KODEKLODE LINUX

What is linux?

Why we use linux?

Why do we care about operating system ?

Linux is by far the most widely used server operating system in the world, covering all its different versions and flavors like Ubuntu, CentOS, Red Hat. According to recent reports, all 500 of the fastest supercomputers run on Linux, 96.3% of the top websites are powered by Linux, and surprisingly, 86% of all smartphones use Linux-based operating systems. This shows how important and popular Linux is in modern technology

How does this relate to devops?

In the cloud and DevOps world, most new tools are first built for Linux. This is because Linux is widely used in servers and data centers. Later, some tools get support for Windows, but Mac is rarely used for servers.

For example, Docker, which helps run apps in containers, was only available on Linux for many years. Only recently did it start working on Windows.

When it comes to Kubernetes, which manages containers, the main part (called the control plane or master nodes) can only run on Linux. Even if your apps run on Windows servers, you still need Linux for the control plane. There are no plans to make a Kubernetes cluster that works only on Windows.

So, when you learn Kubernetes and set up a cluster, you always use Linux for the control plane. This is why Linux is very important for DevOps and cloud work.

Working in a Windows-dominant environment and switching to Linux can bring some challenges, including:

- The Linux command line is text-based and works differently than Windows, so it can feel confusing.

- Linux organizes files and folders in a different way, so finding things takes some getting used to.
- Text editors like vi or vim are keyboard-focused and not like Windows editors.
- There are many types of Linux (Ubuntu, CentOS, etc.), each with its own tools and commands.
- Installing software can be tricky because Linux uses different package managers (like apt, yum, rpm) and sometimes you get errors with missing files or dependencies.
- Setting up networking between Linux virtual machines can be harder than in Windows.
- Linux has strict permission settings, so you might run into issues with files or folders you can’t access.
- Without regular practice, it’s easy to forget commands and how things work.

What is linux shell?

The linux shell is a program that allows text-based interaction between the user and operating system

All types of shells in Linux??

- **Bash (Bourne Again Shell): is the most popular shell in Linux. It is widely used because it has useful features like it suggests commands as you type and lets you create multiple files or folders at once, which are not available in some other shells. Most Linux systems use Bash as the default shell.**
- **Zsh (Z Shell)**:
- **Ksh (Korn Shell):**
- **Csh (C Shell)**:

How can you check which shell you are currently using in Linux?

```
echo $SHELL  #$SHELL  is a environment variable
```

How do you change shell?

First check where the Bourne shell is installed (usually /bin/sh)

```
cat /etc/shells #shows a list of all the available shells on your Linux system.
```

Change shell:

```
csudo chsh -s /path/to/shell #Replace /path/to/shell with the path of the shell you want, like /bin/bash or /bin/zsh.
csudo chsh -s /bin/sh  #-s stand for shell
```

Difference between a variable and an environment variable??

- A variable is like a small box where you can store a piece of information, like a name or a number, for use in your current program or terminal. If you close the program, the box disappears.
- An environment variable is a special kind of box that can be seen and used by many different programs and processes. It stays available as long as your computer is running and can be used by any program that needs it.
- So, use a variable for temporary storage in one program, and use an environment variable to share information between different programs.

How to create and share environment variable from one program to another??

On Linux/macOS:

```
export MY_VAR=value
#After this, you can use $MY_VAR in your shell or scripts.
```

On Windows:

```
set MY_VAR=value
```

How to check value of environment variable?

```
echo $variablename
```

How to see the list of all environment variable??

```
env
```

How to check path?

```
echo $PATH
```

Where is a user's home directory located in Linux, and how is it named?

In Linux, each user has a home directory, which is usually created under the /home folder. The name of the home directory is the same as the username. For example, if the user is named "Bhoomi," their home directory will be /home/bhoomi. Every user has their own unique home directory, so no two users share the same home folder.

Why do we need home directory??

In Linux, the home directory is a special place where each user can store their personal files and folders. Every user has their own unique home directory, and only that user can access it. This means no other user can see or change your files unless you give permission. You can quickly find your home directory by typing `~` (tilde) in the terminal.

Note: Once you are logged into bash shell, the line you see is called the bash prompt

~ = present working directory

$ = User prompt symbol

File Permissions:

Linux is a multiuser system. Every file and directory in your account can be protected from other users by changing its access permissions.

