Introduction à NanoBSD

Daniel Gerzo

$FreeBSD: release/8.4.0/fr_FR.ISO8859-1/articles/nanobsd/article.xml 39632 2012-10-01 11:56:00Z gabor $

$FreeBSD: release/8.4.0/fr_FR.ISO8859-1/articles/nanobsd/article.xml 39632 2012-10-01 11:56:00Z gabor $

Ce document fournit des informations à propos des outils **NanoBSD**, qui peuvent être utilisés pour créer des images du système FreeBSD pour des applications embarquées, adaptées pour utiliser comme support une carte Compact Flash (ou tout autre support de stockage).

Version française de Jean-Loic Tignon <`loic@vigilan.net`>.

Table des matières

1. Introduction à NanoBSD 1

2. Comment utiliser NanoBSD 2

1. Introduction à NanoBSD

NanoBSD est un outil actuellement développé par Poul-Henning Kamp <`phk@FreeBSD.org`>. Il permet de créer une image du système FreeBSD pour des applications embarquées, pouvant utiliser une carte Compact Flash comme support (ou un autre support de stockage)

Il peut être utilisé pour créer des images d'installation spécialisées, conçues pour une installation et une maintenance aisées de systèmes communément appelés « appliances ». Les appliances ont leur matériel et leur logiciel intégrés dans le produit, ce qui signifie que toutes les applications sont pré-installées. L'appliance est connectée à un réseau existant et peut entrer en fonction (presque) immédiatement.

Les fonctionnalités de **NanoBSD** incluent:

- Les logiciels portés et les paquetages fonctionnent comme sous FreeBSD — Chaque application peut être installée et utilisée dans une image **NanoBSD**, de la même façon que sous FreeBSD.

- Aucune fonctionnalité ne manque — S'il est possible de faire quelque chose avec FreeBSD, il sera possible de faire la même chose avec **NanoBSD**, sauf si la fonctionnalité spécifique ou des fonctionnalités ont été explicitement supprimées de l'image **NanoBSD** lors de sa création.
- Tout est en lecture seule pendant l'exécution — Débrancher le cordon d'alimentation est sans danger. Il n'est pas nécessaire d'exécuter fsck(8) après un arrêt inopiné du système.
- Facile à créer et à personnaliser — A l'aide d'une seule procédure et d'un fichier de configuration il est possible de créer des images personnalisées et de taille réduite répondant à n'importe quel type de besoin.

2. Comment utiliser NanoBSD

2.1. L'organisation de NanoBSD

Une fois que l'image est présente sur le support, il est possible de démarrer **NanoBSD**. Le périphérique de stockage est divisé en trois parties par défaut:

- Deux partitions image: `code#1` et `code#2`.
- La partition de configuration, qui peut être montée sur le répertoire `/cfg` à l'exécution.

Ces partitions sont normalement montées en lecture seule.

Les répertoires `/etc` et `/var` sont des disques md(4) (malloc).

La partition de configuration est montée sur le répertoire `/cfg`. Elle contient les fichiers du répertoire `/etc` et est brièvement montée en lecture seule juste après le démarrage du système, par conséquent il est nécessaire de recopier les fichiers modifiés de `/etc` vers le répertoire `/cfg` si l'on souhaite que les changements soient encore effectifs après le redémarrage du système.

Exemple 1. Apporter des changements permanents à `/etc/resolv.conf`

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

> **Note :** La partition qui abrite `/cfg` doit être montée uniquement au démarrage et lors de la copie des fichiers de configuration.
>
> Garder `/cfg` monté en permanence n'est pas une bonne idée, en particulier si le système **NanoBSD** tourne sur un périphérique de stockage qui peut être endommagé par un grand nombre d'écritures sur la partition (c'est à dire quand le contenu des tampons de données est transféré sur les disques).

2.2. Créer une image NanoBSD

Une image **NanoBSD** est créée à l'aide d'une simple procédure `nanobsd.sh`, qui peut être trouvée dans le répertoire */usr*/src/tools/tools/nanobsd. Ce programme crée une image, qui peut être copiée sur le support de stockage à l'aide de dd(1).

