP, NNTP, UNIX ドメインプロトコル, アジソンウェスレイパブリッシャーズジャパン, 1998。ISBN 4-8101-8039-5)

- Vahalia, Uresh. UNIX Internals -- The New Frontiers. Prentice Hall, 1996. ISBN 0-13-101908-2 (訳注: 邦訳は以下のものが出版されています。徳田 英幸 / 中村 明 / 戸辺 義人 / 津田 悦幸 訳。最前線UNIXのカーネル, ピアソンエデュケーション, 2000。ISBN 4-89471-189-3)

- Wright, Gary R. and W. Richard Stevens. TCP/IP Illustrated, Volume 2: The Implementation. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63354-X

- Messmer, Hans-Peter. The Indispensable PC Hardware Book, 4th Ed. Reading, Mass : Addison-Wesley Pub. Co., 2002. ISBN 0-201-59616-4

B.6. セキュリティの参考資料

- Cheswick, William R. and Steven M. Bellovin. Firewalls and Internet Security: Repelling the Wily Hacker. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63357-4 (訳注: 邦訳は以下のものが出版されています。川副 博 監訳。ファイアウォール。ソフトバンク, 1995。ISBN 4-89052-672-2)

- Garfinkel, Simson. PGP Pretty Good Privacy O'Reilly & Associates, Inc., 1995. ISBN 1-56592-098-8

B.7. ハードウェアの参考資料

- Anderson, Don and Tom Shanley. Pentium Processor System Architecture. 2nd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40992-5

- Ferraro, Richard F. Programmer's Guide to the EGA, VGA, and Super VGA Cards. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-62490-7
- Intel Corporation は、自社の CPU やチップセットに関する文書を自社の 開発者向け Web サイト で公開しています。文書のフォーマットは通常 PDF です。
- Shanley, Tom. 80486 System Architecture. 3rd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40994-1
- Shanley, Tom. ISA System Architecture. 3rd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40996-8
- Shanley, Tom. PCI System Architecture. 4th Ed. Reading, Mass. : Addison-Wesley, 1999. ISBN 0-201-30974-2
- Van Gilluwe, Frank. The Undocumented PC, 2nd Ed. Reading, Mass: Addison-Wesley Pub. Co., 1996. ISBN 0-201-47950-8

B.8. UNIX® の歴史

- Lion, John Lion's Commentary on UNIX, 6th Ed. With Source Code. ITP Media Group, 1996. ISBN 1573980137
- Raymond, Eric s. The New Hacker's Dictionary, 3rd edition. MIT Press, 1996. ISBN 0-262-68092-0 これは ジャーゴンファイル (Jargon File) として知られています。
- Saulus, Peter H. A quarter century of UNIX. Addison-Wesley Publishing Company, Inc., 1994. ISBN 0-201-54777-5
- Simon Garfinkel, Daniel Weise, Steven Strassmann. The UNIX-HATERS Handbook. IDG Books Worldwide, Inc., 1994. ISBN 1-56884-203-1. 絶版となりましたが、オンライン で入手できます。
- Don Libes, Sandy Ressler Life with UNIX — special edition. Prentice-Hall, Inc., 1989. ISBN 0-13-536657-7 (訳注: 邦訳は以下のものが出版されています。坂本 文 監訳. Life with UNIX. アスキー, 1990。ISBN 4-7561-0783-4 邦訳が Special 版の訳出か否かは不明)
- BSD 系 OS の系譜図。`https://svnweb.freebsd.org/base/head/share/misc/bsd-family-tree?view=co` か、もしくは、FreeBSD マシンにある `/usr/share/misc/bsd-family-tree` 。
- Networked Computer Science Technical Reports Library. `http://www.ncstrl.org/`

- Computer Systems Research group (CSRG) からの古い BSD リリース集。`http://www.mckusick.com/csrg/` : この 4 枚 CD セットには、1BSD から 4.4BSD までと 4.4BSD-Lite2 が含まれます (残念ながら 2.11BSD は含まれていません)。また 4 枚目の CD には、最終ソースおよび SCCS ファイルが含まれています。

B.9. 定期刊行物、雑誌およびジャーナル

- Admin Magazin (in German), published by Medialinx AG. ISSN: 2190-1066
- BSD Magazine, published by Software Press Sp. z o.o. SK. ISSN: 1898-9144
- BSD Now — Video Podcast, published by Jupiter Broadcasting LLC
- BSD Talk Podcast, by Will Backman
- FreeBSD Journal, published by S&W Publishing, sponsored by The FreeBSD Foundation. ISBN: 978-0-615-88479-0

付録C インターネット上のリソース

訳: 前田 幸範 **<yuki@jp.FreeBSD.org >**、1996 年 8 月 28 日

FreeBSD の進歩は急速であり、印刷したメディアは最新の開発をフォローするのに実用的ではありません。それだけしかない、というわけではありませんが、最新情報を入手する方法としては電子的なリソースがベストです。 FreeBSD はボランティアの努力によって、ユーザコミュニティ自体が、一種の「テクニカルサポート部門」としての役割も通常果たしており、電子メールや Usenet のニュースがこれらのコミュニティにたどり着く最も効果的な方法になっています。

以下に、FreeBSD ユーザコミュニティに連絡を取る場合の最も重要な点についての 概略を示します。 もしここに書かれていない他のリソースに気がついた場合は、それらを FreeBSD documentation project メーリングリスト に送って頂ければ、それらをここに含めるかもしれません。

C.1. メーリングリスト

多くの FreeBSD の開発メンバは USENET を読むことができますが、 もし、`comp.unix.bsd.freebsd.*` のグループの一つに質問を投稿したとしても タイムリにその質問を受け取るということは保証できません。質問を適切なメーリングリストに投稿すれば、私たちか FreeBSD の関係者から、よりよい (そして少なくともより早い) 反応がいつでも得られることでしょう。

さまざまなメーリングリストの憲章を このドキュメントの最後に記載します。私たちは、メーリングリストの質、特に技術面に関する質を高く保つ ために努力しているので、 メーリングリストに参加する前にその憲章を読んでください。 私たちのメーリングリストの参加者のほとんどは、現在非常 にたくさんの FreeBSD に関連したメッセージを毎日受け取っており、メーリン グリストに対するふさわしい 用いられ方をするための憲章やルールを決めるこ とによって、メーリングリストの S/N 比を高くする保つように励んでいます。そうしないと、結果的に、メーリングリストがプロジェクトにとって 事実上のコミュニケーションの手段になってしまうでしょう。

メーリングリストはいずれもアーカイブされており、それらは FreeBSD World Wide Web server で検索することができます。 キーワード検索可能なアーカイブの提供は、良くある質問に対する回答を見つけるすぐれた方法ですから、 質問を投稿する前に調べてみるべきでしょう。

C.1.1. メーリングリストの概説

一般的なメーリングリスト: 以下のものは誰でも自由に参加できる (そしておすすめの) 一般的なものです。

リスト	目的
freebsd-advocacy	FreeBSD の福音伝道
freebsd-announce	重要なイベントやプロジェクトのマイルストン (モデレータ制)
freebsd-arch	アーキテクチャ、設計に関する議論
freebsd-bugbusters	FreeBSD 障害報告データベースおよび関連するツールの管理に関する議論
freebsd-bugs	バグレポート
freebsd-chat	FreeBSD コミュニティに関連する技術的ではない話題
freebsd-current	FreeBSD-CURRENT の使用に関連する議論
freebsd-isp	FreeBSD を用いている インターネットサービスプロバイダの話題
freebsd-jobs	FreeBSD 関連の雇用機会に関する話題
freebsd-newbies	FreeBSD 初心者ユーザの活動と議論
freebsd-questions	ユーザからの質問と技術サポート
freebsd-security-notifications	セキュリティに関する通知 (モデレータ制)
freebsd-stable	FreeBSD-STABLE の使用に関連する議論
freebsd-test	メッセージの送信試験を行なうために、 実際のメーリングリストの代わりに使うアドレス

技術的なメーリングリスト: 以下のメーリングリストは、技術的な 議論のためのものです。それらの利用や内容のためにしっかりとしたガイドラ インがあるので、これらのメーリングリストに入ったり、どれか一つにメール を送ったりする前には、それらのメーリングリストの憲章を注意深く読むべきです。

リスト	目的
freebsd-afs	FreeBSD へのAFSの移植
freebsd-aic7xxx	Adaptec® AIC 7xxx 用ドライバの開発
freebsd-alpha	FreeBSD の Alpha への移植
freebsd-amd64	FreeBSD の AMD64 システムへの移植 (モデレータ制)
freebsd-arm	FreeBSD の ARM® プロセッサへの移植

リスト	目的
freebsd-atm	FreeBSD での ATM ネットワーク使用に関する話題
freebsd-cluster	FreeBSD のクラスタ環境での利用
freebsd-cvsweb	CVSweb の管理
freebsd-database	FreeBSD 上でのデータベースの利用や開発に関する議論
freebsd-desktop	デスクトップでの FreeBSD の利用や改良について
freebsd-doc	FreeBSD 関連ドキュメントの作成
freebsd-emulation	Linux/DOS/Windows® のような他のシステムのエミュレーション
freebsd-firewire	FreeBSD FireWire® (iLink, IEEE 1394) に関する技術的な議論
freebsd-fs	ファイルシステム
freebsd-gnome	GNOME および GNOME アプリケーションの移植
freebsd-hackers	一般的な技術の議論
freebsd-hardware	FreeBSD の走るハードウェアの一般的な議論
freebsd-i18n	FreeBSD の国際化
freebsd-ia32	FreeBSD の IA-32 (Intel® x86) プラットフォームへの移植
freebsd-ia64	FreeBSD の Intel が開発中の IA64 システムへの移植
freebsd-infiniband	FreeBSD での Infiniband の使用
freebsd-ipfw	IP firewall コードの再設計に関する技術的議論
freebsd-isdn	ISDN 開発者
freebsd-java	Java™ 開発者や、FreeBSD へ JDK™ を移植する人たち
freebsd-kde	KDE および KDE アプリケーションの移植
freebsd-lfs	LFS の FreeBSD への移植
freebsd-mips	FreeBSD の MIPS® への移植

リスト	目的
freebsd-mobile	モーバイルコンピューティングについての議論
freebsd-new-bus	バスアーキテクチャに関する技術的な議論
freebsd-net	ネットワークおよび TCP/IP ソースコードに関する議論
freebsd-numerics	高品質な libm 機能の実装に関する議論
freebsd-office	FreeBSD でのオフィスアプリケーションについて
freebsd-performance	ハイパフォーマンス / 高負荷での導入のためのパフォーマンスチューニングに関する質問
freebsd-perl	数多く存在する perl に関連する port の管理について
freebsd-pkg	バイナリ package 管理および package ツールについての議論
freebsd-platforms	Intel 以外のアーキテクチャのプラットフォームへの移植
freebsd-ports	Ports Collection に関する議論
freebsd-ports-announce	Ports Collection に関する重要なニュースと案内 (モデレータ制)
freebsd-ports-bugs	ports のバグや PR についての議論
freebsd-ppc	FreeBSD の PowerPC® への移植
freebsd-qa	品質保証 (通常はリリースの保留) についての議論
freebsd-realtime	FreeBSD 用のリアルタイム拡張の開発に関する話題
freebsd-scsi	SCSI サブシステム
freebsd-security	セキュリティに関する話題
freebsd-small	組み込みアプリケーションにおける FreeBSD の利用
freebsd-smp	対称 / 非対称の マルチプロセッシングの設計に関する議論
freebsd-snapshots	FreeBSD 開発スナップショットのアナウンス

リスト	目的
freebsd-sparc64	FreeBSD の SPARC® ベースシステムへの移植
freebsd-standards	C99 および POSIX® 標準への FreeBSD の適合について
freebsd-threads	FreeBSD 固有の Tcl/Tk に関する議論
freebsd-threads	FreeBSD のスレッドについて
freebsd-tokenring	FreeBSD でのトークンリングのサポート
freebsd-x11	FreeBSD での X11 のメンテナンスとサポート
freebsd-xfce	XFCE の FreeBSD への移植や保守について
freebsd-zope	Zope の FreeBSD への移植や保守について

制限されているメーリングリスト: 以下のメーリングリストはより特化された (そしてより厳しい) メンバーのためのものであり、一般的な興味を惹くようなものでは ありません。 このようなメーリングリストに参加する前に、技術的なメーリングリストで 自らの存在感をアピールするのは良い考えです。 そうすることにより、議論の際のエチケットを学ぶことができるでしょう。

メーリングリスト	目的
freebsd-hubs	ミラーサイトを運営している人達 (基盤のサポート)
freebsd-user-groups	ユーザグループの調整

メーリングリストのダイジェスト版: 上述のメーリングリストのすべてでダイジェスト版を利用できます。 メーリングリストに登録すると、アカウントのオプションセクションで、 ダイジェストのオプションを変更できます。

SVN メーリングリスト: 以下のメーリングリストはソースツリーの さまざまな場所の変更のログメッセージを見ることに 興味のある人向けです。これらは読み専用のリストで、これらにメールを送る事は出来ません。

メーリングリスト	ソースの範囲	(ソースの) 範囲の説明
svn-doc-all	`/usr/doc`	doc subversion リポジトリへ加えられたすべての変更 (`user`, `projects` および `translations` を除く)

メーリングリスト	ソースの範囲	(ソースの) 範囲の説明
svn-doc-head	/usr/doc	doc subversion リポジトリの「head」ブランチに加えられたすべての変更
svn-doc-projects	/usr/doc/projects	doc subversion リポジトリの projects に加えられたすべての変更
svn-doc-svnadmin	/usr/doc	doc subversion リポジトリの管理用スクリプト、フックおよび他のコンフィグレーションデータに対して加えられたすべての変更
svn-ports-all	/usr/ports	ports subversion リポジトリへ加えられたすべての変更
svn-ports-head	/usr/ports	ports subversion リポジトリの「head」ブランチに加えられたすべての変更
svn-ports-svnadmin	/usr/ports	ports subversion リポジトリの管理用スクリプト、フックおよび他のコンフィグレーションデータに対して加えられたすべての変更
svn-src-all	/usr/src	doc subversion リポジトリへ加えられたすべての変更 (user および projects を除く)
svn-src-head	/usr/src	src subversion リポジトリの「head」ブランチ (FreeBSD-CURRENT ブランチ) に加えられたすべての変更
svn-src-projects	/usr/projects	src subversion リポジトリの projects に加えられたすべての変更
svn-src-release	/usr/src	src subversion リポジトリの releases に加えられたすべての変更

メーリングリスト	ソースの範囲	(ソースの) 範囲の説明
svn-src-releng	/usr/src	src subversion リポジトリの releng ブランチ (セキュリティ / リリースエンジニアリングブランチ) に加えられたすべての変更
svn-src-stable	/usr/src	src subversion リポジトリのすべての stable ブランチに加えられたすべての変更
svn-src-stable-6	/usr/src	src subversion リポジトリの stable/6 ブランチに加えられたすべての変更
svn-src-stable-7	/usr/src	src subversion リポジトリの stable/7 ブランチに加えられたすべての変更
svn-src-stable-8	/usr/src	src subversion リポジトリの stable/8 ブランチに加えられたすべての変更
svn-src-stable-9	/usr/src	src subversion リポジトリの stable/9 ブランチに加えられたすべての変更
svn-src-stable-10	/usr/src	src subversion リポジトリの stable/10 ブランチに加えられたすべての変更
svn-src-stable-other	/usr/src	src subversion リポジトリの古い stable ブランチに加えられたすべての変更
svn-src-svnadmin	/usr/src	src subversion リポジトリの管理用スクリプト、 フックおよび他のコンフィグレーションデータに対して加えられたすべての変更
svn-src-user	/usr/src	src subversion リポジトリの user に加えられた実験的なすべての変更
svn-src-vendor	/usr/src	src subversion リポジトリの vender に加えられたすべての変更

C.1.2. 参加方法

メーリングリストに参加するには、上記のリストをクリックするか、 http://lists.FreeBSD.org/mailman/listinfo に進んで興味を持っているメーリングリストをクリックしてください。 メーリングリストのページには、登録に必要な手順が書かれています。

メーリングリストにメールを送るには、ただ <*listname*@FreeBSD.org> にメールを送るだけです。すると、メーリングリストに登録されている世界中のメンバに再配布されます。

メーリングリストから登録を解除する場合は、メーリングリストで配信されているメールの最後にある URL をクリックしてください。 または、 freebsd-[メーリングリストの名前]-unsubscribe@FreeBSD.org にメールを送信することでも登録を解除できます。

再度言いますが、 私たちは技術的なメーリングリストでは技術的な議論を保つよう 要求します。もし、重要なアナウンスにのみ興味があるなら、 FreeBSD announcements メーリングリスト への参加をお勧めします。これには、あまりたくさんのメールは流れません。

C.1.3. メーリングリストの憲章

すべて FreeBSD メーリングリストは誰でもそれらを利用することに 固守しなければいけないという一定の簡単なルールがあります。 これらのルールに従わないと、結果として FreeBSD の Postmaster <postmaster@FreeBSD.org >から 2 回までは警 告を受けます。3 回違反すると、投稿者はすべての FreeBSD のメーリングリストから削除され、そのメーリングリストへのさらなる投稿から締め出されるでしょう。これらのルールや対策が必要なのは残念です。しかし、今日のインターネッ トはずいぶんいやらしい環境になっており、一般の人々は、その (対策の) メカニズムがいかにもろいかという事すら 認識する事が出来ていないと思われます。

道標

- いかなる投稿記事もそのメーリングリストの 基本的な憲章を守るべきです。たとえば、そのメーリングリストが技術的な問題に関するものであれば、 技術的な議論を含む投稿でなければなりません。現在継続中の 不適切な憲章やフレイムは、所属しているすべての人に対してメーリングリスト の価値を下げてしまうだけですし、許される行為ではないでしょう。とくに話題のない自由形式の議論に対しては FreeBSD chat メーリングリスト が自由に認可されているので、かわりに使うべきでしょう。

- 一度に 3 つ以上のメーリングリストには決して投稿すべきでは ありません。2 つのメーリングリストには双方に明確な必要性がある場合にのみ 投稿すべきです。どのリストに対しても、 (メーリングリストの)参加者は (複数のメーリングリ ストに)重複して参加しており、関連する部分が少ない (たとえば、「-stable」と「-scsi」) メーリングリストを除いては、 一度に複数のメーリングリストに投稿する理由は全くありません。もし、Cc に複数の メーリングリストがそのような形で現れて、あなたに届いたのであれ

ば、再びそのメールに返事を出す前に、Cc の部分もまた編集するべきです。元記事を書いたのが誰であっても、あなた自身のクロスポストにまだ責任があります。

- ユーザであれ開発者であれ、(議論の中で) 個人を攻撃したり冒涜したりすることは許されません。 個人的なメールを引用したり再投稿したり する許可をもらえなかったり、もらえそうにない時に、それをおこなう ようなネチケット (訳注: ネットワークにおけるエチケット) に対する ひどい違反は好まれませんが、してはならないと特別に定められている わけではありません。しかしながら、そのような内容がメー リングリストの憲章に沿う場合はほとんどありません。このため、メー リングリストの憲章に違反しているということだけで警告 (または禁止) に値するものと考えていいでしょう。

- FreeBSD 以外の関連する製品やサービスの広告は、絶対に禁止し、spam による違反者が宣伝していることが明確であったら、すぐに禁止します。

個々のメーリングリストの憲章:

freebsd-afs
: Andrew ファイルシステム

  このリストは、CMU/Transarc の AFS の移植や使用に関する議論のためです.

freebsd-announce
: **#######/######**

  これは、単にたまに発表される重要な freebsd のイベントに関心がある人のた めのメーリングリストです。これは、スナップショットやその他のリリースに ついてのアナウンスを含みます。そのアナウンスは新しい FreeBSD の機能のア ナウンスを含んでいます。 ボランティア等の呼びかけがあるかもしれません。 これは流通量の少ないメーリングリストで、完全なモデレートメーリングリストです。

freebsd-arch
: アーキテクチャと設計の議論

  これは、FreeBSD のアーキテクチャに関する議論を行なうためのメーリングリストです。当然、その内容は原則的に技術的なものに限定されます。このメーリングリストにふさわしい話題は以下のようなものです。

  - 複数のカスタマイズされたビルドを同時に行うには、 ビルドシステムをどういじり直せばよいか
  - VFS で Heidemann レイヤを動作させるには、何を修正する必要があるか
  - 同一のデバイスドライバを多数のバス、アーキテクチャに共通で使えるようにするには、デバイスドライバインタフェースをどう改変すれば良いか
  - ネットワークドライバの書き方

freebsd-bugbusters
障害報告の取り扱いに関する調整

このメーリングリストは、 バグマイスター、バグバスター、および他の障害報告データベースに純粋に興味を持っているグループの調整や議論についてのフォーラムです。 このメーリングリストは、個別のバグ、パッチ、障害報告について議論を行うためのものではありません。

freebsd-bugs
バグレポート

これは、FreeBSD のバグレポートのためのメーリングリストです。可能である 場合はいつでも、バグは send-pr(1) を使うか、WEB interfaceを用い て送られる必要があります。

freebsd-chat
FreeBSD のコミュニティに関する 技術的ではない話題

このメーリングリストは技術的ではなく、 社会的な情報について、 他のメーリングリストでは取り扱わない話題を含みます。 これは、Jordan がシロイタチに似ているかどうか、 大文字で打つかどうか、誰がたくさんコーヒーを飲むか、 どこのビールが一番うまいか、誰が地下室でビールを作っているか、などについての議論を含みます。時々重要なイベント (将来開催されるパーティーや、結婚式、誕生日、 新しい仕事など) のお知らせが、 技術的なメーリングリストから でてきます。しかし、 フォローは直接 -chatメーリングリストにするべきです。

freebsd-core
FreeBSD コアチーム

これは、コアメンバが使う内部メーリングリストです。 FreeBSD に関連する深刻なやっかい事の裁定や ハイレベルな綿密な調査を要求するときに、このメー リングリストにメッセージを送る事が出来ます。

freebsd-current
FreeBSD-CURRENT の使用に関する議論

これは FreeBSD-CURRENT のユーザのためのメーリングリストです。 メーリングリストでの話題は、-CURRENT で登場した新しい機能について、その新機能によってユーザに影響することについての注意、 および -CURRENT のままでいるために必要な手順についての説明を含みます。「CURRENT」を走らせている人はこのメーリングリストに 登録しなくてはなりません。 これは技術的なメーリングリストなので、 完全に技術的な内容を要求します。

freebsd-cvsweb
FreeBSD CVSweb プロジェクト

FreeBSD-CVSweb の利用、開発および管理についての技術的な議論の場です。

freebsd-desktop
: デスクトップでの FreeBSD の利用や改良について

  デスクトップでの FreeBSD について議論を行うためのフォーラムです。主として、デスクトップの移植者やユーザが FreeBSD のデスクトップサポートに関する問題点や改良について議論する場です。

freebsd-doc
: ドキュメンテーションプロジェクト

  このメーリングリストは FreeBSD 向けの文書の作成に関連する事柄やプロジェクトについて議論を行なうためのものです。このメーリングリストに参加しているメンバは、「FreeBSD ドキュメンテーションプロジェクト」に参加していることになります。このメーリングリストは公開されているので、参加や投稿は自由に行なうことができます。

freebsd-firewire
: FireWire® (iLink, IEEE 1394)

  このメーリングリストは、FreeBSD における FireWire® (IEEE 1394, iLink) サブシステムの設計と実装について議論を行うためのものです。 標準化、バスデバイスとそのプロトコル、アダプタボード/カード/チップセット、そして、それらに適切に対応するためのアーキテクチャとコードの実装が特に関連するトピックです。

freebsd-fs
: ファイルシステム

  FreeBSD のファイルシステムに関する議論 これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-gnome
: GNOME

  FreeBSD の GNOME Desktop Environment に関する議論。これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-infiniband
: FreeBSD での Infiniband の使用

  FreeBSD における Infiniband, OFED および OpenSM に関する技術的なメーリングリストです。

freebsd-ipfw
: IP Firewall

  これは FreeBSD の IP firewall コードの再設計に関する 技術的な議論のためのフォーラムです。これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-ia64
FreeBSD の IA64 への移植

これは FreeBSD の Intel の IA-64 プラットフォームへの 移植に参加している人達のための技術的なメーリングリストで、 問題を提起し解決策を議論するためのものです。このような技術的な議論に興味を持つ個人は歓迎します。

freebsd-isdn
ISDN コミュニケーション

このメーリングリストは、FreeBSD に対する ISDN サポートの開発の議論を おこなう人のためのものです.

freebsd-java
Java™ の開発

このメーリングリストは、FreeBSD 向けの重要な Java™ アプリケーションの開発や、JDK™ の移植やメンテナンスの 議論をする人のためのものです.

freebsd-kde
KDE

FreeBSD システムにおける KDE に関する議論。これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-hackers
技術的な議論

これは FreeBSD に関する技術的な議論のための フォーラムです。これは最もテクニカルなメーリングリストです。このメーリングリストは、FreeBSD 上でアクティブに活動をしている人のためのもので、 問題を持ち出したり、代わりの解決法を議論します。技術的な議論をフォローするのに興味がある人も歓迎します。これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-hardware
FreeBSD のハードウェアの一般的な議論

FreeBSD が走っているハードウェアのタイプや、 何を買ったり避けたりするかに関する様々な問題や、提案に関する議論。

freebsd-hubs
ミラーサイト

FreeBSD ミラーサイトを運用している人達向けの、アナウンスと議論を行なうメーリングリストです。

freebsd-isp
インターネットサービスプロバイダのについての話題

このメーリングリストは、 FreeBSD を用いたインターネット サービスプロバイダ (ISP) に関する話題の議論のためのものです。これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-newbies
初心者の活動の議論

