How do you list the files and directories ?
ls - list the files and directories
ls -l -- to list the files and directories in long format
ls -lt -- to list the files and directories in long format with time
ls -lrt -- To list the files and directories in long format with reverse of time
pwd -- Present working directory
touch -- is used to create empty file
mkdir -- is used to create a directory
cd -- is used to change directory
cd .. -- to come out of present directory
cd ../.. -- if you want to come out of 2 or more directories
cd - acts as recall between last 2 commands
ls -a -- to list the hidden files and directories
ls -al -- to list the hidden files and directories in long format
cat -- used to display the content of a file on the screen
vi -- is used to edit a file via vi editor

```
        -- Esc i -- used to insert mode in vi editor
        -- Esc:wq! -- used to save and quit the file --- w - write q - quit ! - forcefully
        -- Esc:q! -- quit without saving the content
        -- Esc:set number -- which sets the number for a file
        -- Esc:setnonumber -- which removes the number for a file
        Replace a string in the editor
        syntax: Esc:%s/string1/string2/g
                - s stands for substitute and g stands for globally
        Examples : Esc:1s/string1/string2/g -- changes in 1st line only
        Esc:1,5 s/string1/string2/g -- changes from 1st to 5th line
        Esc: 7,$ s/string1/string2/g -- changes from 7th line end of file
        esc: 1 s/string1/string2/1 -- changes in 1s line 1st occurance only
```

-- Search a pattern /string
Esc :/ pattern -- it will move to 1st occurance then keep pressing n to 2nd and so on

-- How do you delete the whole line in editor
Esc dd

Cut and paste in a file
command : cut -d -f(colname) filename
d - delimeter
example
cut -d- -f2-5 f1.txt
paste -- paste displays lines of multiple files line by line
command:
paste f1.txt f2.txt

cp command (Copy):
It is used to copy content of a file from one to another
If destination file doesn't exist then i will create and copy it . If it is alreasy exist then it will override the content.
syntax: cp file1.txt file2.txt

```
cp -r dir1 dir2
```

mv command (Move):
The source will get deleted and moved to destination. this also can be called as rename

```
mv file1 fil2
mv file dir
mv *.txt dir/
```

chmod command :
Used to change the permission of a file or directory
User Group Others
f or d or l -- file or directory or link
r - read
w - write
x- execute
rwx- users
2nd set is knows as groups
3rd set is knows as others
r - 4   user read 4 write 2 execute 1 4+2+1=7
w- 2   read -4 execute - 1 4+1=5
x- 1   read=4 4

```
chmod 754 f1.txt
chmod -R 777 dir/
```

Remove or add permission
chmod u+w filename -- this sets write permission for the user
chmod g+rwx,o+rwx -- this sets all permission for group and others
chmod g-w,o-w file -- removing write permission for group and others
chmod a+rwx file -- adding all th permission for all user groups and others

Umask : It used to set the default permission on a system
eg : umask 000 dir name
umask 777 file name

Chown command:
Changing the ownership of a file or directory

syntax: chown -R newowner dir
chown newowner filename
chgrp (changing group):
syntax: chgrp new groupname filename

Grep : It is used to search a pattern in a file
syntax : grep "pattern" filename
eg: grep "linux" filename -- it will bring all the lines which consist of linux word
-- grep -i "Linux" filename -- To bring all the lines which consist of linux word case insensitively
-i == case insensitve

grep -e "pattern" -e "pattern" filename --- to search multiple patern in a file
or
egrep can be used instead of -e everywhere
grep -w "linux" filename --- to search only a perticular word
grep -v "linux" filename --- To print the lines except the lines which have a linux word
grep -c : to count the lines if the pattern is present
        eg : grep -c "linux" f1
grep -l * : it will print all the lines which have word linux from all the files
eg : grep -l "linux" *
grep -R -l "linux" -- to search the pattern in subdirectories
grep "^w" filename : to print all the lines that start with a perticular word
grep "S$" filename : to print all the line ends with a perticular letter or word
-- f we want to print line number and patter in the file
grep -i -n "linux" file

