**sshspecialLINUX**

**File sytem Structure of Linux**

The “File system structure” in linux refers to how files & directories are organized and stored on a linux based operating system. It defines how the file system is structured and how files and directories are arranged in a hierarchical manner. The os is made from nothing than a file.

File system includes files, directories and it’s related permissions. File system(fs) stores data in hierarchy of directories and files.

**Types of File systems –**

1. ext3
2. ext4
3. XFS
4. FAT
5. BTRFS

To check the file system of your OS in linux the command is “lsblk -f”. and to check the core directories and files of your system , enter “ls -lrth /”

**File system structure description :**

1. **/ (Root Directory)**:
   * The top-level directory from which all other directories stem.
   * Contains essential system files and directories.
2. **/bin (Binary Binaries)**:
   * Contains critical system binary executable files required for system recovery and maintenance.
3. **/boot**:
   * Holds files needed for the boot process, including the Linux kernel, initramfs, and bootloader configuration.
4. **/dev (Device Files)**:
   * Contains device files representing hardware devices, facilitating interaction with them.
5. **/etc (System Configuration)**:
   * Stores system-wide configuration files and scripts needed to boot the system and configure services.
6. **/home (User Home Directories)**:
   * Home directories for regular users are typically located here.
7. **/lib (Libraries)**:
   * Contains system libraries required for the operation of essential programs and the kernel.
8. **/media**:
   * Mount points for removable media, such as USB drives and optical discs.
9. **/mnt (Mount Points)**:
   * A directory for temporarily mounting file systems or devices.
10. **/opt (Optional)**:
    * Directory for installing additional software packages not included in the distribution's package management system.
11. **/proc (Process Information)**:
    * A virtual file system that provides information about running processes and system configuration.
12. **/root (Superuser Home Directory)**:
    * The home directory for the superuser (root).
13. **/run**:
    * Contains system runtime data, including PID files and UNIX domain sockets.
14. **/sbin (System Binaries)**:
    * Contains system binaries necessary for system administration and recovery.
15. **/srv (Service Data)**:
    * Contains data for services provided by the system.
16. **/tmp (Temporary Files)**:
    * A directory for temporary files that are typically deleted on system reboot.
17. **/usr (User Binaries and Data)**:
    * Contains user binaries, libraries, and data files for system-wide use.
18. **/var (Variable Data)**:
    * Contains variable data such as logs, spool files, and temporary files created by system processes.
19. **/mnt (Mount Points)**:
    * A directory for attaching temporary file systems.
20. **/lost+found**:
    * Used by the **fsck** (file system check) utility to recover files after an unclean shutdown or file system error.

**File commands**

1. **ls** – List files and directories in the current directories
2. **ls** -**ltr** \_ shows the files and directories in a reverse pattern , latest file will be on bottom, no hidden files in it
3. **ls -l** – shows the file and directory in a long format with a detailed list.
4. **Ls -lrth -** shows the file and directory in a long format with a detailed list, and also (r - frequently used files on top, t -sort the list by modification time, h- sizes of files more readable like,mb,kb)
5. **ls – la** – shows the hidden files also with the details ( hidden files always with the ., like .bashrc, .config)
6. **ls – a** -shows all the files including hidden files but no details in it
7. **pwd** – Print the working directory (absolute the path of the current directories you are in )
8. **cd** – change the current directory

(cd .. – previous Directory

Cd –parent directory )

1. **mkdir** – create a new directory
2. **cat > file name** – create a text file/ Overwrite the content of the file
3. **cat>**.file\_name – Create a hidden file
4. **rm** – ( rm -r delete the folder/directories

rmdir – Remove the empty directories only )

rm -delete individual files

1. **cat** – Concatenate, This command read the data from the text file and gives the content as a output
2. **zcat** – It is a useful tool for the displaying the content of compressed file without the decompression
3. **tac** – read the text file in reverse direction
4. **touch** – create an empty text file
5. **touch** .file.txt – Create a hidden file
6. **which –** which command is particularly useful for finding the locations of commonly used commands and utilities in the system.
7. **head** – Select the data from top 1st part like 10 lines
8. **tail** – Select the data from bottom part like 10 lines

( **head -n5 file name –** shows the top 5 lines

**tail -n3 file name** – shows bottom 3 lines)

1. **more/less** – used to view the context of text file in terminal
2. **find** – find the files , hidden and unhidden files from directory
3. **history** – shows the previous commands
4. **tail -f** – it records the real time logs in a file
5. **cp** – copy the files from one directory to another directory
6. **mv** - move the files/directory from one directory to another directory
7. **mv old\_filename\_txt new\_filename.txt** – Rename the file
8. **wc** – word count ( count the total lines, Words, and characters in a text file)
9. **wc -l filename.txt** – count the lines only
10. **wc -c filename.txt** – count the characaters only
11. **cut -c** - used to extract the selected data/lines of text from files.
12. **tee –** its allow you to capture the output of command or save it or process it in a more files while displaying in a terminal

**echo "Hello, world!" | tee output.txt**

1. **sort** – used to sort the data or text files alphabetically or numerically and much more

( **sort -r** – sort by reverse order

**Sort -n** – sort by numerical order)

1. **clear** – is used to clear the terminal screen
2. **dirs** – Shows your current directory
3. **stat -** The stat command in Linux is used to display detailed information about a file or directory, including its inode data, timestamps, file size, and permissions. It provides a comprehensive overview of the file's metadata.
4. **tty –** teletypewriter, This will show the current directory name.
5. **chattr –**

**chattr +a file.txt –**  This command will set ‘append only’ request to file bcz of which the file cannot be modified or overwritten

1. **grep** – used for searching and filtering text from the file

grep “text\_name” file\_name.txt

1. **diff** – differentiate two different files

( **diff – u** – used to diff codes or batch files

**diff -r** – compare directories and subdirectories

**diff -y** – side by side comparison

1. **ln** – used to create symbolic links or softlinks between files and directories

(**ln -s** – to create a soft link, once delete the original file the link will be also deleted

**Ln** – to create hardlink, once delete the original file the link will not be deleted)

1. **vi editor** – Visual editor/ Visual instrument

It’s a powerful and versatile text editor known for its efficiency and flexibility. Its operates in different modes , each with specific purpose.

