**Linux**

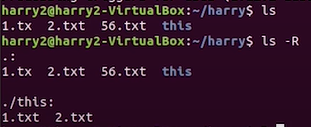
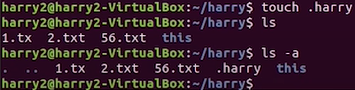
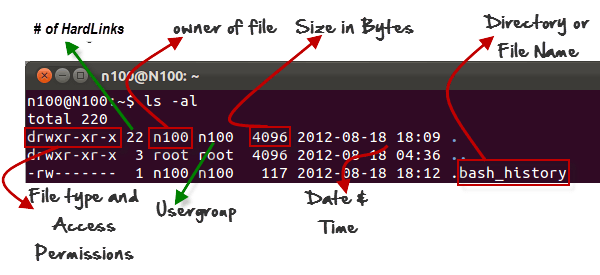
**Jaise Windows me CMD h same way Linux has Terminal for CLI**

C:\Users\Dharam\Downloads\WhatsApp Image 2024-02-12 at 10.39.38 PM.jpeg

Here, **dharamveer** -> username **dharamveer-VirtualBox** -> name of the machine **$** -> regular user (Aage dekhenge) **:~$ ke bhich me current working directory hota h**

**Note: In Linux, All Files are stored in a Tree format. / is the root node/directory.**

**File System Operations**

* **pwd** Shows absolute path of the working directory
* **ls** List files/folders present in the current directory
* **cd subdir** Change directory to subdir
* **cd ..** Go up a directory (Eg, Dharam/java8 me hu cd .. karu to Dharam directory me aa jayega)
* **mkdir dir** Make directory dir
* **touch fileName** Creates a file fileName in the working directory.
* **mv fileName dirName/** Move fileName file to dirName directory. Eg,here moved 1.txt file to **this** directory which is a directory inside the working directory. We can also provide absolute path of the directory agar vo working directory me nhi h to (Eg **cp** me h)
* **cp fileName dirName/** Move fileName file to dirName directory. Eg with abs path
* **ls -R** List whatever is present in the working directory and if subdir present uske andar kya h and uske andar bhi subdir to usme kya h and so on. Full form List Recursively. 
* **touch .fileName** Creates a hidden file named fileName. Ab agr ls kare to hidden file nhi dikhega
* **ls -a** List directories along with hidden files
* **ls -l**  Gives the detail of the creates Files and Folders.

* **ls -r**: It lists files and directories in reverse order. **ls** command ke order ko ulta kr deta h.
* **ls -t**: It lists files and directories, sorting them based on their modification time, with the newest ones appearing first.
* **rm \***:deletes whatever is present in the current directory
* **clear**: terminal clear
* **history**: shows all the commands ran so far.

**Printing on the terminal**



**Users**

**3 types of Users**

Consider 3 users U1,U2,U3

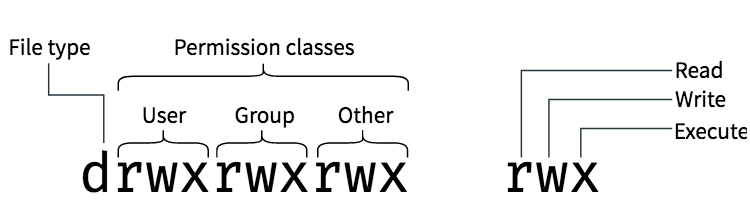
1. **Regular User:** It has it’s own home directory and can perform anything in it’s own home directory but not of other users present in the system. For eg, are three users U1 U2 me khuch nhi krta samjh gaya hoga.
2. **Root User/Super User:** This user can access and peform anything on desired user.
3. **Service User:** vata imp nhi

**sudo su**

**User Permission**

ls -l command run kiya to

****aisa khuch dikhta h which shows upar description h ek ek chiz ka



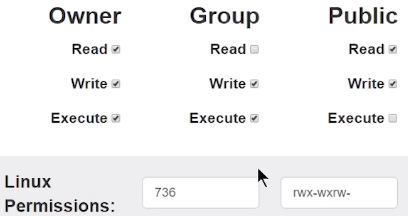
**d** meaning directory, **-** raha to vo file denote krta h

**Changing User Permission**

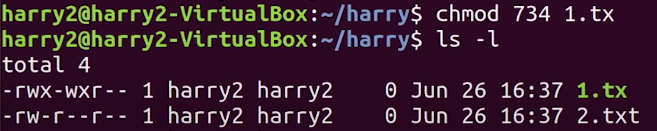
**chmod** command is used to change permission

**chmod number file/directory**: changes file/directory permission

**Calculating the number**

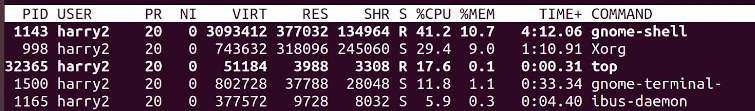
****Consider Ticked box as 1 and unticked as 0

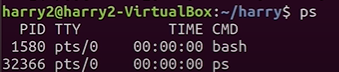
For Owner here it’s 111 ye binary number mila. Convert it in decimal we get 7 same for Group we get 3 and for public/Others it’s 6

If ran **chmod 736 fileName** we can change the user permission for the mentioned file.

