FreeBSD et les polices de caractères

Un guide utilisateur

Dave Bodenstab

Version: 43184
2013-11-13 par hrs.

Résumé

<imdave@synet.net >

Mercredi 7 Aout 1996

Ce document décrit les différents fichiers de polices de caractères qui peuvent être utilisés avec FreeBSD et le pilote syscons, X11, GhostScript et Groff. Des exemples immédiatement applicables sont donnés pour passer l'affichage à la console en mode 80x60 et pour se servir des polices “Type 1” avec les programmes d'application précités.

Version française de Frédéric Haby <frederic.haby@mail.dotcom.fr >.

Table des matières

1. Introduction 1
2. Terminologie élémentaire 2
3. Quels formats de polices de caractères puis-je utiliser? 2
4. Configurer une console virtuelle en mode 80x60 lignes 3
5. X11 et les polices “Type 1” 4
6. Utiliser les polices “Type 1” avec GhostScript 7
7. Utiliser les polices “Type 1” avec Groff 8
8. Peut-on utiliser des polices “True Type”? 10
9. Où peut-on trouver des polices supplémentaires? 11
10. D'autres questions 12

1. Introduction

Il y a de nombreuses sources de polices de caractères disponibles, et l'on peut se demander comment les utiliser avec FreeBSD. La réponse peut être obtenue en détaillant attentive-

ment la documentation du composant voulu. Cela peut prendre beaucoup de temps, ce guide essaie donc de fournir un raccourci pour les autres personnes que cela intéresserait.

2. Terminologie élémentaire

Il y a de nombreux formats de polices de caractères avec chacun son propre suffixe de nom de fichier. Quelques uns d'entre eux seront discutés dans ce document:

`.pfa`, `.pfb`
: Polices PostScript “Type 1”. `.pfa` est le format *A*scii et `.pfb` le format *B*inaire.

`.afm`
: Le fichier de dimensions d'une police “Type 1”.

`.pfm`
: Le fichier de dimensions pour l'impression d'une police “Type 1”.

`.ttf`
: Police “True Type”.

`.fot`
: Référence indirecte à une police “True Type” (ce n'est pas la police elle-même).

`.fon`, `.fnt`
: Police de caractères “bitmap” pour affichage à l'écran.

Les fichiers `.fot` sont employés par Windows un peu de la même façon que des liens symboliques vers le fichiers de police “True Type” (`.ttf`). Les fichiers `.fon` sont aussi utilisés par Windows. Je ne connais pas de moyen de se servir de ces fichiers avec FreeBSD.

3. Quels formats de polices de caractères puis-je utiliser?

Le format de fichier de police de caractères utile dépend de l'application. FreeBSD lui-même n'utilise pas de police de caractères. Ce sont les programmes d'application et/ou les pilotes qui se servent éventuellement de ces fichiers. Voici un petit tableau associant application/pilote de périphérique et suffixe des fichiers de police de caractères:

	`.fnt`	`.pfa`	`.pfb`	`.ttf`	`.afm`
Pilote `sys-cons`	Oui				
GhostScript		Oui	Oui	Oui	
X11		Oui	Oui		

	.fnt	.pfa	.pfb	.ttf	.afm
Groff		Oui			Oui
Povray				Oui	

Le suffixe `.fnt` est assez fréquemment utilisé. Je soupçonne que lorsque quelqu'un veut créer un fichier de police de caractères spécialisé pour son application, il choisit la plupart du temps ce suffixe. Il y a donc des chances pour que des fichiers avec ce suffixe n'aient pas tous le même format; en particulier, les fichiers `.fnt` utilisés par le pilote syscons de FreeBSD n'auront peut-être pas le même format qu'un fichier `.fnt` utilisé dans l'environnement DOS/Windows. Je n'ai jamais essayé d'utiliser d'autres fichiers `.fnt` que ceux fournis avec FreeBSD.