sed command :
 It is used to edit the file without opening it, used to replace the string in a file
 command:
 sed 's/pattern1/pattern2/g' filename
 sed '1 s/pattern1/pattern2/g' filename -- to change the patter in 1 st line
 sed '3,5 s/pattern1/pattern2/g' filename -- to change the pattern from 3 to 5th line
 sed '5,$ s/pattern1/pattern2/g' filename -- to change the pattern from 5th line to end
if you need to change the original file then use -i= insert to a file / modify to a file.
sed -i 's/pattern1/pattern2/g' filename
Note : if you dont use -i it wont affect the original file it will just print the changed output
sed -n '5p' filename it will print 5 th line a file
to print from 4th to 99th line
sed -n '4,99p' filename
to print from 5th lin to end of file
sed -n '5,$p' filename
to delete 5th line in a file
sed '5d' filename

sed -i '5d' filename -- to delete in original file as well

Awk command: is used to cut the content of a file columnwise
syntax: awk -F " " '{print $2,$3}' filename to print 2nd and 3rd cloumn
awk -F " " '{print $NF}' filename is used to print last column
awk -F " " '{print $NF,$(NF-1)}' filename to print last 2 cloumn
awk -F " " '{print $(NF-1)}' 2nd last colname
----

rm command :
Used to remove files or directories
syntax: rm filename
rm -f filename -- to delete the file forcefully
rm -rf dir name -- to delete the directory recursively and forcefully

wc command : It is used to count the number of lines , words and charachters in file
syntax: wc filename
wc -l filename -- it will count all the lines
wc -w filename -- It will count all the words in a file
wc -c filename -- it will count all the charachters in a file

Pipe (|) :
It is used to pass output of one command to next command
    eg cat filename | wc -c

Echo Command :

it is used to print the statement on the terminal
eg : echo "Hi"
echo -e "Hi \nHow are you" --- it will print in 2 lines inorder to print in next line we need to use -e and \n both

Redirect and Append command
Redirect (>) used to write the output of a command to a file, if the file not exist it will create new file and write it, If already present it will override

eg : echo "Hi How are you " > file.txt
ls -lrt > file.txt
du -sh * > file.txt
Append(>>) : It is used to attach the content or output of command to a file at the end, it will not override the existing content of a file
eg: echo "Pradeep" >> file.txt

Head and Tail command

head is used to display 1st pat of a file by head filename -- by default it displays 10 lines

eg :head filename

head -15 filename --- it will display first 15 lines

Tail command : tail is used to display last portion of a file

```
tail -2 filename
tail filename
tail -1 filename
```

```
ls -lrt | tail -1
head -99 filename | tail -1 -- it wil print the 99th line of a fle
head -55 filename | tail -1 -- it will prnt the 55th lne of a file
head -5 filename | tail -3 -- it wil print 2 to 5 lines
```

More and less command:

it is used to display content of the file by more filename , we can scroll down(ENTER) but we cant scroll up usig more command

syn: more filename

less command : it is used to display content of the file, we can scroll up and down and use esc qq to exit

Find Command :

It is used to find the location of a file/dir . find is automatic recursive , it searches in a subdirectories automatically

syntax : find . -name "filename"

examples:

Q) how do you find the files that ends with .log from your pwd

```
find . -type f -name "*.log"
```

Q) find or list the directories that has log from the home location

```
find /home -type d -name "log"
```

Q) Find log directories everywhere in the filesystem with case insensitively

```
find / -type d -iname "log"
```

Examples :

```
find . -type f -iname "*.log"
find . -type d -iname "dir name"
```

To list the files/dir that are modified 90days ago

```
find . -type f -mtime +90
find . -type d -mtime +90
```

How to find the files/ dir that are modified withing 90 day

```
find . -type f -mtime -90
find . -type d -mtime -90
```

how do you list both files and directories that are modified 90 days ago

```
find . -mtime +90
```

how do you find the files/dir that are modified 60 min ago

```
find . -type f -mmin +60
find . -type d -mmin +60
```

How do you find the files/dir that are modified within 30 min

```
find . -type f -mmin -30
find . -type d -mmin -30
```

How do you list all the empty files

```
find . -type f -empty
find . -type d -empty -- lst all the empty dir
find . -empty
find . -type L -empty --- h or L is used to find the links
```

How do you find the files or directories that has permission 777

```
find . -type f -perm 0777
find . -type d -perm 0777
find . -perm 0777
```

how do you list all the non empty files

```
find . -type f ! -empty
```

Max Depth command :

