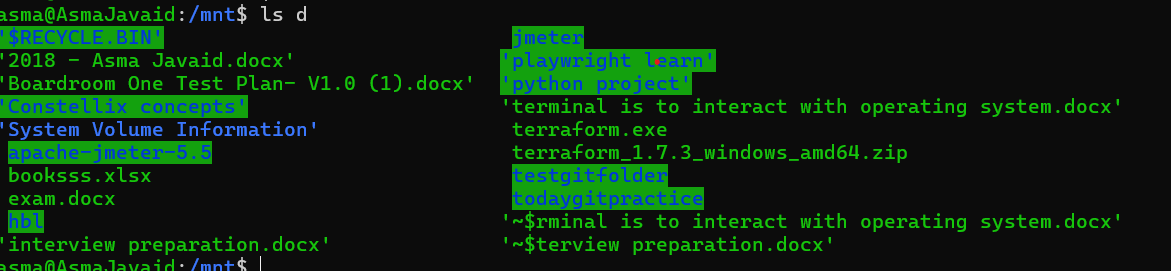
1. terminal is to interact with operating system
2. cd / takes you to the root directory of the entire file system. Example: If you run cd /, you will be in the root directory where you can access system folders like bin, etc, usr, and others.
3. cd ~ takes you to your personal home directory.Example: If your username is john, running cd ~ will take you to /home/john (on most Linux systems). For the root user, cd ~ would take you to /root.
4. cd .. (Move Up One Directory Level): If you are in /home/user/docs, running cd .. will take you to /home/user.
5. cd - toggles between the current and previous directories. if you are in /home/user/docs and move to /var/log, running cd - would take you back to /home/user/docs.
6. clear: Clears the terminal screen, giving you a clean workspace, but does not erase the command history or outputs—you can still scroll back to view them.
7. ls: Lists files and directories in a directory. Ls foldername will list the specific folder such as below  
   
8. Ls -a return all the hidden files and ls -l return the more details about files like something this drwxrwxrwx 1 asma asma 4096 Aug 20 11:05 c
9. mkdir creates a folder
10. touch help you to create files like text, csv,
11. echo null > filename.txt: Writes the word "null" into filename.txt, replacing its contents if it exists.
12. sudo elevates your privileges to execute commands that require administrative rights.
13. sudo rm -rf foldername forcefully deletes a directory and all its contents with superuser privileges, bypassing any permission restrictions.
14. Command without sudo: rm -rf /protected\_folder. If /protected\_folder is owned by the root user or has restricted permissions, you might get a "permission denied" error.

**BASH SCRIPTING**

1. In order to comment multiple lines in bash you can use this command

: ‘

First comment

Second comment

‘

------------------------------------------------------------------------------------

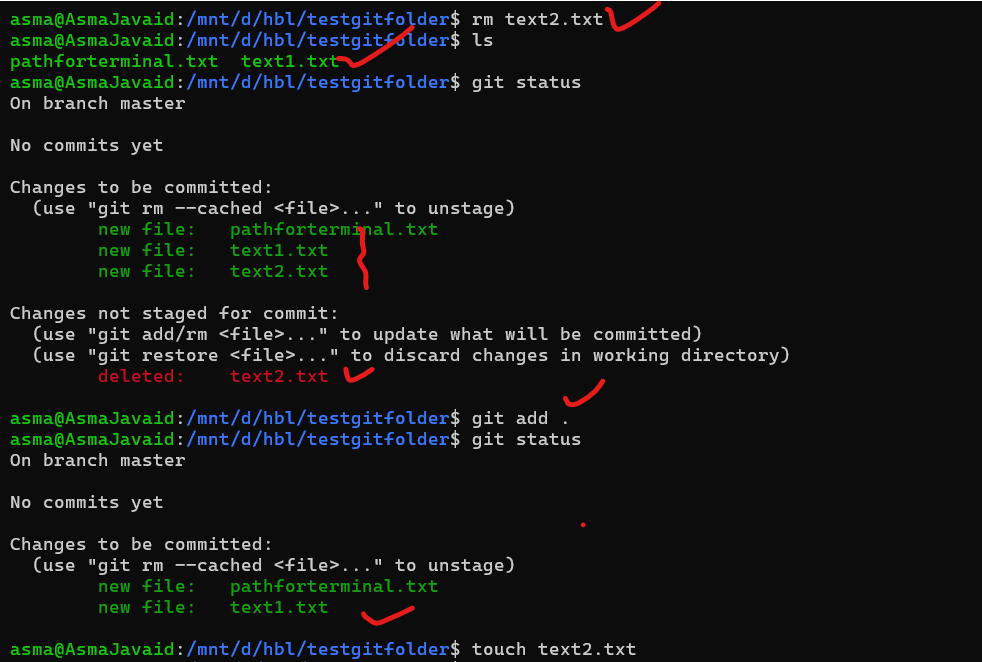
1. The git init command is used to create a new Git repository. When you run git init in a directory, it initializes a new Git repository in that directory, allowing you to start tracking your project's files with Git.
2. The git status command in Git provides an overview of the current state of your working directory and staging area. It shows you which changes have been staged, which haven't, and which files are not being tracked by Git.