どのメーリングリストでも扱われていなかった 初心者の活動すべてをカバーします。独学や問題解決、リソースの見つけ方や使い方、質問のしかた、メーリングリストの使い方、どのメーリングリストを読めばいいのか、普通の世間話、間違いの犯し方、自慢話、アイディアの共有、物語、道徳的な (技術的でない) お手伝い、FreeBSD コミュニティでの貢献のしかたなどの話題をが含まれます。 これらの問題を取り扱い、freebsd-questions へ質問できるようサポートします。また、 初心者の頃にはまった同じ事柄に悩んでいる 他の初心者との出会いの場としてfreebsd-newbies を活用してください。

freebsd-office
FreeBSD でのオフィスアプリケーション

FreeBSD におけるオフィスアプリケーションのインストール、開発およびサポートについての議論の場です。

freebsd-ops-announce
プロジェクトのインフラストラクチャに関するアナウンス

FreeBSD.org プロジェクトのインフラストラクチャの変更や関連した問題について興味を持っている向けのメーリングリストです。

このモデレートメーリングリストは、アナウンスに制限されています (返答や要求、議論、意見を述べる場ではありません)。

freebsd-performance
FreeBSD のチューニングや速度向上に関する議論

このメーリングリストは、ハッカー、管理者および関連グループが、 FreeBSD のパフォーマンスに関するトピックを議論する場です。このメーリングリストで議論されるべきトピックは、 高負荷における FreeBSD の導入において経験するパフォーマンスの問題や FreeBSD の限界に挑むような話題を含みます。FreeBSD のパフォーマンスを改善したいと考えている方は、このメーリングリストに登録してください。このメーリングリストは、FreeBSD の高速化、堅牢さ、拡張性に興味をもっている、経験のある FreeBSD ユーザ、ハッカー、管理者向けの高度な技術的メーリングです。ドキュメントに目を通さずに質問して答えを求めるような Q and A タイプのメーリングリストではなく、 未解決で答えのないパフォーマンスに関連したトピックへの貢献や問い合わせの場です。

freebsd-pkg
バイナリ package 管理および package ツールについての議論

ソフトウェアのインストールにバイナリ package を用いる FreeBSD システム管理のすべての側面に関する議論。バイナリ package のツールキットとフォーマット、それらの FreeBSD における開発とサポート、package リポジトリ管理そしてサードパーティ製 package を含みます。

package の作成に失敗する ports に関する議論は ports の問題として考えるべきであり、このメーリングリストで議論することは適切ではありません。

freebsd-platforms
Intel 以外のプラットフォームへの移植

クロスプラットフォームの FreeBSD の問題。Intel 以外のプラットフォームへの FreeBSD の移植についての一般的な議論や提案。これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-ports
「ports」の議論

FreeBSD「Ports Collection」(`/usr/ports`) に関連する話題や、Ports Collection の基盤および ports の一般的な構成の整備活動に関する議論。これは技術的なメーリングリストなので、厳密に技術的な内容のみが扱われます。

freebsd-ports-announce
FreeBSD「Ports Collection」に関する重要なニュースと案内

「Ports Collection」(`/usr/ports`) の開発者、作成者およびユーザへの重要なニュース。アーキテクチャ/インフラストラクチャの変更、新しい機能、重要なアップグレードの案内、そしてリリースエンジニアリング情報が扱われます。このメーリングリストの流量は少なく、アナウンスを目的としたものです。

freebsd-ports-bugs
「ports」のバグに関する議論

「Ports Collection」(`/usr/ports`) の障害報告や新たな ports や変更についての提案についての議論。これは技術的なメーリングリストなので、厳密に技術的な内容のみが扱われます。

freebsd-questions
ユーザからの質問

FreeBSD に関する質問のためのメーリングリストです。その質問がかなり技術的だと思わないのであれば、「どのようにして」という質問を技術的なメーリングリストに送るべきではありません。

freebsd-scsi
SCSI サブシステム

これは FreeBSD のための SCSI サブシステムについて作業している人向けです。これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-security
セキュリティの関連の話題

FreeBSD コンピュータのセキュリティの話題 (DES, Kerberos, よく知られているセキュリティホールや、それらのふさぎ方など) これは技術的なメーリングリストなので、完全に技術的な議論を要求します。これは、Q and A のメーリングリストではありません。FAQ に対する Q and A (質問と答えの両方) の貢献は、歓迎されます。

freebsd-security-notifications
セキュリティ関連の通知

FreeBSD のセキュリティ問題や、 修正に関する通知を行ないます。 このメーリングリストは議論を行なうための メーリングリストではありません。議論は FreeBSD-security で行ないます。

freebsd-small
組み込み用に FreeBSD を使う

特殊な FreeBSD 小型システムおよび FreeBSD の組み込み技術に関する議論。これは技術的なメーリングリストなので、完全に技術的な内容を要求します。

freebsd-snapshots
FreeBSD 開発スナップショットのアナウンス

このメーリングリストは、head/ および stable/ ブランチからの新しい FreeBSD 開発スナップショットのアナウンスを行います。

freebsd-stable
FreeBSD-STABLE の使用に関する議論

これは FreeBSD-STABLE のユーザ用のメーリングリスト です。メーリングリストでの話題は、-STABLE で登場した新しい機能について、その新機能によってユーザに影響することについての注意、および -STABLE のままでいるために必要な手順についての説明を含みます。「STABLE」を走らせている人はこの メーリングリストに登録すべきです。 これは技術的なメーリングリストなので、完全に技術的な 内容を要求します。

freebsd-standards
C99 & POSIX への適合

C99 および POSIX 標準への FreeBSD の適合に関連した技術的な議論を行うためのフォーラムです。

freebsd-user-groups
ユーザグループの調整のメーリングリスト

これは、ローカルなユーザグループがお互いに、または、コアチームが指定した個人と問題を議論する、それぞれのローカルエリアのユーザグループからの 調整人向けの メーリングリストです。このメーリングリストはユーザグループ間の ミーティングの概要やプロジェクトの調整に 制限されるべきです。

freebsd-xfce
: XFCE

これは、XFCE 環境を FreeBSD へ移植することを議論する、技術的なメーリングリストです。活発に XFCE を FreeBSD に移植する作業を行なっている人に向けたもので、問題を提起したり、新しい解決法を議論することを目的としています。技術的な議論に興味を持っている方の参加も歓迎します。

freebsd-zope
: Zope

これは、Zope 環境を FreeBSD へ移植することを議論する、技術的なメーリングリストです。活発に Zope を FreeBSD に移植する作業を行なっている人に向けたもので、問題を提起したり、新しい解決法を議論することを目的としています。技術的な議論に興味を持っている方の参加も歓迎します。

C.1.4. メーリングリストのフィルタリング

FreeBSD のメーリングリストは、スパム、ウィルスおよび他の不要なメールを配布してしまわないよう、いくつかの方法でフィルタリングを行なっています。この節で説明するフィルタリングは、メーリングリストを守るために使われている方法のすべてというわけではありません。

メーリングリストでは、以下の添付ファイルを送ることができます。以下の一覧以外の MIME content type の添付ファイルを含むファイルは、メーリングリストに流れる前に取り除かれます。

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed

- text/html
- text/plain
- text/x-diff
- text/x-patch

注記

他の MIME content type の添付を許可するメーリングリストもありますが、上の一覧に含まれるものであれば、ほとんどのメーリングリストで適用できます。

HTML と plain テキストを両方含むメールでは、 HTML の部分が削除されます。HTML のみを含むメールは、plain テキストに変換されます。

C.2. Usenet ニュースグループ

2 つの FreeBSD 用のニュースグループがあります。ここでは FreeBSD の議論をするたくさんの様々な人がおり、他にも FreeBSD に関連するユーザがいます。

C.2.1. BSD 用のニュースグループ

- comp.unix.bsd.freebsd.announce
- comp.unix.bsd.freebsd.misc
- de.comp.os.unix.bsd (ドイツ)
- fr.comp.os.bsd (フランス)

C.2.2. 関連する他の UNIX® のニュースグループ

- comp.unix
- comp.unix.questions
- comp.unix.admin
- comp.unix.programmer
- comp.unix.shell
- comp.unix.user-friendly

- comp.security.unix
- comp.sources.unix
- comp.unix.advocacy
- comp.unix.misc
- comp.unix.bsd

C.2.3. X Window システム

- comp.windows.x.i386unix
- comp.windows.x
- comp.windows.x.apps
- comp.windows.x.announce
- comp.windows.x.intrinsics
- comp.windows.x.motif
- comp.windows.x.pex
- comp.emulators.ms-windows.wine

C.3. World Wide Web サイト

中央サーバ, アイルランド, アメリカ合衆国, アルメニア, イギリス, オーストラリア, オーストリア, オランダ, カナダ, スイス, スウェーデン, スペイン, スロベニア, チェコ共和国, デンマーク, ドイツ, ノルウェー, フィンランド, フランス, ラトビア, リトアニア, ロシア, 香港, 台湾, 南アフリカ, 日本.

(UTC 現在)

- 中央サーバ
  - http://www.FreeBSD.org/
- アイルランド
  - http://www.ie.FreeBSD.org/
- アメリカ合衆国

- http://www5.us.FreeBSD.org/ (IPv6)

- アルメニア
  - http://www1.am.FreeBSD.org/ (IPv6)
- イギリス
  - http://www1.uk.FreeBSD.org/
  - http://www3.uk.FreeBSD.org/
- オーストラリア
  - http://www.au.FreeBSD.org/
  - http://www2.au.FreeBSD.org/
- オーストリア
  - http://www.at.FreeBSD.org/ (IPv6)
- オランダ
  - http://www.nl.FreeBSD.org/
- カナダ
  - http://www.ca.FreeBSD.org/
  - http://www2.ca.FreeBSD.org/
- スイス
  - http://www.ch.FreeBSD.org/ (IPv6)
  - http://www2.ch.FreeBSD.org/ (IPv6)
- スウェーデン
  - http://www.se.FreeBSD.org/

- スペイン
  - http://www.es.FreeBSD.org/
  - http://www2.es.FreeBSD.org/
- スロベニア
  - http://www.si.FreeBSD.org/
- チェコ共和国
  - http://www.cz.FreeBSD.org/ (IPv6)
- デンマーク
  - http://www.dk.FreeBSD.org/ (IPv6)
- ドイツ
  - http://www.de.FreeBSD.org/
- ノルウェー
  - http://www.no.FreeBSD.org/
- フィンランド
  - http://www.fi.FreeBSD.org/
- フランス
  - http://www1.fr.FreeBSD.org/
- ラトビア
  - http://www.lv.FreeBSD.org/
- リトアニア
  - http://www.lt.FreeBSD.org/

- ロシア
  - http://www.ru.FreeBSD.org/ (IPv6)
- 香港
  - http://www.hk.FreeBSD.org/
- 台湾
  - http://www.tw.FreeBSD.org/
  - http://www2.tw.FreeBSD.org/
  - http://www4.tw.FreeBSD.org/
  - http://www5.tw.FreeBSD.org/ (IPv6)
- 南アフリカ
  - http://www.za.FreeBSD.org/
- 日本
  - http://www.jp.FreeBSD.org/www.FreeBSD.org/ (IPv6)

C.4. Email アドレス

以下のユーザーグループは、メンバーに FreeBSD に関係する メールアドレスを提供しています。管理者はアドレスがどのような形であれ 悪用された場合に、アドレスを無効にする権利を留保しています。

ドメイン	内容	ユーザーグループ	管理者
ukug.uk.FreeBSD.org	forward のみ	<freebsd-users@uk.FreeBSD.org>	Lee Johnston <lee@uk.FreeBSD.org>

C.5. シェルアカウント

以下のユーザーグループは、FreeBSD Project を活発にサポートしてくれる 人たちにシェルアカウントを提供しています。 管理者はアカウントがどのような形であれ悪用された場合に、アカウントを取り消す権利を留保しています。

ホスト	アクセス方法	提供内容	管理者
storm.uk.FreeBSD.org	SSH のみ	Read-only cvs, personal web space, email	Brian Somers
dogma.freebsd-uk.eu.org	Telnet/FTP/SSH	Email, Web space, Anonymous FTP	Lee Johnston <lee@uk.FreeBSD.org >

付録D PGP 公開鍵

FreeBSD.org オフィサの PGP 公開鍵を以下に示します。 これらの公開鍵は、署名を検証したり、オフィサに暗号メールを送る必要がある場合に使用できます。すべての FreeBSD 公開鍵の一覧は、PGP Keys にあります。また、完全なキーリングは https://www.FreeBSD.org/doc/pgpkeyring.txt からダウンロードできます。