Les commandes nécessaires à la création d'une image **NanoBSD** sont:

```
# 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 Se placer dans le répertoire de base du programme de création de **NanoBSD**.

❷ Démarrer le processus de création.

❸ Se placer dans le répertoire où les images ont été créees.

❹ Installer **NanoBSD** sur le support de stockage.

2.3. Personnaliser une image NanoBSD

C'est probablement la fonctionnalité la plus importante et la plus intéressante de **NanoBSD**. C'est aussi là où vous passerez le plus de temps lors de vos développements avec **NanoBSD**.

L'invocation de la commande suivante va obliger `nanobsd.sh` à lire sa configuration dans le fichier `myconf.nano` situé dans le répertoire courant:

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

La personnalisation est effectuée de 2 façons:

- à l'aide d'options de configuration
- à l'aide de fonctions de personnalisation

2.3.1. Les options de configuration

Grace aux paramètres de configuration, il est possible de configurer des options qui sont passées aux étapes `buildworld` et `installworld` du processus de compilation de **NanoBSD**, ainsi que des options internes qui sont passées au processus principal de compilation de **NanoBSD**. A l'aide de ces options, il est possible de diminuer la taille du système, de façon à ce qu'il tienne dans 64Mo. Vous pouvez utiliser les options de configuration pour réduire encore plus FreeBSD, jusqu'à ce qu'il ne consiste plus qu'en un noyau et 2 ou 3 fichiers dans le système de base.

Le fichier de configuration consiste en une série d'options de configuration, qui surchargent les valeurs par défaut. Les directives les plus importantes sont:

- `NANO_NAME` — Nom de compilation (utilisé pour créer le nom des répertoires de travail).
- `NANO_SRC` — Chemin de l'arbre des sources utilisé pour construire l'image.
- `NANO_KERNEL` — Nom du fichier de configuration utilisé pour compiler le noyau.

- `CONF_BUILD` — Options passées à la phase `buildworld` de la compilation.
- `CONF_INSTALL` — Options passées à la phase `installworld` de la compilation.
- `CONF_WORLD` — Options passées aux étapes `buildworld` et `installworld`.
- `FlashDevice` — Définit le type de support à utiliser. Consulter le fichier `FlashDevice.sub` pour plus de détails.

2.3.2. Les fonctions de personnalisation

Il est possible d'optimiser **NanoBSD** en utilisant des fonctions d'interpréteur de commandes dans le fichier de configuration. L'exemple suivant présente la trame de base des fonctions de personnalisation:

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

Un exemple plus utile de fonction de personnalisation est le suivant, qui change la taille par défaut du répertoire `/etc` de 5Mo à 30Mo:

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

Il existe par défaut quelques fonctions de personnalisation prédéfinies et prêtes à être utilisées:

- `cust_comconsole` — Désactive getty(8) sur les consoles VGA (les périphériques `/dev/ttyv*`) et paramètre la console système sur le port série COM1.
- `cust_allow_ssh_root` — Autorise l'ouverture de sessions `root` via sshd(8).
- `cust_install_files` — Installe les fichiers du répertoire `nanobsd/Files`, qui contient des programmes utiles pour l'administration système.

2.3.3. Exemple de fichier de configuration

Un exemple complet de fichier de configuration pour créer une image **NanoBSD** personnalisée peut être:

```
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. Mettre à jour NanoBSD

Le processus de mise à jour de **NanoBSD** est relativement simple:

1. Créer une nouvelle image **NanoBSD**, comme d'habitude.
2. Télécharger la nouvelle image dans une partition inutilisée d'une appliance **NanoBSD** opérationnelle.

   La différence la plus importante de cette étape par rapport à l'installation initiale de **NanoBSD** est que maintenant au lieu d'utiliser le fichier `_.disk.full` (qui contient une image de la totalité du disque), l'image `_.disk.image` est installée (qui contient seulement l'image d'une seule partition système).

3. Redémarrer le système sur la nouvelle partition.
4. Si tout s'est bien passé, la mise à jour est terminée.
5. Si quelque chose s'est mal passé, redémarrez de nouveau sur la partition précédente (qui contient l'ancienne image, fonctionnelle celle-ci), pour remettre le système en état de marche le plus rapidement possible. Corriger les problèmes de la nouvelle image, et répéter le processus.

Pour installer la nouvelle image sur le système **NanoBSD** opérationnel, il est possible d'utiliser la procédure `updatep1` ou `updatep2` située dans le répertoire `/root`, en fonction de la partition qui est en cours d'utilisation sur le système.

En fonction des services disponibles sur la machine qui dispose de la nouvelle image **NanoBSD** et du type de transfert qui est préféré, il est possible d'utiliser une de ces trois méthodes:

2.4.1. Utiliser ftp(1)

Si la vitesse de transfert est la priorité, utiliser cet exemple:

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

2.4.2. Utiliser ssh(1)

Si un transfert sécurisé est préférable, considérer l'utilisation de cet exemple:

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

2.4.3. Utiliser nc(1)

Utiliser cet exemple si la machine distante n'utilise ni ftp(1) ni sshd(8):

1. Tout d'abord, ouvrez un écouteur TCP sur la machine qui dispose de l'image et faites-lui envoyer l'image au client:

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

   > **Note :** Assurez vous que le port utilisé n'est pas bloqué par un pare-feu afin de recevoir les connexions entrantes de la machine **NanoBSD**.

2. Se connecter à la machine qui dispose de la nouvelle image et exécuter la procédure `updatep1`:

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