Booting Up and Shutting Down

lctseng / Liang-Chi Tseng

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 (1)

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

```
[^_^]  > ll mbr
-r--r--r--  1 root  wheel  512 11 12  2014 mbr
18:54 lctseng@lctseng(10.0.2.15)[/boot]
[^_^]  > ll boot0
-r--r--r--  1 root  wheel  512 11 12  2014 boot0
```

```
[^_^]  > cat boot0 | hd
00000000  fc 31 c0 8e c0 8e d8 8e  d0 bc 00 7c 89 e6 bf 00  |.1.........|....|
00000010  06 b9 00 01 f3 a5 89 fd  b1 08 f3 ab fe 45 f2 e9  |.............E..|
00000020  00 8a f6 46 b7 20 74 07  80 4e b7 40 8a 56 b6 88  |...F. t..N.@.V..|
00000030  56 00 52 e8 f7 00 bb c2  07 31 d2 88 6f fc 0f a3  |V.R......1..o...|
```

```
000001c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00000200
```

MBR – Master Boot Record (2)

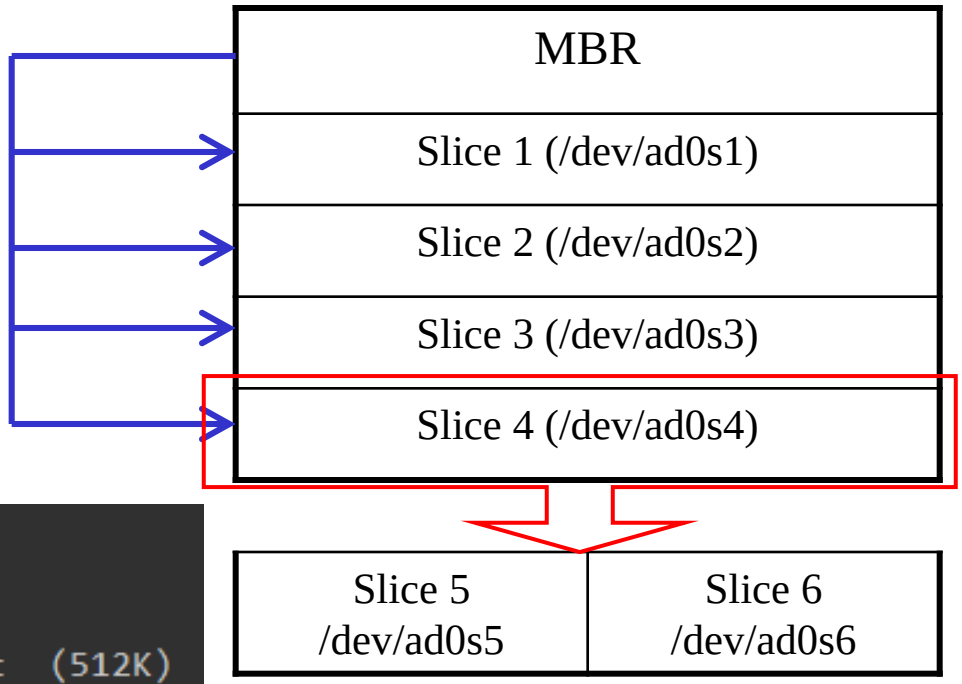
- Responsible to find the boot code on the boot sector of bootable slice.

