Introdução ao NanoBSD

Resumo

Este documento fornece informações sobre as ferramentas NanoBSD, que podem ser usadas para criar imagens do sistema FreeBSD para aplicativos embarcados, adequadas para uso em um cartão Compact Flash (ou outro meio de armazenamento em massa).

Índice

1. Introdução ao NanoBSD

O NanoBSD é uma ferramenta atualmente desenvolvida por Poul-Henning Kamp phk@FreeBSD.org. Ele cria uma imagem do sistema FreeBSD para aplicativos embarcados, adequada para uso em um cartão Compact Flash (ou outro meio de armazenamento em massa).

Ele pode ser usado para construir imagens de instalação especializadas, projetadas para fácil instalação e manutenção de sistemas comumente chamados de "appliances". Os appliances têm seu hardware e software agrupados no produto, o que significa que todos os aplicativos são pré-instalados. O appliance é conectado a uma rede existente e pode começar a funcionar (quase) imediatamente.

Os recursos do NanoBSD incluem:

- Os Ports e os pacotes funcionam como no FreeBSD - Cada aplicativo pode ser instalado e usado em uma imagem do NanoBSD, da mesma forma que no FreeBSD.
- Nenhuma funcionalidade ausente - Se é possível fazer algo com o FreeBSD, é possível fazer a mesma coisa com o NanoBSD, a menos que o recurso ou os recursos específicos sejam explicitamente removidos do NanoBSD quando a imagem foi criada.
- Todo o sistema opera em modo read-only em tempo de execução - É seguro puxar o plugue de alimentação. Não há necessidade de executar fsck(8) após um desligamento abrupto do sistema.
- É fácil de criar e personalizar - Usando apenas um script de shell e um arquivo de configuração, é possível criar imagens reduzidas e personalizadas, satisfazendo qualquer conjunto arbitrário de requisitos.

2. NanoBSD Howto

2.1. O Design do NanoBSD

Quando a imagem estiver presente na midia, é possível inicializar o NanoBSD. O meio de armazenamento em massa é dividido em três partes por padrão:

- Duas partições de imagem: `code#1` e `code#2`.
- A partição do arquivo de configuração, que pode ser montada no diretório /cfg em tempo de execução.

Essas partições são normalmente montadas em modo read-only (somente leitura).

Os diretórios /etc e /var são discos criados em memória (malloc) pelo comando md(4).

A partição do arquivo de configuração persiste no diretório /cfg. Ele contém arquivos para o diretório /etc e é brevemente montado como read-only logo após a inicialização do sistema, portanto é necessário copiar os arquivos modificados de /etc de volta para o /cfg se as alterações precisarem ser mantidas após a reinicialização do sistema.

Exemplo 1. Fazendo Mudanças Persistentes no /etc/resolv.conf

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

A partição que contém o /cfg deve ser montada somente no momento da inicialização ou quando for preciso sobrescrever os arquivos de configuração.

Manter o /cfg montado o tempo todo não é uma boa ideia, especialmente se o sistema NanoBSD for executado em um meio de armazenamento em massa que pode ser afetado negativamente por um grande número de gravações na partição (como quando o sistema de arquivos sincroniza os dados para os discos do sistema).

2.2. Construindo uma imagem NanoBSD

Uma imagem NanoBSD é construída usando um simples shell script nanobsd.sh, que pode ser encontrado no diretório /usr/src/tools/tools/nanobsd. Este script cria uma imagem, que pode ser copiada no meio de armazenamento usando o utilitário dd(1).

Os comandos necessários para construir uma imagem NanoBSD são:

```
# cd /usr/src/tools/tools/nanobsd ①
# sh nanobsd.sh ②
# cd /usr/obj/nanobsd.full ③
# dd if=_.disk.full of=/dev/da0 bs=64k ④
```

① Altere o diretório atual para o diretório base do script de construção do NanoBSD.

② Comece o processo de construção.

③ Altere o diretório atual para o local onde as imagens construídas estão localizadas.

④ Instale o NanoBSD no meio de armazenamento.

2.3. Personalizando uma imagem NanoBSD

Este é provavelmente o recurso mais importante e interessante do NanoBSD. Este também é o lugar onde você passará a maior parte do tempo desenvolvendo com o NanoBSD.

