Complete
Linux Training

-------Module 1(Understanding Linux Concepts)----------

1)What is Linux ?

Ans. Linux is an operating system which sits in the middle of your hardware and users like microsoft, mac.

2)Linux vs Unix ?

Ans. -Unix was first developed for multiuser and multitasking in mid 1970 in Bell Labs by AT&T and Massachusetts Institute of Technology
-Then born Linux in 1991 by Linus Torvalds
-Linux is mostly free
-Linux is open source
-Unix is mostly used by Sun as Solaris, HP-UX, AIX etx.
-Linux is used by developers communities or companies Ex. RedHat, CentOS, Debain etc
-Unix comparatively supports very fewer file systems.
-Linux can be installed on a variety of computer hardware, from mobile phones to tablets to video games.

----------Module 2----------

Linux Distributions :-RedHat used by major companies, Not open source, Provides technical support.
:-CentOS is same and managed by redhat it dont provide technical support and its free.
:-Suse is owned by novell and its free.
:-Debian and Ubuntu.

1)What is VirtualBox ?

Ans. Virtual box is open source hypervisor for x86 computers developed by oracle.
It extends the capatibilty of your existing computer so that it can reun multiple OS on one hardware at
same time

2) CentOS vs CentOS Stream

Ans. CentOS is a community enterprise operating system in 2004 Greg Kurtzer took source code form RedHat and made a CentOS operating system.
Before Feb 2021 Fedora ----> RedHat Enterprise Linux Edition ------> CentOS

After Feb 2021 Fedora ----> CentOS Stream --> RedHat Enterprise Linux Edition.

3) Linux vs Windows.
Ans.
-Linux is free, Windows is not free.
-Linux is not user friendly whereas windows is user friendly
-Linux is reliable but Windos requires reboot
-Linux is enterprise level softwares, Windows are much largers selection of softwares.
-Linux is best for multitasking, Windows multitasking is availablt but high CPU or memory resources required.
-Linux is very secure, Windows is somewhat secure.
-Linux is open source, Windows is not an open source OS.

4) Who uses Linux ?
Ans.
-US Government and Agencies(National, State, Federal and International.) P.S Not all US agencies
-NASA
-HealthCare, Scientists
-Bullet Trains in Japan.
-Traffic Conterol.
-Financial Institutes Ex. New York Stock Exchange.
-World Ecommerce Leaders. Ex. Amazon, eBay, Paypal
-Fortune 500 companies.

5)Keyboard Keys used in Linux
Ans. Esc, ~(tilde), `(Backtak), !, @, #, $, ^, &, *, (,), _, -,{},[],<>, Enter, ' ", \, |, /, ?, Shift, Tab, Ctrl, Alt.

----------Module 3(System Access and File System)----------
1)AWhat are command Prompt?
Ans. A command prompt or simply referred as prompt is a short text at the start of the command line folloed by prompt symbol on a command line interface.
To get your prompt back Ctrl + c.

2) Accessing to Linux System.

Ans. Each operating system has a different protocol or client that is used to access the system.
Ex. Windows = Remote Desktop(If we want to connect to another windows)
VMwareESX = cSphere client
Linux = Putty, SecureCRT
SSH from Linux to Linux.

3) To check ip we can use the command ip or ip addr or ip a or ifconfig.
Host-Only Adpter - Allows communication between your PC and Virtual Machine.
Bridged Adapter - Allows communication between your PC and VM plus allows communication to the internet.

--Important Things to remember in Linux--
-Linux has super user account called root, it is the most powerful account can create, modify , delete or make changes to system configuration files.
-Linux is case sensitive system.
-Avoid spaces when creating files or directories.
-Linux is mostly CLI(Command line INterface) not GUI.
-Linux kernel is a small softare within linux os that takes commands from uses and pass them to system hardware or peripherals.
-Linux is very flexible compared to other OS.

4)Linux File System.
-OS store data on disk drives using a structure called a filesystem, consisting of files directories and the information needed to access and locate them.
-There are many different types of file systems.In general improvements have been made to file systems with new releases of operating system and each new file system has been given
a specific name. Eg. ext3, ext4, XFS, NTFS, FAT etc
-Linux filesystems store information in a hierarchy of directories and files.
/root is the first structure.

5)File System Structure and its Description...

/boot - Contains file that is used by the boot loader(grub.cfg)

/root - root user home directory. It is not same as /
/dev - System devices(e.g - Disk, cdrom, speakers, keyboard etc)
/etc - Configuration files.
/bin -> /user/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 library files needed by commands and apps.
strace -e open pwd
/tmp - Directory for temporary file.
/home - Directory for user.
/var - System logs.
/run - System daemons that start very early(e.g systemd and udev)
/mnt - To mount external filesystem(e.g NFS)
/media - for cdrom mounts.

6) File system navigation commands.

"whoami" - show the username
"cd" - Change Directory or "cd (name of directory)"
"pwd" - present working directory. It tells you where your current location is.
"ls" - list. It lists all the files within a current working directory.
"clear" - clear the screen.
"ls -l" - List all files with their details.
"cd .." - Takes one directory back.

7)What is root ?
Ans. There are three tyoes of root in linux
Root account : Most powerful account and has access to all commands and files.
Root as /: The very first directory in Linux is call root directory.
Root home directory : The root user account directory is the root home directory.

8)File System Paths.
Absolute paths :- Always begins with a /. This indicates that path starts at the root directory.
Ex - cd/var/log/hulk

Relative paths :- Does not start with /. It identifies a location relative to your current position.
Ex - cd/ var
cd log
cd samba

9)Creating files and Directories
-touch -- (It creates a file)
Ex - touch filename
touch course, touch college
touch docs hello -- Creates more than one file in one command
-cp -- (Copying a new file)
Ex - cp course college

-vi -- (creates a file and go to the file editor)
Ex- vi filename
to exit the text editor :wq!

