DAY-6 (29/5/2025)

Emp ID : 110714877

NAME :Kinnera venkatamma

Task1 . RegEX Symbols in linux  & List them down with description

In Linux, regular expressions (often called "regex" or "regexp") are special character sequences used to define search patterns. They're powerful tools for finding, replacing, and validating text within files and other data streams. These patterns can be used in various Linux programs, including grep, bash, and rename.

How Regular Expressions Work:

Regular expressions consist of literal characters (like letters, numbers, and symbols) and metacharacters that give them their power. These metacharacters, such as ^, $, \*, +, ?, [], (), and |, allow you to specify complex patterns.

Key Concepts:

Pattern Matching: Regular expressions define rules for matching patterns in text.

Metacharacters: Special characters that have specific meanings in regular expressions.

Character Classes: Sets of characters that match any single character within the set.

Quantifiers: Symbols like \*, +, and ? that specify how many times the preceding character or group should appear.

Anchors: Symbols like ^ and $ that specify the beginning or end of a line or string.

Common Metacharacters and Their Meanings:

. (dot): Matches any character except newline.

^ (caret): Matches the beginning of a line.

$ (dollar sign): Matches the end of a line.

\* (asterisk): Matches zero or more occurrences of the preceding character or group.

+ (plus sign): Matches one or more occurrences of the preceding character or group.

? (question mark): Matches zero or one occurrence of the preceding character or group.

[] (square brackets): Defines a character class.

() (parentheses): Used for grouping parts of the expression.

| (vertical bar): Matches either the preceding or following expression.

`\` (backslash): Used to escape special characters.

Task2. What are the imp features of Linux os ?

Linux's key features include its open-source nature, allowing for free use, modification, and distribution, and its robust security and stability. It also offers powerful multitasking and multiuser capabilities, as well as advanced networking features, making it suitable for various applications, from servers and desktops to embedded systems.

Here's a more detailed breakdown of important features:

1. Open-Source Foundation: Linux is developed by a global community of developers, making it free to use, modify, and distribute. This open-source nature fosters collaboration and innovation, leading to a constantly evolving and improved OS.

2. Multiuser and Multitasking: Linux supports multiple users simultaneously accessing the system and running multiple applications concurrently. This makes it well-suited for servers, where many users may access the system at the same time.

3. Security: Linux offers robust security features, including user permissions, process isolation, and a well-defined permission system. It also uses authentication and authorization mechanisms to control access to system resources.

4. Stability and Reliability: Linux is known for its stability and reliability, often running for extended periods without requiring a reboot. This is crucial for servers and critical systems where uptime is paramount.

5. Advanced Networking: Linux provides extensive networking capabilities, including support for various protocols and network configurations.

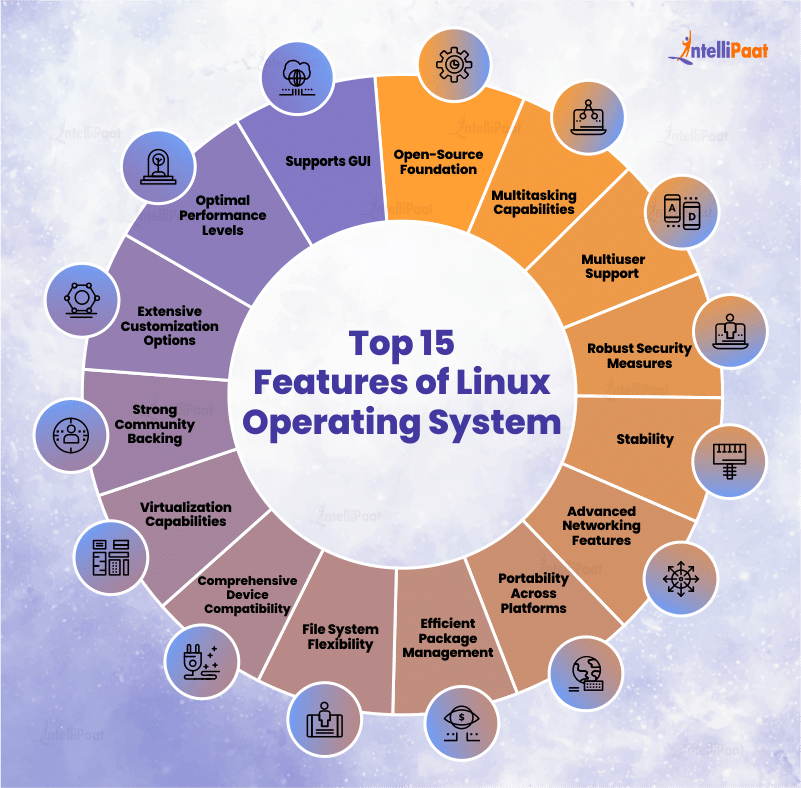
6. Portability: Linux is highly portable, meaning it can run on a wide range of hardware platforms, from embedded systems to supercomputers.