Most common keys –

i – Insert

O – insert on next line

a – insert after a space

esc – escape out of any mode

:q! – quit without save

:wq! – Save and Quit

r – replace

d – delete

u – undo

x – remove one character

backspace –

/ - forward search

? – backward search

ggdG – clear the page

:line number – jump to line number

**Data Sizes Units –**

1. Bit (b): The smallest unit of data storage. A bit can represent either a 0 or a 1, which are the fundamental building blocks of digital information.
2. Byte (B): A byte consists of 8 bits and is the basic unit for representing characters and data in most computer systems. It is often used to measure file sizes and memory capacity.
3. Kilobyte (KB): Equal to 1,024 bytes (or 2^10 bytes). Kilobytes are commonly used to describe the size of small text files, images, or simple documents
4. Megabyte (MB): Equal to 1,024 kilobytes or 1,048,576 bytes (or 2^20 bytes). Megabytes are used for measuring the size of larger files, such as music tracks, photos, or short videos.
5. Gigabyte (GB): Equal to 1,024 megabytes or 1,073,741,824 bytes (or 2^30 bytes). Gigabytes are commonly used to measure the capacity of hard drives, flash drives, and the size of software applications.
6. Terabyte (TB): Equal to 1,024 gigabytes or 1,099,511,627,776 bytes (or 2^40 bytes). Terabytes are used for measuring the storage capacity of large hard drives and data servers.
7. Petabyte (PB): Equal to 1,024 terabytes or 1,125,899,906,842,624 bytes (or 2^50 bytes). Petabytes are used to represent the storage capacity of massive data centers and cloud storage services.
8. Exabyte (EB): Equal to 1,024 petabytes or 1,152,921,504,606,846,976 bytes (or 2^60 bytes). Exabytes are used to quantify the scale of very large datasets, including global data traffic and scientific research data.
9. Zettabyte (ZB): Equal to 1,024 exabytes or 1,180,591,620,717,411,303,424 bytes (or 2^70 bytes). Zettabytes are used to describe the vast amounts of data generated by modern technology, including the internet and large-scale data analytics.
10. Yottabyte (YB): Equal to 1,024 zettabytes or 1,208,925,819,614,629,174,706,176 bytes (or 2^80 bytes). Yottabytes are used in theoretical discussions of future data storage and data transmission capacities.

**File Permisson Command**

Let's break down file permissions and the `chmod` command in a simple way:

**File Permissions:**

**Chmod (change mode):** this command is used to change or modify the permissions of files or directories.

- In Linux, each file and directory has permissions that control who can do what with them.

- There are three types of permissions: **read, write**, and **execute.**

- These permissions are assigned to three categories of users: the **owner** of the file, the **group** associated with the file, and **others (everyone else)**.

Understanding Permission Numbers:

- Permissions can be represented by numbers.

- Each permission is given a numeric value: **read (4), write (2), and execute (1).**

- You add these values to set permissions. For example, read and write permissions together would be 4 + 2 = 6.

Changing Permissions with `chmod`:

- The `chmod` command allows you to change these permissions.

- You can use it to make a file readable, writable, or executable for different users.

To add the permission should use “+” and to remove use “-“

**Basic Usage of `chmod`:**

- To use `chmod`, you need to specify three things:

1. Who you want to change permissions for (the owner, the group, or others).

2. What you want to do (add or remove permissions).

3. Which permission you want to change (read, write, or execute).

Examples:

- To make a file readable for everyone, you'd use: `chmod a+r filename`.

- To give the owner permission to write, you'd use: `chmod u+w filename`.

- To make a script executable, you'd use: `chmod +x script.sh`.

(u – owner/user, g – group, o – others, a – all)

Remember, file permissions are crucial for security and control in Linux, so use them wisely to protect your files and directories.

**Umask :** This command is used to set the default permission of newly created files and directories by specifying which permission should be turned off.

* Umask value is represented in a octal (base 8) number
* Each digit represents the set of 3 permission bits for the owner, group and others
* 1st digit for 3 permissions of the owner(a)
* 2nd digit for 3 permission of the group(g)



* 3rd digit for 3 permission of the everyone(o)

Example – ( what is the permission in 0002

If umask value is 0002 then substract 0002 from 0777

(0777 – all permissions for a,g,o for Directory)

(0666 – all permission for the files, rxcept other)

0777 – 0002 = 0775 (rwxrwxr-x)

Now, lets understand the digits

with their permission,

The default permission of files will never be rwxrwxrwx bcz of security issue. Because the limit for file is 0666.

**chown – change ownership**, This command is used to change the ownership of file and directory

to change the ownership of the filr or directory process –

1. file should be available in both terminals of different users or you can create a file in 1 terminal through shared\_memory

2. now access the file with “sudo chown username filename”

3. with this command ownership will transferred to other user

4. now **chgrp (change group)** also, **same process**

5. this commands like a changing permissions of the file from 1 user to another user for security purposes.

**sudo adduser username** : add new user to terminal , enter the password 2 times and after that “Y”

now to login in that user , command **su username**

**password**

**sudo passwd root :**  add the root user to your terminal ( root user has the highest privileges )

**sudo userdel -r username** – to remove the user from the terminal

su – ( substitute user or switch user)

sudo – (superuser do)

**Class starts here(11/06/2023)**

* **How to create a hidden file in a Linux?**

**->** You can create a hidden file by these 3 methods :

1. touch “.hiddenfilename.txt”

2. vi “.hiddenfilename.txt”

3. echo> “.hiddenfilename.txt”

* **Why files are hidden in a terminal ?**

**-> Reasons –** 1. Hidden files contain system, configuration and sensitive data that are not meant to be viewed or edited directly by users.

2. Users have their own preferences for what files they want to see in the directory, by hiding them users can customize their view of the file system

3. If hidden files are visible on system it distract the users to select the certain files from hidden files, its called clutter reduction.

