

Booting Up and Shutting Down

lctseng

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.

❑ MBR

- http://en.wikipedia.org/wiki/Master_boot_record

❑ FreeBSD Handbook

- <http://www.freebsd.org/doc/en/books/handbook/boot.html>

MBR – Master Boot Record

- ❑ First 512 bytes of disk, outside the FreeBSD area, last 2 Bytes are 0x55AA
 - Corresponding copy in FreeBSD is `/boot/boot0` or `/boot/mbr`

```
nctucs [~] -lctseng- ls -l /boot/boot0
-r--r--r--  1 root  Wheel  512 Nov 12  2014 /boot/boot0
nctucs [~] -lctseng- ls -l /boot/mbr
-r--r--r--  1 root  Wheel  512 Nov 12  2014 /boot/mbr
```

```
nctucs [~] -lctseng- xxd /boot/mbr
00000000: fc31 c08e c08e d88e d0bc 007c be1a 7cbf  .1.....!..
00000010: 1a06 b9e6 01f3 a4e9 008a 31f6 bbbe 07b1  .....1....
...
...
...
000001d0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  .....U.
```

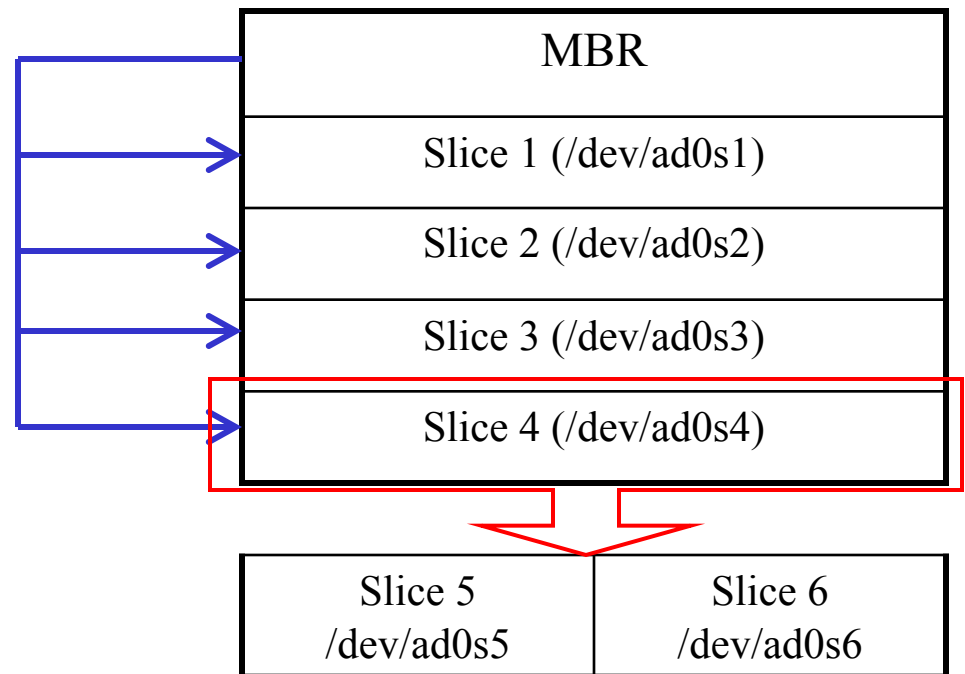
MBR – Master Boot Record

- ❑ Responsible to find the boot code on the boot sector of bootable slice.