A execução do seguinte comando forçará o nanobsd.sh a ler sua configuração do myconf.nano localizado no diretório atual:

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

A personalização é feita de duas maneiras:

- Opções de configuração
- Funções personalizadas

2.3.1. Opções de configuração

Com as definições de configuração, é possível configurar as opções passadas tanto para o estágio `buildworld` quando para o `installworld` do processo de construção do NanoBSD, bem como opções internas passadas para o processo principal de construção do NanoBSD. Através destas opções, é possível reduzir o sistema para que ele caiba, por exemplo, em um cartão de memória de 64 MB. Você pode usar as opções de configuração para reduzir ainda mais o FreeBSD, até que ele consista apenas no kernel e em dois ou três arquivos na área de usuário.

O arquivo de configuração consiste em opções de configuração, que substituem os valores padrões. As diretivas mais importantes são:

- `NANO_NAME` - Nome da compilação (usada para construir os nomes do diretório de trabalho).
- `NANO_SRC` - Caminho para o diretório com o código fonte que será utilizado na construção da imagem.
- `NANO_KERNEL` - Nome do arquivo de configuração do kernel usado para construir o kernel.
- `CONF_BUILD` - Opções passadas para o estágio `buildworld` da compilação.
- `CONF_INSTALL` - Opções passadas para o estágio `installworld` da compilação.
- `CONF_WORLD` - Opções passadas para o estágio `buildworld` e o `installworld` da compilação.

- `FlashDevice` - define o tipo de mídia a ser usado. Verifique o FlashDevice.sub para mais detalhes.

2.3.2. Funções Personalizadas

É possível ajustar o NanoBSD usando as funções do shell no arquivo de configuração. O exemplo a seguir ilustra o modelo básico de funções personalizadas:

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

Um exemplo mais útil de uma função de customização é o seguinte, o qual altera o tamanho padrão do diretório /etc de 5MB para 30MB:

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

Existem algumas funções de customização pré-definidas por padrão e prontas para uso:

- `cust_comconsole` — Desabilita o getty(8) nos dispositivos VGA (os device nodes /dev/ttyv*) e habilita o uso do console do sistema na serial COM1.
- `cust_allow_ssh_root` — Permite que o `root` faça o login via sshd(8).
- `cust_install_files` — Instala arquivos do diretório nanobsd/Files, que contém alguns scripts úteis para administração do sistema.

2.3.3. Adicionando Pacotes

Pacotes podem ser adicionados a uma imagem NanoBSD usando uma função customizada. A seguinte função irá instalar todos os pacotes localizados em /usr/src/files/tools/nanobsd/packages:

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

2.3.4. Exemplo do arquivo de configuração

Um exemplo completo de um arquivo de configuração para criar uma imagem NanoBSD personalizada pode 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_FORTRAN=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. Atualizando o NanoBSD

O processo de atualização do NanoBSD é relativamente simples:

1. Crie uma nova imagem NanoBSD, como de costume.
2. Carregue a nova imagem em uma partição não usada de um dispositivo NanoBSD em execução.

   A diferença mais importante deste passo da instalação inicial do NanoBSD é que agora, em vez de usar . *Disk.full* (que contém uma imagem do disco todo), a imagem \.disk.image está instalada (a qual contém uma imagem de uma única partição do sistema).

3. Reinicie e inicie o sistema a partir da partição recém-instalada.
4. Se tudo correr bem, a atualização está concluída.
5. Se algo der errado, reinicie a partição anterior (que contém a imagem antiga que estava em funcionamento) para restaurar a funcionalidade do sistema o mais rápido possível. Corrija quaisquer problemas da nova compilação e repita o processo.

Para instalar uma nova imagem no sistema NanoBSD, é possível usar o script updatep1 ou updatep2 localizado no diretório /root, dependendo de qual partição o sistema atual está executando.

De acordo com os serviços que estiverem disponíveis no servidor que contem a nova imagem NanoBSD e o tipo de transferência preferido, é possível seguir por uma destas três maneiras:

2.4.1. Usando ftp(1)

Se a velocidade de transferência estiver em primeiro lugar, use este exemplo:

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

2.4.2. Usando ssh(1)

Se uma transferência segura for preferida, considere usar este exemplo:

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

2.4.3. Usando nc(1)

Tente este exemplo se o host remoto que contem a nova imagem não estiver executando o serviço ftpd(8) e nem o serviço sshd(8):

1. Primeiramente, abra um socket TCP em modo escuta no host que serve a imagem e envie a imagem para o cliente:

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

Certifique-se de que a porta usada não esteja bloqueada para receber conexões de entrada do host NanoBSD pelo firewall.

2. Conecte-se ao host que está servindo a nova imagem e execute o script updatep1:

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