Introducción a NanoBSD

Daniel Gerzo

Revisión: 43184

Copyright © 2006 The FreeBSD Documentation Project

FreeBSD is a registered trademark of the FreeBSD Foundation.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and the FreeBSD Project was aware of the trademark claim, the designations have been followed by the "™" or the "®" symbol.

2013-11-13 por hrs.

Resumen

Este documento trata sobre NanoBSD. NanoBSD es una herramienta que permite crear imágenes del sistema FreeBSD para su uso en aplicaciones empotradas y listas para cargarse en una tarjeta Compact Flash (u otro medio de almacenamiento masivo).

Traducción de Germán Marcos <gri.msg@gmail.com >.

Tabla de contenidos

1. Introducción a NanoBSD 1
2. NanoBSD "Howto" 2
Índice 8

1. Introducción a NanoBSD

NanoBSD es una herramienta actualmente desarrollada por Poul-Henning Kamp. Crea una imagen del sistema FreeBSD para aplicaciones empotradas lista para su uso en una tarjeta Compact Flash (u otro medio de almacenamiento masivo).

Puede utilizarse para generar imágenes instalables especiales; está diseñado pensando en una instalación y mantenimiento fáciles en sistemas conocidos como "sistemas empotrados". Los sistemas empotrados tienen hardware y software integrado en el producto, lo

que significa que todas las aplicaciones están preinstaladas. Estos dispositivos pueden conectarse a una red dada y comenzar a trabajar (casi) inmediatamente.

Las características de NanoBSD incluyen:

- Los ports y paquetes funcionan como en FreeBSD: Cualquier aplicación puede instalarse y usarse en una imagen de NanoBSD de la misma forma que en FreeBSD.
- No se pierde funcionalidad: Si es posible hacer algo en FreeBSD es posible hacer lo mismo en NanoBSD, a menos que se haya eliminado explícitamente esa característica o características especiales al crear la imagen de NanoBSD.
- Todo está en modo de sólo lectura durante el funcionamiento del sistema: Un apagado brusco es totalmente seguro. No es necesario ejecutar fsck(8) después de que el sistema sufra un apagón.
- Fácil de generar y personalizar: Usando un solo script de shell y un fichero de configuración pueden generarse imágenes reducidas y personalizadas que satisfagan cualquier necesidad.

2. NanoBSD “Howto”

2.1. El diseño de NanoBSD

Una vez que la imagen está en el medio puede arrancar NanoBSD. El medio de almacenamiento se divide por omisión en tres partes:

- Dos particiones de la imagen: `code#1` y `code#2`.
- La partición que contiene los ficheros de configuración, que puede montarse bajo `/cfg` durante el funcionamiento del sistema.

Estas particiones se montan normalmente en modo de sólo lectura.

Los directorios `/etc` y `/var` son discos md(4) (malloc).

La partición de los ficheros de configuración permanece bajo el directorio `/cfg`. Este directorio contiene ficheros que usa el directorio `/etc` y se monta en modo de sólo lectura inmediatamente después del arranque; por lo tanto, es necesario que los ficheros que se modifiquen en `/etc` se copien en `/cfg` si se desea que los cambios perduren después del reinicio del sistema.

Ejemplo 1. Cómo hacer cambios persistentes en `/etc/resolv.conf`

```
# vi /etc/resolv.conf
```

```
[...-]
# mount /cfg
# cp /etc/resolv.conf /cfg
# umount /cfg
```

Nota

La partición que contiene el directorio `/cfg` debería montarse únicamente en el arranque y mientras se sobreescriben los ficheros de configuración.

No es buena idea montar siempre el directorio `/cfg`, especialmente si el sistema NanoBSD se ejecuta en un dispositivo de almacenamiento masivo que pueda verse afectado negativamente por un número elevado de escrituras en la partición (i.e. cuando el sistema de ficheros sincroniza los datos con el sistema de discos).

2.2. Cómo generar una imagen de NanoBSD

Una imagen de NanoBSD se genera usando el sencillo script de shell `nanobsd.sh` , ubicado en el directorio `/usr/src/tools/tools/nanobsd` . El script crea una imagen que se deberá copiar al medio de almacenamiento mediante la utilidad dd(1).