F1 FreeBSD

Default: F1

```
F1 Win
F2 FreeBSD

Default: F2
```

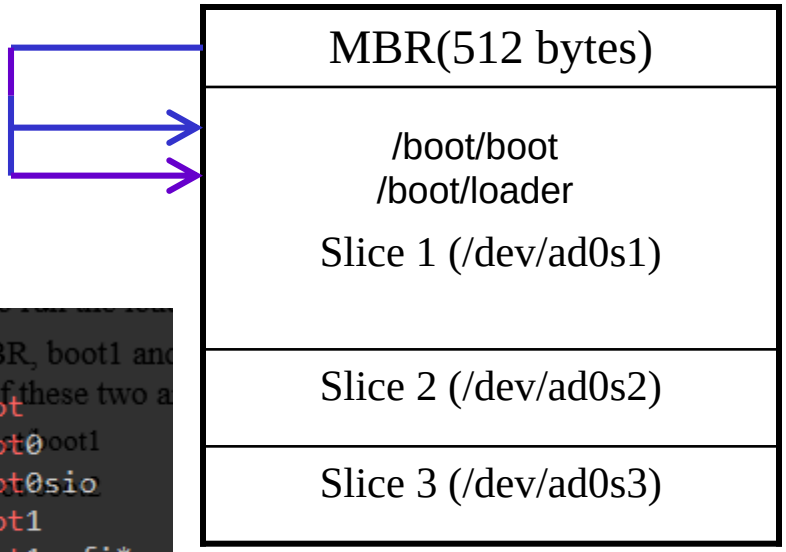



```
18:58 lctseng@lctseng(10.0.2.15)[/boot]
[^_^]  > gpart show
=>       34  104857533  ada0  GPT  (50G)
         34       1024     1  freebsd-boot  (512K)
       1058   98565120     2  freebsd-ufs  (47G)
   98566178    5242880     3  freebsd-swap  (2.5G)
  103809058    1048509        - free -  (512M)
```

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

```
>>FreeBSD/i386 BOOT
Default: 1:ad(1,a)/boot/loader
boot:
```




```
18:58 lctseng@lctseng(10.0.2.15)[/boot]
[^_^]  > ll | grep boot
-r--r--r--   1 root  wheel    8192 11 12  2014 boot
-r--r--r--   1 root  wheel     512 11 12  2014 boot0
-r--r--r--   1 root  wheel     512 11 12  2014 boot0sio
-r--r--r--   1 root  wheel     512 11 12  2014 boot1
-r-xr-xr-x   1 root  wheel   33811 11 12  2014 boot1.efi*
-r--r--r--   1 root  wheel  819200 11 12  2014 boot1.efifat
-r--r--r--   1 root  wheel    7680 11 12  2014 boot2
```

Boot Stage Two

- boot2

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

Boot Stage Three

- loader


```
To get back to the menu, type `menu' and press ENTER
or type `boot' and press ENTER to start FreeBSD.



Type '?' for a list of commands, 'help' for more detailed help.
OK
```

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 bsdlabel, which stores information about the slice, to find and execute boot2. /boot/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