7. Efficient Package Management: Linux offers various package management systems that simplify the installation, updating, and removal of software.

8. Customization and Flexibility: Linux is highly customizable, allowing users to tailor the OS to their specific needs and preferences.

9. Free and Open Source: The Linux kernel and many other components are free and open-source, meaning users can access the source code, modify it, and redistribute it without paying any licensing fees.

10. Versatility: Linux is versatile and can be used for various purposes, including server administration, desktop environments, embedded systems, and more.



Task 3 . What is Kernal and can you explain its functions

A kernel is the core of an operating system (OS), acting as a bridge between software and hardware. It manages system resources, processes, and communication between applications and the hardware. Key functions include process and memory management, device control, and file system management.

Here's a more detailed explanation of the kernel's functions:

1. Process Management:

Scheduling:

The kernel determines which processes (programs) get CPU time and for how long, ensuring fair and efficient execution.

Multitasking:

It manages multiple processes running simultaneously, preventing them from interfering with each other.

Process Creation and Termination:

The kernel handles the creation, deletion, and overall management of processes.

2. Memory Management:

Allocation and Deallocation:

The kernel allocates memory to processes as needed and releases it when processes finish, preventing memory leaks.

Memory Protection:

It ensures that processes have access only to the memory allocated to them, preventing one process from corrupting another's memory.

3. Device Management:

Device Drivers:

The kernel manages communication between the operating system and hardware devices through device drivers.

Input/Output (I/O) Handling:

It handles all input and output requests from software, translating them into instructions for the CPU.

4. File System Management:

File Storage:

The kernel manages how files are stored on storage devices (like hard drives).

File Access:

It provides a mechanism for applications to access files, controlling read and write operations.

5. System Calls:

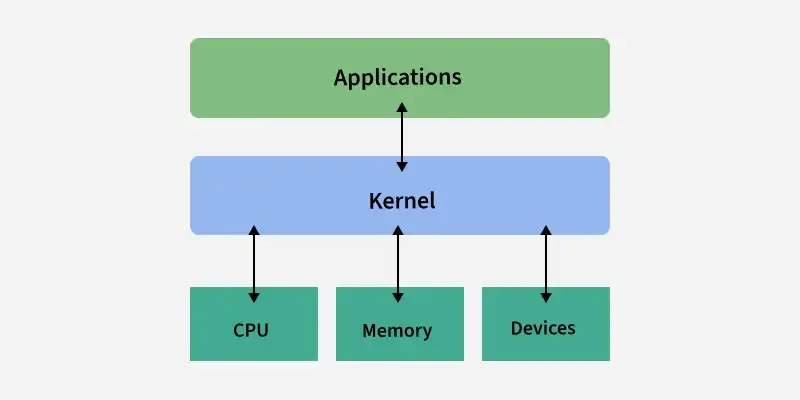
Interface:

The kernel provides a system call interface, allowing user-level applications to request services from the kernel, such as file operations, memory allocation, and device access.

