

Introducción a NanoBSD

Daniel Gerzo

Revisión: [47547](#)

Copyright © 2006 The FreeBSD Documentation Project

FreeBSD es una marca registrada de FreeBSD Foundation.

Muchas de las designaciones usadas por fabricantes y vendedores para vender sus productos son reclamadas como marcas. En aquellos lugares donde dichas designaciones aparecen en este documento y el proyecto FreeBSD estaba al tanto de los reclamos de marcas, dichas designaciones están acompañadas de los símbolos “™” o “®”.

2015-10-13 por wblock.

Resumen

Este documento provee información acerca de las herramientas NanoBSD, las cuales pueden ser usadas para crear imágenes del sistema de FreeBSD para aplicaciones embebidas, adecuadas para ser usadas en una tarjeta Compact Flash (o cualquier otro medio de almacenamiento masivo).

Tabla de contenidos

1. Introducción a NanoBSD	1
2. NanoBSD Paso a Paso	2
Índice	6

1. Introducción a NanoBSD

NanoBSD es una herramienta actualmente desarrollada por Poul-Henning Kamp <phk@FreeBSD.org>. Crea una imagen del sistema de FreeBSD para aplicaciones embebidas, adecuada para ser usada en una tarjeta Compact Flash (o cualquier otro medio de almacenamiento masivo).

Puede usarse para crear imágenes de instalación especializadas, diseñadas para instalar y mantener sistemas comúnmente llamados “aparatos de computación”. Los aparatos de computación incluyen el hardware y software en el mismo producto, lo cual significa que todas las aplicaciones vienen preinstaladas. El aparato se conecta a una red existente y puede comenzar a funcionar (casi) de inmediato.

Las características de NanoBSD incluyen:

- Los ports y packages funcionan de la misma manera que en FreeBSD — Cada aplicación puede ser instalada y usada en una imagen de NanoBSD, de la misma manera que en FreeBSD.
- No hay pérdida de funcionalidad — Si era posible hacer algo con FreeBSD, también será posible hacerlo con NanoBSD, a menos que esa característica específica haya sido removida de la imagen de NanoBSD cuando fue creada.
- Todo es de solo lectura en tiempo de ejecución — Es seguro desconectar el cable. No hay necesidad de ejecutar [fsck\(8\)](#) después de que el sistema se apague de manera brusca.
- Es fácil de crear y personalizar — Haciendo uso de un único script y solamente un archivo de configuración, es posible crear imágenes reducidas y personalizadas para satisfacer un conjunto arbitrario de requisitos.

2. NanoBSD Paso a Paso

2.1. El diseño de NanoBSD

Una vez que la imagen se encuentra presente en el medio es posible arrancar NanoBSD. El medio de almacenamiento masivo está dividido en tres partes por defecto:

- Dos particiones de imagen: `code#1` y `code#2`.
- La partición del archivo de configuración, la cual puede ser montada bajo el directorio `/cfg` en tiempo de ejecución.

Estas particiones normalmente se montan como de solo lectura.

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

La partición del archivo de configuración reside en el directorio `/cfg`. Contiene archivos para el directorio `/etc` y es montada brevemente como una partición de solo lectura tras el arranque del sistema; por consiguiente, es necesaria para copiar archivos modificados desde `/etc` de vuelta hacia el directorio `/cfg` si se espera que los cambios persistan después de reiniciar el sistema.

Ejemplo 1. Hacer Modificaciones Persistentes de `/etc/resolv.conf`

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



Nota

La partición que contiene `/cfg` debería ser montada solo en el arranque y cuando se sustituyan las directivas de los archivos de configuración.

Mantener `/cfg` montado todo el tiempo no es una buena idea, especialmente si el sistema NanoBSD se ejecuta en un medio de almacenamiento masivo que puede ser afectado negativamente por un número alto de escrituras a la partición (como cuando el sincronizador del sistema de archivos vuelca los datos hacia los discos del sistema).

2.2. Compilar una imagen de NanoBSD

Una imagen de NanoBSD se compila usando un simple script de shell `nanobsd.sh`, que se encuentra en el directorio `/usr/src/tools/tools/nanobsd`. Este script crea una imagen, que puede ser copiada al medio de almacenamiento usando la utilidad `dd(1)`.

Los comandos necesarios para compilar 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 Cambie el directorio actual al directorio base del script de compilación de NanoBSD.

- ② Comience el proceso de compilación.
- ③ Cambie el directorio actual por el lugar en donde se encuentren las imágenes compiladas.
- ④ Instale NanoBSD en el medio de almacenamiento.

2.3. Personalizar una imagen de NanoBSD

Esta es probablemente la característica más importante y más interesante de NanoBSD. También es la que le llevará más tiempo cuando se encuentre desarrollando con NanoBSD.

Invocar el siguiente comando forzará a `nanobsd.sh` a leer su configuración desde el archivo `myconf.nano` localizado en el directorio actual:

```
# sh nanobsd.sh -c myconf.nano
```

La personalización puede hacerse de dos maneras:

- Opciones de configuración
- Funciones personalizadas

2.3.1. Opciones de configuración

Con las preferencias de configuración, es posible configurar las opciones pasadas a las etapas `buildworld` y `installworld` del proceso de compilación de NanoBSD, así como también opciones internas pasadas al proceso de compilación principal de NanoBSD. A través de estas opciones es posible achicar el sistema, para que pueda caber en un espacio tan pequeño como 64MB. Pude utilizar estas opciones de configuración para reducir el tamaño de FreeBSD aun más, hasta que consista únicamente del kernel y dos o tres archivos en espacio de usuario.

