

FreeBSD 10-CURRENT

Crear una imagen desde las fuentes en SVN

Matthias Apitz

guru@unixarea.de — <http://www.unixarea.de/>

Agenda

1. Introducción
2. Instalar 9.2 y compilar *10-CURRENT* via SVN
3. Crear una imagen en una llave de USB
4. Instalar *los ports* via SVN e Internet
5. Hacer paquetes de los *ports* y su instalación en la llave de USB
6. Preguntas

—

Introducción

Unas pocas palabras sobre FreeBSD

- FreeBSD es un avanzado sistema operativo libre
- arquitecturas soportadas (actualmente): x86 compatibles (como Pentium® y Athlon), amd64 compatibles (como Opteron, Athlon64 EM64T), UltraSPARC®, IA-64, PC-98 y ARM
- FreeBSD es un derivado de BSD, la versión de UNIX® desarrollada en la Universidad de California, Berkeley
- desarrollado y mantenido por un numeroso equipo de personas
- página principal de web: <http://www.freebsd.org/es/>
- Manual de FreeBSD: http://www.freebsd.org/doc/es_ES.ISO8859-1/books/handbook/
- últimas versiones:
 - Release estable para producción: 9.2
 - Próxima Release 10.0 - BETA3
 - 10-CURRENT (sólo por las fuentes desde SVN)

—

Instalar 9.2 y compilar *10-CURRENT* via SVN (1/6)

- FreeBSD 9.2 se puede bajar desde <http://www.freebsd.org/> como imagen de DVD o USB
FreeBSD-9.2-RELEASE-i386-dvd1.iso
FreeBSD-9.2-RELEASE-i386-memstick.img
- instalación normal de FreeBSD:
 1. arrancar desde DVD o USB
 2. repartir el disco dejando unos 30-40 GByte para el sistema de fichero /usr
 - / -- 512 MByte
 - /var -- 2 GByte
 - swap -- 2 GByte
 - /tmp -- 512 MByte
 - /usr -- todo el resto (mínimum 30 GByte)
 3. instalar sólo el sistema operativo, sin fuentes, sin X11
 4. configurar la red, acceso al sistema por SSH, primeros usuarios 'root' y 'guru' (también en el grupo 'wheel' para poder subir con 'su' al nivel de 'root' ... todo esto dura unos 5 minutos
- el resultado será: un sistema 9.2, listo para arrancar y compilar

Instalar 9.2 y compilar *10-CURRENT* via SVN (2/6)

- checkout las fuentes del núcleo y de todo el 'userland' via SVN:

```
# svn checkout svn://svn.freebsd.org/base/head /usr/src
```

más información sobre SVN: <http://wiki.freebsd.org/SubversionPrimer>
- compilar todo (sistema y 'userland') e instalar sólo el núcleo:

```
# cd /usr/src
# echo 'WITH_PKGTOOLS=yes' > /etc/src.conf
# make buildworld
# make buildkernel KERNCONF=GENERIC
```

el paso siguiente sería

```
# make installkernel KERNCONF=GENERIC
# shutdown -r now
...
```

pero no, queríamos instalar en una llave de USB

Instalar 9.2 y compilar 10-CURRENT via SVN (3/6)

- Recomendación: operaciones en USB son lentas, por eso vamos a usar el siguiente truco:
 1. copiamos la llave desde /dev/da0 a un fichero con dd(1);
 2. usamos mdconfig(8) para hacer del fichero un disco en la memoria, y montamos ese disco virtual con mount(8)
 3. hacemos todo el trabajo en el disco virtual
 4. al final vamos a copiar el disco virtual a la llave de USB
- acá van los detalles:

```
# dd if=/dev/da0 of=da0.img bs=8m
# mdconfig -a -t vnode -f da0.img
md0
# dd if=/dev/zero of=/dev/md0 bs=8m # borrar todos los bloques a zero
```

- ahora creamos la partición para FreeBSD en este fichero / disco virtual

```
# fdisk -I md0
# fdisk -B md0
# bsdlablel -w md0s1 auto
# bsdlablel -B md0s1
# bsdlablel -e md0s1 # edit the disk label and change partition "a" from "unuse
```

Instalar 9.2 y compilar 10-CURRENT via SVN (4/6)

- creamos el sistem de fichero

```
# newfs /dev/md0s1a
# mount /dev/md0s1a /mnt
```

- instalamos el sistema compilado antes hacia /mnt:

```
# make installworld DESTDIR=/mnt
# make installkernel DESTDIR=/mnt KERNCONF=GENERIC INSTALL_NODEBUG=t
# make distrib-dirs DESTDIR=/mnt
# make distribution DESTDIR=/mnt
```

- algunos toques para el futuro sistema: /etc/fstab, ...

```
# echo /dev/da0s1a / ufs rw,noatime 1 1 > /mnt/etc/fstab
```

Instalar 9.2 y compilar *10-CURRENT* via SVN (5/6)