Fig. boot0 Screenshot

F1 Win
F2 FreeBSD

Default: F2



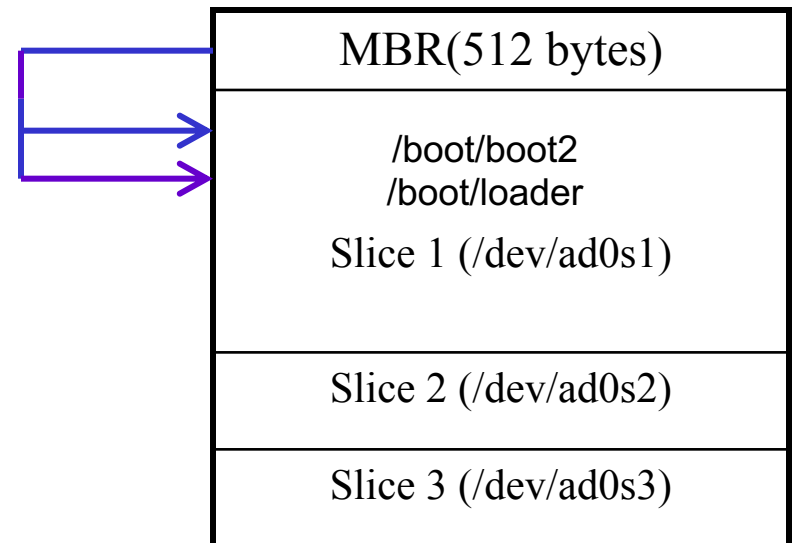
Boot Stage One and Stage Two

□ boot1 and boot2 ($/boot/boot1 + /boot/boot2 = /boot/boot$)

- 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`

Fig. boot2 Screenshot

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```



Boot Stage Three

❑ Boot Stage Three: The loader

- Provide a user-friendly interface to configure booting choice.
- /boot/loader
 - /boot/loader.rc use processing commands in /boot/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

Files in /boot/

❑ /boot/mbr (Standard)

- Simplified version of boot0, blindly boot the partition marked active

❑ /boot/boot0 (BootMgr)

- bootmanager

❑ /boot/boot{1,2}

- boot1 is very simple, since it can only be 512 bytes in size, and knows just enough about the FreeBSD **bsdlablel**, which stores information about the slice, to find and execute boot2.
- boot2 is slightly more sophisticated, and **understands the FreeBSD file system enough to find files on it**, and can provide a simple interface to choose the kernel or loader to run /boot/loader

❑ /boot/loader

- load the kernel from disk

❑ /boot/kernel/kernel

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 "2" in the menu
Linux	LILO: linux single
Solaris	Press "STOP" and "a" to enter the boot PROM and Press "boot -s"

Insecure single user mode

- ❑ Single user mode requires **no password** by default
- ❑ 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

default 0

timeout 30

fallback 1

For booting GNU/Linux

title GNU/Linux

kernel (hd1,0)/vmlinuz root=/dev/hdb1

For booting FreeBSD

title FreeBSD

root (hd0,2,a)

kernel /boot/loader

For booting Windows NT or Windows95

title Windows NT / Windows 95 boot menu

root (hd0,0)

makeactive

chainloader +1

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`
 - `/boot/kernel/*.ko`

```
/boot/loader.conf  
if_em_load="YES"  
vboxdrv_load="YES"  
vboxnet_enable="YES"
```

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	kthreadd, kflushed, kupdate Kpiod, kswapd
SunOS	sched	init	pageout

Steps in the boot process – Operator intervention

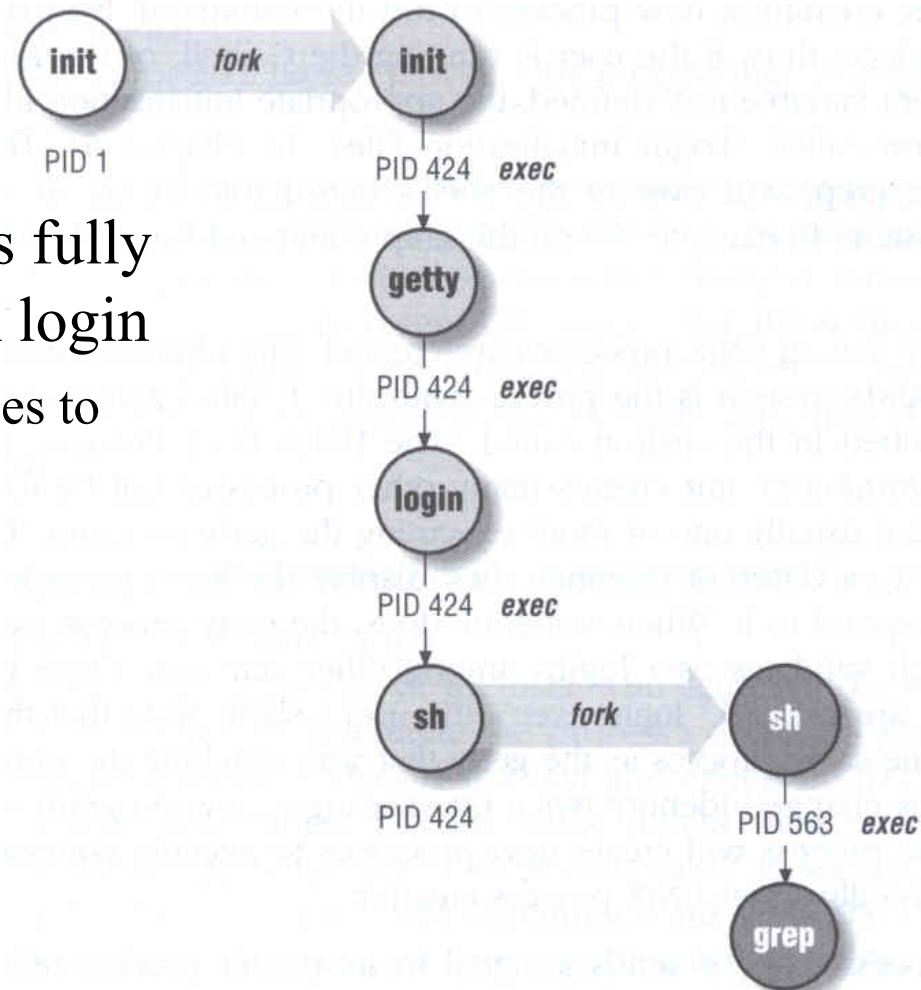
- ❑ Manual boot only (boot into single)
- ❑ Only the root partition is mounted and mounted as read only
 - `mount -u /`
 - Indicates that the status of an already mounted file system should be changed
 - `mount -a -t ufs` (or other external types)
 - Mount all file systems with specific type
 - `swapon -a`
 - Enable all swap

