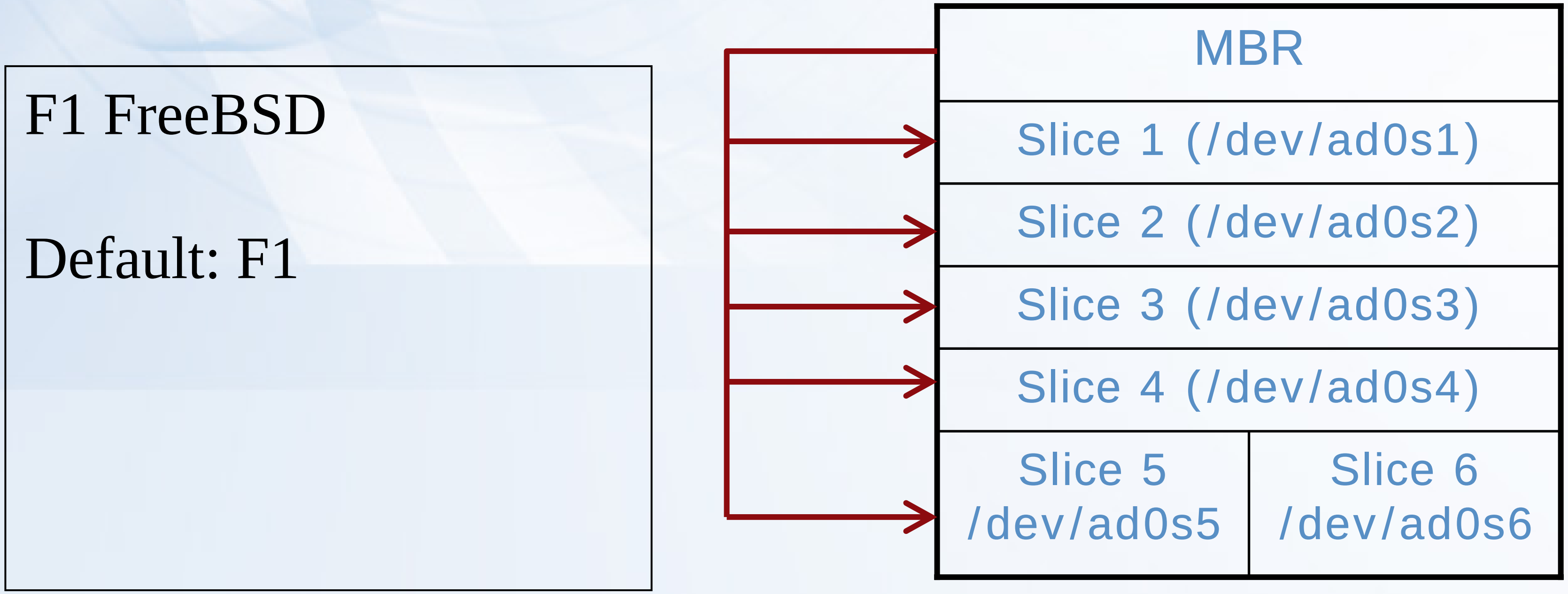
Chapter 2
Booting and Shutting Down

Bootstrapping

- 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 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、Boot Stages one、two、three

- MBR
  - First 512 bytes of disk, outside the FreeBSD area
  - Corresponding copy in FreeBSD is /boot/boot0
  - Responsible to find the boot code on the boot sector of bootable slice.



F1 FreeBSD

Default: F1

MBR

Slice 1 (/dev/ad0s1)

Slice 2 (/dev/ad0s2)

Slice 3 (/dev/ad0s3)

Slice 4 (/dev/ad0s4)

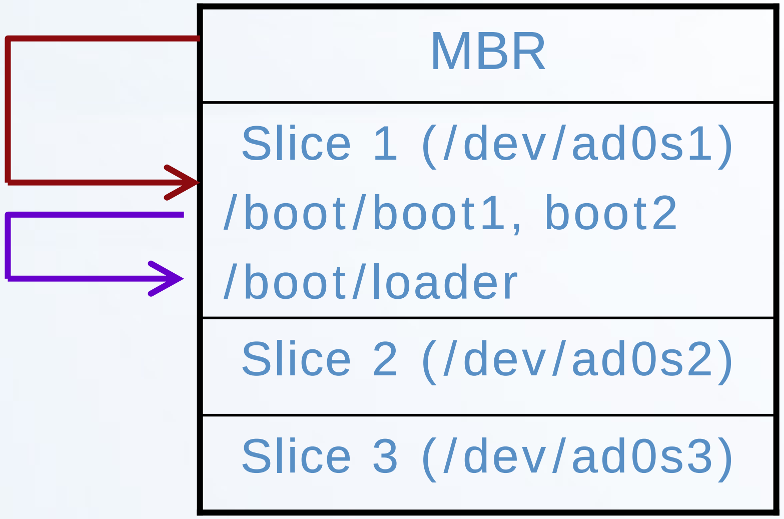
Slice 5 /dev/ad0s5

Slice 6 /dev/ad0s6

Boot Stages one and two

> boot1 and boot2

- Conceptually both are part of the same program, but split into two due to space constrain.
- 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**