Privileged Access:

System calls allow applications to interact with the kernel's privileged resources.

]In summary, the kernel is the heart of the OS, responsible for managing and controlling the system's resources, allowing applications to run smoothly, and ensuring efficient and secure operation.



Task 4. What is BASH? Full form with explanation.

BASH stands for Bourne Again SHell. It's a command-line shell and scripting language used on Linux, macOS, and other Unix-like operating systems. BASH is the default shell on many Linux distributions, and it allows users to interact with the operating system by typing commands and scripts.

Explanation:

Shell:

A shell is a command-line interpreter, a program that takes commands from the user and executes them on the operating system.

Bourne Shell:

The original Unix shell, written by Stephen Bourne, is the predecessor to BASH.

Bourne Again Shell:

BASH is an enhanced and improved version of the original Bourne shell, incorporating features from other shells like Korn shell and C shell.

Command-Line:

BASH operates in a text-based environment, where users type commands to perform tasks.

Scripting Language:

BASH can also be used to create scripts, which are sequences of commands that can be executed automatically.

Task 5 . What is the difference between window and linux

Linux: Linux could be a free and open supply OS supported operating system standards. It provides programming interface still as programme compatible with operating system primarily based systems and provides giant selection applications. A UNIX operating system additionally contains several severally developed parts, leading to UNIX operating system that is totally compatible and free from proprietary code.

Windows: Windows may be a commissioned OS within which ASCII text file is inaccessible. it’s designed for the people with the angle of getting no programming information and for business and alternative industrial users. it’s terribly straightforward and simple to use. The distinction between Linux and Windows package is that Linux is completely freed from price whereas windows is marketable package and is expensive. Associate operating system could be a program meant to regulate the pc or computer hardware Associate behave as an treater between user and hardware. Linux is a open supply package wherever users will access the ASCII text file and might improve the code victimisation the system. On the opposite hand, in windows, users can’t access ASCII text file, and it’s a authorized OS.

| **S. No** | **Linux** | **Windows** |
| --- | --- | --- |
| 1. | Linux is an **open-source** operating system. | Windows is **not** an open-source operating system. |
| 2. | Linux is **free of cost**. | Windows is **paid** and requires a license. |
| 3. | **File names are case-sensitive**, meaning file.txt and File.txt are different. | **File names are case-insensitive**, meaning file.txt and File.txt are treated the same. |
| 4. | Uses a **monolithic kernel**. | Uses a **hybrid kernel**. |
| 5. | **More efficient and stable**, especially for servers and developers. | **Less efficient** due to resource-intensive processes. |
| 6. | Uses **forward slash (/)** for directory separation. | Uses **backslash (\)** for directory separation. |
| 7. | **More secure** with better user control and fewer vulnerabilities. | **Less secure** due to higher susceptibility to malware and viruses. |
| 8. | Preferred by **hackers and security experts** due to its open-source nature and control. | **Not widely used for hacking** as it lacks built-in security tools. |
| 9. | Has **3 types of user accounts**: (1) Regular, (2) Root, (3) Service Account. | Has **4 types of user accounts**: (1) Administrator, (2) Standard, (3) Child, (4) Guest. |
| 10. | **Root user** has all administrative privileges. | **Administrator user** has all administrative privileges. |
| 11. | In Linux, you **can have two files with the same name** but different cases (File.txt and file.txt). | In Windows, **you cannot have two files with the same name** in the same folder. |

Task 6. Define the basic components of Linux

The core components of a Linux operating system include the kernel, system libraries, and system utilities. The kernel manages hardware resources and interacts directly with the system. System libraries provide essential functionality, enabling applications to access kernel features without direct kernel interaction. System utilities handle specialized tasks like file manipulation and system administration.

