Boot Process:

* System goes through POST (reads BIOS settings and initializes hardware)
  + Power is initialized
  + Bios is initialized
  + POST tests internal and external components
  + Hard drives and RAID systems are initialized
* System determines primary boot device
* System reads Master Boot Record of device (512mb)
  + Stage 1 Bootloader
* Depending on installed boot loader, Grub or LILO is loaded
* Boot loader determines and decompresses the default kernel image
* Kernel loads modules for all the hardware and initializes it
* Kernel starts the init process
* Init reads its configuration file (/etc/inittab) to determine the default runlevel
* Init starts processes located in the default runlevel’s rc directory
* Init starts processes in rc.local
* Init starts TTY sessions and user is presented with a login prompt

GRUB/LILO +Pros/Cons

* Grub – Grand Unified Boot Loader use to load the operating system
  + Stage 1: Embedded on the MBR of a disk or in the boot sector of a partition. The image file loads Stage 1.5 or 2 directly
  + Stage 1.5:
    - This image serves as a bridge between Stage 1 and Stage 2
    - Allows GRUB to access several file systems and helps locate the Stage 2 image as a file within the file system
  + Stage 2:
    - Contains the actual code to load the kernel that boots the OS
* LILO – Linux Loader boot manager
  + LILO is a two stage boot loader
    - The first stage loads LILO itself into the memory and prompts you for booting instructions with the lilo: prompt or colorized boot menu
    - The second stage boots the linux operating system
* Pros/Cons
  + Grub can load a Linux Kernel from an ext2 or ext3 File system, LILO cannot

Run Levels

* 0 = Halt
* 1 = Single User
* 2 = Multiuser/No Network File System (NFS)
* 3 = Full multiuser mode
* 4 = User defined
* 5 = Multiuser using an X Windows system login
* 6 = Reboot

Directory Structure

* Ext2
* Ext3
  + Journalized file system
    - File system keeps a record of all writes to the actual disk. This can be used in data recovery situations to redo all the transactions.
* / - root filesystem
  + /boot – contains kernel images and Grub if it is the bootloader
  + /etc – contains config files
  + /bin – contains system binaries
  + /sbin – contains secure system binaries
  + /lib – contains system library files
  + /home – user directories
  + /var – contains log files, and generally runtime files (lock, spool, etc…)
  + /dev – Virtual filesystem that contains devices
  + /proc – virtual filesystem that holds kernel parameters and process information
    - /prc/cpuinfo – P242-252
* File permission p112-118
* EDAC – Error Detection and Correction - The 'edac' kernel module goal is to detect and report errors that occur within the computer system running under linux.
  + EDAC sysfs Interface resides in the /sys/devices/system/edac directory
    - Two components
      * mc memory controller(s) system
      * pci PCI control and status system
* ECC – error checking and correction ECC (either "error correction [or correcting] code" or "error checking and correcting") allows data that is being read or transmitted to be checked for errors and, when necessary, corrected on the fly. It differs from parity-checking in that errors are not only detected but also corrected. ECC is increasingly being designed into data storage and transmission hardware as data rates (and therefore error rates) increase.
* Kernel - The kernel is a program acting as chief of operations. It is responsible for starting and stopping other programs such as editors, handling requests for memory accessing disks, and managing network connections.