4. Configurer une console virtuelle en mode 80x60 lignes

Il faut d'abord charger une police de caractères 8x8. Le fichier `/etc/sysconfig` [1] doit donc contenir les lignes:

```
# choose font 8x8 from /usr/share/syscons/fonts/* (or NO for default)
font8x8=/usr/share/syscons/fonts/cp437-8x8.fnt
```

La commande pour changer de mode d'affichage est vidcontrol(1):

```
bash$ vidcontrol VGA_80x60
```

Différents programmes en mode plein-écran, tel vi(1), doivent pouvoir déterminer la taille actuelle de l'écran. Elle peut être configurée avec stty(1):

```
bash$ stty crt rows 60 columns 80
```

Pour rendre cela moins fastidieux, ces commandes peuvent être introduites dans une procédure exécutée au démarrage du système. Voici une façon de le faire:

1. Modifiez `/etc/sysconfig` comme expliqué plus haut,

2. Ajoutez les lignes suivantes à `/etc/rc.local` :

```
for tty in /dev/ttyv?
do
  vidcontrol VGA_80x60 <$tty >/dev/null 2<&1
```

[1] N.d.T.: `/etc/rc.conf` dans les versions récentes de FreeBSD.

```
done
```

3. Ajoutez les lignes suivantes à `/etc/profile`:

```
TTYNAME=`basename \`tty\``
if expr "$TTYNAME" : 'ttyv' > /dev/null
then
  stty crt rows 60 columns 80
fi
```

Références: stty(1), vidcontrol(1).

5. X11 et les polices “Type 1”

X11 peut employer les formats de police de caractères `.pfa` et `.pfb`. Il y a une référence croisée de chaque fichier sur son nom X11 par l'intermédiaire du fichier `fonts.dir` de chaque répertoire.

Il y a déjà un répertoire appelé `Type1`. La façon la plus immédiate d'ajouter une police de caractères est de la mettre dans ce répertoire. Une meilleure manière de procéder est de mettre toutes les nouvelle polices dans un répertoire distinct et d'utiliser un lien symbolique sur ces nouvelles polices. Il est alors plus facile de distinguer ses propres polices de celles qui sont fournies d'origine. Par exemple:

- *Créer un repertoire pour les fichiers de police de caractères.*

```
bash$ mkdir -p /usr/local/share/fonts/type1
bash$ cd /usr/local/share/fonts/type1
```

- *Y mettre les fichiers `.pfa`, `.pfb` et `.afm`. On peut aussi vouloir y conserver les fichiers `README` et autres documentations concernant les polices de caractères.*

```
bash$ cp /cdrom/fonts/atm/showboat/showboat.pfb .
bash$ cp /cdrom/fonts/atm/showboat/showboat.afm .
```

- *Tenir à jour un index pour créer les références croisées sur les polices.*

```
bash$ echo showboat - InfoMagic CICA, Dec 1994, /fonts/atm/
showboat >>INDEX
```

Pour pouvoir maintenant utiliser une nouvelle police de caractères, il faut mettre le fichier à disposition, et mettre à jour le fichier des noms de polices. Les noms de police X11 se présentent comme suit:

```
```

```
-bitstream-charter-medium-r-normal-xxx-0-0-0-0-p-0-iso8859-1
     |        |      |    |   |     |  | | | | | |    \    \
     |        |      |    |   |     \  \ \ \ \ \ \     +----+- jeu ↺
de caractères
     |        |      |    |   \      \  \ \ \ \ \ +- largeur moyenne
     |        |      |    |    \      \  \ \ \ \ +- espacement
     |        |      |    \     \      \  \ \ \ +- résolution ↺
verticale
     |        |      |     \     \      \  \ \ +- résolution ↺
horizontale
     |        |      |      \     \      \  \ +- points
     |        |      |       \  largeur   \  +- pixels
     |        |      |        \            \
   casse   famille graisse inclinaison style supplémentaire
```

