

Chapter 2

Booting Up and Shutting Down

Booting Up

- ◎ Starting up a computer

- > Load kernel into memory and execute it.

- (1) BIOS load and run the MBR (Master Boot Record)

- (2) MBR searches for the bootable slice (partition) on the disk and then run the code on the slice to load OS.

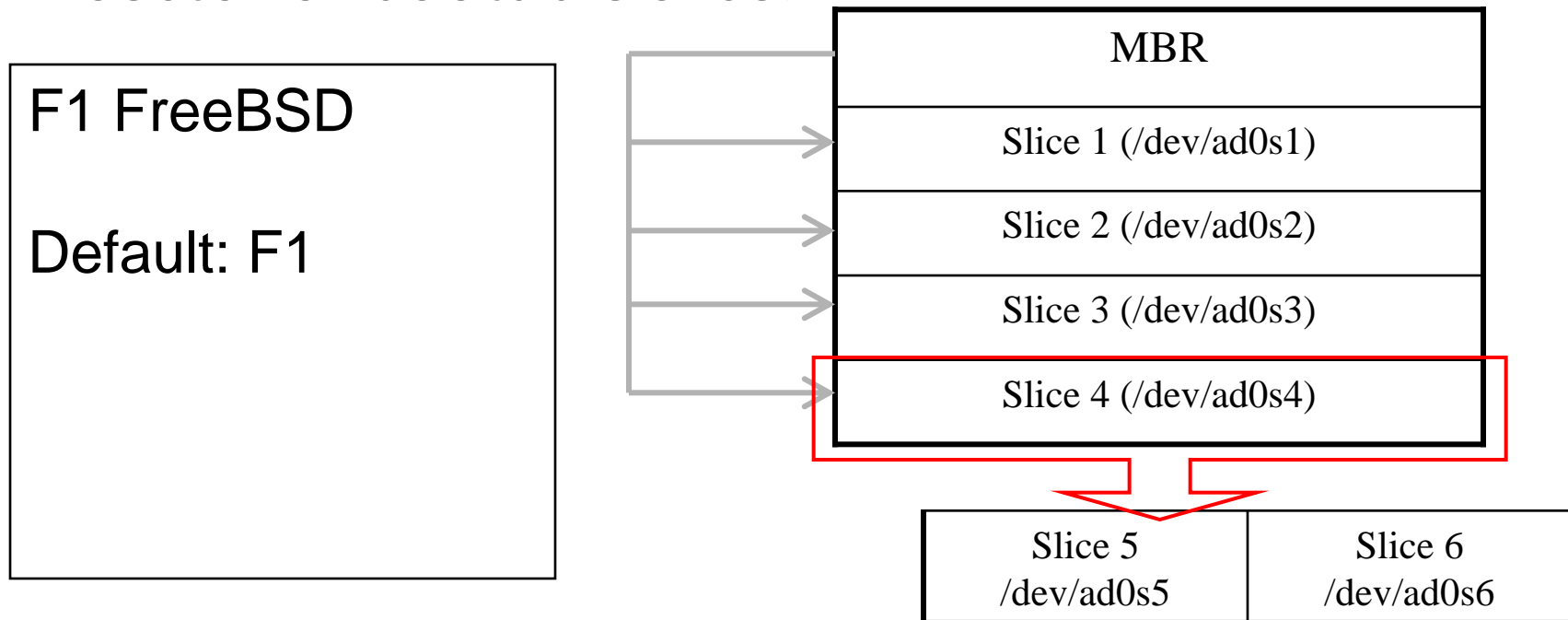
- (3) kernel is loaded into memory, and then probing, initialization, init process.

- ◎ FreeBSD Architecture Handbook

- ◎ <http://www.freebsd.org/doc/en/books/arch-handbook/>

MBR: Master Boot Record

- First 512 Bytes of booting media, last 2 Bytes are 0xAA55
 - > Corresponding copy in FreeBSD is /boot/boot0
- Responsible to find the boot code on the boot sector of bootable slice.

