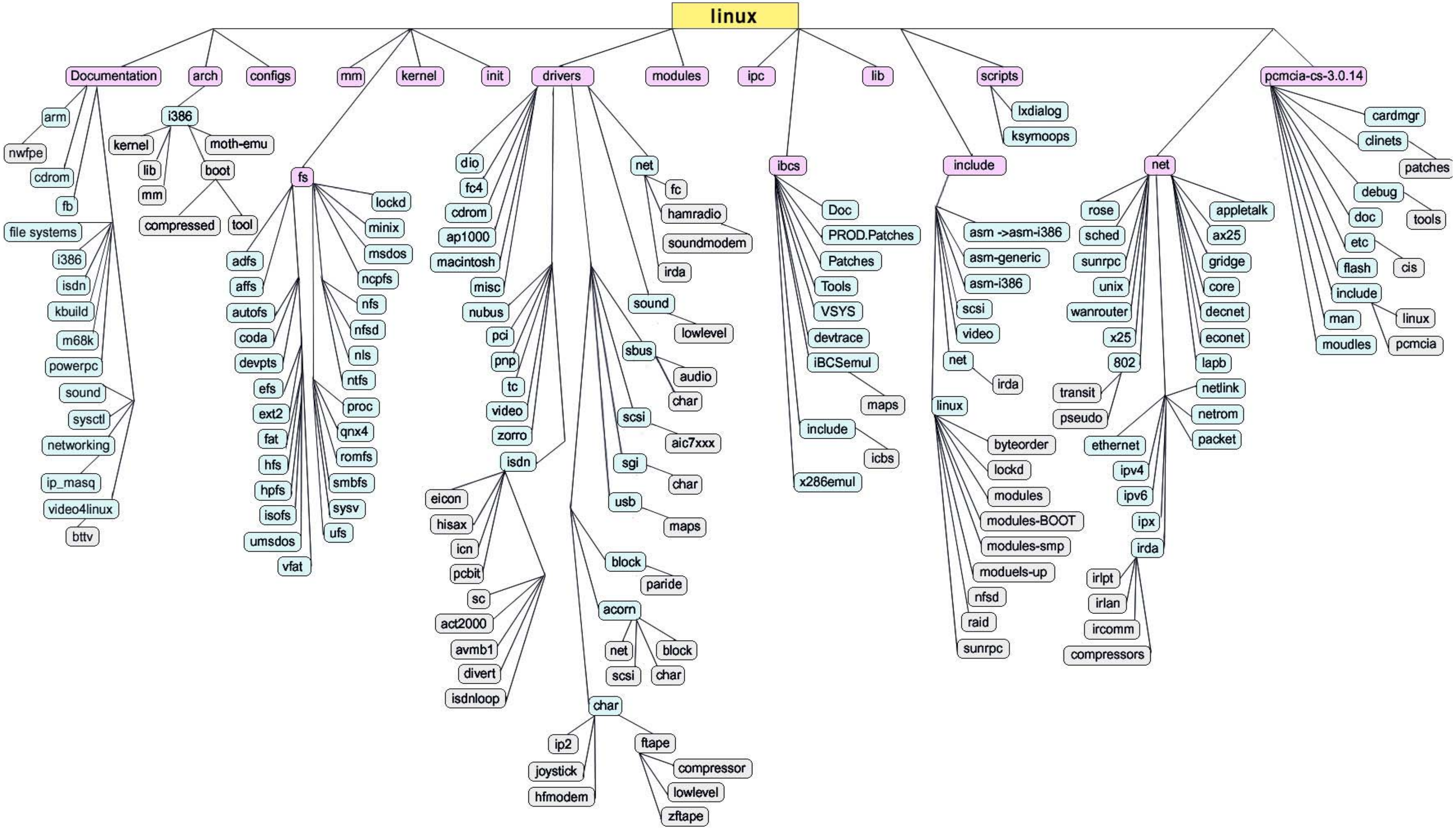
Embedded Systems

Prof. Myung-Eui Lee (A-405)

melee@kut.ac.kr

한국기술교육대학교
korea University of
Technology and Education

Linux File Structure



- linux
  - Documentation
    - arm
      - nwfpe
    - cdrom
    - fb
    - file systems
    - i386
    - isdn
    - kbuild
    - m68k
    - powerpc
    - sound
    - sysctl
    - networking
      - ip_masq
    - video4linux
      - bttv
  - arch
    - i386
      - kernel
      - lib
      - mm
      - boot
        - compressed
        - tool
      - moth-emu
  - configs
  - mm
  - kernel
  - init
  - fs
    - adfs
    - affs
    - autofs
    - coda
    - devpts
    - efs
    - ext2
    - fat
    - hfs
    - hpfs
    - isofs
    - umsdos
    - vfat
    - lockd
    - minix
    - msdos
    - ncpfs
    - nfs
    - nfsd
    - nls
    - ntfs
    - proc
    - qnx4
    - romfs
    - smbfs
    - sysv
    - ufs
  - drivers
    - dio
    - fc4
    - cdrom
    - ap1000
    - macintosh
    - misc
    - nubus
    - pci
    - pnp
    - tc
    - video
    - zorro
    - isdn
      - eicon
      - hisax
      - icn
      - pcbit
      - sc
      - act2000
      - avmb1
      - divert
      - isdnloop
    - net
      - fc
      - hamradio
      - soundmodem
      - irda
    - sound
      - lowlevel
    - sbus
      - audio
      - char
    - scsi
      - aic7xxx
    - sgi
      - char
    - usb
      - maps
    - block
      - paride
    - acorn
      - net
      - scsi
      - block
      - char
    - char
      - ip2
      - joystick
      - hfmodem
      - ftape
        - compressor
        - lowlevel
        - zftape
  - modules
  - ipc
  - ibcs
    - Doc
    - PROD.Patches
    - Patches
    - Tools
    - VSYS
    - devtrace
    - iBCSemul
      - maps
    - include
      - icbs
    - x286emul
  - lib
  - include
    - asm ->asm-i386
    - asm-generic
    - asm-i386
    - scsi
    - video
    - net
      - irda
    - linux
      - byteorder
      - lockd
      - modules
      - modules-BOOT
      - modules-smp
      - moduels-up
      - nfsd
      - raid
      - sunrpc
  - scripts
    - lxdialog
    - ksymoops
  - net
    - rose
    - sched
    - sunrpc
    - unix
    - wanrouter
    - x25
    - 802
      - transit
      - pseudo
    - appletalk
    - ax25
    - gridge
    - core
    - decnet
    - econet
    - lapb
    - netlink
    - netrom
    - packet
    - ethernet
    - ipv4
    - ipv6
    - ipx
    - irda
      - irlpt
      - irlan
      - ircomm
      - compressors
  - pcmcia-cs-3.0.14
    - cardmgr
    - clinets
      - patches
    - debug
      - tools
    - doc
    - etc
      - cis
    - flash
    - include
    - man
      - linux
      - pcmcia
    - moudles

