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

Types of Command execution

- Synchronous execution
  - Shell waits for command to be completed.
  - Default execution
- Asynchronous execution
  - Shell doesn't wait for command to be completed.
  - To execute a command asynchronously, use "&" at the end of the command.
  - terminal> gedit &

locate

- locate hello.c
- locate *.txt

zip and unzip

- zip file.zip filename.txt
  - -d - removes file from zip
  - -u - updates file to zip
  - -r - zip directory recursively
- unzip file.zip

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

chmod

- chmod permissions filename

- Permissions:
  - +x/r/w - add permission to all
  - -x/r/w - remove permission of all
  - u+x/r/w - add permission to user
  - u-x/r/w - remove permission of user
  - g+x/r/w - add permission to group
  - g-x/r/w - remove permission of group
  - o+x/r/w - add permission to others
  - o-x/r/w - remove permission of others
  - 0774 - apply rwx to user, rwx to group and r to others

chown

- sudo chown username filename
- sudo chown :groupname filename
- sudo chown username:groupname filename

adduser

- sudo adduser username

passwd

- sudo passwd username

deluser

- sudo deluser username
  - home directory is not removed
- sudo deluser -r username
- sudo deluser --remove-home username

Links

- Links are shortcuts to access deeply located files quickly.
- There are two types of links in Linux/UNIX.
  - 1. Symbolic Link
  - 2. Hard Link

Symbolic Link

- terminal> ln -s /path/of/target/file linkpath
- Internally use symlink() syscall.
  - man symlink
  - int symlink(const char *target_path, const char *link_path);

symlink() syscall

- A new link file is created (new inode and new data block is allocated), which contains info about the
  - target file (absolute or relative path).

- Link count is not incremented.
- If target file is deleted, the link becomes useless.
- Can create symlinks for directories also.

Hard Link

- terminal> ln targetfilepath linkfilepath
- Internally use link() syscall.
    - man link
    - int link(const char *target_path, const char *link_path);

link() syscall

- A new directory entry is created, which has a new name and same inode number. No new file (inode and data blocks) is created.
- Link count in the inode of the file is incremented.
- If directory entry of target file is deleted (rm command), file can be still accessed by link directory entry.
- Cannot create hard link for directories, because it may lead to infinite recursion (while traversing directories recursively e.g. ls -R)

Directory permissions

- r (read): Can read directory entries.
    - ls command can list the directory.
    - readdir() can get directory entry.
- w (write): Can add new directory entry, modify existing directory entry, or delete directory entry.
    - new directory entry: when create new sub-directory (e.g. mkdir) or file (e.g. touch, ...).
    - modify existing directory entry: rename file or sub-directory.
    - delete directory entry: delete file or sub-directory (e.g. rm, rmdir, ...)
- x (execute): Can browse the directory
    - cd command (chdir() syscall) will work

VI Editor

- sudo apt-get install vim
- VI editor works in two modes
    - command mode
    - insert mode
- press i - to go into insert mode
- press Esc - to go into command node
- VI editor commands:
    - :w - write/save into file
    - :q - quit vi editor

- :wq - save and quit
- :y - to copy current line
- :ny - to copy nth line
- yy - to copy current line
- nyy - copy n lines from current line
- :m,ny - copy fomr mth line to nth line
- :d - to cut current line
- :nd - to cut nth line
- dd - to cut current line
- ndd - cut n lines from current line
- :m,nd - cut fomr mth line to nth line
- press p - to paste copied line on next line of current line
- yw - copy from current position upto next word
- yiw - copy current word
- y$ - copy from cursor position upto end of line
- y^ - copy from cursor position upto start of line
- vim -o - to open multiple files horizontally
- vim -O - to open multiple files vertically
- ctrl + ww - to go into next file
- /pattern - to search the pattern
- n - to go on next occurance
- 😒/pattern1/pattern2/ - find and replace only first occurance of current line
- 😒/pattern1/pattern2/g - find and replace all occurances of current line
- :%s/pattern1/pattern2/g - find and replace all occurances of file

- To do setting in vi editor
  - create file into home directory as below:

```
vim ~/.vimrc
```

- add below content into it

```
set number
set autoindent
set tabstop=4
set nowrap
```

Shell Scripting/Programming

- Shell Script - set of commands with some programming constructs, written for shell
- commands - shell commands
- Programming construct - control flow structures, functions, arrays
    - Decision control structure (if else)
    - Selection control structure (case)
    - Loop control structure (while, until, for)
- Shell scripts are interpreted by shell

Applications

- Shell scripts are used by adminstration people for installations, setups, automations
- Administrative tasks e.g. User creations, Troubleshooting, Router/Firewall config on commandline, etc.
- Automation of tasks e.g. Cron jobs, Scripts for building softwares, etc.
- Application installers

Limitations

- Slow execution -- Interpreted. Cannot be used for performance centric applications.

Proceedure to write and execute scripts

```
* Step 1 - create file with extension .sh
    * $ vim script.sh
* Step 2 - execute the script
    * Option 1
        * $ bash script.sh

    * Option 2
        * $ chmod +x script.sh
        * $ ./script.sh
```