LINUX CLASS-2

FILE COMMANDS:

- touch filename : used to create a file
- touch aws azure gcp : used to create multiple files
- touch linux{1..5} : this command will create 5 files (linux1,inux2, linux3 .... linux5)

- rm filename : remove file with permissions (rm means remove)
- rm -f filename : remove file without permissions (-f means forcefully)
- rm -f aws azure gpc : remove multiple files without permissions
- rm -f linux{1..5} : this command will remove linux files from 1 to 5 without permissions
- rm -f * : used to delete all files
- rm -f *.txt : used to delete all files with .txt extension
- rm -f a* : this command will delete all files which are stared with “a” letter

FOLDER COMMANDS:

- mkdir foldername : used to create a folder
- mkdir git maven jenkins : used to create multiple folders
- mkdir docker{1..5} : used to create 5 folders at the same time

- rmdir foldername : used to remove empty folder
- rmdir git maven jenkins : will remove multiple empty folders
- rmdir docker{1..5} : used to remove 5 docker empty folders
- rmdir * : used to remove all empty folder
- rm -rf * : used to remove all files & folders and also it is used to remove non empty folders

LIST THE FILES:

ll : used to get list of files

ls : used to get list of files

ll vs ls

ll : will give the full info about files/folders

ls : it will give only file/folder names

ll -t : will gives the files based on time

ll -r : will gives the files on reverse order

ll -a : used to show hidden files

CHANGE DIRECTORY:

- cd foldername : used to change directory
- cd : used to go to root directory
- cd - : used to go to previous folder
- cd ../ : used to go to one folder back
- cd ../../ : used to go to 2 folders back

DIRECTORY COMMANDS:

- mkdir folder1/folder2 : this will creates folder2 inside folder1
- ll folder1 : used to get list of files & folder which are present in folder1
- touch folder1/aws.txt : used to create file inside a folder
- mkdir -p aws/azure/gcp/ccit : used to create parenting folder (folder inside the folder) automatically

COPY COMMAND: used to copy

SYNTAX: cp source destination

command: cp file1 file2

By the above command, the data from file1 copies into file2. But the problem is it will overwrite the data which are present in file2.

To overcome this issue we will use cat command.

cat source_file (file1)>> destination_file (file2)

MOVE COMMAND:
SYNTAX: mv source destination

COMMAND : mv file1 file2
This is also called renaming a file

VIM EDITOR: It is used to edit the files in linux machines, It has 3 modes

1. Command mode
2. Insert mode
3. Save & quit mode

To open any file in vim editor : vim filename (or) vi filename

1. **COMMAND MODE**: this is the default mode in vim editor, It is used to perform some actions like used to copy the data, delete the data and we can make undo and redo the changes as well.

gg : used to go to 1st line of the file

G : used to go to last line of the file

M : used to go to middle of the file

4gg : used to go to 4th line of the file

17gg : used to go to 17th line of the file

:23 : used to go to 23rd line of the file

:set number : used to set numbers of the file

yy : used to copy the line

4yy : used to copy 4 lines from our cursor

p : paste the copied content

10p : paste the copied content 10 times

dd : used to delete the line

5dd : used to delete 5 lines from the cursor

u : used to undo the changes

crtl + r : used to redo the changes.

/word : used to search for a word in a file

?word : used to search for a word in a file

:%s/old/new/ : used to replace

2. INSERT MODE: This mode is used to insert the data or make any modifications in a file.

To go to insert mode : **i**

To go back to command mode : **esc**

To go to the ending of the line : **A**

To go to starting of this line : **I**

To create a new line above the cursor : **O**

To create a new line below the cursor : **o**

DIFFERENCE B/W COMMAND MODE KEYS & INSERT MODE.

If we perform command mode keys we will be in command mode only

If we perform insert mode keys we will go to insert mode.

3. **SAVE & QUIT MODE**: This is used to save the data and quit from vim editor

To save the data - **:w**

To quit from vim editor - **:q**

To quit forcefully - **:q!**

To save & quit at a time - **:wq**

To save & quit forcefully at a time - **:wq!**

USER COMMANDS:

- to create user : useradd username
- to see all users : getent passwd (or) cat /etc/passwd
- to check a specific user : id username
- to set a password to user : passwd username
- to switch to user : su - username
- to delete a user : userdel username (this will delete only user & group)
- to delete a user : userdel -r username (this will delete all user, group & path also)

NOTE:

1. WHENEVER WE CREATE A USER THEN AUTOMATICALLY **GROUP** WILL BE CREATE.
2. WHENEVER WE CRATE A USER, THEN AUTOMATICALLY **PATH** WILL BE CREATE.
3. WE CANT DELETE THE GROUPS DIRECTLY WHICH ARE CREATED BY THE USERS, IF WE DELETE USER THEN AUTOMATICALLY GROUP WILL BE DELETE

UNDERSTAND THE USER:

shiva:x:1001:1001::/home/shiva:/bin/bash

shiva == username

x == it stores the users password

1001 == UID (User ID)

1001 == GID (Group ID)

/home/shiva == this is user path

/bin/bash = users shell path

GROUP COMMANDS:

to create a group : groupadd groupname

to see the list of groups : getent group (or) cat /etc/group

to delete a group : groupdel group-name

to add a user to a group : usermod -a -G group-name user-name