Type of permissions

- r - read → 4
- w - write → 2
- x - execute = running a program ⟶ 1

Total sum = 1+2+4 = 7 (all permissions)

Each permission (rwx) can be controlled at three levels.

- u - user = yourself
- g - group = can be people in the same Project
- o - other = everyone on the system

Command to change the permission

- **chmod 777 <file/dir>**
- First 7 → for User
- Second 7→ for group
- Third 7 → other group

How you give permission to files or folders in Linux

To give permission to files or folders in Linux, you use commands like chmod and chown:

1. Change ownership (if needed):

Use the chown command to assign a specific user or group as the owner.

Example:

```
sudo chown username:groupname filename
#Replace username and groupname with the actual names
```

2. Set permissions:

Use the chmod command to set read, write, and execute permissions.

```
chmod u+rwx filename
# This gives the owner (user) read, write, and execute permissions.
```

3. For numeric mode (common for scripts):

```
chmod 755 filename
# Here,7 means read+write+execute for the owner,5 means read+execute for group and others.
```

4. Check permissions:

Use the “ ls -l” filename command to see the current permissions and ownership.

5. Best practices:

- Only give the minimum permissions needed.

```
chmod -R 755 foldername
#chmod changes permissions on files and folders.
#-R means "recursive," so it applies the permission change to all files and sub folders inside the specified folder.
#The owner (you) gets read, write, and execute permissions (7).
#Group members and all other users get read and execute permissions (5),but cannot write or modify files.
```

This is commonly used for folders because it lets everyone see and open files inside, but only the owner can add, delete, or change files. For example, if you run ”chmod -R 755 myfolder”, every file and folder inside myfolder will have these permissions.

Why is the Linux command-line interface important compared to the graphical interface?

The Linux command-line interface (also called the shell) lets you work on your Linux computer more effectively. While the graphical interface looks easier and more attractive for new users, it has limits in what it can do. The command-line is more powerful and gives you more control to perform tasks and manage your system.

What is graphical interface?

A graphical interface, also called a GUI (Graphical User Interface), is a way to interact with a computer using visual elements like icons, buttons, windows, and menus instead of typing commands. You can click on these elements with a mouse or touchpad to open files, run programs, or change settings, making it easier for beginners to use computers. Most operating systems like Windows, macOS have a graphical interface to help users work without needing to remember complex commands

why graphical interface not good?

A graphical interface (GUI) is not always good because it can be slow and doesn’t let you do everything you need, especially on servers. It uses more computer power, which is not good for systems that need to run fast and smoothly.

Linux has a GUI to help beginners use the system easily by clicking icons and menus. But for advanced work, servers, and cloud environments, Linux mostly uses the command-line (CLI) because it is faster, uses less power, and gives more control over the system.

What are type of command in linux?

1. Internal or build-in Commands: Internal commands are built right into the shell, so they work immediately and don't need any extra files. they are always available when you use the shell.

```
echo,cd,pwd etc
```

1. External Commands: External commands are separate programs or scripts stored as files in the system, usually in directories like `/bin` or `/usr/bin`. so your system must find them to run them

```
ls,cat,grep #These can be pre-installed with the OS or added by the user.
```

Note: To check if a command is internal or external, use the type command. For example,

"type echo" will say it's a shell built-in (internal), but "type ls" will show a file path (external)

what is command, option, flags, switch ,argument??

```
cp -r folder1 folder2

#Option/Flag/Switch: Extra settings you add to change how the command wor
ks.
#Argument: The specific file, folder, or value the command works on.
```

- `cp` is the command (copy).
- `r` is the option/flag/switch
- `folder1` and `folder2` are the arguments (the folders to copy from and to).

So, options/flags/switches are extra instructions, and arguments are the things the command acts on.

What is absolute path and relative path?

- An absolute path is the full path to a file or directory, starting from the root directory (/). It always begins with a / and gives the complete address. For example, if you are in /home/bhoomi and want to go to a directory called asia inside your home directory, the absolute path would be /home/bhoomi/asia.
- A relative path is the path in relation to your current directory. It does not start with a / and is based on where you are right now. For example, if you are in /home/bhoomi, you can simply run cd asia to enter the asia directory. Here, asia is a relative path because it is relative to your current locatio

Note: The alternative of this is “pushd” command

Difference between dir and pwd?