Here's a more detailed breakdown:

Kernel:

This is the core of the Linux operating system, directly interacting with the hardware. It manages resources, processes, and memory, ensuring efficient operation of the system.

System Libraries:

These libraries provide a set of functions that applications can use to perform various tasks without directly interacting with the kernel. For example, they handle input/output operations, file access, and networking.

System Utilities:

These are programs that perform specific tasks at the system level, such as managing files, users, and system resources. They are also used for administrative tasks like installing software or configuring the system.

Task 7. Is it legal to edit Kernal?

Yes, it is legal to edit a kernel, particularly the Linux kernel, as it is released under the General Public License (GPL). The GPL allows for modification and redistribution of the software, meaning anyone can make changes and distribute their modified version, provided they also license it under the GPL.

Elaboration:

GPL and Open Source:

The Linux kernel is open-source, meaning its code is publicly available and anyone can examine, modify, and distribute it, according to barrgroup.com. The GPL ensures that any derivative works also remain open-source and under the same licensing terms.

Modifying the Kernel:

You can make changes to the kernel code for your own personal use or for contribution to the community.

Distributing Modified Kernels:

If you intend to distribute your modified version, you must also license it under the GPL, ensuring that the downstream recipients also have the freedom to modify and redistribute.

Example: Google's Android Modifications:

Google, for instance, modifies the Linux kernel to create Android, and their modifications are GPL compliant. This allows them to add proprietary blobs (closed-source code) for device-specific functionalities, while also contributing to the mainline kernel.

Task 8. Can you explain LILO

LILO (Linux Loader) is an old, yet reliable, boot loader used in Linux-based systems. It was a popular choice in the early days of Linux, allowing users to boot Linux alongside other operating systems. While LILO's development is discontinued, it's a simple and direct way to load the operating system kernel.

Here's a more detailed explanation:

What it is:

LILO is a program that loads the operating system kernel into memory when a computer is turned on or restarted.

How it works:

When the computer boots, LILO is loaded into memory by the BIOS (Basic Input/Output System). It then displays a menu of boot options, allowing the user to select which kernel to load.

Configuration:

LILO uses a configuration file (typically /etc/lilo.conf) to define where the bootloader should be placed, what boot options to offer, and kernel parameters.

Advantages:

LILO is known for its simplicity, stability, and ability to boot older systems. It also allows for customization through the configuration file.

Disadvantages:

LILO lacks some advanced features found in modern boot loaders like GRUB, such as the ability to edit boot parameters at boot time. Additionally, if LILO's configuration is corrupted, recovering from boot issues can be more challenging than with GRUB.

Task 9. What is shell? How many shells are there and what are they ? can you explain.

In the context of computers, a "shell" is a program that acts as an interface between the user and the operating system's kernel. It allows users to interact with the computer by typing commands and executing them. There are many different types of shells, but they generally fall into two main categories: command-line shells and graphical shells.

Types of Shells:

Command-line shells (CLI):

These shells provide a text-based interface for interacting with the operating system. Users type commands, and the shell interprets and executes them. Examples include Bash (Bourne Again SHell), Zsh, and others.

Graphical shells (GUI):

These shells provide a graphical user interface, allowing users to interact with the system through icons, menus, and windows. Examples include GNOME, KDE, and Windows Explorer.

How many shells are there?

There are numerous shells available for different operating systems, each with its own features and capabilities. Some popular examples include:

Linux/Unix: Bash, Zsh, Fish, Korn Shell (ksh), C Shell (csh), Bourne Shell (sh).

Windows: PowerShell.

Explanation:

Function:

A shell takes the user's input (commands), translates it, and passes it to the operating system's kernel for execution.

Purpose:

Shells provide a way for users to manage files, run programs, automate tasks through scripting, and interact with the operating system at a low level.

Command-line shells:

These shells are often used for tasks like system administration, scripting, and automation, as they offer more control and flexibility than GUI shells.

