Introduction to FreeBSD Additional Topics

Pre-SANOG VI Workshop

January 10, 2005
Thimphu, Bhutan

Hervey Allen
Brian Candler

Topics

- How FreeBSD boots (more detail)
- Recompiling the FreeBSD kernel
- Kernel loadable modules and hardware support
- Firewalls
- X Window vs. Gnome vs. KDE
- cvs in detail
- FreeBSD file system: UFS
- Logs
- Use of `su`
- crontab
- Lots more commands

How FreeBSD Boots

Initial boot items are in /boot (this resides under "/", or in it's own partition).

boot0:

Copy of MBR is in /boot/boot0. MBR is at start of the boot disk and is 512 bytes in size. If you use lilo, grub, or other MBR then this is not relevant.

boot1/boot2 or Stage 1 and 2:

/boot/boot1 is 512 bytes in size and runs /boot/boot2.

/boot/boot2 is more complex and runs /boot/loader.

How FreeBSD Boots cont.

Stage 3 or /boot/loader:

- Probes for consoles and disk
- Reads in this order:
  - /boot/loader.rc
  - /boot/defaults/loader.conf
  - /boot/loader.conf to override previous
- Kernel and modules are loaded after a 10 second wait for key press. Interactive prompt available.

For more discussion and examples see:
http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/boot-blocks.html

How FreeBSD Boots cont.

The init process:

- After the kernel boots it hands over control to the user process /sbin/init.
- If filesystems look good then init begins reading the resource configuration of the system. These files are read in this order:
  - /etc/defaults/rc.conf
  - /etc/rc.conf (overrides previous)
  - /etc/rc.conf.local (overrides previous)
- Mounts file systems in /etc/fstab

How FreeBSD Boots cont.

The init process cont.:

- Once file systems are mounted then the following starts:
  - Networking services
  - System daemons
  - Locally installed package daemons (/usr/local/etc/rc.d scripts)

Init process and shutdown:

- When `shutdown` is called then init runs the scripts /etc/rc.shutdown.

The FreeBSD Kernel

- You might rebuild a kernel to add hardware support, additional filesystem support, etc.
- Kernel source, if installed, is in /usr/src/sys
  - "If there is not a /usr/src/sys directory on your system, then the kernel source has not been installed. The easiest way to do this is by running /stand/sysinstall as root, choosing Configure, then Distributions, then src, then sys." (FreeBSD Handbook 9.3)
- To rebuild your kernel you use the default configuration file, update settings as needed, then recompile the kernel, installing it in /boot.

Recompiling the FreeBSD Kernel

See FreeBSD Handbook section 9.3

- Config file in `/usr/src/sys/arch/conf`
- Example (old style):
  - `cp GENERIC /root/kernel/MYNEWKERNEL`
  - `ln -s /root/kernel/MYNEWKERNEL`
  - `/usr/sbin/config MYNEWKERNEL`
  - `cd ../compile/MYNEWKERNEL`
  - `make depend, make, make install`

Recompiling the FreeBSD Kernel cont.

Example (new style):

- `cd /usr/src`
- `make buildkernel kernconf=MYNEWKERNEL`
- `make installkernel kernconf=MYNEWKERNEL`

- Kernel installed as /boot/kernel
- Old kernel is in /boot/kernel.old
- If new kernel does not boot, go to boot loader prompt and type:
  - `unload`
  - `boot kernel.old`

Recompiling the FreeBSD Kernel cont.

The kernel config file has many options. For a more complete explanation of the various options see (e.g. on a PC with Intel CPU):

- /usr/src/sys/i386/conf/NOTES

Or look at the FreeBSD Handbook section 9.4 for some more examples.

Kernel and Hardware Support

FreeBSD is moving towards "modularizing" hardware support. That is "drivers" (kernel loadable modules) are loaded at boot time to support your systems' hardware.

Some hardware is still supported by statically loaded software directly in the kernel.

Some hardware use is optimized by setting kernel state using the sysctl facility.

