What is typeset in linux ?

Basic Ubuntu Commands For Beginner:

$ : dollar sign used for regular user

# : pound sign used for sudo user ( note sudo user is root user )

Sudo : (superuser DO) :this is useful when , for example we need to modify files in a directory that our user wouldn’t noramally have access to . It provides admistrative privileges just like, Run as admisnistrator .

1.apt-get : It is used to install ,update , upgrade , and remove any package : a. sudo apt-get update

b.sudo apt-get upgrade c. sudo apt-get install d.sudo apt-get remove e. sudo apt-get purge : it removes the software completely . f. sudo apt-get autoremove : it is used to remove unwanted software after uninstlling a package to remove .

2. ls : provides lists all files and folders in current working directory .

3.cd : change directory :-> cd / : takes to the root directory , cd .. = takes to paprent directory

Cd - = takes to the previous directoy

4. pwd : full path name of the current working directory

5. cp = it copies a file . 6. mv = move 7. rm = remove a. rmdir = remove an empty directory

b.rm -r = remove recursively , removes a directory along with its content .

8. mkdir = it creates directory ( folder) 9. History = it gives previous command what we used .

9. df = display information about the disk space usage of all mounted filesystem ,

10. du (directory usage) = command displays the size of a directory and all of its subdirctories .

11.Uname -a = provides a wide range of basic information about the system .

12.free – display the amount of free space available on the system .

13. top = Display the processes using the most system resources at any given time .

14 . q = it can be used to exit .

15. man man = provides information about the manual itself .

16.man intro = displays a bried introduction to linux commands

17. info = similar to man but often proves more detailed or prepcise information .

18. passwd = it is used to change user password using Terminal .

19. whatis = whatis command shows a brief description of what is the functionality of specific built in linux command . eg : whatis cd , whatis man , whatis man

20. find /etc -type f ( it will give all the file under etc folder)

Ubuntu Terminal Shortcuts :

Ctrl +Shift +T = open new tab on current terminal

Ctrl+Shift+w = clost the current tab

Ctrl+A = move cursor to beginning of line

Ctrl +E = move cursor to end of line

Ctrl+U = clears the entire current line

Ctrl +k = clears the from the cursor Right

Ctrl+W = delete the word before the cursor

Ctrl+R = allows to search history for commands matching what I have typed

Ctrl+C = kill the current process

Crtl +Z = suspend the current process by sending the signal SIGSTOP

Ctrl +l = clears the terminal output

Alt +F = move forward one word

Alt + B = move backward one word

Ctrl+Shift+c = copy the highlighted command to the clipboard

Ctrl+shift+v or shift+insert = paste the contents of the clipboard

Up / down arrow keys = to scroll through command history and perform quickly

Tab = used to complete the command which we are typing

File creation :

1. If we do : touch .shambhu --- > it will create file Shambhu which is not seen in ls command
2. If we do : ls -a -- > it will help to show even hidden file
3. If we do : history -- > it will give all list of command we used in terminal
4. Ctrl+shift+O -- > to make terminal big
5. Chmod -- > this is the command to change permission , for this we can search in google “chmod calculator” which gives idea about chmod used Number like 400 or 700 etc .
6. If we do : top -- > it will give resource consumed hierarchy
7. If we do : ps and ps -a --- > it will give all the process running list
8. If we do : kill pid -- > it will kill given process id (pid can be found when we use “ps or ps -a” command)
9. How this chmod calculator work : -- > there is : read , write and execute permission in file . and each user (user , group , and others) may have all rwe permission in file . so we start for USER : if we gave rwx permission to USER then it becomes 111 . if we gave only rw permission then it became 110 . So here we give 0 to x because we are not giving permission to execute . Now we convert this binary to integer or decimal (use google to convert binary to integer) . Then after converting binary , we get one number whether it is 4 or 5 or 6 etc . So this number is for USER . Likewise we do the same for group and may ger 5 or 4 or 7 etc and get one single number after converting binary to Integer or decimal .Likewise we do same for OTHERS and get single number .
10. So finally we may have 756 – meaning : 🡪 7 coming from USER , 5 coming from GROUP and 6 coming from OTHERs . in this way permission number works.

How to use alias in linux or Kubernetes?

Ans : go to : vi ~/.bashrc -- > write alias (alias a =”ls”) -- > save and quit

@Now we need to inform our terminal that we are using alias written in “bashrc”

@Write in terminal : source ~/.bashrc (now our terminal is aware about our alias or code written in bashrc)

How to change the owner of the file or folder in linux >( this is the good resources in google)

?

<https://linuxize.com/post/linux-chown-command/>

permission for linxu

1. Check how many useers are in linx : cat /etc/passwd ,

**Shell script: Redirecting stream**

The numbers are file descriptors and only the first three (starting with zero) have a standardized meaning:

0 - stdin

1 - stdout

2 - stderr

So each of these numbers in your command refer to a file descriptor. You can either redirect a file descriptor to a file with > or redirect it to another file descriptor with >&

The 3>&1 in your command line will create a new file descriptor and redirect it to 1 which is STDOUT. Now 1>&2 will redirect the file descriptor 1 to STDERR and 2>&3 will redirect file descriptor 2 to 3 which is STDOUT.

So basically you switched STDOUT and STDERR, these are the steps:

1. Create a new fd 3 and point it to the fd 1
2. Redirect file descriptor 1 to file descriptor 2. If we wouldn't have saved the file descriptor in 3 we would lose the target.
3. Redirect file descriptor 2 to file descriptor 3. Now file descriptors one and two are switched.

Now if the program prints something to the file descriptor 1, it will be printed to the file descriptor 2 and vice versa.

EG. Command:

Find /etc -type f 2> /dev/null (here stderr(2) is redirected to /dev/null which ignores all stderr in this directory)

**Sign**

**$?** -- > it will give the command success or not result : 0 - > success , 1 -- >

Eg. Use command first : ls -- > and use command : echo $? -- > 0 (zero) - > meaning command is success

* > We don’t need to mention 1 as STDOUT because it works by default

Eg. Ls -l > output.txt -- > output will be list of the file with long description

Ls -l >&2 -- > it means we are redirecting STDOUT to STDERR

We can do at one time to print both STDOUT and STDERR

Eg.