**Similary we can use chgrp command to change the group of a file/directory.**

**Few other commands**

**top:** Lists processes which are consuming resources in descending order 

**ps**: shows running processes

**kill processId:** stops the mentioned process basically firebox open h to usko close krne ke liye CLI way hai kill command.

**Editing a file**

Install vim text editor first

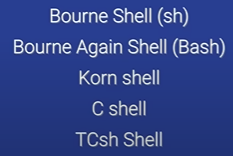
1. Hit command **vim fileName**
2. File open ho jayega first type **i** to type something in the file.
3. Press **Esc** then **:q!** for unsave and exit, **Esc** then **:wq** for save and exit.

**Shell Scripting**

Writing set of commands in a file and executing the file results in sequentially run of all the commands is termed as Shell Scripting

Assume koi task baar baar krna h repeatedly saare commands baar baar run krne se better script bana lo and jb need ho execute karo. We can aslo schedule (kis time pe execute krna h) a script too.



Bahut types of shells are present

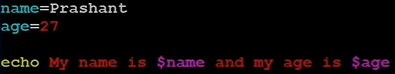
 : shows which shell are using.

**Creating a Script**

1. Open a file using **vim scriptName** command.
2. First write **#!/bin/bash** (Bash is a Shell and jb script execute krte h to ye linux ko batata padta h apn ye shell use kr rahe h)
3. Phir jo commands likhna h likh le quit and save krle
4. To excute a script make sure user has execute permission (rwx). Nhi raha permission to Access denied batayega execute krne gaye to.
5. Giving user execute permission command : **chmod u+x scriptName (**Here u -> user, x- > execute)
6. Ab execute kr sakte h, command -> **./scriptName**

**Note:** Better use extension **.sh** to differentiate between Normal file and a Script. Eg script1.sh. Due to this linux bhi samjh jaata h shell create rahe h and Code Editor jaise text different colors me likhata h

**Syntax for Shell Scripting**

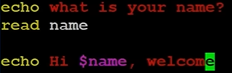
**Variable Intialization and printing variable**

****

**Constant Variable** ab apn var\_name ki value change nhi kr sakte.

**Storing a Command in a variable**

**Output ->** Note: Prashant was the hostname for youtuber bhai

**Taking input from the user**

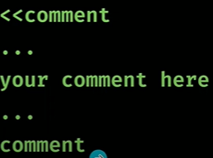
Another way here NAME is the variable

****

**Printing Script Name**

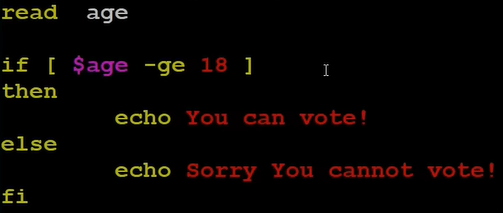
**Comments**

****

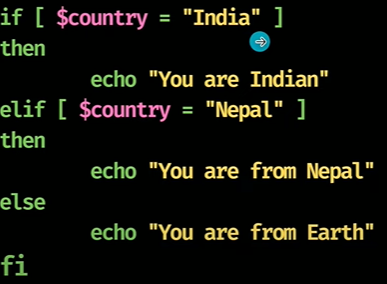
****

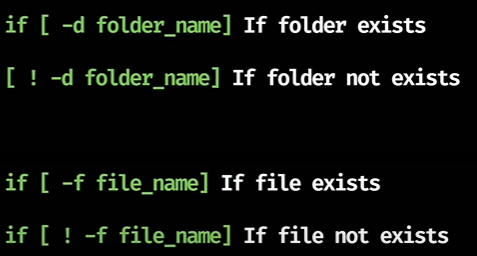
**Types of Operator**



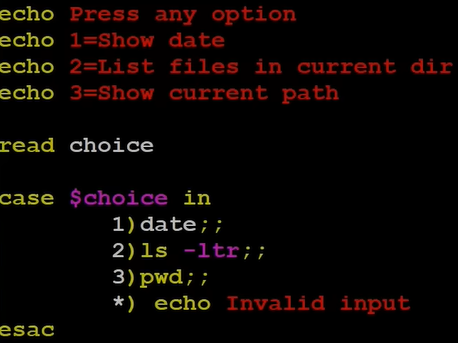
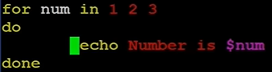
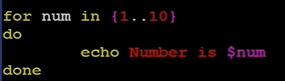
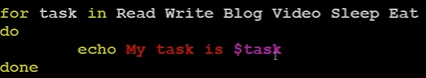