Commands

* Everything is treated as a file
* File Lengths can go to 255 chars. Keep to 80 chars. as this is the width of a normal terminal window.
* Everything is case sensitive
  + WildCards:
    - \* matches any character zero or more times.
    - ? matches a single character.
    - [] matches either a set of single characters [12] or a range of characters separated by a hyphen [1-3].
  + Symbols and File Types:
    - \*=Executable
    - /=Directory
    - @=Symbolic link
    - |=FIFO
    - (=)=Socket
  + Shell Color Codes:
    - Default=Regular file
    - Green=Executable
    - Blue=Directory
    - Magenta=Symbolic link
    - Yellow=FIFO
    - Magenta=Socket
    - Red=Archive(.tar,.zip,.deb,.rpm)
    - Magenta=Images(.jpg,.gif,.png,.tiff)
    - Magenta=Audio(.mp3,.ogg,.wav)
  + Stringing cmds:
    - ;= allows you to string multiple commands together. This cmd runs regardless of a successful completion of the last cmd
    - &&= allows you to string multiple commands together. This cmd will halt if one of the commands in the string fails
    - ||= allows you to string multiple commands together. This command will only run the second cmd if the first one fails (Either/or)
    - ()= allows the command output from one command and plugs it into another command as though you had typed that output in directly
    - |= when placed between 2 cmds, it takes the output from the first and uses it as input for the second
    - >= redirects a cmd's output to a file
    - >>= appends a cmd's output to a file
    - <= uses a file as input for a cmd
  + What if statement, an array, a float, a for loop, a while loop, how variables are declared.
* Basic commands –
  + Cd Changes the current directory
  + cd ~ Takes you back to the home directory
  + cd - Takes you back to the previous directory, then it automatically runs the pwd cmd for you
  + ./ - default for root is that the path does not include the current directory you are in. Just to save yourself from shooting yourself in the foot so you are saying ./hostiquiry that means in my current directory run the command hostinquiry it's a safety so that if you are in a dir and run a command as root it won't work because it doesn't know where it is so ./ means right here
  + Pwd - Displays the full, absolute path of the current working directory p123,131,424-425
  + ~= Means home directory
  + touch (-a=access)(-m=modify)= Updates the access and modification times of a file
  + touch -t= This variant allows you to change the time to any time as opposed to the current time
  + touch= Creates a file when you specify a file that is not there
  + mkdir= Creates a directory
  + mkdir -p= Creates subdirectories
  + mkdir -v= Tells you what mkdir is doing
  + cp= Copy files
  + cp -v= Displays the progress of the copy command
  + cp -i= Warns you so you do not overwrite files when copying them
  + cp -r= Allows you to copy directories w/the full content
  + cp -a= Allows you to make perfect backup of files
  + mv= Moves/renames a file (uses the same options as cp)
  + rm= Removes a file
  + shred= Deletes a file for good by writing over it 25x
  + rmdir= Removes an empty directory
  + rm -RF= Will delete everything (Be careful when using as root)
  + su = Means Switch User | Gives root access
  + su <username>= Allows you to temporarily act as another user
  + su -l= Allows you to temporarily act as another user with their environment variables
  + su -= Allows root access with its environment variables
  + whoami =Displays the current user
  + man= (manual) Gives information about various commands
  + /(keyword)= Searches the man page for your keyword
  + q= Quits the manual
  + -k= Search for a command based on what it does
  + -f= Find out what a command does based on its name
  + -u= Rebuilds the man's DB
  + -t= Prints the man page
  + info= Like man page but is broken down into nodes and subnodes
  + ctrl + 1= Redraws the screen if the info page becomes distorted
  + q= Quits the info page
  + whereis= Displays the path for commands, executable file, source files and man pages
  + -b= binaries
  + -m= man pages
  + -s= sources
  + whatis= like man -f this displays the man page definition for a command
  + which=displays which version of a command will run
  + /
  + List Commands:
    - ls=list ; ls works with relative or absolute paths
    - ls -R= lists the contents of several subdirectories
    - ls -1= formats to a single column
    - ls -m= formats with commas
    - ls -a= displays all files including hidden
    - ls -f= displays all info per files
    - ls --color= displays results by color codes
    - ls -L= displays permissions
    - ls -x= displays content by file extension
    - ls -t= displays content by time
    - ls -s= displays content by size
    - ls -h= makes the display easier to read
* View Files:
  + cat= p125 prints the file to the screen and places you back at the prompt
    - used to read contents of files
    - to redirect output use “>” and file name.
      * i.e. cat inittab > test. Instead of displaying on monitor the output from cat to the file “test”
      * cat > ABC (Enter) type text (Enter) writes text to the ABC file
      * Ctrl+D to restore command prompt
    - less= prints a large file and lets you navigate it easier
    - -Q= quit less
    - -v= opens a text editor in less
    - head= prints the first 10 lines in a file, head file1... or to display multiple files head file1 file2
    - tail= prints the last 10 lines in a file
    - -f= allows tail to stay open and allows to to watch the changes to a log file as they happen
  + Advanced commands
    - IOstat – Report CPU statistics and input/output statistics for devices and partitions. The **iostat** command generates reports that can be used to change system configuration to better balance the input/output load between physical disks.
    - Dmesg - used to examine or control the kernel ring buffer. The program helps users to print out their bootup messages. Instead of copying the messages by hand, the user need only: dmesg > boot.messages

and mail the *boot.messages* file to whoever can debug their problem.

