Operating System: An OS is system software that manages computer hardware and software resources, and provides common services for computer programs.

Types of OS :-

1. Desktop OS – Microsoft windows, macOS, linux such as ubuntu
2. Server OD – windows server, linux distributions like centOS, RHEL
3. Mobile OS – Android, iOS, Windows mobile
4. Embedded OS – Routers, smart TVs, automobiles, home appliances
5. Real-time OS(RTOS) – medical equipment, car ECUs, aerospace, defence, network firewalls, home security systems

Linux is an open source OS that manages computer hardware and software resources.

Developed by linus Torvalds

Linux Distributions

1. Ubuntu
2. Fedora
3. Debian
4. RHEL
5. CentOS
6. Arch Linux
7. openSUSE
8. linux mint
9. gentoo linux
10. slackware
11. alpine linux
12. kali linux

Install Oracle Virtualbox to use linux systems - <https://www.virtualbox.org/>

Filesystem – used by an OS to manage files. The system controls how data is saved or retrieved.

/boot – contains file that is used by the boot loader

/root – root user home directory

/dev – system devices(eg. Disk,cdrom,speakers etc)

/etc – configuration files

/bin - /usr/bin – everyday user commands

/sbin - /usr/sbin – system/filesystem commands

/opt – optional add on applications(not part of OS apps)

/proc – running processes(only exist in memory)

/lib - /usr/lib – C programming lib files needed by commands and apps

/tmp - directory for temporary files

/home – directory for user

/var – system logs

/run – system daemons that start very early to store temporary runtime files like PID files

/mnt – to mount external filesystem

/media – for cdrom mounts

cd – change directory

pwd – print working directory

ls – list

whoami

ls -l

cd ..

su - -> become root user

ls -lrt

ls -ld – to list only the directories

**Type of files**

drwx-xr-x – directory

lrwxrwxrwx – link

-rw-w-r-- - file

- ->regular file

D -> directory

l –> link

c –> special file or device file

p –> named pipe

b –> block device

change password – passwd

FILE SYSTEM PATHS

Absolute Path – begins with /– cd /var/log

Relative Path – does not begin with /, identifies a location relative to your current position – cd /var , cd log

Creating files

touch filename -can create multiple files ( touch filename1 filename2)

cp filename filename1

cp /var/log/messages . – dot represents current directory

vi

creating directories

mkdir foldername foldername1

copy directory – cp -r sourcefolder destinationfolder

find a file - find . -name “filename” or find / -name “filename”

to find the location of filename - locate filename

To update local database – updatedb and then give locate command so that it shows the location. Only when the database is updated the locate command works.

Wildcards - \* ? []

rm abc\*

touch abc{1..9}-xyz

ls -l abc\*

ls -l ?bcd\* - ? can be any character

ls -l \*[cd]\* | more – gives anything that has c or d letter in a file

rm \*xy\*

\ - as an escape character

^ - the beginning of the line

$ - the end of the line

Soft and Hard Links

Hard link – ln

Soft link – ln -s filename(go to the directory to create soft link for the file)

Hard link retains data even when removed from the folder, but soft link deletes completely in both source and destination.

------------------------------------------------------------------------------------------------------------------------

sudo -i - root user

mkdir ops devops

touch devops{1..10}.txt -> creates 10 files

copy directory - cp -r source destination

move files/directories – mv ops /dev , mv \*.txt textdir/

mkdir testdir{1..5}

cat /etc/os-release

vi command mode –

gg – to go to the beginning of the page

G – to go to end of the page

w – to move the cursor forward, word by word

b - to move the cursor backward, word by word

nw - to move the cursor forward to n words(5W)

nb – to move the cursor backward to n words(5B)

u – to undo last change(word)

U – to undo the previous changes(entire line)

Ctrl+R – to redo the changes

yy – to copy a line

nyy – to copy n lines(4yy)

dd – cut the line

ndd – delete n number of lines

p – to paste line below the cursor position

P – to paste line above the cursor position

dw – to delete the word letter by letter(like backspace)

x – to delete the word letter by letter(like del key)

ESC+:se nu – shows numbers in the file(set numbers)

/word – to search for a word in vi editor

File Types

ls -l – long listing

ls -lt – sort according to time stamp

ls -ltr – reverse the sort

ls -ld – to list only the directories

ls -l | grep drw | wc -l

grep word filename | wc -l

Hyphen(-) – regular file

d – directory

c – character file

b – block file

l – link (symbolic link file)

mkdir -p /opt/ops/dev/devops/test

ln -s dest source

ctrl+d to exit / exit – same

history

to check the filetype – file <filepath>

Filters

Grep – global regular expression print

grep word filename – searches word in file

grep -i firewall filename – ignore case sensitivity

grep -i word < filename -input redirection

grep -i word \* - searches in current working directory

grep -iR word \* - to search inside the sub directory as well

grep -R word /etc/\*

grep -vi word filename – does not show anything related to word and shows the remaining, reverse search