- The dir command is used to list the contents of a directory, showing files and folders inside it, similar to the ”ls” command in Linux.
- The pwd command stands for "print working directory." It shows the full path of the directory you are currently in. For example, if you are in /home/bhoomi, running pwd will display /home/bhoomi

What is mv Command??

Moves a file or directory from one place to another. The original is removed from its old location and placed in the new location. It can also be used to rename files

This command require two argument

1. The first argument is the source directory, current file location
2. The second argument is the destination directory, Where we want it to be moved

Using absolute path, this can be done easily using the command

```
mv /home/bhoomi/Europe/Morroco /home/bhoomi/africa/
```

or

Using relative path

```
mv Europe/Morroco Africa/
```

How to renamed directory name?

We use “mv” command

```
# mv source directory destination directory
mv Asia/India/Mumbai Asia/India/delhi
```

What is cp command do??

Makes a duplicate of a file or directory. The original stays in place, and a copy is created in the new location.

```
# mv source directory destination directory
mv Asia/India/delhi/city.txt Africa/Egypt/Cairo
```

The main difference between cp and mv in Linux is

- **cp (copy)**: Makes a duplicate of a file or directory. The original stays in place, and a copy is created in the new location.
- **mv (move)**: Moves a file or directory from one place to another. The original is removed from its old location and placed in the new location. It can also be used to rename files

How do you check if something is file or folder?

```
ls -l foldername/filename
```

You can tell if something is a file or a directory by looking at the very first character in the output of ls -l:

- If the first character is a -, it means it's a regular file.
- If the first character is a d, it means it's a directory.

What rm command do??

```
rm Cairo
```

This will delete the file named Cairo from your current directory.

Note: The rm command deletes files permanently, so be careful. If the file is protected, you may need to confirm before deleting it

What cat command do?

The cat command is used to display the contents of a file. For example, cat filename shows what’s inside that file

```
cat file.txt
```

What is “cat > filename” command do?

```
cat > file.txt
# > forward arrow symbol
```

The cat > command is used to create a new file or overwrite an existing file with new content.

For example, cat > filename lets you type text, and when you press Ctrl+D, it saves that text in the file. If the file already exists, its content will be replaced

What touch command do?

The touch command in Linux is used to create empty files

```
touch myfile.txt
```

what vim command do?

The vim command opens the Vim text editor, which lets you create, edit, and manage text files directly from the termina

```
vim file.txt
```

Some basic vim commands:

- “i” to enter insert mode and start typing.
- “Esc” to exit insert mode and return to command mode.
- “:wq” to save changes and quit.
- “:q!” to quit without saving.
- “dd” to delete a line.
- “u” to undo the last change.
- “yy” to copy (yank) a line, p to paste it.

What “more” or “less” command do?

you can view text file in scrollable manner

```
more file.txt
```

What ls(Long list) command do??

ls is useful to list the contents of directory in the linux file system

```
ls
ls -l #provide more details related for file and folder such as ownership or che
ck something is file or folder
ls -a #list all files including hidden and hidden files are those that start with a d
ot (.)
ls -ltr #to do reverse and print files from oldest to newest, use the
```

what mkdir(make directories) command do?

The mkdir command in Linux is used to create new directories (folders)

```
mkdir directories_name
mkdir snake frog  # This creates two directories: snake and frog (both at the s
ame level), not nested inside each other.
mkdir -p /home/bob/fish/salmon  # this is nested directories one inside anoth
er
#-p stand for parents
```

What is operating system??

An operating system (OS) is the main software that makes your computer work. It manages everything like running programs, storing files, and connecting to devices. Examples are Windows, Linux, and macOS. The OS helps you use your computer easily by handling all the basic tasks behind the scenes.

What is linux kernel?

The Linux kernel is the core part of the Linux operating system. It acts as the main bridge between your computer's hardware (like the processor and memory) and the software (like programs and apps).