-mkdir -- (It makes a directory)
Ex - mkdir newfolder

10) Copying Directories

*Command to copy a directory is cp
*To copy a direcotry ion linux you have to execute the "cp" command with the "-R" option for recursive and specify
the source and destination directories to be copied.
Command :-
cp -R <source_folder> <destination_folder>

11)Linux file types
- - Regular file.
- d - Directory file.
- l - Link file.
- c - Special or device file.
- s - Socket file.
- p - Named pipe
- b - Block Device

12) Finding files and Directories.

Two main command used to find files and directories
* find
Ex- find . -name "filename" # (.) means in this directory

* locate
Ex- locate filename

13) Find and Locate
* Locate uses a prebuilt database which should be regularly updated, while find iterates over a filesystem
to locate files. Thus, locate is much faster than find, but can be inaccurate if the database is not updated.
* To update locate database run updatedb.
By becoming root we can use updatedb.

14) Changing password.
- You should change our initial password as soon as you login.
Command:-
* passwd userid (In userid enter the username)

15) Wildcards(*,?,^,[])
- A wildcard is a character that can be used as a substitute for any of a class of characters in a search.
* :- represents zero or more characters.
? :- Represents a single character.
[] :- represents a range of characters

Ex-
touch abcd{1..9}-xyz -- This will create abcd1-xyz, abcd2-xyz....upto abcd9-xyz

rm abc* -- * wild card does is it will remove the files named abc and any file whose name starts with abc--

ls -l ?bcd -- Give the files that have anything in place of the question mark.

ls -l [cd] -- Give the files that have a or b.

Wildcards can be used in many of the commands.

#some more wildcards
\ = as an escape character.
^ = the beginning of the line.
$ = the end of the line.

16) Soft and Hard Links
inode = Pointer or number of a file on a hard-disk.
Soft Links = Link will be removed if file is removed or renamed.
Hard Link = Deleting or renaming or moving the original file will not affect the hard link.
Command:-
* ln (used to link)
* ln -s
#We cannot create soft or hard link within the same directory with the same name.
#We can find the inode by using the command ls -ltri or ls -ltri filename

----------Module 4(Linux Fundamentals)----------

1)Command Syntax

man ls -- It gives manual about all the different option #To change page press space and to quit hit q

*Command options and arguments.
*Commands typically have the syntax: command options(s) argumment(s)

Options:-
Modify the way that a command works.
Usually consists of a hyphen or dash followed by a single letter.
Some commands accept multiple options which can usually grouped together after a single hyphen.

Arguments:-
Most commands are used together with one or more arguments.
Some commands assume a default argument if none is supplied.

Arguments are optional for some commands and required by others.
Ex- ls is a command and -ltr is option
ls -ltr filename -- Here the filename is an argument
rm -r -- It is to remove a directory.

2) Files and Directory permissions.

* UNIX is a multi user system. EVery file and directory in your account can be protected from or made accessible to other users by changing its access permissions. Every user has responsibility for
controlling access to their files.
* Permissions for a file or directory may be restricted to by types
* There are 3 types of permissions
:- r - read
:- w - write
:- x - execute=running a program.

* Each permission (rwx) can be controlled at three levels.
:- u - user=yourself
:- g - group=can be people in the same project.
:- o - other=everyone on the system.

* File or directory permission can be displayed by running ls -l command.
Ex - -(type pf file)rwx(user)rwx(group)rwx(other)

* Command to change permission
chmod

Ex - Let a file name jerry have permissions -rw-rw-r--
#) We want to remove write permissions from group user, so we will use the command:- chmod g-w jerry

#) We want to remove the read permission from every user we will use:-
chmod a-r jerry # a represents every group user, group and others

#) If we want to give all the permission then we use the command:-
chmod u+rw jerry # + is used to give permission to all the users, groups and others.

#) If we want to give permissions to the group then we can use the code:- chmod g+rw jerry

#) If we dont have x(execution) permission to a directory we cannot access a directory.

3) Permission using Numeric mode.

* Permission to a file and directory can also be assigned numerically.

:- chmod ugo+r filename
OR
chmod 444 filename

* The below data assigns numbers to permission types

0 --- No Permission ---
1 --- Execute --x
2 --- Write -w-
3 --- Execute+Write -wx
4 --- Read r--
5 --- Read+Execute r-x
6 --- Read+Write rw-
7 --- Read+Write+Execute rwx

Ex - chmod 764 filename (# The first number is for the user, The second number is for group and the third number is for the others.)

:- This is going to assign Red+Write+Execute to the user.

:- This is going to assign Read+Write to the group.

:- This is going to assign Read to the other group.

Ex -
chmod 777 jerry

It will give all read, write and execute permission to all the user, group and other users.

chmod 600 jerry

It will give read and write permission for the user. and have no permission for group and others.

4) File Ownership Commands.

* There are two owners of a file or directory User and Group.
* Command to change file ownership.
chown :- chown changes the ownership of file.
chgrp :- chgrp changes the group ownership of a file.

* Recursive ownership change option(Cascade)
-R

After using ls -l command the third column tells who owns the file and fourth column tells which group owns it.

Ex- If we want to change ownership we need to go to root user by using sudo -i command #If we want to exit from the root we use exit command.

Then to change the ownership we need to go to that file directory and then use the command
chown root jerry # Here we changed the ownership of the jerry file to root.

chgrp root jerry # Here we change the group of the file jerry as root.

5) Access Control List (ACL)

* ACL provides an additional more flexible permission mechanism for file systems. It is designed to assist with UNIX file permissions. ACL allows you to give permissions
for any user or group to any disc resource.

* Use of ACL :- Think of a scenario in which particular user is not a member of group created by you but still you want to give some read and write access, How can you do it

without making user a member of group, Here comes Access Control Lists. ACL help us to do the trick.