* **How to delete lines with a matching pattern/Duplicate/Repeated lines?**

-> You can do this with 3 methods –

**1. sed(stream editor) command**

**sed /’matching\_pattern\_lines/d’ file\_name > newfile\_name**

(you have to create newfile name , without it you cannot make the changes)

**2.grep(global regular expression print) command**

**grep -v ’matching\_pattern\_lines’ file\_name > newfile\_name**

**3. awk(!) command**

**awk ‘!/matching\_pattern\_lines/’ file\_name > newfile\_name**

* **How to exit from cat> file\_name.txt**

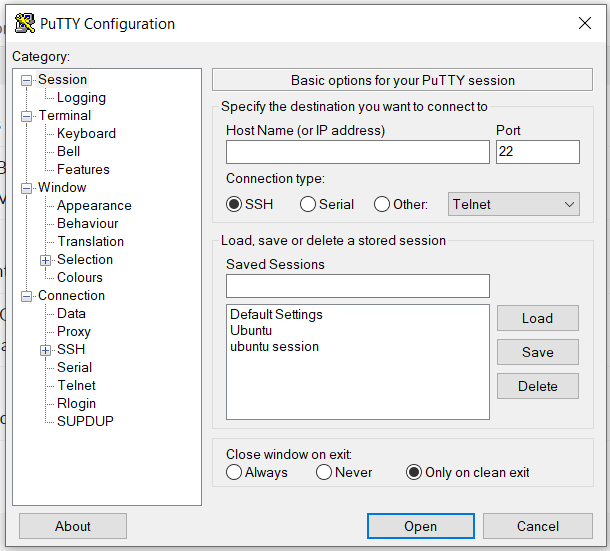
-> 1. ctrl+D – save and exit

2. ctrl+C – cancel and exit

* **Remote Server –** A remote server is physically located somewhere else , that you can access and use over a network, like a internet. example. GCP(google cloud platform)(its secured with username and password)

**PuTTY**

**PuTTY –** It’s a windows software which is used to connect linux machine from windows. ( It’s a terminal emulator and secure shell client for windows that allows you to connect to remote server and network devices for various purposes)



PuTTY is a versatile suite of network tools that goes beyond file transfer. While it's known for its PuTTY Secure Copy protocol tool (PSCP) and Secure File Transfer Protocol (SFTP) capabilities, PuTTY is primarily used for secure remote access to other computers and network devices. Here are some common uses of PuTTY other than file transfer:

1. **SSH (Secure Shell) Access:** PuTTY is widely used as an SSH client to connect to remote servers and devices securely. It provides a command-line interface for secure interactive sessions and administration.
2. **Telnet Access:** PuTTY includes a Telnet client, allowing you to connect to devices that use the Telnet protocol. However, SSH is generally recommended over Telnet for security reasons.
3. **Serial Console Access:** You can use PuTTY to access devices with serial ports, such as network switches and routers, via a serial console cable. This is useful for configuring and troubleshooting hardware.
4. **Remote System Administration:** System administrators use PuTTY to perform tasks like server management, configuration, software installation, and troubleshooting on remote servers and devices.
5. **Secure Remote Desktop and X11 Forwarding:** PuTTY can be used for secure remote desktop access (with X11 forwarding) to access graphical applications and desktop environments on Linux or Unix-based systems.
6. **Tunneling and Port Forwarding:** PuTTY can create SSH tunnels and port forwarding to securely access services on remote servers as if they were local. This is helpful for accessing web services, databases, or other resources securely.
7. **SSH Key Management**: PuTTY provides tools for generating, converting, and managing SSH keys, which are used for authentication when connecting to SSH servers.
8. **SSH Authentication Agent (Pageant):** Pageant, a component of PuTTY, acts as an SSH authentication agent. It allows you to load and use SSH keys for authentication without needing to enter your passphrase each time.
9. **Cross-Platform Remote Access:** PuTTY is used for connecting to various types of remote systems, including Windows, Linux, Unix, routers, switches, and more. It's especially valuable in heterogeneous network environments.
10. **Network Device Configuration:** Network engineers use PuTTY to configure and manage network devices such as routers, switches, and firewalls via SSH or Telnet.
11. **Terminal Emulation:** The PuTTY terminal emulator provides a user-friendly and customizable command-line interface, making it suitable for interactive shell sessions and text-based applications.
12. **Multi-Protocol Support:** PuTTY supports various protocols, including SSH, Telnet, and serial console access, making it a flexible tool for managing different types of devices.

**File Transfer Commands**

**scp,ftp,sftp**

1. **scp – Secure Copy Protocol (Port Number – 22)**

It’s a linux command used for securely copying files and directories over Linux local and remote servers. It encrypts data during transmission for security over a network. If you wants to connect to windows system from linux you will need a scp server or utility installed in your windows system.

**Command:** **scp file\_name user@ip\_address:/path/destination**

1. **ftp – File Transfer Protocol (Port Number – 21)**

ftp is a standard network protocol used to transfer the files between the client and computer network. One of the oldest and widely used protocol developed in 1970’s. Due to security concerns associated with ftp especially when used without encryption. (So basically, its like there is a website on internet(like facebook, youtube) while you have the access to use it then you can simply enter the ftp command with website address and it will ask for your username and password to sign, after that you can access and modify the site contents)

**Command :**

**$ ftp ftp.example.com**

Connected to ftp.example.com.

220 (vsFTPd 3.0.3)

Name (ftp.example.com:your-username): **your-username**

331 Please specify the password.

Password: **your-password**

230 Login successful.

Remote system type is UNIX.

Using binary mode to transfer files.

