Week 1 Notes

Terminal Emulators

- Terminal
- Konsole
- xterm
- guake

Command Prompt

- `username@hostname:~$`
  - `~$` is the path

Commands and Flags

- `uname`
  - prints the name, version and other details about the current machine and the operating system running on it
  - the -a displays hidden files that have a dot in front of them
- `pwd`
  - Present Working Directory
- `ls` `-a` : all . displays hidden files `-l` : use a long listing format `-i` : print index number of each file (inode) `-s` : shows blocks occupied by each file `-1` : each file name on a separate line
  - output of `ls -l` : `drwxr-xr-x 5 ckg ckg 12288 Nov 25 10:00 Documents` (`d` is file type ; `rwxr-xr-x` owner,group,others permissions ; `5` no of hard links ; `ckg` is owner ; `ckg` is group ; last modified time stamp ; filename)
  - `ls F*` gives a list of all files starting with F
- `rm`
  - remove a file
  - `rm -i` prompts before every removal (it can be set using `alias rm="rm -i"`)
  - works only with write permission
  - use `-d` for removing directories
  - `rm -r mydirectory`
- `mv`
  - move , rename
  - `mv file1 ..` (moves file to parent dir)
  - `mv file1 file1a` (renames file1 to file1a)
- `ps`
  - currently running processes
- `clear`
  - or ctrl+l
- `exit`

    - or ctrl+d
- `man`
    - get help on any command in linux. eg : man ls
    - man sections (1 to 9) eg : man 1 ls
    - 1 - Executable programs or shell commands
    - 2 - System calls provided by Kernel
    - 3 - Library calls
    - 4 - Special files usually found in /dev
    - 5 - File formats and conversions
    - 6 - Games
    - 7 - Misc : macro packages and conventions
    - 8 - System admin commands
    - 9 - Kernel routines
- `cd`
    - change directory eg cd .. - goes to parent directory
    - `cd` without any arguments will take you to the home directory
    - `cd /` takes you to the root folder
    - `cd -` takes you to previous directory
    - `cd ~` takes you to home directory
- `cp`
    - copy command : cp file1 file2
- `date`
    - date and time
    - `date -R` gives in RFC 5322 standard (used for email communications)
- `cal`
    - calendar of a month
    - eg : `cal aug 1947`
    - `ncal` gives calendar in flipped orientation
- `free`
    - memory statistics
    - use `h` flag to make it human readable
- `groups`
    - groups to which a user belongs
- `file`
    - what type of file
    - `-f` allows you to pass a file in which file names are separated by lines (ls -1 > files.txt; file -f files.txt)
    - `file *` will give a list of file name and types directly
- `mkdir`
    - create a directory
    - default permissions (umask)
- `touch`
    - used to change the last modified timestamp of a file

    - also used to create empty files
- `chmod`
    - `chmod 777 file.txt`
    - `chmod g-w file.txt` (removes write permissions from the group)
    - `chmod o-x file.txt` (removes executable permission from others)
    - `chmod u-r file.txt` (removes read permission from owner)
- `whoami`
    - prints username
- `less`
    - allows you to read a file page by page
- `ln`
    - used to create a hard link or a symbolic link (symlink) to an existing file or directory
    - `s` flag is used to create a soft link
    - usage : `ln file1 file2 ; ln -s file1 file2`
- `cat`
    - stands for concatinate
    - allows you to view the contents of a single file or multiple files (gets concatinated)

File types

- output of `ls -l` : `drwxrwxrwx or lr-x--x--x` (l indicates symbolic link and d indicates directory)
- `-` Regular file
- `d` Directory
- `l` Symbolic link
- `c` Character file (usually found in /dev ; typically the terminal)
- `b` Block file (usually found in /dev ; typically the hard disk)
- `s` Socket file
- `p` named pipe

Viewing and Adding to files

- `cat` - to view the contents of a file
- writing to a file : `>` eg : `echo "Hello world" > test.txt`
- appending to a file : `>>` eg : `echo "Helo world" >> test.txt`

Hard links and Soft links

- inode - An entry in the filesystem table about the location in the storage media
- hard link points to the same inode
- soft link points to a hard link
- hard link must be on the same partition while soft link can point to a file at a totally different geographical location.
- inode is metadata for the file . eg : size ,permissions,blocks etc.