How do you restric recursive search for the dir search in find command

```
find . -maxdepth 2 -iname "test"
```

xargs

It is used to pass the args to next command

Output of one command will be passing as args to next command

Q) how do you delete the files that are modified 3 months back ?

```
find . -type -f -mtime +90 | xargs rm
```

Note : Arguement - Arranging words in horizontal row or in a series in its spaces

To Check the size of a file or dir

```
du -sh filename
```

du --- disk usage

s -- size

h --- human readable

```
du -sh -- to list all the file size
du -sh * to list all the files size in a dir
df -h . --- to check the present disk space
df -h -- to check all the disks
```

-- To check the linux version

```
uname -a -- all info
```

uname -- just the version

to check how many users logged into system
who

-- whoami -- it will display who logged in to system

ps -- it will show all the processes that are running in the system

ps -ef | grep git -- it will check whether git process is running or not
to check the perticular process or application running or not -- ps -ef | grep "process name "
kill -9 process id --- to kill the process forcefully
Note : dont kill system related process

htop and top :- to check which proces is taking high cpu usage or memory
free -m -- to check the free memory or RAM

Links : There are two typs lins 1. Soft links and 2. hard links
-- Soft links or Symbolic links or synm links : I is a shortcut to a file, if i modfy actual file it will get reflected in a link also and if i delete the original file or actual file soft link will not work
syn: ln -s filename linkname

Hard Links: it is also a shortcut to a file, if i delete the original file hard link will work because it points to inode of file (inode is a unique identification number which indicates memory address of a file)
syn : ln filename linkname

how do add user in linux
command useradd <username> -- to add user
groupadd <groupname> -- to add group
useradd -g <groupname> <username> -- to create a user and add it to existing group
useradd -d <directory> wasim
userdel <username> --- to delete user
groupdel <groupname> --- to delete a group
passwd <username> -- to set the password to user

su - (superuser/switch user) -- it is used to login as other user (normally login as root user)
sudo -- is used execute the commands with root/ admin privilege
how to login as a root

commands

```
sudo su
sudo bash
su -
su root
```

ssh ---
it used to login to remote server
syntax: ssh user@server2 -- to login from server 1 to server2

To enable passwrd authentication / public key authentication follow the steps

--- vi /etc/ssh/sshd_config
uncomment or chnage it to yes the followng line
PubkeyAuthentication yes
uncomment or chnage it to yes --- PasswordAuthentication yes
save it -- wq!
restart ssh service --- systemctl restart sshd

scp --- is used to trasfer files from one server to the other server
syntax:
scp filename username@server2:path

rsync: it is used to copy files from one server to anoter and also within the server while copying the data if copy is stopped dueto network issue in between, if i use scp it will start copying from begining and if i use rsync u will start copying from where it is stopped
syntax : rsync filename user@server2:path

Note: Itwill sync to the destnation from source it is not reversible
To reverse we have to login to server 2 and give the same command

Q) What is the difference between rsync and scp
while copying the files from one server to another server, if copying stopped in between due to network problem, once the system is back online if you re issue the command scp will start from the begining ut rsync will start copying from where it stopped

Ping: It is used to check whether the server is up and running
syntax : ping <serverip or address>
telnet : it is used t authenticate remote server account credentials
used to break firewall
it uses port 23
ssh uses the port 22

```
telnet <ip address> <port number >
```

tee : it is used to write the output of command to a file as well as displace on the console
syn:

```
cat <filename> | tee log1
```

ports
http:80 /8080
https:443 /8443
telnet:23
smtp:25
ssh:22
rsync:22
sftp:22

top: To check which process is taking high cpu usage or memory

tree : To show tree structure of file and direction

Netstat: To find all the network ports

Q) What are the options in netstat
netstat -a : show all the listening and non listening sockets with the interface option
syntax : netstat -a | more

netstat -at : To list all tcp ports
netstat -nptl : to check the service ports
netstat -au : to list all the udp ports
netstat -l : to list only the listening ports
netstat -lt : to list only the listening tcp pots
netstat -lu : to list only the listening udp ports
netstat -s : to list all the statistic of all the ports
netstat -st : to list all the statistic of tcp prts
netstat -pt : to display PIC nd progrm name
netstat -c : To print all the netstat info continuation
netstat --verbos : to get the non supportive address families in the system
netstat -r : to gt to know th kernel routing information
netstat -ap | grep ssh : To get the port on which a program is running
netstat -an |greo '80' : to get the process which is using given port
netstat -i : to get the list of network intefaces
netstat -ie : to display extended information on the interface

ifconfig : to see the ipaddress
hostname -i : ipaddress (private)
hostname : Domain name and ip address
tracert(trace route): This utility is used to see the exact path the data packets is tking on the way to destination
nslookup : soource ip and info of url