The kernel is responsible like

- Memory management- keeps track of how much memory is used to store what and where
- Process management- determine which processes can use the CPU, when and for how long
- Device Drivers- act as a mediator or an interpreter between the hardware and processes

Memory is divided into two areas

- **Kernel space:** This is the protected area of memory where the kernel (the core of the operating system) runs and manages hardware, processes, and system

resources. Only the kernel can access this space, which helps keep the system secure and stable. it is used by the operating system kernel like managing hardware, memory, and security

- **User space:** This is the area where user programs and applications run. Each program gets its own part of user space, and cannot access kernel space or the memory of other programs. This separation keeps programs safe and prevents them from interfering with the system or each other. **User space** is used by regular programs and applications, like web browsers, text editors

why we use sudo?

We use sudo to do important tasks in Linux, like installing software or changing system settings. Regular users don’t have permission to do these things for security reasons. When you use sudo, you temporarily get extra permissions to do these tasks safely. This helps protect the system from mistakes or harmful changes

What is the use of sudo su?

sudo su is a Linux command that switches you to the root user (full admin) using your own user’s sudo permissions.

How to helps protect the system from mistakes or harmful changes??

Using sudo helps protect the system by making sure only authorized users can do important tasks. When you use sudo, you must enter your password, which proves you are allowed to do that action. This stops accidental or harmful changes, like deleting important files or changing system settings, because regular users can't do these things without permission.

What is boot process?

The boot process is what happens when you turn on your computer.

There are four type of boot process

1. **BIOS POST (Power-On Self-Test):**

When you turn on your computer, the BIOS checks if all hardware is working. If something is wrong, the computer stops and shows an error.

2. **Bootloader:**

   After the hardware check, the BIOS loads a small program called the bootloader (like GRUB2) from the hard disk. The bootloader lets you choose which operating system to start and then loads the Linux kernel into memory.

3. **Kernel Initialization:**

   The kernel (the core of Linux) is loaded and starts running. It sets up hardware, manages memory, and prepares the system for running programs. Once ready, it looks for the next step.

4. **Init Process:**

   The kernel starts the init process, which is usually systemd these days. Systemd starts up all the services and programs needed for you to use the system. In the past, a different process called SysV-Init was used, but systemd is faster because it starts services at the same time.

5. **Login Prompt:** Once everything is ready, you get a screen to log in and use your computer.

what is systemd?

Systemd is the main program that starts up your Linux computer. It loads all the necessary services and programs so your system is ready to use. It is faster than older systems because it starts everything at once, not one by one. Most Linux computers now use systemd

What systemd command do?

Used to manage services and the system

```
sudo systemctl start servicename #Start a service
sudo systemctl stop servicename  #Stop a service
```

```
sudo systemctl restart servicename #Restart a service
systemctl status servicename  #Check status of a service
sudo systemctl enable servicename  #Enable a service (start at boot
sudo systemctl disable servicename #Disable a service (don't start at boot)
systemctl list-units --type=service  #List all services
```

What is the difference between start the service and enable the service?

- Starting a service means running it right now, but it won’t start automatically next time you restart your computer.
- Enabling a service means setting it up to start automatically every time your computer boots. Enabling does not start the service immediately; it just makes sure it will run when the system starts.

What is files?

In Linux, "files" are storage used to save information like text, programs

Types of files

- Regular files: These store data like text, images, scripts, or programs. They are shown with a - in the ls -l output.
- Directories: These are folders that contain other files and directories. They are shown with a d in the ls -l output.
- Symbolic links: These are shortcuts to other files or directories. They are shown with an l in the ls -l output.
- Character device files: These represent devices that handle data byte by byte, like keyboards or mice. They are shown with a c in the ls -l output.
- Block device files: These represent devices that handle data in blocks, like hard disks. They are shown with a b in the ls -l output.
- Named pipes (FIFO): These allow processes to exchange data. They are shown with a p in the ls -l output.

- Sockets: These are used for inter-process communication. They are shown with an s in the ls -l output

Linux File System Structure

Folder	What It Stores?
`/` **(Root)**	The **main** directory (everything starts here).
`/home`	Stores **user files** (documents, downloads).
`/bin`	Basic **system commands** (`ls` , `cp` , `cat`).
`/sbin`	Commands for **admin users** (`shutdown` , `fdisk`).
`/etc`	Holds system configuration files. ex: User, network , system app
`/var`	Where variable data such as logs and databases are stored
`/usr`	User-installed **programs and apps**.
`/lib`	/lib is a folder that stores important system files to help Linux run commands and programs correctly
`/tmp`	Temporary files created by system and users
`/dev`	/dev is a special folder where Linux stores files that control hardware devices, like hard drives, USBs, and keyboards!
`/mnt` **&** `/media`	**External drives** (USB, CD).
`/boot`	Files needed to **start the system** (Linux Kernel, GRUB).
`/proc`	**System and process info**.
`/opt`	Third-party **software and applications**.
`/root`	Home folder for the **admin (root) user**.