D.1. オフィサ

D.1.1. セキュリティオフィサチーム <security-officer@FreeBSD.org >

```
pub   1024D/15D68804CA6CDFB2 2002-08-27 [expires: 2014-01-01]
      Key fingerprint = C374 0FC5 69A6 FBB1 4AED  B131 15D6 8804 ↺
CA6C DFB2
uid                          FreeBSD Security Officer <security-
officer@FreeBSD.org>
sub   2048g/E8C96EEFA3071809 2002-08-27 [expires: 2014-01-01]

pub   4096R/ED67ECD65DCF6AE7 2013-09-24 [expires: 2018-01-01]
      Key fingerprint = 1CF7 FF6F ADF5 CA9F BE1B  8CB2 ED67 ECD6 ↺
5DCF 6AE7
uid                          FreeBSD Security Officer <security-
officer@FreeBSD.org>
sub   4096R/B64357A343D9CBAE 2013-09-24 [expires: 2018-01-01]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGiBD1rpGQRBACJ1CQS7VnTTvH3wjscXQed2RoeVi+n3HtxaF9ApJbxb77dXk+/
DL1ZR0bcZ8s7uQ1D5BkrqSHevoA9FlEN02MM9qyIerXter2/ZEporVOG+/XMkIiV
rd3AgVwUnawhOMKTlYmttcOpADKr9RkYvCT6QMqFDXJssbW7gPlEqOzhYwCgoIdD
ygZ5RdfXm/hBnp+oTWadeIED/2WvL/Iy0YheRTSmTvEdK+Cd4xPhmY2SrrvF2+lE
oFIn94C0fJhqKhJp+wGXmQ/h3yF0gcr1NfFBm6y1iztEz2n0ciaEmMf1tu0Y+u+Y
E0/1Igpoj9Kj5xxRJD5wYyDi0qzxP8BhvJ6sKJtO+f6/OIgZ0ITYWakim7d3RrNV
0ditA/0XUvDgdEB0hm7iqR8FbwKNmS8DVKGs+CYrFwSBJ0vUH65WFapbdWbi2uwm
8CDKgSWpS16/PVr/ql84ePWdiVhHYmkkjuWPUFHSUcDiYL8YG9rnymw6Enx3Nyyr
ewiUOJFzWN6/u3O4x2M9ljrQQ1FmmAbw9R4KT/KHOyBC0W+xHbQ3RnJlZUJTRCBT
ZWN1cml0eSBPZmZpY2VyIDxzZWN1cml0eS1vZmZpY2VyQEZyZWVCU0Qub3JnPoha
BBMRAgAaBQI9a6RkBQsHAwIBAxUCAwMWAgECHgECF4AACgkQFdaIBMps37J/wQCf
epaHjByMpiZ4IJ7I5va1CwHjTkwAoIWSaoQOtqTuUupwzv4bNpPSBWbBiGAEExEK
ACAFCwcDAgEDFQIDAxYCAQIeAQIXgAUCUkUymwUJFVe2HAAKCRAV1ogEymzfspbW
AKCDvRJCLhfcdt+Rs8j6rxNZeaiPugCdGf70yZDhLmHuZJcirtNKShReca2IRgQT
EQIABgUCPWurSgAKCRCMMoz/FgbbldR2AJ9TkXexibjUd/bysiVJnNAxq3S2FwCg
j2AK9NlD1soRBvC0VVYiTbB5TxWInAQTAQEABgUCPWurawAKCRBVLh4uc9KIpYgV
A/9RhH/BsCRrvLRyTQgtXVFF0bZdKeZxvgxJZ/8tnAx+ZpDncwL6kdtFZsxmplWX
CshFKjCf2pG+YVPSnjtF0nlOgpLwbwcj29Un+2M1TZth9216WvBk++M4noyfj6vO
Rsvc7NQ5kuokLQsq8+gEyZM3OxbtDaDIQx8i6MFueG/PdokBHAQTAQEABgUCPWuv
```

```
bQAKCRBnwoCPM8Fie7k1B/95K8U9+QunBDYcYbn/afd195xb5TYUEeHV5Qs2RES1
f78CLE+95jnAno2XfPW9ip6Vk3bHD/66MT3ZuMQhk9BYzzLbEZnbKck9wiwhOhva
Y5C/5R4+AK6fcTfE9EKj45vSOG5PjkxKhIVh5PDXGAwtsIY3sujBMAbeDPK3IkAs
Ya5rC/5X11SCnofkZW/u6RM3Q44MWD0b149sueSvb9NKNk/0oi2HgBgsA6Ziodyy
y5b9QIdvNC+gOajstneVLCWahONnr9npAJse9fS90/SYMbH8/BcRpVHT6TG1Jwoe
6fWegEEDjwDAw021msQc9FUfW/FpJMZiou2cFXXP5vpSiEYEEhECAAYFAj1r6bYA
CgkQbGPaBITQ1+flVQCgxKicYCuXL7hT3Iz/i5YK8vyZy/YAnRpwkzbKaEMnVzMn
smeFMTxejsexiEYEExECAAYFAj1sgqQACgkQhDa3C3+GgmhvrgCfUEAGrporit4V
G/xddNf33zi5jFIAniCdksJJ0DhTekfIGxBOvwq0Nh1viEYEExECAAYFAj1tVWgA
CgkQObaG4P6BelCYDQCgnyVaUr7s/PJhJYpwi256vJ/Ha4EAn1KbR6Wc8JZzv6Bh
+iXkwP5fvgVgiQCVAwUQPdMiy01WKCF5BQwRAQFr5gP/XnIcMS41EP7ooB5Q0th7
QSBtLFCauRCoib7uKmOM5RfftQlSVqvnLOkrIFe/9a9iS9TGSUrphJF3dztcvtp3
0VzsgjZ59NRI7Nlg//FR4re0UFKf3gvHlyYaMd/hYc8M4NDPacAqoSjbMwyXAIKH
UrPYZj+ELOqaW2mWsRGNQUCInAQTAQIABgUCPd1MsAAKCRDhyUtG3x3UcTr0A/48
M61zbAKzmJ0kHZ+Q6ssXtBb1Cd8YSCUGqlr7i+LvyJgdB+ebyUzzXE8KOektIjxj
y0TtVTgsZE8YLW/EdoLKoa6Zw+BQKLuUxH/xEaz7la3kOwrXXOVnmA3zWrKzXFFT
aqaFf8dddCb5iJ/f0aafBNdocaUphSyoJZwyVo4Z2YhGBBMRAgAGBQJAkt8pAAoJ
EOuA3h2LcdOkAuMAoKCz8SumnyZCf1T1C3rMdbmbYNkBAJ9YCNF2x/dPRsxSeKbu
SmBaHlPYH4hFBBMRAgAGBQI/bZczAAoJEN68VxqalnmwSaMAn1l/3BJQzSdWwKd5
1a9x+kxActRyAJi6s5sCKTv9opL9Wj9rWrb9ZPj3iEYEExECAAYFAj9i2VEACgkQ
TyzT2CeTzy0btgCg3BceMu8hKtRCW16fAd0vtHoSp1sAnRLvJuDmypHO3mgvg3TE
Y7gYNsEziEYEExECAAYFAj9iF9kACgkQyIakK9Wy8PuqIACgmj3gpbfiPpkx/tG4
Yf7mxzq4juEAn3cLYlJwL+dcZ074wY2pXfd7cHNliEYEExECAAYFAj4cXQYACgkQ
2MO5UukaublYAQCggRDpTQCl7fpUnunAxxJzK0fOuW0AoIjRbboNnilOL2o4fmt1
TaCRxp0jiEYEExECAAYFAj4cWrwACgkQbZTbIaRBRXHJjACePSDDtm3jIlezrxKu
F+RRo4w+3OkAn1BqRElRGWu8HrG9xhO9dE0jIG0LiEYEEhECAAYFAkBR5+EACgkQ
OO1t8kNcHJqBnQCeIHpxX8Zg3ZYa6ivBJe5AKZiWd8MAn2np9GJi9BMTTLTX2sL8
ekC2lzjaiEYEEhECAAYFAkA+wrsACgkQqh18sBJn8JIccQCfTTO0ZIdIp5CAqlha
drSuNhU48ygAoMmB5mXUlcaV8ee/VlLqSupEqJ51iEYEEhECAAYFAj9h5/cACgkQ
2MoxcVugUsM2vQCgrsl5oItD6odM0TkfSYx273P7WCEAoJDyxJ7b3Xy67sOf1xPh
FXbwmKOaiEYEEhECAAYFAj8+BuMACgkQ2z/V939+MWMItwCfTZsRZJujvGiFYCT3
8JXYY10DOE4AoIPcHvW1WvgMLothdYGLuSADkWrSiEYEEhECAAYFAj67eiwACgkQ
72rIAB3Lz7eyEwCgv5WXThEaxz4fnvKpe8M9xC1qJhsAmgJxO/as8NajfegYmW4u
Gwxt772biEYEERECAAYFAkBHsmYACgkQMEuQSofRzg/UawCglF93sAXFzUFgrPAh
AJldt6l5fkUAmgKVuaJVV5em8kpR/5iz77WV2TsKiEUEEBECAAYFAkAiqvAACgkQ
1hDu5GjD2mW+VgCgxPsh6evF8lP5m9qOA8Dw03SOVOUAl0UuwwXoTlIi550tPdZ4
apxDpFyIRgQQEQIABgUCP+P6fgAKCRCT72NexbJb07I5AJ9g5j8LUeynnc8Qdaa3
2/ELvsgMxgCgmAilm0OaV+GI6VrWMwra+oy4CU6IRgQQEQIABgUCP33yEgAKCRDn
Qgt4utuOEaxTAKCMA67/PUI62JfKwUuow/6NL26W2gCgnMthTlLATPwz2tNeL+Ek
g3o+hF6IRgQQEQIABgUCP3HtvQAKCRBNrPLCwxI7HxuyAJ9/a0KffzpWw2g1lAT4
P7c9eD709wCgkcR0gypA4PPqNlmWlwAWycLb08+IRgQQEQIABgUCPuGahQAKCRDF
CSV+DzAaLn9JAKCWJ4Boe6dag2ukC6jFugnYcloLDACfd5nTjGj2mjJiv16rP6es
KAkz4luIRgQQEQIABgUCPmKk6wAKCRBuxMs5TCWbRzvoAJ4rBT4UpsnuSXHm+W7H
Z1D2wf1wrwCgy0ZnBUT3fFOd1VCDSJLUpSrbE8WIRgQQEQIABgUCPmKdCgAKCRDh
qZsPqaYP4ycuAJwL7n7cri4NzC9yvqW9Gl9nZptpUACfdIRzYqBamZbyO4rOa6Jl
uhH8JAaIRgQQEQIABgUCPlxY8wAKCRDnZkDIb0WuacbQAJ97zdAe3/5VX+d6A8vK
/keJbkKcJACfS7e2yRBp6i/VC5G7bqZvYZqxKYGIRgQQEQIABgUCPkf7+wAKCRB+
kn2FdkGYY2SPAJ0WZWC2fNfdq6cLZPkiWGVnLY/PuwCg+t143ijjKQj61IGoU3ch
hRkYl7yIRgQQEQIABgUCQDeVLQAKCRAqbV2p6xmZrAJJAJ0U+DCVqGxkdSAcfbgm
hkSxW9yuZACcDN88/I6tTSm3/l/occuN7mQGOiOIRgQTEQIABgUCQH+KJAAKCRA1
nhUBgs2eQzJKAJ9hLUeoJ1KhzvmSftxFIAdjoyaB+QCfQZ6UDZDksdUYES0HyZaM
LoEnNsSIRgQTEQIABgUCQSeEqwAKCRA/oN4IoNORaOFPAKCE7SBgOuugO9OOq7LX
nziKHN/PlACfS6tHZbyt4A5HQfwh8G6ay43/7N2IRgQTEQIABgUCQLYRzAAKCRD3
```

```
RQ1yObRVQXmtAJ9NP/a9aYNJ6oLx0rlmxIuK6q0ezgCdHg++QjH5OSeEpTYTBxUT
kJWkfDqIRgQTEQIABgUCQLIoXAAKCRB5A4OpdGbmU+7CAKCi8EmsmWPUHVYzCksh
JL6rjcC4IwCeNdpDKcYiOqjrBGmBXYLHN62FgYWIRgQTEQIABgUCQKrpNwAKCRAV
G6mUEXXC4zQ6AJwJjeizo5nnCZ0Qzeo4keC1XPkUpQCffz/qkjK07PRGEfP3FE8m
OuUYIyWIRgQTEQIABgUCQKqnpAAKCRAHYXOxkoTEUrHuAJ93vaIg6+mJX3HDSb2f
vuJQiR2AHgCdHd+yVP5sqYXGeZ+ragpL9sP6K7+IRgQTEQIABgUCQKqCfwAKCRAP
WuglNDguUY1NAJ4wcyUIfCQMXvQ7wqToAH2sJULFJACgqlUFO1GoXQKTvG4JGnfn
0A8D/WuIRgQTEQIABgUCQKjgbwAKCRBxzq+s7KKK22EwAJ42eTYNPJ2bEHL5bgnl
+sgxDCx6GQCfXIFX+AgG9d5TMVf7qj/JKDQXU9KIRgQTEQIABgUCQKjfpgAKCRCg
7/ngeafIcHhRAJ9Vt5ZVFnqstoF7PS+Sl9mybiDdgACggm566eWLJjlax7v4YgDV
P80r8l6IRgQTEQIABgUCQKhLnwAKCRD9/49Y5NtE8tVrAJ0UoRfpoYzGFafo5xOn
tCl6ijp4EQCg0tAVYXnuE4egIEPRB5vtTui5ZL6IRgQTEQIABgUCQKhLnAAKCRDS
D9QFytUJxv8/AJ9fThcbzZTiJv68+i9CrWeZdIUY9gCfbBZoHsaX3GoWQvVLXozg
UxQD1OiIRgQTEQIABgUCQKfupgAKCRB9vQuV7YwgQnJSAJ4mDNsLfr8rBJZgKaks
zvb9W7HRFgCgmFN1I64Cnjr/gET0a05XLSWpnN6IRgQTEQIABgUCQKd8nAAKCRDf
7jeUa+yYCrC+AJ423DDnUbT3auMicWgsBTRioFOHBACfe/773KoNw9MA+0NFygQx
KS0+WYmInAQTAQIABgUCQKnAwAAKCRAff6kIA1j8vdTTA/9UzhCtYCc4vFlD8KDp
m4jGxfGxy420u+VdytDMJdpKWxiGTH7mKq87KGKKzsRli7m/Aeeyy6qezw8LgHlc
AkC5H/438Qfy3gqc//KohzTCngp+lVh/A82q71e+aqM6Zdq/qpg4ZIcNyzKyIBN4
3MSsMVuZApPoR4ecyMXgdllt+ohGBBIRAgAGBQJA5EuHAAoJED8Szz1kFZUJw60A
niDmbeslCVAQZJNxJsxZs2E4kV0ZAJ9Qjjh5d1cm05cTAjjOzEV2SLXpbIhGBBIR
AgAGBQJAsOzKAAoJEH63kt8ZH82KwNsAoKkZuz+38bJ961/LczZF92x0hdxIAJ9Y
HM8/GzeWZG0zq9XHevuibrfdyYhFBBIRAgAGBQJAquvkAAoJEEuzpm9+s1JA58QA
l2BPvrAyoyIcMODMfz80XcD/V9QAnjqI53HdvHKEusjWgeBFz2LMFtiZiEYEEhEC
AAYFAkCo+K8ACgkQM4SDxAv8uX4H9ACeJhldCoPQ8jPLXLeFvoIFLI2M5OwAn14Q
+n+iH5pyXZUO4+crfghC1B4ViEYEEhECAAYFAkCoC7EACgkQDMt+/gswqTtE8wCe
OYv6sCNDH1LS081Io1+4WddTM2YAn0UXd+aIt17uSqgNJD+31mbpldEdiEYEEhEC
AAYFAkCn3TMACgkQSUWlN9d7Q/tBoACghHZaTNqIV03NVSPW94hifHXH6Z8AoIXZ
43KpCmgk4cFI8dzGauB2ggZriEYEEhECAAYFAkCm/AkACgkQF47idPgWcsWfhwCc
DdggxPA9FNa9CFUZeoRQgz7vrUgAoIIaIC3f5Ci/flk4LIpD+8OTkETJiEYEEREC
AAYFAkCm2DEACgkQ6SYtkGO1mF+woACfX9IlrfU63iR+LvT8RUO4whUug/QAn2c0
AlM5wsSQUVYnRl7E5KphSKNSiEoEEBECAAoFAkCm0/gDBQh4AAoJEPYcyLWu8zhH
NIkAn0xhy9EjBPURUFO6teiTB9wcwno6AKCGwl2XUa9TyrKcnpNlHR2nWhvEkYhG
BBARAgAGBQJAqukfAAoJEBUbqZQRdcLjAn0An0TdGpuJxIHokIf0VMI2kXLc/6fe
AJ9WF3rNR2/zc/fk9psqHpdb9W6ItIhGBBARAgAGBQJAqWMbAAoJEGlqm6oW1qT4
8HAAn0drFNBEJ/q970omFYwptBqNZ68JAJoCJ5wIzYKmtytEuc0fgmLOQR8/yohG
BBARAgAGBQI+eTKEAAoJEJAtvZGMOKkKnOQAniJX3xzZ7uWHHTnnSFVQ+cQIdHAE
AKCDOhDg8BFu+brIv63YgzvxGhJcU4hGBBARAgAGBQI/fWYfAAoJELcM/uw7sga/
ulMAn1/jrciw6qJ4Zzp9fXj4tNKkI3hFAKDfiJyUaUf0KJn2buZVpZQIzBsJ8IhG
BBARAgAGBQJAyTLtAAoJEEcxdTMMgeE8SCcAoMUQPwGijQMIEhOqYVKgpHtoJMJw
AKCRT4bUCO4RMGX2QZe9Wt59QUkBbIhGBBARAgAGBQJA5WgKAAoJEFhOU3zw+3u3
UKMAn1WW2WZBBmuhZSA+qxyPuKdRqVgSAJ9B4KhrMFFuzxiKFa14/4wMOIVZhIhG
BBARAgAGBQJB9Na3AAoJEKH3GNLIXe3AXWMAoJzU1RKakiSrTaDWGRk/Ly0zVr4s
AJ9pt+bsB+ArJTjUyrbkDwDpitDV1YhGBBARAgAGBQJCSIaFAAoJEGmo7Unq2nxZ
NokAnA8WM1WobqfbQ9xJbAZpneezHTf/AKC4kbDUDr+b0Dxr0tocT2Efk3yao4hG
BBARAgAGBQJCT17CAAoJEBi2sjIC/3GyOKkAnA1o/lqo3WdBR0aqj6Qmg9dHqyQP
AJ49/qJFJrNO5kwk8azN4CCJzGY0rohGBBARAgAGBQJCe2iZAAoJEPMxmA5OilAb
UgoAoNd6HscseNFee9fE8305ujhGfcuxAJ4nT1RlwoOEpcF6YRzbNxgz2pVe7ohG
BBARAgAGBQJCinGSAAoJEN+ig2JUF1no1NsAn1ZGfKRP2L7njOuzwOEW7swas6UW
AJ0Tf+IBf8fuuo2Ihc6Np1ze67Ti2IhGBBARAgAGBQJCqgu3AAoJEKK7Smn1q0T/
n6EAn32upJu7p8WWtYbR27LLKrXpl/H4AJ9QGsowCK3VyMAES4irU73T9BVtgYhG
BBARAgAGBQJC7B6hAAoJEINk48Y0qnRPlEkAoNXnLLGCNWILyMUvhxSXAyZ5xFs3
AKDqeqqPVWZxzgF0qa/GetzAYPCJNIhGBBARAgAGBQJDb3R6AAoJEK+1mC+KAcSn
```

```
tRUAn0kl7pUHCOU77xfrjLWvszlq97giAJ9hQHMzuEqrEnpP+JWLNTy1+rVCAohG
BBARAgAGBQJDzhwGAAoJEDl84qgJDKm0EzcAnitO6fkU1KmnC0hqcpDQCNzJT743
AKCOSf5lEeLQeRjJLAjWBLstOEM57IhGBBARAgAGBQJD4aXHAAoJEMMQ14pUoOQX
VzMAniGfPL9myk46V/ESjoD3HHp6rZxdAJ9kBWJirGJMf1xLR+P/1/xhQ6AVaIhG
BBARAgAGBQJEUlIyAAoJEAssGHlMQ+b1UGEAoJECFUozma7E9Asmq5/SfaxqNTvM
AJ9uDRNRY8cVU+jZe5IAdLX8mTlgr4hGBBARAgAGBQJEkExNAAoJEJjt8eIHzJ5l
vFkAnj/yQBZE3ozWTVkGpySNwhx9JshzAJ4j0UHi+FeuyM+/1zAuBUPJfSM664hG
BBARAgAGBQJE1vEOAAoJEKIjlRMQhVQt3nAAn3aZ+RIOG5GhLjpvqy1OOr178Q1A
AJsEQk83rTJdBmXp3L43RD3crzFr8YhGBBARAgAGBQJFFsqyAAoJEKctGR6SoTMk
ROkAn3X+A+3ztaJ3TzQb5zyMTzkb77RsAJ0Q/Iu+xxIksgIMKj6e/3YdMG+m9ohG
BBARAgAGBQJFVx0aAAoJEDDUOm5k6+Ig7XcAnjNDKUxQwTH7pXu3H/mZU2LoOLNt
AJwMcjCeNwq3QYhlq4RjZivS6SzLY4hGBBARAgAGBQJFm24aAAoJEFpDCyQ8LuZq
eMwAnjvYrEl0MYUWDJON71k3AE80KPhHAJ94HusVFkxy8AIshFd7ykyHxbqehIhG
BBARAgAGBQJF+uPGAAoJEKenCzN5XdlxXuYAn2cKkEBQPgl+/sK70vs2kR/sFuEP
AKCfo09LN9cWyUyGVlyMmOM/u4ApeYhGBBARAgAGBQJGuXwNAAoJEMNToZJ25W+2
52IAni2jS3152HJklqJuNaH5AcAp712vAJ900+YK76yeMjSkC6pXzQIPD3vHeIhG
BBARAgAGBQJGyEG/AAoJELNRWTiXjTGseW0AoKMx1NIg8v0QVOeJLerufrXyojqC
AJ9bX3re1+sLTyAGuKZtMNlJr2bcZIhGBBARAgAGBQJGyEOpAAoJEF+0x22hWnfu
TksAn3+KEMREtXdpGMAU/3Vslc99IGh1AJ0RgkjBif65Oo+sbpid256jODaNxIhG
BBARAgAGBQJJJhYoAAoJEAoQd4z8f0YhVh0AnRNd7kx3gy54FgDkMS59ogKnpMJf
AJ4jW9kd9CnDewdqQQvgA9qWwyjElYhGBBARAgAGBQJKL5R6AAoJEKgTSad+1XPT
1+oAn0z7hK2tQ3TP3zVMtX1BJNlHqiX2AKCVN4GMJ0GtfUV/Ro9IKITqbJ4OQohG
BBARAgAGBQJLIgzIAAoJELTjE/U2ZxFeSnUAnRbPqh79z2K+Y4CYvlFsSLRw91nK
AJ9BZUWrhuqN8WaU2Vy8KzHCsAi2fohGBBARAgAGBQJLtOjFAAoJEGfzMRpuD7SU
+UcAn2zoN6Mz7jsU74iaIdDGl8g5qVh1AJ9RrXqLxHQFxaoQ7Ho+dXVhlOogTIhG
BBARAgAGBQJMRpL4AAoJEE/BMsN8gQR2QAkAoJ1OvW07kFwhFv1WXEN2VFbd9xL/
AJ98ipsE2CmW49OYax8H9+RBG7VUYohGBBARAgAGBQJMZpp4AAoJEFfAdbIXpfJc
dM0AnjomwQIi4csicV/KXOYyFf5ZEBcxAKCCiKBcE15makROEnHazgLpIwYRBIhG
BBARAgAGBQJNbsEQAAoJEE0sDPzVimehi9kAnAma3cts1K0o7cO92WYjJ1VhUco2
AJ4oNnprsH9kB+REAJROg2tb0kEAMohGBBERAgAGBQJLSCKFAAoJEG09p+pjnF0Q
cwAAoMstTzBlHEm1iKoY5ZmyxmXeU6dpAKC3j0aZFfw8hMov+UsNIeUCBTdaqYhG
BBIRAgAGBQI/bSymAAoJEBj1A4AkwngCAZwAoLYGe6+hh1eT95+T2K6lhfZzV5lG
AJ9s/ytvHef5qt3I66rzLb+Evqwq5ohGBBIRAgAGBQI/bSyzAAoJECH5xbz3apv1
w4gAnRGfACThXTlxxtvEc0d1rPsl9V+EAKCz/8yOT+wlxpaxHgW0qt/XHREaOIhG
BBIRAgAGBQJCcRScAAoJEEsiCRufMca1I9kAniwU0GNZDVXzKslbVu4G1EKEHjWg
AJ4hupCGN5Cnzy6ELhc/cXKzux/MDohGBBIRAgAGBQJDHl/wAAoJEPW0eMZmqaUQ
2PgAnRc4o7Id2njS/f4R1JdOCJGdl17xAJwOcnGAwN6I7HSh4KZ51Ks4GnN8aYhG
BBIRAgAGBQJDVnhBAAoJEGbPHiVU26Rh3ngAnjlURkEL/3EqB0gyMgitGbaSm7us
AKDv1bQ25mYhwv8vwXI1fz5MUKfFTYhGBBMRAgAGBQJA4+GnAAoJEOgkW4kiRO2p
qQYAn0xAuwT0FaBtf2nBST6clBcxGyqmAJ9smzk/bOtKBuTKm6M+eTIeME3I/ohG
BBMRAgAGBQJBhLIpAAoJEL9L0OYEnbh5BpkAnA2rOpPzo3Cn53N15UT/4sGmfWrc
AKDoMHtpmWF192QJAmgRGsiSCUnrsYhGBBMRAgAGBQJBmwt2AAoJEMdeyVAbfju7
hQMAn3MCk6kP3/Gr80VFFEZdt+MMNPN2AJ9SRHkmWrbOHKR885L9nb2eENAVQohG
BBMRAgAGBQJBnWMMAAoJEL6YDgZWajXgkjkAn2dOeURnR4RH6ML4/viKf3F++Zpb
AJ4jqD7ftRCxLa27aV+74VtmnR1DB4hGBBMRAgAGBQJCqJF4AAoJEDIrCnSoXv2X
doIAnRskFgXun20T/BEKwFFIk/tdjaIqAJ0W76fYR68dV9DWhWYhkxlHQzEgvYhG
BBMRAgAGBQJC30tQAAoJEGuSvENlxpT3ZnkAn37o3ziLVtmCoSnzHn24LtQzNYmK
AJ9dXs8VxXJEP0Ka6DqPxML56EoYoohGBBMRAgAGBQJEu+5dAAoJEJki45vXY/+i
Z0cAn2IoDE1U25fF2v7fjvG8qxduHM2+AJ990FV84qxE4fS4g4kR1Ahel+tDr4hM
BBARAgAMBQJDuNMtBQMCCMCAAAoJEHSdKVBj61zIIlMAnR6I3IIh0EzwQHezKKHe
jHhVlcK1AJoCbUgOQ8m5nyHMF0bl0VaBGhMrH4haBBMRAgAaBQI9a6RkBQsHAwIB
AxUCAwMWAgECHgECF4AACgkQFdaIBMps37J/wQCgje4X7iqjNbVDgwpk+98vc+/H
oE4An1usSnfAlNcEcd+05ksTw1gPh+h2iF0EExECAB0CHgECF4AFAkNGu/0GCwkI
```

```
BwMCBBUCCAMEFgIDAQAKCRAV1ogEymzfsp7eAJ0dbFbiegRXFnp6X1a8B1eTDNdX
WgCeLmzXUp83gjnUnVrJ3sJUREreKVqIXgQQEQgABgUCTJ9xAgAKCRB/urM2KlaH
OGhXAP0X4sBAkxjxf5AcUrbFvyElsACYou25SILHiBMjVzbL6AD/TQpi3dqC01OP
lmSHD/0kADdJm8qI2QdJ6POqj4RTl0yJARwEEAECAAYFAkM9Z0gACgkQgdpXm5x3
8d1Blgf/VEE+rXnWGqITLcnvhNGWE221fd43dJZwWBfw8lkuPMXyRlI1jdVStON6
DMiCS9+Ex9c4nzyGmkKneqkyuhW04+DgCoKpbflAM9tLpTG2Cz43pLMYfiKTPY9Z
4MIlWT8bzpF9jP2YSOt7RSoJna8hiBr3NCxRsll/SZZ5q/bjO8W/aLHGh3VmQFMO
kdsYy5J7NGdv7oTYAnIzyuc3QLESHD80qaJAjrmR7r7clDPPRXfy4GbBI4ZtuRrk
49SdTfz0OM90pGOBPEaZuP+MRLeitfMnRlHirfCv8TMK+Dlk1yH/eYQdVVfeqK7j
XHOCmYWn9OUCjsnP49iUI2lIUHmng4kBHAQQAQIABgUCRGDOXAAKCRCJsz+f21Oa
7WcoB/4kqfVfFZs+i8IvLmibdDL/sR48/SCjE0KSnWyQna6yHpId4t1kRQtuIJSI
7Z7DHNZlfs8xZHFHYRBiot9nfA8GPxw4RhR7MJMEnrPByqEqmtOUGFrCbYfTkSxd
SGXB+2U7MNilkGEeTxyYQ9Pyd0C2eoLGJ3II/fCs4TSb277C0X7G1YDG3/yycdPq