How do you setup a passwordless connection between servers?
we need to run ssh-keygen command on server1 take the public key go to server2 paste the key in authorized_keys file in .ssh folder, next time we will login it will be a passwordless connection

Very Imp:
.bashrc is a executable for interactive non login shells that means if you have already logged into your computer open a new window then .bashrc s executed before windows command promt
.bash profile is executed for login shells that means when login via console bash profile is executed to configure your shell before the initial command promt

----

Git
--- What is version control system ?
Used to track history or version of files and directories
--- Different types of version control system
Git
svn
clearcase
TFS
CVS
amazon code commit
bitbucket

Steps to create git folder --
Create a folder- mkdir git-repo
cd git-repo
ls -al -- check .git is present
git init -- if .git is not present you need to initialise git repo
if git command not found error then instal git
yum install git -y
once install do git init and ls -al and check .git folder appears then

add some files

□ Git status (used to check whether files are workspace, staging area or in git repo
□ Git add test1 (this will move file from workspace to staging area)
□ Git status (it will shows changes to be committed)
□ Git commit –m "message" (this will move files from staging area to git repo)
□ Git status (this will show working directory clean)
□ If It wont work use for 1st time to configure,
o Git config –global user-email p@gmail.com
o Git config –global user-name ―p‖
□ Git commit –m ― ―
□ Git status
□ Git log (used to check the repo history)
□ Git log filename (specified filename)
□ Git log -2 (last 2 commits)
□ Git log -2 filename (last 2 commits of the file)
□ Git checkout commitId (used to switch to a previous version , used to switch to branch, also to switch to tags)
□ Git checkout master (gives to the latest version)

Tags
□ Tag is a name given to set of versions of files and directories. It indicates milestones of a project we can easily remember tags in the future. If I want good code in the future we tag it.
□ Command to list tags is,
o Git tag
□ To create a tag,
o Git tag tagname (it tags the latest version of code by default)
Branching
□ Is a parallel development, two teams will work on same peace of code on two different branches, later they can integrate the changes by merging.
□ Git branch – to list the branches
□ Git branch branchname –to list branchnames
□ If ‗*‘ is on the branch , when I run git branch branchname, branch will get created from the branch which has a ‗*‘.
□ To switch to a branchname
o Git checkout branchname
□ Git merge
o Is used to integrate two branches
□ Git merge branchname
□ Merging Conflicts

o Will occur when the same peace of code is changed on 2
different branches, when we try to merge those two brnaches,
then merging conflict will occur,
o To resolve this issue, I don‘t know whose change should I take
to merge , so I contact developers changes the code, person
who modified code of branch1 and branch2. Then they will
decide and tell us whose changes should I take into merge.
o Then I take that change and I commit it. I get to know who
modified the code on branch1 and branch2 using git log
command.
□ to delete branch or tag
o git branch –d branchname
o git tag –d tagname
□ used to create branch2 and automatically checkout to that branch
o git checkout –b branch2
□ how do you create a branch from tag?
o You need to checkout to that tag and create a branch using git

Assignment :
Checkout to master, include 2 files big3.c and fact.c
o Then create a tag, inside tag create a branch1,inside that create
palindrome.c then create a tag called release1 then in tat
include reverse.c then again tag as release 2, then merge it to
master with a tag name as release2.1.
o In the master again create a branch2, in tat include even.c and
prime.c program and create a tag for those and merge it to
master,then use tag as release 3.1 in master.