Linux Directories

- **File System Standard (FSSTND)**
  - » Proposed in 1994, every LINUX system should contain a set of standard files and directories
- **Linux Directories**
  - » **root** - The home directory for the root user
  - » **home** - Contains the user's home directories
  - » **bin** – shell command & essential command for both system admin and normal users
    - ◆ all commands needed to boot the system or run it in single-user mode
    - ◆ cp, ls, cat, mv, grep, kill, ps, rm, rpm, vi, etc...
  - » **sbin** – system admin command(not intended for normal users)
    - ◆ fdisk, fsck, halt, shutdown, insmod, mkfs, etc...
    - ◆ Utilities used for system admin are stored in /sbin, /usr/sbin, and /usr/local/sbin

system install program {
- → /sbin : binaries essential for booting, restoring, recovering, and/or repairing the system in addition to the binaries in /bin.
- → /usr/sbin : binaries for graphic or Office program(less important)

user install program
- → /usr/local/sbin : Locally-installed(source compiled) system administration programs

Linux Directories

- **usr** – User application program, most user command
  - → Contains all commands, libraries, man pages, games and static files for normal operation(shareable and read-only data)
  - **bin** - Almost all user commands(gcc, gdb, vim,…), some commands are in /bin or /usr/local/bin.
  - **sbin** - System admin commands not needed on the root filesystem. e.g., most server programs(httpd, pppd,..)
  - **include** - Include files used in C programs such as stdio.h .
  - **lib** - Libraries for programming and packages
  - **local** - The place for locally installed software and other files.
    - → needs to be safe from being overwritten when the system software is updated
  - **man** - Manual pages (실제 파일 /usr/share/man)
  - **share -** Architecture-independent data
    - → **info** - Info documents
    - → **doc** - Documentation
  - **src** – source programs used to build system (linux-2.4.20-8)
  - **X11R6** - The X windows system files.

Linux Directories

- » **etc** - Configuration files specific to the machine.
  - **profile.d** - contains scripts that are run by /etc/profile upon login.
  - **init.d** - contains most of the initialization scripts (crond, apmd,...)
  - **xinetd.d** – Super demon service directory (RH 7.0 later)

    Demon {

  - **rc.d** - contains a number of shell scripts that are run on bootup at different run levels.
    - → /etc/rc.d/init.d – Link to /etc/init.d
    - → /etc/rc.d/rc*.d – Contains script files for services to be started (S) and stopped (K) at that run level (S?? : number means process sequence)
      - » where ``*'' is a number corresponding to the default run level.
      - » Run Level : “ **init *** ” command

        0 Halt the system

        1 Single-user mode (for special administration)

        2 – 5 Normal operation (user defined)

        6 Reboot

        2 – Multiuser (without NFS)
        3 – Full multiuser mode
        4 – unused
        5 – X11

      - » these files are symbolic links to the initialization scripts themselves, which are in /etc/rc.d/init.d.
  - **skel** - When a home directory is created it is initialized with files from this directory(.bash_profile, .bashrc, ...)
  - **sysconfig** - Files that configure the linux system for devices(vi keyboard)

Linux Directories

- **var** - Contains files that change for mail, news, printers log files, man pages, temp files
  - **cache** - Application cache data
  - **lib** - Variable state information
  - **local** - Variable data for programs installed in /usr/local
  - **lock** - Lock files
    - Used by a program to indicate it is using a particular device or file
  - **log** - Log files from programs such as login and syslog
  - **run** - Data relevant to running processes
    - information about the system that is valid until the system is next booted
  - **spool** – Directory for mail/new, and application spool data(printer)
  - **tmp** - Temporary files preserved between system reboots
  - **www** – Web server directory (httpd - index.html)
- **dev** - Device files
- **boot** - Files used by the bootstrap loader (GRUB or LILO). Kernel images are often kept here. the binary of the Linux kernel is in /boot/vmlinux-2.4.20-8smp file.

Linux Directories

- **lib** - Shared libraries needed by the programs on the root file system
  - **modules** - Loadable kernel modules
- **proc** – Kernel and process information(virtual file system), Each process has a directory below proc.
  - **1** - A directory with information about process number 1= /sbin/init
- **mnt** - Mount points for temporary mounts by the system administrator.
- **tmp** - Temporary files. Programs running after bootup should use /var/tmp.

Linux Files

- **Typical Linux Files**
  - **/boot/vmlinuz** - the typical location and name of the Linux kernel(symbolic link to ***vmlinuz-2.4.20-8***)
  - **/dev/fd0** - first FDD, **/dev/fd1** – second FDD
  - **/dev/fd0H1440** - driver for the first floppy drive in high density mode.
  - **/dev/hda** - first IDE HDD, **/dev/sda** – first SCSI/USB HDD
  - **/dev/cdrom** - the IDE cdrom drive. Most often, a symbolic link to the true cdrom driver file (/dev/hdb).
  - **/dev/null** - a virtual-file that can be written to. Data written to this file gets discarded.
  - **/etc/rc.d/rc.sysinit** - run once at boot time
  - **/etc/rc.d/rc** – run rc?.d directory Snn / Knn (K:kill / S:start)
  - **/etc/rc.d/rc.local** - bash script that is executed at the end of login process (executed *after* all the other init scripts: S99local).
  - **/etc/profile** – system-wide environment variables for all users.
  - **/etc/bashrc** - system-wide default functions and aliases for the bash shell.

Linux Files

- **/etc/modules.conf** - aliases and options for configurable modules
- **/etc/crontab** - shell script to run different commands periodically (hourly, daily, weekly, monthly, etc.)
- **/etc/DIR_COLORS** - used to store colors for different file types when using ls command. The ***dircolors*** command uses this file when there is not a .dir_colors file in the user's home directory.
- **/etc/exports** - specifies hosts to which file systems can be exported using NFS. “man exports” contains information on how to set up this file for remote users.
- **/etc/hosts** - contains a list of host names and absolute IP addresses.
- **/etc/resolv.conf** - contains a list of domain name servers
  - /etc/sysconfig/network-scripts/ifcfg-eth0 확인
- **/etc/hosts.allow** - hosts allowed (by the tcpd daemon) to access Internet services
- **/etc/hosts.deny** - hosts forbidden (by the tcpd daemon) to access Internet services
- **/etc/group** - similar to /etc/passwd but for groups