o5grlvikaIFrnP9UsQd9MYFeCM4KUw5Wb+QkxVtElBChBT3KKlYex5wx4IAV913x
P0DfzkUGlpuh8vlyXWriUxJimjUzV6HCyYr+zt+dIaqSqvgDCsq1eNgNO+N1qinW
8BefBW5UNxU7oW2YOaSKdIcW36WhiQEcBBABAgAGBQJNeUGZAAoJEEjb1pAwnDBe
gPIIAIf0exOxFLlv2buqwnPbAwCQUk9+tV4NumL49fs0++JLZnvWs4TlC/llnkTD
aJkd+BSuO+rInccO431RXsm9a9OpjEFzyF5KhFjJicfnFLa1bJoQxsmcmVxEHU4B
OSvDLnEs3NAkYRwDriGNdTiLI9IoBwfYriLgXAVU/PJ+hYKtka5R+akpXEaM7w8X
d1cweXCwl9FSzMEEul1RzxHK1U+7NMYE8XlfPCh7efkR0Vm/07xyNR/tW0jHf6uq
Ioj2WGzW7mJIq006YzBMTFmIOpPHKDF7qFpGakZjTXx7ljB44A9gQXR+WUnJy35p
Ms/RGXxL3BDxyRNf9PBM5Eqrl9CJARwEEQECAAYFAlI8HXIACgkQRG9u+TkF4/0T
vwf+J+nTvymJJNIk5tOH5m1qCdF85xYej2Ey0W+QJgdomfOJ/qfpZZFXnVSInl6y
XhV24iuFC7VfNh1sHvkI8Mz5pOdUWn3wH8NNL+RD8KHK+YVcwjs/eZg6EtAlUfLm
77p9w4tXdsRfE35zGtmNRRGp/CfXolX0UNyEZoTnqyRVjp91PepkY6yOHeLtAcmj
c7+OUM/f/z2lCe/ZXbRTvx0yuE3YGKsvVyNocucSUI67S3KyVXgDz9Gr2CMehjv8
Xx81NmfY76IVpOLRxxSXG8pahCw6xclWQo2BmG17wrELvEoiNTK9kp4Mi+tArcbE
u3y/9wCnkfTnGeofLxlpZ9I+0YkBHAQSAQIABgUCTox57QAKCRB5eCsGL5NY+7+9
B/4y30T1N8UDAPyy4A9D69bfElvULMNaJNbOVP6FQ6eJWWvir1kCkvqVnh5hCfP9
+sF9sdEd4UvmvgIb9bQcPxmtROVbMhK7CouPDbJ3PcFgIewt8G4z6TfMgAbIbNIA
ySo963RI+Hx4hc7UWruMYG/i7OXcRdoVKK73ROO5zxt4XFtSNcst4ThcWxlaWO8B
QnRWYaJIVjgkp++q5fNPUK4Fq2iKq3h65TGtVjD2jdnu37gdSpu0SVLVHjs7jzK9
qzlrMjub4JH38hWmII8d9LJ39izvYxTSY+9Vb10/rD9NjR/J5o/JYkbtLP3s4lht
CyFpfj+VkzMxk9dr86HShplUiQEcBBIBAgAGBQJO9LVPAAoJEOgBcD7A/5N89iUI
AOg+F4XnFFQvMLYodkUJiwOYjw0I/7Z0hfrNKHIj6wUpQKUD3n/fTFCrX+DihQ2d
jrcUrIza2ZQwoRnTyA6zJWrADGqqPlPCJb5zangYwVAyc7+yH/qJtK2TqdVYSgo3
MjvpOJHrQLqlVpUl9nXg75XuCU9BjlMDL+i5BRDOy1TcHQUUbhhPmTmpdeQpxnWY
tuCF2L5IAm8DOH6zkeHNATR7yr+/z3/s97+H0SfxfdocSHUAtpAbeb/HfYzQg15H
SwpLnFg+otJVDaeMiT79jd2G3Jy9MZD0HkuFSiKJzDGA7zr/cO9g/R175WRqFyHS
4zhI4uuVgbZmERpWZ2yETsSJASIEEAECAAwFAkL+Q7QFAwASdQAACgkQlxC4m8pX
rXyxwwgAvjFEl/lyPAFPXTNzHbjGsp3iPo0DxSSHlqCgHA/zcP0veRCsWyDmJmpN
tFnmoCiT23aNtTe/iHhibLcQ8hPbR9oZOiLU/J1A5wvdmK5NqCocbzDPI1u8h72l
YIyvUWvpWGv8e6xnuQQvtX1uxUXK/DPDlHB76TrqVjKVT2CUQ+8vNtgovRE+PHB6
hCEVrtnzoEWGWopruWKBXmdAlqna9os6AWDcK9+KA02KJnALX2XBwPzHU3a3xLJb
aVfqfsIeiGwMQXdaXBHAozM/4exdh9srGmGkHqoA1OmYf7etUe3wwykLCvmhcfBV
dPYu8LYaNUhBvYrCgXqt7ZYrrarvaIkBIgQQAQIADAUCQw98LAUDABJ1AAAKCRCX
ELibyletfEGvB/9/yJIqGF1PrMXhIs7jAhBF5KEqvmvQxnGKQabfYSKciXwewiR9
aSrSrXqGn1lt4ABsc3wqgiFKZBCRfAl/3QrQj46n3gTaiO5FBz5MBJ1VpYUL01+D
JILKfwWT9BbQs9cZuVrLawbId4vBmXvG5EN6bUhVPTgpHRYx1V60v7bjs86c2/du
ExM69o+gL7oXXRgdBhYmkbTewV7uERCvvgrXLUgUN3vuJj1JxBFksFSzGLZ/9ABQ
gBpSbHJlwrX8cXRPvOqu0YoeLuS6Cn7iq/xxLkdSxyZAhsYPUqrteGKLfs4ixzV5
9M1Xu82eNGaWDfCSYHPy6Pdu/ZEkLKBtpqi/iQEiBBABAgAMBQJDEMykBQMAEnUA
AAoJEJcQuJvKV618m8UIAJnp1WA2XMJ5mZ9rNGKCTow4Zs+Fn+8PSWjD+DWxCIUD
```

```
AcMqZaUGIv+TJ2YwZ8YqteCAzV7dvr6yjQpNn4XLTcxyQAqFGR39QvyVC4D6u4rW
v+NKgRk2o2J0BooudbEGRk2gwsjzo09OZfaCwtiOgPw9a6Sy4rPjd+fjZYx4EWT4
6u30sUsXiv0gIrUjzd4WPVeLn0j5QPnnDKa199Ekpj0XP5O7YQZUy2Rbh8sCJQQc
z+ewzziodRUsV8cHb3re2wpGHImJCXvMrXZJ6r4aipB60h7SlN0zHW2m9NrNOiHa
by8shlZe1lStADhOQ5BNMy5xvjVn3cNTmUJWxmd4nRSJASIEEAECAAwFAkMSyAsF
AwASdQAACgkQlxC4m8pXrXwPQggAsN8MgCCA2p7+KLETSIsAxOJMi8Sit1+QftkP
f7uuay6BeCyljwuLazl5KiLMjiIx0NWZn9hKYnETvaJAcEFk6VM/4KKc3Q8r/WIX
bqeCqZySRSNYIKXpQcUw7+f++coEiiXK+nHJykWp0z7PmjIVOEiUMwjP/hkE+YYt
/XMOl9p3mIHfQc2zxGm+Te4N7PhBX7QqMMTLJjXB40ajssQDdndbov85ZgTxlOv8
+rygAOfjGX49X3PO5QexTp8dcQUUn1qXVbMe7m5YSBtIVRbi3uTLc0RTWCFyUwdN
GAiwdwKnRrjnQhSN8fmMJ7YMgLeNqICs4pc5pJszdsQbsh3m0YkBIgQQAQIADAUC
QxNwkwUDABJ1AAAKCRCXELibyletfIYLCADIJqYvAp33q9UJzKrhXheAVDlZaNs+
z1XGUUSY+GJJUZ4jlf7UzLdUD3mUzfOSmkkQGzkM8jTz2GzX7LX5EZ9vLGWJXCFa
RauLApB4SW55SELqe2JrUhDJ1GAxmCTOWsjGdVatiiT2tR67z6tnjpcF0neaJiMz
VCXlM9dC/f4odPM71w4e62nSRvDvVKoFVwqKp0Ihwa2PXZBH8+M8V5a/kFt/Jqt2
ooSDM/WVY+ttqbnivh8o4Qvdw4FF9vyJr+buehyW1PZzf7SCG+q+3CKntDo30FAU
nmLU8eZ30rbqPqL7QfwpYRqW5Dc1vUFMAYbrGrcrwbcePHTqhgQ6rs3kiQEiBBAB
AgAMBQJDFBmlBQMAEnUAAAoJEJcQuJvKV618fUQH/3ueYtS/qV6sDgdjLaCTMfNN
dEPFTOTmTMlM8HMb78bYMhtt7KhFQ3z0qbvAZCvawM92fXmUbxCj1rgX1FINPeIx
Gg5FVz1TQjBT3WBLnVvVQgyq+PbBypnrrY2KzcVSg2MG8SRm29PXzZ3Z32IjzW85
f6GkWjq4V+5a9TVI9QUbj14KpcqbVaLXk4HxAQcXtzhUhyxpNs3Z1PlfxPaosZa/
02f7ys9vSMqPZeTionI08p+AkKYjYRrywW2KkGsu7vu0ASWNIf9Bc2qX5SUyv/I2
+CASIiSs9Y6OIECRDWopphEfjWAVmAGOvVFR6xNnKrOXPITipK5TSyQveEjhjYqJ
ASIEEAECAAwFAkMWE5EFAwASdQAACgkQlxC4m8pXrXyAXggAi7kl+4WYEZZex2Hi
t8Q7xpZnkXe9bpRZD7b6Ms50qv5XkCWz2YVKi+IXECFNI3EqvMYq1J8cG3iN2eBz
weXR00WFMnVwDLgY0ijdDKsiuFlaoPXHCR1ql7LKavcHHZYiX5PwHo4EFgNT4adR
eK3tLJtO1h9Fsu+377VzwRWhgFXf0+MxHpr+gkpgSLyxt2zIAYKm5Ekl/OJm6UIM
Qre6yEyYQ5r768s6UsseJoQvxVjuYp7ZJCGFTcRuAp04QFzgRAg47J/GR+CPbwTv
iy0PcnGD1Ag7ZN0t5QC4/gF1kD2GEVDGVOj0RYzuHg1E7pElFgHRYze0a3XeYzrZ
CIEWOIkBIgQQAQIADAUCQxdlmgUDABJ1AAAKCRCXELibyletfHejB/9mY9hrnyeJ
6EPkJumm96b/xCdojboUJmz59aX49DhOaqBLd7lZ4XkyxFxM+n3siJOxjXVc+5hv
AEE4F0laVlVqHiL2wLkGS/tOWlmkQ5DiSQPyhjZuS5JMuBPR8Qza1IIpjEb/hW2T
HongeH0rdICj3ksTgB6ppL7D73BxocO0kHdzqnPVH+kpc68oRsC4OeNKom1Sg2R2
rgeeyTA8HMvgJGpjo3zH3InYonNqkpQG0VXdpAIg/H8KeU/G6nX7dHvMzxOrG4dj
Bik43iOUpnPUeRN8EpZjUMHkTFZ9OTvxUS5/MVAbs9++IHDm9PNX/r1FLxI7ry7D
9XTzaONXx6esiQEiBBABAgAMBQJDGLZNBQMAEnUAAAoJEJcQuJvKV618klIH/RYe
XwfWfXmDk9hwEhl7mx3Tp37MpBFHVg+xAMbJp9RLzr+pMS7bpjq3b0WWbXVwjuIS
UV7lnHwKrBDM3WtNhIWkQbSRhi+B3a6Ky4Hpug6gSvrHce3sOHNYCyNatL1Dgm6i
3lv7xs3NzlmSDNuftEU3Gp6Jv5X5vASjnSL9Y/E4xN3gfot/ltDW+H6SAdFde3z7
IAxXvl7wjxot3M1WNTLqmtLN2MRFsukyX6xVKVboNSMbY48lfcehDwv6uJXkwnO+
aK0fJ75fXRrHonBf1hiiFmer5Oi7WhwGPxlEjmxzKLlyUWmqp3uExuzRDbeXPgbB
IJ1Y3GR4kzY1zt8DLgKJASIEEAECAAwFAkMZX5AFAwASdQAACgkQlxC4m8pXrXxH
Gwf+NSKiOGVC0jpp3IjY5+pZWkCG8qvB34dQga2YxRcvOA0op4pKbIXXsrYmWveX
q7+iK5TSCS7iQyBlLendaNZ7y8dGSS0rxlMKvWePVKmZBXY9uYmJdDEeDcMfj519
wVd8pPmu4Snpc0opaj2gu3n4Dr4tatdvA3sB4AiCiXocDcEub7EsunQab4I0Qvin
j8ApouRMJSTC9Udytow7beh1p7t7SfyoLdlsI/1a4T3TE58jPk3eir/DEAmh88Xy
hXtRq7GBTx48wSa3lGRXfWPJW1ODYS4FViNyhRnZ8q3wkPEJxaHs58MkQqOwxGdP
B/p14HW3T+IV59ACStlXLdfg5IkBIgQQAQIADAUCQxtZCQUDABJ1AAAKCRCXELib
yletfE3PB/9TfKN/QBfAOAmeC1S9Wn6o0YB0WL0i16gtkrxavhvUCkWp29BOuCoe
d8wh2ir6mg9KQ0i/QGTS028slOeO8IVX113aElUN7UsFgP6Oqx/aKtdCO+8ZppGv
6G2QbvebgzPeHKO+UJ5tboA47c2NB+E6Zx4X5dGXQoL1KXSxcUHral9yB13+YBQc
nMZWll+Lb2J4d5L2xG80/qZECxKspsSfTXtUdwYpQ0EN94DiWOt3YIVk+Fxv8328
81RJFMvuR2jY/C8+Finw1BYkVloaXqpyBF4HclJQ+q6xRckv6CaR3pRAqfbSECCS
```

```
ZmGpSHKDztTK8wWhnsi8GGb18U321arQiQEiBBABAgAMBQJDG1kJBQMAEnUAAAoJ
EJcQuJvKV618Tc8H/1N8o39AF8A4CZ4LVL1afqjRgHRYvSLXqC2SvFq+G9QKRanb
0E64Kh7////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
////////////////////////////////////+JAZwEEAECAAYFAlF1oN0ACgkQ
jw7rxHtHFsmrwgv/aVGvQnxFX1BGQse85UTZig5GvslhktVGRcdBb86YKzsLxFRE
Pc8IOqItTSxBtvSTQEyQuYXMZfP1+iw1uQm+OyqP0cEipeo/fCcUXDjndMslHb2O
5jE1kqOGh3SvvQzUtS8Y6O/iKiR6urQFJYXGF4gkyvBRw9MyIf60HnSxM8QX8AMh
C3JOoDrTIhFLq8WFkrdPU37zvJet/k80+uFXL7vToO8AIvzynRKzuQLRn0DlFUDA
hWvvy+lXsquL9sGzyE9oOQDcBmkSArNpJ5zFi9g8p/45dvjHWcqNYNe35zq+7QG1
ctN7kjPvJNWFuFE0PwwQ/LCNwg2XYoOUDDnNQXhcZAa/eD5bUmJtD71AYIx0SlmM
X8xoCh83SD6qK/eML6gKSOMc2Kxoq7BEHG230/sjSqUrWHW31ikcPTxB1q8aSW1X
pSNpBcpVNYeJfCImEi/FH+pUY3ueIEt1B2NzuUTmopg7kYqQfEnOFYnNjdV1G4D3
VwwDdWhVVET7x8ltiQGcBBABAgAGBQJRdaDiAAoJEBrKdusyNTqzoLcMAJsY/oTM
zdqj6rAd0rLulQ8ZrTb5VFGS6bhFrv98h8mn/nJ7nM368A7F0GoxjAHTgvXndgxl
7+xuxCCBdcxT0/oFGwU7T1chvZ/MEa6ErXLsJb2jXpI/tXMSuwkhX4Tkza063v+D
yfjDDgIgoblUUSQzJsfrnAGniq1kXl7EdlMTjIRUHKfXLnOqdvWq2cloP0W7RzXC
YMC03w7nOUSbz4PBHHGPareNPz//wEAUeCIt4GcqPNh8n+zRrylklVebO4HMaVuf
r/6F66Q56En8DvyVw4NtGvuo5bZjhmrM6muAvVqHc8qnAb6fhM7VmT57smWRUTDF
2wJeOr6JyAz6A6rRwKI2WUrSgHjBENDJnPJmTeX3O3XTDcN4Y8JeswjsMDBkr5io
qEdwykEEudMstGVV3negPYRQtOiZVPiHSRnrrnKGLHDKtwiwwAl9NmGCFpHqwGNW
+hse1Ze2hFVqlU/EO61TQO/dqwQTmfx4QanKxAIkFxWRFpkbDGiUnzDsCYkCHAQQ
AQIABgUCTDQ2jgAKCRDn+Npl/acoMW3ZD/9YJQejYaKOHz1YAH+jV/BQY+b1X100
ERsA6RzvuAT8Bh5RB0yHlt1cVAjJQlEnEzJuTSwT2EKbDb6MBHtdLjl89PkbvdHF
wVE18h2k2bQWQUHprDQKPjhACA9+ZXdYhVrj4d/W+zsWaFSEs/s1pSf1l1lEDutG
UQD/bmBiIaM10YlKp8YhRBCGPduH5/4p5NY/oF/gfZWDx7/Fy5SYlQc6OpJIx0/7
2V0dNC/ScnaJUHXhLR1D88ste3l0o6fLycUBqfVS2eztvzslXXxNYInhUMuH0SYX
zjJCyjrNxUCrh0g+Npsa9zqmULrPFwtNy/p/7wFww5v5DPAEdzCEw2x19/zWTw3/
EYum6Mv/dL5uvVx1Sm9hIknSmvSgpH5EdBWiJHDjBTD7bkQ4JlGUlzgjqXIFTemc
8VgzL4D/yik+/rlKgh+UHL/CVXcbjr9zsGFb4auZ02koxyWJUUyl+ScDzLd3cq3v
ZDZZTMBep0p+EXJrmxDAgknUe1H+PKRHb16319TaH1+V9JThr1+BAcFqTLJVgb9t
jiDChXxWe0/pTO3LnGp0GLZqP8KPabwI3wFtYYQKBdJBmMfQe0nMOuzg8aREC7qj
uwTncXH4Eqe1I2xtZgfUDx9cWWqLnBktm9b7OVXJ8+7lI4q2PGmDjHgGUpAAjRKp
KphxXsQJwRNYHIkCHAQQAQIABgUCTDz4zAAKCRB0m//TuofodBMTD/9Eh7Sjxn9Y
Z4vwXwcIpEAdJeCrwstNXQFb2MHJere3Ee4sjGQRV+Y8Y8f6axqFVxrpksvcNMuH
ysTo42E7etUWa9I3ZBRbHdzbbKzukUT0Gn9pHSmmAKtmjfZPsIYDQtOeRfjLUgEl
QfGTnexZ66BPqPbORVOGqw7MAniMz3nTtSOkfwJ6TPqBZFx4p+U5spWgw1jdzxwy
bWjAwDC2jronE+ssH3xQc5lb06y6PKYU5bv1D1eMh54yNsC2/R0mCszAB4TuuS5E
314ZuTJwyLgdnZrYqFg5k/lhl90gbJyTa1tADg/HgtkUwiag2gQbB3BmFfxlWwLO
ZkUzEvTIVSv7YqLb0XfYJJfOJHCiE3IyeZvtkX6p9qTH9DgvEgSxLC3dpAWtQYxP
p4KhQMpLoE7EIARRH4dtAr7+y6t4fR54mPJRDmbndErO3+v5YjQvQSPxt+lwDBvm
0pERrWpv7znL4TEZ/e8WA1JDDk9ym5TE1S/QGet8J/psWOgfIXDBGZfMzIgWNzkn
7esnUqG9Y0InWTf+Xtbkg4hPZJGuirI9Ofuzs4IBxuBZVx8tng13fvkIjonNLQxU
lGvCThSAz4KngQvy2nkoeeMa0QPADyFMCSZpL6yWakvY1QcaJkodlv0eP39yca4J
RwIJTGC0aPXlmcrOczj7eNa9zwdvccMY14kCHAQQAQIABgUCTtVixgAKCRBZeIn2
zl348lzrEAChOE/xwQN/deypoRF9+hIF/PAiijRctv0SZ7sqEZnKCSApVOE0i4LJ
X9g0EOC2kxh0D1YBPBdojXMl1uFDRmHQKJX+s+eEGLVWcO1gQVQKxARLtaigdFxv
TqEl+NjDHXOdxY6nksurxvJAgMUk++U+4Taz8qRdjp8YW4nYBAfiAdjTFLl4ub5A
0l/PgK2KsYyxV4e9eVF6HsTPUyZF7sVMOxFxf3j3niRiadLAnzGRF3RkZndhOuEU
+P0qOiZQHSzE29Iu8Eze1PBYpbC6a1T1YNpPKGlPIn5ZbCi0i6o5ZAuyc85qyxBT
yuoYrvixOD5yE2y5edMo4CBd427+V7fX27UU4vH2Vv9Cp1F2YkCZagXpQPYDCE7R
```

```
Bi1wmeTx165YOtiic0ScuFaBEa9Tr4VkI5xv04KvEZVyYL//NubNU0ul7Xhrghvz
27s9aZQIFWVjlIm5iGPTFm9I5evqo0LsFvDmxIUk3qooNv5adrN17PAmFLVyQQew
sjrZSN9T6MjP0IpvoomVGtCE6IqzNLqc1pimt871BNHF2p3zHQxx2KfV0lPGz67m
lEKYafCNqJTF+hE9GNyr78++rQnJUK27Ig3RVM9IpQp8QOhRloiQsGQtD//CvSWs
h54c36K15Yl9g5fQ8HUYkbcetiub3uBr0mMdVa7wgYfF1/VZB8bPJYkCHAQQAQIA
BgUCT5TSEQAKCRCjOKb8j8gZbNT+EACmzXhlgI33H0I0PzbSviO4yM0Sa1STzTK0
M1lm51g3Uc43vstS0ruQKZjpLXf1SUhQVyV3QIalx/cQsqxBXonTLfGvEr5EOerz
xrGNRzSNRyV8KNpZ0yoYjvxHGjbYDAEM5HkdBOt0eSJM6Mn5f5W1MwB93YDCg4RE
0ElyvKEezsuhY5tl4nF9X2GeciJBRrd9tE4MdJaj/nj0xcDx4NXNEFF5XMdh10DI
S1ZDVTCFs75coEedoW4GtZAQg6WkKX4yqJ8Vk80tJd4h6IYNYXmATXcuPJ40F28p
WphBbSJmwwLSJO+N8zYTHpbrN0+PmC1WsflMmpy01hp2/+6WFpDgaWToYptZwvau
Apoh3Lo9BMb5+cZd+knogDLB9RGMy9AV9Lt987dKlXexHfwUbfRZzdXuH/vJLwbv
ziYvTXO3N1PXSNE7AQ787KVc8dIPx00RVJf52Es6QuhOFl2Eb2L7sgHDbDpyhwoq
feFKdkH6kWVg90t+uKVuGALTvI0q8xxVnljvD2L2vqZMrmdGzqiylnOPyYXWzYDr
jBJXGaI2bR9mTEE2TWb60ptlKBOpWHLJC0ta+AiHItaDJUs77LPHmozT3UxS6Xtr
Wl40owVS2BdnveYtbS6ShKWYfiwSFqvGxMO6zTiMMLSJqHlM+yVuFVUsF64sICEh
rBZr9ofWZYkCHAQQAQIABgUCURQ6HQAKCRD2xo8/nF8DuUr9D/9r0Mpt+5SV9h+t
l8sYHQQev6odjAQ2u5wG5G7p/2nXFYs3dRpO09up0foqdOPqLYPhe1PaIyp4KFu4
R13yeVbh4iq6hjrOO5ftADj3bBWQTGKlUCaeKXi1TRouoejgO55KBABcVY6PORru
Kz7LQ03ADZFwUkzrG/31yxT/LEU0uHljpmvEmp0OEIIMEjIxUiUF5hOoMhnH+hMH
ENliPGlJJ6H/bvqPvKhyFIusleRktHMEKfNvY8QO4dAcziIcOSo/Mbu9HGBoE+WD
8GDcJE3DdqYOY0uZpghr+V0yvejt0vej5KhbRLO2Cgyora9FO7KUyhcGjMvK3Ti7
3Fe43lr/B1C7ahUmdPBrXKKJMnlVWUC2+8fXvjVQm37/v3JzjAN7lky8041JVNHR
k9Ve1rgTBq4X6bYkHNzCXwb1o06Y4bQr8UiRFLrwue1yUzYp7xTQ4vpsaqkqVVTK
YpKoeURH2qbZ93GJDJV/Pkmn1+VH0TNGzLE/KRtIZy9536YvmZovxF9jVs4kVNIW
0Q17D5mqH64nlRDJgCpOPuTW2jSg0JNzidIY6zvf70BmjiaityHIAtPH8LFyQ2/g
SyzHZ4UJENEw5mMkZG3j7fXN0Pt/NhlQ/mhl7ZbCtQIBnA/3d1fVxBWHuojFqgyZ
IAgIxvlUSwtTEq+KTJlIwuj8N+QrMYkCHAQQAQIABgUCUV2QewAKCRC7m0cb+U6H
svcwD/9LEjuSGv4vfTU8pKOcIzrwKScq6xTgJ9wAyNZ+rC0PJjHHEhLw8j27qZAv
32uohHt3T0F7n/iRKHsCszTKUkYLWmUVBVaOl3y6HufZE7sMDJuidiPhSi11tOgM
t9ekySI1uNBfnLLR1rOUMbxjoG1NbdJIYjUPHCpArlKp6zd0lW+TQdgEQhsWxuM7
Pw3F79s1SVf7twjdj4NbWfnw2ByHX9HGuDytKuoEm/OgjHDapUpD4Ctc8K7l8WCa
ve4YkikxebC6K3C9NAyRmMup1wC1PRpzLD7UNkFXxo+Zt0bETuac6g8UCSR2vs7W
aw8ZiARcLaml3uJvadLP76TkvM86y7zdPWoxPySDtaXtAfEdsBwkmoSma7qK4c1R
IHUGj77TNDsJp4rJW54bFuWMcyY1KYfp5ofqmvXx7nU/7+SG1g/T/e4foBbEsTQ2
nUTMGeEQokHpsDc15a4fTNHL+Yn2ngeO8k1/AtRp6MWPQvVDGWYUO4HFYnihDmMc
6c5H/aKtGepdIqd7vct8Nfdn7ABgNyb5o7znQ02PsClf6G1pQP3erJ1ryywSwKKF
QiIwIL+n/SpAwGO8BfcOeVM3tkLTtnZdvvzaA+VS7407J7AKiyxOFyKdg2NN6woX
dY5YASI4EN9z3TvmEOLtqmCVzUCIN7fL6a50cCdicIeohqej0YkCHAQRAQIABgUC
Ujx7ZwAKCRAnffdJdtkFnwGBEACYxYy1VjQKp6cc5CQQgWju6xTacJjJYMc3nZkq
X8OSuBby+bXLAAaCp+6lhckdtmdOCsw33b7D5/S+GbXoeakxonhN0nNy+zKRz3tl
lNwtNtKgWcM0pJOSH3+X3fPcvQSXY+SMUOtCcBFgg1Xo8dWwIof+M+ZoBghiwZ6O
T2QDoTUUPL6chV1/6FqNK6SoIuIafWTqFOT3mFBhXEd3felA3njkMnsgpGS7XG4i
A+nLB9PmKdkPvz/QBH/zMitJr/JgRGHQCiynh36PkQ8bmZN2fBoviuTxJgTA4jT3
YYAaLQDJjyadl9680TYXs7QX81ZObV5pw9L15qt0locm+eYRpbjJyTreWzrHsglT
MvqOF5RgH1xDX2D3dPLFgZcrHU0uMhqzsHbI/DzTrI9rlkJ6jfbiSTEmn16GThuo
ONVUJ1M3KayFgmkPgYH6OKngwYVynhUSY5YN54MEowuITq7eXfh7Vu9ZhWDeY4yO
pTw/4qdbrZ7AlpaiN74SXvfvm42oyZG4XhLOD7Vnt8zSYvOiHzUx8ci/B88TiX9P
C35OPOi+zxh8Sl4V3o4CqwcQg9SSm040p/CcMJIkan8Ql8UUH/2TiYjXc/89Oi7M
8mH3AW3eSelP1y1zjm9RLdBMsPYUZ1LRTFSsyL8vswwei0554YMWSZCv4ANdm4V0
SYixoIkCHAQTAQIABgUCQZpZ5wAKCRCQnUi5NkQ5u21zEACvJPiTSJhdmKhYmC5O
BoZJwT3kxYhWB4Lr2wsmH9qI/DrnRaCKYVYu39mFWR7i+dQrQT4I0a2HpxRKZRrR
pVlEh0nPHUkgslUyUN4W5XiPW3IsBfBNIsWsDf0ROAHjzuBtThYHDxTyYd/EYwQs
```

```
4i2sNVkMbu9BV7s/HElQmlIFSCu51YWbOuq53/19Ma7HMJW4CiX223pWX38n5mhV
e3+mNTO+jSgASNRuDq3pXyuu9gOeUBptlCCfkM63W740kbzlwz/9dYHasV1BX/61
ZWwGGylSrWQaNwMnBCxIfERqfXrsvWd1Y8wmlmGJn8ZUhnpspYbnJkSlV8rKx7+c
JopdZkTv5bzVKGAD6/0nge3iOKzwXY7YdyoboA5HJDNk6vY+HSLYwzHeM1BA/VnJ
JoDlI3XsRDvHbTUcYwp4RGnIsZWNwyQbWEcsDqshkSUM8p8ODkOzmetEBILIDUPb
l0UaUF4gbRUc3Rh+3UiB+MdQiqaPoBx5sKVeUd4SQc958a+z7vx/HrSxP9R9Qpm9