Graphical shells:

These shells are more user-friendly and are commonly used for everyday tasks like browsing files, running applications, and managing the desktop environment.

Task 10. What is Swap space ?

Swap space, also known as virtual memory, is a storage area on a computer's hard drive or SSD that acts as an extension of its physical RAM (Random Access Memory). It's used when the RAM is full, temporarily storing inactive data that is not immediately needed, preventing system crashes due to memory shortages.

Here's a more detailed explanation:

Purpose:

When RAM is filled, the operating system can move inactive data (processes or files) to swap space on the hard drive or SSD. This frees up RAM for currently running applications, preventing the system from crashing or becoming unresponsive.

Location:

Swap space is typically located on a dedicated partition of the hard drive or SSD. It can also be implemented as a swap file, a special file within the file system.

Access Speed:

Since swap space is located on the hard drive or SSD, access to it is slower than accessing RAM. This means that while swap space helps to prevent memory shortages, it can also introduce performance delays when data is swapped between RAM and swap space.

Benefits:

Prevents system crashes: By providing extra storage, swap space prevents the system from crashing due to insufficient RAM.

Enables running more programs: It allows a computer to run more programs than would be possible with just the available RAM.

Limitations:

Slower performance: Accessing data from swap space is slower than accessing it from RAM, potentially causing delays in application responsiveness.

Not a replacement for RAM: While useful, swap space should not be considered a substitute for having sufficient RAM.

Task 11. What is Mount ? how do you mount and unmount file system in Linux?

In Linux, "mount" refers to attaching a filesystem (like a USB drive or a hard disk partition) to a directory in the file system, making it accessible to the operating system and users. Unmounting is the opposite, removing the filesystem from that directory, making it inaccessible.

Here's how to mount and unmount a filesystem in Linux:

Mounting:

1. Find the device:

Determine the name of the device file you want to mount (e.g., /dev/sdX1 for a USB drive). You can use ls -l /dev/disk/by-label to find devices by their label or ls -l /dev/ to list all devices.

2. Create a mount point:

Choose a directory where you want to mount the filesystem. A common location for temporary mount points is /mnt or /media. You can create a new directory using mkdir if necessary.

3. Use the mount command:

Run the command mount [device] [mount\_point] with appropriate options. For example: mount /dev/sdb1 /mnt/usb to mount the partition /dev/sdb1 to /mnt/usb. You may need sudo (if you're not logged in as root) to have the necessary permissions.

4. Check the mount:

Verify the filesystem is mounted by using the mount command without any arguments. This will list all currently mounted filesystems.

Unmounting:

Unmount the device: Use the umount [mount\_point] command to unmount the filesystem. For example: umount /mnt/usb. Again, you may need sudo.

Important Considerations:

Root privileges: Mounting and unmounting require root privileges or the use of sudo.

Mount points: Choose a directory that's not in use by other processes or files.

Persistence: For permanent mounts, you'll need to add an entry to the /etc/fstab file.

Error handling: If the filesystem is busy (i.e., someone is accessing files on it), unmounting may fail. Try unmounting the filesystem after any open processes are closed.

Task 12. What is chmod command ? how to use it?

The chmod command in Linux is used to change the permissions (read, write, execute) of files and directories. It uses a symbolic or numeric mode to specify the desired changes.

Syntax:

chmod [OPTION]... MODE FILE...

Explanation:

[OPTION] (optional): Flags like -R (recursive, applies to subdirectories), -h (follow symbolic links), etc.

MODE: Specifies the desired permissions. Can be in symbolic or octal format.

FILE...: The files or directories whose permissions are to be changed.

Common Usage:

1. Symbolic Mode:

chmod u+x filename: Makes the owner (u) executable (x).

chmod g-w filename: Removes write (w) permissions from the group (g).

chmod o=r filename: Sets read (r) permissions for others (o).

chmod a+r filename: Adds read permissions for all users (a).