Kernel Loadable & Static Modules

- Static (in conf) – built-in during recompile vs.
- Kernel loadable (kld) /boot/kernel modules.
- Autoloading using /etc/rc.conf directives and/or using loader.conf
- Address security in FreeBSD vs. Linux and modules.
- Commands `kldload, kldstat, kldunload`

Firewalls

Building an appropriate firewall ruleset for your situation requires thought:

- See FreeBSD Handbook section 10.8 to get started.
- Enable IP FireWall support (IPFW) by adding one, or more options to kernel configuration file.
- ipfw was updated to "ipfw2" in July 2002.
- Starting and stopping in /etc/rc.conf and /etc/rc.firewall.
- ipfw rules and firewall set are in /etc/rc.firewall.
- You can dynamically control ipfw as well:
  - `ipfw flush, ipfw enable, ipfw disable, ipfw flush, etc.`

Installing a Binary File

This is much less common, but you can precompile a program for a specific version of FreeBSD.

Clearly this would be something that might be done with commercial applications that have restrictive licensing agreements.

Normally installation is done using a shell script that copies compressed files to the appropriate locations and updates configurations as needed.

Adobe's Acrobat Reader, Macromedia Flash plugin, etc. are examples (/usr/local/bin/acroread).

Installing with CVS

CVS: Concurrent Versions System

Somewhat detailed FreeBSD Handbook entry:

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/cvsup.html

- Typical use for CVS and FreeBSD (other than software projects) is to keep your Ports collection up-to-date.
- To do this be sure you have installed the Ports collection at initial installation.
- Now install cvsup-without-gui from source if necessary:

Install cvsup

If you are using KDE or Gnome, then check:

```
pkg_info | grep cvs
```

If CVS is installed you can skip this. Otherwise:

```
cd /usr/ports/net/cvsup-without-gui
```

or

```
cd /usr/ports/net/cvsup
make
make install
make clean
```

Install cvsup cont.

Now copy the cvsup configuration file needed to tell CVS to upgrade your ports collection. A sample is located in /usr/share/examples:

```
cp /usr/share/examples/cvsup/ports-supfile /root/.
```

Edit this file to look like this (line 50):

```
# IMPORTANT: Change the next line to use one of the CVSup mirror sites
# listed at http://www.freebsd.org/doc/handbook/mirrors.html.
*default host=cvsup.th.FreeBSD.org
*default base=/usr
*default prefix=/usr
*default release=cvs tag=.
*default delete use-rel-suffix
```

Install and Use cvsup

At this point you are ready to update your entire Ports collection with one simple command:

```
cvsup -g -L 2 /root/ports-supfile
```

"`-g`" : don't use graphical interface.

"`-L 2`" : verbosity level. Level 2 is verbose.

CVS Summary

CVS is a powerful and complex tool. For some more hints and information see:

```
man cvsup
info cvs
```

FreeBSD Handbook:

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports-using.html

X Windows – Gnome – KDE

The first thing to understand is that Gnome and KDE use the X graphical subsystem. Generally KDE programs run in Gnome and vice-versa.

For a server you do not need to run, or install, any of these.

You can run one, both, or other window managers like fwvm, windowmaker, etc.

X – Gnome – KDE cont.

- Which desktop environment is better? There's no correct answer to this.
- To configure how X runs you specify this in the file /etc/X11/XF86Config.
- You can configure everything using menu-based tools, but if you understand how things work you can use edit /etc/X11/XF86Config directly. (xf86cfg may be required in FreeBSD).
- To exit X you can press ALT-CTRL-Backspace.
- You can, also, go directly to a terminal using alt-ctrl-f2 through f8. alt-ctrl-f9 returns to X. Alt-ctrl-f1 displays X informational messages.

The FreeBSD Unix File System

Taken from Wikipedia:

UNIX file system (UFS) is a file system used by many unix operating systems. It is derived from the Berkeley Fast File System (FFS), which itself was originally developed from FS in the first versions of UNIX developed at Bell Labs.

Nearly all BSD unix derivatives including FreeBSD, NetBSD, OpenBSD, NeXTStep, and Solaris use a variant of UFS. In Mac OS X it is available as an alternative to HFS. In Linux, partial UFS support is available and the native linux ext2 filesystem is derived from UFS.