UYZwrIa03CrZMaKEfqInOs74GiA4qkADgw9b8uGXgvpgWMTz5AGSLZqN6B290NDq
GOnhOIZl21UOHrCwEu/qdw+3NqUBfkcLrHqLU9dZ6rHxR4TwwrZ4/nkaA3hS0quO
d66/IDnomgSWtY0vU/AdmZCut4kCHAQTAQoABgUCUaVYaAAKCRAIh22TLlSrLFiw
EAC6UdDRPB/VJnX6Wkg0FB8Y077cQFwnB9gw5jBKP/1kV2nNSQFZJthuKfa8R30l
5pwUBChblOad6aW5cyV43P0n24B5FE9anRpjqX5I6eB04IW/km0Dfg5d5z0PRsSV
EExwLQyxvJJx3gQKvhjzuygWzEfsGSc11+Qie+GEdLr3oNA5EQZ87Jz2En3d8UtQ
Q4zoJfOHaRtjuWMnzp4dxRB0cK2CsZLSsA2/aCygkVkO6wxkchBQLm/m6/cVDqUu
kWpDtZKYxeKcYl+ypwwY1taSoH9XpX/w4zVlkHblesOvfrBtgj6/51YaRhyNOAAf
ijYIyG7n4uNWRrgPNfXbo02NscRg1f2ey9BffZxAzi2lrzyuU536p93wn9wLpUOJ
R06FQyG4DVUEi10iy+jsSwDweqiNJGY0euG14P3770HtdRGmxJkikJRS2evIzzve
/34q/M5cIHOHYzcVX1+207sqH6EGGjypAeHS44CMonXsyV5CK5habTkbxz+X5G03
JKmJEu0yA7udQBPLsmOcQ3yD7BCh28GU0+LNIIuqIa1+Hl/NDkLnvs+u7HsQuqyK
F3M1W4iNre0JANuEN3uU5SOXDKY+sqjrFd8C6ADgR1Mm0nB0LBbudVjrJtF7OYKo
erY/DhCI2t776MPsEY/zIPJQ7QkVXg9i+BQnduEGNB3w8IkCSAQQAQIAMgUCRVcc
NSsaaHR0cDovL3d3dy5wYWVwcy5jeC9ncGcvc2lnbmluZy1wb2xpY3kuYXNjAAoJ
ECZJ5ijF000FvsAQAI16T+yMp+Wif0qllqKzzRrmEvSJi0v4Yj/WEj7fMDj2OHlD
3FPUMm+rZ2pkC+U3ULbUx/qtsRBGHLI5ZUUuw3/wYbMF7L0wSfBcyiqoMu4PF8dS
3E0QS5Y8XpPkPB44daZksCLj8nsMO6cSnGJt70hD6tXYJ8L6Wn6pEeYQ7RvQVA3H
1W37/SwBR+fO8iYUNOhDBxWZI4PfQoDN6uHNvAGm+GOUL6xyMOs2urJQf3TNF0Ct
U486BDp79/XvUXLLqb1NUh0ynHYk4aCXytPBnYYo1QBlStl+u79r45WS1pjmO7fN
vdG8R807jGaUz4Wrv+PC8SPT/W+W3E2FprwOCiYo39FJLy+fd3wrA1hN2zjuiT6k
fS+Mqq8fiolUXC/GpOtDR9d0XC6h5ZjNb9vpYyScAmfFalrpw/y8h+d4tP9+LvmW
a0QMlH04xOq+o+L2jeEpu8aq+9TGRQx5MikojocyNj6Gn1vsBFlRxrHS9Zv4984V
KZaSQbFWYJAL9IxTYiVJxgrZ2g4JO6FINSQLTXLrtS8+m8qjIXUD0QZqUB/JYghp
DtRl4y1GkFg6COqPOEviWbCd/26EFtgKzZWt2x0ZfW/EZOTqGrB46RCol8igwEFO
BAaTaASTiNVyxWweHlAh+Uw+tHyLAZMn9sRrZ9j8jQQxT4Y2R44CRGidk74WiQIc
BBABCAAGBQJSTYTDAAoJECC3DeE/HR5PJWAP/jkgCkQspG7iIpEt525V8W5ikBP2
hlEtDpGqyAvT3HcCuxT01PwrtepaEvyOdtX5TY8h03T8vUesFSMXywNiwt5sTVVs
JJ4jqSRnPjVxdYpf+vk40qZuAs/JHykefrtHpBoIBshKniV2mmNZcbRMZaFvfIRE
7TbQjid+c2dZ/v0Nyy1fD1aRQh4+QgE83Tx3oSGduyiPe6uqenM0+w5duy/xpiKL
d+8JXZDxiRZ77Nr6CCNLlDJc9apNsrFpJCt6/5kw/Q+HIOBQOhc5XS5qkmhkkQGc
JELIr4WpJBGFafBS36/OdbOBUiCDtUL7Fmr7axxMdnKRQCUQV+YqiUaFZnq8tSxZ
0DEMYkQuDZ2Tkyd96QlmmMD9TKUJWpRl6AtiO0bXtK6p3Y1ntQSPE+t0tcOtV3zD
Bxkw8uCKU1iR47OkeIsiBGiSS2GebSS6sFPEC16N3VBgaR2Vn9nM92B1YPhGIkyV
bsJAIwV4T9eH5kerTCgkpaSvTcr2m7MdfIneKk4pLp3lH0rlhL2v2Rr0XbYVRF0Y
LKsZTCZVWl/kamAAQDcwQdW++8n49AWqsOdutpz5rJ5I0CMQAeDb+5e+ofA7Hm9Q
MmHKV2sN+S/DfYkqOvwRUVRmCCK8s20J1CoWEQZR+efi9D1M7OfjyjX9kSn/Nbd2
rttCRoVu6BFJqAfauQINBD1rpGkQCADyZeINQyOAsPXyOkPR5OOj8LaYIs+Iw3vm
KndG0lKD+JBQ+w+jzuHIC6js+tfZTLMylbDtTw815tadrUiT4yGrpJ6ieKW1FhIS
Ae/gpCtAfIp1W82N8w14dVPBDVKyAa8w5Bvdk1iEKkyLaNMt9YFVvkvB1DKgbyqL
ZpwYE6vCPgVsyPuCxqDJzg+e2cMHSiOlY59DSesAr0UcDYwVnxK0p5b/CaxXKCLA
Yk1EJIK7v5SWHOseItOwauS8+EfroNDbOfx9HK8AabdLyu2BB5gBhVIjSmbslAUl
s24mMYRGHnOryse9gCG56xsRWvL6Y7Jtfnzn8lUCXc6cZ69nXtqjAAMFB/9IUrUP
dUVEqdysECd1NBoJ1DtIcEZGRCbONC1pKcG5QGVt+iFAbXjZkHeykw2j3DR9jwRe
cHLtgIOvg7SKf8w1958ifZ2sPmq8yh3+b8qxiwBGqqyKJ65v9vb9U8pRYxqRXpjh
/SZyhxieqPeWUoLZeyWF0Q/70nxjc55zHqCZ65bLxAnMWLrTTvqhRm1aYLznrnzK
4VoPcnv2zTBrSTqjvxa2Zd5Eev02kyPN5WyuR7EqlEN81IyCMFDiisUYJQUMsuq6
```

```
g6eMwVdfFfZHa299ENt6lFh5l9uOJ0E2U5P1evVfbVtxsdYRJAuZFm87QS1gRxGG
ntG3oCE/8vveBHgtiEwEGBEKAAwFAlJFMokFCRVXthcACgkQFdaIBMps37IyOwCe
IAXR+JM4sHsiOw4tfniC2LAhmvAAoJ1w1Osdp1sKIp47wyBJOmQPuOtcmQINBFJB
jOYBEADuKnefrbTVFTZf9mITVx1lFAqwDHPRHZeWBr2Vq1B/Y1eKKsenBKbK/O/C
XaLuGFRn/6Ptvi9eLuWnho88qzaPU1Aa7BFRRiZlN+WrTmaDwdONJnJQp1LTPjqH
mLVAkD7mFZe/H8Glxot62zEqY7LrEs+ZuxQ8oI51YKjhGaACvkrFMinO09+TDey1
fupVH1+yskVKQZo1zp//Hl/IrPbZKfGCxIGePQowZF7YLvl8DKPo4jI5KO4tZ1kO
PcPL2CqwhuCDy0fpUhrQZBswp6tsGx5mRJxDxfgePRBYDK4tMK+BSVsRputIKOZ4
zoBf12hYFiJ8Yd7e9cqxTiPa7AhxPbAjppiH7qJ3NJKCXOOp9DcSvrfbymu9cbDI
PNwh/LQ1wt3T+U8QkD6a1a2kJL5+mdg03Ny+8Ej8hUyuJOEx+sxLs+JX4TS1KRre
LzxN7Ak21dNMr8361lB+Uprgi9lOBNLO31TWPABtJhIzwBOhohSqstB9w6I2ZsPp
LqUp/p9BrWlw6+UfOqNDFILZ0CqL1CyFIyrkjutXrUshqniSc/u1VbTURlIcufZh
N3FtW1P6ktUq5ss4dqEh/QZfR1WxBYRMbKXXAN61XO8M2t44I+44DHi7jOs1q6jr
bfAli1ZGYam/5wjOJkvQ3xemP6SaDKnCKOnPHC45EAt2SEVGywARAQABtDdGcmVl
QlNEIFNlY3VyaXR5IE9mZmljZXIgPHNlY3VyaXR5LW9mZmljZXJARnJlZUJTRC5v
cmc+iQI9BBMBCgAnBQJSQYzmAhsDBQkIB+1BBQsJCAcDBRUKCQgLBRYCAwEAAh4B
AheAAAoJEO1n7NZdz2rnKEkQAJWJ2ctNY7vg2pqrabavfRZ4UOWrLi4AgOMnKrsm
4ozZ1mc7NVMRj0Ve8jLLHrySW5QaSmp8TcaI6twxKD8FfTOFYjBU35DUliyRlcbZ
msBk7aG561TPwaK0XnF47RyPZWKbHrO7WgiDveGx52AmBdm2VRyMBwnue3b5RlKn
NVMMSm4RLmrolkL0SAZNAWZGG4FqFtaxPRZo7LR9fEv/NydQN91b2cR8SnLc2F2y
iVc5mq/1f/t8dMBEbNx2+NoFaqP1O+1JeGYgmA/vE9fk1oDnn1pHej8OhoJJ9SsQ
EuaITvzKP9bU+5/o/UqYzAX+y8QbTthjhzpkRwjqwjuMVmp6/f/o8ivlnzD5K1lQ
OP/OJAki63h5LDUC/JHYkT/XN/bbgoSNveFSGV7cdocdSpCoBaZUJ9pfzZpqRxyp
RB57f7bKBCI36E42KJKJ3wo873MJeElAeo31tXi2pBvTN/Idmrl6sDCNPWwgsIOm
u4Xd2FG5lanbTsXHKebCDPh/KK51mWra5judWWFVxChsNSwRHJACBXVa2fPsahfz
4GAEVp0/VbC114m8CHrgm3nh/ZAyNjgJQN5jJ37gQjx2LFsAhW5WKK8U0Es5YXff
jLEiNOnmJ+q8IZj6Mj5lWXkbCvrqjfNTOKnzzZGws+6y4gRQkgkSY3BPp+mpCQPj
ORc/iEYEEBEKAAYFAlJBjuoACgkQFdaIBMps37Jv6QCeJjxijseWZzn/z7Cv3zSw
SFMAWPwAnig7ZgzoqKqwpvnwAXsQpGSnE8K5iQIcBBABCgAGBQJSQZHeAAoJELI
Q0VtpqZu8r8P/jHm+xi5yMz3DVj6emMazJdXLtnnGrKTNw5xL1X10a1Rvmo+sj4J
1gmL+Cy2hM6fl6r054E/BYt9GVGaIC4eYiF6DUzlcPWkwniDKfi1lNJzNIja4qha
nuGrK7EJtZXACRhUuNr2EzEm4dd3nXNaBQZv9FlIn79tk4vVho7wK7uiIT7nseUM
WDh7T0h4IVSs2LWdvP71WDx8acoyfspI35C2pKXB5GRWxnzN+wOl+V0kDn2fGd+n
L7ZEb/c/01h6AfyYJGetCXY1omkXSzgD9KKu/RqZuxL8TMMjNN6z4SAyMTthOHW0
lTK/5h55dJYSquBQwuEAX0Z8RT8S4Nva5LKGr25IpIJuP/TxaHIgdncrin4D0Ftu
G0JMOxjuzNdo2lOiMZ/lqZ75l61C68GuKAhU2Rn1toqc/NReL1yLhHoM1o3EvovA
fZmzX3sOugU2N8L+oiTnFFXezpY5Huup5KUkrX+C5EErBIVfvKjNyhhKFru6Jwy9
z3qiGhxNUFAAzftVYhNT1lDkMNqa4jPjOrcWS6+gwVfQAo9k0p5uwPNbIw59RA2q
/wwhZuRoai4nqN9WkgnwmWn0sS9XO87jwN3uvK0IF97MGPSXNcmAGXlxzF3GBFHY
f/bpagrvT4v+DE+gLpgfplo86oZbjDPsXGhVNu1iffC64R+vecw7r3DiiQIcBBAB
AgAGBQJSRaaeAAoJECZJ5ijF000F4jIP+weCFBeCkY7sprDa61kp10GNF4YujiZ1
QKQDgrQA9ipgv3pN+5ovC/ClzZm5baVGi+j5zWD/blG9YZAApM/kkpAIvCPYIuQ9
b+/crOUjuxyywuE2HSbaFuh66lW7Eox3NT8NNMEl6Zry6m8RDHqTZIpwJPBiCgEc
Nqr/dcbtE0XgzJj94NOWSuq1URpP4wIT9aAVBqdj+0KQDkDk6Sqvmf59Cjt8hihv
XAhOqcguKo8y262ABEO8kxwfqvRYECCE+eDEAPUEyOi/6uI0dQjQMytTWKogPIYg
4wQjpG+Pa7wl7AnxOTBp4WvoS0BuCgjSYaxnwVKHBMvxSCuDHBurLN0wqOaKSg9i
b6m/Vy2vfi9ak8crXJFZ6eLrIxt73gyiozfKEfvd6LBOJ9AeXstnubEs7ltNq9qK
yW4+vR9eABmn/wABxCsHNjW+mmi8xAVhhc1KqZC/D4vm6r8ZwrVAsmTADqcTr6A4
8J15FmIwcaQRQWQ4oytxTGA7rHRFVjrt3YIj/WP62byp8s59HOKJE+mA9q7ksAvn
ToLfrMiNA8/18Zm4CADKUny6GLzpuKgcYwTucqE/zBWUszI2NrJNtaKWafdXyEAw
gBxNIl1FiYF9+ntoMWlqDQROPZLYChRThJvRnNNsT+WwcuSHSFexLl14yrPJ3MBE
e7e+2Vpj9HR2iQIcBBABAgAGBQJSSFmrAAoJEDpFFvNRg85IHx8P/3exX3fATzNw
qfININlvYjxMzuGIHdV03w2pHrOllmPX28/UUHSQL9yRRNhzimm/9v3dvu5XHzjU
```

```
zCEozoAa74DnICe8wUfju8sGmN5FKolbvSz7VvcW4mAC5RY85zk+7luTg2wHZIId
girTDrgPSirtYkm+qpuX/k5LAkwmYtH6gghqv7rnYNKUChh+Ga+4yNbsdD7blWYr
52UwnfT3evbgI5GqBMZEbghmqNiR2fcII6trNnuawH646UcucwogxPtLxLuZnslE
pWiHQlAVvHlrCMoEkYqS+NRXOwZF04zTwRpLCUlj0PxlRInvTrEpBd1KVejbkNWK
K7wfyL/bF3rR9pMGWuDC32/9BfjtGgNDXJhQMDGntyAeQfiI3Ml5b5SA8bT5DsR/
FIQDg0UDe5jjeVIEGZKunmRT/IqOLFMpZoMHqNqWW8YrHlpN2o2c0/VqWSLzPKmo
cgqLwlkx5oqvn/F12xUzazGhFTFp6IXpqQVTlkSPdDsVJuidj9ZJLMRoKfFD9tIS
qTocGw3suLqp8u5KZf43THWspBi4tD4IoN5rlrLWtPnkteffyO62NZOOyg7rPUGJ
YlpgAMIDkXmsp58CyXqrL1/art0Ymcy5z8ea1eUCnq/ZJJxrj+HrXuwko4fXTewf
+nzSbJ2GEL/fMBkzAOKl9j5bOPAKwiD9iEYEEBECAAYFAlJKlYkACgkQ20zMSyow
1ymmfwCeLqsUDHBH8JnuaJjEUYqACGWZo88An0wcNy95yGdSJtgBFXNPZQJL2gSu
iF4EEBEIAAYFAlJNSA0ACgkQUYUJaGx+XoKvBAD/bUBqzL0oZtaF7WUDXchb4yki
f0ko+zh832R2Ad0KfygBAKNEUUKOnZFLJ8GZqAXmIWktgMiWFOMSxAXDLsyionoh
iQEcBBABCAAGBQJSRqY/AAoJEFF75hSlwe7HvwsIAJUnlLFMOBLvlBrRuxVeAO6X
8DhytdD5YlRzt866cXq6A/dw57O9qwyyDy3upJIGRy6hYlL18ngGZXv5djcw7Rch
QmvBJ9ROkmkCHLe3+fYn668nkxtgQJHWADd90MGFHkLDWa4Pbu5yJKqkTy3tqx2N
mBDEz317F6mMtyTP56QI8PVnh1p6w0McQIVctS3LOC3u4Wjbw7l3Hwof9Pl3u4BZ
L/gJz5KAozUa5TqNV4SLwtUqXBg7kipwfshXVuQekG9XfMC84GaFMqEKTExscHoF
VdSzrBKHn6VlEl1sdhcdS9aKSOsqMXB25xhBe0hOl4Ddw63j7b47XCqcyqAE5eiJ
AhwEEAEIAAYFAlJHAsIACgkQ8cUWs8g1l1OXkhAAvXUR237vXF/sZCZgG0748Dp0
eOhish/c4ODgW3JRehVWAyAlTAit/+xK6oI5xkQA+z3KO6+/bAtnDQgikAkykgpt
VeVW/6v4GGBarUTc/CTcofEpC3rsrEm1ZwPLyva3YuFFnYHATq/2Qi1a5PnSfj5C
O3fZrOgJTXsm6eNt21bH7RYF4DYi4kDNQHxtBOaEcUhcIkS1MsMz5F+/YeqOd12/
FrcIPDq8c0G3Ol+QsHFx+Y6b5Fp/HgkQem9Pzu7XkNcf7nj5UFJw+qx+BivaVYhJ
8Ugq3pXYkNkhYSy/AP/YYp7moOgpo2tY5e+fqho4pVlrHoPqWTNKJJrfYg2Mg/vP
e0nPxiCU3anmFXhfeZy87QLrA2BrO0I45StbU3uBhzT1dfNW2BIgxg+LqUZyTrZ2
qHq8TOPsnplu5Xn/UjEDQ5soTq1zDpslEjCX36R8wL3eai74HUTjstF4xq+kiXmK
bX7HhGKD9TILRjU+toOPXY0ffbS7FOUijLqOJqWEW1nBpoYoHbGfMHn2g2rNFGzz
wiLZgbL2HZsC+kDoog33s60b//A9E3yFIIiPtk668kQmiobs9Iel3RC+eOdHP8lD
gcMN/Rc/5B1S9a+wYC8VTf6KInUTq5YwC0veKbg1s+Ow7tB9ejqgxtHT7iFjR5NB
oOpVkI4UtHDpewRAW9SJAhwEEAEIAAYFAlJIEEoACgkQi+h5sChzHhzyGQ//e6o3
y+pnFTS4UWjUxFTKCtqJeqtS84jvcbXhXFGKfnXX15atLYkVoD2LcO5yvrFRNvY6
PjRkxJmLo2Lb/MpoDupRMfR1PxotFYuNYodmoHxVUun+1eIFQ5XUSiQSsIsjcUYd
EcOoZFzMfWIHZUOA1cGAtb8WL/Ql6cLcZT3fhPjEO253O8XcxKmU7sJ1sCCh3tyL
CY0dvLffA0jgxEXUYmf3DpC6p+MNkPU3EDk60OUzy4/C2HT26Lt4NR6TNcEZg6O/
lPvmD1/ATO9fAHCb4uEIkqR3VLdeg31EHND32gO/2HXc4Xp2dbV8qs+ts13w5L26
D+94PSsTwYF+85mfgu8nBhPOOn7lqWxIO/1MnOrEIVNu+K/fwh4lu8v/6PJYEYIn
LtYkDH3/LcKTsK6N/2KLbtROlHXeNKXyt0UliINteDlV9xYkn6TtzUcTrZ4Xa3HM
yN5mi+a0vptJFBPxyonMMHDAXRkLR8BexxUJqdk2aupIs0Y0Cet6Vk+8Q9bn04gl
pKjTjnnarJJsTlhrdmVobkDhbEGYB3KyrjZp2JmdYYzAbHXbdp3T7yJ4R3/7aQRg
XJIQgEHjmgFf0Wwzxs1JIN2URDZS8k2pyuI6M8ndPtJiYbwqy1Wcflz57aWYAOVf
b/G4IEsicSd1mHjYjsaMV/kp1kGrWihB/Dt79nWJAhwEEwECAAYFAlJJfnUACgkQ
cTWO1j93QHkxbA//SKb0a0wo5dTJpMp7pUL4pkCx1gR3YCZMyiJHAGnC0vHoTmxI
+6+YAU9DBFWjQk2uqqn+GW+3AxLEN08s2xYvNoxJHUB1bF43HI9lXscGmzfjDR62
cIptcWtggeMw6W66UStdFWUudwDM6WV8BTxg2LYD3upeY69GnN92HinMj90D6PMc
iQjfUdZxZAYLKEhic12dKHpWRC0PH9NIAS0EchARkZQmjyPc4trWevAyhmpqdw+H
gxh9EBH2I194SvIXVuU5Gyl/l3a/6ntEUZnitBijU3uUjRnkS5XkJfqy1MjdrJ0o
ymo8mlxOVFKV879ez10KBnE1BLe9ioylOeGQRNcyYehFE7GmzkZHbOk+Pqd1Meaf
AjNIgQxrqgh8pJ2F8Zd8pGDrYspjICGbbdR0WRNcoN4kckJruTWFQ1xr//Kfwp1b
kCQWRwYcRL/RNVVZuHGgvTiTa2wZNbWfZk3tF9cXaYHIqhYU8l7Lc1zK0Fhv2E1t
Phw4pu495RbGRAFOE14S+QmknIy+DgIkTzQ1s36vnI4SVw9zs0D4Np6d1mF1p4gi
VVrgTQnlF3poZNppCUK9Rih8s5kMnyuRruGm/Lod4jL3wcbBz4sxBkCgrc2pyU1M
SNAjM2V8c7cGLgPOqX0eVqgXJoTnlNItF07aIZyFEA6e7YeiTeXxPfU10Q2JAhwE
```

```
EAEIAAYFAlJNhQYACgkQILcN4T8dHk8Ifg/+JzwtYSnxoksuU5H4NIH0fchwRLfq
6VAscqaZYxz/KxH9suEaEGoXxMzeHO91OqPqnvMxkpOGEopUssHGOVXYwtw5XCEL
NCjD8PwSlSpDDe5+lYNjMIjtIXieiGt6ZeOO/0VlVXzRCHEtKoN96ikgEaxkPq/m
ZmfQK1PSEFcPWujBxlWjZl2DHv8eAvgFEfX1kyIoxV2nfrllDMaVFU1NvDB+zXdR
Mg1xyEDiCBsldfmHmhSjylunfJeyjpwye65rAVEO7XkmNBy2SloPIHRCiFLPeLku
oD3XaRFHWsRCOBcfwZy519DrvUUpn5InuXB36zu91Qwh8Bd+UJQIowsBoU9AH8n9
lPsUTCU7dl4UqtZxiaTHQB/3+J4o7+m12I5/Y5ftW8ToCRF5EGKoB4r0zhp2BLWG
e+z5B08HjR1NcQVG6Tv6FwSqpqf5m4yFaiEmUCFMfFMXxXGXSjI3JeJeImKYkxOw
aa8XbH65D9Lj7syDz4DSgZiPC+cUL7SNY73YjH0zfL66nGRzwo4zX9T2ermnvCN7
kw11wIfJVOlLG+D9sNpBiikpMPppW73i7g6VuFReSLgmdNCXCoVWQYeelb4E9ulD
KqFj63VChpRaBEv6fz5YFUKqUVAXy8iiptgY/hbF5V/KVlN9JVYOglQ8oq9sSzKG
lUWPldPvS8nzroSJAhwEEAECAAYFAlJNN0YACgkQTaEU5cSi5X+5JA/+L/Ilu9WT
FeVZmGTYkWEOllp7B0tNQKSCwN5L0zt917Vj81udXBKb9O3PKwjpc9rmUh5dRNOV
vAaIj9moU8NoOm1SYvnVvdyAxF5bajnN8u2cNlkdg+fzCiwwUyGPbCQ5elC/sM4k
FB/kw2c8e5uUHBjTmjh18MEqLQYpVaXxmQcica9EQnDvAXq6Ri2dZA4hpb/+qZXC
iS/fojYQmiigV2XugWFr4+rWfcOFACCKWfr8zP/3p+fs29i91tCUwaW49EA1W4qN
8/3NCugXwGaFZBsQdkZotP4WwPToD8KNaUqRvdiz83TAOL2RDZ7P8NmGNeAExeEm
t0+Z5MQeudfvTUCb7YMJKnPttuQ5rIgsLmDHwNariGMa7km0ZykkTgCw3r11efiv
/DwhR5ygZkb8KNVDIBxHGwhz2c4mbNsmRAas/wDboijT/GvA6NTaAaRhH4RpHej1
Bry1j+5mlhve3fKH5vQ+qfyks4yemjXq4meLf+0hj+SKoGcYXnfJUuOV6TB85FcF
EVncY2uh7bU5et9sdDv0HK0yNMGxBqF9ox0VkXVAg28Q10n49CGHHtHaDzTLGPru
hrQX9bTN2pgNticzZu0zDz2a/+rV/TGZ21pMlfPpmks9jcy0NYIn8twoMpRCqfJL
teKUP3kd0WdyT0Y8pB4X+aCzliVB5BmDxxKISgQQEQoACgUCUk3NEAMFAXgACgkQ
OfuToMruuMAgxQCfScnmgUcnT0J07KNsLKLMGW/6ffAAn2J50o8KV/wu8auCY1o6
EkjpiJt/uQINBFJBjOYBEAC2oNVWMm9p1UwMmKl7srU84rhC1wWzCIpgDBzQk6Q5
4zS0OKuis/zr2B0e2S4qvd8S5bSu0h3k54CNIIj00iKFfSvQDaInU+t2GGV6hXsI
XS7QPFNUCj9n0dKa5BahPPfOvTVdfJvulLMlvygYwsYW5DhfXI1FnD/R1oY3eNib
FFYsmP7++VRrO/O3wvbgl0kng8RndM1M46imFkOOfPEYxHbp30VvcxX2QJwEiki/
d7UjwgonKKCaU5SoZEKa6/oIwAMzQ1YQZEGQBSD5iM/sLblBHsO0UtLKiuknZBdR
rHYHCDwXZvX7nil9dtA7bydhGzPLT/JKKtiNqGtP4uIo6Ao3kctOfq8hv8pmCZo9
HgXVxUlg+OXEOJu7bqREiUcEhm5gn12JlKmb+6anhlfLlHzjU2OgZkGkgWx+biiv
Tacu7ESh/qSHLYrWX1Y7xT0CMbTlrM1CEMaKO/gYHgpd+cvENbnWrw9laY/HAESL
uZBuH93YPKrNUuchCJRYyTg44IHdUQdbNLSww5/00EdY4LyOGUdqT3PMHeo2wnrH
UNcgcLd/gPyjAUCrLrPYFWQpDKzubFfNyJO/JgiqtvnKdG2wsvYYx2fU14wXOIHn
XIxqT0EMwYbKZc5tjcaaDbaAXCdv5kHH6s0Aa3hHeeCT78LSN5cfIZA2ezrDCgLK
4wARAQABiQIlBBgBCgAPBQJSQYzmAhsMBQkIB+1BAAoJEO1n7NZdz2rn4csP/3gl
2XgdJvZsDo3WT5KdqO/LsLbEJLoak4wiQNoij4CjB7zmLFwl6qI0ziUGvw4YyoB7
bPRwyzgG88e502Y3/hx4GzHBSeZwKvWEmIRpCvh4BH+UML+nPqC+QKd0MpJ46+Dk
WKJcip/qxNeky7h65ptA7jjzmhtIFoXv/fM5R87dG1p3DSHMRy/9dqIJOgDx/AYU
2MaECaX87u5o+YAjet6XgcwQc3EiCoBEyJg2YU/ydWAmLs6rPqu/rn8T2yG01VCI
cGARcZl/+WyvEGxAmyAbZWP6CCQNk9fkB9PsoJXhSse0z51ffIpvCJbCiw/AqaDN
jFHmpfolnICv7vZmzn95vno0YQZQlgouZYl3znMJAdNmKsWwMi5mzzuhh2sNiYWv
ChaajFmpIt4EI1tRG78Fs7ieclbOvd/CWpY7os87usJp9Qrr+Z1g8m3gKmYN7ega
e10/9RUDXRlDupZgdPM0raF4Gbg0djRAwFdigATlscwIOc1hU3hBFXFTKOxcp+CM
7KLSNkdf738IeKEhdoKo4jgx0vBHHt1TCGgo63nX39aWHvXDSq+D2RW3rcDsS3Kv
vGP8g+kQZREN8P8SFdefSh99Yvz4EpwtinVNun2Al7cBv8XdU5a5p8yWk434iLhg
R6bnoCX8SLywMD4E3tynDujld/4cAbvQJ1xEOftW
=Ba2T
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.2. セキュリティチーム書記 <secteam-secretary@FreeBSD.org >