Why is the File System Important?

- **Keeps files organized**
- **Controls file access (security)**
- **Manages storage devices (HDD, SSD, USB)**

File Path Format in Linux vs Windows ??

- **Linux paths** use `/` and start from `/` (root).
- Windows paths use \ and start from C:\ (drive letter).

Types of file system?

1. ext4(Fourth Extended File System): Default Linux file system
2. **XFS(Extended File System) :** Good for large files
3. Btrfs(B-Tree File System) : Supports backups & recovery
4. **FAT32(File Allocation Table 32-bit) :** Works with Linux, Windows & USB drives
5. exFAT(Extended File Allocation Table) : Best for USB drives & external storage

Note: For high-throughput and extensive parallel processing capabilities , XFS is typically more effective than ext4

How to check your linux Filesystem ??

- lsblk: Shows **file system type** for all storage devices.
- df -Th: Shows **file system type and disk usage**.
- cat/etc/fstab: Displays **file system details** used when the system starts.

What is package manager?

A package manager is a tool that makes it easy to install, update, and remove software

- **apt-get (Advanced Package Tool):** Used in Ubuntu
- **yum (Yellowdog Updater, Modified):** Used in amazon
- **npm** (Node Package Manager) is a tool used in JavaScript and Node.js to install, update, and manage software packages and libraries

Difference between apt and apt-get?

apt and apt-get are both package managers used in Debian-based Linux systems (like Ubuntu) to install, update, and remove software. The main differences are:

- apt is a newer, user-friendly and already installed by default command. It provides a progress bar, and gives more helpful output during tasks. It’s meant for everyday users.
- apt-get is an older, low-level command. It’s more predictable and is often used in scripts or by system administrators because it remains backward compatible. It doesn’t show a progress bar and has less user-friendly output.

what is redhat?

Red Hat is a company that creates Linux-based operating systems for businesses and organizations.

what is Redis?

Redis is an open-source, in-memory data structure server, often used as a database, cache, or message broker. It stores data in memory, which makes it extremely fast for reading and writing. Redis supports various data types like strings, lists, hashes, sets, and more. It's widely used for real-time applications, caching, session storage, and messaging because of its speed and flexibility.

What is telnet?

Telnet is a simple tool that lets you connect to a remote computer over the network. Once connected, you can use keyboard commands to control the remote system, just like if you were sitting right in front of it. You can run programs, create or delete files, manage folders, start or stop services, and check if ports are open or closed. This way, you can do almost anything on a remote server, even if you are far away

What is nginx?

Nginx is a lightweight and fast web server that can handle many users at the same time. It is designed to serve static content (like images, HTML, CSS, and JavaScript) very quickly and efficiently. Nginx can also work as a reverse proxy, meaning it can forward requests from users to other servers and send back the response. This helps in managing high traffic and makes websites faster and more reliable. Many big websites use Nginx because it uses less memory and

resources, scales easily(means that Nginx can handle more users or traffic by simply adding more servers or resources, without major changes to the system) , and provides a smooth experience even during high traffic.

Nginx Features?

- **Scalability:**

  Nginx can handle many users at once and can easily add more servers to manage increased traffic, making it great for high-traffic websites.

- **Load Balancing:**

  Nginx distributes incoming traffic across multiple servers, preventing any one server from getting overloaded and improving reliability.

- **Caching:**

  Nginx stores copies of frequently requested files, so it can serve them quickly without reprocessing, making websites faster.

- **Compression:**

  Nginx compresses files before sending them to users, reducing file size and speeding up page loads

At a time nginx is connected to many server??

Yes, Nginx can connect to many servers at the same time. This is especially useful for load balancing, where Nginx distributes incoming traffic across multiple backend servers. Nginx can also serve multiple websites or applications on the same machine by using server blocks (virtual hosts). This means you can run many services or websites behind a single Nginx instance, making it efficient and scalable

What is web server ?