grep -c keyword filename – gives the count of keyword

grep -n keyword file – display the matched lines and their line numbers

grep -v keyword file – display everything but keyword

grep keyword file | awk ‘{print $1}’

grep keyword file | awk ‘{print $1}’ | cut -c1-3

egrep -i “keyword1|keyword2” file – to search for two words

less filename – can search for words and quit using q, can use up and down arrows

more filename – enter to read more

head -2 filename – shows first two lines of the file

tail -2 filename – last two lines of the file

tail -f filename – shows dynamic content of the file

/var/log – log files of the system

cut -d: -f1 /etc/passwd – delimiter(d),Column1(f1) – prints column 1

cut -d: f6-7 /etc/passwd

cut -c1 filename – prints first letter of all lines in file

cut c1,2,4 filename – prints first,second and fourth letter of all lines in the file

cut -c1-3 filename – list range of characters

cut -c1-3,6-8 filename

cut -b1-3 filename – based on byte size

ls -l | cut -c2-4

if there is no delimiter we can use awk

awk -F’:’ ‘{print $1}’ /etc/passwd – prints the first column

awk ‘{print $1} filename

ls -l | awk ‘{print $1,$3}’ prints 1st and 2nd column of ls -l

ls -l | awk ‘{print $NF}’ – prints last column

awk ‘/jerry/ {print}’ filename- search for a word in file

awk -F: ‘{print $1}’ /etc/passwd

echo “Hello Tom” | awk ‘{$2=”adam”, print $0}’ – replace words

awk ‘length($0) > 15’ filename – give length of line greater than 15(bytes/characters)

ls -l | awk ‘{if($9==”seinfeld”) print $0;}’ – lists the name matching with Seinfeld

ls -l | awk ‘{print NF}’ – counts the number of fields

Esc+:%s/word/replacementword – replaces all word with replacementword in a line

Esc+:%s/word/replacementword/g – replace globally(g)

Esc+:%s/word//g – replace with nothing

sed ‘s/word/replacementword/g’ filename – will just print on screen will not make any changes to file

sed -i ‘s/word/replacementword/g’ filename – will change the word in file

sed 's/pisces/aquarius/2g' astrology.txt - replaces the word **“pisces”**with **“aquarius”**from the second occurrence until the last one in the **astrology.txt** file

Redirections

Redirection is a process where we can copy the output of any command(s), file(s) into a new file. There are two ways of redirecting the output into a file using > or >> filename after the command.

echo “Welcome” > filename

echo “Welcome Again!” >> filename – to append to the previous one

uptime > /tmp/sysinfo.txt

ls > /tmp/sysinfo.txt – store the values present in current working directory

uptime >> /tmp/sysinfo.txt – to append the content to previous values

free -m – memory utilisation

df -h – hard disk utilisation

freeee -m 2> /tmp/error.log

freeee -m 2>> /tmp/error.log – redirect error to log file, 2 is for standard error

free -m 1>> /tmp/error.log – 1 is for standard output

free -m &>> /tmp/error.log – redirect any output to file

wc -l /etc/passwd – counts line numbers

ls | wc -l – ls output goes as input to right hand side command – and prints the number of files

ls | grep host – all file names that start with host

tail -20 /var/log/messages | grep -i vagrant

free -m | grep Mem

ls -l | head – first ten files

ls -l | tail – last ten files

find /etc -name host\*

locate host

tee – store and view the output of any command

echo “Welcome” | tee filename

echo “Welcome” | tee -a filename – to append the content

ls -l | tee filename

Pipes

ls -l | more – to refine the output using more

Users & Groups

To control access to files and resources

id username – gives info about the user

Users password is stored(encrypted) in /etc/shadow file.

/etc/passwd

/etc/group

useradd jenkins

groupadd devops

usermod -aG devops jenkins – adding jenkins user to devops group

passwd jenkins – can be done as root user , setting password

lsof -u username – lists all files opened by the user

userdel username – delete a user

userdel -r username – delete along with home directory

groupdel groupname

Permissions

User group others

-rwxr-xr-x – read write execute permissions

- -> filetype

useradd

groupadd

chown -R ansible:devops /opt

chmod o-r /opt - > remove read permission for others

chmod g+r /opt -> add read permission for group

chmod a-r /opt -> remove read permissions for all user,group,others

u – user, g – group, o – others(everyone on the system)

4- read, 2-write, 1 -execute

chmod 777 filename – full permissions

Sudo

sudo -i

/etc/sudoers - Only user in /etc/sudoers file or /etc/sudoers.d dir can use sudo -i command to switch to root user