Los comandos necesarios para generar una imagen de NanoBSD son:

```
# cd /usr/src/tools/tools/nanobsd   ❶
# sh nanobsd.sh  ❷
# cd /usr/obj/nanobsd.full   ❸
# dd if=_.disk.full of=/dev/da0 bs=64k   ❹
```

❶ Cambia el directorio actual por el directorio base del script de generación de NanoBSD.
❷ Comienza el proceso de generación.
❸ Cambia el directorio actual por el directorio donde se encuentra la imagen generada.
❹ Instala NanoBSD dentro del medio de almacenamiento.

2.3. Personalización de una imagen de NanoBSD

Esta es probablemente la característica más importante y más interesante de NanoBSD. Tambié es donde usted pasará la mayor parte del tiempo cuando esté desarrollando con NanoBSD.

La invocación del siguiente comando forzará al script `nanobsd.sh` a leer su configuración desde el fichero `mi-configuracion.nano` ubicado en el directorio actual:

```
# sh nanobsd.sh -c mi-configuracion.nano
```

La personalización se hace de dos maneras:

- Opciones de configuración
- Funciones personalizadas

2.3.1. Opciones de configuración

Por medio de ajustes de configuración es posible configurar las opciones que se pasan a las fases `buildworld` e `installworld` del proceso de generación de NanoBSD. Mediante estas opciones puede reducirse el tamaño del sistema de manera que pueda incluso encajar en algo tan pequeño como un dispositivo de 64MB. Es posible recortar FreeBSD incluso más, hasta que consista únicamente en el kernel y dos o tres ficheros de "userland".

El fichero de configuración está compuesto por opciones de configuración que sobreescriben los valores por omisión. Las directivas más importantes son:

- `NANO_NAME` : Nombre de la generación que estamos ejecutando (se usa para dar nombres a los directorios donde encontraremos el resultado del proceso).
- `NANO_SRC` : Ruta al código fuente que se usará para generar la imagen.
- `NANO_KERNEL` : Nombre del fichero de configuración del kernel que se usará para generar el kernel.
- `CONF_BUILD` : Opciones que se pasan a la fase `buildworld`.
- `CONF_INSTALL` : Opciones que se pasan a la fase `installworld`.
- `CONF_WORLD` : Opciones que se pasan tanto a `buildworld` como a `installworld`.
- `FlashDevice` : Define el tipo de medio de almacenamiento que se usará. Para más información consulte el fichero `FlashDevice.sub` .

2.3.2. Funciones personalizadas

Puede ajustar NanoBSD de forma muy precisa mediante el uso de funciones de shell en el fichero de configuración. En el siguiente ejemplo vemos un modelo básico de función personalizada.

```
cust_foo () (
 echo "bar=topless" > \
   ${NANO_WORLDDIR}/etc/foo
)
customize_cmd cust_foo
```

A continuación vemos un ejemplo más útil de función personalizada, que cambia el tamaño por omisión del directorio `/etc` de 5MB a 30MB:

```
cust_etc_size () (
 cd ${NANO_WORLDDIR}/conf
 echo 30000 > default/etc/md_size
)
customize_cmd cust_etc_size
```

Hay unas pocas funciones de personalización predefinidas listas para su uso:

- `cust_comconsole` : Deshabilita getty(8) en los dispositivos VGA (los nodos de dispositivo `/dev/tty*`) y habilita el uso del puerto serie COM1 para que sea la consola del sistema.
- `cust_allow_ssh_root` : Permite a `root` acceder a través de sshd(8).
- `cust_install_files` : Instala ficheros de configuración desde el directorio `nanobsd/files`, que contiene scripts útiles para la administración del sistema.