Il faut créer un nouveau nom pour chaque nouvelle police. Si la documentation qui l'accompagne vous donne quelques informations, elle peut servir de base pour définir ce nom. Si vous n'avez aucune information, vous pouvez utiliser la commande strings(1) sur le fichier de police. Par exemple:

```
bash$ strings showboat.pfb | more
```

```
%!FontType1-1.0: Showboat 001.001
%%CreationDate: 1/15/91 5:16:03 PM
%%VMusage: 1024 45747
% Generated by Fontographer 3.1
% Showboat
 1991 by David Rakowski.  Alle Rechte Vorbehalten.
FontDirectory/Showboat known{/Showboat findfont dup/UniqueID ↺
known{dup
/UniqueID get 4962377 eq exch/FontType get 1 eq and}{pop false}ifelse
{save true}{false}ifelse}{false}ifelse
12 dict begin
/FontInfo 9 dict dup begin
 /version (001.001) readonly def
 /FullName (Showboat) readonly def
 /FamilyName (Showboat) readonly def
 /Weight (Medium) readonly def
 /ItalicAngle 0 def
 /isFixedPitch false def
 /UnderlinePosition -106 def
 /UnderlineThickness 16 def
 /Notice (Showboat
 1991 by David Rakowski.  Alle Rechte Vorbehalten.) readonly def
end readonly def
/FontName /Showboat def
--stdin--
```

A partir de ces informations, le nom pourrait être:

```
-type1-Showboat-medium-r-normal-decorative-0-0-0-0-p-0-iso8859-1
```

Les composantes de ce nom sont:

Casse
: Appelons simplement toutes nos nouvelles polices `type1`.

Famille
: Le nom de la police.

Graisse
: Normal, gras, médium, semi-gras, etc. D'après les résultats de strings(1) ci-dessus, la police a une graisse *médium*.

Inclinaison
: *r*oman, *i*talique ou *o*blique. Comme *ItaliqueAngle* vaut 0, nous utiliserons *roman*.

Largeur
: Normale, large, condensée, étendue, etc. Jusqu'à ce que nous la déterminions à l'usage, nous supposerons qu'elle est *normale*.

Style supplémentaire
: Habituellement non renseigné, mais nous nous en servons pour indiquer que la police contient des majuscules décoratives.

Espacement
: Proportionnel ou fixe. Comme *isFixedPitch* est faux, nous utilisons *Proportionnel*.

Tous ces noms sont arbitraires, mais il faut essayer de rester compatible avec les conventions existantes. Une police est connue d'une application X11 sous un nom qui peut éventuellement comporter des caractères de substitution, il faut donc choisir un nom significatif. On peut commencer en utilisant simplement:

```
...-normal-r-normal-...-p-...
```

comme nom, puis se servir de xfontsel(1) pour visualiser la police et affiner son nom en fonction de ce à quoi elle ressemble.

Donc, pour compléter notre exemple:

- *Rendre la police accessible à X11*

```
bash$ cd /usr/X11R6/lib/X11/fonts/Type1
```

```
bash$ ln -s /usr/local/share/fonts/type1/showboat.pfb .
```

- *Editer `fonts.scale` et `fonts.dir` pour y ajouter une ligne décrivant la nouvelle police et incrémenter le nombre de polices qui est défini en première ligne*

```
bash$ ex fonts.dir
:1p
25
:1c
26
.
:$a
showboat.pfb -type1-showboat-medium-r-normal-decorative-0-0-0-0-p-0-
iso8859-1
.
:wq
```

- *`fonts.scale` et `fonts.dir` sont apparemment identiques*

```
bash$ cp fonts.dir fonts.scale
```

- *Informer X11 des modifications*

```
bash$ xset fp rehash
```

- *Visualiser la nouvelle police*

```
bash$ xfontsel -pattern -type1-*
```

Références: xfontsel(1), xset(1), *The X Window System in a Nutshell*, O'Reilly & Associates.

6. Utiliser les polices “Type 1” avec GhostScript

GhostScript référence les polices via son fichier `Fontmap`. Il doit donc être modifié de la même façon que le fichier X11 `fonts.dir`. GhostScript peut utiliser des fichiers de police de caractères aux formats `.pfa` ou `.pfb`. Voici comment nous utiliserions la police de l'exemple précédent avec GhostScript:

- *Mettre la police dans le répertoire des polices de GhostScript*

```
bash$ cd /usr/local/share/ghostscript/fonts
bash$ ln -s /usr/local/share/fonts/type1/showboat.pfb .
```

- *Editer le fichier `Fontmap` pour que GhostScript ait connaissance de la nouvelle police*

```
bash$ cd /usr/local/share/ghostscript/4.01
bash$ ex Fontmap
:$a
/Showboat       (showboat.pfb) -; % From CICA /fonts/atm/showboat
```

```
:wq
```