Linux Files

- **/etc/inittab** – linux initialization table

  id:runlevels:action:process **->** ***man inittab***
  - setting the default runlevel
  - running the rc.sysinit script contained in /etc/rc.d
  - running the rc script in l5:5:wait:/etc/rc.d/***rc 5*** → running the script in /etc/rc.d/rc?.d
  - running the last rc.local script in /etc/rc.d → link to S99local
  - setting up virtual login terminals
  - running xdm for a graphical login prompt (only if the default runlevel is 5)
- **/etc/issue** - pre-login message.
- **/etc/motd** - message of the day(motd) file, printed immediately after login.
- **/etc/grub.conf** - configuration file for grub(GRand Unified Bootloader) boot loader
- **/etc/mtab** - shows currently mounted devices and partitions and their status
- **/etc/fstab** - contains information on partitions and file systems used by system to mount different partitions and devices

Linux Files

- **/etc/smb.conf** - config file for the SAMBA server.
- **/etc/passwd** - contains passwords and other information concerning users who are registered to use the system. this is readable only by root and encoded password is in /etc/shadow.
  - ***passwd*** account name
  - melee:x:501:500::/home/melee:/bin/bash
    - → login name : dummy passwd : user ID : group ID: user information : home directory : shell location
- **/etc/printcap** - shows the setup of printers
- **/etc/termcap** - ASCII database defining the capabilities and characteristics of different consoles, terminals, and printers
  - console = physical keyboard and monitor(only one for one computer)
  - terminal = can be physical as well as virtual.
- **/etc/X11/XF86Config** - X configuration file.
- **/proc/cpuinfo** - cpu information
- **/proc/filesystems** - prints filesystems currently in use
- **/proc/interrupts** - prints interrupts currently in use

Linux Files

- /**proc/ioports** - contains a list of the i/o addresses used by various devices connected to the computer
- /**proc/kcore** - The command ***ls -l /proc/kcore*** will give the amount of RAM on the computer. (= " ***free*** " command to get the same information and more).
- /**proc/version** - prints Linux version and other info
- /**var/log/messages** - used by syslog daemon to store kernel boot-time messages
- /**var/log/dmesg** – print the kernel ring buffer(bootup messages)
  - "***dmesg***" command
- /**var/log/lastlog** - used by system to store information about last boot

Dot Files

- **Typical Dot Files**
  - » . **files** = Hidden Files
    - Not visible by default to normal directory-browsing
      - → files named with a leading dot are not normally presented in directory listings ("*ls –a*" command)
    - For this reason, many programs define one or more dot files in which startup or configuration information may be optionally recorded. – gnome, x windows, mozilla,….
  - » **.bash_logout** - file executed by bash shell on logout (=*clear*)
  - » **.bash_history** – bash shell command history
  - » **.bash_profile** - initialization of bash shell run only on login.
    - Bash looks first for a .bash_profile file when started as a login shell
  - » **.bashrc** - initialization command run when bash shell starts up
  - » **.cshrc** - initialization commands that are run automatically when C shell is initiated
  - ❖ **Note** : The process of executing start-up shell scripts

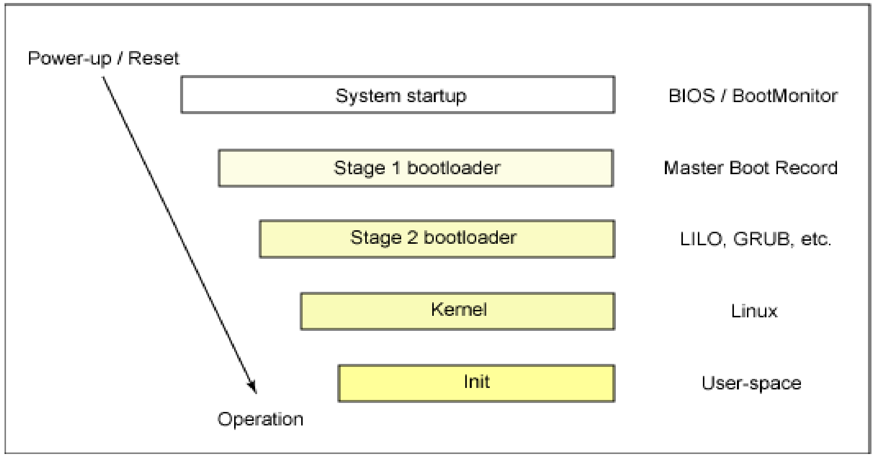
    1. /etc/profile → 2. ~/.bash_profile → 3. /etc/bashrc → 4. ~/.bashrc

Boot Terminology

- **Loader**
  - » Program that moves bits from disk (usually) to memory and then transfers CPU control to the newly loaded" bits (executable).
- **Boot loader / Bootstrap**
  - » The computer term bootstrap began as a metaphor in the 1950s.
  - » In computers, pressing a bootstrap button caused a hardwired program to read a bootstrap program from an input unit and then execute the bootstrap program which read more program instructions and became a self-sustaining process that proceeded without external help from manually entered instructions.
  - » Program that loads the "first program" (the kernel)
  - » LILO, GRUB, NTLDR
- **Boot ROM / ROM Monitor / BIOS**
  - » Persistent code that is "already loaded" on power-up
- **Boot Manager**
  - » Program that lets you choose the "first program" to load

Booting Process

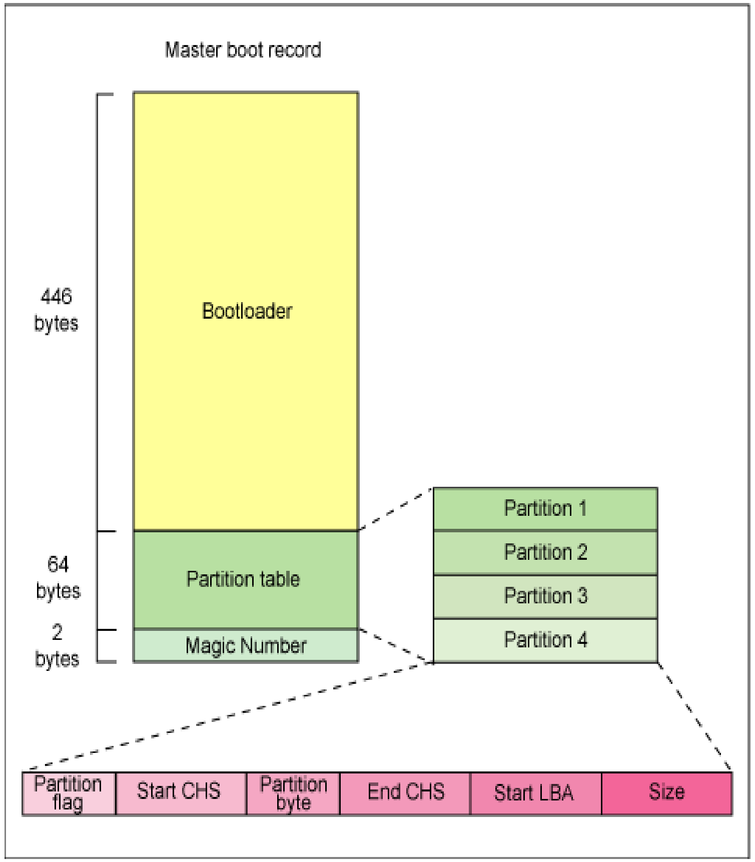
1. Computer Turn On
2. CPU jump to address of BIOS (0xFFFF0)
3. BIOS runs POST (Power-On Self Test)
4. Find bootable devices
5. Loads and execute boot sector form MBR (INT 19 / Stage 1)
6. GRUB loads the kernel into memory and passes control on to the kernel (Kernel loader / Stage 2)
7. Kernel running and /sbin/init start → /etc/inittab script start
8. /etc/inittab → /etc/rc.d/rc.sysinit and /etc/rc.d/rc (rc 5) script start



