**LINUX COMMANDS PRACTICE**

Whoami

uname -s kernel name

uname -r release

uname -v version

uname -n node name

uname -p processor

usage

uptime

uptime -p pretty format

uptime -s

\*useradd

Useradd roy -d /home/roy adding roy in /home

Useradd roy -e 2018-05-30 from that date acct will be disabled

Useradd roy -f 6 can login for 6 days to change his passwd

Useradd roy -g staff set roy as staff

Userdel

Userdel loginname delete user account

Userdel -r loginname del account with home dir and all files

Userdel -f loginname when logged in

Usermod have to check this

Sleep 5 , 2m , 3h , 5d

Exit or logout

Reboot -p

Reboot -f forcibly

Passwd

Passwd -S display passwd status

Passwd -Sa display for all accounts

Passwd -d roy removing passwd of user

Passwd -e expiry immediately

Passwd -l roy lock

Passwd -u roy unlock

Ls list files

Ls -l lists all files with permission and dates

Ls -a shows hidden files with .

Ls -lh sizes in human readable format

Ls -F add / at end of each dir

Ls -R long listing dir

Ls -r reverse order

Ls -I inode num

Ls -version

Cd /usr/roy changes cd to roy

cd - move one dir back

cd .. curr dir to par dir

cd -- show last working dir

cd ../../ move two dir up

cd ~ move to users home dir

chmod r - 4 , w - 2 , e - 1

chmod 777 1st 7 user 2nd 7 group 3rd 7 everybody

chown have to check

mkdir 1 2 3 creates 1,2,3 directories

mkdir -p roy/roy1 creates roy1 dir under roy dir

pwd

rmdir -rf roy delete dir if it has files and sub dir

rm -I file deletes interactively

echo have to check its use

id UID AND GID

id -u roy UID

id -g roy GID

vi editor have to check this

cp copy

cp 1 2 3 /users/roy

cp -r dir /users/roy copy dir recursive

touch 1 2 3 creates multiple files

touch - c test avoids creating new files

cat file file1 displays contents

cat > roy.txt redircts

cat >> roy.txt appends

cat file1 > file 2 copies content

tail file displays last 10 lines

less command have to check this

zip file1 file2

ls archive.zip

unzip archive.zip

zip -d archive.zip file1

gzip compresses and deltes old file

gzip -c file keeping original file

gunzip file.gz uncompress

gzip -r file.gz compress all files

gunzip -r file.gz uncompress

gzip -fast file.txt compress fastly

tar have to check

ln -s path of original file path of soft link file to create soft link

ln path of original file path of soft link file hard link

top - show processes sorted

when top is running press M for memory , R reverse , k kill

scp source file path destination file path send file from local to remote

wget - download

wget -c resume download

wget -b download file in back ground

curl <http://www.google.com> > google.html redirects web content to html file

du -a disk uage of all files

du -ah

du -ahc grand total

du -sh only total

df file system d

vmstat memory usage

vmstat -a active memory

free system memory

free -k

free -m

free -g

vmstat display memory usage

ifconfig assigned ip address

ifconfig -a active interface

netstat -a active internet connections

ping inc or dec time interval btw packets

ping -c N google.com send N packets

traceroute google.com shows ip address to web

ps - currently running process

ps -ef e display f format

kill -9 PID kills process

kill -9 PID1 PID2

chkconfig whether service is config for start

crontab used for scheduled jobs

lsof list of all files for active ps

awk print all lines of file

grep search for given string

find list all files in current and sub dir

uniq removes duplicate lines

wc line words wc

GIT

STEP BY STEP OF GIT

Git --version

git config --global user.name "roy4455"

git config --global user.email “[roypranay23@gmail.com](mailto:roypranay23@gmail.com)”

git init create local repo

git status

git add file or git add \*

git commit -m "Commit message"

git remote add origin <server> connect to remote repo

git push origin master

git -help for any help

Branch

create branch

checkout branch

merge branch to master

Git branch MyBranch

Git checkout MyBranch

Git push -u origin MyBranch

Git merge MyBranch

Git push -u origin master

git branch -d “branch name” delete from local

git push origin --delete “branch name” delete from remote

tags

git tag v1.0

git tag -a v1.1 -m "ver 1.1" (to create annotated tags)

Display or Show tags

git tag

git show v1.0

git tag -l “v1.\*”

Push tags to remote

git push origin v1.0

git push origin --tags

git push --tags (to push all tags at once)

Delete tags (if required only)

to delete tags from local : git tag -d v1.0

git tag --delete v1.0

to delete tags from remote :

git push origin -d v1.0

git push origin --delete v1.0

git push origin :v1.0

to delete multiple tags at once:

git tag -d v1.0 v1.1 (local)

git push origin -d v1.0 v1.1 (remote)