**elif**



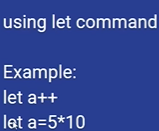


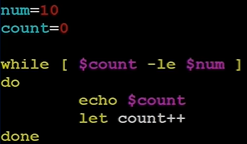
Commands to execute

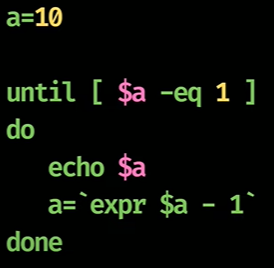
In shell scripting, the let command is often used for arithmetic operations, including incrementing and decrementing variables. The let command allows you to perform arithmetic operations directly within the shell script

Eg of let command



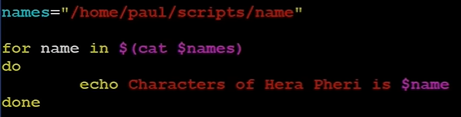


Until loop opposite of while loop i.e jb tk condition true na ho jaye tb tk loop chala



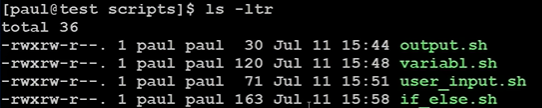
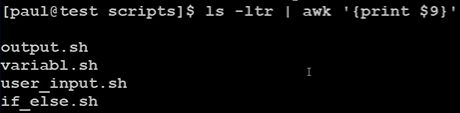
We can also use continue and break keywords here bss keyword likhde and kaam khatam.

Iterating values from External Files using For Loop

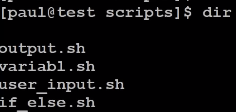


File name name



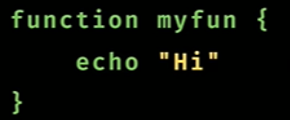
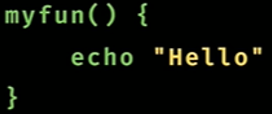
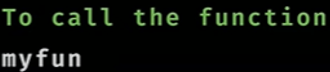
$ sign variable print krne ke kaam aata h yaha \ lagaya taaki batane ke liye ki $ bss part of the command h koi variable print krne ka try nhi kr rahe hm. Just like C/C++/Java



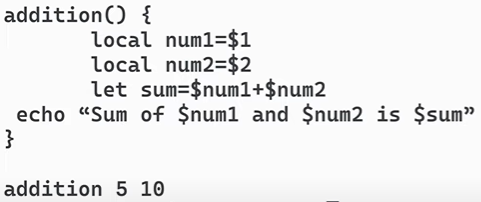
./fileName: Mtlb current directory ka hi koi script h

directoryName/fileName: me apn bata rahe h ye waale directory me dhundo to apn directoryName ke parent ke parent directory me bhi kyu na ho we can still execute desired file

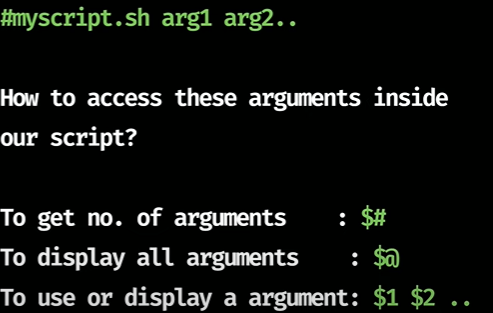
**Function**

**** **** 

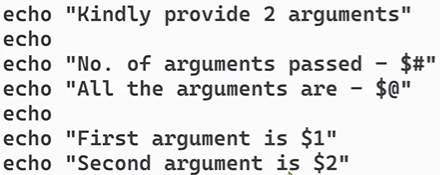
**Function with paramaters**

****

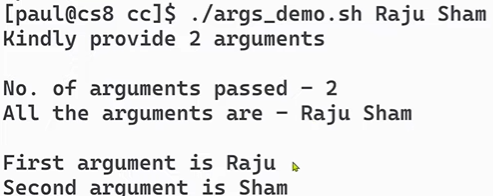
**Arguments in Script**

****

**Code**

****

**Passing the arguments along with execution of the script**

****

** Jaha se bhi debugging shuru krna h vaha ye likh lo**