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



FreeBSD startup scripts

- ❑ The BSD-style booting
- ❑ 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

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	-g <u>secs</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

Halt ? Poweroff ?

□ Halt

- Terminate all processes, write data back to disks
- When everything is ready, tell user to turn off the power
 - Or reboot by press any key
- In older systems, you need to manually do so

```
The operating system has halted.  
Please press any key to reboot.
```

It's now safe to turn off
your computer.

您現在可以放心關機

Poweroff

- ❑ Halt + Turn off the power
- ❑ ACPI / APM
 - Advanced Configuration and Power Management
 - Advanced Power Management
- ❑ In FreeBSD,
 - (1) Try “shutdown -p now”
 - (2) Compile this into kernel **In case it does not work...**
device apm0 at nexus?flag 0x20
 - (3) Rebuild the kernel
 - (4) Edit /etc/rc.conf
apm_enable="YES"
apmd_enable="YES"
 - (5) Reboot
 - (6) Try “shtudown -p now” (or poweroff)

Other Booting Manager

- ❑ Besides BSD-style booting, another line is System-V
- ❑ Used by many Linux distributions
 - Solaris, Debian

System-V Startup Scripts

□ 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/rc.d/rc0.d/	Halt
1	/etc/rc.d/rc1.d/	Single user mode
2	/etc/rc.d/rc2.d/	Multiuser without NFS
3	/etc/rc.d/rc3.d/	Full multiuser mode
4	/etc/rc.d/rc4.d/	User defined
5	/etc/rc.d/rc5.d/	Multiuser with graphical interface
6	/etc/rc.d/rc6.d/	Reboot

Ways to shut down or reboot – telinit

- ❑ Only for SystemV systems (and Systemd)
- ❑ telinit: change run-level
- ❑ Halt/poweroff
 - % telinit 0
- ❑ Reboot
 - % telinit 6
- ❑ Single user mode
 - % telinit 1

Systemd

- ❑ Modern system/service manager for many Linux distributions
 - Ubuntu, Debian, ...
- ❑ Evolved from System-V
 - Another booting manager beside BSD
 - Used by older versions of Linux distributions
 - Debian < 8.0 : System-V
 - Debian >= 8.0 : Systemd
- ❑ Similar to System-V, but faster and easier to use

Systemd

❑ Use ‘targets’ to replace run-levels

SysV Run Level	Systemd targets	Meaning
0	runlevel0.target, poweroff.target	Poweroff
1	runlevel1.target, rescue.target	Single user mode
2	runlevel2.target, multi-user.target	User defined. Default: same as level 3
3	runlevel3.target, multi-user.target	Multiuser mode
4	runlevel4.target, multi-user.target	User defined. Default: same as level 3
5	runlevel5.target, graphical.target	Multiuser with graphical interface
6	runlevel6.target, reboot.target	Reboot