* Untracked Files: Files in your working directory that are not being tracked by Git. These files have not been staged or committed.
* Changes Not Staged for Commit: Files that have been modified but are not yet staged for the next commit. These are changes you’ve made that Git is aware of, but they won’t be included in the next commit unless you stage them.
* Changes to Be Committed: Files that have been added to the staging area and are ready to be committed. These changes will be included in the next commit.
* Current Branch Information: Displays the name of the current branch and shows if your branch is ahead, behind, or diverged from the remote branch, along with any unmerged files if applicable.

1. The git add command is used to add changes in your working directory to the staging area

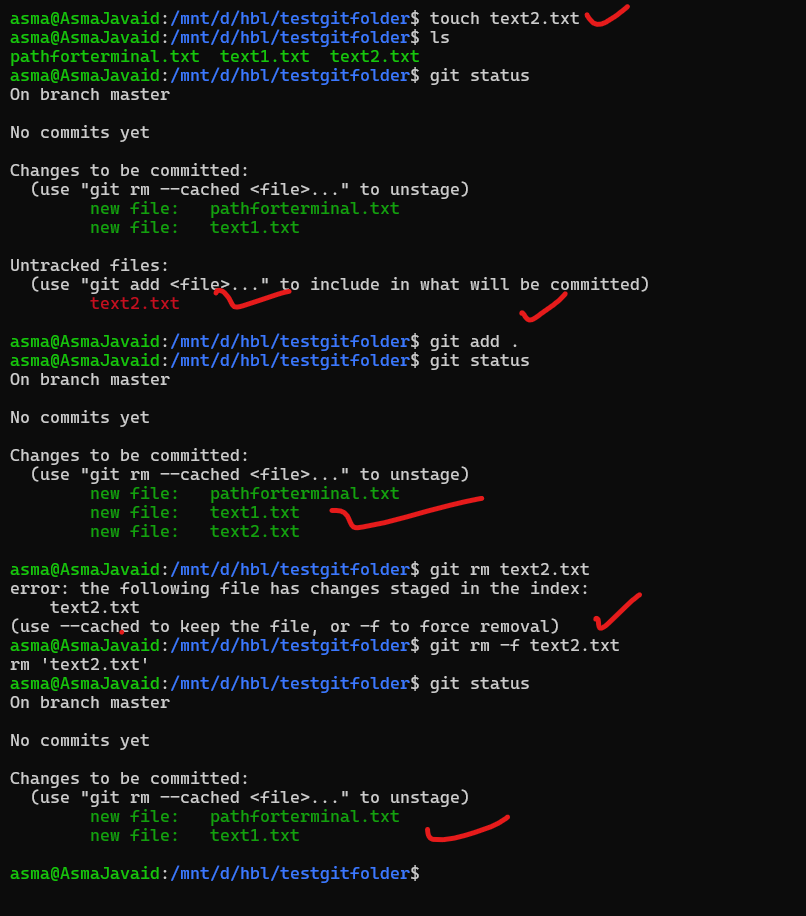
* Staging Changes: When you modify files in your working directory, those changes are not automatically included in the next commit. You use git add to specify which changes you want to include in the next commit.
* Files and Directories: You can use git add to stage individual files, multiple files, or entire directories.