```
```

```
pub   4096R/3CB2EAFCC3D6C666 2013-09-24 [expires: 2018-01-01]
      Key fingerprint = FA97 AA04 4DF9 0969 D5EF  4ADA 3CB2 EAFC ↺
C3D6 C666
uid                          FreeBSD Security Team Secretary ↺
<secteam-secretary@FreeBSD.org>
sub   4096R/509B26612335EB65 2013-09-24 [expires: 2018-01-01]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFJBjIIBEADadvvpXSkdnBOGV2xcsFwBBcSwAdryWuLk6v2VxjwsPcY6Lwqz
NAZr2Ox1BaSgX7106Psa6v9si8nxoOtMc5BCM/ps/fmedFU48YtqOTGF+utxvACg
Ou6SKintEMUa1eoPcww1jzDZ3mxx49bQaNAJLjVxeiAZoYHe9loTe1fxsprCONnx
Era1hrI+YA2KjMWDORcwa0sSXRCI3V+b4PUnbMUOQa3fFVUriM4QjjUBU6hW0Ub0
GDPcZq45nd7PoPPtb3/EauaYfk/zdx8Xt0OmuKTi9/vMkvB09AEUyShbyzoebaKH
dKtXlzyAPCZoH9dihFM67rhUg4umckFLc8vc5P2tNblwYrnhgL8ymUaOIjZB/fOi
Z2OZLVCiDeHNjjK3VZ6jLAiPyiYTG1Hrk9E8NaZDeUgIb9X/K06JXVBQIKNSGfX5
LLp/j2wr+Kbg3QtEBkcStlUGBOzfcbhKpE2nySnuIyspfDb/6JbhD/qYqMJerX0T
d5ekkJ1tXtM6aX2iTXgZ8cqv+5gyouEF5akrkLi1ySgZetQfjm+zhy/1x/NjGd0u
35QbUye7sTbfSimwzCXKIIpy06zIO4iNA0P/vgG4v7ydjMvXsW8FRULSecDT19Gq
xOZGfSPVrSRSAhgNxHzwUivxJbr05NNdwhJSbx9m57naXouLfvVPAMeJYwARAQAB
tD9GcmVlQlNEIFNlY3VyaXR5IFRlYW0gU2VjcmV0YXJ5IDxzZWN0ZWFtLXNlY3Jl
dGFyeUBGcmVlQlNELm9yZz6JAj0EEwEKACcFAlJBjIICGwMFCQgH7b8FCwkIBwMF
FQoJCAsFFgIDAQACHgECF4AACgkQPLLq/MPWxmYt8Q/+IfFhPIbqglh4rwFzgR58
8YonMZcq+5Op3qiUBh6tE6yRz6VEqBqTahyCQGIk4xGzrHSIOIj2e6gEk5a4zYtf
0jNJprk3pxu2Og05USJmd8lPSbyBF20FVm5W0dhWMKHagL5dGS8zInlwRYxr6mMi
UuJjj+2Hm3PoUNGAwL1SH2BVOeAeudtzu80vAlbRlujYVmjIDn/dWVjqnWgEBNHT
SD+WpA3yW4mBJyxWil0sAJQbTlt5EM/XPORVZ2tvETxJIrXea/Sda9mFwvJ02pJn
gHi6TGyOYydmbu0ob9Ma9AvUrRlxv8V9eN7eZUtvNa6n+IT8WEJj2+snJlO4SpHL
D3Z+l7zwfYeM8FOdzGZdVFgxeyBU7t3AnPjYfHmoneqgLcCO0nJDKq/98ohz5T9i
FbNR/vtLaEiYFBeX3C9Ee96pP6BU26BXhw+dRSnFeyIhD+4g+/AZ0XJ1CPF19D+5
z0ojanJkh7lZn4JL+V6+mF1eOExiGrydIiiSXDA/p5FhavMMu8Om4S0sn5iaQ2aX
wRUv2SUKhbHDqhIILLeQKlB3X26obx1Vg0nRhy47qNQn/xc9oSWLAQSVOgsShQeC
6DSzrKIBdKB3V8uWOmuM7lWAoCP53bDRW+XIOu9wfpSaXN2VTyqzU7zpTq5BHX1a
+XRw8KNHZGnCSAOCofZWnKyJAhwEEAEKAAYFAlJBjYgACgkQ7Wfs1l3PaudFcQ//
UiM7EXsIHLwHxez32TzA/0uNMPWFHQN4Ezzg4PKB6Cc4amva5qbgbhoeCPuP+XPI
2ELfRviAHbmyZ/zIgqplDC4nmyisMoKlpK0Yo1w4qbix9EVVZr2ztL8F43qN3Xe/
NUSMTBgt/Jio7l5lYyhuVS3JQCfDlYGbq6NPk0xfYoYOMOZASoPhEquCxM5D4D0Z
3J3CBeAjyVzdF37HUw9rVQe2IRlxGn1YAyMb5EpR2Ij612GFad8c/5ikzDh5q6JD
tB9ApdvLkr0czTBucDljChSpFJ7ENPjAgZuH9N5Dmx2rRUj2mdBmi7HKqxAN9Kdm
+pg/6vZ3vM18rBlXmw1poQdc3srAL+6MHmIfHHrq49oksLyHwyeL8T6BO4d4nTZU
xObP7PLAeWrdrd1Sb3EWlZJ9HB/m2UL9w9Om1c6cb6X2DoCzQAStVypAE6SQCMBK
pxkWRj90L41BS62snja+BlZTELuuLTHULRkWqS3fFkUxlDSMUn96QksWlwZLcxCv
hKxJXOX+pHAiUuMIImaPQ0TBDBWWf5d8zOQlNPsyhSGFR5Skwzlg+m9ErQ+jy7Uz
UmNCNztlYgRKeckXuvr73seoKoNXHrn7vWQ6qB1IRURj2bfphsqlmYuITmcBhfFS
Dw0fdYXSDXrmG9wad98g49g4HwCJhPAl0j55f93gHLGIRgQQEQoABgUCUkGO5gAK
CRAV1ogEymzfsol4AKCI7rOnptuoXgwYx2Z9HkUKuugSRwCgkyW9pxa5EovDijEF
j1jG/cdxTOaJAhwEEAEKAAYFAlJBkdUACgkQkshDRW2mpm6aLxAAzpWNHMZVFt7e
wQnCJnf/FMLTjduGTEhVFnVCkEtI+YKarveE6pclqKJfSRFDxruZ6PHGG2CDfMig
J6mdDdmXCkN//TbIlRGowVgsxpIRg4jQVh4S3D0Nz50h+Zb7CHbjp6WAPVoWZz7b
Myp+pN7qx/miJJwEiw22Eet4Hjj1QymKwjWyY146V928BV/wDBS/xiwfg3xIVPZr
RqtiOGN/AGpMGeGQKKplkeITY7AXiAd+mL4H/eNf8b+o0Ce2Z9oSxSsGPF3DzMTL
kIX7sWD3rjy3Xe2BM20stIDrJS2a1fbnIwFvqszS3Z3sF5bLc6W0iyPJdtbQ0pt6
```

```
nekRl9nboAdUs0R+n/6QNYBkj4AcSh3jpZKe82NwnD/6WyzHWtC0SDRTVkcQWXPW
EaWLmv8VqfzdBiw6aLcxlmXQSAr0cUA6zo6/bMQZosKwiCfGl3tR4Pbwgvbyjoii
pF+ZXfz7rWWUqZ2C79hy3YTytwIlVMOnp3MyOV+9ubOsFhLuRDxAksIMaRTsO7ii
5J4z1d+jzWMW4g1B50CoQ8W+FyAfVp/8qGwzvGN7wxN8P1iR+DZjtpCt7J+Xb9Pt
L+lRKSO/aOgOfDksyt2fEKY4yEWdzq9A3VkRo1HCdUQY6SJ/qt7IyQHumxvL90F6
vbB3edrR/fVGeJsz4vE10hzy7kI1QT65Ag0EUkGMggEQAMTsvyKEdUsgEehymKz9
MRn9wiwfHEX5CLmpJAvnX9MITgcsTX8MKiPyrTBnyY/QzA0rh+yyhzkY/y55yxMP
INdpL5xgJCS1SHyJK85HOdN77uKDCkwHfphlWYGlBPuaXyxkiWYXJTVUggSjuO4b
jeKwDqFl/4Xc0XeZNgWVjqHtKF91wwgdXXgAzUL1/nwN3IglxiIR31y10GQdOQEG
4T3ufx6gv73+qbFc0RzgZUQiJykQ3tZK1+Gw6aDirgjQYOc90o2Je0RJHjdObyZQ
aQc4PTZ2DC7CElFEt2EHJCXLyP/taeLq+IdpKe6sLPckwakqtbqwunWVoPTbgkxo
Q1eCMzgrkRu23B2TJaY9zbZAFP3cpL65vQAVJVQISqJvDL8K5hvAWJ3vi92qfBcz
jqydAcbhjkzJUI9t44v63cIXTI0+QyqTQhqkvEJhHZkbb8MYoimebDVxFVtQ3I1p
EynOYPfn4IMvaItLFbkgZpR/zjHYau5snErR9NC4AOIfNFpxM+fFFJQ7W88JP3cG
JLl9dcRGERq28PDU/CTDH9rlk1kZ0xzpRDkJijKDnFIxT2ajijVOZx7l2jPL1njx
s4xa1jK0/39kh6XnrCgK49WQsJM5IflVR2JAi8BLi2q/e0NQG2pgn0QL695Sqbbp
NbrrJGRcRJD9sUkQTpMsLlQTABEBAAGJAiUEGAEKAA8FAlJBjIICGwwFCQgH7b8A
CgkQPLLq/MPWxmZAew//et/LToMVR3q6/qP/pf9ob/QwQ3MgejkC0DY3Md7JBRl/
6GWfySYnO0Vm5IoJofcv1hbhc/y3OeZTvK4s+BOQsNokYe34mCxZG4dypNaepkQi
x0mLujeU/n4Y0p0LTLjhGLVdKina2dM9HmllgYr4KumT58g6eGjxs2oZD6z5ty0L
viU5tx3lz3o0c3I9soH2RN2zNHVjXNW0EvWJwFLxFeLJbk/Y3UY1/kXCtcyMzLua
S5L5012eUOEvaZr5iYDKjy+wOxY4SUCNYf0GPmSej8CBbwHOF2XCwXytSzm6hNb3
5TRgCGbOSFTIy9MxfV5lpddQcdzijmuFSl8LySkL2yuJxjlI7uKNDN+NlfODIPMg
rdH0hBSyKci6Uz7Nz/Up3qdE+aISq68k+Hk1fiKJG1UcBRJidheds29FCzj3hoyZ
VDmf6OL60hL0YI1/4GjIkJyetlPzjMp8J7K3GweOUkfHcFihYZlbiMe7z+oIWEc7
0fNScrAGF/+JN3L6mjXKB6Pv+ER5ztzpfuhBJ/j7AV5BaNMmDXAVO4aTphWl7Dje
iecENuGTpkK8Ugv5cMJc4QJaWDkj/9sACc0EFgigPo68KjegvKg5R8jUPwb8E7T6
lIjBtlclVhaUrE2uLx/yTz2Apbm+GAmD8M0dQ7IYsOFlZNBW9zjgLLCtWDW+p1A=
=5gJ7
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.3. コアチーム書記 <core-secretary@FreeBSD.org >

```
pub   4096R/36A7C05FE1ECF9BB 2014-07-09 [expires: 2015-07-09]
      Key fingerprint = C07B F5E3 10AE 64BF 6120  B0F6 36A7 C05F ↺
E1EC F9BB
uid                          FreeBSD Core Team Secretary <core-
secretary@freebsd.org>
sub   4096R/7B5150C8D7CE5D02 2014-07-09 [expires: 2015-07-09]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFO9HvEBEADRfuWeoNUwib7ZjNmhg0Kt1kjiGEEosf3O2yMDfYuAXt4De6qK
S4KECe5+vZH2T8g+zmNLl/7JxdqHiWj9cnoZ6T3bqKh7w7pW7QzC/Q2k4mZsQkGl
xzhStHvaHSPKw5808TME0d3ewAfs0dQkDuA0eari0HipCbOVzqHUMTIROr/syPXs
jHxb2bj0KVzzq7wgy+vF4Cv25VzaAPBVgPv3HAoO/gLOr4SnXqBCw2vgprWx335t
QX1JslWlsUDmwwq40q4+eMnSFPZ0ing1DgfhMb+Dnrl6Rbxhb0pwPhbwubppUKfe
W6owOrTuUbATVoAhsfNySmUWQKc2p9w/8uFV/jJj9HOSgIMKrNONvqekPrjWOQn9
/lcQtGhldWmtPbMogOfaQisBEn1XjMZ3VEOagQxIe/6LDjU7GGoYvSdwf8Z0wXUY
/qDntPwudjJA4wQid1Tzf53gpUjr0tYq7aclpiBGs3F5EOs4HMXq5/xlwRGtBDHY
```

```
i9RNAlbRSfSD2s1nGsfsImPowlpjtLa+3PqYs/cRLGDu51DsgV/p/CqtAyebG+9O
WsF0Ydt4Q62jEuU8HY7SOj+AuKJVdUkyAZGk5vkPvsKzjdZUqRslurme7d3LqKai
FjBGj8UyId/IomDCjth3baGc/Y4e+JKyx1XDXgFY2HoQ2KzEoANrizjy5QARAQAB
tDhGcmVlQlNEIENvcmUgVGVhbSBTZWNyZXRhcnkgPGNvcmUtc2VjcmV0YXJ5QGZy
ZWVic2Qub3JnPokCPQQTAQoAJwUCU70sxwIbAwUJAeEzgAULCQgHAwUVCgkICwUW
AwIBAAIeAQIXgAAKCRA2p8Bf4ez5u2R4D/4o7Qb/hFz3XFRiBLtcsr+v5CS1fYgk
VN2Xp40dOTtt+Xo257+1oQDsS6McNXT7XSAFOTrFW+XID8GBnY6+ZpRYzontMCJd
25qGHzGBs2aFA/ROfhdvpOkhngxsYG89+IOUjS6SRNqG85Iro6eI7ZLi8sznEhZd
lH1jEWHiJhEubOvcedWS1zSHpOZwNIpyQV5d2O3/EhAOJRWgy8VmegeQLGg0zqdL
r9MKfg8/OwNQlysjrsA5MRnZMZAm6kEO381zYg8+Rwe+HtspzL/1cu4M7k5aBw0m
a8nShgXTqfgX+pwW2zBBL8c42p5D3Jh4Oe7KQ1KEvysZJGWO5ZjR9p4KLu49X2Yl
9cuD3Ii6+Nz9/cUlEGaMm4I8N3jpSNH+GCj7k1Wa9KWMh2ZxmVO5AML8h4g+A7aT
qNasKpG4XLXpcZ38sOw2/h6G2kwnrCel1ViCe0Z4pxeW95bxREqs6pkkym7Wz4sy
F5Mkba0uwFhWg5zBANBteXfOS/WU3fLEWeHXvZu9t44vyvP84qqZfNzBhz9puDVi
kPAY4uiZK4BTDkYZZr3YNKIIwEQ/rjcJSDKZKGJ6dsHKBSi0oGxFZpCE+YMH/Zqq
hTQHgXbMKWLgJ7xxRYJS/apvgnTd1Dc7zc5J7IT7XCdML0+Qj8eD48DwyMu+uAAq
/048iyhu/R8RP4kCHAQQAQoABgUCU70tAgAKCRADb2ye5/Oevxk0D/4wEJarg3nO
/B8ks+s7DadvpmNZrNSGeeR4qSEeZlH1ye30xD1HDE+ekJXC97Td4mKWKTrbRVEX
F1y1aBCe8tL5BbsRDmwzKwpv3OqQ1rhowlAKZD8hdBEyEn15ciaoRisFKpWAv+mm
mZeTnG5sYOo9BF1v36WrX+9AOvy7yduVcZE3kLNoivS+8bXe5ULj6uMoyo/eC5uk
q35f1p1qBzrplMc+vs7z8Npc7YFP2ORhR/mVg0rYFLkPVdPV3uFkrkb/Z2SVu02B
IFWEeJojEBEbFNfziHUaQ8Z7ZdtkZ9OIS1Z0Wuy64xjgORWVcTlRe4AwZKVC8DJs
Fs9Gk9gnsPmfLgM46pcKaac445w9uUQzKLxdCBKLPxMV/JHBRrMghyahevwnpsuB
Ok+ldsCNqjFaeAnVAmv3YtTbZ1JdorupeFa65JyjQon7l1JwWD67P7N0xrn1gxJ+
9uGUeHwGAvDF3IRr5ECISKAW92NfLYuNpyv6re/bt6A+FxaSA+E8P7Z+I4Gl5l37
czz+9ciYipiusweDuxectOlEnh13/WKXuUb68v6ZzbwGjkW8wunavhAN9fWxqqgF
W0tKtmpAOHFzmNOw5WGPfGdKOzc+g5ZqRt3CREVHj/uizNJsHzfadkTqYum+kQTa
y4tSgGH0S3EAiHGsEy4YZ4M/nxdIyakmsIkBHAQQAQIABgUCU701xQAKCRBNWP3N
LKSXdmOGCADEySzz4Q6wKsx/gLiAyhYNbEJbiv1MirxhjIYGP9MqNpxxI1+Q3kuj
01K6ELIMuAhehoQOgU4AssJQxu7q78+hz2O7s+V0Syl+pvEOl2zUCgAmOYfle+BQ
75ZEEiINBuh6SOXBVLhfNp90FZ55KUSW4EeyoT+A4nRGHRgCTEfZ5WHi3lGlaLQd
Z9viLfNKA/DxrLWww+joTPIEhc3eU1mgDrcmfxo/L95EmTyUa5BtE0WuLwQEaY8H
J3eBgA9Y130ubuzzY4jGl4SCNedMzeIroHw2Bogd3V+E5aFtGd8gZUjXXr8rM6yX
PpttP2Hc8Bie2YXI2NffqwVqpL0dxo3uiQIcBBABAgAGBQJTvTYRAAoJEMATMJ1t
fkRccM4P/Rbg0W6l4KPFUvyKcUE6odRwoXExRGHdG9qW8Vf6xtW5eXUX/AZoCnXD
f3yWWttxgzN1e8iNRh0aYfuNSFTuHcHut/xw6GZ1yqASbuDmGWQ6uTb0yHYQcwQ5
ioaRaZZo5cpnSs0qZUpnrSzdUzyVmlKsD+1ut0/Z8yM8WGRyhplWX0dfXKnUUxJG
yh4GQc2dQon1vrsiuDTD2hr3EVues7le4WU+csegZTGPgPjhTSH6ZNFdDs4Y5KPi
unjXx+X6avPKPSJCnC9YlPMkI0RcokVLJW+K3+4QnbqU8m2MpZWVaaOo5s9PCx1I
208EHQ77A7EAFYNFrPZmtSV5X3BhU2rYNoRu0fpsNqJC34Oi7JdZdplPoO7FHRAC
AyQJyv8KUG8VVzK7m6Kt/Okq7LBc8RuvLQpUHSv1Z19fQvFgTegM5Pcpp3/ful/H
QIIc7XRElM57e+t8kbsoRpOlaKa3Okl3KisXdksWB4Fu6XdVArY/jIIQGs6dCpYa
jhRZcjkjHUAPvY/OqD2mBSwj0YwP0RoMVvFHMP1cgB3gjaB37A+DJeiKEXTWzGe1
fKC1TxCcOUZsrcqXnUyy23lKV9CXC7za3eB23dPIfWzJnD9BsVgYsemRVJx8r3Sv
QIL5zjKVDRAuV3M/HbCtSOgO68MExC0TFEl/8LPIMW8oVCY4+iqliQIcBBABCgAG
BQJTvUdSAAoJEO1n7NZdz2rnXxsP/R8WHku1nxjELqdM9M72JLD8UBlaAIwLStDy
hnTvLa0GO6eN0r2eJ1+tG8mKB+PZKOvNt8eZcSO/kjUvTIBILt7fAtN1BhsWpjQz
n+tuVWs4GVoPQssM4N09AYUzx8ni4byADY6n9l4zof2HsPsjXvuw/bzYXctKTQxg
gD3nswtLGY3q6unYewIChyaG8DStihFLcXIhXbwc6EOqdPN3VWwNInG/602UTO2L
eXoEM+tTaXkE51P5otACVH37AW0Vqqh1GxklYlMYLrKKn/YIBRvLVS5G+95iKs3g
MJhnaeFND2s9dmOTXyKyfTUffr/XTL/PVJSCbdqwiuXZQp8J77MtYyJn1262H8ko
59OlPtqvpBNuywco0/F8BOFvSTw5sS8CmU0EHvyunKaofS3mxCFd2B0DzX89+AoY
ZY7CKU8OYt/VqhsfsL0C+DL4+XschBOUoTg6HrqG4F69+gerkK4Ps2984vOeTXe3
```

```
IqlYN/Bn92m9rGy9PKkpqG5C6w5X58BgvfeWTAkM4X32rZHKOmyYSCdRJQw7MGSR
24OaWKPmgKvMaH8MQaJx9oSaAgF1y1892+ykVI9ntCVwywkmxNg1lfMuVFM4Vh9j
+C7OngnbQbhYtbFG90z9zfoMln39z8KT2yDP4A7Hklw0xmmT6t8KduGD2tfmaW4+
oz9attTyiQIcBBABCgAGBQJTvUdkAAoJEJLIQ0VtpqZuLNQP/RazXTtk1mixmLfu
nrScgh/1Gtb6XPVDTP5SGb/8HVdY1a+dDBClUIAFbEAUBIcBO4/NVlW9un4IHyri
xdD/ijE/Cr/BMLnSFU5EmHKn8yOc3Bv6eTTRbJ8EYru0Cj5MdSIAoq8JKF8Lbxba
hBFw9ZSIpREPGlxnKI3TEuHJQreSZR07/GPk50suTK2CXxzEsk4VZesSTnwt/1m+
hrdtyNke3+wy8R0Use+KKgmTzpQ9phc8wq3uhHFzLJ5HbE21VRHgFd5+osZuSDuH
jw5o/zU5o0Bq9DDY3TWXPD9lgqKQ2PgH5DG6od9gjpjJdvXpXpck08HJDu2V+u7M
zW5lsnthDsW39YVLRD4ZwZJQaw+0wBuGDDxV+8x4fYhYJnXD8ZtUHCQ56FlcMyzL
m00WRLNxBSJjRvH12geg5xK3JWZ8V6Ce+XhvIAduDkajuUKQoWvTMzYaviqQhfV1
zXNNt5rxDNT+jMiwnAtDeCTZPMfGvzOPw9sYBz5RUa4liVPrGiEW/snAhcMB9JDK
jMAsKHvJwIvB9QrRC9sz6JIgEqv+jlsdkslEVU2AALlcJ3QuXlW0j9Z26q0sNGt8
5FVqhc4DOyXtZhKR5Ru8Lyc3swYRi2ofu9Roycq3L0swcMoGnOikRbj2PuuRdQiU
ozJ7mT6JmNF8ynlx/1+uzniVneEluQINBFO9HvEBEACynbl7EgcRIGWP7O6h1O6m
rYXNZ2JpJBgYosqizdDHyru2nQSrNfgiwAM1feB2NLJC0coQzRO1sDK2JP770+eK
3ZhbWSP5BWN2toSFVEGlVpGWLBGoefaeZnZA22IDzpOIjIi7iC92JBsTXESsBoV8
iG1rylQ15pcE03IQEuuDu9r7H8RJ3vTfX1c+a+B8MUHn56kn3QkdG2blV0/3gjFq
qavZeOxZpAmyn9n9Vc3yCCPkagtNQwleNyZOSOLjjVpBjncE6dATdLOj85phfOU6
eO/0bMXAgTr7mY41EIqYqdPQYrY93ySGgBvBkyNaH5AlDNZZwJ4ddtDMFoP8nUhB
oRrf5ApYyHcEmSXahLfW3a2qrPm/w5VLEGLt53/6GZvEetpP+TtBLAxX6XaC2SXA
OrzfSZENdYt/Ew6F/dTCZ622m0eW65iVwSi1sNZD2hNFPs/12a2tem7DAWqD2bi8
BltKRbO+8T7BARwIl5hXGq5+YnO+DgTIf4SYkSt8aiPYwDAF3YSkzpiUmZoBSRt6
Sb9sZ3zIxpfnrtLFmSeujzinyCVNzFdn+HKxZvI9Mc3Tv/LqPruVuWHt1Aj+eygH
5bRZw4PTsMNX1FxM/K8hRY91A6Fyp3GCkb5RzqdEGuSONBseaZirC0d+EYZ4smy1
jydpzwT1O8VjY4wi5BdgwQARAQABiQIlBBgBCgAPBQJTvR7xAhsMBQkB4TOAAAoJ
EDanwF/h7Pm7gzUP/Auq4I876RUqAP2DgPNArjuc7jqvIIsgqBvwS2Vdvxg0pFQm
sL13nvlmFHR1dp+yOhiz1WmXIVGa9UoBrVdmDrzqCifHUVlFct99DLJlM+hNmQnH
gTABTsClR4idkekOvyEgAW/gsgddYfqmN8nRKSrokFZvwoR7HlCcSXEuATbqgHjJ
C8IuJuIgDWjshy3yVLIo5v/g/Yuio4bxtRTJm72DtUIdDH3tDejc37wd945Rzk4h
Kzet64inF6XfRhqBsRob1m6/wtP62xPcGbb1GlzxYg9WllRJ/ZOTktB4BemVjRn4
/0KU93YyxFxHOtXsoZpRGDlovQBUhRkdb2JEyHaLnFVKa7w7JFKW9d+OdGjV93l/
/6139QEQlcuocV4QciXZOXTZkWsaKQOb6WkrSwkQwvNFS2yeUd0x6HD0xebztmSG
oXL3uCKyeIxP/8MJjJUkWHh/+DLue4khlHwva7Wym0cvYoS3RzfYd26t9OgM7rfl
E0tOeJovaoaKP6Aant/EkQj2Wb1MssO/Ia44NyMTrLARAPlAEGW/HI5JL9DZDnmU
tByf29yqaHDuJCZS3nvR3T/4q+pblTVdxQgwopUWjiKkg72yCXMhr/6FcOr9Xgqn
++GuMZzh/TFdEoFoJn9stUaScYsoMoihym2NFbd1l2Pw/ifTQk5/KpQD/Hkw
=zQ/h
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.4. ports 管理チーム書記 <portmgr-secretary@FreeBSD.org >