**ftp> ls**

**ftp> get remote-file.txt**

**ftp> put local-file.txt**

**ftp> bye**

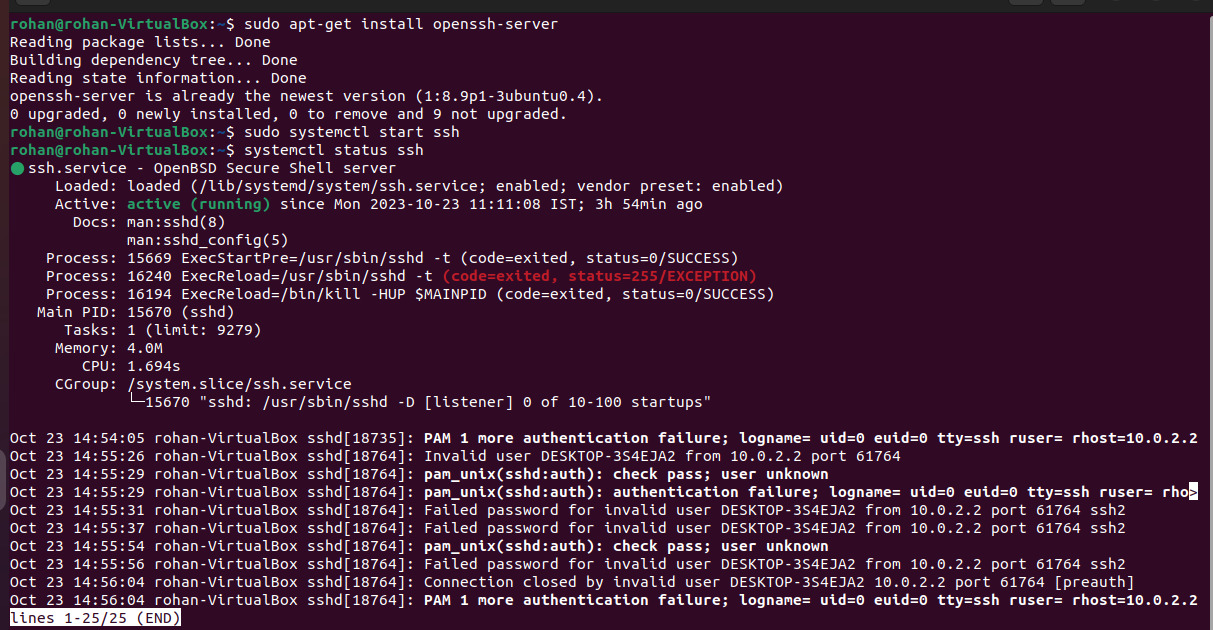
**sftp - Secure File Transfer Protocol/ Secure Shell(ssh) file transfer protocol/ Secure file transfer program (Port No- 22)**

It’s a secure network protocol used for securely transferring files between a client and remote server or both remote server . It is a subsystem of secure shell(ssh) protocol , Its encrypts data during transmission.

It will work on similar or cross platform, you need a access from remote server with username and password.

To access sftp server you have to be installed ssh in both platforms,see below

1. this is for linux( Ubuntu)



1. For Windows
2. Open settings
3. Open apps, Inside apps there is option of ‘optional features’ click on it
4. Select openssh server from there and install it.

**Command :**

**$ sftp your-username@host\_ipaddress**

**Password:**

**sftp> put file\_name.txt** (send the file to server)

**sftp> get file\_name.txt** (receive the file from server)

**sftp> mget \*.txt** (get multiple files of .txt from server current directory)

**sftp> mput \*.txt** (send multiple iles of .txt from local server current directory to remote server directory)

**sftp>!**

**sftp>ls** (list all the files of remote server)

**sftp>!ls** (list all the files of local server)

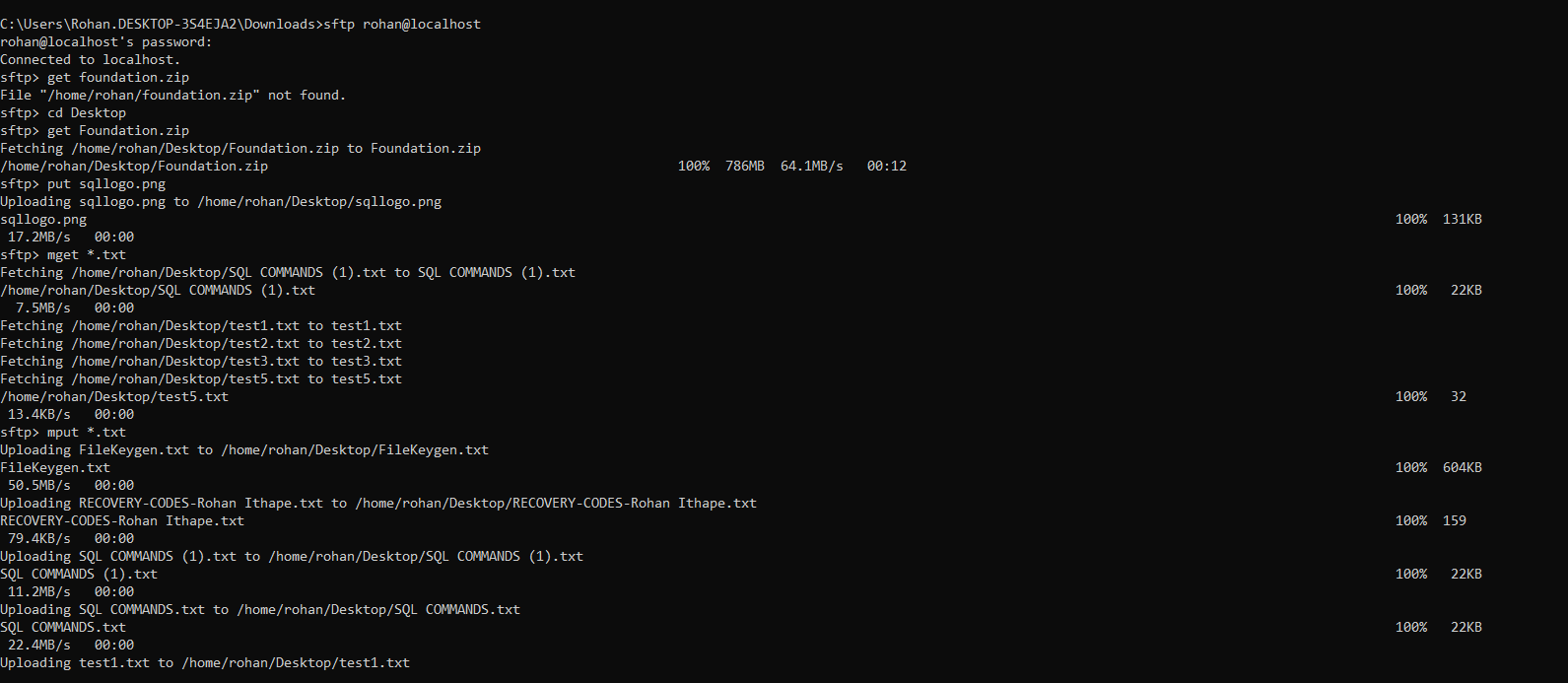
**sftp>pwd** (present directory of remote server)

**sftp> quit** (exit from the server)

**Process :** ( Listed commands are the connection of windows to Linux)

1. Get the files from linux server to windows server in a same system.

2Open ‘Command Prompt’, Run ‘sftp hostname@localhost/ipaddress.

1. Enter remote server host password, and you are connected.
2. (sftp>get filename.txt)
3. Check the below image for the reference.
4. (If there are multiple files of .txt in a remote server and you want only selected files from it, then you need to 1st either move or transfer those selected files to another directory and then you can get/send it with get/mget or put/mput command)
5. Use the below image for the reference
6. 