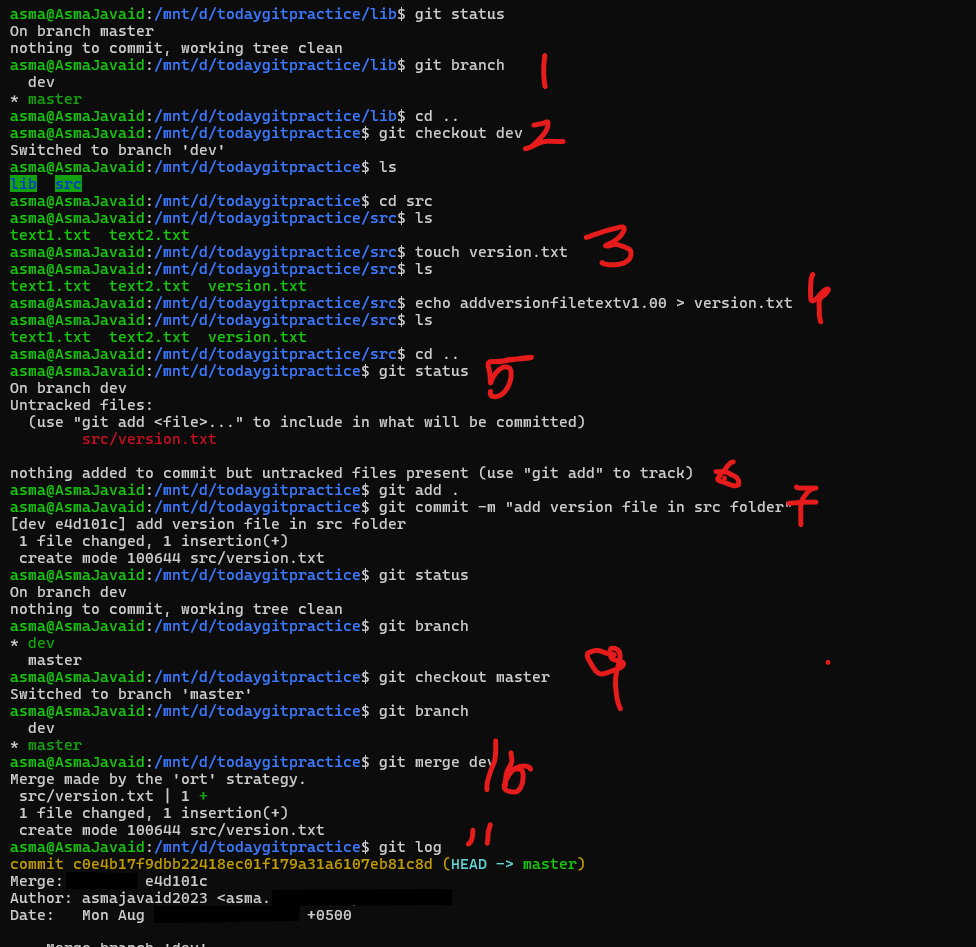
**Note1**: If add file and then delete it using rm text1.txt command then it will delete the file but it will not be at stagging until we do git add . like in below image. Which means **rm text1.txt**: Removes the file from your working directory but does not stage the deletion. You need to manually stage the change with git add.