A web server is **a system (hardware, software, or both) that stores, processes, and delivers web content like webpages, images, and videos to users over the internet via their browsers using the HTTP protocol**

Difference between Apache and nginx??

- Apache is flexible(creates a new process for each request) and good for dynamic websites. It uses more memory and CPU, especially with many users
- Nginx is fast and efficient for static websites and high traffic.

Dynamic website and Static websites?

Dynamic websites are sites that change content based on user actions, time, or other factors. Examples include:

- **Amazon:** Shows personalized product recommendations and updates prices and availability in real time.
- **Instagram:** Updates your news feed and shows new posts as they happen.

Static websites are sites that show the same content to every visitor and do not change unless manually updated. Examples include:

- **Personal blogs**
- **Company or portfolio websites**
- **Image or video galleries**

How we can check the size of a linux?

The “du” command, which stand for disk usage, is a popular command to inspect the size of the files.

```
du -sh test.img
#du: Stands for "disk usage" and is used to estimate the space used by files o
r directories.
#-s: Shows only the total size of the specified file or directory, not individual s
ubdirectories or files.
#-h: Formats the output in a human-readable way (e.g., KB, MB, GB), making i
t easier to understand the size.
```

What is the use of “tar” command?

Tar is used to group multiple files or directories into s single file, it is especially useful for archiving data.

```
tar -cf test.tar file1 file2 file3
#-c: Stands for "create" — tells tar to create a new archive
#-f: Specifies the filename for the archive. In this case, it's test.tar.
#test.tar: The name of the output archive file.
#file1 file2 file3: The list of files to include in the archive.
```

```
tar -tf test.tar
#-t: Stands for "list" — tells tar to show the files inside the archive.
#-f: Specifies the archive filename, in this case test.tar.
```

```
tar -xf test.tar
#-x: Stands for "extract" — tells tar to unpack the contents of the archive.
#-f: Specifies the filename of the archive to extract, in this case test.tar.
```

```
tar -zcf test.tar file1 file2 file3
#tar: The command used to create, extract, or manage archive files.
#-z: Tells tar to compress the archive using gzip, resulting in a compressed .ta
r.gz file
#-c: Stands for "create" — tells tar to create a new archive.
#-f: Specifies the filename for the archive, in this case test.tar.
```

What is compression?

Compression is the technique used to reduce the size by file

```
bzip2 test.img
gzip test1.img
```

```
xz test2.img
gunzip test.img
```

What locate command work?

The locate command is used to quickly find files and directories by name

```
locate city.txt
```

```
#example
sudo find /etc -type f -name "dummy.service" > /home/bob/dummy-service
#sudo: Runs the command with administrative privileges, ensuring access to all directories under /etc, even those restricted to regular users.
#find: The command used to search for files and directories.
##/etc: The starting directory for the search.
#-type f: Limits the search to regular files (not directories or links).
#-name "dummy.service": Specifies the filename to search for.
#>: Redirects the output (the list of matching file paths) to a file instead of dis playing it on the screen.
```

What is the use to grep command?

The grep command is used to search for specific text patterns within files, such as words, phrases.

```
grep word_name file_name
grep second file.txt


#-i: Case-insensitive search.
#-n: Shows line numbers of matches.
#-r: Searches recursively in directories.
#-w: Matches whole words only.
#-v: Shows lines that do not match the pattern.
```

Redirect the output

The > operator redirects standard output (stdout) to a file, overwriting its contents if the file exists.

```
echo $SHELL > shell.txt #overwriting any existing content in the file
#output: /bin/bash
```

The >> operator appends stdout to a file, adding new content at the end.

```
echo "this is the bash shell" >> shell.txt
#appends the string "this is the bash shell" to the end of shell.txt without delet
ing what was already there.
```

What is Command line pipes (|) ??

Command line pipes (`|`) are used to connect multiple commands together, so that the output of one command becomes the input for the next. command1 | command2 | command3, and each will process the data passed from the previous one

```
ls | grep ".txt"  #lists all files in the current directory and then filters only those
with a .txt extension
ps aux | grep "nginx" | awk '{print $2}' #shows all processes, filters for those
named "nginx", and then prints only the process IDs
```