ACL are used to make flexible permission mechanism in Linux.
From Linux man pages, ACLs are used to define more fine grained discretionary access rights for files and directories.
*Commands to assign and remove ACL permission are:
setfacl AND getfacl

List of Command for setting up ACL:
i) To add permission for user
setfacl -m u:user:rwx /path/to/file

ii) To add permission for a group
setfacl -m g:group:rw /path/to/file

iii) To allow all files or direcotries to inherit ACL, entries from the directory it is within
setfacl -dm "entry" /path/to/dir

iv) To remove a specific entry
setfacl -x u:user /path/to/file (For specific user)

v) To remove all entries
setfacl -b path/to/file (For all users)

Note:
* As you assign the ACL permission to a file/Directory it adds + sign at the end of the permission.

To get all the information about file we will use the command getfac1 filename

Ex -

setfacl -m u:ssingh:rw /tmp/jerry
Here user ssingh has read and write permission to specific file jerry.

setfacl -m g:ssingh:rw /tmp/jerry
Here group ssingh has read and write permission to specific file jerry.

setfacl -x u:ssingh /tmp/jerry
Here user ssingh has been removed execution permission from a specific file jerry.

setfacl -b tmp/jerry
Here every permission is removed from all the users from the file jerry.

6) Help Commands

There are three types of help commands

whatis commandname
commandname --help
man commandname

Ex -
whatis ls # Tells the info about ls command
whatis cd # Tells the info about cd command

ls --help # Tells the detailed info about ls command
cd --help # Tells the detailed info about cd command

man ls # It gives a nice manual of information about the ls command
man pwd # It gives a nice manual of information about the pwd command

7) TAB completion and UP arrow Keys.

Hitting TAB key completes the available command, files or directories
chm TAB
ls j<TAB>
cd Des<TAB>

Hitting up arrow keyreturns the last command ran.

8) Adding Text to Files(Redirects).

#) 3 Simple ways to add text to a file.

i) vi
ii) Redirect command output > or >>
iii)echo > or >>

Ex-
-> echo "hi name is Jerry"
O/P - hi my name is Jerry

echo "Hello everyone how are you" > jerry # Works in already created files.
#) The above statement is added to the file name called jerry.

cat jerry # cat command is used to give out the output in the file name jerry.
O/P - Hello everyone how are you

-> echo "myself a new linux machine" >> jerry # If we use a single '>' sign our new line will overwite our old line
but if we use double '>>' sign our new line will also be added to the jerry file.

cat jerry
O/P - hi my name is Jerry
myself a new linux machine

-> ls -ltr > listingdir(a new file) # It will insert all the data of ls -ltr into the filename called listingdir

9) Input and Output Redirects(>,>>,<,stdin,stdout and stderr)

There are three redirects in Linux
i)Standard input (stdin) and it has file descriptor number as 0.
ii)Standard output (stdout) and it has file descriptor number as 1.
iii)Standard error (stderr) and it has file descriptor number as 2.

Output(stdout) - 1
By default when running a command its output goes to the terminal.
The output of a command can be routed to a file using>symbol
Ex - ls -l > listings
pwd > findpath

If using the same file for additional output or to append to the same file then use >>
Ex - ls -l >> listings
echo "Hello World" >> findpath

Input (stdin) - 0
Input is used when feeding file contents to a file.
Ex- cat < listings
mail -s "Office memo" allusers@abc < memo

Error (stderr) - 2
When a command is executed we use a keyboard that is also considered (stdin -0)
That command output goes on the monitor and that output is (stdout - 1)
If the command produced any error on the screen then it is considered (stderr - 2)
We can use redirects to route errors from the screen
Ex - ls -l /root 2> errorfile
telnet localhost 2> errorfile

It is used to store error messages in error files.

10) Standard Output to a File (tee command)

"tee" command is used to store and view the output of any command. The command is named after the T-splitter used in plumbing. It basicallt breaks the output of a program so that it can be both displayed and saved
in a file. It does both the tasks simultaneously, copies the result into the specified files or variables and also display the result.

11) Pipes (|)

A pipe is used by the shell to connect the output of one command directly to the input of another command

command1 [argument1] | command2 [argument2]

Ex -

ls -ltr | more # It will give content page by page not one a time.

ls -l | tail -1 # tail is another command which gives last line of output.

12) File maintanence Commands

cp :- Copy one file to another
Ex - cp george david # It also create a file and rename it as david
cp david /tmp # The david file will be copied into the tmp directory

rm :- Remove a file

mv :- Move a file
Ex - mv lex luther # It will rename the lex file into luther within the same directory

mkdir :- Make a directory
rmdir or rm -r :- Remove a directory
chgrp :- Change group
chown :- Change ownership

13) File display commands

cat :- View entire content of file
more/less :- Helps in viewing one page at a time.
Ex -
more jerry #If their is so much messages more command helps in viewing one page at a time.

head :- Can view the first lines followed by number of line to view.
Ex - head -2 jerry #Shows first two lines of the file
head -7 jerry #Shows first seven lines of the file

tail :- Can view last line of the file.
Ex - tail -2 jerry #Shows last two lines of the file
tail -7 jerry #Shows last seven lines of the file

14) Filters/ Text Processing Commands.

cut - Allows to cut the output.
awk - Allows you to list by the columns
grep and egerp - Allows you to search by keywords.
sort - It sorts out the output in alphabetical order.
uniq - It will not show any duplicates of files.
wc - It tells how many word, letter, and lines in a file.

15) cut - Text Processor Commands.

cut is a command line utility that allows you to cut parts of lines from specified files or piped data and print the result to standard output.
Ex -
cut filename = does not work

cut -c1 filename = Takes out the first character
Suppose we have a file names friends which has
Ravi
Shyam
Rohan
Jay
Zoran

If we use the command:- cut -c1 friends It will return

R
S
R
J
Z

If we want specific characters then
cut -c1,2,4 friends

If we want ranges of characters
cut -c1-5 friends # It will print upto 5th character.