MBR

Slice 1 (/dev/ad0s1)
/boot/boot1, boot2
/boot/loader

Slice 2 (/dev/ad0s2)

Slice 3 (/dev/ad0s3)

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, 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”
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

0x7 means 0111 and boot menu will include at least three options

Multibooting (2)

> Linux

- Using lilo or GRUB

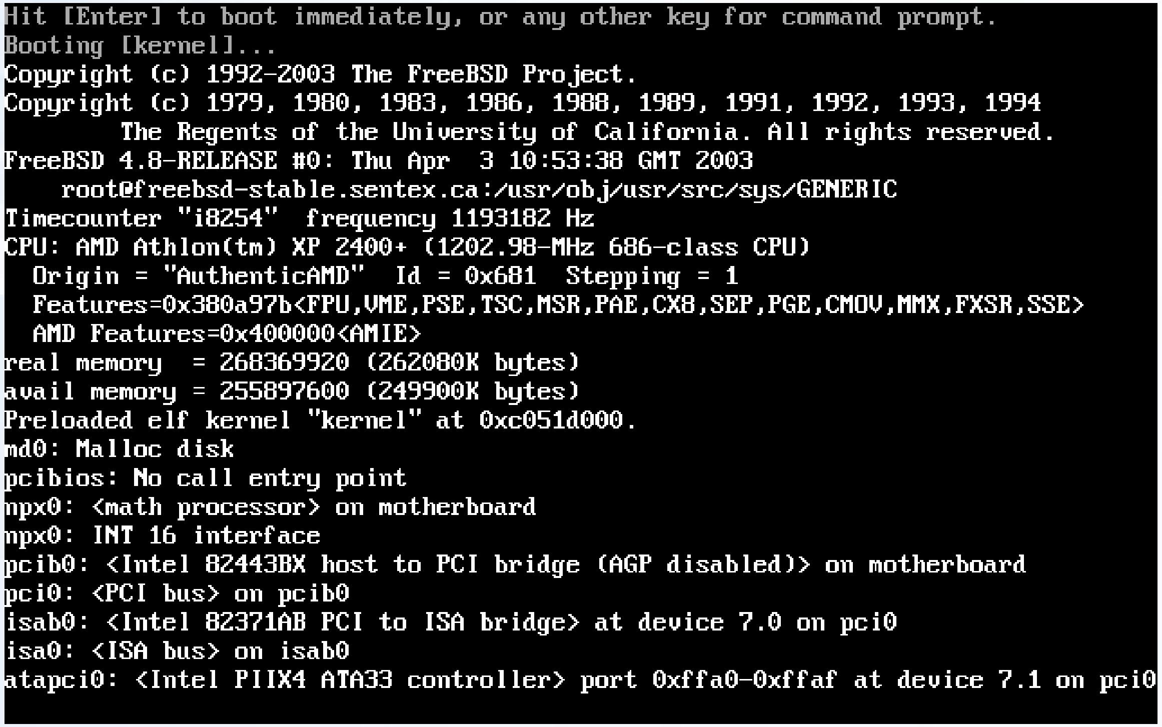
```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=Linux2_4_18

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