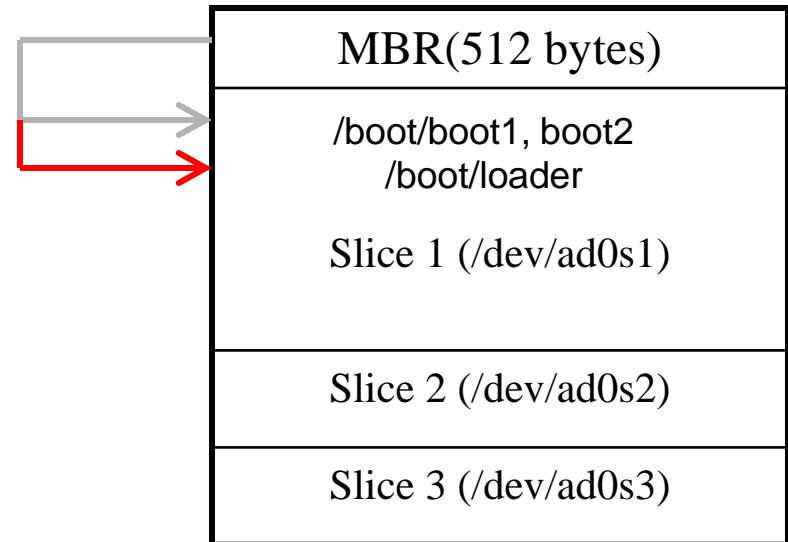


Boot Stages one and two

◎ boot1 and boot2

- > Members of booting chain
- > Used to run the loader.
- > As MBR, boot1 and boot2 are outside the FreeBSD, and the copy of these two are

- /boot/boot1
- /boot/boot2



Boot Stage Three

◎ Boot Stage Three: The loader

- > Provide a user-friendly interface to configure booting choice.
- > /boot/loader
 - /etc/loader.rc use processing commands in /etc/loader.4th to manipulate loader.conf
 - Wait for 10 seconds then autoboot

```
/boot/default/loader.conf
```

Default loader behavior

```
/boot/loader.conf:  
autoboot_delay="10"  
password="ooxx"
```

User-defined loader behavior

MBR recover

- ⦿ If MBR is overwritten by MS (or others), and you want to replace it with FreeBSD MBR:
 - > Boot with CD or Floppy
 - > `$ fdisk -B -b /boot/boot0 ad0`
 - or
 - > `$ boot0cfg -B /dev/ad0`
- ⦿ If you want to replace it with MS MBR
 - > Boot with DOS floppy
 - > `C:\fdisk /mbr`

-B means reinitialize the boot code contained
in sector 0 of the disk

-b is used to specify the boot code

Boot in single user mode

OS	Command
FreeBSD	Interrupt the boot loader and type "boot -s" Or type "4" in the menu (5.x~)
Linux	LILO: linux single
Solaris	Press "STOP" and "a" to enter the boot PROM and Press "boot -s"

Insecure single user mode

- When the physical security to the console is considerable,
 - > Set console to be insecure in /etc/ttys

```
# name  getty          type  status  comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
# console none          unknown off secure
console none          unknown off insecure
```


Multibooting (1)

◎ FreeBSD

- > FreeBSD's boot loader will try to detect bootable partitions
- > You can also declare the bootable partitions explicitly with `boot0cfg`
 - `% boot0cfg -B -m 0x7 ad0`

-m means mask

Specify slices to be enabled/disabled,
ex. `0x7` means `0111`, boot menu will detect
slice1~3 to show the options

Multibooting (2)