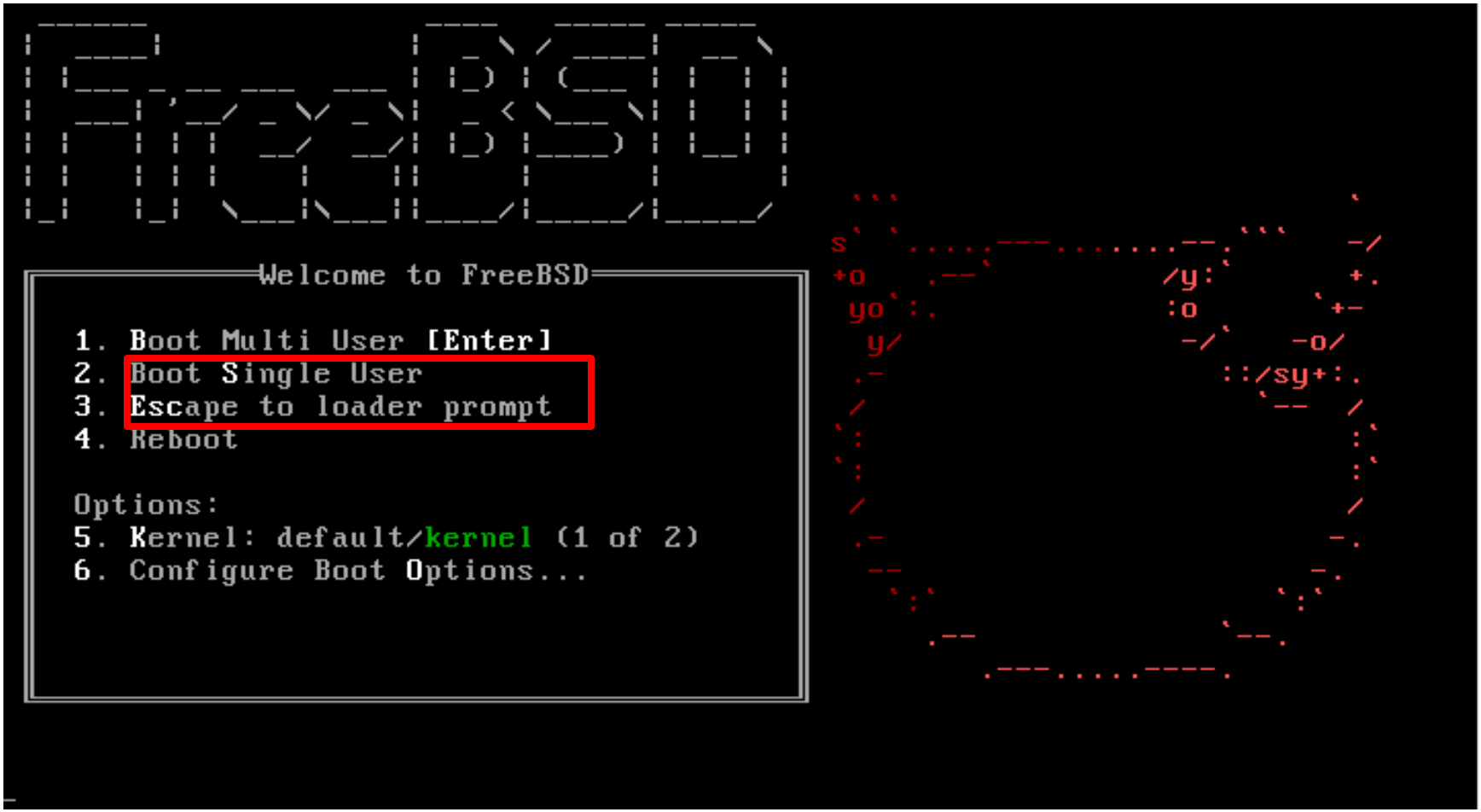
-B means reinitialize the boot code contained in sector 0 of the disk

-b is used to specify the boot code

Boot in single user mode (1)

OS	command
FreeBSD	Enter OK mode 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"

Boot in single user mode (2)



Boot in single user mode (3)

- The OK mode (loader prompt)

```
_
To get back to the menu, type `menu' and press ENTER
or type `boot' and press ENTER to start FreeBSD.



Type '?' for a list of commands, 'help' for more detailed help.
OK _
```

- Type “boot -s” to enter single user mode

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/module/*.ko

/boot/loader.conf

```
if_em_load="YES"
vboxdrv_load="YES"
```

Steps in the boot process – System Processes

- Spontaneous process
  - Not created by the normal UNIX fork mechanism
  - View by `ps [pid]`

OS	Pid 0	Pid 1	Pid 2 and more
FreeBSD	kernel	init	g_event, cam, …
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 /
    - mount –a –t ufs
    - swapon -a

---

mount –u indicates that the status of an already mounted file system should be changed