**Openssh server vs Openssh Client**

OpenSSH server and OpenSSH client work together to establish secure connections and provide encrypted communication between a local machine (client) and a remote server (server). The server handles incoming connections, while the client initiates connections and interacts with the remote server. Both components play essential roles in enabling secure remote administration and data transfer over SSH.

**RDP – Remote Desktop Protocol/Remote Desktop conection**

It is a technology developed by Microsoft for Windows operating system, it allows user to connect to and control a remote computer or server from another computer or device over a network.

**Advantages –**

1. Resource sharing – users can access files, application and resources on a remote computer from anywhere over an internet

2. centralized management – RDP is a centralized management and administration of remote computers, making it easier for software update, security patches and troubleshoot

3. security – RDP data is typically encrypted, enhancing data security .

4. Collaboration – Teams at different location can collaborate on project by sharing a remote desktop

**Disadvantages –**

1. Network Dependency – RDP depends on fast and stable internet connection, slow or unreliable network leads to poor user experience.

2. Compatibility – Not all applications or software works seamlessly on RDP , especially graphics intensive or hardware related software.

3. Offline access – Its totally works on an internet so users who needs to work without network connection may face limitation

4. Security – Weak passwords can pose security risks.

5. Licensing – Some versions of rdp requires licensing fees

Eg. Anydesk, RealVNC, Teamviewer, Chrome Remote Desktop, Splashshot, Nomachine, Radmin

**Protocol**

It is a “set of rules” which are used in a digital communication to connect network devices and exchange information between them.

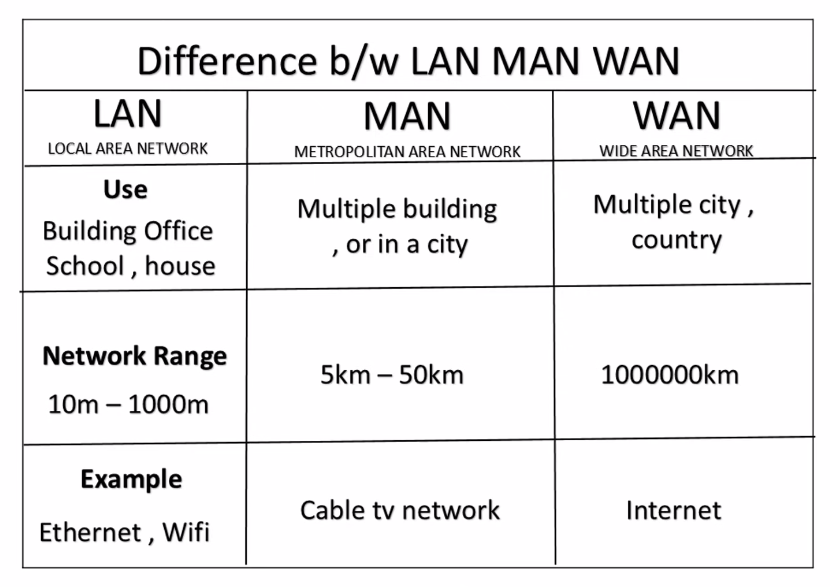
**PORT NUMBER -**  It is a 16 -bit unsigned integer used to identify specific processes or services that are running on a computer within a IP network.

**16-bit :** It refers to number of binary digits. 1 bit can be 1 or 0 which makes the 2^16 =65536 possibilities.

**Unsigned :** It means the integer can only represent positive numbers.

**Explanation in simple way –** 1 boy has 65536 friends , he has all 65536 unique secret codes to speak to the friends. Whenever he wants to speak with his specific friend(protocol) he has to 1st tell the secret/unique code(port number ) to his friend, which enables him to share the information with his friend.

**Types of Networks**



**FileZilla vs PuTTY**

|  |  |
| --- | --- |
| **FileZilla** | **PuTTY** |
| 1. Function : FileZilla is primarily an FTP (File Transfer Protocol) client, although it also supports the more secure SFTP (SSH File Transfer Protocol). 2. Main Use Case: FileZilla is used for transferring files to and from remote servers. It's often used by web developers to upload website files, manage directories, and perform file-related tasks. 3. Features: It provides a user-friendly graphical interface for file transfers 4. Encryption: Supports encryption through SFTP for secure file transfers 5. FileZilla is primarily a file transfer tool with a graphical user interface | 1. Function : PuTTY is a collection of terminal software tools, with PuTTY itself being an SSH and Telnet client. 2. Main Use: PuTTY is used to establish secure shell (SSH) connections to remote servers for command-line access and terminal-based tasks. 3. Features: PuTTY provides terminal emulation and SSH access to perform a wide range of tasks on a remote system, including running commands 4. Encryption: Offers secure SSH connections for remote access. 5. PuTTY is a terminal client for secure shell access, typically used for command-line administration and remote server control |

**LOGIN COMMANDS**

"login command" is not commonly used to refer to SSH or PuTTY. Instead, they are tools used for remote access and secure login to remote systems.

**ssh, PuTTY**

ssh is a protocol that enables secure remote access and communication, while putty is specific software for windows that implements the ssh protocol. PuTTY acts as a bridge that allows windows users to connect to remote systems that supports ssh.

ssh – Secure Shell

Secure shell is a secure and encrypte Before Telnet was much popular that ssh, but it has security issues

d protocol for securely connecting to and managing remote computers and servers over a network. It’s a secure alternative for non-protected login protocols (telnet) and file transfer protocol(ftp)

Check the file installed or not in your system –

**Check the file /etc/ssh/sshd\_config**

Installation command

**Install openssh-clients openssh-server**

Check status of ssh active/inactive

**systemctl status sshd**

Commands :

1. ssh username@ipaddress or open putty enter ip address and connect, now enter login\_id and password ( you should already have the account in remote server , to use same login id and password)

2. Now you will be inside the remote server and then same commands like scp

**PuTTY –**

Primarily Designed for Microsoft Windows only and is not intended for use on linux. It is possible to use in linux using compatibility layers or emulators(enables one computer system to mimic the functions of another computer system in order to run software or execute tasks ) like wine.

PuTTY was written and is maintained primarily by **Simon Tatham**, a British programmer.

It’s a software application , from where you can keep the records of remote servers , and its more easy to handle than any other software or commands.

Its connect to remote server by Ip address and port number. After that using ssh command in a terminal you can access the files and directories from remote servers.