```
Hit [Enter] to boot immediately, or any other key for command prompt.
Booting [kernel]...
Copyright (c) 1992-2003 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.
FreeBSD 4.8-RELEASE #0: Thu Apr  3 10:53:38 GMT 2003
    root@freebsd-stable.sentex.ca:/usr/obj/usr/src/sys/GENERIC
Timecounter "i8254"  frequency 1193182 Hz
CPU: AMD Athlon(tm) XP 2400+ (1202.98-MHz 686-class CPU)
  Origin = "AuthenticAMD"  Id = 0x681  Stepping = 1
  Features=0x380a97b<FPU,VME,PSE,TSC,MSR,PAE,CX8,SEP,PGE,CMOV,MMX,FXSR,SSE>
  AMD Features=0x400000<AMIE>
real memory  = 268369920 (262080K bytes)
avail memory = 255897600 (249900K bytes)
Preloaded elf kernel "kernel" at 0xc051d000.
md0: Malloc disk
pcibios: No call entry point
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Intel 82443BX host to PCI bridge (AGP disabled)> on motherboard
pci0: <PCI bus> on pcib0
isab0: <Intel 82371AB PCI to ISA bridge> at device 7.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <Intel PIIX4 ATA33 controller> port 0xffa0-0xffaf at device 7.1 on pci0
```

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	/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
  - 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**

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	swapper	init	pagedaemon
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

- 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

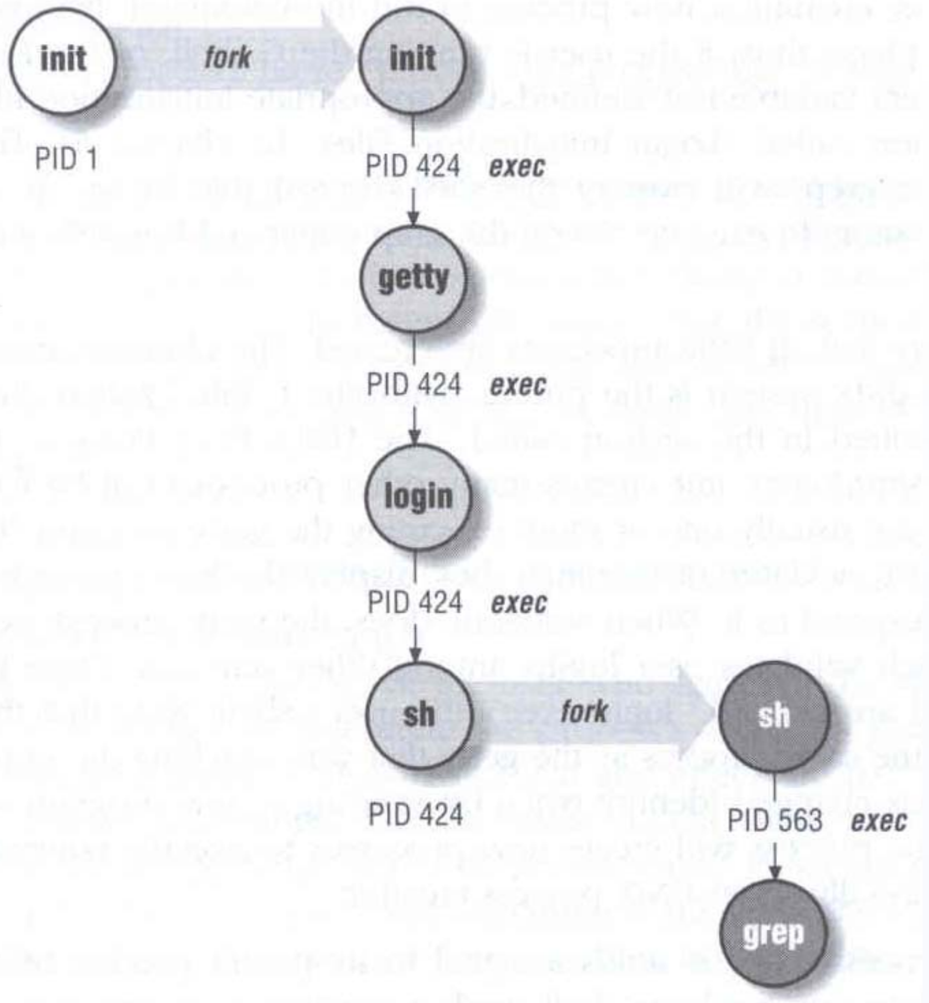
mount –a –t means mount all ufs file systems

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

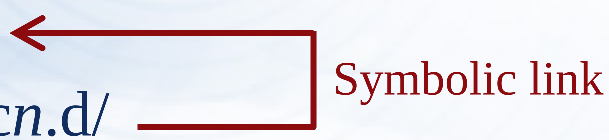
- From now on, the system is fully operational, but no one can login
  - init will spawn getty processes to listen for login



Startup Scripts

> SystemV-style startup scripts

- ccsun1~ccsun7, linux1~linux20
- /etc/init.d/
- /etc/rc.d/rc*n*.d/
- Each script is responsible for one daemon or one aspect of system.



Example: sshd in ccsun2