- `ls -i <name>`
- `ln` and `ln -s` is used for creating hard links and soft links
- inode is unique for every file : if there are multiple entries of inode then it means that they are all hard links
  - if there is a dir level1 with inode = 18874686
  - when you cd into that dir . will also have inode = 18874686
  - if i make a dir level2 inside level1 and then cd into level2 .. will have inode = 18874686 (no of hard links will increase by 1)
  - as number of sub directories increases the number of hardlinks also keeps increasing
- users cannot create hard links for directories (level1 to level2 and level2 to level1 will create a back and forth)

Permissions

- Files and directories do not inherit the parent directory permissions
- `rwxrwxrwx` (777)
  - 7 rwx
  - 6 rw-
  - 5 r-x
  - 4 r--
  - 3 -wx
  - 2 -w-
  - 1 --x
- rwx rwx rwx : Owner Group Others
- only owners can change permissions of a file
- Execute permission is required on a directory to cd into it (Even ls and tocuh to a dir will not work)
- If you want to access a file, all its parent direcories should have **x** permission. This works even without r and w permissions if you know the path.
- r and w permissions along with x is required to ls a directory or touch a file into a directory
- Removing a file works only if it has write permission

Linux Virtual Machine

ISO

- image of Linux OS (Ubuntu 20.04 LTS for x86_64 platform)

Hypervisor

- (eg: Oracle VirtualBox or VMWare Workstation Player)
- A Hypervisor creates and runs virtual machines
- It allows running multiple operating systems while sharing hardware resources

Command Line Environments

- Cloud - replit and cocalc
- Phone - Termux by Fredrick Fornwall

File System of Linux OS

- Filesystem Hirearchy Standard FHS 3.0 (June 03, 2015) (refspecs.linuxfoundation.org/fhs.shtml)
- `/` is root directory and field separator or delimiter for sub-directories
- `.` references the current directory (. is a special file in every directory)
- `..` references the parent directory (.. is a special file in every directory)
- Path for traversal can be absolute or relative
- `boot` directory is where the kernel is located
- `/usr/bin` contains commands that we will use
- `/bin` - essential command binaries
- `/boot` static files of the bootloader
- `/dev` device files (different character in long format of file listing 'c' instead of 'l' or 'd'. 'c' indicates character file - means you can read from it character by character. if first character is 'b' they are block devices typically hdds - the block devices are made available as files.)
- `/etc` Host specific system configuration (.conf files)
- `/lib` Essential shared libraries and kernel modules (Typically contain files with version number at the end)
- `/media` mount points for removable devices
- `/mnt` mount points
- `/opt` add on application software packages
- `/run` Data relevant to running processes
- `/sbin` essential system binaries
- `/srv` data for services
- `/tmp` temporary files (normally flushed when system is rebooted)
- `/usr` secondary hierarchy
  - `/usr/bin` : user commands
  - `/usr/lib` : libraries
  - `/usr/local` : local hierarchy
  - `/usr/sbin` : non vital system binaries
  - `/usr/share` : architecture dependent data
  - `/usr/include` : header files included by c programs
  - `/usr/src` : source code
- `/var` variable data (`/var/log` contains logs for various services)
  - `/var/cache` : Application cache data
  - `/var/lib` : Variable state informtion
  - `/var/local` : variable data for /usr/local
  - `/var/lock` : lock files
  - `/var/log` : log files and directories
  - `/var/run` : data relevant to running processes
  - `/var/tmp` : temporary files preserved between reboots

	Shareable	Unsharable
static	`/usr` and `/opt`	`/etc` and `/boot`
variable	`/var/mail`	`/var/run` and `/var/lock`