```
pub   2048R/D8294EC3BBC4D7D5 2012-07-24
      Key fingerprint = FB37 45C8 6F15 E8ED AC81  32FC D829 4EC3 ↺
BBC4 D7D5
uid                          FreeBSD Ports Management Team ↺
Secretary <portmgr-secretary@FreeBSD.org>
sub   2048R/5CC117965F65CFE7 2012-07-24
sub   4096R/CA20328577064EB7 2013-10-05
sub   4096R/8B114B3613867E00 2013-10-05
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBFAOzqYBCACYd+KGv0/DduIRpSEKWZG2yfDILStzWfdaQMD+8zdWihB0x7dd
JDBUpV0o0Ixzt9mvu5CHybx+9lOHeFRhZshFXc+bIJOPyi+JrSs100o7Lo6jg6+c
Si2vME0ixG4x9YjCi8DisXIGJ1kZiDXhmVWwCvL+vLInpeXrtJnK8yFkmszCOr4Y
Q3GXuvdU0BF2tL/Wo/eCbSf+3U9syopVS2L2wKcP76bbYU0ioO35Y503rJEK6R5G
TchwYvYjSXuhv4ec7N1/j3thrMC9GNpoqjVninTynOk2kn+YZuMpO3c6b/pfoNcq
MxoizGlTu8VT4OO/SF1y52OkKjpAsENbFaNTABEBAAG0R0ZyZWVCU0QgUG9ydHMg
TWFuYWdlbWVudCBUZWFtIFNlY3JldGFyeSA8cG9ydG1nci1zZWNyZXRhcnlARnJl
ZUJTRC5vcmc+iQE4BBMBAgAiBQJQDs6mAhsDBgsJCAcDAgYVCAIJCgsEFgIDAQIe
AQIXgAAKCRDYKU7Du8TX1QW2B/0coHe8utbTfGKpeM4BY9IyC+PFgkE58Hq50o8d
shoB9gfommcUaK9PNwJPxTEJNlwiKPZy+VoKs/+dO8gahovchbRdSyP1ejn3CFy+
H8pol0hDDU4n7Ldc50q54GLuZijdcJZqlgOloZqWOYtXFklKPZjdUvYN8KHAntgf
u361rwM4DZ40HngYY9fdGc4SbXurGA5m+vLAURLzPv+QRQqHfaI1DZF6gzMgY49x
qS1JBF4kPoicpgvs3o6CuX8MD9ewGFSAMM3EdzV6ZdC8pnpXC8+8Q+p6FjNqmtjk
GpW39Zq/p8SJVg1RortCH6qWLe7dW7TaFYov7gF1V/DYwDN5iQEcBBABAgAGBQJQ
DuVrAAoJENk3EJekc8mQ3KwIAImNDMXAF8ajPwCZFpM6KDi3F/jpwyBPISGY1oWu
YPEi1zN94k5jS90aZb3W8Y8x4JTh35Ewb6XODi3uGLSLCmnlqu2a80yPfXf5IuWm
IQdFNQxvosj9UHrg+icZGFmm+f0hPJxMTsZREv3AvivQfnb/N3xIICxW4SjKSYXQ
cq4hr4ObhUx7GKnjayq+ofU2cRlujr87uOH0fO3xhOJG4+cX5mI1HGK38k0Csc1z
qYa/66Qe5dnIZz+sNXpEPMLAHIt1a45UB967igJdZSDFN33bPl1QWmf3aUXU3d1V
ttiSyHkpm4kb9KgsDkUk1IJ5nUe9OXydWtoqNW5afDa5N0aIRgQQEQIABgUCUA7l
wwAKCRB59uBxdBRinNh2AJ41+zfsaQSRHWvSkqOXGcP/fgOduwCfUJDT+M1eXe2u
dmKof/9yzGYMirKJASIEEAECAAwFAlAaIT8FAwASdQAACgkQlxC4m8pXrXwCHAf+
J7l+L7AvRpqlQcezjnjFS/zG1098qkDflThHZlpVnrBMJZaXdvL6LzVgiIYVWZC5
CSSazW9EWFjp9VjM7FBHdWFZNMV7GAuUt0jzx6gGXOWwi+/v/hs1P11RyDZN5hIC
HdPNmyZVupciDxe+sIEP9aEbVxcaiccqzM/pFzIVIMMP5tCiA42q6Mz3h0hy6hnt
UKptS8Uon6sje5cDVcVlKAUj1wO2cphCqkYlwMQfZV5J9f/hcW5ODriD3cBwK8So
cA2Cq5JYF8kYDL1+pXnUutGnvAHUYt87RWvQdKmfXjzBcMFJ2LlPUB1+IFvwQ13V
9R8j9B/EdLmSWQYT9qRA2rkBDQRQDs6mAQgAzNxJYpf5PrqV8pdRXkn36Fe45q67
1YtbZ2WrT7D0CVZ8Z+AZsxnP/tiY1SrM2MepCeA2xBAhKGsWBWo1aRk5mfZOksKs
iXsi2XeBVhdZlCkrOMKBTVian7I1lH59ZnNIMX0Nl0tlj3L1IjeWWNvfej43URV8
1S9EmSwpjaWboatr2A+1oJku5m7nPD9JIOckE1TzBsyhx7zIUN9w6MKr7gFw8DCz
ypwUKyYgKYToVm8QlkT/L3B0fuQHWhT6ROGk4o8SC71ia5tc1TzUzGEZ1AQO8bbn
bmJLBDKveWHCoaeAkRzINzoD9wAn9z4pnilze59QtKC1cOqUksTvBSDh6wARAQAB
iQEfBBgBAgAJBQJQDs6mAhsMAAoJENgpTsO7xNfVOHoH/i5VyggVdwpqPX8YBmN5
mXQziYZNQoiON8IhOsxpX4W2nXCj5m6MACV6nJDVV6wyUH8/VvDQC9nHarCe1oaN
sHXJz0HamYt5gHJ0G1bYuBcuJp/FEjLa48XFI7nXQjJHn8rlwZMjK/PWj1lw2WZi
ekviuzTEDH8c3YStGJSa+gYe8Eyq3XJVAe2VQOhImoWgGDR3tWfgrya/IdEFb/jm
jHSG5XUfbI0vNwqlf832BqSQKPG/Zix4MmBJgvAz4R71PH8WBmbmNFjDelxVyfz8
0+iMgEb9aL91MfeBNC2KB1pFmg91mQTsiq7ajwVLVJK8NplHAkdLmkBCO8MgMjzG
hlG5Ag0EUk+ViAEQALkCj95JmvmfgytOxnR1w8xnQBuUxtYxf+BWwqU0IfOMBxAm
XDRfbfe9Bc3DDZygmjrZ0RCDcVJ4OmDNRc/vvoTst870so49dM1h2i6aWUhhD4Hy
cNjkx0wAIyZ9jXidT3LyAhCTSf/GjtpcORCvmAsXcJHzhhbJ5eM2SBkXmNdn4MeC
/hCbfUfzN64fmsGR4tKKIlPPayyQaQv/dOP3ofEh4SWRc8KtjEA6uIyHztWMWEJQ
KN/7nSeVuwDpVJL9l4yUGB0TqlK0p/necoQkOolnHUX5oK7Emin9TjPYCGqYAEWu
M1BbWXiYIdivlEm3ZU+vqq6CbSw/SKDZ2ZuK4UzDGQnW7WycsbXqZ18aewh1mdf+
5+YBxeXazRJvfFapB/WvtVqi84LOWior1IBMMg3PCyBiCjhHwXvoV5V6M3lthrnV
gyiCTreLXvTLNgIKQQFUvwW2Jscuwa3HhBgkhkJBSCPmpWAIZ9eA1RugvFQ893Xh
t8vy+z33WqxgUzhD8ZonKCBEZa29Zj7SeIYXcZmVmxCbRt4PYkgFoq6VC+93qtXu
OaOvC+0b9qkmNj5EqvozRt6SNv0sDe844T0LacR6QS9Pq/YMjAs8ao3gw0iwOAlV
vpiZfoEb6purf0Tz0h/URRHUZngEBsFmhciZxuCg9g/GaD8o/+PX0nzOWH4PABEB
```