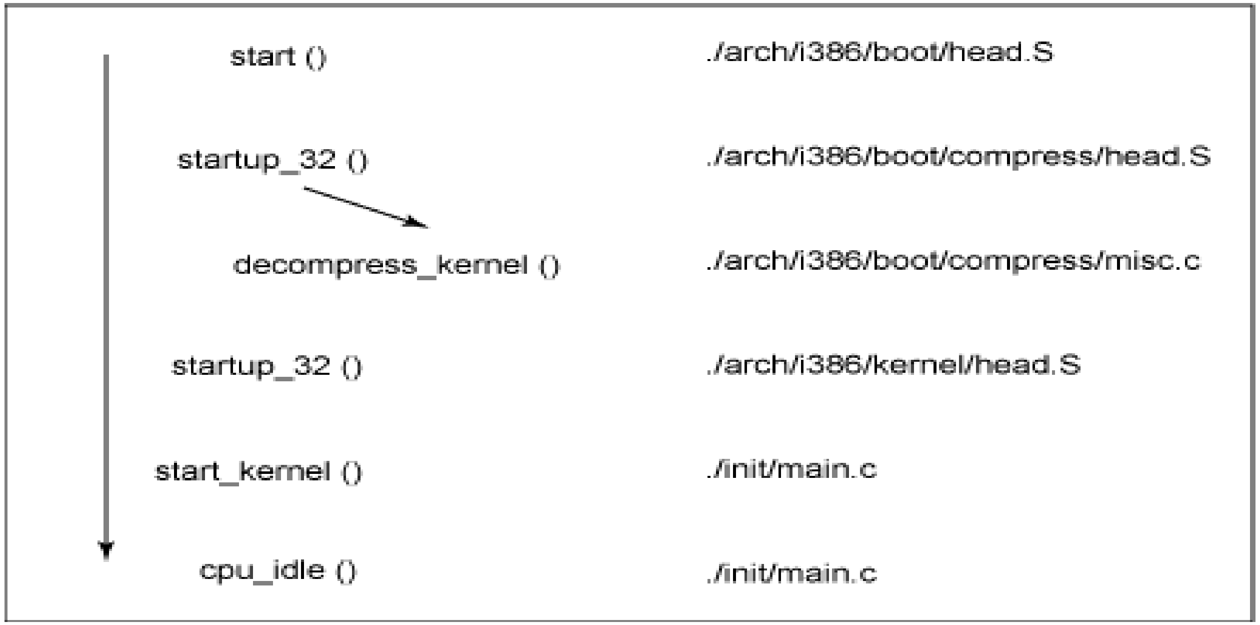
```
* /etc/inittab script
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
.....
l5:5:wait:/etc/rc.d/rc 5

* /etc/rc.d/rc script
# Now run the START scripts
for i in /etc/rc.d/rc$runlevel.d/S*; do

* /etc/rc.d/rc5.d/S99local last script
link to /etc/rc.d/rc.local
```

Master Boot Record

- **OS is booted from a hard disk, where the Master Boot Record (MBR) contains the primary boot loader**
- **The MBR is a 512-byte sector, located in the first sector on the disk (sector 1 of cylinder 0, head 0)**
- **After the MBR is loaded into RAM, the BIOS yields control to it.**
- **The MBR ends with two bytes that are defined as the magic number (0xAA55). The magic number serves as a validation check of the MBR**



Linux kernel boot



start ()	./arch/i386/boot/head.S
startup_32 ()	./arch/i386/boot/compress/head.S
decompress_kernel ()	./arch/i386/boot/compress/misc.c
startup_32 ()	./arch/i386/kernel/head.S
start_kernel ()	./init/main.c
cpu_idle ()	./init/main.c

1. With the kernel image in memory and control given from the stage 2 boot loader, the kernel stage begins
2. start assembly routine does some basic hardware setup and invokes the startup_32 routine : currently no start, only startup_32
3. startup_32 sets up a basic environment (stack, etc.) and clears the Block Started by Symbol (BSS)
4. The kernel is then decompressed through a call to a C function called decompress kernel (located in … /compress/misc.c)

Linux kernel boot

5. When the kernel is decompressed into memory, another startup_32() function(/arch/i386/kernel/head.S) is called. In the new startup_32() (also called the swapper or process 0), the page tables are initialized and memory paging is enabled. The type of CPU is detected.
6. With the call to start_kernel(), a long list of initialization functions are called to set up interrupts, perform further memory configuration, and load the initial RAM disk.
7. In the end of start_kernel(), a call is made to res_init() → kernel_thread() (in arch/i386/kernel /process.c) to start the init() → /sbin/init , which is the first user-space process(= spawn init() process as a "kernel thread")
8. After the call to cpu_idle() (=become idle process), the scheduler can now take control .
9. With interrupts enabled, the pre-emptive scheduler periodically takes control to provide multitasking.

initrd

- During the boot of the kernel, the initial-RAM disk (initrd) that was loaded into memory by the stage 2 boot loader is copied into RAM and mounted.
- This **initrd** serves as a temporary root file system in RAM and allows the kernel to fully boot without having to mount any physical disks.
- Since the necessary modules needed to interface with peripherals can be part of the initrd, the kernel can be very small
- After the kernel is booted, the root file system is pivoted (via pivot_root) where the **initrd** root file system is unmounted and the real root file system is mounted.

sbin/init process

- The first thing the kernel does is to execute init program
- init is the root/parent of all processes executing on Linux
- init process is identified by process id "1“
- init is responsible for starting system processes as defined in the /etc/inittab file

- Note –

* init() begins life as a “kernel thread” and ends by starting the user-level init process (/sbin/init)

* init() process in linux/init/main.c : kernel space program

* /sbin/init : user space program

Demon Services

- **Important Daemon Services** ; ***ntsysv, ps -e***
  - » **amd** - runs the automount daemon for remote filesystem mounting such as nfs (user)
  - » **anacron** - checks delayed `cron' tasks at boot time and executes them. Useful if you have cron jobs scheduled but don't run your machine all the time.
  - » **apmd** - Advanced Power Management BIOS daemon. For use on machines, especially laptops, that support apm. Monitors battery status and can shut down the system if power is too low.
  - » **arpwatch** - keeps watch for ethernet IP address pairings that are resolved using the ARP protocol(find mac address).
  - » **atd** - runs jobs at specific time by ‘***at***’ command(run only once)
  - » **autofs** - control the operation of automount daemons, used to mount and unmount devices on demand (kernel)
  - » **bootparamd** - allows computers to boot from a Linux machine using the BOOTP network protocol. A server process that provides information to diskless clients necessary for booting.

