UNIX

In order to understand the popularity of Linux, we need to travel back in time, about 30 years ago...

Imagine computers as big as houses, even stadiums. While the sizes of those computers posed substantial problems, there was one thing that made this even worse: every computer had a different operating system. Software was always customized to serve a specific purpose, and software for one given system didn't run on another system. Being able to work with one system didn't automatically mean that you could work with another. It was difficult, both for the users and the system administrators.

Computers were extremely expensive then, and sacrifices had to be made even after the original purchase just to get the users to understand how they worked. The total cost per unit of computing power was enormous.

Technologically the world was not quite that advanced, so they had to live with the size for another decade. In 1969, a team of developers in the Bell Labs laboratories started working on a solution for the software problem, to address these compatibility issues. They developed a new operating system, which was

1. Simple and elegant.

2. Written in the C programming language instead of in assembly code.

3. Able to recycle code.

The Bell Labs developers named their project "**UNIX**."

The code recycling features were very important. Until then, all commercially available computer systems were written in a code specifically developed for one system. UNIX on the other hand needed only a small piece of that special code, which is now commonly named the kernel. This kernel is the only piece of code that needs to be adapted for every specific system and forms the base of the UNIX system. The operating system and all other functions were built around this kernel and written in a higher programming language C.

Introduction to Linux

Linux is an open-source Unix-like operating system-based family on the Linux kernel, and the OS kernel was first published on 17 September 1991 by Linus Torvalds. Typically, Linux is packaged as the Linux distribution, which contains the supporting libraries and system software and kernel, several of which are offered by the GNU Project. Several Linux distributions use the term "Linux" in the title, but the Free Software Foundation uses the "GNU/Linux" title to focus on the necessity of GNU software, causing a few controversies.

Famous Linux distributions are Ubuntu, Fedora Linux, and Debian, the latter of which is composed of several different modifications and distributions, including Xubuntu and Lubuntu. Commercial distributions are SUSE Linux Enterprise and Red Hat Enterprise Linux. Desktop distributions of Linux are windowing systems like Wayland or X11 and desktop environments like KDE Plasma and GNOME.

Originally, Linux was designed for personal computers that were Intel x86 architecture-based, but it have since been moved to more environments than other operating systems.

Including Android, Linux has the biggest installed base of every general-purpose operating system because of the control of the Linux-based Android over smart phones as of May 2022.

However, Linux is used by just around 2.6% of desktop computers as of November 2022.

Linux is one of the most outstanding examples of open-source and free software collaboration. The source code may be distributed, modified, and used non-commercially or commercially by everyone under the conditions of its respective licenses, like the GNU GPL (General Public License). For example, the Linux kernel is licensed upon the GPLv2.

Uses of Linux OS

- Web servers
- Laptops and desktops
- Mobiles devices
- Film production
- Government uses

Advantages of Linux

Linux is an open-source operating system like Windows and MacOS. It is not just limited to the operating system, but nowadays, it is also used as a platform to run desktops, servers, and embedded systems. It provides various distributions and variations as it is open source and has a modular design. The kernel is a core part of the Linux system.

Linux system is used to manage various services such as process scheduling, application scheduling, basic peripheral devices, file system, and more. Linux provides various advantages over other operating systems such as Windows and macOS. So, it is used in almost every field, from cars to home appliances and smartphones to servers (supercomputers).

Following are top 20 advantages of the Linux operating system:

1. Open Source

As it is open-source, its source code is easily available. Anyone having programming knowledge can customize the operating system. One can contribute, modify, distribute, and enhance the code for any purpose.

2. Security

The Linux security feature is the main reason that it is the most favorable option for developers. It is not completely safe, but it is less vulnerable than others. Each application needs to authorize by the admin user. The virus is not executed until the administrator provides the access password. Linux systems do not require any antivirus program.

3. Free

Certainly, the biggest advantage of the Linux system is that it is free to use. We can easily download it, and there is no need to buy the license for it. It is distributed under GNU GPL (General Public License). Comparatively, we have to pay a huge amount for the license of the other operating systems.

4. Lightweight

Linux is lightweight. The requirements for running Linux are much less than other operating systems. In Linux, the memory footprint and disk space are also lower. Generally, most of the Linux distributions required as little as 128MB of RAM around the same amount for disk space.

5. Stability

Linux is more stable than other operating systems. Linux does not require to reboot the system to maintain performance levels. It rarely hangs up or slow down. It has big up-times.

6. Performance

Linux system provides high performance over different networks. It is capable of handling a large number of users simultaneously.

7. Flexibility

Linux operating system is very flexible. It can be used for desktop applications, embedded systems, and server applications too. It also provides various restriction options for specific computers. We can install only necessary components for a system.

8. Software Updates

In Linux, the software updates are in user control. We can select the required updates. There a large number of system updates are available. These updates are much faster than other operating systems. So, the system updates can be installed easily without facing any issue.

9. Distributions/ Distros

There are many Linux distributions available in the market. It provides various options and flavors of Linux to the users. We can choose any distros according to our needs. Some popular distros are Ubuntu, Fedora, Debian, Linux Mint, Arch Linux, and many more.

For the beginners, Ubuntu and Linux Mint would be useful and, Debian and Fedora would be good choices for proficient programmers.

10. Live CD/USB

Almost all Linux distributions have a Live CD/USB option. It allows us to try or run the Linux operating system without installing it.

11. Graphical User Interface

Linux is a command-line based OS but, it provides an interactive user interface like Windows.

12. Suitable for programmers

It supports almost all of the most used programming languages such as C/C++, Java, Python, Ruby, and more. Further, it offers a vast range of useful applications for development.The programmers prefer the Linux terminal over the Windows command line. The package manager on Linux system helps programmers to understand how things are done. Bash scripting is also a functional feature for the programmers. It also provides support for SSH, which helps in managing the servers quickly.

13. Community Support

Linux provides large community support. We can find support from various sources. There are many forums available on the web to assist users. Further, developers from the various open source communities are ready to help us.

14. Privacy

Linux always takes care of user privacy as it never takes much private data from the user. Comparatively, other operating systems ask for the user's private data.