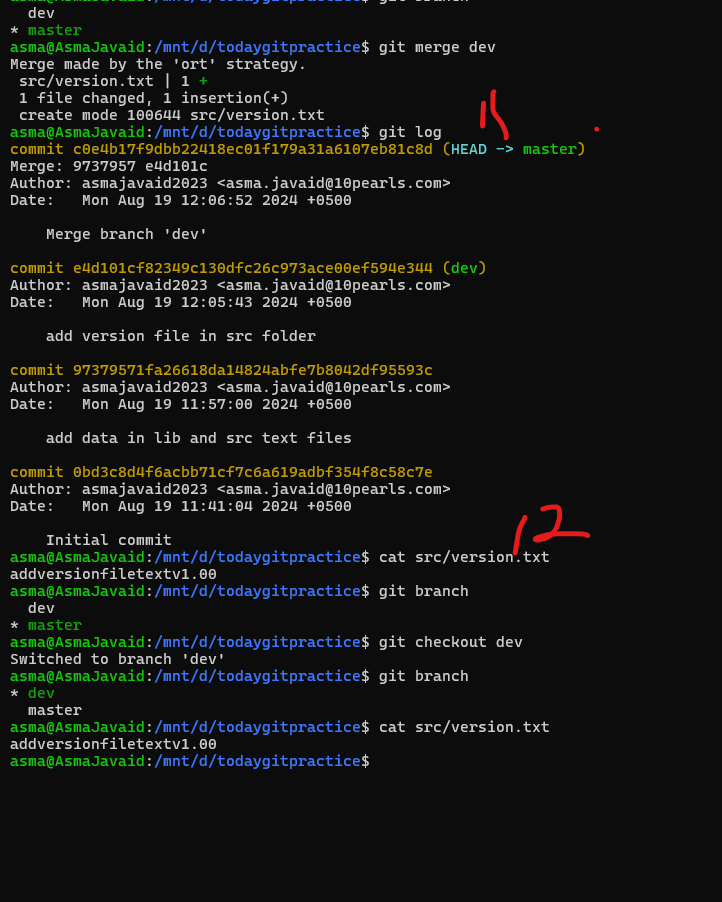


**Note2**: Now if I add a new file text2.txt and want to delete it using a command git rm -f text2.txt then it will automatically be at stagging and we dont need to do git add . in order to remove from git repository like in below image, which means **git rm -f text2.txt**: Removes the file from both your working directory and stages the deletion in one step, making it ready for the next commit.

1. Now if i create a folder **todaygitpractice** and create a new Git repository and in that folder i also create two more folders name lib and src and each of these folders i create 2 text files. Now when i do git add . inside the main folder name **todaygitpractice** then all files will be adding to staging but master branch didn’t exist until we do git **commit -m “messagefor commit”** then these files will be commited and we will be able to create a git branch. 
2. Git branch (branch name) is used to **create or add branch e.g git branch bugs.** If you want to checkout to this branch then you do **git checkout bugs** and if you want to go back to master branch then you do **git checkout master.** Also if you want to **delete** a branch you can do **git branch -d bugs. Git branch** command will display all the branches . **Git log** command displays the commit history of a repository, showing details like commit hash, author, date, and commit message



Note: If i create dev branch by using git branch dev command and then checkout in dev branch using git checkout dev and made changes in the folder in dev branch like adding data or folder and then add and commit changes then when checkout to master branch and merge dev changes into master using git merge dev command then dev changes will merge into master and you can verify it using this command cat src/version.txt. Follow bellow images to see steps  




**Protocals:** describe how 2 computers will interact with each other over a network and **network protocals** ensures that devices understand each other like if someone has macbook and someone has windows then both have functions and need a common language or protocal to understand each other function also **network protocal**  are like rules that how data should be sent received correctly and understood by the network

**Router**: connects all devices to the internet and to each other  
**Directing Traffic:**

* When you want to visit a website (e.g., www.example.com), your computer sends a request. This request goes to the router first.
* The router forwards the request out to the internet through the ISP.
* The website’s data (like the webpage content) comes back to the router, which then sends it to your specific computer based on your internal IP address

**Real-Life Office Example:**

* **Internet Access:** Every time someone in your office checks their email or visits a website, the data first goes through the router. The router decides whether to send that data out to the internet or to keep it within the office network.
* **Connecting Remote Offices:** If your office has a second location, the router helps connect the two offices securely, allowing employees to share files and access resources across both locations as if they were on the same local network.
* The router can also include basic firewall functions, filtering incoming and outgoing traffic to protect the office network.
* Example: The router might block certain types of traffic (like unauthorized access attempts) to keep the network secure.