Commands –

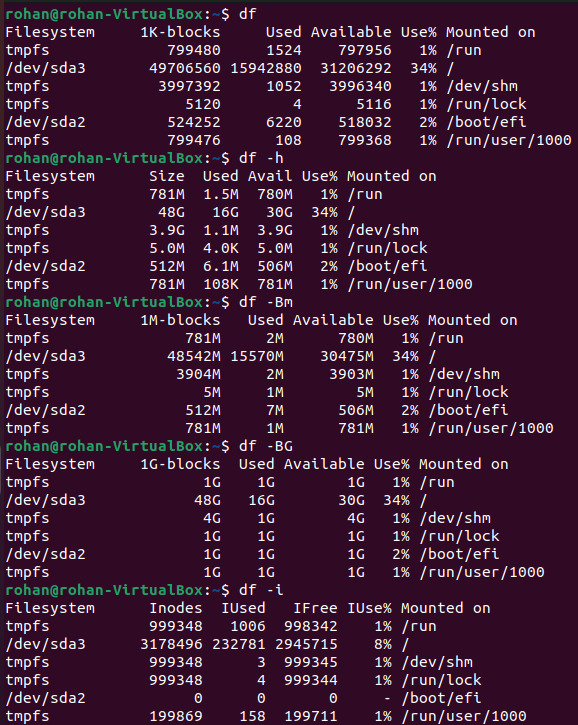
1. open putty enter ip address and connect,

2. now enter login\_id and password

( you should already have the account in remote server , to use same login id and password)

**Disk Usage Comands/ System Monitoring Commands**

It is used to get full summary of available & used disk space



**Referrence : ‘df’ command**

usage of linux system

ex. If you want to deploy the application or software on

system , to check the space on server you

can use this commands)

1. **df – Disk Free**

It displays information about disk space

usage on mounted filesystems, including

their total size, used space, free space

and the percentage of space used . It’s only for the

mounts and not for the files.

By default it provides information in 1k blocks.

df : Default kb format

df -h: In Human readable format

df -BM: In MB format

df -BG: In GB format

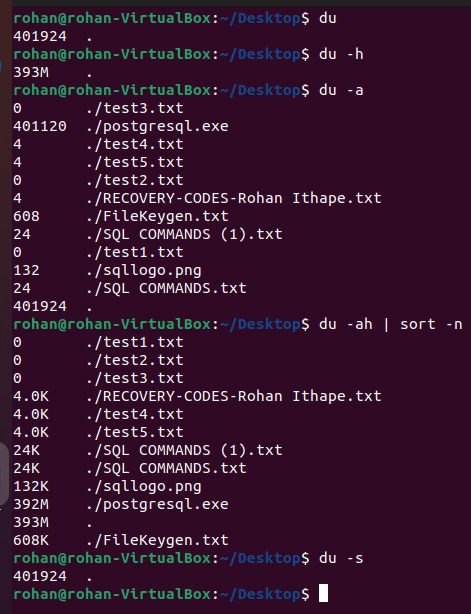
df -i: inode format

(Inodes are important because they are used to track and

manage files and directories on the filesystem. If you run out of

available inodes, you won't be able to create more files or directories

on that filesystem, even if there is still disk space available. This can happen in situations where many small files are created.)



Reference ‘du’ command

1. **du – Disk Utilization/ Disk Usage**

It is used to display the disk space usage of directories and files

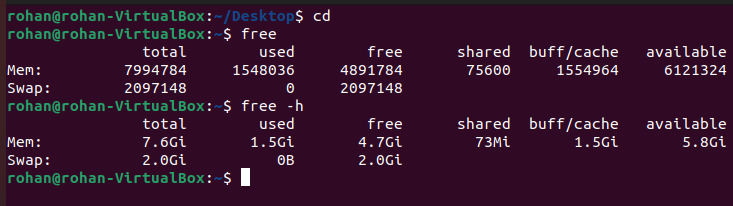
du : all system files & directories in defauly kb format.

du -h:( human readable format)

du -ah | sort -n: All data in ascending order with

readable format

du -s: current directory total default size in kb



Referrence ‘free’ command

1. **free:** Display amount of free and used memory in the system

**free -h**

**free -h -s 3 (This command will give you free and used space on terminal for every 3 seconds, to stop enter ‘kill -9 PID’**

**Difference between System memory(RAM) vs Disk Space**

RAM (system memory) is used for temporary, fast-access storage of data required for immediate processing by the CPU. It significantly affects system performance, and having enough RAM is essential for smooth multitasking. Disk space, on the other hand, is used for long-term storage of data, applications, and the operating system. It doesn't directly impact CPU performance but is crucial for overall system usability and file storage. Both RAM and disk space are important components of a computer, each serving its specific role.

**Searching Commands**

Searching commands used in a linux to find directories, files or contents within your files.

**grep: Global Regular Expression Print**

It is used to search a pattern into the files & shows the output.

1. grep ‘pattern’ filename.txt

This will search the pattern in a current file only

**grep “text\_pattern” File\_name.txt**

1. This will ignore the Uppercae and Lowercase and will show the exactly same matching pattern (I – ignores case sensitiveness)

**grep -i “text\_pattern” File\_name.txt**

1. This will count the total number of matches in it ( c – count the matches)

**grep -c “text\_pattern” File\_name.txt**

1. This will hide the matching pattern and will show the rest of the content (v – verbose)

**grep -v “text\_pattern” File\_name.txt**

1. If looking for that pattern in multiple files it will display the file name in it

**grep -l “text\_pattern” File\_name.txt File\_name1.txt**

1. This will display the lines which starts with the letter “mentioned”

**grep ‘^mentioned’ File\_name.txt**

1. This will display the lines ends with letter “mentioned”

**grep ‘mentioned$’ File\_name.txt**

**find Command:**

This command is used for real time searches or complex criteria.

**Command –**

**find - name file\_name**

**locate command:**

This command is used to quickly locate the file or directory in a system.

Command –

**locate file\_name**

**Process Commands:**

A set of commands and utilities that allow you to interact with and manage the execution of programs and tasks running on your computer.

1. ps – shows the process of current shell/ It shows the currently running processes on a current system.

PID – Unique process ID

TTY - Terminal type of user logged into

TIME – Ammount of cpu in min and sec that process has been running

CMD – name of the command that launched the process.

ps – e/ps -A /ps -aux – Display information about all the processes running in a system.