15. Networking

Linux facilitates with powerful support for networking.

The client-server systems can be easily set to a Linux system. It provides various command-line tools such as ssh, ip, mail, telnet, and more for connectivity with the other systems and servers. Tasks such as network backup are much faster than others.

16. Compatibility

Linux is compatible with a large number of file formats as it supports almost all file formats.

17. Installation

Linux installation process takes less time than other operating systems such as Windows. Further, its installation process is much easy as it requires less user input. It does not require much more system configuration even it can be easily installed on old machines having less configuration.

18. Multiple Desktop Supports

Linux system provides multiple desktop environment support for its enhanced use. The desktop environment option can be selected during installation. We can select any desktop environment such as GNOME (GNU Network Object Model Environment) or KDE (K Desktop Environment) as both have their specific environment.

19. Multitasking

It is a multitasking operating system as it can run multiple tasks simultaneously without affecting the system speed.

20. Heavily documented for beginners

There are many command-line options that provide documentation on commands, libraries, standards such as manual pages and info pages. Also, there are plenty of documents available on the internet in different formats, such as Linux tutorials, Linux documentation project, Server fault, and more.

CLOUD COMPUTING

Cloud computing is a technology that allows individuals and organizations to access and use computing resources (such as servers, storage, databases, networking, software, and more) over the internet on a pay-as-you-go basis. Instead of owning and maintaining physical hardware and data centers, users can leverage the cloud to quickly provision and scale resources as needed.

There are several key characteristics of cloud computing:

1. On-Demand Self-Service: Users can provision and manage computing resources as needed without requiring human intervention from service providers.
2. Broad Network Access: Cloud services are accessible over the internet from a variety of devices, such as laptops, smartphones, and tablets.
3. Resource Pooling: Cloud providers use multi-tenant models, where resources are shared among multiple customers. This allows for cost efficiencies and resource optimization.
4. Rapid Elasticity: Cloud resources can be quickly scaled up or down to meet changing workloads and demands. This scalability ensures that you only pay for the resources you use.
5. Measured Service: Cloud usage is metered and billed based on actual consumption. This pay-as-you-go model provides cost control and flexibility.

Cloud computing can be categorized into three main service models:

1. Infrastructure as a Service (IaaS): IaaS provides virtualized computing resources over the internet. Users can rent virtual machines, storage, and networking components, which they can configure and manage. Examples of IaaS providers include Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).
2. Platform as a Service (PaaS): PaaS offers a platform and environment for developers to build, deploy, and manage applications without the need to worry about underlying infrastructure. PaaS providers include Heroku, Google App Engine, and Microsoft Azure App Service.

3. Software as a Service (SaaS): SaaS delivers software applications over the internet on a subscription basis. Users can access these applications via web browsers without the need to install or maintain software locally. Examples of SaaS offerings include Google Workspace, Microsoft 365, and Salesforce.

Cloud computing is a fundamental technology that has transformed the way businesses and individuals manage and utilize IT resources. It provides cost-efficiency, scalability, flexibility, and accessibility, making it a crucial enabler of digital transformation and innovation.

AWS – AMAZON WEB SERVICES

Amazon Web Services (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs (Application Programming Interfaces) to individuals, companies, and governments, on a metered pay-as-you-go basis. AWS offers a wide range of cloud services, including computing power, storage, databases, machine learning, analytics, content delivery, and more. These services enable organizations to scale and deploy applications and infrastructure without the need to invest in and maintain their own physical servers and data centers.

AWS is one of the leading cloud service providers globally and has a vast global network of data centers, allowing customers to host their applications and data in various regions around the world. This global infrastructure, along with a wide array of services and tools, has made AWS a popular choice for businesses looking to leverage the cloud for their computing and storage needs.

Some of the most well-known AWS services include Amazon EC2 (Elastic Compute Cloud) for scalable virtual servers, Amazon S3 (Simple Storage Service) for object storage, Amazon RDS (Relational Database Service) for managed databases, Amazon Lambda for serverless computing, and Amazon SNS (Simple Notification Service) for messaging and notifications, among many others.

AWS provides both infrastructure as a service (IaaS) and platform as a service (PaaS) solutions, allowing users to choose the level of control and flexibility they require for their specific applications and workloads. It has become a cornerstone of the cloud computing industry and is widely used by startups, enterprises, and governments for a variety of purposes, including hosting websites, running machine learning models, storing and analyzing data, and more.

AWS Account Creation (Free tier account)

Creating an AWS Free Tier account is a straightforward process, and it allows you to access a range of AWS services with certain usage limits for free during the first 12 months of your account.

Here are the steps to create an AWS Free Tier account:

1. Visit the AWS Free Tier Signup Page: Go to the AWS Free Tier signup page by following this link: https://aws.amazon.com/free/.

2. Click "Create an AWS Account": On the AWS Free Tier page, click the "Create an AWS Account" button.

3. Sign In or Create a New AWS Account:

   If you already have an Amazon.com account, you can sign in with those credentials. If you don't have an Amazon.com account, you'll need to create one. Follow the on-screen instructions to sign in or create an account.

4. Provide Account Information:

   Fill in the required information for your AWS account, including your email address, password, and AWS account name. You'll also need to read and accept the AWS Customer Agreement, the AWS Service Terms, and the AWS Privacy Notice.

5. Contact Information: Enter your contact information, including your name, address, and phone number.

6. Payment Information: To verify your identity, you'll need to provide valid credit card information. AWS uses this information to ensure that you are a real person and to prevent misuse of the Free Tier. Your credit card will not be charged unless you exceed the Free Tier usage limits or choose to use paid services.

7. Identity Verification: You'll need to provide a phone number where you can receive a call from AWS. Follow the instructions to receive a verification code via phone call/sms. Enter the code to complete the identity verification process.

8. Choose a Support Plan: AWS offers a free Basic Support plan, for the Free Tier, the Basic Support plan is sufficient.

9. Complete the Signup: Review the information you've provided and click "Create Account and Continue" to complete the signup process.

