Linux commands

- cd
    - cd ~ - change working directory to home directory
    - cd - - change working directory to old working directory
    - cd .. - change working directory to parent directory
- ls
    - ls - list the contents of present working directory
    - ls path - list the contents of given path
    - ls -l - list the contents in detail format
        - type and permissions
            - Types of files
                - Regular file (-)
                - Directory file (d)
                - Link file (l)
                - pipe file (p)
                - socket file (s)
                - character special file (c)
                - block specical file (b)
            - Permissions of files
                - r - read, w - write, x - execute
                - (rwx)user/owner, (rwx)group, (rwx)others
        - link count
        - user/owner
        - group
        - size
        - timestamp
        - name
    - ls -a - list all contents along with hidden
    - ls -A - list all contents along with hidden except . and ..
    - ls -i - list contents with indoe number
        - inode number is unique number given to every file
    - ls -s - list content with size (number of blocks)
    - ls -S - list content in descending order of their sizes
- touch
    - if file does not exist, empty file is created
    - if file exist, timestamp of that file is changed
- stat
    - stat file - display information of file
    - stat file1 file2 - display information of file1 and file2
    - stat -c "format" file - display file information in given format

- head
    - head file - display first 10 lines
    - head -5 file - display first 5 lines
- tail
    - tail file - display last 10 lines
    - tail -4 file - display last 4 lines
- sort
    - sort file - sort the content by alphabetically
    - sort -n file - sort the content by their value
        - sort command do not modify file content
- uniq
    - uniq file - display contents uniquely (truncate duplicate)
        - truncate duplicate content if it is consecutive
- rev filepath
    - Print each line reversed.
    - File contents are not modified.
- tac filepath
    - Print all lines in reverse order. The first line printed at last, while last line printed first.
- stat path
    - Display info about file or directory.
- alias
    - alias list="ls -l"
        - list will be alias/nick name to ls -l
        - list will give output same as ls -l
- unalias
    - unalias list
- which
    - which command
        - display the location of command executable.
- whereis
    - whereis command
        - display the location of command executable and also manual paage location.

Types of commands

- Internal commands
  - commands are part of shell program only.
  - No seperate executable is present
  - eg. alias, unalias, cd
- External commands
  - commands are not part of shell program
  - Seperate executable is available in any one of the bin directory
  - eg. ls, cp, mv

Redirection

- for every command input is taken from terminal, output is printed on terminal and error is also printed on terminal
- Standard streams (by defualt for every process, three files are opened)
  - stdin
  - stdout
  - stderr
- There are three types of redirections
  - input redirection
    - input will be taken from file instead of stdin
    - to do input redirection '<' symbol is used
    - command < file
  - output redirection
    - output will be written into file instead of stdout
    - to do output redirection '>' or '>>' symbol is used
    - command > file
      - older content of file will be over written
    - command >> file
      - content will be appneded into file at the end
  - error redirection
    - error will be written into file instead of stderr
    - to do output redirection '2>' or '2>>' symbol is used
    - command 2> file
      - older content of file will be over written
    - command 2>> file
      - content will be appneded into file at the end

Pipe

- Using pipe, we can redirect output of any command to the input of any other command.
- Two processes are connected using pipe operator (|).

- Two processes runs simultaneously and are automatically rescheduled as data flows between them.
- If you don't use pipes, you must use several steps to do single task.
- command1 | command2
    - output of command1 will be given as input to command 2
- E.g.
    - who | wc

Shell meta characters

- '*' - zero or more occurances of any character
- '?' - one occurance of any character

File Permissions/Mode

- File permission types: r (read), w (write), x (execute)
- Permission levels: u (user), g (group), o (other)
- File mode: u-rwx g-rwx o-rwx
- Example: hello.txt -- user can read/write and execute, group can read and execute, other can only read.
    - hello.txt mode: u=rwx g=r-x o=r--
        - rwx r-x r--
        - 111 101 100 = 754
    - chmod 754 hello.txt
- Decimal and Binary
    - 000 -- 0
    - 001 -- 1
    - 010 -- 2
    - 011 -- 3
    - 100 -- 4
    - 101 -- 5
    - 110 -- 6
    - 111 -- 7
- File mode/permissions
    - read=4, write=2, execute=1
    - user -- rwx -- 4+2+1 = 7
    - group -- r-x -- 4+1 = 5
    - other -- r-- -- 4 = 4
- File mode can be changed by the file owner (or superuser).
    - rw- rw- r--
    - 110 110 100

    - 6 6 4
    - chmod 664 f1.txt
    - rw- rw- r--
    - 110 110 100
    - 111 100 100
    - 7 4 4
    - chmod 744 f1.txt
- chmod +/- r/w/x file
    - to add or remove permissions of file
- chmod u/g/o +/- r/w/x
    - to add or remove permissions of specific level

Directory permissions/Mode

- From end user perspective, directory is a container that contains subdirectories and files.
- From kernel perspective, directory is a special file (d) that contains directory entries for each subdirectory and file.
- Directory entry = File/Subdir name + Inode number
- Default directory permissions: rwx rwx r-x
- Dir read permission: Can list dir contents (i.e. can read dir entries)
- Dir write permission: Can add, delete or modify dir entries i.e. Create new subdir/file in it, Can delete subdir/file in it, Can rename subdir/file in it.
- Dir execute permission: Can changed to that directory (cd command).

Kernel

- The core OS that does basic minimal functionalities of the kernel is referred as "kernel".
- These basic functionalities are
    - CPU scheduling
    - Process Management
    - Memory Management
    - File & IO Management
    - Hardware abstraction layer
- Kernel has four types (as per architecture)
    - Monolithic kernel
    - Micro kernel
    - Modular kernel
    - Hybrid kernel

Monolithic kernel

- All source files are compiled into a single binary kernel image.
- This is kernel is very fast (all components in the same file).
- However if any component fails, whole OS (kernel) crash.
- Modifying any component needs to rebuild whole kernel and redeploy it.
- e.g. Linux (vmlinuz), UNIX.

Microlithic kernel

- Kernel is very small including minimal functionalities. So referred as micro-kernel.
- All additional functionalities are implemented as independent processes (called as servers).
- All these servers and kernel communicate with each other by message passing. This reduce performance.
- If any component fails, the rest of system can continue to function. So this kernel is robust.
- Modifying any component needs to rebuild and redeploy only that component.
- e.g. Symbian.

Modular kernel

- Modular kernel is small. Additional functionalities are implemented as dynamically loadable modules (.sys/.ko).
- These modules are loaded into the kernel process at runtime and execute.
- This kernel execute faster, as all modules are part of same kernel process.
- If any component fails, whole OS (kernel) crash.
- Modifying any component needs to rebuild and redeploy only that component.
- e.g. Windows (ntoskrnl.exe)

Hybrid kernel

- This kernel is made by combining source code of multiple kernels.
- e.g. Mac OS X kernel is made up of MACH kernel and BSD UNIX kernel.

Linux kernel

- Linux kernel has static and dynamic components.
- Static components are
  - Scheduler
  - Process management
  - Memory management
  - IO subsystem (core)
  - System calls
- Dynamic components are
  - File systems (like ext3, ext4, xfs)
  - Device drivers
- Static components are compiled into the kernel binary image. They are kernel components. The kernel image is /boot/vmlinuz.
- Dynamic components are compiled into kernel objects (*.ko files). They are non-kernel components. They are located in /lib/modules/kernel-version.