```
AAGJAz4EGAECAAkFAlJPlYgCGwICKQkQ2ClOw7vE19XBXSAEGQECAAYFAlJPlYgA
CgkQyiAyhXcGTrfF9g/8C6JhufyxnYO0DmPURCFV0em3WjXcxQkhQa03u/sU8OFN
yp2UOB8jxI3XAsGe4RnclPF273K5G5OfUAJ2hjWFGrMqXGNLlsuJvDJCUmdHnI6c
sw1RvOn2myMzUSTqxPQynkvntughqiyXbGQaoxQJVWjruhogcIkVYvOT4nMGXCyk
9Q+uooLNw8xpzsZBw83znWhjjzemu/BbOfjXbf3qdK9NWMkBfcaR9sDhRA+YY4bc
o7oTKXFfh3gMN1pqTuAM8dbGc0/bS558yNQuFaJ+cX3lbSLXz5oibJi6kudyL9E/
CkVvMFWTLcQQWCPS5rK0dctodW1x8JnwVnaG80trFIxmQ+wk4S156CdYwTv/1LbW
GKcVz9Og1e9wIL2LsVaLD9OIRnx4G8dLqZJz4pmGEES7CX6rrJuSpXxNOg4PthiP
s0z/q9deohKJkUgj6tln0OCgI8203GSSLq3FTAFer2VS+m1XMkA2mqk1Wv6tZeyY
MnYAZCBzc16F2zMg5dVKmV114bHmRTX0b5QQNB7JU3C34kdTagjLr4dz/5BhXgof
zjP9HgVQKCp7JvTdUT7N0y/k+mRMmnK8vnWdbOZH48IOELDhMkU5QIiKHoJaXsw7
wVo68LtSQItfIoA/m6EORGZxUWCi/8G+kBK6NEExR1dlAW+M+fRhzNgqvqoLrcfR
4wgAkQGq3M2/hGdU/Z1j6CDKEvA1/iSRpnBnfVM8KUR661QrEctmIo1YwSU7x5+g
H5lfDUjU2cIlD+HgfxERKwl+hb2KY9OyLq2AUhhf8rAvG2dU9djDp7TWIvF3wexL
liu4C6EWcwlnEecCpkPUYv7/PKb0h0xAx32Umb6dzkfd24miWZTI/Gg7R1Qyl1DC
wBM9kgPRgEhnV3ummsD3KPXf5UwzHPslhqWTO1p2iML4exnXlknL7mJKjj4d3gyi
5LxVVSzt2xi5MyCkFhiHFkSfItihcRpHdzxsBW+7YWw1ELlcKMeOkZQcyBm3VpE/
ZD4DQK6zwGfk4y8WS7A5SJEjJrkCDQRST5W/ARAAwbT4loEoK7ZY8fzt8hyhdBgS
bpXFWAB6yeyGDYp9ucG4ySjlfZIAew/EOzIR/68pKiGzIE0+kxKjlvVDcJoqWfrr
gomondGK2oyK35qOsYOlae6tsZm73InJuGSC+fZ1Vv0HBIn8JZpJXFV7z4FotaG5
iaKDdZCjxawVEWDl7z0zDSLKgPLtK4uSsEVsQIhPo5YLSqT7mCnleCqdNu3S4cun
tXaZmmSZnfv3qnkiPNNNSagZrMUK349fVCdvnJ4hKDbMAEgbxZVwEeB4HFicjwAN
UQTO1qSc+h8bwkrN/RgaT0Zz3nak+DJOOrKRV5VWWemx4Oy2oT08SqMRbhip/veI
LvrV+rKH+lb9uxp0vYWnRvWodOlEla0NeBzfz6Sv8ZI2D+xLJ260mIB4f4BUYCBi
SDu8UKTDffYba+50I9RgV9+umJ2WTcp/PA5/59M9e6R2F3dcM0Qey8hVDcf+rPY2
thXymS85IHcPauDjvjpHpx1xvhoEu2iX0yJwBDUX/xVWwH/lei7DT0cd1dB1pKiP
zNvWSoPp6zhqcB6tUvE69CLRzotmb2OiAkFOReEVkcATL+bG9PBN0zWtSXJuCSL5
WAb+syRzBGwLA7+iQOY0yXCZ+Etl6BG8rsyQkBRwpTG5jXzDKUTl2hmKGFmU72xc
cPeNxhbq1tGMGfDVVmkAEQEAAYkBHwQYAQIACQUCUk+VvwIbDAAKCRDYKU7Du8TX
1cEOB/0WwX3yF/MC8upI0YAbjht+KG4cLgU6qo1ydZgQyHpAf8cttzq/uCDu4wLE
g0CDmreTXoiNR9W0ULhkn1LFkZQI6Z2uSH/wqQUJrE4P9Oo61TXClIzvjBoEufkS
3bg3wuAKiQ4cD1XxyThUG3qa4nbGVvKi4eWYuubpzJof7QD75LudHmLneY4mtCNA
ZgmgEWA2Utn0GWN1QNfoy1NGeLt0kza0VFjm6l6KeMc28ULZZztg0KbuFmQIreN9
JHQbJrrqX8ev57SqEtanoPyX4IjxVJFEmTn4xVSlRXY2uFZZtlB1SuyEPYmrdnjb
pDN9ZUgVOk/c+0+5u+G3JBRYu6P1
=fyjO
-----END PGP PUBLIC KEY BLOCK-----
```

FreeBSD Glossary

This glossary contains terms and acronyms used within the FreeBSD community and documentation.

A

ACL	Access Control List参照
ACPI	Advanced Configuration and Power Interface参照
AMD	Automatic Mount Daemon参照
AML	ACPI Machine Language参照
API	Application Programming Interface参照
APIC	Advanced Programmable Interrupt Controller参照
APM	Advanced Power Management参照
APOP	Authenticated Post Office Protocol参照
ASL	ACPI Source Language参照
ATA	Advanced Technology Attachment参照
ATM	Asynchronous Transfer Mode参照
ACPI Machine Language	Pseudocode, interpreted by a virtual machine within an ACPI-compliant operating system, providing a layer between the underlying hardware and the documented interface presented to the OS.
ACPI Source Language	The programming language AML is written in.
Access Control List	A list of permissions attached to an object, usually either a file or a network device.
Advanced Configuration and Power Interface	A specification which provides an abstraction of the interface the hardware presents to the operating system, so that

	the operating system should need to know nothing about the underlying hardware to make the most of it. ACPI evolves and supersedes the functionality provided previously by APM, PNPBIOS and other technologies, and provides facilities for controlling power consumption, machine suspension, device enabling and disabling, etc.
Application Programming Interface	A set of procedures, protocols and tools that specify the canonical interaction of one or more program parts; how, when and why they do work together, and what data they share or operate on.
Advanced Power Management	An API enabling the operating system to work in conjunction with the BIOS in order to achieve power management. APM has been superseded by the much more generic and powerful ACPI specification for most applications.
Advanced Programmable Interrupt Controller	
Advanced Technology Attachment	
Asynchronous Transfer Mode	
Authenticated Post Office Protocol	
Automatic Mount Daemon	A daemon that automatically mounts a filesystem when a file or directory within that filesystem is accessed.

B

BAR	Base Address Register参照
BIND	Berkeley Internet Name Domain参照
BIOS	Basic Input/Output System参照
BSD	Berkeley Software Distribution参照

Base Address Register	The registers that determine which address range a PCI device will respond to.
Basic Input/Output System	The definition of BIOS depends a bit on the context. Some people refer to it as the ROM chip with a basic set of routines to provide an interface between software and hardware. Others refer to it as the set of routines contained in the chip that help in bootstrapping the system. Some might also refer to it as the screen used to configure the boostrapping process. The BIOS is PC-specific but other systems have something similar.
Berkeley Internet Name Domain	An implementation of the DNS protocols.
Berkeley Software Distribution	This is the name that the Computer Systems Research Group (CSRG) at The University of California at Berkeley gave to their improvements and modifications to AT&T's 32V UNIX®. FreeBSD is a descendant of the CSRG work.
Bikeshed Building	A phenomenon whereby many people will give an opinion on an uncomplicated topic, whilst a complex topic receives little or no discussion. See the FAQ for the origin of the term.

C

CD	Carrier Detect参照
CHAP	Challenge Handshake Authentication Protocol参照
CLIP	Classical IP over ATM参照
COFF	Common Object File Format参照
CPU	Central Processing Unit参照
CTS	Clear To Send参照
CVS	Concurrent Versions System参照

Carrier Detect
: An RS232C signal indicating that a carrier has been detected.

Central Processing Unit
: Also known as the processor. This is the brain of the computer where all calculations take place. There are a number of different architectures with different instruction sets. Among the more well-known are the Intel-x86 and derivatives, Sun SPARC, PowerPC, and Alpha.

Challenge Handshake Authentication Protocol
: A method of authenticating a user, based on a secret shared between client and server.

Classical IP over ATM

Clear To Send
: An RS232C signal giving the remote system permission to send data.
Request To Send参照

Common Object File Format

Concurrent Versions System
: A version control system, providing a method of working with and keeping track of many different revisions of files. CVS provides the ability to extract, merge and revert individual changes or sets of changes, and offers the ability to keep track of which changes were made, by who and for what reason.

D

DAC
: Discretionary Access Control参照

DDB
: Debugger参照

DES
: Data Encryption Standard参照

DHCP
: Dynamic Host Configuration Protocol参照

DNS
: Domain Name System参照

DSDT
: Differentiated System Description Table参照

DSR
: Data Set Ready参照

DTR
: Data Terminal Ready参照

DVMRP — Distance-Vector Multicast Routing Protocol参照

Discretionary Access Control

Data Encryption Standard — A method of encrypting information, traditionally used as the method of encryption for UNIX® passwords and the crypt(3) function.

Data Set Ready — An RS232C signal sent from the modem to the computer or terminal indicating a readiness to send and receive data.
Data Terminal Ready参照

Data Terminal Ready — An RS232C signal sent from the computer or terminal to the modem indicating a readiness to send and receive data.

Debugger — An interactive in-kernel facility for examining the status of a system, often used after a system has crashed to establish the events surrounding the failure.

Differentiated System Description Table — An ACPI table, supplying basic configuration information about the base system.

Distance-Vector Multicast Routing Protocol

Domain Name System — The system that converts humanly readable hostnames (i.e., mail.example.net) to Internet addresses and vice versa.

Dynamic Host Configuration Protocol — A protocol that dynamically assigns IP addresses to a computer (host) when it requests one from the server. The address assignment is called a「lease」.

E

ECOFF — Extended COFF参照

ELF — Executable and Linking Format参照

ESP — Encapsulated Security Payload参照

Encapsulated Security Payload

Executable and Linking Format

Extended COFF

F

FADT — Fixed ACPI Description Table参照

FAT — File Allocation Table参照

FAT16 — File Allocation Table (16-bit)参照

FTP — File Transfer Protocol参照

File Allocation Table

File Allocation Table (16-bit)

File Transfer Protocol — A member of the family of high-level protocols implemented on top of TCP which can be used to transfer files over a TCP/IP network.

Fixed ACPI Description Table

G

GUI — Graphical User Interface参照

Giant — The name of a mutual exclusion mechanism (a `sleep mutex`) that protects a large set of kernel resources. Although a simple locking mechanism was adequate in the days where a machine might have only a few dozen processes, one networking card, and certainly only one processor, in current times it is an unacceptable performance bottleneck. FreeBSD developers are actively working to replace it with locks that protect individual resources, which will allow a much

greater degree of parallelism for both single-processor and multi-processor machines.

Graphical User Interface — A system where the user and computer interact with graphics.

H

HTML — HyperText Markup Language参照

HUP — HangUp参照

HangUp

HyperText Markup Language — The markup language used to create web pages.

I

I/O — Input/Output参照

IASL — Intel's ASL compiler参照

IMAP — Internet Message Access Protocol参照

IP — Internet Protocol参照

IPFW — IP Firewall参照

IPP — Internet Printing Protocol参照

IPv4 — IP Version 4参照

IPv6 — IP Version 6参照

ISP — Internet Service Provider参照

IP Firewall

IP Version 4 — The IP protocol version 4, which uses 32 bits for addressing. This version is still the most widely used, but it is slowly being replaced with IPv6.
IP Version 6参照

IP Version 6 — The new IP protocol. Invented because the address space in IPv4 is running out. Uses 128 bits for addressing.

Input/Output

Intel's ASL compiler — Intel's compiler for converting ASL into AML.

Internet Message Access Protocol — A protocol for accessing email messages on a mail server, characterised by the messages usually being kept on the server as opposed to being downloaded to the mail reader client.
Post Office Protocol Version 3参照

Internet Printing Protocol

Internet Protocol — The packet transmitting protocol that is the basic protocol on the Internet. Originally developed at the U.S. Department of Defense and an extremely important part of the TCP/IP stack. Without the Internet Protocol, the Internet would not have become what it is today. For more information, see RFC 791.

Internet Service Provider — A company that provides access to the Internet.

K

KAME — Japanese for「turtle」, the term KAME is used in computing circles to refer to the KAME Project, who work on an implementation of IPv6.

KDC — Key Distribution Center参照

KLD — Kernel ld(1)参照

KSE — Kernel Scheduler Entities参照

KVA — Kernel Virtual Address参照

Kbps — Kilo Bits Per Second参照

Kernel ld(1) — A method of dynamically loading functionality into a FreeBSD kernel without rebooting the system.

Kernel Scheduler Entities — A kernel-supported threading system. See the project home page for further details.

Kernel Virtual Address

Key Distribution Center

Kilo Bits Per Second
: Used to measure bandwidth (how much data can pass a given point at a specified amount of time). Alternates to the Kilo prefix include Mega, Giga, Tera, and so forth.

L

LAN
: Local Area Network参照

LOR
: Lock Order Reversal参照

LPD
: Line Printer Daemon参照

Line Printer Daemon

Local Area Network
: A network used on a local area, e.g. office, home, or so forth.

Lock Order Reversal
: The FreeBSD kernel uses a number of resource locks to arbitrate contention for those resources. A run-time lock diagnostic system found in FreeBSD-CURRENT kernels (but removed for releases), called witness(4), detects the potential for deadlocks due to locking errors. (witness(4) is actually slightly conservative, so it is possible to get false positives.) A true positive report indicates that「if you were unlucky, a deadlock would have happened here」.

  True positive LORs tend to get fixed quickly, so check http://lists.FreeBSD.org/mailman/listinfo/freebsd-current and the LORs Seen page before posting to the mailing lists.

M

MAC
: Mandatory Access Control参照

MADT
: Multiple APIC Description Table参照

MFC
: Merge From Current参照

MFP4
: Merge From Perforce参照

MFS — Merge From Stable参照

MIT — Massachusetts Institute of Technology参照

MLS — Multi-Level Security参照

MOTD — Message Of The Day参照

MTA — Mail Transfer Agent参照

MUA — Mail User Agent参照

Mail Transfer Agent — An application used to transfer email. An MTA has traditionally been part of the BSD base system. Today Sendmail is included in the base system, but there are many other MTAs, such as postfix, qmail and Exim.

Mail User Agent — An application used by users to display and write email.

Mandatory Access Control

Massachusetts Institute of Technology

Merge From Current — To merge functionality or a patch from the -CURRENT branch to another, most often -STABLE.

Merge From Perforce — To merge functionality or a patch from the Perforce repository to the -CURRENT branch.
Perforce参照

Merge From Stable — In the normal course of FreeBSD development, a change will be committed to the -CURRENT branch for testing before being merged to -STABLE. On rare occasions, a change will go into -STABLE first and then be merged to -CURRENT.

This term is also used when a patch is merged from -STABLE to a security branch.
Merge From Current参照

Message Of The Day — A message, usually shown on login, often used to distribute information to users of the system.

Multi-Level Security

Multiple APIC Description Table

N

NAT — Network Address Translation参照

NDISulator — Project Evil参照

NFS — Network File System参照

NTFS — New Technology File System参照

NTP — Network Time Protocol参照

Network Address Translation — A technique where IP packets are rewritten on the way through a gateway, enabling many machines behind the gateway to effectively share a single IP address.

Network File System

New Technology File System — A filesystem developed by Microsoft and available in its「New Technology」operating systems, such as Windows® 2000, Windows NT® and Windows® XP.

Network Time Protocol — A means of synchronizing clocks over a network.

O

OBE — Overtaken By Events参照

ODMR — On-Demand Mail Relay参照

OS — Operating System参照

On-Demand Mail Relay

Operating System — A set of programs, libraries and tools that provide access to the hardware resources of a computer. Operating systems range today from simplistic designs that support only one program running at a time, accessing only one device to fully multi-user, multi-

	tasking and multi-process systems that can serve thousands of users simultaneously, each of them running dozens of different applications.
Overtaken By Events	Indicates a suggested change (such as a Problem Report or a feature request) which is no longer relevant or applicable due to such things as later changes to FreeBSD, changes in networking standards, the affected hardware having since become obsolete, and so forth.

P

p4	Perforce参照
PAE	Physical Address Extensions参照
PAM	Pluggable Authentication Modules参照
PAP	Password Authentication Protocol参照
PC	Personal Computer参照
PCNSFD	Personal Computer Network File System Daemon参照
PDF	Portable Document Format参照
PID	Process ID参照
POLA	Principle Of Least Astonishment参照
POP	Post Office Protocol参照
POP3	Post Office Protocol Version 3参照
PPD	PostScript Printer Description参照
PPP	Point-to-Point Protocol参照
PPPoA	PPP over ATM参照
PPPoE	PPP over Ethernet参照
PPP over ATM	
PPP over Ethernet	

PR	Problem Report参照
PXE	Preboot eXecution Environment参照
Password Authentication Protocol	
Perforce	A source code control product made by Perforce Software which is more advanced than CVS. Although not open source, its use is free of charge to open-source projects such as FreeBSD. Some FreeBSD developers use a Perforce repository as a staging area for code that is considered too experimental for the -CURRENT branch.
Personal Computer	
Personal Computer Network File System Daemon	
Physical Address Extensions	A method of enabling access to up to 64 GB of RAM on systems which only physically have a 32-bit wide address space (and would therefore be limited to 4 GB without PAE).
Pluggable Authentication Modules	
Point-to-Point Protocol	
Pointy Hat	A mythical piece of headgear, much like a `dunce cap`, awarded to any FreeBSD committer who breaks the build, makes revision numbers go backwards, or creates any other kind of havoc in the source base. Any committer worth his or her salt will soon accumulate a large collection. The usage is (almost always?) humorous.
Portable Document Format	
Post Office Protocol	Post Office Protocol Version 3参照
Post Office Protocol Version 3	A protocol for accessing email messages on a mail server, characterised by the messages usually being downloaded from the server to

	the client, as opposed to remaining on the server. Internet Message Access Protocol参照
PostScript Printer Description	
Preboot eXecution Environment	
Principle Of Least Astonishment	As FreeBSD evolves, changes visible to the user should be kept as unsurprising as possible. For example, arbitrarily rearranging system startup variables in `/etc/defaults/rc.conf` violates POLA. Developers consider POLA when contemplating user-visible system changes.
Problem Report	A description of some kind of problem that has been found in either the FreeBSD source or documentation. See Writing FreeBSD Problem Reports.
Process ID	A number, unique to a particular process on a system, which identifies it and allows actions to be taken against it.
Project Evil	The working title for the NDISulator, written by Bill Paul, who named it referring to how awful it is (from a philosophical standpoint) to need to have something like this in the first place. The NDISulator is a special compatibility module to allow Microsoft Windows™ NDIS miniport network drivers to be used with FreeBSD/i386. This is usually the only way to use cards where the driver is closed-source. See `src/sys/compat/ndis/subr_ndis.c` .

R

RA	Router Advertisement参照
RAID	Redundant Array of Inexpensive Disks参照
RAM	Random Access Memory参照

RD	Received Data参照
RFC	Request For Comments参照
RISC	Reduced Instruction Set Computer参照
RPC	Remote Procedure Call参照
RS232C	Recommended Standard 232C参照
RTS	Request To Send参照
Random Access Memory	
Revision Control System	The Revision Control System (RCS) is one of the oldest software suites that implement「revision control」for plain files. It allows the storage, retrieval, archival, logging, identification and merging of multiple revisions for each file. RCS consists of many small tools that work together. It lacks some of the features found in more modern revision control systems, like CVS or Subversion, but it is very simple to install, configure, and start using for a small set of files. Implementations of RCS can be found on every major UNIX-like OS. Concurrent Versions System, Subversion参照
Received Data	An RS232C pin or wire that data is received on. Transmitted Data参照
Recommended Standard 232C	A standard for communications between serial devices.
Reduced Instruction Set Computer	An approach to processor design where the operations the hardware can perform are simplified but made as general purpose as possible. This can lead to lower power consumption, fewer transistors and in some cases, better performance and increased code density. Examples of RISC processors include the Alpha, SPARC®, ARM® and PowerPC®.
Redundant Array of Inexpensive Disks	

Remote Procedure Call

repocopy — Repository Copy参照

Repository Copy — A direct copying of files within the CVS repository.

Without a repocopy, if a file needed to be copied or moved to another place in the repository, the committer would run `cvs add` to put the file in its new location, and then `cvs rm` on the old file if the old copy was being removed.

The disadvantage of this method is that the history (i.e. the entries in the CVS logs) of the file would not be copied to the new location. As the FreeBSD Project considers this history very useful, a repository copy is often used instead. This is a process where one of the repository meisters will copy the files directly within the repository, rather than using the cvs(1) program.

Request For Comments — A set of documents defining Internet standards, protocols, and so forth. See www.rfc-editor.org.

Also used as a general term when someone has a suggested change and wants feedback.

Request To Send — An RS232C signal requesting that the remote system commences transmission of data. Clear To Send参照

Router Advertisement

S

SCI — System Control Interrupt参照

SCSI — Small Computer System Interface参照

SG — Signal Ground参照

SMB — Server Message Block参照

SMP — Symmetric MultiProcessor参照

SMTP — Simple Mail Transfer Protocol参照

SMTP AUTH — SMTP Authentication参照

SSH — Secure Shell参照

STR — Suspend To RAM参照

SVN — Subversion参照

SMTP Authentication

Server Message Block

Signal Ground — An RS232 pin or wire that is the ground reference for the signal.

Simple Mail Transfer Protocol

Secure Shell

Small Computer System Interface

Subversion — Subversion is a version control system, similar to CVS, but with an expanded feature list.
Concurrent Versions System参照

Suspend To RAM

Symmetric MultiProcessor

System Control Interrupt

T

TCP — Transmission Control Protocol参照

TCP/IP — Transmission Control Protocol/Internet Protocol参照

TD — Transmitted Data参照

TFTP — Trivial FTP参照

TGT — Ticket-Granting Ticket参照

TSC — Time Stamp Counter参照

Ticket-Granting Ticket

Time Stamp Counter — A profiling counter internal to modern Pentium® processors that counts core frequency clock ticks.

Transmission Control Protocol — A protocol that sits on top of (e.g.) the IP protocol and guarantees that packets are delivered in a reliable, ordered, fashion.

Transmission Control Protocol/Internet Protocol — The term for the combination of the TCP protocol running over the IP protocol. Much of the Internet runs over TCP/IP.

Transmitted Data — An RS232C pin or wire that data is transmitted on.
Received Data参照

Trivial FTP

U

UDP — User Datagram Protocol参照

UFS1 — Unix File System Version 1参照

UFS2 — Unix File System Version 2参照

UID — User ID参照

URL — Uniform Resource Locator参照

USB — Universal Serial Bus参照

Uniform Resource Locator — A method of locating a resource, such as a document on the Internet and a means to identify that resource.

Unix File System Version 1 — The original UNIX® file system, sometimes called the Berkeley Fast File System.

Unix File System Version 2 — An extension to UFS1, introduced in FreeBSD 5-CURRENT. UFS2 adds 64 bit block pointers (breaking the 1T barrier), support for extended file storage and other features.

Universal Serial Bus
: A hardware standard used to connect a wide variety of computer peripherals to a universal interface.

User ID
: A unique number assigned to each user of a computer, by which the resources and permissions assigned to that user can be identified.

User Datagram Protocol
: A simple, unreliable datagram protocol which is used for exchanging data on a TCP/IP network. UDP does not provide error checking and correction like TCP.

V

VPN
: Virtual Private Network参照

Virtual Private Network
: A method of using a public telecommunication such as the Internet, to provide remote access to a localized network, such as a corporate LAN.

索引

シンボル

-CURRENT, 579
  using, 580
  コンパイル, 580
  使用, 580
-STABLE, 579
  利用する, 581
  構築、コンパイル, 582
.rhosts, 526
/boot/kernel.old, 291
/etc, 35
/etc/gettytab, 615
/etc/groups, 441
/etc/login.conf, 438
/etc/mail/access, 679
/etc/mail/aliases, 679
/etc/mail/local-host-names, 679
/etc/mail/mailer.conf, 679
/etc/mail/mailertable, 679
/etc/mail/sendmail.cf, 679
/etc/mail/virtusertable, 679
/etc/remote, 623
/etc/ttys, 617
/usr, 35
/usr/bin/login, 614
/usr/local/etc, 404
/usr/local/etc/rc.d, 405
/usr/share/skel, 433
/var, 35
10 base 2, 745
386BSD, 9, 10
386BSD Patchkit, 9
4.3BSD-Lite, 10
4.4BSD-Lite, 5, 6
802.11 (参照 無線ネットワーク)
アカウンティング
  ディスク領域, 542
アカウント
  daemon, 429
  nobody, 429
  operator, 429
  user, 430
  グループ, 441
  システム, 429
  スーパーユーザ (root), 431
  パスワードの変更, 437
  制限, 438
  削除, 434
  変更, 432
  追加, 433
アドレス転送, 794
アプリケーション
  Mathematica, 370
  Oracle, 372
  SAP R/3, 376
イエローページ (参照 NIS)
インストール
  トラブルシューティング, 64, 143
  ネットワーク
    FTP, 109
インターネットソフトウェアコンソーシアム (ISC), 767
インターネット接続共有, 791
イーサネット
  MAC アドレス, 371, 700
エディタ, 188
  emacs, 188
  vi, 188
エンコーディング, 554
オフィススイート
  Apache OpenOffice , 253
  Calligra, 252
  LibreOffice, 254
カーネル
  NOTES, 288
  カスタムカーネルの構築, 285
  コンフィグレーションファイル, 288
  構築 / インストール, 290
  設定, 262, 792
カーネル (kernel), 419
  起動フラグ, 424
  起動時の応答, 424
カーネルのチューニング, 373, 384
カーネルオプション
  LINUX, 366
  options BRIDGE, 726
クォータ (quotas), 438
グループ, 441
ゲートウェイ, 700

コアチーム, 12
コマンドライン, 185
コミッター, 12
コンソール, 157
コンソール (console), 426
コントリビュータ, 13
コンパイラ
  C, 6
  C++, 6
サウンドカード, 262
サブネット, 700, 701
サービス, 405
サービス妨害 (DoS), 446, 454
シェル (shell), 185
システムの設定と最適化, 403
シリアルコンソール, 626
シリアルポート, 302
シリアル通信, 601
シングルユーザモード, 423, 426, 587
シンボリックリンク, 368
スケルトンディレクトリ, 433
スタートアップスクリプト, 158
スライス, 175, 502
セキュリティ, 445
  crypt, 458
  DoS 攻撃 (参照 サービス妨害 (DoS))
  IPsec, 486
  OpenSSH, 493
  OpenSSL, 486
  S/Key, 459
  アカウント不正利用, 446
  ファイアウォール, 474
  安全性を高める, 448
  裏口 (バックドア), 447
セントロニクス (参照 パラレルプリンタ)
ソフトアップデート, 413
ソフトリミット, 544
ソースコード, 6
ゾーン
  例, 774
ダイアルアウトサービス, 622
ダイアルインサービス, 613
ダイナミックホストコンフィギュレーション
プロトコル (DHCP) (参照 DHCP)
テキストエディタ, 188
テープメディア, 521
  AIT, 524
  DDS (4mm) テープ, 521
  DLT, 523
  Exabyte (8mm) テープ, 522
  QIC テープ, 521
  QIC-150, 522
ディスク
  ファイルベース (4.X), 535
  ファイルベース (5.X), 536
  メモリ, 535
  メモリディスクの切り離し, 540
  メモリファイルシステム (4.X), 538
  メモリファイルシステム (5.X), 539
  仮想, 535
  暗号化, 546
  追加, 502
ディスククォータ, 438, 542
  チェック, 542, 545
  制限, 543
ディスクレスワークステーション, 734
ディスクレス稼働, 734
  カーネル設定, 739
  読み取り専用の /usr, 742
  起動, 736
ディレクトリ, 162
ディレクトリの階層構造, 165
デバイスノード, 265
デフォルトルート, 702
デフォルトルート (default route), 700
デュアルホームホスト, 703
ヌルモデムケーブル, 149, 602, 626
ヌルモデム用ケーブル, 297
ネットグループ, 759, 761
ネットワークアドレス変換 (NAT), 725
ネットワークプリンタ, 333, 334
ハードリミット, 544
バイナリ互換性
  BSD/OS, 6
  Linux, 6
  NetBSD, 6
  SCO, 6
  SVR4, 6
バイナリ互換機能
  Linux, 365
バックアップソフトウェア
  Amanda, 529

cpio, 528
pax, 528
tar, 527
ダンプ / リストア, 526
バックアップフロッピー, 524
バナーページ (参照 ヘッダページ)
バーチャルホスト, 406
パラレルライン IP, 800
パリティ, 297, 302
パーティション, 175, 502
ビデオ packages, 273
ビデオ ports, 273
ファイアウォール, 7, 474, 725, 726
ファイルの許可属性 (permissions), 162
ファイルシステム
fstab を使ったマウント, 177
HFS, 512
ISO 9660, 511, 512
Joliet, 512
アンマウント, 180
スナップショット, 540
マウント, 178
フォント, 372
LCD スクリーン, 231
TrueType, 228
アンチエイリアス, 229
スペーシング, 230
フォントのアンチエイリアス, 229
フロッピーディスク, 524
フローコントロールプロトコル, 297
ブラウザ
web, 246
ブリッジ, 724
ブートストラップ (bootstrap), 417
ブートマネージャ, 419
ブートマネージャ (Boot Manager), 418
ブートローダ (Boot Loader), 418
ブートローダ (boot-loader), 421
プリエンプティブマルチタスキング, 5
プリンタ, 562
USB, 296
ケイパビリティ, 303
シリアル, 296, 302, 308, 315
ネットワーク, 333, 334
パラレル, 296, 301
使い方, 348
利用制限, 337
プリンタスプール, 306
プリントジョブ, 294, 306, 315, 340, 349
制御, 340
ヘッダページ, 303, 326
ページ課金, 326
ボーレート, 297
マスターブートレコード (Master Boot Record: MBR), 418
マスターブートレコード (MBR), 419
マニュアルページ, 189
マルチキャスト
options MROUTING, 707
マルチユーザモード, 427
マルチユーザ機能, 5
メモリ保護, 6
メーリングリスト, 583
メールサーバデーモン
exim, 678
postfix, 678
qmail, 678
sendmail, 678
メールホスト, 679
モデム, 743
ユーザ
FreeBSD を利用している大規模サイト, 8
ユーザに対する制限, 438
coredumpsize, 439
cputime, 439
filesize, 440
maxproc, 440
memorylocked, 440
memoryuse, 440
openfiles, 440
sbsize, 440
stacksize, 441
クォータ, 438
リゾルバ, 773
リフレッシュレート, 220
ルータ, 8, 704, 725
ルーティング, 700
ルーティングの伝搬, 706
ルートゾーン, 773
ルートファイルシステム, 177
ディスクレス稼働, 740

ループバックデバイス (loopback device), 700
ログインクラス, 556, 557
ロケール, 384
ロケール (locale), 556
ロケール(locale), 554, 558
ローダ, 421
ローダの設定, 421
ワイドスクリーンフラットパネルコンフィグレーション, 226
仮想コンソール, 157
仮想ディスク, 535
仮想メモリ, 6
共有ライブラリ, 367
印刷, 293, 348
  フィルタ, 309, 311, 314
    apsfilter, 324
  ヘッダページ, 305
危険な専用, 175
国コード, 554
国際化 (internationalization) (参照 地域化 (localization))
圧縮, 525
地域化
  ドイツ語, 564
  ロシア語, 561
  日本語, 564
  繁体字中国語, 564
  韓国語, 564
地域化(localization), 553
垂直同期周波数 (参照 リフレッシュレート)
垂直走査レート (参照 リフレッシュレート)
基本入出力システム (Basic Input/Output System) (参照 BIOS)
対称型マルチプロセッシング (SMP), 6
水平同期周波数, 220
水平走査レート (参照 水平同期周波数)
無線ネットワーク, 708
  アクセスポイント, 709
  暗号化, 712
猶予期間, 545
環境変数 (environment variables), 186, 186
画像スキャナ, 279
砂場 (sandbox), 450
端末, 157, 607
繁体字中国語 (Traditional Chinese)
  BIG-5 エンコーディング, 556
自動マウントデーモン, 731
表計算ソフトウェア
  Gnumeric, 258
  KMyMoney, 258
言語コード, 554
許可属性
  シンボル, 164
許可属性 (permissions), 162
課金
  プリンタ, 326, 343
起動, 417
逆引き DNS, 773
電子メール, 7, 677

A

AbiWord, 252
ACL, 498
adduser, 433, 557
AIX, 747
Amanda, 529
amd, 731
Apache, 8
Apache OpenOffice , 253
Apple, 8
apsfilter, 324
ASCII, 359, 554
AT&T, 10

B

BGP, 704
BIND, 686, 773
  キャッシュネームサーバ, 782
  砂場での実行, 782
  設定ファイル, 776
  起動, 775
BIOS, 91, 418
bit/秒, 302
bits-per-second, 601
Bluetooth, 714
BOOTP
  ディスクレス稼働, 737
Bourne シェル (Bourne shells), 187
BSD Copyright, 11
BSD パーティション, 503

C

Calligra, 252
CD ライタ
  ATAPI, 512
  ATAPI/CAM ドライバ, 512, 517
CDROM
  ブータブル (起動可能な) CDROM の作成, 513
  作成, 511
  書き込み, 513, 514
chpass, 435
Chromium, 250
chroot, 782
Cisco, 8
Coda, 535
color
  contrast, 147
Computer Systems Research Group (CSRG), 6
comsat, 450
Concurrent Versions System (参照 CVS)
coredumpsize, 439
cpio, 528
cputime, 439
crypt, 458
CTM, 583, 823
cu1, 149
cuaa, 606
CUPS, 357
CVS, 12
CVS リポジトリ, 12

D

DCE, 601
Deleting obsolete files and directories, 591
DES, 458
device.hints, 425
DGA, 270
DHCP
  dhcpd.conf, 771
  インストール, 770
  サーバ, 769
  ディスクレス稼働, 736
  必要条件, 768
  設定ファイル, 769, 772
disklabel, 530, 534
DNS, 408, 678, 691, 773
  レコード, 780
DNS サーバ, 8
Documentation (参照 Updating and Upgrading)
documentation package (参照 Updating and Upgrading)
DOS, 77, 561
DoS 攻撃 (参照 サービス妨害 (DoS))
DSL, 725
DSP, 265
DTE, 601
dump, 526

E

editors
  ee, 188
ee, 188
ELF, 396
  branding, 396
emacs, 188
email, 7, 677
  MTA の変更, 682
  トラブルシューティング, 686
  受け取り, 678
  設定, 689
ePDFView, 256
Etherboot, 738
execution class loader, 396
Experts Exchange, 9

F

fdisk, 502
filesize, 440
finger, 450
Firefox, 246
fix-it フロッピー, 530
Flash, 247
FORTRAN, 319
Free Software Foundation, 10, 191
FreeBSD プロジェクト
  history, 9
  目標, 11
  開発モデル, 11

freebsd-update (参照 updating-upgrading)
Freecode, 196
FreshPorts, 195
FTP
  anonymous, 116, 137
  HTTP proxy 経由, 110
  Passive モード, 109
FTP サーバ, 7

G

getty, 614
Ghostscript, 317
GIMP, 252
GNOME, 235
GNU Compiler Collection, 8
GNU General Public License (GPL), 11
GNU Lesser General Public License (LGPL), 11
GNU ツールチェイン, 369
GnuCash, 257
Gnumeric, 258
GQview, 256
Greenman, David, 10
Grimes, Rod, 9
gv, 256
gzip, 525

H

HCI, 716
hostname, 408
hosts, 409
HP-UX, 747
HPLIP, 358
Hubbard, Jordan, 9
hw.ata.wc, 413

I

I/O ポート, 265
ICMP_BANDLIM, 456
IEEE, 528
ifconfig, 714
IMAP, 678, 679
init, 419, 426
installation, 15, 67
  from Windows, 153
  headless (serial console), 148
  network
    Ethernet, 154
    FTP, 152
    NFS, 155
    parallel (PLIP), 154
    serial (PPP), 154
Intel i810 graphic chipset, 225
ip aliases, 406
IP サブネット, 724
IP マスカレード, 791
ipfw, 476
  有効化, 477
  設定, 478
IPsec, 486
IPX/SPX, 747
IRQ, 265
ISDN, 725, 742
  カード, 743
  スタンドアロン ブリッジ/ルータ, 745
ISO 9660, 511

J

Jolitz, Bill, 9

K

KDE, 237
  ディスプレイマネージャ, 238
Kerberos, 450, 457, 466
  インストール, 466
kern.maxfiles, 414
kernel 制限のチューニング, 414
kernel.old, 423
keymap, 559
KLD (kernel loadable object), 366
KMyMoney, 258
Konqueror, 249

L

L2CAP, 717
LCD スクリーン, 231
LibreOffice, 254
Linux, 747
  /proc ファイルシステム, 365
  ELF バイナリ, 369

Linux ライブラリのインストール, 366
Linux バイナリ互換機能, 365
LISA, 529
log files, 410
LPD スプーリングシステム, 293
LPRng, 357
ls, 162

M

MacOS, 464
make.conf, 587
MAKEDEV, 607
maxproc, 440
MD5, 458
memorylocked, 440
memoryuse, 440
mencoder, 275
mergemaster , 590
Microsoft Windows, 90
MIME, 556, 558
MIT, 466
modem, 613
mount, 534
mount8, 148, 149
mountd, 728
moused, 559
MPlayer, 273
MS-DOS, 90, 359, 464
MX レコード, 690
MySQL, 561

N

NAT, 7
natd, 791
Net/2, 10, 10
NetApp, 8
NetBSD, 747
Netcraft, 9
NetEase, 9
Netflix, 9
newfs, 534
newsyslog.conf, 410
NFS, 535, 545, 727
エクスポートの例, 729
サーバ, 728
ディスクレス稼働, 738
マウント, 730
使用法, 731
複数のマシンにインストール, 595
設定, 728
nfsd, 728
NIS, 747
クライアント, 750
クライアントの設定, 756
サーバ設定, 752
スレーブサーバ, 750, 754
ドメイン, 748
ドメイン名, 751
パスワード形式, 766
マスタサーバ, 749
マップ, 752
NOTES, 288
Novell, 10
ntalk, 450
NTP, 788
ntp.conf, 789
ntpd, 788
サーバ選択, 788
設定, 788
ntpdate, 788

O

OBEX, 722
Okular, 257
OpenBSD, 747
openfiles, 440
OpenSSH, 493
secure copy, 494
クライアント, 494
トンネリング, 496
有効化, 493
設定, 495
OpenSSL, 486
インストール, 486
Opera, 248
OS/2, 359, 504
OSPF, 704

P

packages, 193
Pair Networks, 9
pairing, 719

partition layout, 35
passwd, 437
pax, 528
PCI, 262
PCL, 301, 360
PDF
  viewing, 256, 257
  ビューア, 255, 256
pgp公開鍵, 863
PLIP, 800
POP, 678, 679
portmap, 728, 748
portmaster, 210
ports, 193
  installing, 205
  アップグレード, 209
  ディスク容量, 212
  削除, 209
ports collection, 366
portupgrade, 212
POSIX, 528, 555
PostScript, 296, 300
  エミュレーション, 317
  ビューア, 256
PPP, 687, 744
pw, 438, 557

R

RAID
  CCD, 505
  Vinum, 509
  ソフトウェア, 505, 509
  ハードウェア, 510
Rambler, 8
rc files
  rc.conf, 403
rc ファイル
  rc.serial, 606, 618
rc ファイル群, 427
resolv.conf, 408
restore, 526
RFCOMM, 718
RIP, 704
rlogind, 450
rmuser, 434
root パーティション, 534
RPMs, 380
RS-232C ケーブル, 602, 603
rshd, 450

S

S/Key, 459
sbsize, 440
scp, 494
screenmap, 559
SCSI, 91
SDL, 270
SDP, 720
sendmail, 451, 679
serial console, 148
shutdown, 427
signal 11, 593
Sina, 9
SMTP, 690
Solaris, 396, 747
Sony Japan, 9
src.conf, 588
SSH
  sshd, 116
ssh, 457
sshd, 450
stacksize, 441
su, 448, 503
Subversion, 12, 582, 582, 826
Subversion Repository
  ミラーサイト, 828
Subversion リポジトリ, 12
SunOS, 751
SVN (参照 Subversion)
swap partition, 35
swap sizing, 35
sysctl, 411, 412, 452
sysctl によるチューニング, 412
sysctl.conf, 411
sysinstall, 559, 768
  ディスクの追加, 503
syslog.conf, 410
sysutils/mkisofs, 512

T

tar, 525, 527
TCP/IP ネットワーキング, 6

tcpwrapper, 758
TELEHOUSE America, 9
telnetd, 450
TeX, 294, 352
　DVI ファイルの印刷, 318
TFTP
　ディスクレス稼働, 738
traceroute, 707
troff, 313
Tru64 UNIX, 753
TrueType フォント, 228
ttyd, 606
tunefs, 413
TV カード, 276

U

U.C. Berkeley, 10, 10
UDP, 768
Unicode, 561
UNIX, 162
Updating and Upgrading, 566, 575, 577
USENET, 7
UUCP, 687

V

vfs.vmiodirenable, 412
vi, 188
vipw, 557

W

Walnut Creek CDROM, 10
Weathernews, 9
web サーバ, 7
WEP, 712
wheel, 448
Williams, Nate, 9
Windows, 464
Windows NT, 748
world の再構築, 583
　時間, 589

X

X Window System, 6, 8
　Accelerated-X, 8
X ディスプレイマネージャ, 232
X11, 220
X11 Input Method (XIM), 560
X11 True Type フォントサーバ, 560
X11 のチューニング, 224
XML, 229
Xorg, 220
xorg.conf, 224
Xpdf, 255
XVideo, 270

Y

Yahoo!, 8
Yandex, 8

奥付

この本は「The FreeBSD Documentation Project」に対するたくさんの人々の共同作業によって完成したものです。文書は DocBook DTD を用いた XML 形式で管理されており、XSLT を使って XML からさまざまな可読形式に組版されています。もし、Donald Knuth 氏による組版言語 TeX、Leslie Lamport 氏による LaTeX および Sebastian Rahtz 氏による JadeTeX マクロパッケージがなかったら、この文書の書籍版を実現することはできなかったでしょう。