10. Welcome to AWS: Your AWS Free Tier account is now created. You can access the AWS Management Console and start using the free services within the Free Tier limits.

Remember to monitor your usage and stay within the Free Tier limits to avoid unexpected charges. AWS Free Tier offers a variety of services at no cost for the first 12 months, but some services may have usage limits. Always check the AWS Free Tier documentation for the most up-to-date information on available services and usage limits.

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service offered by Amazon Web Services (AWS) that provides resizable compute capacity in the cloud. With Amazon EC2, you can launch virtual servers known as "instances" to run applications, host websites, and handle various workloads. Here are some key points and features related to EC2 instances:

1. **Instance Types:** EC2 instances come in various types, each optimized for specific use cases. These types range from general-purpose instances to compute-optimized, memory-optimized, storage-optimized, and GPU-accelerated instances. You can choose the instance type that best suits your workload's requirements.

2. **Operating Systems:** EC2 supports a wide range of operating systems, including Linux, Windows, and various BSD variants. You can select the operating system that suits your application needs.

3. **Instance Lifecycle:** EC2 instances can be launched, terminated, and stopped as needed. You are only billed for the compute capacity you use while the instance is running.

4. **Amazon Machine Images (AMIs):** AMIs are pre-configured templates that define the software, settings, and operating system for your EC2 instances. You can choose from a variety of AMIs provided by AWS or create custom AMIs.

5. **Key pair:** Key pairs are an essential security feature in AWS that helps protect your instances and data from unauthorized access. Always make sure to generate and manage your key pairs carefully and follow security best practices to ensure the safety of your AWS resources

6. **Security:** EC2 instances can be launched within Virtual Private Clouds (VPCs), and you can apply security groups and network access control lists to control inbound and outbound traffic to your instances. Additionally, you can use Identity and Access Management (IAM) for fine-grained access control.

7. **Storage Options:** EC2 instances can be attached to various types of storage, including Amazon Elastic Block Store (EBS) for block storage and instance store (also known as ephemeral storage) for temporary storage. EBS volumes are network-attached, allowing data to persist even if the instance is terminated.

8. **Instance Metadata:** EC2 instances have access to instance metadata, which includes information about the instance, such as instance type, IP address, and security group settings. This metadata can be used by applications running on the instance.

EC2 instances are a fundamental building block of AWS infrastructure, providing the flexibility and scalability needed to deploy a wide range of applications and workloads in the cloud. The choice of instance type, storage, and configuration depends on your specific use case and performance requirements.

Launching EC2 instances on Amazon Web Services (AWS) can be done through the AWS Management Console, AWS Command Line Interface (CLI), or AWS SDKs. Below, I'll provide step-by-step instructions for launching an EC2 instance using the AWS Management Console:

Step 1: Sign In to Your AWS Account

- If you don't already have an AWS account, you'll need to create one. Once you have an account, sign in to the AWS Management Console at https://aws.amazon.com/console/.

Step 2: Navigate to the EC2 Dashboard

- In the AWS Management Console, locate and click on the "Services" dropdown in the top left corner, then select "EC2" under the "Compute" section. This will take you to the EC2 Dashboard.

Step 3: Choose an Amazon Machine Image (AMI)

- On the EC2 Dashboard, click the "Launch Instance" button to start the instance creation process.
- Give the name of your server
- In the "Choose an Amazon Machine Image (AMI)" step:
  - Select an AMI from the provided list. You can choose from AWS's pre-configured AMIs .Click on AMAZON LINUX AMI

Step 4: Create or Select a Key Pair

- You'll be prompted to create a new key pair or use an existing one. Click on create new key pair, give key pair name and create key pair. After creating a new key pair, make sure to download the private key file and keep it secure folder for further use.

Step 5: Choose an Instance Type

- In the "Choose an Instance Type" step:Select the instance type that meets your computing needs. EC2 offers a variety of instance types optimized for different workloads.Click on t2.micro which is free tier eligible

Step 6: Configure Instance Details

- In the "Configure Instance Details" step, you can:
  - Choose the number of instances to launch.

Step 7: Configure Security Group

- In the "Configure Security Group" step, you can:
  - Choose an existing security group or create a new one.
  - Configure inbound and outbound rules to control network traffic to your instance.
  - Click on edit option on network settings and add security rules
  - Allow all traffic and allow anyip4 address

Step 8: Review and Launch

- In the "Review" step, review the configuration settings for your instance. Make sure everything is set up correctly.

Step 9: Launch the Instance

- After reviewing the instance details, click the "Launch" button.

Step 10: View Your Instances

- You'll see a confirmation message indicating that your instances are launching. You can track the status of your instances by going back to the EC2 Dashboard and clicking on "Instances" in the left-hand navigation pane.

That's it! You've successfully launched EC2 instances on AWS. You can now connect to your instances using SSH (for Linux instances) and start using them for your specific workloads. Make sure to manage your instances, monitor their performance, and follow best security practices for AWS resources.

Connecting ec2 instances to Git bash

To connect to an Amazon EC2 instance using Git Bash (a Bash shell for Windows), you can use SSH (Secure Shell) if you're connecting to a Linux-based EC2 instance. Here are the steps to connect your EC2 instance to Git Bash:

Prerequisites:

1. Make sure you have already launched an EC2 instance in your AWS account.
2. You should have downloaded the private key file (e.g., **your-key.pem**) when launching the instance.

Step 1: Locate Your Private Key

- Ensure that the private key file (**your-key.pem**) is stored in a directory on your local machine. For security reasons, it's a good practice to keep your private key in a secure location.

Step 2: Open Git Bash

- Create a newfolder on desktop and copy paste the keypair.pem file in this folder
- Now right click the mouse and open gitbash here

Step 3:Login to console and launch instances

- ➢ click on created instances and connect

Step 4: Set Appropriate Permissions for Your Private Key

- SSH requires that the private key file is not publicly accessible, so you should change the permissions on the key file to make it more secure.Use the following command:

```
chmod 400 your-key.pem
```

And copy in git bash terminal

Step 5: Connect to Your EC2 Instance

- Use the **ssh** command to connect to your EC2 instance.

  ssh -i your-key.pem ec2-user@YourInstancePublicIP (ssh -i "awskeypair.pem" ec2-user@ec2-3-87-128-182.compute-1.amazonaws.com)

- If you're connecting to an Ubuntu instance, replace **ec2-user** with **ubuntu**.