Difference between merge and rebase
Merge
-- it will merge latest content of 2 branches
-- simple and easy to understand merging concepts
-- maintain original context as source code

Rebase
-- it will add the history also
-- unifies the line development by rewriting changes from source branch so that they appear as children
-- simplifies your history
Git revert --
-- Used to undo the committed chnages, history will not be removed, we can track the reverted

chnages in the git log
syntax: git revert Head
git revert commit id
Git reset
Used to undo the committed changes but hostory will be removed

There are 3 types of reset
-- soft reset -- it will move the files from git repo to staging area it reset to specified commit if and history will be removed
syntax: git reset --soft <commit id > or HEAD
mixed reset -- It is uded to move files from repo to workspace
syntax: git reset --mixed <commit id > or HEAD
--- hard reset
it is used ti reset the previous committed changes there will no clue that you have committed the changes, hard reset will remove files from git repo staging area abd from workspace
syntax: git reset --hard <HEAD or commit id >

Note : In interview if they ask difference between and revert and reset then always gives revert and hard reset

Note :
There are total 40 digits in the commit id, git copies only 12 digit when we edit something in branches and merge it to a branch it copies only chnages

git reflog --- it is used to fetch all the log of all the activities and we can fetch deleted branch, tag files from git reflog

How do you list only merge commit
git branch --merged
git branch --no --merged
Note : Never do rebase on a public branch

Git stash :
if am workng on one branch. if i get some crtical work which needs to be fixed on the other branch, i dont want ro commit changes n the other branch then before i switch to the other branch i need to stash on the current branch which store the local files on a temporary location (it will not store in workpace staging area)_ once i am back to the current branch after completing critical work i can bring back the changes using git stash pop

syntax: to stash -- git stash
to bring back -- git stash pop
to delete stash id -- git stash delete stash id

git stash -u -- to list the untracked files in stash
git stash apply -- to apply the changes in working directory as well as stashing area
Q) how do you delete all the stash list

git stash list -- to list all the stash
git stash drop -- to drop all the stash list

Git cherry-pick :
it is used to merge specific commit to a branch
it also used to pick a perticular change in a branch
syntax: git cherry-pick <commit-id>
note : if i want to commit 3 commits use
git cherry-pick commitid1 commitid2 commitid3

Git conflit:
It is same as merge conflict
Sometime you get merge conflict when merging or pulling froma branch
git will then tell you something like conflict (content) merge conflict in a fakefile
it also tell you to fix conflict and commit the changes

git bisect :
it is used to search to fnd the commit that introduced a bug
syntax: git bisect <subcommand> <options>
-- This command uses a binary search algorithm to find which commit in your project history introduced a bug
-- can be used to find the commit that changed any propertyof your project
git bisect start
git bisect bad
git bisect good
git besect reset

There are 2 types of repository in git
bare repo
Non bare repo

Bare repo
-It acts as a central-repo, we can only push and pull the changes to the repository.
—in base repository, you cant run any git operation on bare repo.

Non bare repo -- its a local repo we can modify the files and push to a central repo we can run all the git commands
syntax : to create a non bare repo-- git init
bare repo : It acts a central repository we can only push and pull the changes to the repository we cannot do any git operations here
syntax : git init --bare

git clone : clone is used to bring the remote repository to local workspace for the first time
We need to clone it where git repo should not be there or else it will show error
syntax: git clone user@serverip:remote repo path

git push :
It is used to push the local repo changes to remote repository
git push user@server:path of remote repo

git pull :
Pull will bring the changes from remote repo to and merge to local repo automatically
git pull user@server:path

git fetch :
fetch will bring the changes from remote repo and stores in a separate branch where you can reviw the changes and merge if required

syntax: git fetch user@server

Note : The branch that fetch data is stored is FETCH HEAD
Git pull = git fetch + git merge

 Any source code execution
o .c
o Preprocessing
o Compilation (.s)
o Assemebly conversion (.o)
o Linking and loading
o Executable files (.exe)
 Try this
o Mkdir central_repo
o Mkdir workspace1 workspace2

- Cd central_repo
- Git init --bare
- Cd ..
- Cd workspace1
- Git clone /./../central_repo/
- Cd ../..
- Vi big3.c
- Git add and commit
- Git push /./../central_repo/
- Cd ../..
- Cd workspace2
- Git clone /./../central_repo/
- Cd workspace2/central_repo
- Vi rev.c
- Git add and commit
- Git push /./../central_repo/
- Cd ../..
- Cd workspace1/central_repo
- Git pull /./../central_repo/
- Vi pali.c
- Git add and commit
- Git push /./../central_repo/
- Cd workspace2/central_repo/
- Git fetch /./../central_repo/
- Git checkout FETCH_HEAD
- Git checkout master
- Git merge FETCH_HEAD