```
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

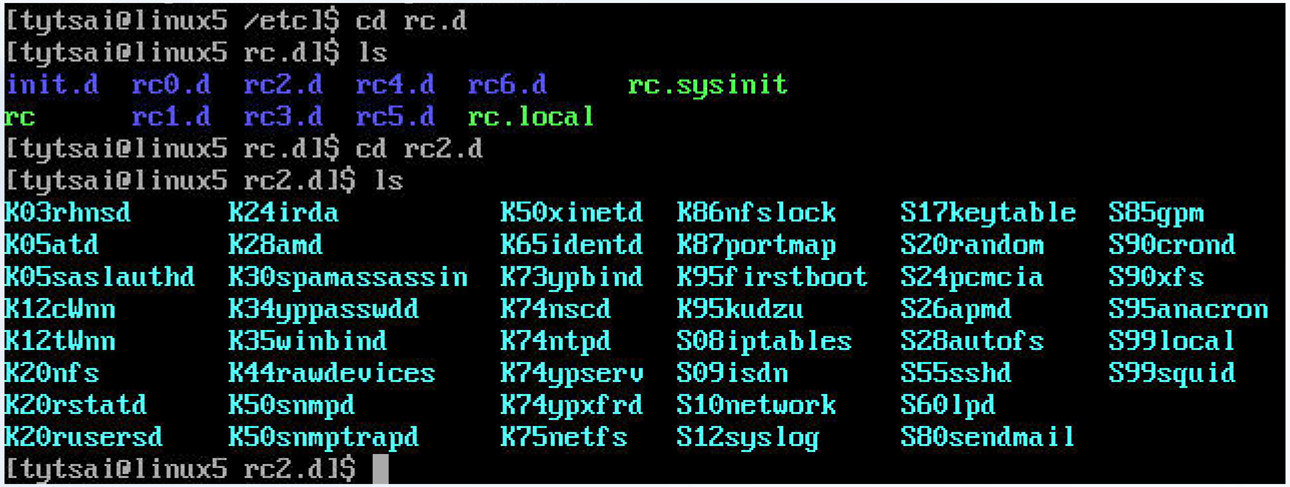
Example: inittab in linux5

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/	Unused
5	/etc/rc.d/rc5.d/	X11
6	/etc/rc.d/rc6.d/	reboot

Startup Scripts – SystemV-style startup scripts (2)

> /etc/rc.d/rc*n*.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**



```
[tytsai@linux5 /etc]$ cd rc.d
[tytsai@linux5 rc.d]$ ls
init.d  rc0.d  rc2.d  rc4.d  rc6.d     rc.sysinit
rc      rc1.d  rc3.d  rc5.d  rc.local
[tytsai@linux5 rc.d]$ cd rc2.d
[tytsai@linux5 rc2.d]$ ls
K03rhnsd      K24irda          K50xinetd  K86nfslock   S17keytable  S85gpm
K05atd        K28amd           K65identd  K87portmap   S20random    S90crond
K05saslauthd  K30spamassassin  K73ypbind  K95firstboot S24pcmcia    S90xfs
K12cWnn       K34yppasswdd     K74nscd    K95kudzu     S26apmd      S95anacron
K12tWnn       K35winbind       K74ntpd    S08iptables  S28autofs    S99local
K20nfs        K44rawdevices    K74ypserv  S09isdn      S55sshd      S99squid
K20rstatd     K50snmpd         K74ypxfrd  S10network   S60lpd
K20rusersd    K50snmptrapd     K75netfs   S12syslog    S80sendmail
[tytsai@linux5 rc2.d]$
```

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 rc*n*.d

- ln –s /etc/init.d/initiald /etc/rc2.d/S61initiald
- ln –s /etc/init.d/initiald /etc/rc0.d/K33initiald

Startup Scripts – SystemV-style startup scripts (4)

- In linux
  - /etc/sysconfig/ contain config data used by startup scripts
  - Ex:
    - **network**
      - Set global network option (hostname, gateway, ..)
        - HOSTNAME=linux5
        - GATEWAY=140.113.209.254
    - **network-scripts/**
      - Contain accessory scripts and network config file
      - EX: ifcfg-eth0
        - DEVICE=eth0
        - BROADCAST=140.113.209.255
        - IPADDR=140.113.209.145
        - NETMASK=255.255.255.0
        - ONBOOT=yes

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.conf.local**
  - /etc/rc then executes the following scripts in predefined order
    - **/etc/rc.diskless1**
    - **/etc/rc.early**
    - **/etc/rc.sysctl**
    - **/etc/rc.serial**
    - **/etc/rc.pccard**
    - **/etc/rc.network**
    - **/etc/rc.network6**
    - **/etc/rc.{arch}**
    - **/etc/rc.syscons**
    - **/etc/rc.local**

Ways to shut down or reboot

- Turning off the power
- 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	-g*secs*	-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”