Demon Services

- **crond** - automatic task scheduler. Manages the execution of tasks that are executed at regular(/etc/crontab).
- **cups** - daemon for print services under the Common Unix Printer System, a replacement for lpd
- **dhcpd** - implements the Dynamic Host Configuration Protocol (DHCP) and the Internet Bootstrap Protocol (BOOTP). Used to lease out IP addresses to remote machines.
- **fetchmail** - daemon to fetch mail at regular intervals from mail servers
- **ftpd** - ftp server daemon
- **gpm** - useful mouse server for applications running on the Linux console.
- **httpd** - the Apache webserver hypertext transfer protocol daemon
- **inetd** – listens for service requests on network connections
  - In the version of Red Hat 7.0, it has been replaced by xinetd.

Demon Services

- **innd** - Usenet news server daemon
- **ipchains** - daemon for packet forwarding. Used for configuring a gateway/firewall.
- **keytable** - loads the appropriate keyboard map from /etc/sysconfig/keyboard
- **kudzu** - detects and configures new or changed hardware during boot (***kudzu –p***)
  - detect the current hardware, and check it against *\/etc/sysconfig/hwconf*.
- **lpd** - line printer and print spooler daemon
- **mysql** - database server daemon
- **named** - provides DNS services
- **netfs** - network filesystem mounter. Used for mounting nfs, smb and ncp shares on boot.
- **numlock** - locks numlock key at init runlevel change
- **pcmcia** - generic services for pcmcia cards in laptops
- **portmap** - needed for Remote Procedure Calls

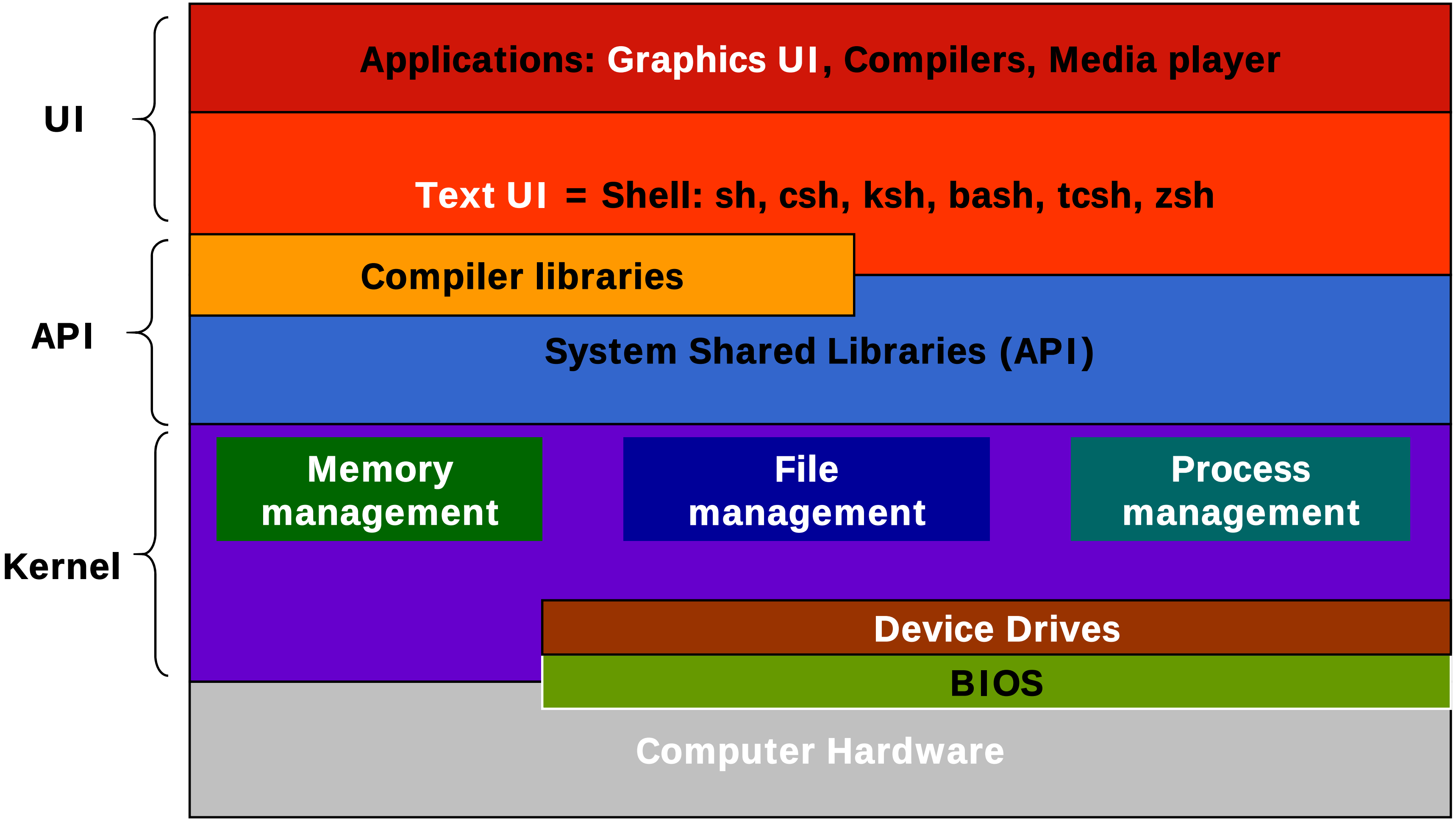
Demon Services

- **random** - random number generating daemon, related to security and encryption
- **sendmail** - mail transfer agent. This is the agent that comes with Red Hat.
- **smb** - needed for running SAMBA
- **snmpd** - provides Simple Network Management Protocol support
- **sound** - daemon for managing sound
- **squid** - web page proxy server daemon
- **network** - activates all network interfaces at boot time by calling scripts in */etc/sysconfig/network-scripts* (ifcfg-eth0 script)
- **nfsd** - used for exporting nfs shares when requested by remote systems
- **nfslock** - starts and stops nfs file locking service
- **syslogd** - manages system log files
- **telnetd** - telnet server daemon

Demon Services

- **usb** - daemon for devices on Universal Serial Bus
- **xfs** - X font server
- **xinetd** – On-demand super demon(more modern replacement for inetd)
  - xinetd replaces inetd as the default network services daemon in Red Hat 7.0.
  - Service configuration files in /etc/xinetd.d/ directory : tftp, bootp, ...
- **xntpd** - Network Time Protocol daemon. Provides a means to syncronize time over the network.
- **ypbind** - NIS binder. Needed if computer is part of Network Information Service domain(Sun directory service).

