Linux

What is Linux.?

Just like Windows, iOS, and Mac OS, Linux is an operating system. In fact, one of the most popular platforms on the planet, Android, is powered by the Linux operating system. An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn’t function.

Putty:

Putty is a free and open source terminal emulator, serial console and network file transfer application.

* Download putty and install.
* Open putty with server IP and login with user id and password.
* You can also open the session by giving “username@IP address”.

[mkrishna@mytrainer ~ ] $

User name server name working disk prompt

Unix is multi user environment. You can login any time in the system.

* You can give spaces before and after the commands but not in the middle of the commands

**Changing Password/Creating Password:**

Command “**Passwd**” to change your own password

“$ passwd”

Linux commands have the following format

**$ command options arguments**

Ex: $ ls -l file

Here “ls” is the command, “-l” is the option and “file” is argument.

Options always starts with “- “(hyphen)

Ex: - $ ls -l -r file1 file2 file3

Here ls is command, -l is option1, -r is option2, file1 is arg1, file2 is ar2 and file3 is ar3.

* We can merge options. For example -l and -r can be write as “-lr”.
* We can’t merge arguments.
* The behaviour of command altered or changed by using option.
* When we write two options, we need to give space between them.
* If we want to merge two options we should give single hyphen(-) and write options without space.

**COMMANDS: -**

$ logname - to see our username

$ whoami - to see our username

$ exit

$ logout to exit terminal

<ctrl+d>

$ clear – to clear screen

$ cal – to open the calendar

$ passwd – to change the password

$ ls –

$ date – it will show the current date in your laptop

$ echo – it writes what follows to the screen

$ banner – it writes character strings in large letters to the screen

$ who – it shows who are logged into the system

$ who am I – it tells self-details like time and Ip address of the system and login time

$ w – it is giving uptime, how many users and servers load, user name tty, from which Ip address they have logged in, login time, idle time, JCPU, PCPU and what.

$ w -u – it gives data same as given by $who but it also gives login and logout time.

**Keyboard Shortcuts: -**

* <ctrl-c> - to terminate current command and return to the shell
* <ctrl-d> - end of transmission or end of file
* <ctrl-s> - Temporarily stops output to the screen
* <ctrl-q> - Resumes output which is stopped by <ctrl-s>
* <ctrl-u> - it erases entire line
* <ctrl-e> - moves the courser to the last character of command

**Server**

User screen 1

Pseudo terminal slave

Server system

User screen 2

Sessions

console

User screen 3

192.168.2.46

* Here server Ip address is192.168.2.46
* The server has its own system called as console
* These terminals were not connected with direct Ip servers, these terminals has individual Ip address. So, these were called as pseudo terminal slave(pts).

APPLICATION

SHELL

KERNEL

HW

CPU,MEMORY and Input/Output devices are System hardware.

**Kernel:**

Kernel is core operating system program. It is an important component which manages system hardware like cpu,memory,input output devices.

Kernel is stored in / booy directory and it name starts with vmlinuz-its version release number. You can see same with uname -r command.

**Shell:**

Shell is an interpreter that provides you an interface to execute commands. The default shell used by RHEL(Red Hat Enterprise Linux) is bash (borne again shell).

**bsh:** bourne shell

**tsh:** turbo shell

**csh:** cshell

**ksh**: Korn shell

note: bash is having automatic command completion by using tab key.

* Unix doesn’t have first name or second name it only has file name.

**Commands: -**

* $ cat (concatenation) 🡪 to print file content on the screen
* $ ls (long list) 🡪 it shows all the files present in the directory.
* $ ls -l 🡪 it shows all the content of the files present in the directory.
* $ wc (word count) 🡪 it shows number of lines, words, characters in the file
* $ wc -l 🡪 it shows number of lines.
* $ wc -w 🡪 it shows number of words
* $ wc -c 🡪 it shows number of characters.
* $ uname -r 🡪 is to know the release version
* $ uname -a 🡪 is to know all the details of the version
* $ hostname 🡪 it shows the server’s name
* $ head [FileName ]🡪 it print first few lines of file. By default first ten lines will be printed . if we want first three lines of file the syntax is

**$ head [-n] filename**

Here **n** is how many lines we want prepared

* $ tail 🡪 it pints last ten lines by default.

Syntsx: $ tail -n filename.

* $ mesg y 🡪 it is command which will give permission to interact with other users.
* $ mesg n 🡪 it deny the permission to interact with other users.
* $ write 🡪 it is used to chat or interact with other users.

Ex:

$ write username.

To end the interaction we can give <ctrl+c>. then it shows EOF to other user.

**$ sort filename:** it will sort the file data in to ascending order(A-Z)

**$ sort -r filename:** it gives tha file data in reverse order that is descending order(Z-A)

* **‘\’** back slashes having different meanings in the shell. For example

Ex: “Hi there\”

Here we opened double quotes and we closed beside the back slash with out any space, so here the command will not executed. The space should be given after the back slash (“Hi there\ “)

* Usually the sort command takes the strings. If we want to sort the numbers file or numerical file we should give ‘-n’. Here n is the numeric sort.

**$ sort -n number.txt :-**  it sorts the numbers in ascending order.

**$ sort -n -r number.txt** :- it sorts the numbers in descending order.

**$ more filename:** More command used to print the pages one by one. So you can read page by page. If you want to quit in between then click ‘q’.

**Date command with formatted output:**

**Syntax: $ date ‘+ %h’**

**Date command arguments**

a= Mon

A= Monday

b or h = jan

B= January

c = complete date format

C = century

d = date

m = month

y = year short form

Y = year full form

H = hours

M = minutes

S = seconds

%n = new line

Examples: $ date ‘+ %a’

$ date ‘+ %A’

$ date ‘+ %d-%m-%y’ (21-06-2023)

$ date ‘+%H:%M:%S’ (11:11:11)

* **%t** is used to tab space. Example $ date ‘+ %H%t%M%t%S’ it gives (11 30 53)

**$ history**: it is used to see history that what the commands we used previously.