FreeBSD UFS cont.

UFS2 and Soft Updates make for a powerful combination:

- Data is clustered on cylinders to reduce fragmentation.
- Block level fragmentation to avoid wasting disk space when large block sizes are used.
- Extended attribute support.
- Support for 1TB file systems.
- Fast file system creation using “lazy” inode initizializtion.
- Soft updates to dramatically improve metadata operations.
- UFS is journaled so no need for fsck on large drives.

FreeBSD UFS cont.

To learn more about UFS and Soft Updates:

UFS Definition from Wikipedia:

http://en.wikipedia.org/wiki/UFS

Little UFS2 FAQ:

http://lists.freebsd.org/pipermail/freebsd-current/2003-April/001444.html

Disk Tuning (Soft Udpates):

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/configtuning-disk.html#SOFT-UPDATES

Inode Definition from Wikipedia:

http://en.wikipedia.org/wiki/Inode

devfs

DEVice **F**ile **S**ystem:

Basically a way to interact with new devices at the kernel level in the global file system namespace. DEVFS allows the sytem to adapt to hardware changes more cleanly.

- USB, firewire, etc. device mounts are cleaner.
- Included by default in FreeBSD 5.0 and above.
- No longer need to use makedev to create device nodes for new hardware.

Installing FreeBSD (5.2.1)

Sample install session...

```
Boot from CD-ROM
Pick default FreeBSD install
Choose Express install option
Say OK to geometry warning
Delete any slices laying around
Use "A" (entire disk) for FreeBSD slice
Q to finish disk partition
Install FreeBSD BootMgr
Create partitions like this:
    - / 1GB
    - /var 1GB with SoftUpdate on
    - swap 1GB
    - /usr (rest of disk with Softupdates on)
Q to finish partition creation
Choose 5, or X-Developer for install
Choose to install Ports
Click Exit twice to get to media dialogue
Choose to install from CD/DVD
Say "Yes" to last chance to set options
    - set root password
    - add users
    - configure addtion network interfaces
        - configure dc0
            - no dhcp
            - host = int/labnn
            - domain = workshop.th
            - ipv4 gw = 203.159.31.1
            - nameserver = 203.159.0.1
            - ipv4 address =
203.159.31.nnn
            - netmask (calc) =
255.255.255.0
    - configure your mouse, turn on
mouse daemon
    - configure xfree86
        - monitor at 1280x1024 @ 74 Hz
        - define screen size
    - save /etc/X11/XF86Config
    - Configure Gnome
        - Get from CD/DVD
    - Add package bash
        - install packages
            - shells
    - Add package sudo (under security)
    - Set timezone
remove CD-ROM
reboot
```

Installing FreeBSD cont.

First pick the type of install:

- Standard
- Express
- Custom

During install you must partition and slice.

After install use Configure to:

- Install Distribtutions
- Packages
- Configure network, accounts, XFree86, Timezone, mouse, startup, etc.

More Commands

ps	ProceSs list. Show information for running processes
cat	ConCATenate a file to the default ouput (screen)
less	Display file pausing each page & allowing movement
more	Display file pausing each page, but no movement
tail	Display the end of a file (see "-f" option)
gzip	Compress file(s) using Lempel-Ziv coding
gunzip	Decompress zip'ped files
bunzip2	Uncompress files compressed with bzip2
tar	Manipulate Tape ARchive files.
grep	Search text/files for patterns (many variations)

Even More Commands

- apropos
- bg
- bzip2
- chgrp*
- chmod
- clear
- chown*
- "ctrl-u"
- date
- exec
- df
- dmesg
- du
- export
- file
- find
- gcc
- hexdump
- history
- id
- ifconfig*
- info
- init*
- kill
- ln
- locate
- lsof**
- mkdir
- "|" pipe
- man
- mkisofs
- mount*
- netstat
- nmap**
- ping
- pkg_add
- pkg_delete
- pkg_info
- printenv
- ps
- pwd
- reset
- route*
- rmdir
- script
- set
- su
- sysinstall
- sysctl
- swapinfo
- tcpdump
- top
- touch
- traceroute
- uname
- unset
- unzip
- users
- watch
- whereis
- which
- whoami

*root only for changes **Not installed by default in FreeBSD

Basic vi Commands

Impress your friends...