2. Numeric (Octal) Mode:

chmod 755 filename: Owner has read, write, and execute (7), group and others have read and execute (55).

chmod 644 filename: Owner has read and write (6), group and others have read only (44).

Explanation of Symbolic Mode:

u: User (owner)

g: Group

o: Others

a: All (user, group, others)

+: Add permissions

-: Remove permissions

=: Set permissions exactly

Explanation of Octal Mode:

Each digit represents permissions for user, group, and others.

Values:

4: Read

2: Write

1: Execute

Example: 755 (4+2+1 for read, write, execute for user, 4+1 for read, execute for group and others)

Task 13. Can you add a new user account? Crate a new user in different ways and paste ss

To add a new user account, you can use different methods depending on your operating system or application. In Windows, you can add users through the Settings app or using the net user command. In Android, you can add users through the Settings app. For Linux systems, you can use the useradd command or a graphical user manager. For Google Workspace, you can add users through the Admin console. Google Help

Methods to Add New User Accounts:

1. Windows:

Using the Settings app: Open the Settings app, go to Accounts > Other Users, and then select "Add other user".

Using the net user command: Open Command Prompt with administrator privileges and use the command net user username password /add.

Using the GUI: Click Start, go to Settings, navigate to Accounts > Family & other users, and then add a new user.

2. Android:

Open the Settings app, go to System > Multiple users, and then tap "Add user".

3. Linux:

Using the useradd command: Open a terminal and use the command useradd -d /home/newuser -p password newuser (replace newuser and password with your desired username and password).

Using a graphical user manager: Red Hat Enterprise Linux provides a graphical user manager to create users.

4. Google Workspace:

Sign in to your Admin console, go to Menu, then User, and click "Add new user".



Task 14. Can you change the password of a user? How do you do that? Plz share ss

Yes, it is possible to change a user's password, especially if you are an administrator or have the necessary permissions. The specific steps depend on the operating system or system you're using, but generally involve using a management tool or command-line interface.

Here's how to change a user's password in common scenarios:

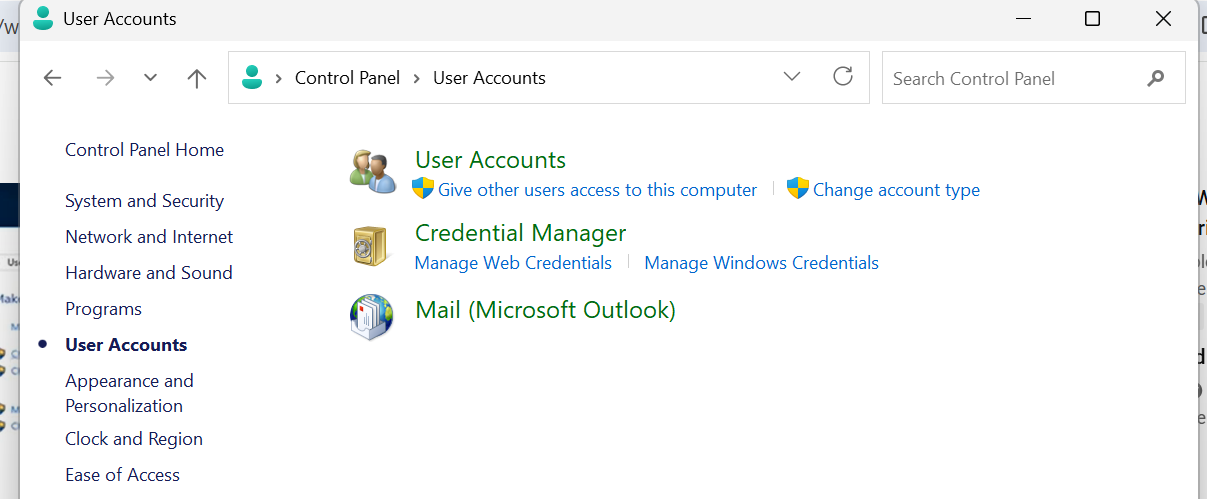
Windows:

Open Control Panel and navigate to User Accounts.