Ps -u username - It shows all the processes running under this user.

1. **top – (Table of Processess)**

Its same like ps but it provides the real time view of system process and kernel managed tasks

The top command is an essential tool for system administrators and users who need to keep a close eye on their system's performance, identify resource-hungry processes, diagnose issues, and take action to optimize system utilization and stability. It's a versatile and valuable utility for system monitoring and management.

PID: Process ID.

USER: The user running the process.

PR: Priority of the process.

NI: Nice value (priority adjustment).

VIRT: Virtual memory used by the process.

RES: Resident memory (physical RAM) used by the process.

SHR: Shared memory used by the process.

S: Process state (e.g., R for running, S for sleeping).

%CPU: CPU usage percentage.

%MEM: Memory usage percentage.

TIME+: Total time the process has been running.

**bg – Backgroung**

This command in Linux is used to send a suspended process to the background, allowing it to continue running without blocking the terminal.

It is used to manage and control processes running in the background, making it a valuable tool for multitasking and optimizing your workflow in a Linux operating system.

Command –

Asssume , sleep 100 is your current job in your system – ctrl+Z - stopped

bg %sleep – Now this job went background

**fg - foreground**

The fg command in Linux is used to bring a background process to the foreground, allowing you to interact with it in your terminal session. This command is especially useful when you have one or more background processes running, and you want to switch your attention back to a specific process.

Asssume , sleep 100 is your current job in your system – ctrl+Z - stopped

bg %sleep – Now this job went background **(instead of sleep you can use the job id number also)**

fg %sleep – It will start show on the terminal

**nohup – no hang up**

If you want your process keep running even after you close your terminal you can use ‘nohup’ command.

**nohup ./echo.sh &**

(This is an scripting command , now even if you close the terminal it will keep running in the background.)

**Sar – System Activity Report**

**Sadc – system activity data collector**

This command in linux is used to collect, report and save system activity information. It provides a wide range of system information , such as cpu usage, memory usage, disk acitivity, network activity and more.

vmstat – virtual memory statistics

This command provides detailed information about system performance with a focus on virtual memory statistics.

vmstat 2 5

( This means that it will show the statistics for every 2 seconds for 5 times.)

**Kill**

The kill command in Linux is used to terminate or send signals to processes.

**Command – Kill PID\_NO**

Here are some common signals you can use with the kill command:

TERM (15): Terminate (default). This asks the process to exit gracefully. Most processes will respond to this signal by stopping.

KILL (9): Kill. This forcefully terminates the process. It doesn't allow the process to clean up or save data.

HUP (1): Hang Up. Some processes reload their configuration when they receive this signal.

INT (2): Interrupt. Similar to pressing Ctrl+C in the terminal. This is often used to stop a running command.

STOP (19): Stop. Pauses a process.

CONT (18): Continue. Resumes a stopped process.

**System Commands:**

System administration commands or administrative tools. It is used to manage and configure various aspects of the system.

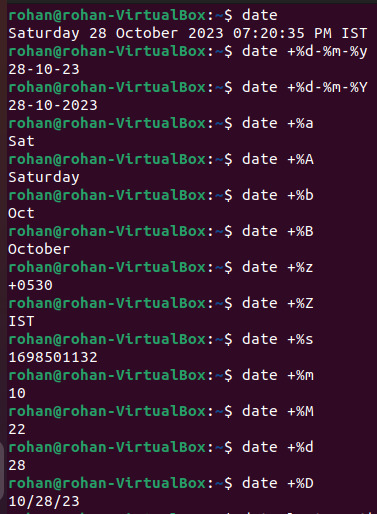
**cal –** calendar

It is used to display a calendar for a specified month and year.

1. cal – Display a current month calender
2. cal 8 2020 – Specific month
3. cal 2020 – full year
4. cal -3 : previous, current & next month
5. cal -j – julian calendar

**time –** It shows the command execution time/ Time takes to finish the command & task.

**Command –** 1. time



1. **time file\_name/task**

**date**

It is used to display the day date and

time along with time zone.

**date**

**date +%A** – day of the week

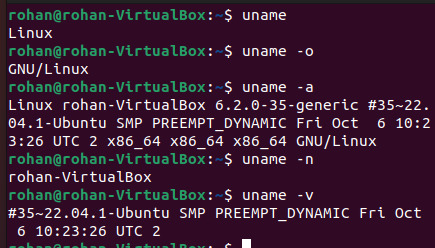
**date +%B** – month of the year

**date +%Z** – display the current timezone

**date+%s** – total no of seconds from 1 Jan 1970.

**uname**

It is used to display system information, including the operating system name, kernel version, host name, machine hardware, and more. It's a versatile command that provides basic system information, which can be useful for various administrative and scripting tasks.



**uname** : Display os name

**uname -o** : kernel name

**uname -a** : all information

**uname -n** : nodename/ hostname

**uname -v** : kernel version

**uptime**

The uptime command in Linux is used to display the system's uptime and various load averages. It provides information about how long the system has been running since its last reboot, as well as the system's current load average over different time intervals.



**19:47:59** Indicating that the system has been up for 19 hours, 47 minutes, and 59 seconds.

**up 1:02** indicates that the system has been running for 1 hour and 2 minutes.

**1 user** signifies that there is one user currently logged in to the system.

**load average: 0.12, 0.18, 0.18** provides the system's load averages. the load averages are 0.12 for the last minute, 0.18 for the last 5 minutes, and 0.18 for the last 15 minutes. These numbers indicate the system's current load and can be useful for understanding its performance. A load average of 1.0 typically means that the system's resources are fully utilized, so values lower than 1.0 are generally considered good.

**who -** The who command in Linux and Unix-like operating systems is used to display information about users who are currently logged into the system. It provides details about their usernames, terminal sessions, login times, and more.

**Whoami –** It is used to display the username of the current user who is logged into the terminal session.

**Which** – It display the location of the particular file in a system

**id** – It is used to display information about the current user or a specified user. It provides information about the user's UID (User ID), GID (Group ID), groups the user belongs to, and some additional information about the user.

**users –** It is used to display the list of users who are currently logged into the system.

**sudo – Superuser do**

It is used to allow authorized users to run commands with the privileges of another user, usually the superuser or "root."

It is a powerful tool for granting temporary administrative access to users while maintaining system security.