- algunos toques para el futuro sistema: /etc/fstab, ...

```
# cat <<EOF > /mnt/etc/rc.conf
tmpmfs="YES"
tmpsize="128m"
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
ifconfig_bge0="DHCP"
hostname=tiny-r255948
sshd_enable="YES"
moused_enable="YES"
EOF
```

- creamos un usuario 'normal' para entrar por SSH más tarde:

```
# chroot /mnt /usr/sbin/adduser
```

—

Instalar 9.2 y compilar *10-CURRENT* via SVN (6/6)

- umount el disco virtual y copiarlo hacia la llave:

```
# umount /mnt
# mdconfig -d -u md0
# dd if=da0.img of=/dev/da0 bs=8m
```

- resultado después de arrancar la llave de USB:

```
# uname -a
FreeBSD tiny-r255948 10.0-ALPHA4 FreeBSD 10.0-ALPHA4 #1: Fri Oct 18 12:10:57 CE
```

—

Instalar *ports* via SVN e Internet (1/3)

Unas palabras sobre *los ports*:

- son esqueletos de unas 30.000 aplicaciones, listas para su compilación en FreeBSD
- por ejemplo el ambiente gráfico KDE4:

```
$ ls -Cl /usr/ports/x11/kde4
Makefile      <--- cómo se compila el port
files/...     <--- acá los parches para FreeBSD
pkg-descr     <--- acá la descripción del port
scripts/...
```

- se instala el árbol de los esqueletos de los *ports* desde el DVD (hay una pregunta para hacerlo durante la instalación del sistema), o mejor por ser más actual via SVN e Internet:

```
# cd /usr
# svn checkout svn://svn.freebsd.org/ports/head
```

Instalar *ports* via SVN e Internet (2/3)

- la compilación de un *port* se hace así:

```
# cd /usr/ports/x11/kde4
# make install
```

(¿cómo funciona esto?, exige unas palabras más)

- las fuentes bajadas de Internet se quedan en `/usr/ports/distfiles/ ...` (miles) para su próximo uso
- se busca por un *port*, por ejemplo por Gnome, con

```
# cd /usr/ports
# make search name=gnome | more
```

y da entre otras informaciones el lugar donde queda, en este caso: `/usr/ports/x11/gnome2`

Instalar *ports* via SVN e Internet (3/3)

El uso práctico de los *ports*: renovar una granja de computadoras; se necesitan:

- una máquina libre para hacerlo (la tengo en forma de una máquina virtual)
- un script para la compilación de los ports que me interesan: [myPorts.sh](#)
- y compilo los *ports* así:

```
$ ssh vm-azul
$ su
# nohup ./myPorts.sh
```

resultado: unas 1200 aplicaciones (y sub-aplicaciones) están instaladas

- y de estas se hace paquetes binarios para su instalación en otras computadoras (isin Internet!)

Hacer paquetes de los *ports* y su instalación en la llave de USB(1/4)

- El proceso "make install" de cada port (y de sus sub-ports de los que depende) crea al final un paquete y lo instala; se lo puede ver con:

```
# pkg_info kde-4.5.10_1
Information for kde-4.5.10_1:
```

```
Comment:
The "meta-port" for KDE
```

```
Description:
KDE is a powerful Free Software graphical desktop environment for Linux
and Unix workstations.
...
```

- Todos los paquetes instalados tienen una entrada en la base de datos, en /var/db/pkg, por ejemplo:

```
# ls -Cl /var/db/pkg/kde-4.5.10_1/
+COMMENT
+CONTENTS
+DESC
+...
```

Hacer paquetes de los *ports* y su instalación en la llave de USB(2/4)

- se puede hacer de los paquetes instalados ficheros para su distribución, así:

```
# mkdir PKGDIR
# cd PKGDIR
# pkg_create -Rnb kde-3.5.10_1
```

resultado:

```
# ls -l kde*
-r--r--r--  1 guru  wheel      4732 27 mar  2009 kde-3.5.10_1.tbz
-r--r--r--  1 guru  wheel  9372072 27 mar  2009 kdeaccessibility-3.5.10_1.tbz
...
```

—

Hacer paquetes de los *ports* y su instalación en la llave de USB(3/4)

- La manera más fácil de crear todos los paquetes que están instalados para su distribución es:

```
# mkdir PKGDIR
# ls /var/db/pkg > listaDeLosPaquetes.txt
# cd PKGDIR
# while read name; do
  pkg_create -Rnb $name
done < ../listaDeLosPaquetes.txt
```

resultado: unos 1200 ficheros para su distribución a otras computadoras y su instalación allá con nada más fácil que:

```
# pkg_add kde-4.5.10_1
```

—

Hacer paquetes de los *ports* y su instalación en la llave de USB(4/4)

- Ahora vamos a instalar los paquetes en nuestra llave de USB:

```
# mount /dev/da0s1a /mnt
# cp -Rp PKGDIR /mnt
# chroot /mnt pkg_add kde-4.5.10_1
# chroot /mnt pkg_add ...
```

—

Preguntas

¡Gracias por la atención! Y suerte con la instalación. ¿Preguntas?