El archivo de configuración consiste en opciones de configuración que sobrescriben los valores por defecto. Las directivas más importantes son:

- `NANO_NAME` — Nombre de la compilación (utilizada para crear los nombres de los directorios de trabajo).
- `NANO_SRC` . Ruta al archivo fuente utilizada para compilar la imagen.
- `NANO_KERNEL` — Nombre del archivo de configuración del kernel utilizado para compilar el kernel.
- `CONF_BUILD` — Opciones pasadas a la etapa `buildworld` del proceso de compilación.
- `CONF_INSTALL` — Opciones pasadas a la etapa `installworld` del proceso de compilación.
- `CONF_WORLD` — Opciones pasadas a las etapas `buildworld` y `installworld` del proceso de compilación.
- `FlashDevice` — Define el tipo de medio que se usará. Consulte `FlashDevice.sub` para más detalles.

2.3.2. Funciones personalizadas

Es posible configurar cada aspecto de NanoBSD utilizando funciones del shell en el archivo de configuración. El siguiente ejemplo ilustra el modelo básico de funciones personalizadas:

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

Un ejemplo más útil de una función de personalización es el siguiente, el cual cambia el tamaño por defecto 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
```

Estas son algunas funciones de personalización incluidas por defecto y listas para ser usadas:

- `cust_comconsole` — Deshabilita [getty\(8\)](#) en los dispositivos VGA (los nodos de dispositivo `/dev/ttyv*`) y habilita el uso del puerto en serie COM1 como la consola del sistema.
- `cust_allow_ssh_root` — Permite que root se autentique mediante [ssh\(8\)](#).
- `cust_install_files` — Instala archivos desde el directorio `nanobsd/Files`, el cual contiene algunos scripts útiles para la administración del sistema.

2.3.3. Agregando Packages

Pueden agregarse packages a la imagen de NanoBSD utilizando una función personalizada. La siguiente función instalará todos los paquetes situados 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 archivo de Configuración

Un ejemplo completo de un archivo de configuración para crear una imagen personalizada de NanoBSD podría ser:

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

CONF_BUILD='
WITHOUT_KLDLOAD=YES
WITHOUT_NETGRAPH=YES
WITHOUT_PAM=YES
'

CONF_INSTALL='
WITHOUT_ACPI=YES
WITHOUT_BLUETOOTH=YES
WITHOUT_FORTTRAN=YES
WITHOUT_HTML=YES
WITHOUT_LPR=YES
WITHOUT_MAN=YES
WITHOUT_SENDMAIL=YES
WITHOUT_SHAREDOCS=YES
WITHOUT_EXAMPLES=YES
WITHOUT_INSTALLLIB=YES
WITHOUT_CALENDAR=YES
WITHOUT_MISC=YES
WITHOUT_SHARE=YES
'

CONF_WORLD='
WITHOUT_BIND=YES
WITHOUT_MODULES=YES
WITHOUT_KERBEROS=YES
WITHOUT_GAMES=YES
WITHOUT_RESCUE=YES
WITHOUT_LOCALES=YES
'
```

```
WITHOUT_SYSCONS=YES
WITHOUT_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. Actualizar NanoBSD

El proceso de actualización de NanoBSD es relativamente simple:

1. Compile una nueva imagen de NanoBSD, como de costumbre.
2. Suba la nueva imagen a una partición sin usar de un aparato NanoBSD que se encuentre en ejecución.

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

3. Reinicie y arranque el sistema desde la partición recientemente instalada.
4. Si todo terminó correctamente, la actualización habrá finalizado.
5. Si algo sale mal, reinicie en la partición anterior (que contiene la antigua imagen que funciona correctamente), para restaurar la funcionalidad del sistema tan rápido como sea posible. Arregle los problemas de la nueva imagen y repita el proceso.

Para instalar la nueva imagen en el sistema NanoBSD actual, es posible usar los scripts `updatep1` o `updatep2` que se encuentran en el directorio `/root`, dependiendo de la partición desde la que esté ejecutando el sistema actual.

Según que servicios que se encuentren disponibles en el huésped sirviendo la nueva imagen NanoBSD y el tipo de transferencia que se prefiera, es posible utilizar uno de estos tres métodos:

2.4.1. Usando ftp(1)

Si la velocidad de transferencia es una prioridad para usted, utilice este ejemplo:

```
# ftp myhost
get _.disk.image "| sh updatep1"
```

2.4.2. Usando ssh(1)

Si prefiere una transferencia segura, considere usar este ejemplo:

```
# ssh myhost cat _.disk.image.gz | zcat | sh updatep1
```

2.4.3. Usando nc(1)

Puede utilizar este ejemplo si el huésped remoto no está ejecutando los servicios [ftpd\(8\)](#) o [sshd\(8\)](#):

1. En primer lugar, abra un puerto TCP en el huésped que se encuentra sirviendo la imagen y haga que envíe la imagen al cliente:

```
myhost# nc -l 2222 < _.disk.image
```



Nota

Asegúrese de que el puerto usado no esté bloqueado para recibir conexiones entrantes del huésped NanoBSD por el firewall.

2. Conéctese al huésped sirviendo la nueva imagen y ejecute el script `updatep1`:

```
# nc myhost 2222 | sh updatep1
```

Índice

N

NanoBSD, 1