List specific range of character

cut -c1-2,6-8 friends #It will print characters from 1 to 3 then 6 to 8 letters

List by byte size
cut -b1-3 friends # Print characters by byte size.

cut -d: -f 6 /etc/passwd #List first 6th column separated by :

ls -l | cut -c2-4 # Only print user permissions of files/dir.

16) awk - Text Processor Commands

awk is a utility designed for data extraction. MOst of the time it is used to extract field from file or from output.
awk --version = Give the version of the awk

awk '{print$1}' friends # It gives the first coloumn of the file.

ls -l | awk '{print $1,$3}' # It will print first and third coloumn of the listing of ls -ltr

ls -l | awk '{print $NF}' # It will print the last coloumn of the listing ls.

awk '/Jerry/ {print}' friends # It will find Jerry from our friends file.

awk -F: '{print $1}' /etc/passwd #It will tell the first field of /etc/passwd

echo "Hello Tom" | awk '{$2="Adam"; print $0}' # It will replace the Tom word by Adam.

cat friends | awk '{$2="Singh"; print $0}' # Replace every characters last name by Singh.

awk 'length($0) > 15' friends #It will give all the characters name who length is greater than 15 characters.

ls -l | awl '{if($9 == "singh"} print $0;)' # It will give all the characters which matches "singh"

ls -l | awk '(print NF}' # It tells the column in ls -l

17) grep/egrep - Text Processor Commands.

grep command stands for global regular expression print, it processes text line by line and prints any lines which matches a specified pattern.

grep singh friends # It gives the lines in friends file whose name matches with Singh

grep -c singh friends # It gives the number how many times the file occur.

grep -i singh friends # It ignores the case sensitive in our file.

grep -n singh friends # It gives the name and also the line number in which singh is found in the friends file.

grep -v singh friends # It gives the remaining file which don't contain singh from our file.

grep -vi seinfield seinfield-characters | awk '{print $1}' | cut -c1-3
It gives the first 3 characters which dont contain the word seinfield.

ls -l | grep Desktop # It will find the word desktop in the desktop folder.

egrep -i "seinfield|costanza" seinfield-characters # It will find both the characters which contains seinfield and costanza.

18) sort/uniq - Text Processors Commands

Sort command sorts in alphabetical order.
Uniq command filters out the repeated or duplicate lines.

Ex-

sort seinfield-characters # Its will sort characters in alphabetical order.

sort -r seinfield-characters # Its will sort characters in reverse-alphabetical order.

sort -k2 seinfield-characters # It will sort alphabetically by second column.

uniq seinfield-characters #It will print unique characters after sorting or remove duplicates.

sort seinfield-character | uniq # It will remove the duplicate.

sort seinfield-character | uniq -c # Show the number of character how much it repeated.

sort seinfield-character | uniq -d # It will show the character which is repeated.

19) wc - Text Processor Commands.

The command reads either standard input or a list of files and generates : newline count, word count and byte count.

wc seinfield-characters # It will tell lines, words and bytes.

wc -l seinfield-characters # It will give the line count.

wc -w seinfield-characters # It will give the word count.

wc -c seinfield-characters # It will give the byte count.

wc seinfield

ls -l | wc -l #It will give how many files are their in the directory of ls -l.(Whatever value you get minus it by 1 that many files are their.)

grep seinfield seinfield-character | wc -l # It will show the number how many characters have seinfield.

20) Compare files (diff and cmp)

diff (Line by Line)
cmp (Byte by Byte)

21) Compress and uncompress (tar, gzip, gunzip)

tar
gzip
gzip -d OR gunzip

tar cvf compressed.tar /home/iafzal # tar or zip all the files in home directory into compressed.tar file.

tar xvf compressed.tar #It will extract all the files present in compressed.tar.

gzip compressed.tar # Compressed the files more than tar.

gzip -d compressed.tar.gz # Uncompress the files.

rm -rf directoryname # Force remove directory.

22) Truncate File Size(truncate)

The Linux truncate command is often used to shrink or extend the size of a file to the specified size.

Command:-
truncate -s 10 filename

Ex:-
touch seinfieldwords # We created a file seinfieldwords
echo puffyshirt giddyup yadayada kavorka serenitynow festivus > seinfieldwords # We inserted the words in the file seinfieldwords

truncate -s 40 seinfieldwords # THe truncate command will reduce or increase the size to 40 bytes or any specified size.
It will remove words/letters to make it to the specified size.

23) Combining and Splitting Files.

Multiple files can be combine into one and one file can be split into multiple files.

cat file1 file2 file3 > file4
split file 4

Ex-
split -l 300 file.txt childfile #It will split file.txt into 300 lines per file and give the output childfileaa, childfileab, childfileac.....

We create a file countries with usa, uk, uae, canada, france, swiz, japan.

split -l 2 countries sep #It will split the file countries and create files sepaa, sepab, sepac, sepad with each file having two countries in it.

24) Linux vs. Windows Commands
Windows/Linux:-
Listing of a directory:- dir / ls -l
Rename a file:- ren / mv
Copy a file:- copy / cp
Move a file:- move / mv
Clear Screen:- cls / clear
Delete File:- del / rm
Compare contents:- fc / diff
Search for word:- find / grep
Display command help:- command/? / man
Display your location:- chdir / pwd
Display the time:- time / date

----------Module 5(System Administration)----------

1) Linux File Editor(vi)

A text editor is a program which enables you to create and manipulate data(text) in a Linux file
Their are several standard rext editors available on most Linux systems.
vi - Visual Editor (Using available in almost every linux dsitribution)

ed - Standard Line Editor
ex - Extended Line Editor
emacs - A full Screen Editor
pico - Beginners Editor
vim - Advance Version of vi

vi supplies command for:
Inserting and Deleting Text, Replacing Text, Moving around a file, Finding and substituting strings, Cutting and Pasting Text.

Most Common Keys:
i - Insert # After creating a file using vi command press i to write anything in the file.
Esc - Escape out of any mode

(The below command will only work in navigate mode)
r - Replace # In the create file mode navigate to the letter you want to edit using arrow and then press r and replace your own letter.
d - Delete # Delete a line
can also press x to delete a character in navigate mode.
:q! - Quit without saving
:wq! - Quit and Save
/lesson - Find the letter lesson in vi Editor.

