**BASH SCRIPTING**

Bash is a scripting language which allows users to automate, execute commands and manage system operations. To create a shell script the extension should be **.sh**

* Commands for shell scripting

1. **Touch**: - to create a file
2. **Vi/vim**: - To open the file if already created. If it is not created already then it will create a file and opens it
3. **Ls**: - To list the files
4. **Ls -ltr**: - to list the files with time stamp
5. **Man**: - Describes the functionality of a command
6. **Cat**: - It is used to print the content of the file without opening the file
7. **Sh filename (or) ./filename: -** It is used to execute the file. Which gives the output of the file
8. **Chmod:** - It is useful to grant the permissions of the users. **Ch 🡪 change**
9. **History: -** It is used to get the list of commands that we have used so far
10. **pwd:** - It is used to know in which directory we are at. Stands for **present working directoty**
11. **mkdir: -** It is used to create a directory**(folder).** Stands for **make directory**
12. **cd: -** It is used to go into the directory**(folder).** Stands for change directory
13. **cd .**. :- It is used to get back from the directory
14. **rm -rf: -** It is used to delete a directory
15. **nproc:** - It is used to list the cpu’s on the machine
16. **free -g:** - It is used to get the ram details
17. **top:** - used to get the list of processes that are running on the machine. Used to analyze the node status
18. **df -h:** - used to get the machine detail. Disc space
19. **set -x**: - debug mode in the shell script. It will print the output along with the command given
20. **set -e**: - it exits the script if there is an error. Without this command even if there is an error in the 1st line the script will execute, if we use this it will fail to execute the script at line 1 itself.
21. **Set -o pipefail**: - it is used along with **set -e**. because without using this, if there is an error in a line with “|” at the end of the line it wont stop the execution.
22. **ps -ef**: - used to get the processes details running in the machine
23. **| grep “ ”**: - **used to search for specific patterns of text**
24. **| :-** it is a pipe command. Used to send the output of the first command to the second command
25. **Date: -** used to get today’s date and time
26. **Awk -f: -** used to get the single column information.

**EX:- ps -ef |grep amazon | awk -f “ ” ‘{print $2}’ 🡪 it prints the second column**

1. **Curl: -** it will retrieve the information from the internet. **Curl https://...........**
2. **Wget: -** it will retrieve the information from the internet and store in a file. If we want to see the content we need to open the file, but curl directly shows the output without storing it in any files
3. **Sudo su -: -** used to go to root user. **Sudo 🡪 substitute user do, su 🡪 switch user**
4. **Find: -** used to find the location of a file. **EX: - sudo find / -name pam**
5. **Trap: -** it is used to trap the manual signals
6. **Kill: -** used to kill a process. **Ex: - kill -9 $process**

* Shell scripting must start with **#!/bin/bash** 🡪 **(#!/) it is called as shebang**
* **Bash/dash/sh/ksh** 🡪 **these are called executables**
* **Previously sh will automatically redirect to bash using linking concept, but now sh is redirecting to dash as default**
* **Echo: - It is used to print something**
* **To write something in a file** first we need to go to insert mode, we can achieve that by **(click on esc then click on i)**
* **To save the file** we need to **(click on esc and enter :wq!)**
* **To exit from the file without saving** we need to **(click on esc and enter :q!)**
* **:wq! 🡪 to save the file and exit**
* **:q! 🡪 exit from the file without saving**
* **Chmod** has 3 categories

1. What are the permissions for a root user **(you)**
2. Which group has access **(your group)**
3. What are your permissions **(all users)**

**Linux uses chmod as 4, 2, 1 manner which says**

**4 as read**

**2 as write**

**1 as execute**

* **If we give chmod 777 then you, your group and all users has permission for read, write and execute the file**
* **If we give chmod 774 then you and your group has permission to read, write and execute but for all users it will be only read permission**
* **# 🡪** it is used to write the comments in shell script
* **Date | echo “Today’s date”** 🡪 this will give the output as today’s date. Because date is a system default command and it sends the output to stdin. But pipe (|) doesn’t have the ability to receive the info from stdin, it only takes from stdout.

**If else in shell scripting**

If [ condition ]

Then

Statement

Elif [ condition ]

Then

statement

Else

Statement

Fi

**For loop in shell scripting**

For condition

Do

Script

done