Step 6: Confirm the Connection

- When prompted, confirm the SSH connection by typing "yes" and pressing Enter.
- You should now be connected to your EC2 instance via SSH in Git Bash.

You can now use Git Bash to interact with your EC2 instance, execute commands, and manage your files on the server. Make sure to disconnect from the instance when you're done by typing **exit** and pressing Enter.Stop/terminate the instances after practice.

Remember to keep your private key secure and never share it with others. Additionally, ensure that the security group associated with your EC2 instance allows SSH traffic (port 22) from your local IP address.

LINUX COMMANDS

File commands	
sudo su -	Switching ec2 user to root user
touch filename	Create the files
mkdir dirname	Create the directories
ls	Directory / files listing
ll	Long listing the files/directories
ls -la	Formatted listing with hidden files
ls -lt	Sorting the Formatted listing by time modification
ls -lrth	Sorting the Formatted listing by time modification in reverse order
cd	Changing directory to home directory
cd /	Changing directory to root
cd /opt	Navigating into Optional directory to create the files/directories
cd dirname	Change directory to dirname
cd ..	Coming out of current directory
cd ../../	Coming out of two directories
cd -	To go to the previous directory
cd ~	Changing directory directly to user home directory
pwd	Show current working directory
mkdir dir1 dir2 dir3	Creating a multiple directories
mkdir –p d1/d2/d3/d4	Creating nested directories
cat filename	View data in the file
more filename	Output the contents of the file
head filename	Output the first 10 lines of the file
tail filename	Output the last 10 lines of the file
tail -n filename	Output display only last 10 lines
touch file1 file 2 file3	Creating multiple files
rm file	Deleting the file
rm -r dir	Deleting the directory
rm -f file	Force to remove the file
rm -rf dir	Force to remove the directory dir
cp file1 file2	Copy the contents of file1 to file2
cp -r dir1 dir2	Copy dir1 to dir2;create dir2 if not present
mv file1 file2	Rename or move file1 to file2,if file2 is an existingdirectory

yum install tree	installing tree package
tree	To display the files and directories in tree structure
echo data > fname	This command writes the text "data" to the file fname.
Echo data >> fname	This command appends the text "data" to the file fname

System info commands	
date	Show the current date and time
cal	Show this month's calendar
uptime	Show current uptime
w	Display who is on line
whoami	Who you are logged in as
Sleep <time>	To hold the terminal by the specified amount of time
uname -a	Show kernel information
cat /proc/cpuinfo	Cpu information
cat proc/meminfo	Memory information
man command	Show the manual data for command
df -h df -Th	Show the disk usage Show the disk usage and limit listing to file systems of type TYPE
du du -sh du -h	Show directory/file space usage shows folder size and print sizes in human readable format shows file size and print sizes in human readable format
free	Show memory and swap usage
who –r or runlevel : In Linux, run levels represent different operating states or modes of the system. Historically, run levels were used to define the state of the system during system initialization or	displays run level init 0/1/2/3/4/5/6 0 = stop/hang 1 = singleuser 2 = multiuser

shutdown	3 = multiuser + n/w 4 = reserved 5 = cli +gui 6 = reboot
whereis appname	Show possible locations of app
which appname	Show which applications will be run by default
yum install packagename	To install package
yum remove packagename	To remove installed pacakage
hostname	Displays the server name
hostname -I	Displays IP-address of server
hostnamectl set-hostname <name>	To set hostname
cat /etc/passwd	To view the user configuration

User commands	
sudo su -	switch to root user
su - username	Switching user to another user
id	To display the user ID
useradd username	To add a user on a Linux server
passwd password	To create the password for a user
userdel –r username	To delete a user
usermod –c “comment” user	Adding comment to the user
User stanza username:userid:usergroup:comment:location:excitation ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash	

process management commands	
kill process id	kill the process for the given process id
kill -9 process id	Kill the process forcibly for the given process id
lscpu	Displays information about cpu
ps	To display the currently working processes
top	Displays all running process

Network commands	
ping google.com	To check connectivity with two nodes
whois domain	Get whois information for domains
dig domain	Get DNS information for domain
wget url	Download file
netstat	Display connection information

Compression commands	
zip –r data.zip filename	Create zip file named data.zip containing filename
unzip data.zip	Extracting .zip files from zip archive
tar -xvf file.tar	Extract the files from file.tar
tar cvf file.tar.gz filename	Change normal file to tar file

File permission commands

The chmod command in Linux is used to change the file permissions

chmod [options] [File_name]	Change the permission of file to octal,which can be found separately for user,group,world by adding, • 4-read(r) • 2-write(w) • 1-execute(x)
Change file permission The numeric mode is a combination of these values. For example: 7 – read, write, and execute (4 + 2 + 1) 6 – read and write (4 + 2) 5 – read and execute (4 + 1) 4 – read only	chmod <permission> <file name> u (Owner) - Permissions used for the owner of the file. g (Group) - Permissions used by members of the group. o (Other) - Permissions used by all other users. r (read) – permit to read the file. w (write) – permit to write the file. x (execute) – permit to execute the file.