2.3.3. Cómo añadir paquetes

Se pueden añadir paquetes a la imagen de NanoBSD usando una función personalizada. La siguiente función instalará todos los paquetes que se encuentren en `/usr/src/tools/tools/nanobsd/packages` :

```
install_packages () (
    mkdir -p ${NANO_WORLDDIR}/packages
    cp /usr/src/tools/tools/nanobsd/packages/* ${NANO_WORLDDIR}/
packages
    chroot ${NANO_WORLDDIR} sh -c 'cd packages; pkg_add -v *;cd ..-;'
    rm -rf ${NANO_WORLDDIR}/packages
)
customize_cmd install_packages
```

2.3.4. Ejemplo de fichero de configuración

El siguiente es un ejemplo completo de fichero de configuración para generar una imagen personalizada de NanoBSD:

```
NANO_NAME=custom
NANO_SRC=/usr/src
NANO_KERNEL=MYKERNEL
NANO_IMAGES=2

CONF_BUILD='
NO_KLDLOAD=YES
NO_NETGRAPH=YES
NO_PAM=YES
'

CONF_INSTALL='
```

```
NO_ACPI=YES
NO_BLUETOOTH=YES
NO_CVS=YES
NO_FORTRAN=YES
NO_HTML=YES
NO_LPR=YES
NO_MAN=YES
NO_SENDMAIL=YES
NO_SHAREDOCS=YES
NO_EXAMPLES=YES
NO_INSTALLLIB=YES
NO_CALENDAR=YES
NO_MISC=YES
NO_SHARE=YES
'

CONF_WORLD='
NO_BIND=YES
NO_MODULES=YES
NO_KERBEROS=YES
NO_GAMES=YES
NO_RESCUE=YES
NO_LOCALES=YES
NO_SYSCONS=YES
NO_INFO=YES
'

FlashDevice SanDisk 1G

cust_nobeastie() (
 touch ${NANO_WORLDDIR}/boot/loader.conf
 echo "beastie_disable=\"YES\"" >> ${NANO_WORLDDIR}/boot/loader.conf
)

customize_cmd cust_comconsole
customize_cmd cust_install_files
customize_cmd cust_allow_ssh_root
customize_cmd cust_nobeastie
```

2.4. Cómo actualizar NanoBSD

El proceso de actualización de NanoBSD es relativamente simple:

1. Generar una nueva imagen de NanoBSD de la forma habitual.

2. Copiar la nueva imagen sobre la partición no utilizada del sistema empotrado que está usando NanoBSD.

   La diferencia más importante entre este paso y la instalación inicial de NanoBSD es que ahora en lugar de usar el fichero `_.disk.full` (que contiene una imagen de todo el disco) se instala la imagen `_.disk.image` (que contiene una imagen de una sola partición).

3. Reiniciar y arrancar el sistema desde la partición que se acaba de instalar.

4. Si todo va bien la actualización ha terminado.

5. Si algo va mal reinicie desde la partición anterior (que contiene la imagen vieja, pero que funciona) para recuperar el funcionamiento del sistema lo antes posible. Arregle los problemas de la nueva generación y repita el proceso.

Para facilitar la instalación de una imagen nueva en su sistema NanoBSD en funcionamiento dispone de los scripts `updatep1` y `updatep2`, ubicados en el directorio `/root`. La elección de uno u otro script depende de qué partición esté ejecutando el sistema, la primera o la segunda.

Dependiendo de los servicios disponibles en la máquina que va a servir la nueva imagen de NanoBSD y del tipo de transferencia de datos que prefiera uno de estos tres métodos le resultará más o menos interesante:

2.4.1. Uso de ftp(1)

Si la prioridad es la velocidad de la transferencia use usted este ejemplo:

```
# ftp mi-maquina
get _.disk.image "| sh updatep1"
```

2.4.2. Uso de ssh(1)

Si la prioridad es la seguridad de la transferencia seguramente ejecutará algo muy parecido a lo siguiente:

```
# ssh mi-maquina cat _.disk.image.gz | zcat | sh updatep1
```

2.4.3. Uso de nc(1)

Si la máquina remota no dispone de servicios de ftp(1) ni de sshd(8) puede recurrir al siguiente ejemplo:

1. Primero abra un puerto TCP de escucha en la máquina que sirve la imagen y envíela al cliente:

   ```
   mi-maquina# nc -l 2222 < _.disk.image
   ```

   Nota

   Asegúrese de que el puerto elegido no tenga bloqueadas las conexiones entrantes desde la máquina NanoBSD por ningún cortafuegos.

2. Conéctese a la máquina que va servir la nueva imagen y ejecute el script `updatep1`:

```
# nc mi-maquina 2222 | sh updatep1
```

Índice

N

NanoBSD, 1