2)Differnce between vi and vim Editors.
Ans. vi is installed in more places, shorter name, and its simpler
vim has spellcheck, comparison, merging, unicode, regular expressions, scripting languages, plugins, GUI, folding, syntax, highlighting.
https://www.openvim.com/
https://www.vim-adventures.com/

3) "sed" Command
Ans. It Replace a String in a file with a new string
Find and Delete a line.
Remove empty lines.
Remove the first or n lines in a file.
To replace tabs with spaces.
Show defined lines from a file.
Substitute within vi editor.

(:- NOTE that this sed will only change what is visible on screen it will not change the file, To change the file enter -i before 's/Kenny/...' :-)

Ex:-
sed -i 's/Kenny/Lenny/g' seinfield-characters # It will substitute or replace kenny into lenny in the file seinfield-characters # We use g to make it global i.e if their is more than one kenny in that file.
#) If we dont use -i then it will replace Kenny in screen only it will not make changes to the file.

sed -i 's/Costanza//g' # Remove the Costanza from the screen.

sed '/Seinfield/d' seinfield-characters # Will deletet he line that has seinfield in it from screen.

sed '/^$/d' seinfield-characters # Delete the Empty lines from the screen.

sed '1d' seinfield-characters # Delete first line from the file on the screen.

sed '1,2d' seinfield-characters # Delete the first and second line from our file on the screen.

sed 's/\t/ /g' seinfield-characters # Replace TAB spaces into space in the screen.

sed -n 12,18p seinfield-characters # It will show on the screen all the characters have 12 to 18 letters.

sed 12,18d seinfield-characters # It will show on the screen all the characters except 12 and 18 line.

sed G seinfield-characters # Add empty line space between each line in our screen.

sed '8!s/Seinfield/S/' seinfield-characters # It will change all the seinfield names into s except the eighth line in file on the screen.

:%s/Seinfield/Peter/ # Change all the Seinfield characters into Peter in the vi Editor.

4) User Account Management (useradd, groupadd, usermod,userdel, groupdel)

Commands:- Files:-
useradd /etc/passwd
groupadd /etc/group
userdel /etc/shadow
groupdel
uesrmod

Ex:-
useradd -g superheros -s /bin/bash -c "user description" -m -d /home.spiderman spiderman.

(:- First Become Root :-)

useradd spiderman # It will create a user named spiderman.
Also change password of the new user using passwd spiderman
groupadd superheroes # It will add a group named superheroes.
userdel -r spiderman # It will delete the user as well as directory of the user spiderman.
groupdel nonewgroup # It will delete the group nonewgroup.
usermod -G superheroes spiderman # It will add the spiderman user into the superheroes group.
chgrp -R superheroes spiderman # It will change the group of spiderman into superheroes.
cat /etc/shadow # Passwords for the users.
useradd -g superheroes -s /bin/bash -c "Ironman Character" -m -d /home/ironman ironman # Create a user ironman with the group superheroes.

5) Enable Password Aging

The chage command - per user
Ex-
chage [-d lastday] [-m mindays] [-M maxdays] [-W warndays] [-I inactive] [-E expiredate] user

* -d = 3 Last Password change: Days since Jan 1,1970 that password was last changed.
* -m = 4 The minimum number of days required between passwrod changes i.e the number of days left before the user is allowed to change his/her password
* -M = 5 The maximum number of days the password is valid.
* -W = 6 The number of days before password is to expire that user is warned that his/her password must be changed.
* -I = 7 The number of days after password expireds that account is diasbled.
* -E = 8 The account is disabled and can no longer be used.

File = /etc/login.def

chage -m 5 -M 90 -W 10 -I 3 -E 7 babubutt # Change the password configs for the file babubutt

*To check the configs we can use :-
grep babubutt /etc/shadow

6) Switch Users and Sudo Access(su, sudo)
Commands:-
su - username
sudo command
visudo # Edits configuration files of sudo

File
/etc/sudoers

Ex:-
su - spiderman # We will become user spiderman.
sudo fdisk -l # Display info about the user.

7) Monitor Users (who, last, w, id)

who # This command tells who logged in our machine
last # This will tell all the info about every user who logged in the system.
w # w works pretty much same as who command.

(:--We need to install finger on our linux machine using the command yum install finger -y |:-)
finger

id # It will give all the information about ourself.

8) Talking to Users (users, wall, write)
Commands:-

Ex:-
users # Know which users are logged in

wall # All users who are connected will see the below message.
Please logoff. This system is coming down for maintanence
(ctrl + D)

write spider
Hi spider, your web is getting bigger. Please stop spreading your web
Message will go to only the user spider.

9) Linux Directory Service - Account Authentication

Types of Accounts
* Local accounts
* Domain/Directory accounts

10) Difference between Active Directory, LDAP, IDM, WinBIND, OpenLDAP etc.
Ans.
Active Directory is for Microsoft.
IDM is Identity Manager build by RedHat.
WinBIND is used to communicate with Windows(Samba build this add-on).
OpenLDAP(opensource) is a directory service like IDM.
IBM Directory Server.
JumpCloud
LDAP = Lightweight Directory Access Protocol

11)System Utility Commands (date, uptime, hostname, uname, which, cal, bc)
Ans.

date # This command gives the date and time.
uptime # This command tells you how many users logged in and how much time the system in on.
hostname # This command tells who is the host in Linux.
uname # This command tells which machine is our system.

which # This command tells where the command is located
For Ex.
which pwd # This will show where pwd command is located.
which date # This will show where date command is located.

cal # This will gives calender of this month.
Ex.
cal 9 1977 # This will give month calender of september year 1977
cal 2016 # This will give calender of all months of year 2016

bc # This code we will go to a basic calculator
For quitting the calculator we will write quit.