CHOWN :FILE & DIRECTORY OWNERSHIP

The chown command in Linux is used to change the ownership of files and directories. This command allows you to change both the user and group ownership of a file or directory.

chown [options] user[:group] file(s)	1. Change the Owner of a File: chown newuser filename 2. Change the Group of a File: chown :newgroup filename 3.Change Both the Owner and Group of a File: chown newuser:newgroup filename 4. Change the Owner and Group of Multiple Files:

	chown newuser:newgroup file1 file2 file3 5.-R – Change ownership recursively for directories and their contents: chown -R newuser:newgroup /path/to/directory

Searching commands	
grep pattern file	Search for pattern in file
grep -r pattern dir	Search recursively for pattern in dir
yum install locate	install locate package
updatedb	update db
locate filename	Find all instances of filename
locate –n 5 filename	locates top 5 locations of given filename
ps -ef | grep tomcat	Searches for the process id of tomcat

Filter commands	
cut -d(delimiter) -f(columnNumber) <fileName>	To select a specific column of a file. The '-d' option is used as a delimiter, and it can be a space (' '), a slash (/), a hyphen (-), or anything else. And, the '-f' option is used to specify a column number.
comm< fname1> <fname2>	To compare two files or streams
tree	To view files/directories in tree format
wc <fname> wc –l fname wc –w fname wc –c fname	To count the lines, words, and characters in a file line count word count character count
sort <fname>	To sort files in alphabetical order

sed command	
Syntax: s/pattern/replacement/flags sed 's/madhu/madhukiran/' test > test2	Replaces occurrences of a pattern with a specified replacement. And stores in test2 file
sed 's/madhu/madhukiran/' input.txt	Replace first occurrence of 'madhu' with 'madhukiran' in each line
sed 's/madhu/madhukiran/g' input.txt	Replace all occurrences of 'madhu' with 'madhukiran' in each line
sed 's/madhu/madhukiran/2' input.txt	Replace the second occurrence of "madhu' with 'madhukiran' in each line

Curl command	
curl -O https://example.com/file.zip	Download a file
curl -X POST -d "username=user&password=pass" https://api.example.com/login	Send a POST request with data
curl -I www.google.com	The curl -I command is used to fetch only the headers of a URL
curl -m 10 https://example.com	The -m option in the curl command sets a maximum time for the entire operation

Monitoring & performance commands	
top	Displays real-time information about system processes, CPU and memory usage.
htop	An interactive and improved version of `top`, providing a more user-friendly and colorful interface.
iostat	Reports CPU and input/output statistics for devices and partitions.
lscpu	Displays detailed information about the CPU architecture.
free -g	Displays the total, used, and free memory in gigabytes.
vmstat	Reports virtual memory statistics.
sar	Collects, reports, or saves system activity information.
mpstat	Reports CPU usage per processor.

Shortcuts	
ctrl+c	Halts the current command
ctrl+z	Stops the current command, resume with fg in the foreground or bg in the background
ctrl+d	Logout the current session, similar to exit
ctrl+w	Erases one word in the current line
ctrl+u	Erases the whole line
ctrl+r	Type to bring up a recent command
ctrl+l	clear the entire screen
!!	Repeats the last command
exit	Logout the current session

Files system Hierarchy Standard (FHS)

The Files system Hierarchy Standard (FHS) is a set of guidelines that define the structure and organization of directories and files on Unix-like operating systems, including Linux. It aims to ensure consistency and compatibility between different distributions of Linux, making it easier for software developers, administrators, and users to navigate and understand the file system. Here are the key components of the FHS:

Root Directory (/)

/bin: Essential command binaries that are required for system operation (e.g., ls, cp, mv).

/boot: Files needed for the boot process, including boot loader configuration files and kernels.

/dev: Device files representing hardware devices attached to the system (e.g., /dev/sda for the first SATA drive).

/etc: Configuration files for the system and installed applications (e.g., /etc/passwd for user information).

/home: Home directories for regular users (e.g., /home/user1).

/lib: Shared libraries needed by programs in /bin and /sbin.

/media: Mount points for removable media (e.g., USB drives).

/mnt: Mount points for temporary mounted filesystems.

/opt: Optional software packages that are not part of the operating system distribution.

/sbin: System binaries essential for system administration, typically used by the root user.

/srv: Data for services provided by the system (e.g., web server data).

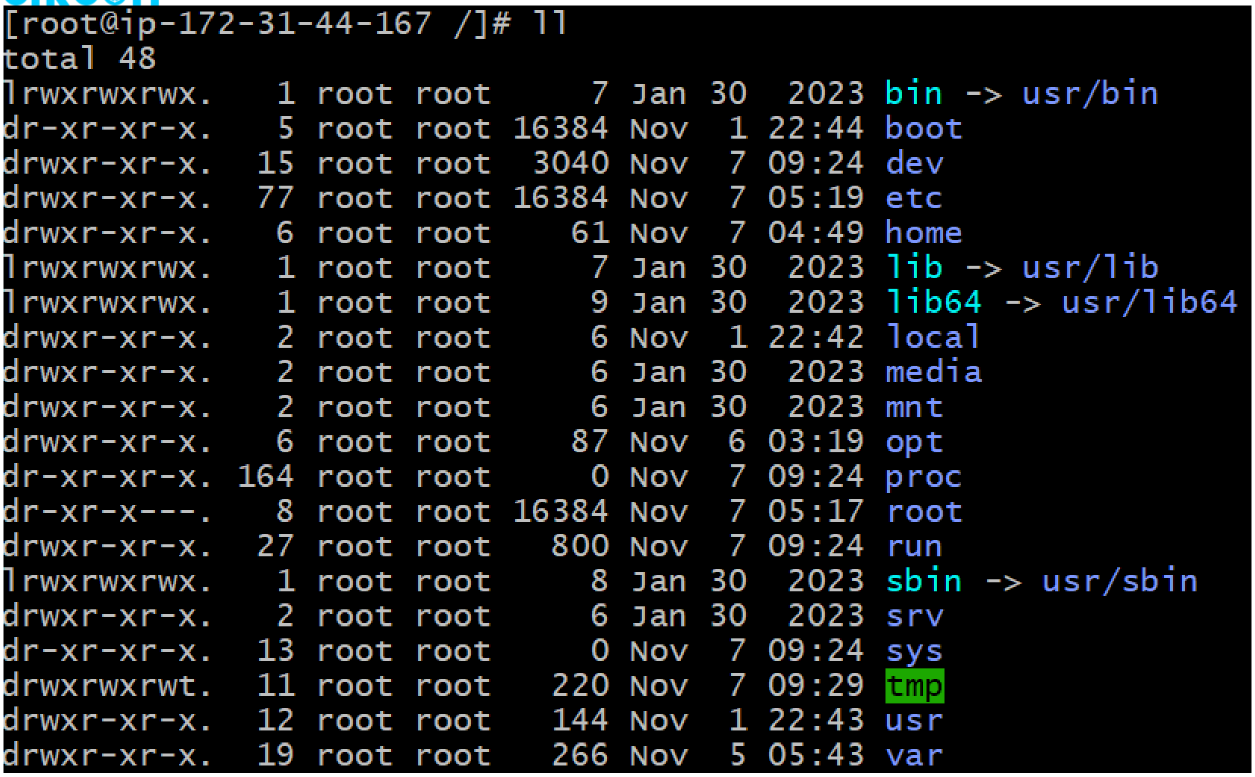
/tmp: Temporary files that may be deleted between reboots.