- ◎ Linux
 - > Using lilo or GRUB

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=Linux2_4_18
```

```
image=/boot/vmlinuz-2.2.17-14
    label=Linux2_2_17
    root=/dev/hda7
```

```
image=/boot/bzImage
    label=Linux2_4_18
    root=/dev/hda7
```

Steps in the boot process

- ⦿ Loading and initialization of the kernel
- ⦿ Device detection and configuration
- ⦿ Creation of spontaneous system processes
- ⦿ Operator intervention
- ⦿ Execution of system startup scripts
- ⦿ Multiuser operation

Steps in the boot process – Kernel initialization

- ⦿ Get kernel image into memory to be executed
- ⦿ Perform memory test
 - > Allocate kernel's internal data structures

OS	Kernel image path
FreeBSD	/boot/kernel/kernel
Linux	/boot/vmlinuz
Solaris	/kernel/genunix
SunOS	/vmunix

Steps in the boot process – Hardware configuration

- ◎ Devices specified in kernel configuration file
 - > Kernel will try to locate and initialize it
- ◎ Devices not specified in kernel configuration file
 - > Kernel tries to determine the other information by probing the bus
 - If the driver is missing or not responsible to the probe, device is disabled
 - > We can load kernel module to support this device.
 - kldload, kldstat, kldunload
 - See also: kld(3)

Steps in the boot process – System Processes

◎ Spontaneous process

- > Not created by the normal UNIX fork mechanism

OS	Pid 0	Pid 1	Pid 2 and more
FreeBSD	kernel	init	[g_event] ...
Linux	-	init	kflushed,kupdate Kpiod,kswapd
Solaris	sched	init	various handlers
SunOS	swapper	init	Pagedaemon

Steps in the boot process – Operator intervention

◎ Manual boot only (boot into single)

- > Only the root partition is mounted and mounted as read only
 - `mount -u /`
 - `mount -a -t ufs`
 - `swapon -a`

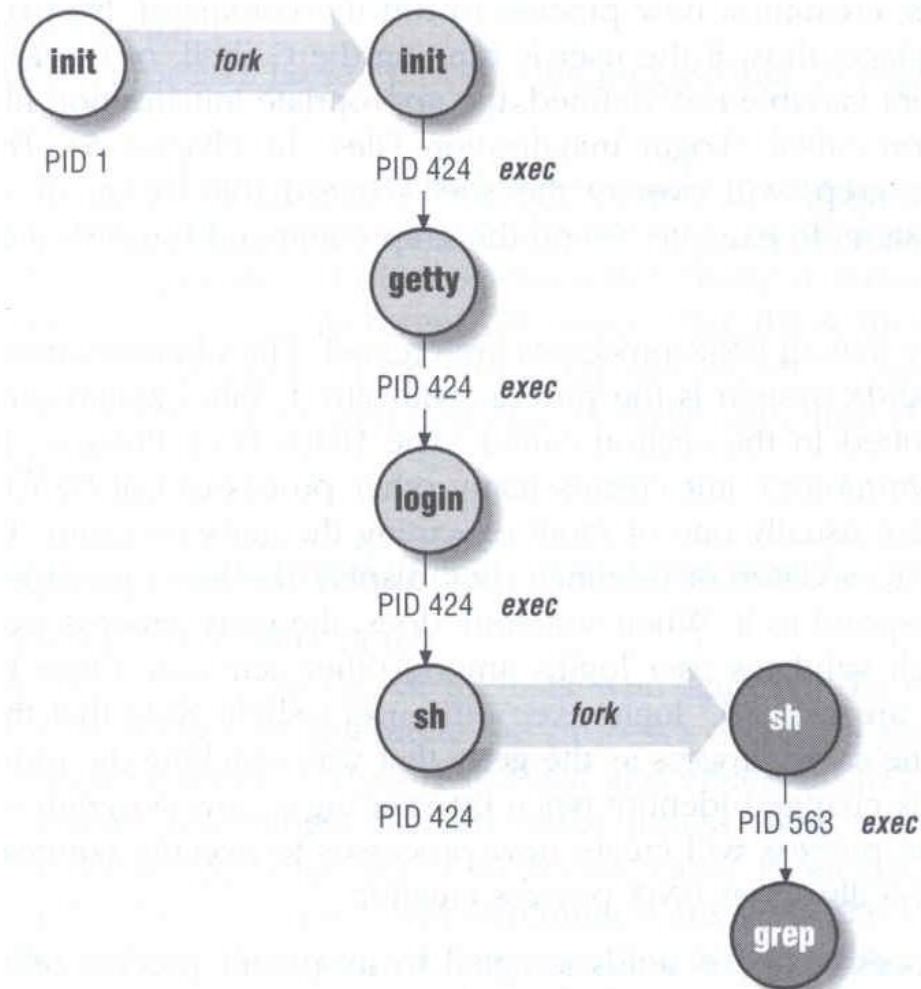
`mount -u` indicates that the status of an already mounted file system should be changed
`mount -a -t` means mount all ufs file systems

Steps in the boot process – Execution of startup scripts

- ◎ The startup scripts are selected and run by **init**
- ◎ Typical works are:
 - > Setting the name of the computer
 - > Setting the time zone
 - > Checking the disk with fsck
 - > Mounting the system's disks
 - > Removing files from /tmp directory
 - > Configuring network interface
 - > Starting up daemons and network services

Steps in the boot process – multiuser operator

- From now on, the system is fully operational, but no one can login
 - init will spawn getty processes to listen for login



Startup Scripts

SystemV-style startup scripts

- > sun, linux
- > /etc/init.d/ ←
- > /etc/rc.d/rcn.d/
- > Each script is responsible for one daemon or one aspect of system.

Symbolic link

Example: sshd in Sun OS