12) Processes, Jobs and Scheduling.
Ans.
Application also referred as service: It is a program which runs in your computer.
Script: Script is something which is written in file and packaged in a way so that it can be executed.
Process: When you start a application it generates a Process.
Daemon: It is something that continously runs in the background.
Threads: Every Application can have multiple threads.
Job: A job is something that is created by scheduler to run a process.

Commands:
systemctl or service
ps
top
kill
crontab

at

13) sytemctl command
systemctl is a new tool to control system services.
It ia used to start linux or any third party services.

Usage example #To use this command we need to be root
systemctl start|stop|status servicename.service (firewalld) # Ex: systemctl start firewalld
systemctl enable servicename.service
systemctl restart|reload servicename.service
systemctl list-units --all

The output has following columns:
UNIT:The systemd unit name.
LOAD:Whether the units configuration has been parsed by systemd. The configuration of loaded units is kept in memory.
ACTIVE:A summary state about wwhether the unit is active. This is usually a fairly basic way to tell if the unit started successfully or not.
SUB:This is lower entry level state that indicates more detailed information about that unit. This often varies by unit type, state and actual method in which the unit runs.
DESCRIPTION:A short textual description of what the unit is/does.

*To add a service under systemctl management:
Create a unit file in /etc/systemd/system/servicename.service

*To control full system using systemctl
Commands:-
systemctl poweroff (It will poweroff the system)
systemctl halt
systemctl reboot (It will restart the system)

14) ps command

ps command stands for process status and it displays all the currently running processes in the linux system

Usage Examples
ps = Shows the processes of the current shell

PID = the unique process ID
TIY = terminal type the user logged in
TIME = amount of CPU in minutes and seconds that the process has been running
CMD = name of the command

ps -e = Shows all the runnig processes
ps aux = Shows all runnignprocesses in BSD format # BSD is a special format
ps -ef = Shows all the running processes in full format listing.(Most commonly used)
ps -u username = Shows all the processes by specific user.

15) top command

top commands is used to show the linux processes and it provides a real-time view of the running system
This command shows the summary information of the system and list of processes or threads which are currently managed by linux kernel.
When the top command is executed then it goes into interactive mode and you can exit out by hitting q
Usage: top
PID:Shows task's unique process id.
USER:Username of owner of task
PR:The "PR" field shows the scheduling priority of the process from the perspective of kernel.
NI:Represents a nice value of the task. A negative nice values implies higher priority and positive nice value means lower priority.
VIRT:Total virtual memory used by the task.
RES:Memory consumed by the process in RAM.
SHR:Represents the amount of shared memory used by a task.
S:This field shows the process state in the single letter form.
%CPU:Represents the CPU usage.
%MEM:Shows the Memory usage of task.
TIME+:CPU Time, the same as 'TIME' but reflecting more granularity through hundreths of second.

Some more commands
top -u pritishtripathy = shows tasks/process by the user owned.

top "then press c" = shows command absolute path.
top "then press k" = kills a process by PID within top session # First enter top command then press k, It will ask for PID after entering PID we can kill the desired process.
top "then press M and P" = It sorts all Linux running processes by Memory usage.

Note:- top command refreshes the information within 30 seconds.

16) kill command

kill command is used to terminate process manually
It sends a signal which ultimately terminates or kills a particular process or group of processes

Usage:
kill [OPTION] [PID]
OPTION = Signal name or signal number/ID
PID = Process ID

Commands:-
kill -l = To get a list of all signal names or signal number.

Most used signals are:
kill PID = kill a process with default signal
kill -1 = Restart
kill -2 = Interrupt from a keyboard just like Ctrl C
kill -9 = Forcefully kill the process
kill -15 = Kill a process gracefully.

Ex:-
kill 750 # It will kill the process whose process ID is 750.
kill -1 6290 # That process is restarted

Other similar kill commands are:
killall
pkill

17) crontab command

crontab command is used to scheduled task

Usage:
crontab -e = Edit the crontab
crontab -l = List the crontab entries
crontab -r = Remove the crontab
crond = crontab daemon/service that manages scheduling.
systemctl status crond = To manage the crond service
format = day(0-6, Sun-Sat), month(1-12), day of the month(1-31), hour(0-23), minute(0-59)

*Create a crontab enty by scheduling a task.
crontab -e
schedule, echo "This is my first crontab entry" > crontab-entry

Example:-
first enter crontab -e command, It will take to crontab editor
then in editor
21 16 * 10 * echo "This is my first crontab entry" > crontab-entry

Here 21 means minute, 16 is the hour, * means everyday of the week, 10 means october, * means everyday
At the given schedule our system will create a crontab-entry file at the specified time.

18) at command

at command is like crontab which allows you to schedule jobs but only once.
When the command is run it wwill enter interactive mode and you can get out by pressing Ctrl D

Usage:
at HH:MM PM = Schedule a job
atq = List the at entries
atrm # = Remove at entry (# means entry number which can be obtained by atq command)
atd = at daemon/service that manages the scheduling.
systemctl status atd = To manage the atd service.

* Create at entry by scheduling a task:
at 4:45PM -> enter
echo "This is my first at entry" > at-entry
Ctrl D

Example:
at 5:05PM
at> echo "This is my first at entry" > at-entry #After entering at command it will go to the at> mode.
Ctrl D

It will create a at-entry file in our directory at the specified time.

Other future scheduling format
at 2:45 AM 101621 = schedule a job to run on october 16th, 2021 at 2:45am
at 4PM + 4 days = Schedule a job at 4pm four days from now.
at now +5 hours = Schedule a job to run five hours from now.
at 8:00 AM Sun = Schedule a job to 8am on coming sunday.
at 10:00 AM next month = Schedule a job to 10am next month.