/usr: Secondary hierarchy for read-only user data; contains the majority of user utilities and applications.

/var: Variable data files, such as logs, spool files, and temporary files that persist between reboots.

cd /

root --> /



```
[root@ip-172-31-44-167 /]# ll
total 48
lrwxrwxrwx.   1 root root     7 Jan 30  2023 bin -> usr/bin
dr-xr-xr-x.   5 root root 16384 Nov  1 22:44 boot
drwxr-xr-x.  15 root root  3040 Nov  7 09:24 dev
drwxr-xr-x.  77 root root 16384 Nov  7 05:19 etc
drwxr-xr-x.   6 root root    61 Nov  7 04:49 home
lrwxrwxrwx.   1 root root     7 Jan 30  2023 lib -> usr/lib
lrwxrwxrwx.   1 root root     9 Jan 30  2023 lib64 -> usr/lib64
drwxr-xr-x.   2 root root     6 Nov  1 22:42 local
drwxr-xr-x.   2 root root     6 Jan 30  2023 media
drwxr-xr-x.   2 root root     6 Jan 30  2023 mnt
drwxr-xr-x.   6 root root    87 Nov  6 03:19 opt
dr-xr-xr-x. 164 root root     0 Nov  7 09:24 proc
dr-xr-x---.   8 root root 16384 Nov  7 05:17 root
drwxr-xr-x.  27 root root   800 Nov  7 09:24 run
lrwxrwxrwx.   1 root root     8 Jan 30  2023 sbin -> usr/sbin
drwxr-xr-x.   2 root root     6 Jan 30  2023 srv
dr-xr-xr-x.  13 root root     0 Nov  7 09:24 sys
drwxrwxrwt.  11 root root   220 Nov  7 09:29 tmp
drwxr-xr-x.  12 root root   144 Nov  1 22:43 usr
drwxr-xr-x.  19 root root   266 Nov  5 05:43 var
```

inode:

In Linux and Unix-like operating systems, an inode (index node) is a data structure that stores metadata about a file or directory. Every file and directory on the filesystem is associated with an inode, which contains important information about the file's attributes and location on disk.

File Metadata:

File type: -- regular file, directory, symbolic link, , etc.
Permissions: (read, write, execute) for the owner, group, and others.
Owner and Group: User ID (UID) and group ID (GID)
Timestamps:
  Last access time (atime):   Last time the file was accessed.
  Last modification time (mtime): Last time the file's content was modified.
  Last status change time (ctime): Last time the file's metadata (permissions, owner, etc.) was changed.
  File Size: Size of the file in bytes.

Softlink vs hardlink

In Linux, both soft links (symbolic links) and hard links are ways to reference files. They serve different purposes and have distinct characteristics.

Examples:

```
ln -s /path/to/target/file   /path/to/link

ln  /path/to/target/file   /path/to/hardlink
```

Environment Variables:

These variables are dynamic environment variables provide a way to customize the behavior of software and the operating system itself.

User specific environment variables

```
# Display all environment variables
env

# Print string "USER"
echo USER

# Print the value of the USER environment variable
echo $USER

# Print the value of the USER environment variable using printenv
printenv USER

# Set an environment variable VAR and print its value
export VAR=testmadhu
printenv VAR
echo $VAR

# Filter environment variables to show those containing "USER" and "VAR"
env | grep USER
env | grep VAR

# Unset the VAR environment variable
```

```
unset VAR

# Edit .bash_profile to add a new environment variable
vim ~/.bash_profile
# Add the line: export VAR2="Madhu kiran"
# Save and exit the editor

# Source .bash_profile to apply the changes
source ~/.bash_profile

# Print and echo the new environment variable
printenv VAR2
echo $VAR2
-----------------
```

Global/system specific environment variables

```
# Change directory to /etc
cd /etc/

# List files in /etc
ls

# List files in /etc with detailed information
ll

# Open the environment file with vim
vim environment
# Add the line: export GLOBALVAR="kiran madhu"
# Save and exit the editor

# Source the environment file to apply changes
source /etc/environment

# Print and echo the new environment variable
echo $GLOBALVAR
printenv GLOBALVAR
```

PATH

The PATH variable is a specific type of environment variable that determines the directories where executable files are searched for when you type a command in the shell without specifying its full path.

If PATH is set to /usr/local/bin:/usr/bin:/bin, and you type ls, the system will look for ls in /usr/local/bin, then /usr/bin, and finally /bin.

PATH=/home/ec2-user/.local/bin:/home/ec2-user/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

===

~ vs # vs $

~ ----> represents home dir
# ----> a root user prompt (super user privileges')
$ ----> a regular user prompt.

==
Key Points:

.bash_profile vs .bashrc

Environment variables: Typically set in .bash_profile.
Aliases and functions: Can be placed in either file, but .bashrc is more common for frequently used ones.
Shell prompt customization: Usually done in .bashrc.
Efficiency: Avoid heavy operations in .bashrc as its executed more often.

VI EDITOR

What is Vi?

The vi editor is elaborated as visual editor. It is installed in every Unix system. In other words, it is available in all Linux distros. It is user-friendly and works same on different distros and platforms. It is a very powerful application. An improved version of vi editor is vim.

The vi editor has two modes:

- Command Mode: In command mode, actions are taken on the file. The vi editor starts in command mode. Here, the typed words will act as commands in vi editor. To pass a command, you need to be in command mode.
- Insert Mode: In insert mode, entered text will be inserted into the file. The Esc key will take you to the command mode from insert mode.