A switch is a network device that connects multiple devices (like computers, printers, servers) within the same local network (LAN). It enables these devices to communicate with each other efficiently.

**Scenario:** You're in an office where multiple employees need to access shared resources like files, printers, and internal applications hosted on servers.

**How the Switch Works:**

1. **Managing Data Traffic:**
   * When your computer sends data (like a file to print or a request to access a shared document), the switch receives this data.
   * The switch examines the data to determine the destination device by checking the MAC address. The MAC address is a unique identifier assigned to each network interface card (NIC) on every device.
2. **Directing Data Efficiently:**
   * Instead of broadcasting the data to every device on the network, the switch intelligently sends the data only to the intended recipient.
   * Example: If you send a file to the printer, the switch directs the data specifically to the printer's MAC address, rather than sending it to every device on the network.

**What is a Firewall?**

* A firewall is a security device or software that monitors and controls incoming and outgoing network traffic based on predefined security rules.
* The firewall sits between your office network and the internet, monitoring all data entering and leaving the network.
* Example: When you try to access a website or an external service, the firewall checks the request against its set of security rules before allowing the data to flow through.
* Example: If an employee unknowingly clicks on a link that tries to download malware, the firewall can detect the threat and block the download, preventing the malware from infecting the office network.
* Example: If someone tries to send a file containing sensitive company data to an external email address, the firewall can block this action if it violates security policies.

**ARP** maps IP addresses to MAC addresses, which is essential for communication on a network. This mapping is necessary because devices use IP addresses to identify each other, but switches use MAC addresses to forward data.  
**Communicating on the Same Network**

* **How Devices Talk:** When one device on the network wants to send data to another device, it needs to know the recipient's IP address. If both devices are on the same network (like your laptop and your friend's laptop connected to the office Wi-Fi), they can communicate directly.
* **Role of the MAC Address:** Before sending data, the device needs to know the MAC address of the recipient. This is where ARP (Address Resolution Protocol) comes in—it helps match the IP address to the correct MAC address, ensuring the data is sent to the right device.

**How Switch and Arp Work Together:**

* **Initial Communication:**
* When one device on a network wants to send data to another device, it knows the other device's IP address but might not know the MAC address.
* The device uses ARP to ask, "Who has this IP address?" The device with that IP address replies with its MAC address.
* **Switch Uses MAC Addresses:**
* Once ARP provides the MAC address, the sending device can frame the data with the correct destination MAC address.
* The switch then uses its MAC address table (a list of MAC addresses and the corresponding ports on the switch) to forward the data only to the port where the target device is connected, rather than flooding the data to all devices.

**FTP** stands for File Transfer Protocol. It's a method used to transfer files between computers over the internet or a network.

**Simple Explanation:**

Imagine you're sending a package to a friend. FTP is like the postal service that helps you send the package from your computer to your friend's computer. It has a set of rules that both computers follow to make sure the files are sent correctly.

**How FTP Works:**

* **Sending Files:** If you want to share a file (like a photo or a document) with someone, you can use FTP to upload the file from your computer to a server (a special computer that stores files).
* **Receiving Files:** The other person can then use FTP to download the file from the server to their own computer.

**SMTP** stands for Simple Mail Transfer Protocol. It's a system that helps send emails from one computer to another.  
**How SMTP Works:**

1. **Composing an Email:** When you write an email and hit "send," your email client (like Gmail, Outlook, etc.) uses SMTP to start the delivery process.
2. **Sending the Email:** SMTP takes your email, figures out where it needs to go (based on the recipient's email address), and sends it off to the recipient's email server.
3. **Receiving the Email:** The recipient's email server then stores the email in their inbox, where they can retrieve it using another protocol like IMAP or POP3.