**sudo -l** : This command will show wheather the user has a sudo access or not.

**Shutdown –**  It is used to shutdown or restart a computer or server.

-h or -P: Shutdown and power off the system after the shutdown is complete.

-r: Restart the system after the shutdown .

-c: Cancel a previously scheduled shutdown.

-t: Specifies the delay time in minutes before the shutdown or restart occurs.

**reboot -** The reboot command in Linux and Unix-like operating systems is used to reboot or restart the computer.

**passwd -**

The passwd command in Linux and Unix-like operating systems is used to change a user's password. It allows both regular users and system administrators to update their passwords, providing a level of security for user accounts.

**chpasswd -** It is used to change user passwords in batch mode. It allows you to change passwords for multiple users at once by reading the new password information from a file or standard input.

**Compression commands**

1. zip : It is used for compress the files on linux system. This command provides compression as well as packaging functionality as well as bunch functionality. Extension of zip files is ‘zip’. Default behaviour of zip command is keeping the original files as it is.

It allow you to reduce the size of files and directories for storage or transfer.

gzip – Compress the file

gunzip – Decompress the file

tar – compress the folder

tar -zcvd file\_name.tgz folder\_name

**Networking Commands**

**ping -** The ping command is a network utility tool used to test network connectivity and check the reachability of a remote host or server.

(So basically this command send a signal to the ip address of the host over a neetwork, if it reflects back it means the device is under a network.)

**telnet – Teletype Network**

Terminal Emulation program

It is a network protocol used for remote terminal emulation, which allows a user to

The telnet command is used for establishing text-based, interactive network connections to remote hosts using the Telnet protocol ( It’s same like ssh/putty, Need a ipaddress and port number for connection)

**Ipconfig :** It is used to display the network configuration information for all the network interfaces on your computer.

netstat – It is used to display network statistics, routing tables and active network connections.It is often used for network troubleshooting and to minor the network activity.

**nslookup** : This command will provide you with the IP address associated with the hostname. It also displays information about the DNS server that was used to perform the lookup.

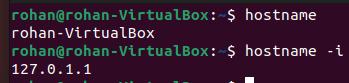
**traceroute :** Tracks the route packets on how the data travels on internet from your computer to destination.

The only required parameter is the name or ip address of the destination host.

**hostname**

The hostname commansedd is used to display the system or server name of the local machine.

hostname



hostname -i

hostname -a

**Special Commands**

**Sed -**  Stream Editor

This command is used for manipulating and transforming text from the files in linux box. It’s also used for the purposes like find & replacement, print the specific lines, print or delete the lines into the range.

1. Find & Replacement

1). sed ‘s/a/b/’ a.txt (s – Substitute)

This command will replace and print ‘a’ to ‘b’ in a.txt file, It will only replace 1st matching word from each line. If there are multiple same words in 1 line it will replace only 1st word. This will not make any permanent changes in original file.

2). sed -i ‘s/a/b/’ a.txt

This will replace ‘a’ to ‘b’ in a.txt file permanently.

3). sed ‘s/a/b/g’ a.txt (g – global replacement)

This will replace the ‘a’ to ‘b’ from a.txt in a whole file. While Normal command will replace only 1st word of the file.

4). sed ‘s/a/b/i' a.txt ( i – Ignore Uppercase/Lowercase)

This will replace not only ‘a’ to ‘b’ but will also replace ‘A’ to ‘b’ if there is ‘A’ in the text file. only the 1st word of the line will be replaced. If want to replace it from the whole file need to use ‘ig’

5). sed ‘s/a/b/2’ a.txt (2nd occurances)

This will replace all the 2nd occurances of ‘a’ to ‘b’ from a.txt file from each line.

6). sed ‘s/a/b/2g’ a.txt

This command will will replace ‘a’ to ‘b’ from 2nd occurrence’s from from each line. If there are three occurrence’s in 1 line it will replace only 2nd occurrence and will keep rest 2 as it is.

1. Print

1) sed ‘s/a/b/p’ a.txt

This command will replace ‘a’ to ‘b’ in a.txt, while it will print the replaced lines 2 times along with data as it is.

2) sed -n ‘s/a/b/p’ a.txt

This command will replace ‘a’ to ‘b’ in a.txt as well as It will print the replaced line.

3) sed -e ‘s/a/b/ a.txt -e ‘s/c/d/ a.txt

This command is used to replace the multiples words/patterns from a same or different file.

**Crontab –** It is used to create, view, edit and manage scheduled tasks or jobs that are executed at a specific time or intervals. These tasks also referred to as a ‘cron jobs’.

It is exactly similar like ‘task schedular’ in windows operating system.

1. Crontab -l

crontab -e

\* \* \* \* \* /bin/sh /home/user/Desktop/file\_name.sh (ctrl x)

cd Desktop

vim file\_name.sh

touch /home/user/Desktop/file.txt (:wq)

chmod u+x file\_name.txt

mailx command

bc – Basic Calculator

It is a command-line calculator that provides a simple and interactive way to perform arithmetic calculations.

expr – expression

same like 1bc ,expr = 6 + 4

10

tr command – Translate or Delete

It is used to perform character by character replacement, deletions or translations on input text.

Command :

Cat file\_name | tr [:lower:] [:upper:] (lower to uppercase)

Cat file\_name | tr -d [:punct:] (delete punctuals)

Cat file\_name | tr -d [:digit:] (delete the digits)

Sed – It is primarily used for processing and transforming text using regular expressions.

Command :

sed -n ‘ 2p’ file\_name.txt ( print the 2nd line from the text)

sed -n ‘$p’ file\_name.txt (print the last line from the text)

sed ‘2i textline\_name’ file\_name.txt ( insert extra line in a text )

sed ‘s/text1/text2/p’ file\_name.txt (substitute text1 to text2)

awk – It is used for processing and analyzing text files, particularly structured data like tables or log files. (same like cut command)

command –

awk ‘{print $2}’ file\_name.txt (print 2nd column)

awk ‘{print $NF}’ file\_name.txt (print last column column)

ls -ltr | awk ‘{print $NF}’ ( print last column of current directory)

echo - It is primarily used to display text or variables on the terminal or in shell scripts.

Command :

echo -e ‘hello \frnd’

* hello

frnd

echo -e ‘hello\tfrnd’

* hello frnd

v=’hey’

u=’frnd’

* echo $v $u
* hey frnd

**Network Types –**