Linux Operating System


UI
API
Kernel
Applications: Graphics UI, Compilers, Media player
Text UI = Shell: sh, csh, ksh, bash, tcsh, zsh
Compiler libraries
System Shared Libraries (API)
Memory management
File management
Process management
Device Drives
BIOS
Computer Hardware

Linux System

- **User Interface :** **command.com in DOS**
  - » Interface between OS and User
  - » Divided into text based and graphical based
  - » Allow user to make commands to the system
- **System Shared Libraries(API)**
  - » Comprise a set of functions that can be used by the applications and library routines to use the services provided by the kernel
- **Kernel**
  - » The part of an OS where the real work is done
    - ◆ Memory management
      - → Memory in a computer is divided into main memory (RAM) and secondary storage (usually refer to hard disk)
      - → Memory is small in capacity but fast in speed, and hard disk is vice versa
      - → Data that are not currently used should be saved to hard disk first, while data that are urgently needed should be retrieved and stored in RAM
      - → The mechanism is referred as memory management

Linux System

- Process Management
  - For a multitask system, multiple programs can be executed simultaneously in the system
  - When a program starts to execute, it becomes a process
  - Kernel manages processes in terms of creating, suspending, and terminating them
  - A process is protected from other processes and can communicate with the others
- File Management
  - Control the creation & removal of files, and provide directory maintenance
  - For a multiuser system, every user should have its own right to access files and directories

- **Device drivers**
  - Interfaces between the kernel and the BIOS (Hardware if no bios)
  - Different device has different driver

Shell Command

- **Shell types supported in RH9**
  - ***vi /etc/shells*** : sh(bourne), ash, csh, bash(default), tcsh
- **Some file names are special**
  - / The root directory (not to be confused with the root user)
  - . The current directory
  - .. The parent (previous) directory
  - ~ My home directory (cd)
    - pwd : display absolute path of working directory
  - ***./command_name*** - run a command in the current directory when the current directory is not on the path
- **Pathnames**
  - Absolute:
    - List of directory names from **root directory** to desired file or directory name, each separated by / -> /usr/share/xpdf
  - Relative:
    - List of directory names from **working directory** to desired file or directory name, each separated by / -> usr/share/xpdf

Shell Command

- **Shell variables**
  - Standard variables
    - **DISPLAY** – the name of X window display
    - **HOME** – the name of home directory : /etc/passwd
    - **USERNAME** – the name of user name
    - **SHELL** – path to shell (/bin/bash) : /etc/passwd
    - **MAIL** – path to incoming mailbox
    - **PWD** – shell's current directory
  - ***echo "My home is $HOME"***
  - user variable : ***var=3; echo $var***
    - ***$*** : to refer to a value
- **Quoting**
  - Single quote : literal = constant
    - ***echo ' HOME has value $HOME'*** -> HOME has value $HOME
  - Double quotes : variable
    - ***echo "HOME has value $HOME"*** -> HOME has value /root

Shell Command

- **Escaping**
  - ₩ : backward slash (escape character)
    - Override variable : force literal
  - Variable
    - ***echo "HOME has value $HOME"*** -> HOME has value /root
  - Literal
    - ***echo "HOME has value ₩$HOME"*** -> HOME has value $HOME

- **Combining commands**
  - ***command 1 ; command 2 ; command 3***
    - To invoke several commands in sequence on a single command line
  - ***command 1 && command 2 && command 3***
    - Stop execution if any of them fails
  - ***command 1 || command 2 || command 3***
    - Stop execution as soon as one succeeds

Shell Command

- **Redirection**
  - » ***command > file*** direct output of *command* to *file* instead of to standard output (screen), replacing current contents of *file*
  - » ***command >> file*** as above, except output is **appended** to the the current contents of *file*
  - » ***command < file*** *command* receives input from *file* instead of from standard input (keyboard)
  - » ***cmd1 | cmd2*** "pipe" output of *cmd1* to input of *cmd2*
    - ◆ ***ls | more, ls | grep test***
  - » ***tee*** read from std input and write to std output and file
    - ◆ ***sort < unsorted | tee sorted***

Shell Command

- **Control characters (special keystrokes)**
  - ***Ctrl-C*** break, kill almost any program that is running
  - ***Ctrl-D*** end-of-file(input from the keyboard) or logout from current terminal
  - ***Ctrl-L*** refresh the window
  - ***Ctrl-S*** pause screen output. ***Ctrl-Q*** key to resume output
  - ***Ctrl-U*** erase the entire line you're typing.
  - ***Ctrl-Z*** suspend the job that is running
    - To send it to run in the background, hit "***Ctrl-Z***", then type “***bg***".
- **Control + Alt**
  - ***[Alt][Ctrl][F1]*** switch to the first virtual text console
    - ***[Alt][Ctrl][Fn]*** - switch to the nth virtual text console. Typically, there are six on a Linux PC system.
  - ***[Alt][Ctrl][F7]*** switch to the first GUI console, if there is one running.
  - ***[Alt][Ctrl][Del]*** switch to run level 6(reboot)
    - text command mode only (logout menu display at x window)

Shell Command

- **Create a File**
  - » ***cat > file*** enter text and end with ***ctrl-D***
  - » ***vi file*** edit *file* using the ***vi*** editor
- **Make a Directory**
  - » ***mkdir directory***
- **Remove File / Directory**
  - » ***rm file*** remove (delete) a file
  - » ***rmdir dir*** remove an empty directory
  - » ***rm -rf dir*** remove a directory and its contents
  - » ***rm -i file*** remove file, but prompt before deleting
    - ◆ default : alias at .bashrc (***rm, mv, cp***)
- **Move (or Rename) Files and Directories**
  - » ***mv src-file dest-file*** rename *src-file* to *dest-file*
  - » ***mv src-file dest-dir*** move a file into a directory
  - » ***mv src-dir dest-dir*** rename *src-dir (=* move to *dest-dir*)
  - » ***mv -i src dest*** prompt before overwriting

Shell Command

- **List Files and Directories**
  - ***ls*** list contents of directory
  - ***ls -a*** include files with "." (dot files)
  - ***ls -l*** list contents in long format (show modes)
  - ***ls -F*** list files in current directory and indicate the file type
    - * - executable , @ - symbolic link
- **Copy Files**
  - ***cp src-file dest-file*** copy *src-file* to *dest-file*
  - ***cp src-file dest-dir*** copy a file into a directory
  - ***cp -r src-dir dest-dir*** copy one directory into another
  - ***cp -a src dest*** copy with archive option all copy with no symbolic link
    - ***a*** = ***dpr*** : no-dereference(no symbolic link), preserve(mode), recursive
- **Comparing Files**
  - ***diff file1 file2*** line by line comparison
  - ***cmp file1 file2*** byte by byte comparison