By default, the vi editor starts in command mode. To enter text, you have to be in insert mode, just type 'i' and you'll be in insert mode. Although, after typing i nothing will appear on the screen but you'll be in insert mode. Now you can type anything.

To exit from insert mode press Esc key, you'll be directed to command mode.

Using vi

The vi editor tool is an interactive tool as it displays changes made in the file on the screen while you edit the file.

In vi editor you can insert, edit or remove a word as cursor moves throughout the file.

Commands are specified for each function like to delete it's x or dd.

The vi editor is case-sensitive. For example, p allows you to paste after the current line while P allows you to paste before the current line.

vi syntax: vi <filename>

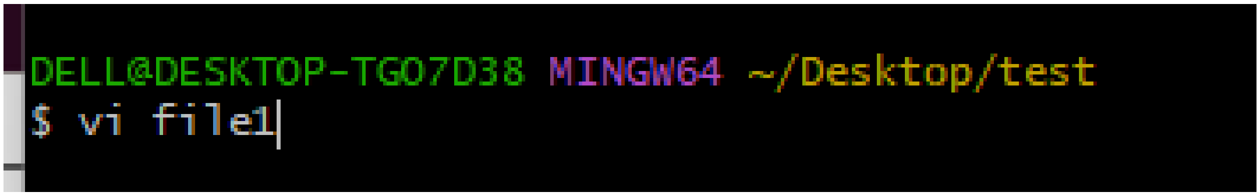
In the terminal when you'll type vi command with a file name, the terminal will get clear and content of the file will be displayed. If there is no such file, then a new file will be created and once completed file will be saved with the mentioned file name.

Let's understand vi through an example:

To start vi open your terminal and type vi command followed by file name. If your file is in some other directory, you can specify the file path. And if in case, your file doesn't exist, it will create a new file with the specified name at the given location.

Example:

1. vi file

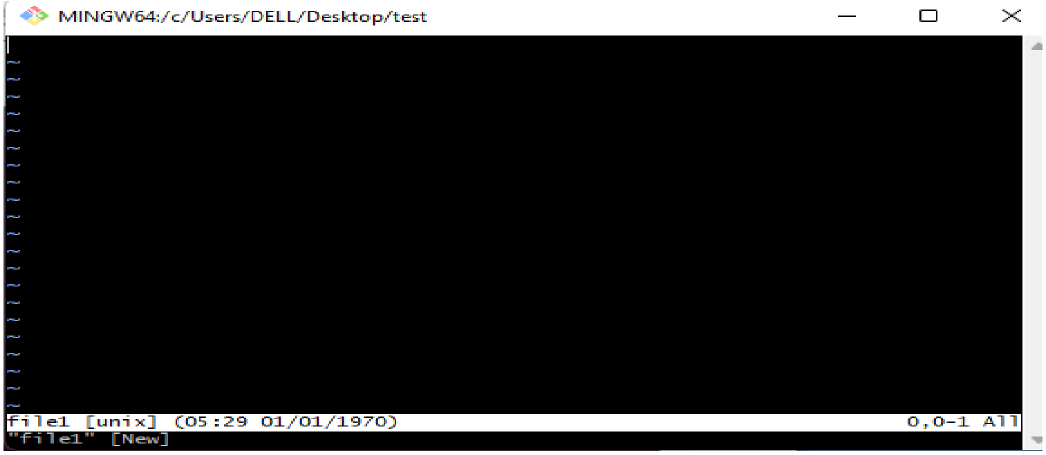


```
DELL@DESKTOP-TGO7D38 MINGW64 ~/Desktop/test
$ vi file1
```

Look at the above snapshot; we are creating a new file file

Command mode

This is what you'll see when you'll press enter after the above command. If you'll start typing, nothing will appear as you are in command mode. By default vi opens in command mode.

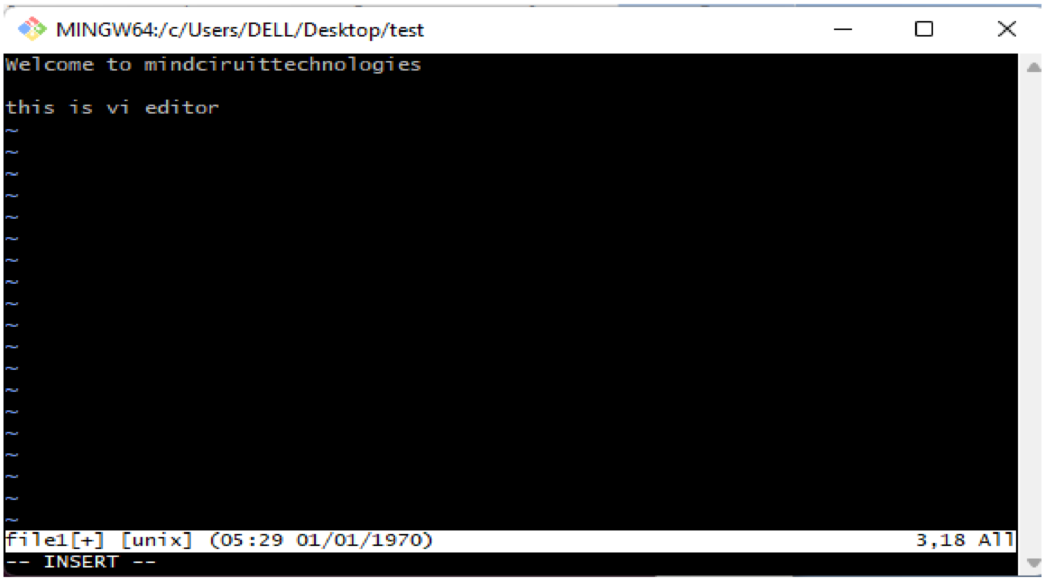


```
MINGW64:/c/Users/DELL/Desktop/test
|
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
file1 [unix] (05:29 01/01/1970)                                  0,0-1 All
"file1" [New]
```

Look at the above snapshot; it is blank as it is a new file. To start typing, you have to move to the insert mode. At the end of the terminal window, directory name and file name are displayed.