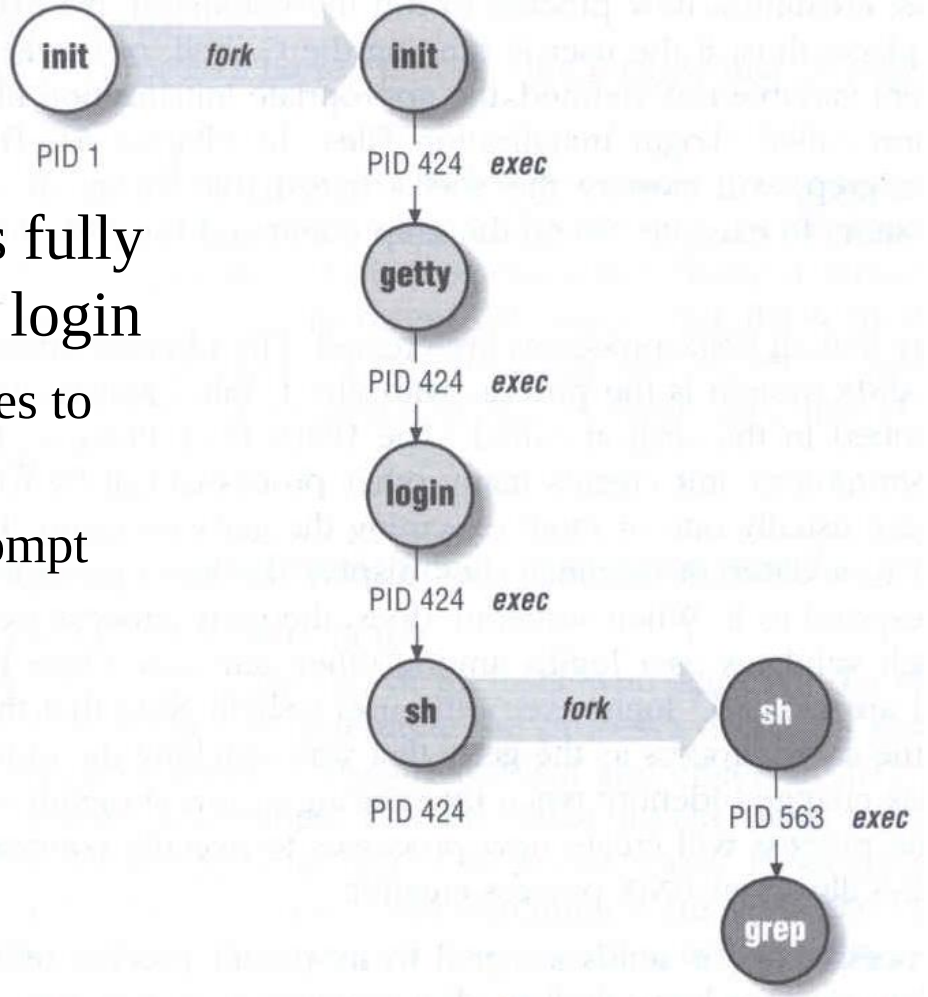
mount –a –t means mount all ufs file systems

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
  - login: read username and prompt for password



FreeBSD startup scripts

- 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 !
  - May cause disk failure / filesystem dirty
- Using the shutdown command
  - Using the halt and reboot command
    - halt = shutdown –h
    - reboot = shutdown –r
    - poweroff = shutdown -p
- Sending init a TERM signal
  - kill –TERM 1 (go into single user mode)
  - Using telinit to change init's level (for Linux using System-V)

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 (after m minutes)

hh:mm ➔linux

yymmddhhmm/now ➔ FreeBSD

Poweroff

- ACPI / APM
  - Advanced Configuration and Power Management
  - Advanced Power Management
- In FreeBSD, (if cannot poweroff automatically)

  (1) Try “shutdown -p now”

  (2) Compile this into kernel

  device apm0 at nexus?flag 0x20

  (3) Rebuild the kernel

  (4) Edit /etc/rc.conf

  apm_enable=“YES”

  apmd_enable=“YES”

  (5) Reboot

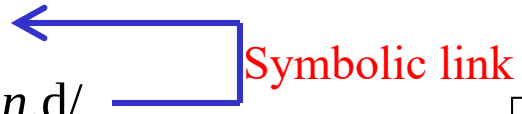
  (6) Try “shtudown –p now”

Appendix

System-V

Startup Scripts

- SystemV-style startup scripts
  - sun, linux
  - /etc/init.d/
  - /etc/rc.d/rc*n*.d/
  - Each script is responsible for one daemon or one aspect of system.



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/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

Ways to shut down or reboot – telinit

- Only for SystemV systems
  - Linux, Solaris
    - % telinit 1