Shell Command

- **Display File Contents**
  - » ***cat file*** display contents of *file*
  - » ***more file*** display contents of *file* one screen full at a time.
  - » ***view file*** read only version of vi
  - » ***head file*** display the first 10 lines of *file*
  - » ***head -20 file*** display the first 20 lines of *file*
  - » ***tail file*** display the last 10 lines of *file*
  - » ***tail -20 file*** display the last 20 lines of *file*
- **Printing**
  - » ***lpr file*** print *file* on default printer
  - » ***lpr -Pprinter file*** print *file* on *printer*
  - » ***lpr -c# file*** print *#* copies of *file*
  - » ***lpq*** show print queue
  - » ***lprm -#*** remove print request *#* (listed with ***lpq***)

Shell Command

- **Changing Access Modes**
  - ***chmod mode file1 file2 ...*** (changes **files**)
  - ***chmod -R mode dir*** (changes **all files and directories** in *dir*)
    - Mode Settings
      - → **u** user (owner) : **g** group : **o** other
      - → **+** add permission : **-** remove permission
      - → **r = 4** read : **w = 2** write : **x = 1** execute
    - ***chmod go +rwx file***
      - → Adds read, write, and execute permissions for *group* and *other* on file.
    - ***chmod 7 5 5 file***
      - → Full permission for the owner, read and execute access for the group and others.
    - ***chmod +x file***
      - → Make the file executable to all users.

Shell Command

“grep” : g /re/ p (***ed***)
search globally for lines matching the regular expression, and print them.

Search Files and Text within Files

- ***grep string file (or filelist)*** show lines containing *string* in any file in *filelist*
- ***grep -v string file (or filelist)*** show lines not containing *string*
- ***grep -i string file (or filelist)*** show lines containing *string*, ignore case
- ***grep –r string /dir*** look for and list all files containing *string* (start from */dir*)
- ***find / -name file*** starting with the root directory, look for *file*
- ***which file*** show the subdirectory containing the executable file called *file* : ***which vi***
- ***whereis file*** Locate the binary, source and man page files for a command.
- ***updatedb*** create or update the database of files on all file systems attached to the linux root directory
- ***locate file*** find a file called *file* using the ***locate*** command – this assumes you have already used the command ***updatedb***

Shell Command

- **Compressing / Decompressing Files**
  - ***compress file*** encode *file*, replacing it with *file.Z*
  - ***uncompress file.Z*** decode *file.Z*, replacing it with *file*
  - ***tar cvf file.tar file or dir*** compress *file* or *directory*
  - ***tar xvf file.tar*** decompress *file.tar*
  - ***gzip file.tar*** resulted file is *file.tar.gz*
  - ***gzip –d file.tar.gz*** uncompressed file is *file.tar*
  - ***tar –czvf file.tar.gz file*** or ***dir*** compress file or directory, first ***tar*** and then ***gzip***
  - ***tar -xzvf file.tar.gz*** or ***tar -xzvf file.tgz*** decompress the files contained in the zipped and tarred archive.
  - ***tar -xjvf file.tar.bz2*** decompress the files contained in the zipped and tarred archive.
  - ***zcat file.Z*** display compressed file

Shell Command

- **Filesystem Management**
  - ***dump*** Used to back up an ext2 filesystem. Complement is restore.
  - ***restore*** Used to restore an ext2 filesystem.
  - ***fdisk*** Used to fix or create partitions on a hard drive.
  - ***fdformat*** Formats a floppy disk.
  - ***fsck*** Used to check and repair a Linux file system.
    - Must not be run on a mounted file system (use e2fsck)
  - ***mkfs*** Initializes a Linux filesystem. This is a front end that runs a separate program depending on the filesystem's type.
    - ***mke2fs*** Create a Linux second extended filesystem.
  - ***mount*** Used to mount a filesystem. Complement is umount.
  - ***umount*** Unmounts a filesystem. Complement is mount.

Shell Command

“job” definition
Programs and scripts run by users in a bash shell

- **Linux Job Management**
  - ***at*** Similar to cron but run only once.
    - ***at 23:55 12/31/09 < script file*** , ***at -l***
  - ***atq*** Lists the user's pending jobs. *= at -l*
  - ***atrm number*** Deletes *at* job *number*(at ***at –l*** command).
  - ***batch*** Executes commands when system load levels drop below 0.8 (/proc/loadavg). **same as at> *command*** [cnt+d]
  - ***cron*** A demon used to set commands to be run at specific times. Starts the commands in the ***crontab*** file.
    - ***30 8 * * mon /root/scripts/backup.sh***
      - → run custom script every Monday at 8:30AM
  - ***nice*** Run a program with modified scheduling priority from -20(hightest) to +19(lowest).-> 10 by default
    - ***nice -5 sort file1 > file2***
  - ***nohup*** Run a command immune to hangups (allows a process to continue after you log out).
    - ***nohup sleep 1000***

Shell Command

- **Linux Process management**
  - ***bg*** Starts a suspended process in the background
  - ***fg*** Starts a suspended process in the foreground
  - ***jobs*** report current jobs and job-id numbers
  - ***kill pid or %job_id*** Kill a process or jobs
  - ***Kill -9 pid or %job_id*** The -9 flag forces the process to die
  - ***ps*** Lists all current running processes and pids
  - ***ps ax |more*** to see all processes including daemons
  - ***pstree*** Display the tree of running processes.
  - ***top*** Display the processes that are using the most CPU resources.
  - ***&*** Run a job in the background.
- **Information on Users**
  - ***finger*** *or* ***finger user*** get information on a *user*
  - ***who*** who is currently logged in
  - ***w*** show who is logged in and what they are doing

Shell Command

- **Network setup and commands**
  - ***ftp*** File transfer program.
  - ***ifconfig*** Configure a network interface.
  - ***ifdown*** Shutdown a network interface.
  - ***ifup*** Brings a network interface up.
  - ***iptables*** Administration tool for packet filter and NAT
    - ***iptables –F : flush all the rules***
  - ***netconfig*** GUI step by step network configuration program.
  - ***netstat*** Displays information about the systems network connections, including port connections, routing tables, and more. The command "***netstat -r***" will display the routing table.
  - ***nslookup*** Used to query DNS servers for information about hosts.
  - ***ping*** is there anybody out there? Check a host for existence
  - ***portmap*** RPC program number mapper. Must be running to make RPC calls.

Shell Command

  - » ***route*** Show or manipulate the IP routing table.
  - » ***showmount*** Show mount information for an NFS server.
  - » ***tcpdump*** Dump traffic on a network. Prints out headers of packets that match the boolean expression.
  - » ***telnet*** User interface to the TELNET protocol.
  - » ***traceroute*** show me how to get from here to there.