* + - * -c - Clear the ring buffer contents after printing.
      * -s*bufsize -* Use a buffer of size *bufsize* to query the kernel ring buffer. This is 16392 by default. If you have set the kernel buffer to be larger than the default then this option can be used to view the entire buffer.
      * -n*level-* Set the *level* at which logging of messages is done to the console. For example, -n 1 prevents all messages, expect panic messages, from appearing on the console. All levels of messages are still written to */proc/kmsg*, so syslogd can still be used to control exactly where kernel messages appear. When the -n option is used, dmesg will *not* print or clear the kernel ring buffer. When both options are used, only the last option on the command line will have an effect.
    - Grep –
    - VMstat – reports information about processes, memory, paging, block IO, traps, and CPU activity.
      * -a displays active/inactive memory
      * -f displays the number of forks since boot.
      * -m slabinfo
      * -n causes the header to be displayed only once rather than periodically
      * -s table of various event counters and memory statistics.
      * Delay – is the delay between updates in seconds
      * Count – is the number of updates
      * -d reports disk statistics
      * -p followed by some partition name for detailed statistics
      * -S followed by k or K or m or M switches outputs between 1000,1024, 1000000, or 1048576 bytes
      * -V displays version information
    - ps – the ps command lists all the processes in a system, their state, size, name, owner, CPU time, etc. p131
      * Common options
        + -a Shows all processes with a controlling terminal
        + –r only running processes
        + –x processes that do not have a controlling terminal
        + –u process owners
        + –f displays parent/child relationships among processes
        + –l produces a list in long format
        + –w process’s command-line parameters (up to half a line)
        + –ww process’s command-line parameters (unlimited width fashion)
    - Chmod – Used to set permission values. p116
    - Chown – allows you to change the ownership of a file to someone else (only the root user can do this)p115
    - Ifconfig – Primarily responsible for setting up network interface cards p315, 301
      * Resides in the /sbin directory
    - fstab - /etc/fstab a configuration file that mount can use. p173
      * It contains a list of all partitions known to the system
      * During the boot process this list is read and the items in it are automatically mounted with the options specified therin.
    - Vi – Text editor
      * Shift+zz to save
      * I/insert to change/edit
      * Help
      * q! (press escape several times -> type q! to force an exit without saving
      * wq save
  + Performance Management/Troubleshooting
    - Explain built-in disk utilities and files and what they do
      * Iostat – reports CPU statistics and I/O statistics for devise and partitions
      * Scsi\_info – SCSI device descriptions
      * Cat /proc/scsi/scsi – SCSI device lists
      * Cat /proc/partitions – list of system partitions
    - What is vmstat?
      * Virtual memory statistics utility
    - What is MTU and how do you set it?
      * MTU is Maximum Transfer Unit, which measures the size, in bytes, of the largest packet that a communication protocol can pass, and it is set through ifconfig
    - What is a for I loop?
      * Conditional loop that allows repeated executions (iteration loop)
    - A new disk partition you created works fine during the first mount. After a system restart, the partition is no longer mounted or available. What steps do you have to take to mount the partition on future reboots?
      * You must add an entry to /etc/fstab
    - What’s the difference between a symbolic link and a hard link?
      * Symbolic link creates a new inode and contains a pointer to the original file/directory
    - You’ve rebooted it, how would you find out what caused the crash?
      * /var/log/\*
      * Remote logging
        + How would you set this up?

Syslog.conf

* + - * Remote crashdumps
        + How would you set this up?

netdump

* + - fsck
      * Why/when is it run?
        + Usually on boot, particularly if you’ve crashed the server
        + Runs on boot after a number of remounts set in the superblock
        + How do you set/change this number?

Mkfs, tunefs

* + - * + Can be manually run
        + How do you skip it on bootup?

Flag file e.g. .fastboot or .autofsck

What can break?

* + - * + How do you fsck when your root partition is dead?

Boot from non-local medium e.g. PXE or CD

* + - * + How do you recover dfrom a bad superblock?

Use alternate superblock

How do you find that?

Mkfs –n (or equivalent “no-write” flag)

* + - Problem: “Can’t log into server”
      * Network responds (ping)
      * Ssh doesn’t allow you in
        + Is SSH running? How can you tell (remotely)?

telnet to port 22/ssh -v

* + - * reverse DNS not working (need to check exactly how ssh responds!)
      * MTU issues (tcpdump will spot this, also manually tweaking mtu)
      * User not in passwd file
      * User’s shell is crippled
        + Radius server unreachable
      * Recover from one of the above
    - You get a trouble ticket that State my server is experiencing slow IO, what tools do you use?
      * IOstat
      * vmstat

Misc.

* Swap file? Alternatively referred to as a **page file** or **paging file**, a **swap file** is a file stored on the computer hard disk drive that is used as a temporary location to store information that is not currently being used by the computer RAM. By using a swap file a computer has the ability to use more memory than what is physically installed in the computer. A swap file allows an operating system to use hard disk space to simulate extra memory. When the system runs low on memory, it swaps a section of RAM that an idle program is using onto the hard disk to free up memory for other programs. Then when you go back to the swapped out program, it changes places with another program in RAM. This causes a large amount of hard disk reading and writing that slows down your computer considerably.

This combination of RAM and swap files is known as virtual memory. The use of virtual memory allows your computer to run more programs than it could run in RAM alone.

The way swap files are implemented depends on the operating system. Some operating systems, like Windows, can be configured to use temporary swap files that they create when necessary. The disk space is then released when it is no longer needed. Other operating systems, like Linux and Unix, set aside a permanent swap space that reserves a certain portion of your hard disk.

Permanent swap files take a contiguous section of your hard disk while some temporary swap files can use fragmented hard disk space. This means that using a permanent swap file will usually be faster than using a temporary one. Temporary swap files are more useful if you are low on disk space because they don't permanently reserve part of your hard disk.

* inode
  + A Block on the hard drive that contains the pointer to the file and all its attributes (size, permissions, etc.)
* Changing the default runlevel using Vi
  + /etc/inittab
  + Default runlevel 3