- *Utiliser GhostScript pour visualiser la police*

```
bash$ gs prfont.ps
```

```
Aladdin Ghostscript 4.01 (1996-7-10)
Copyright (C) 1996 Aladdin Enterprises, Menlo Park, CA.  All rights
reserved.
This software comes with NO WARRANTY: see the file PUBLIC for ↺
details.
Loading Times-Roman font from /usr/local/share/ghostscript/fonts/
tir_____.pfb...
 /1899520 581354 1300084 13826 0 done.
```

```
GS> Showboat DoFont
```

```
Loading Showboat font from /usr/local/share/ghostscript/fonts/
showboat.pfb...
 1939688 565415 1300084 16901 0 done.
>>showpage, press <return> to continue<<
>>showpage, press <return> to continue<<
>>showpage, press <return> to continue<<
```

```
GS> quit
```

Références: `fonts.txt` de la distribution de GhostScript 4.01.

7. Utiliser les polices “Type 1” avec Groff

Maintenant que nous pouvons utiliser la nouvelle police avec X11 et GhostScript, comment faire pour s'en servir aussi avec groff? Tout d'abord, comme nous nous occupons de polices PostScript “Type 1”, le périphérique groff à utiliser est `ps`. Il faut créer un fichier de police pour chaque police utilisée par groff. Le nom d'une police groff se résume à un fichier dans le répertoire `/usr/share/groff_font/devps` . Dans notre exemple, ce pourrait être `/usr/share/groff_font/devps/SHOWBOAT` . Il faut créer ce fichier avec les outils fournis par groff.

Le premier outil est `afmtodit`. Il n'est pas normalement installé et doit donc être extrait de la distribution sous forme de sources. Je me suis aperçu qu'il fallait modifier la première ligne du fichier, voici donc ce que j'ai fait:

```
bash$ cp /usr/src/gnu/usr.bin/groff/afmtodit/afmtodit.pl /tmp
bash$ ex /tmp/afmtodit.pl
```

```
:1c
#!/usr/bin/perl -P-
.
:wq
```

Cet outil crée le fichier de police groff à partir du fichier de dimensions de la police (suffixe `.afm`). Pour continuer avec notre exemple:

- *De nombreux fichiers `.afm` sont au format Mac... avec des lignes terminées par des ^M*

  Il faut les convertir au style Unix avec des lignes terminées par des ^J

```
bash$ cd /tmp
bash$ cat /usr/local/share/fonts/type1/showboat.afm |
       tr '\015' '\012' >showboat.afm
```

- *Créons maintenant le fichier de police groff*

```
bash$ cd /usr/share/groff_font/devps
bash$ /tmp/afmtodit.pl -d DESC -e text.enc /tmp/showboat.afm ↺
generate/textmap SHOWBOAT
```

La police peut maintenant être référencée par le nom SHOWBOAT.

Si l'on utilise GhostScript pour piloter les imprimantes du système, il n'y a rien d'autre à faire. Si, par contre, l'on utilise de vraies imprimantes PostScript, il faut alors charger la police sur l'imprimante pour pouvoir l'utiliser (à moins que l'imprimante ne dispose de la police “showboat” en interne ou sur une disquette de polices accessible). La dernière étape consiste à créer la police chargeable. L'outil `pfbtops` est utilisé pour créer le format `.pfa` de la police et le fichier `download` est modifié pour faire référence à la nouvelle police. Le fichier `download` doit indiquer le nom interne de la police. Ce nom peut être facilement connu à l'aide du fichier de police groff, comme le montre l'exemple:

- *Créer le fichier de police `.pfa`*

```
bash$ fgrep internalname SHOWBOAT
internalname Showboat
```

- *Dire à groff qu'il faut charger la police*

```
bash$ ex download
:$a
Showboat      showboat.pfa
.
:wq
```

Pour tester la police:

```
bash$ cd /tmp
bash$ cat >exemple.t <<EOF
```

```
.sp 5
.ps 15
C'est un exemple de police Showboat:
.br
.ps 48
.vs (\n(.s+2)p
.sp
.ft SHOWBOAT
ABCDEFGHI
.br
JKLMNOPQR
.br
STUVWXYZ
.sp
.ps 16
.vs (\n(.s+2)p
.fp 5 SHOWBOAT
.ft 5
Utilisée comme première lettre d'un paragraphe. Cela ressemblera à:
.sp50p
\s(48\f5V\s0\fRoici la première phrase d'un paragraphe qui utilise
la police showboat pour sa première lettre. Il faut augmenter
l'espacement vertical pour laisser de la place pour la première
lettre.
EOF
bash$ groff -Tps exemple.t >exemple.ps
```