- **Starting & Stopping**
  - » ***shutdown -h now*** Shutdown the system now and do not reboot
  - » ***halt*** Stop all processes - same as above
  - » ***shutdown -r now*** Shutdown the system now and reboot
  - » ***reboot*** Stop all processes and then reboot – same as above
  - » ***Poweroff*** Shutdown the system now and power-off

Shell Command

- **Module and Kernel Management**
  - ***depmod*** Handle loadable modules automatically. Creates a makefile-like dependency file. /lib/modules/2.4.20-8/modules.dep
  - ***insmod*** Install loadable kernel module.
  - ***modprobe*** Used to load a set of modules. first check module dependency, and then install modules. /etc/modules.conf
  - ***lsmod*** List currently installed kernel modules.
  - ***rmmod*** Unload loadable modules.
  - ***dmesg*** Print or control the kernel ring buffer. This shows the last kernel startup messages.
  - ***genksyms*** Generate symbol version information. *.ver
- **Timesavers**
  - ***alias*** *string command* : ***alias dir='ls –lF', alias cls='clear'***
  - ***history*** show command history
    - ***!num*** - repeat command with history number *num*
    - ***!!*** - repeat previous command
    - ***!$*** - the last parameter from the previous command

Shell Command

System Information

- ***arch*** Display machine architecture.
- ***df*** Shows disk free space .
- ***du*** Shows disk usage, disk directory and all its files contain.
- ***free*** Display used and free memory on the system.
- ***lsdev*** Display information about installed hardware via files in the /proc directory.
- ***lsof*** List open files.
- ***lspci*** List PCI devices .
- ***procinfo*** Display system status gathered from proc.
- ***runlevel*** Find the current and previous system runlevel.
- ***strace*** Trace sytem calls and signals for a binary program.
- ***stty*** Display and change terminal line settings.
- ***tty*** Print the filename of the terminal connected to standard input.
- ***uname*** display the machine and operating system name

Shell Command

- **Linux User Management**
  - » ***ac*** Print statistics about users' connect time.
  - » ***useradd*** Adds a user to the system
  - » ***adduser*** same as *useradd* (symbolic link to *useradd*)
  - » ***userdel*** Delete a user account and related files.
  - » ***usermod*** Modify a user account.
  - » ***users*** Print the user names of users currently logged in.
  - » ***chgrp*** Changes the group ownership of file.
  - » ***chown*** Change the owner of file to another user.
  - » ***ash, bash, csh*** Change the login shell.
  - » ***finger*** See what users are running on a system.
  - » ***groupadd*** Create a new group.
  - » ***groupdel*** Delete a group.
  - » ***groupmod*** Modify a group.
  - » ***passwd*** Set a user's pass word.

Shell Command

- ***su*** Create a shell with substitute user ID
- ***sulogin*** Single user mode login, need root password
  - not part of any currently supported standard; it is an extension of AT&T System V
- ***vigr*** Edit the group file = vi /etc/group
- ***vipw*** Edit the password file = vi /etc/passwd
- ***wall*** Send a message to everybody's terminal.
- ***whoami*** Print current user id.
- ***quota*** Display users' limits and current disk usage.
- ***quotacheck*** Used to check a filesystem for usage, and update the quota.user file.
- ***repquota*** Lists a summary of quota information on filesystems.

Environment

- ***env*** Show all environment variables.
- ***export*** Set the value of a variable so it is visible to all subprocesses.
- ***reset*** Restores runtime parameters for session to default values.

Shell Command

- **Miscellaneous Commands**
  - » ***clear*** clear the screen
  - » ***echo*** display a message on the screen
  - » ***file*** Display file information
  - » ***dd*** Convert and copy a file.
    - ◆ ***dd if=boot.img of=/dev/fd0***
  - » ***ln*** Make links(alias) between files.
    - ◆ Hard link : default, same size, same inode (***ls –i***)
    - ◆ Symbolic link : -s option, pointer(small size)
  - » ***mknod*** Make a block or character special file.
  - » ***touch*** Change file timestamps to the current time. Make the file if it doesn't exist.
  - » ***patch*** Apply a diff file to an original.
  - » ***sleep number*** Delay for a specified *number* of second.
  - » ***wc file*** Count lines, words, characters in *file*.

Shell Command

- **Linux Programming**
  - » ***as86*** Assembler
  - » ***ld86*** Linker for as86
  - » ***awk*** programming utility - allows finding of lines with specific characters (Pattern scanning and processing language)
    - ◆ ***gawk '{sum += $1}; END {print sum}' file.txt***
    - ◆ **Sum *the first column* of numbers in a *file.txt* *(cat > file.txt)***
  - » ***gawk*** GNU's implementation of awk.
  - » ***gcc*** GNU c and c++ compiler.
  - » ***gdb*** Debugging program.
  - » ***ld*** GNU linker.
  - » ***make*** GNU make utility to maintain a group of programs

Shell Command

- **System time**
  - » ***cal*** Calendar.
  - » ***clock*** Used to change or get current time.
  - » ***date*** Print or set the system date and time.
  - » ***hwclock*** Set or read the hardware CMOS clock.
  - » ***uptime*** Reports how long the system has been running.

- **rpm**
  - » ***rpm -ihv name.rpm*** Install the rpm package called name
  - » ***rpm -Uhv name.rpm*** Upgrade the rpm package called name
  - » ***rpm -e package*** Delete the rpm package called package
  - » ***rpm -ql package*** List the files and state the installed version of the package called package
  - » ***rpm -qa*** list all packages installed

Documentations

- **Editor**
  - vi (vim), gedit, kedit, emacs
  - vi editor
    - Command mode – initial state
    - Insert mode – entered with ***a***, ***i***
    - Last line mode – entered with **:** ***?*** , ***/***
- **Documentation**
  - ***man command*** display on-line manual pages (/usr/share/man)
  - ***man -k string*** list one-line summaries of manual pages containing *string*
  - ***apropos keyword*** - Show all commands with the keyword in their description. The same as the "***man -k***" command.
  - ***whatis*** Search the whatis database using manual pages.
    - /etc/cron.daily/makewhatis.cron → makewhatis command
  - ***info command*** readable Info documents (more detail /usr/share/info)
  - ***/usr/share/doc***

Shell Script

- **Shell script**
  - Shell Script is **series of command** written **in plain text file**(batch file)
    - Useful to create our own commands.
    - Save lots of time.
    - To automate some task of day today life.
    - System Administration part can be also automated.
- **Following steps are required to write shell script:**
  - 1. Use any editor like vi to write shell script.
    - The first line of the file must be ***#!/bin/csh***
  - 2. After writing shell script, set execute permission for the script as follows*:*
    - ***chmod +x script-name***
  - 3. Execute the script as*:*
    - ***bash script-name, sh script-name, ./script-name***