Package management

rpm -qa – list all rpm packages in the system

arch / uname -m - to find arch of the system

curl link -o filename/outputfile(.rpm) / wget url– to download

rpm -ivh filename(.rpm) – to install

rpm -qa | grep telnet

rpm -e filename(.rpm) – to remove the package

yum install httpd/ dnf install httpd

dnf remove httpd

dnf install httpd -y – installs without asking

Services

sudo systemctl start httpd

sudo systemctl restart httpd

reload/status/

systemctl is-active httpd – to check the status

systemctl is-enabled httpd – checks if the file is enabled

Processes

top – shows dynamic process based on cpu and ram, will get sorted based on cpu utilisation, q quit

ps aux – similar to top command

ps -ef | grep httpd | grep -v ‘grep’ – do not show grep and show the process running for httpd

kill / kill -9 processid – to stop the process, -9 for forcefull

ps -ef | grep httpd | grep -v ‘grep’ | awk ‘{print $2}’ – awk prints the second column

ps -ef | grep httpd | grep -v ‘grep’ | awk ‘{print $2}’ | xargs kill -9 – executes a particular command

Archiving

tar -czvf jenkins.tar.gz directoryname/path – create, compress, verbose, file – create a tar file

tar -xzvf jenkins.tar.gz – extract the tar

zip -r httpd.zip jenkins – archive a file

unzip httpd.zip – unzip an archive

tar cvf filename.tar /home

Text processors Commands

sort filename – to sort in ascending order

sort -r filename – reverse sort, descending order

sort -k2 filename – sort by second column

ls -l | sort k9

uniq filename – removes the duplicates

sort filename | uniq

sort filename | uniq -c – sort file and list count

sort filename | uniq -d – shows only repeated line

wc – word count

wc filename - lists newline count, word count and byte count

wc -l filename – line count

wc -b filename – byte count

wc -w filename – word count

ls -l | grep drw | wc -l

grep word filename | wc -l

Compare files

diff filename1 filename2

cmp filename1 filename2

UBUNTU COMMANDS – based on Debian architecture

adduser devops

export EDITOR=vim – normally used is nano but to make it vim use export

dpkg -i filename.deb – install in ubuntu

apt install apache2

apt upgrade

apt remove apache2

apt purge apache – clean uninstall

------------------------------------------------------------------------------------------------------------------------Bash Scripting

Bash scripting is used to automate repetitive tasks and simplify system administration.

#! – Shebang/hashbang is used to specify the interpreter that should be used to execute the script.

/bin/bash – tells the system to use the bash shell located at /bin/bash to interpret the script

#!/bin/bash

echo “The uptime of the system is” //prints The uptime of the system is

uptime //run commands

to execute the script - ./script.sh

# - comments

#!/bin/bash

sudo yum install wget unzp httpd -y

sudo systemctl start httpd

sudo systemctl enable httpd

mkdir -p /tmp/webfiles

cd /tmp/webfiles

wget <https://www/tooplate.com/zip-templates/2098_health.zip>

unzip 2098\_health.zip

sudo cp -r 2098\_health/\* /var/www/html

systemctl restart httpd

rm -rf /tmp/webfiles

SKILL=”DevOps”

echo $SKILL – DevOps

#!/bin/bash

TEMPDIR=/data/sample/example

#Creating temp directory

mkdir -p $TEMPDIR

cd $TEMPDIR

when we give $Y without giving the variable a value, we can give the values in arguments

example –

#!/bin/bash

echo “Value of 0 is”

$0

echo “Value of 1 is”

$1

Execute the script – ./script.sh Linux

Output – Value of 0 is

./script.sh

Value of 1 is

Linux

To print exit status of last command – echo $?

0 – true

Non zero - false

Use double quotes to print $ value in output

VIRUS=”COVID19”

echo “Due to $VIRUS”

or

VIRUS=”COVID19"

echo ‘Due to \$VIRUS’

prints -> Due to COVID19

.bashrc – environment variables(root user)

/etc/profile – globally for all users

read – to enter in console

#!/bin/bash

echo “Enter your skills:”

read SKILL

echo “Your $SKILL skill is in high demand”

MM HH DOM mm DOW COMMAND

\* \* \* \* \* /opt/scripts/11\_monit.sh $>> /var/log/monit\_httpd.log

LOOPS

Decision making – if

#!/bin/bash

read -p “Enter a number: “ NUM

echo

if [ $NUM -gt 100 ]

then

echo “num is greater than 100”

sleep 3

date

fi

elif

#!/bin/bash

value=$(ip addr show | grep -v LOOPBACK | grep -ic mtu)

if [ $value -eq 1 ]

then

echo “1 active network found”

elif [ $value -gt 1 ]

then

echo “value greater than 1”

else

echo “No active interface found”

fi

fi – to mark an end for if statement

For loop

#!/bin/bash

for VAR1 in java .net python ruby php

do

echo “Looping”

echo “Value of VAR1 is $VAR1”

date

done

#!/bin/bash

MYUSERS-“alpha beta gamma”

for usr in $MYUSERS

do

echo “Adding users $MYUSERS”

useradd $usr

id $usr

done

for i in {0..10..2}

do

echo “$i”

done

for (( i=0; i<=5; i++ ))

do

echo "Element $i"

done

infinite loop – for (( ; ; ))

While loops

#!/bin/bash

counter=0

while [ $counter -lt 5 ]

do

echo “Looping”

echo “Value of counter is $counter”

counter=$(( $counter+1 ))

done

Infinite loop – while true