**Example:**

* **Sending a Message:** Let's say you want to send an email to your friend Sarah. You type the message and press "send." SMTP takes over, sending your email from your email server to Sarah's email server, where she can then open and read it.

HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) are both protocols used for transferring data over the web, but they differ in terms of security.

**1. HTTP (Hypertext Transfer Protocol):**

* **Unencrypted:** HTTP is the standard protocol for transferring data between a web browser (like Chrome, Firefox, etc.) and a web server. However, the data sent and received using HTTP is not encrypted, meaning it's in plain text.
* **Vulnerability:** Because the data is not encrypted, it's possible for someone with the right tools to intercept and read the information being transmitted. This can include sensitive data like passwords or credit card numbers.
* **Usage:** HTTP is commonly used for websites where security is not a primary concern, such as informational websites or blogs.

**2. HTTPS (Hypertext Transfer Protocol Secure):**

* **Encrypted:** HTTPS is the secure version of HTTP. It uses encryption (via SSL/TLS certificates) and secure version of HTTP, where data between your browser and the website you’re visiting is encrypted, ensuring privacy and security. This means that even if someone intercepts the data, they won't be able to read it because it's encrypted.
* **Security:** HTTPS ensures that the data sent and received is secure and private. It protects against eavesdropping, tampering, and man-in-the-middle attacks.
* **Usage:** HTTPS is used on websites where security is critical, such as online banking, shopping sites, and any site that requires users to enter sensitive information like passwords or payment details.
* **Browser Indication:** Most browsers show a padlock icon in the address bar when you're on an HTTPS site, indicating that the connection is secure.

**Summary of Differences:**

* **Encryption:** HTTP is unencrypted, while HTTPS encrypts data for security.

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are both cryptographic protocols designed to secure communication over a computer network

SSL/TLS ensures that the data sent and received hasn’t been tampered with. If the data is altered in any way during transmission, SSL/TLS can detect this and will terminate the connection to prevent compromised data from being delivered.  
  
  
DNS stands for **Domain Name System**.   
DNS is the system that translates the domain name you type into the corresponding IP address, allowing your browser to find and connect to the right website.

**How DNS Works:**

1. **You Type a Domain Name:** You enter a web address (like www.example.com) into your browser.
2. **DNS Lookup:** Your browser sends a request to a DNS server to find out the IP address associated with that domain name.
3. **Finding the IP Address:** The DNS server looks up the domain name in its records and responds with the correct IP address.
4. **Connecting to the Website:** Your browser uses the IP address to connect to the website's server and load the webpage.

DHCP **Dynamic Host Configuration Protocol** is a protocol that automates the process of assigning IP addresses to devices (like computers, smartphones, printers, etc.) on a network. Instead of manually configuring each device with a static IP address, DHCP automatically assigns IP addresses from a pool of available addresses. It also provides other necessary configuration details like the subnet mask, default gateway, and DNS server addresses.

TCP (Transmission Control Protocol) TCP is a connection-oriented protocol, meaning that a connection is established and maintained until the application at each end has finished exchanging messages. TCP breaks data into smaller packets, sends them to the recipient, and ensures that all packets arrive in the correct order.  
**Example Use Cases:**

* **Web Browsing:** When you load a webpage, your browser uses TCP to request and receive the page data from a web server. This ensures that all elements of the page load correctly.
* **Email:** Sending an email via protocols like SMTP relies on TCP to ensure that the entire message arrives intact.

**UDP (User Datagram Protocol)**

* **What It Is:** UDP is a connectionless protocol, meaning that data is sent without establishing a connection between the sender and receiver. UDP is faster than TCP but less reliable, as it doesn't guarantee that all data will be received or that it will arrive in the correct order.