- *Pour utiliser GhostScript/GhostView*

```
bash$ ghostview exemple.ps
```

- *Pour l'imprimer*

```
bash$ lpr -Ppostscript exemple.ps
```

Références: `/usr/src/gnu/usr.bin/groff/afmtodit/afmtodit.man` , groff_font(5), groff_char(5), pfbtops(1).

8. Peut-on utiliser des polices “True Type”?

Le format de police “True Type” est utilisée par Windows, Windows 95 et le Macintosh. Il est assez répandu et il y a de nombreuses polices de ce format disponibles.

Malheureusement, je connais peu d'applications qui puissent utiliser ce format: GhostScript et Povray viennent à l'esprit. Le support par GhostScript est, d'après la documentation, assez rudimentaire et il y a des chances que le résultat soit moins bon qu'avec les polices “Type 1”. Povray version 3 est aussi capable d'utiliser des des polices “True Type”, mais je doute que beaucoup de personnes créent des documents sous forme de pages générées par lancer de rayon :-).

Il est difficile d'utiliser une police “True Type” avec groff parce que groff a besoin d'un fichier de description de la police, et je ne connais pas d'outil pour déterminer les dimensions d'une police “True Type”. De plus, il faudrait charger la police sur les imprimantes PostScript avec le format approprié et, de nouveau, groff ne peut gérer de cette façon les polices “True Type”.

Cette situation plutôt désolante changera peut-être bientôt. Le Projet FreeType développe actuellement un ensemble d'utilitaire FreeType:

- Le serveur de polices `xfsft` pour X11 peut gérer les polices “TrueType” en plus des polices ordinaires. Bien que ce soit encore une version béta, elle a la réputation d'être à peu près utilisable. Voyez la page de Juliusz Chroboczek pour plus d'informations. Il y a des instructions de portage sous FreeBSD sur la page logicielle de Stephen Montgomery.

- `xfstt` est un autre serveur X11, disponible sur ftp://sunsite.unc.edu/pub/Linux/X11/fonts.

- Il y a un programme appelé `ttf2bdf` qui peut créer des fichiers BDF utilisables en environnement X à partir de fichiers “TrueType”. Les binaires pour Linux sont censés être disponibles sur ftp://crl.nmsu.edu/CLR/multiling/General/.

- Pour les personnes qui ont besoin de polices “True Type” asiatiques, il peut valoir la peine de jetter un coup d'oeil au serveur de polices XTT. Il y a des informations sur XTT à l'adresse: http://hawk.ise.chuo-u.ac.jp/student/person/tshiozak/study/freebsd-at-random/x-tt/index-en.html.

- et d'autres ...

9. Où peut-on trouver des polices supplémentaires?

Il y a de nombreuses polices disponibles sur l'Internet. Elles sont soit complètement gratuites ou en “partagiciel”. Il y a aussi de nombreux CD-ROMs peux onéreux qui contiennent beaucoup de polices. Voici quelques sites Internet (au mois d'Août 1996):

- ftp://ftp.winsite.com (auparavant CICA),

- http://www.simtel.net/,

- ftp://ftp.coast.net/,

- http://af-pc-plloyd.ecel.uwa.edu.au/fonts/index.html,

- http://www.esselte.com/letraset/index.html,

- http://www.inil.com/users/elfring/esf.htm.

10. D'autres questions

- Que sont que les fichiers `.pfm`?
- Peut-on générer des fichiers `.afm` à partir de fichiers `.pfa` ou `.pfb`?
- Comment générer des fichiers de correspondance de caractères pour groff pour les polices PostScript dont les caractères ont des noms qui ne sont pas standard?
- Les périphériques `xditview` et `devX??` peuvent-ils être configurés pour accéder aux nouvelles polices?
- Il serait bien d'avoir des exemples d'utilisation de polices “True Type” avec GhostScript et Povray.