19) Additional cronjobs (hourly, daily, weekly, monthly)

By default their are four different types of cronjobs :- hourly, daily, weekly, monthly
*All the above crons are setup in
/etc/cron.__(directory)
*The timing for each are set in
/etc/anacrontab -- except hourly
*For hourly
/etc/cron.d/0hourly

20) Process management

Background = Ctrl-z, jobs and bg
Foreground = fg
Run process even after exit = nohup process &
OR = nohup process > /dev/null 2>&1 &
Kill a process by name = pkill
Process priority = nice (e.g. nice -n 5 process)

*The niceness scale goes from -20 to 19. The lower the number more priority that task gets.
Process monitor = top
List process = ps

Example:-
sleep 100
Ctrl + Z
jobs # It will list the current jobs
bg # It will keep on running our jobs in the background.
fg # It will show our job in the terminal itself
nohup sleep 75 # It is running job in background and will not impact our terminal.
nohup sleep 73 > dev/null 2>&1 & #It will send all the messages to specified file.
nice -n 5 sleep 10 # It sets the priority.

21) System Monitoring Command

top, df, dmesg, iostat 1, netstat, free, cat /proc/cpuinfo, cat /proc/meminfo

df # This will report disk usage
dmesg # This will show the error messages if any or remaining infos.
iostat 1 # It will show input output stat every 1 second or any desired time, To quit out of it press CTRL C.
netstat -rnv # It will show the info about all the connections, subnet mask, required connected devices etc.
cat /proc/cpuinfo # Used to read all CPU info.
cat /proc/meminfo # Used to read all memory info.

21) System Logs Monitor (/var/log)

Another and most important way of system administration is log monitor
Log Directory = /var/log
boot
chronyd = NTP
cron
maillog

secure # tail -f secure secure will keep on running an let us know if anyone logged in or put any wrong password.
messages
httpd

To view a log use the 'more' command.'
To exit from log file press Ctrl + C

22) System Maintanence Commands(shutdown, init, reboot, halt)

shutdown
init 0-6 #It wil set time for reboot
reboot
halt

23) Changing System Hostname(hostnamectl) # We should be root

Command:-
hostnamectl set-hostname newhostname
Example:- hostnamectl set-hostname mrserosin
After changing reboot your system

File used to save hostname
Edit /etc/hostname or /etc/sysconfig/network

24) Finding System Information (uname, dmidecode)

cat /etc/redhat-release # It will tell what we are using

uname -a # Use to tell about version, name, kernel version, time architecture etc.

We need to be root
dmidecode # It will tell about bios, virtualbox, rom size, memory info, manufacture info, version etc.

25) Finding System Architecture.(arch)

Differences between a 32-bit and 64-bit CPU.

The big difference between them is the number of calculation per second they can perform.
Multiple cores allow for an increased number of calculations per second that can be performed

Linux = arch

26) Terminal Control keys

Several key combinations on your keyboard usually have a special effect on the terminal.
These "control"(CTRL) keys are accomplised by holding the CTRL key while typing the second key.
The most common control keys are listed below:
CTRL u = Erase everything you've typed on the command line.
CTRL c = strop/kill a command
CTRL z = Suspends a command
CTRL d = Exits from an interactive command

27) Terminal Commands (clear, exit, script)

clear = Clears your screen
exit = Exits out of the shell or a user session.
script = This command stores terminal activity in a log file and name can be set by the user.
Ex:-
script mylogfile.log
--After that whatever you/or any command you use will be recorded in your logfile.--
exit #To exit from the script log

more mylogfile.log # To view the log file.

28)Recover Root Password (single usermode)

* Restart your computer
* Edit grub
press e, search for ro remove it, then write rw init=/sysroot/bin/sh
ctrl x
chroot

passwd root
exit

* Change password
* Reboot

29) SOS Report

What is SOS Report?
Ans.Collect and package diagnostic and support data
Package name = sos-version
Command = sosreport

30) Environment Variables

What are environment variables?
Ans. An environment variables is a dynamic-named value that can affect the way running processes will behave on a computer.
They are part of the environment in which a process runs.
In simple words, set of defined rules and values to build an environment.

To view all environment variables
*Commands
printenv OR env

*To view one enevironment variable
echo $SHELL

*To set the environment variables
export TEST=1
echo $TEST

*To set environment variable permanently
vi .bashrc
Test='123'
export TEST

To set global environment variable permanently
vi /etc/profile or /etc/bashrc
Test='123'

export TEST

31) Special Permissions with setupid, setgid, and stickbit
All Permissions on a file or directory are reffered as bits.
-rwxrwxrwx #First rwx are user, next rwx are groups and third rwx are others.
all are reffered as bits.

There are 3 additional permissions in Linux:
setuid: bit tells Linux to run a program with effective userid of the owner instead of the executor(ex. passwd command) /etc/shadow.
setgid: bit tells Linux to run a program with effective group id of the owner instead of the executor(ex. locate or wall command)
NOTE: This bit is present for only files which have executables permissions.
sticky bit: a bit set on files/directories that allows only the owner or root to delete those files.

To assign special permissions at the user level
chmod u+s xyz.sh

To assign special permission at the group level
chmod g+s xyz.sh

To remove special permission at the user or group level
chmod u-s xyz.sh
chmod g-s xyz.sh

To find all executables in Linux with setupid and setgid permission
find / -perm /6000 -type f

Add a sticky bit :-
chmod +t /allinone (Other user cannot delete that directory even if it has all permission.

----------Module 6(Shell Scripting)----------

1) Linux Kernel
What is a Kernel?

Ans.It is an interface between hardware and software.

2) What is a Shell.
Ans. It is like a container. It is interface between users and Kernel/OS. Ex. CLI is a shell.
to find your shell we can use echo $0, Available shells "cat/etc/shells", Your Shell?/etc/passwd
Windows GUI os a shell, Linux KDE GUI is a shell, Linux sh, bash etc. is a shell.

3) Types of Linux-Shells.
Ans. Gnome, KDE, sh, bash, csh and tcsh, ksh.