**Example Use Cases:**

* **Live Streaming:** When you watch a live video stream or listen to live audio, UDP is often used because speed is more important than perfect accuracy. A few lost packets won't significantly affect the stream, and retransmitting them would cause delays.
* **Use Cases:**
  + **TCP:** Used where accuracy and reliability are critical (e.g., web browsing, email).
  + **UDP:** Used where speed is essential and some data loss is acceptable (e.g., live streaming, gaming).

**Example in Practice:**

* **TCP Example:** When you download a file from the internet, TCP ensures that every part of the file is received correctly. If any part is missing or out of order, TCP will request it again from the server.
* **UDP Example:** When making a voice call over the internet (VoIP), UDP is used to send the audio data. Some packets might get lost, but the conversation continues without noticeable delays, which is more important than perfect accuracy.
* **Use Cases:**
  + **TCP:** Used where accuracy and reliability are critical (e.g., web browsing, email).
  + **UDP:** Used where speed is essential and some data loss is acceptable (e.g., live streaming, gaming).

**Example in Practice:**

* **TCP Example:** When you download a file from the internet, TCP ensures that every part of the file is received correctly. If any part is missing or out of order, TCP will request it again from the server.
* **UDP Example:** When making a voice call over the internet (VoIP), UDP is used to send the audio data. Some packets might get lost, but the conversation continues without noticeable delays, which is more important than perfect accuracy.

In networking, the terms **Layer 2 (L2)**, **Layer 3 (L3)**, and others like Layer 1 (L1) and Layer 4 (L4) refer to different levels in the **OSI (Open Systems Interconnection) model**. The OSI model is a conceptual framework that standardizes the functions of a telecommunication or networking system into seven distinct layers. Each layer in the OSI model has specific responsibilities and interacts with the layers directly above and below it. This model helps different networking and telecommunication technologies to interoperate smoothly by ensuring they follow a common set of standards.Here’s a simplified explanation of these layers, focusing on Layers 2 and 3, since those are the most commonly discussed in networking.

**Layer 1: Physical Layer (L1)**

* **What It Is:** The Physical Layer is the first layer of the OSI model, and it deals with the actual physical connection between devices. It includes the hardware elements like cables, switches, and network interface cards, and it manages the transmission and reception of raw data bits over the physical medium.
* **Example:** Ethernet cables, Wi-Fi signals, and fiber optics are all part of Layer 1.

**Layer 2: Data Link Layer (L2)**

* **What It Is:** The Data Link Layer is responsible for establishing a reliable link between two directly connected nodes (devices). It handles the physical addressing (using MAC addresses), error detection and correction, and controls how data is placed on the network medium.
* **Functions:**
  + **MAC Addresses:** Layer 2 uses MAC addresses to ensure data is sent to the correct physical device on a local network.
  + **Switches:** Devices like network switches operate at Layer 2, directing data based on MAC addresses.
* **Example:** When you send a file to a printer on the same local network, the switch uses Layer 2 to send the data directly to the printer.

**Layer 3: Network Layer (L3)**

* **What It Is:** The Network Layer is responsible for moving data between different networks. It handles logical addressing (using IP addresses) and routing, which involves determining the best path for data to travel from the source to the destination across different networks.
* **Functions:**
  + **IP Addresses:** Layer 3 uses IP addresses to identify devices and route data to the correct network.
  + **Routers:** Routers operate at Layer 3, directing data between different networks based on IP addresses.
* **Example:** When you visit a website, your data travels from your local network (using a router) to the internet, which involves Layer 3 routing.

**Layer 4: Transport Layer (L4)**

* **What It Is:** The Transport Layer is responsible for ensuring that data is delivered error-free, in sequence, and with no losses or duplicates. It manages end-to-end communication between devices, using protocols like TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).
* **Functions:**
  + **TCP/UDP:** These protocols handle things like breaking data into segments, managing data flow, and retransmitting lost data.
* **Example:** When you stream a video, Layer 4 ensures that the data packets arrive in the correct order and that any missing packets are retransmitted.

Questions:

Killall is not kiiling the process i used command killall teams or killall Notepad but in each case i get error how this commands work