Insert mode

To move to the insert mode press **esc i .**



MINGW64:/c/Users/DELL/Desktop/test

```
Welcome to mindciruittechnologies

this is vi editor
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
file1[+] [unix] (05:29 01/01/1970)                                    3,18 All
-- INSERT --
```

Look at the above snapshot, after pressing i we have entered into insert mode. Now we can write anything. To move to the next line press enter.

Once you have done with your typing, press esc key to return to the command mode.

To save and quit

You can save and quit vi editor from command mode. Before writing save or quit command you have to press colon (:wq) .

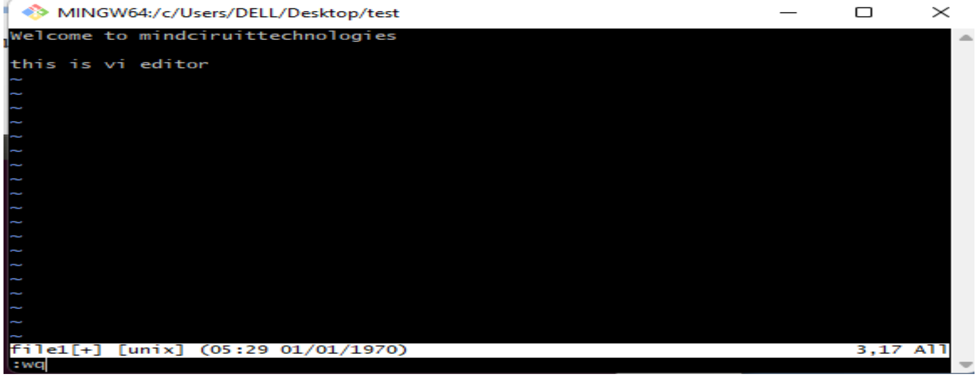
exit vi table commands:

Commands	Action

:wq	Save and quit
:w	Save
:q	Quit
:w fname	Save as fname
:wq!	save and quit forcefully
:q!	Quit discarding changes made
:w!	Save (and write to non-writable file)

To exit from vi, first ensure that you are in command mode. Now, type :wq and press enter. It will save and quit vi.

Type :wq to save and exit the file



```
Welcome to mindciruittechnologies

this is vi editor
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
file1[+] [unix] (05:29 01/01/1970)                                3,17 All
:wq
```

Look at the above snapshot, command : wq will save and quit the vi editor. When you'll type it in command mode, it will automatically come at bottom left corner.

Vi Commands

Linux vi editor is different from other editors. You have to use different keys to use different functions. Although, it's quite easy and interesting to use vi editor.

The vi editor commands are case sensitive.

Have a look at the vi commands in the following table.

To switch from command to insert mode(esc):

Command	**Action**
i	Start typing before the current character
I	Start typing at the start of current line
a	Start typing after the current character
A	Start typing at the end of current line
o	Start typing on a new line after the current line
O	Start typing on a new line before the current line

To move around data in a file:

Commands	**Action**
j	To move down
k	To move up
h	To move left
l	To move right

To jump lines:

Commands	**Action**

G	Will direct you at the last line of the file
gg	Will direct you to your first position in the file

To delete:

Commands	**Action**
x	Delete the current character
X	Delete the character before the cursor
r	Replace the current character
xp	Switch two characters
dd	Delete the current line
D	Delete the current line from current character to the end of the line
dG	delete from the current line to the end of the file

To repeat and undo:

Commands	**Action**
u	Undo the last command
.	Repeat the last command

Command to cut, copy and paste:

Commands	**Action**
dd	Delete a line
yy	(yank yank) copy a line
p	Paste after the current line
P	Paste before the current line

Command to cut , copy and paste in blocks:

Commands	**Action**
<n>dd	Delete the specified n number of lines
<n>yy	Copy the specified n number of lines

Start and end of line:

Commands	**Action**
θ	Bring at the start of the current line
^	Bring at the start of the current line
$	Bring at the end of the current line
dθ	Delete till start of a line

d$	Delete till end of a line

Joining lines:

Commands	Action
J	Join two lines
yyp	Repeat the current line
ddp	Swap two lines

Move forward or backward:

Commands	Action
w	Move one word forward
b	Move one word backward
<n>w	Move specified number of words forward
dw	Delete one word
yw	Copy one word

<n>dw	Delete specified number of words

More commands

:set nu or :set number	to set the numbering in vi editor
:set nonu or :set nonumber	to remove the numbering in vi editor
/ word or character	to search a word or character
:nohl	To remove search highlights in vi editor

awk command	
awk command :	awk is a scripting language used for manipulating data and generating reports
awk '{print}' fname	To display data
awk '{print $1,$4}' fname	Prints 1st and 4th column only
awk '/word/{print}' fname	Prints all the line which matches with the 'word'.
awk '{print NR,$0}' fname	NR prints all the lines along with the line number.
awk '{print NR "- " $1 }' fname	To print the first item along with the row number(NR) separated with " – " from each line

<table>
<tr><td colspan="2"><b>scp Command</b>
The scp (secure copy) command is used to securely transfer files between a local host and a remote host, or between two remote hosts. It uses SSH for data transfer and provides the same level of security and encryption.</td></tr>
<tr><td>scp [options] source destination</td><td>syntax</td></tr>
<tr><td>scp localfile.txt username@remotehost:/path/to/destination/</td><td>Copy a file from a local system to a remote system</td></tr>
<tr><td colspan="2"><b>winscp Command</b>
winscp is a popular SFTP and FTP client for Windows, allowing users to securely transfer files between local and remote systems. It also has a command-line interface (CLI) for scripting purposes.</td></tr>
</table>

Linux interview question

- https://qaremote.jobs/blog/linux-interview-questions/?gclid=Cj0KCQiAuqKqBhDxARIsAFZELmKZ7M7vGMlBWvfViOhD4AWW9E5FHo6hHfpf3mWS23vjGthc4YblXmYaAjAkEALw_wcB

- https://www.geeksforgeeks.org/linux-interview-questions/