4) Shell Scripting
Q. What is a Shell Script
Ans. A shell script is an executable file containing multiple shell commands that are executed
sequentially. The file can contain:
* Shell (#!/bin/bash)
* Comments (# comments)
* Commands (echo, cp, grep etc) # go to the editor after the bash command then write echo "Hello World"
* Statements (if, while, for etc.)

Shell script should have executable permissions(e.g. -rwxr-xr-x)
Shell script has to be called from absolute path(e.g. /home/userdir/script.bash)
If called from current location then ./script.bash

5) Basic Shell Scripts

Output to screen using "echo"
Creating tasks
Telling your id, current location, your files/directories, system info
Creating files or directories
Output to a file ">"
Filters/Text processors through scripts(cut, awk, grep, sort, uniq, wc)

Ex-
* Go to the vi editor

```
#!/bin/bash
#Define Small tasks
whoami
echo                    # Here echo command is used to give space
between the line.
pwd
echo
hostname
echo
ls -ltr
echo
```

* Defining Variables

input:-

```
#!/bin/bash
# Example of Defining Variable
a=Sumit
b=Singh
c='Linux Class'                    --When their is a space in your
word put single quotes.
echo "My first name is $a"
echo "My last name is $b"
echo "I am studying $c"
```

output:-

```
My first name is Sumit
My last name is Singh
I am studying Linux Class
```

6) Input and Output of Script

```
read
echo
```

input:-

```
#!/bin/bash
echo Hello, My name is Sumit Singh
echo
echo What is your name?
```

read namecontainer --- read is used to take info from user, and we have taken its variable name as namecontainer
echo
echo Hello $namecontainer
echo

output:- Another Example
What is your name? input:-
Jerry Seinfield #!/bin/bash
a=`hostname`
Hello Jerry Seinfield echo Hello, My server name is $a

output:-
Hello, My server name is LinuxFirstVM

another example:-
#!/bin/bash
echo What is your name?
read a
echo What are you studying
read b
echo Which is your favourite show?
read c
echo Hello, Mr $a
echo
echo But Medical is better than $b
echo
echo Really $c is your favourite show?

7) if-then Scripts

If then statement
If this happens = do this
Otherwise = do that

input:-

#!/bin/bash
count=100
if [$count -eq 100] -- Here -eq means equals

```
then
echo Count is 100
else
echo Count is not 100
fi
```

output:-
Count is 100

another example
input:-

```
#!/bin/bash
clear
if [ -e /home/iafzal/error.txt]          # -e is option which checks if a file
exist or not
then
echo "File Exist"
else
echo "File does not Exist"
fi
```

output:-
File does not Exist   # Because their is no file as
error.txt

8) for-loop Scripts

It keeps running until specified number of variables.
e.g. variable = 10 then run the script 10 times
OR
variable = green, blue, red (then run the script 3 times for each color)

Example Programs

input:-

```
#!/bin/bash
for i in 1 2 3 4 5
do
echo "Welcome $i times"
```

output:-

```
Welcome 1 times
 Welcome 2 times
Welcome 3 times
Welcome 4 times
```

done Welcome 5 times

input:- output:-

```
#!/bin/bash                    See Imran eat
for i in eat run jump play     See Imran run
do                             See Imran jump
echo See Imran $i              See Imran play
done
```

9) do-while Scripts

The while statement continually executes a block of statements while a particular condition is true or met.

e.g. Run a script until 2pm

```
while [ condition ]
do
command1
command2
command3
done
```

input:- output:-

```
#!/bin/bash                              10 seconds left to stop this
process
count=0                                  9 seconds left to stop this process
num=10                                   .
while [ $count -lt 10 ]                                              .
do                              .
echo                                     1 seconds left to stop this process
echo $num seconds left to stop this process $1       process is stopped!!!
echo
sleep 1
num=`expr $num - 1`
count=`expr $count + 1`
done
echo
echo $1 process is stopped!!!
```

echo

10) Case Statement Scripts

If option a is selected = do this
If option b is selected = do this
If option c is selected = do this

e.g. run "df -h"
Command

input:-

```
#!/bin/bash
echo
echo Please chose one of the options below
echo
echo 'a = Display Date and Time'
echo 'b = List file and directories'
echo 'c = List users logged in'
echo 'd = Check System uptime'
echo
read choices
case $choices in
a) date;;
b) ls;;
c) who;;
d) uptime;;
*) echo Invalid choice - Bye.
esac
```

output:-
will give the required output according to choices.

11) Check Remote Servers Connectivity.

A script to check the status of remote hosts.
Check the document ping-hosts.txt

```
#!/bin/bash
```

```
ping -c1 192.168.1.1
if [ $? -eq 0 ]
then
echo OK
else
echo NOT OK
fi
```

Change the IP to 192.168.1.235

Check the document ping-hosts.txt

12) Aliases (alias)

Aliases is a very popular command that is used to cut down on lengthy and repetitive commands

```
alias ls="ls -al"
alias pl="pwd; ls"                    # Here pwd and ls both will work in the command pl
alias tell="whoami; hostname; pwd" # Here tell command with give whoami, hostname and pwd.
alias dir="ls -l | grep ^d"
alias lmar="ls -l | grep Mar"
alias wpa= "chmod a+w"
alias d="df -h | awk '{print \$6}' | cut -cl-4"
```

Alias command will used to assign long commands and more than one command in a single alias or name such as ls, pl, tell etc.

To know the created aliases we can use the command alias.
To remove the alias we can use the command = unalias dir (unalias aliasname)

13) User and Global Aliases

Creating User or Global Aliases
* User = Applies only to a specific user profile
* Global = Applies to everyone who has account on the system.
* User = /home/user/.bashrc # User aliases
* Global = /etc/bashrc # Global aliases

To edit user aliases go to the required folder and edit using the vi editor.

14) Shell History (history)

All Commands are recorded, Very effective command to troubleshoot.

Command:-
history # Check every command we ran in the system.

To find something !406 or any number

The file where history of yur shell commands saved = /home/yourname/.bash_history