```
case "$1" in
'start')
    if [ -x /usr/local/sbin/sshd ]; then
        echo "Starting the secure shell daemon"
        /usr/local/sbin/sshd &
    fi
    ;;

'stop')
    echo "Stopping the secure shell daemon "
    pkill -TERM sshd
    ;;

*)
    echo "Usage: /etc/init.d/sshd { start | stop }"
    ;;
esac
exit 0
```

Startup Scripts –

SystemV-style startup scripts (1)

- ◎ Run-level

- > /etc/inittab

- > init follow the inittab from level 0 to level k

Example: inittab in sun1

Run Level	Startup scripts	Meaning
0	/etc/rc0.d/	Halt
1	/etc/rc1.d/	Single User Mode
2	/etc/rc2.d/	Multiuser without NFS
3	/etc/rc3.d/	Full multiuser mode
4	/etc/rc4.d/	Unused
5	/etc/rc5.d/	X11
6	/etc/rc6.d/	reboot

Startup Scripts – SystemV-style startup scripts (2)

- ◎ */etc/rcn.d/*
 - > When init transitions from lower run level to higher one,
 - it runs all the scripts that start with "S" in ascending order with "start" argument
 - > When init transitions from high run level to lower one,
 - it runs all the scripts that start with "K" in descending order with "stop" argument

Startup Scripts – SystemV-style startup scripts (3)

- ◎ If you write a daemon and want init to start/stop it,
 - > write a script and put in `/etc/init.d`
 - > make suitable symbolic link in `rcn.d`
 - **`ln -s /etc/init.d/initiald /etc/rc2.d/S61initiald`**
 - **`ln -s /etc/init.d/initiald /etc/rc0.d/K33initiald`**

FreeBSD startup scripts

- ◎ No concept of run level
 - > init will run /etc/rc
 - > /etc/rc will reads the following configuration
 - /etc/defaults/rc.conf
 - /etc/rc.conf
 - /etc/rc.d/
 - > Manual -> rc(8)

Ways to shut down or reboot

- ⦿ Turning off the power ← Please Don't !
- ⦿ Using the shutdown command
 - > Using the halt and reboot command
 - halt = shutdown -h
 - reboot = shutdown -r
- ⦿ Sending init a TERM signal
 - > kill -TERM 1
 - > Using telinit to change init's level
 - > Killing init

Ways to shut down or reboot – shutdown command

OS	Pathname	Time	R	H	S	F
FreeBSD	/sbin/shutdown	time	-r	-h		
Linux	/sbin/shutdown	time	-r	-h		
Solaris	/usr/sbin/shutdown	- <u>gsecs</u>	-i6	-i0	-is	
SunOS	/usr/sbin/shutdown	+mins	-r	-h		-f

R=Reboot, H=Halt, S=Enter Single user mode, F=Skip fsck

time format can be

+m

hh:mm → linux

yymmddhhmm → FreeBSD

Ways to shut down or reboot – telinit

- ◎ Only for SystemV systems
 - > Linux, Solaris
 - \$ telinit 1

Poweroff ?

- In linux,
 - > You can use “poweroff” to shutdown the system and turn the power off.
- In FreeBSD,
 - (1) Compile this into kernel
 - device apm0 at nexus?flag 0x20
 - (2) Rebuild the kernel
 - (3) Edit /etc/rc.conf
 - apm_enable=“YES”
 - apmd_enable=“YES”
 - (4) Reboot
 - (5) Try “shtudown -p now”