**LINUX**

Linux is not an operation system. It consist of 3 layers

1.User

2.Kernel

3.Hardware

* LINUX IS A COMMAND LINE INTERFACE

**Basic LINUX Commands**

$ touch <file name> // file creation

$ mkdir <folder name> //folder creation

$pwd //present working directory

$ls //listing of files

$ls -a //listing of .files

$ll //see the file permision

$uname //kernal name

$cat /etc/os-release //log files

$ cd , cd .. //changing directories forward and backward

**Identify normal user or root user**

~ 🡪 home directory of user

$ 🡪normal user

# 🡪root user

command for login or switch into root user

$sudo su //sudo-Super User Do

**Cat commands**

cat {view the content from file}

cat <source file> > <destination file> //override(erase the pervious data) only d.f

cat <source file> >> <destination file> //appending s.f + d.f

cp <source file> <destination file>

mv <source file> <destination file>

**grep**

eg: $grep “word” <filename>

**Linux EDITORS**

Nano ---------------->$sudo nano <filename>

VI -----------------> $sudo vi <filename>

**Utility Commands**

$head <filename> //print first 10lines or 5paragraphs

$tail <filename> //print last 10lines or 5paragraphs

$more <filename> //scroll down content

$less <filename> //scroll up and down

**File Management**

Absolute method:

Read – r -------> 4

Write – w ------->2

Execute – x ------>1

eg :$chmod 600 java.txt (user, group, others)

**Processmanagement Service ctive/down**

$sudo systemctl status

$sudo systemctl start

$sudo systemctl restart

**GIT**

Version control system

Working area(Red color) --> Staging area(Green color) --> Local area

$git config --global user.email <email address>

$git config --global user.name<user name>

$git clone <url>

$cd <repo folder name>

$git add <filename>

Commit: The location where code and its changes are stored or The changes made in repository will be saved as a new commit with a new commit id.

$git commit -m ‘’<commit message>”

$git push -u origin main

Branch: Branch is a representation of different isolated versions of code.

$git branch <branchname>

$git checkout <branchname>

Merge: A technique that is used to include the changes from one branch to other branch.

(new commit name and commit id are created)---> be on the branch that you want to merge into.

$git merge <branch name>

Rebase: Rebase is similar to git merge, but logs are modified in this technique.

(no change in commit name but commit id will get override)

$git rebase <branch name>

* Git merge and rebase commands are used to combine the work of multiple developers in one code(End objective of both the commands is same).

**Stash**:

Git stash temporarily saves your data safely without committing so you can work on something else and re-apply them later on.

$git stash

$git stash pop

$git status

$git stash list

**Fetch**:

Git fetch as similar behaviour to git pull but git fetch is used to download the changes from a remote repository but does not automatically merge them with the local repository.