- **Open:** vi fn, vi -r fn, vi + fn, vi +n fn, vi +/pat fn
- **Close:** :w, w!:, :wq, :wq!, :q, :q!
- **Movement:** h,j,k,l w, W, b, B, :n (+arrow keys)
- **Edit:** A, i, o, x, D, dd, yy, p
- **Search:** /pattern, ?pattern, n, N

/etc/group

Format is:

```
wheel:*:0:root,hervey,test
```

-Group name. 8 characters or less.
-Encrypted password. Rarely used. "*" as placeholder.
-Group Identifying number (GID).
-List of group members seperated by commas.
-User's login shell.

Using the `su` Command

The "`su`" command is used to become a different userid, like root, without having to logout and log back in.

To use "`su`" to become root your userid has to be given permission to do this in "/etc/sudoers".

You can allow users to run specific privileged commands using "/etc/sudoers" and "`sudo`".

You can assign users to the "wheel" group and using "/etc/sudoers" you can allow them to run all commands (or some, but unusual).

Use "visudo" as root to allow users or groups to use `sudo`. Users in the "wheel" group can run `su`.

More Uses for the su Command

Instead of having to open a root shell, you can run a privileged command like this:

```
sudo command
```

For example:

```
sudo mount /mnt/cdrom
```

And, if you wish to open a different user shell and run their login scripts do:

```
su - userid
```

Looking for More Information

Not only can you use commands to find information about your system, but you can look inside several files, and you can use the sysctl facility as well.

Example of files with useful information:

- /etc/motd
- /etc/resolve.conf
- /etc/services
- /etc/X11/XF86Config
- /etc/fstab

More information cont.

If you are used to "/proc" it's possible to compile support for this in to the kernel, but not normally used ("options LINPROCFS" in kernel conf file).

You can look in /boot/kernel for modules available and use "kldstat" to see what's loaded (kernel loadable modules = "kld").

Use "dmesg" to see what is reported during startup, including hardware and addresses.

Use of sysctl, such as:

```
sysctl -a,
sysctl -aN
sysctl kern.maxproc
```

And, see /etc/sysctl.conf

Logs – How to Know What's Up

- To configure what happens to events that are logged by applications using syslog, edit the file /etc/syslog.conf (see "`man syslog.conf`").
- Take a look at the file /var/log/messages. The "`tail`" command is very useful for this.
- To troubleshoot start by typing:

```
tail -f /var/log/messages
```

and in another terminal start and stop the service you are trying to debug.

Logs cont.

There are many log files. For example, if you run a webserver, like apache, all of the webserver logs are likely to be in /var/log/httpd

sendmail uses /var/log/maillog

There are multiple software packages to read and automatically generate reports based on logfiles. See:

- http://nsrc.org/security/index.html#logging

for some examples of available packages.

Mounting Filesystems

- If you want to mount a filesystem not listed in /etc/fstab then you need to use the mount command.
- First, you need to know what entry in the /dev directory describes the device you wish to mount (a cd, floppy, another hard drive, etc.).
- You, also, need to know what type of filesystem.
- For example, mounting a dos formatted floppy:
  - `mount -t msdos /dev/fd0 /mnt/floppy` or
  - `mount_msdosfs /dev/fd0 /mnt/floppy`

And, a USB flash pen drive:

- `mount -t msdos /dev/da0s1 /mnt/usb`

Crontab

- The "cron" service allows you to automatically run programs when you want.
- This is configured in /etc/crontab, and /var/cron/tabs/
- Use the command `crontab` in order to change the files that control how the cron daemon works.

Crontab cont.

- In addition you can specify who may and may not use cronjobs with /var/cron/allow and /var/cron/deny

A cron file that shows how a service is going to run has the following format:

```
Minute Hour Day Month Weekday Command
```

An example:

```
1 4 1 4 * /bin/mail user@dot.com < /home/user/joke
```

Send an email on the first of April.