Select "Manage another account".

Choose the user whose password you want to change.

Click "Change the password" and enter the new password.



Task15. What is diff between Process and Thread?

A process is a program in execution, while a thread is a unit of execution within a process. Processes are independent, with their own memory space, while threads within the same process share memory and resources. Threads are lighter than processes, meaning they require fewer resources and have faster context switching.

Here's a more detailed comparison:

Process:

Independent:

Each process runs in its own memory space, isolated from other processes.

Heavyweight:

Processes consume more resources and have slower context switching compared to threads.

Communication:

Processes can communicate with each other, but it requires more overhead and resources, typically through inter-process communication (IPC) mechanisms.

Fault Isolation:

If one process crashes, it generally doesn't affect other processes, thanks to their isolated memory spaces.

Example:

Each application running on your computer (like your web browser, or a word processor) is typically a process.

Thread:

Lightweight:

Threads within a process share the same memory space, code, and resources, making them lighter and more efficient to create and manage.

Shared Resources:

Threads within the same process can directly access and modify shared data, which simplifies communication between them.

Context Switching:

Threads within the same process can switch context quickly, as they don't require interacting with the operating system.

Example:

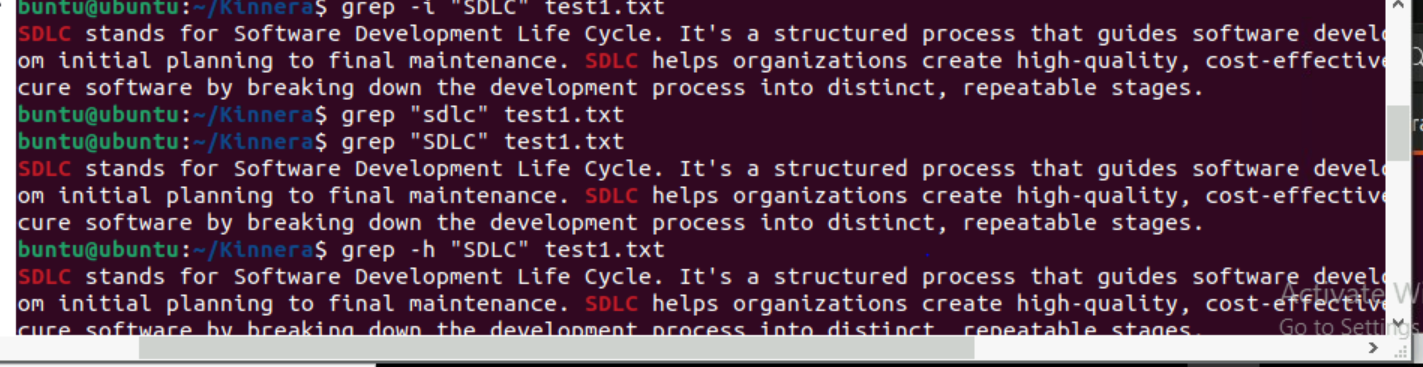
Within a web browser, each tab or window may run in a separate thread.

* Key Differences Summarized:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Process** | **Thread** |
| Memory Space | Independent, separate memory space | Shared memory space with other threads in the same process |
| Resource Usage | Heavyweight, consumes more resources | Lightweight, requires fewer resources |
| Communication | Requires more overhead (IPC) | Direct shared memory access, simpler communication |
| Context Switching | Slower, involves OS interaction | Faster, often no OS interaction needed |
| Fault Isolation | High, one process crash won't affect others | Lower, a thread crash might affect the entire process |
| Example | A running application (e.g., your web browser) | Multiple tabs or windows within a web browser running concurrently |
|  |  |  |

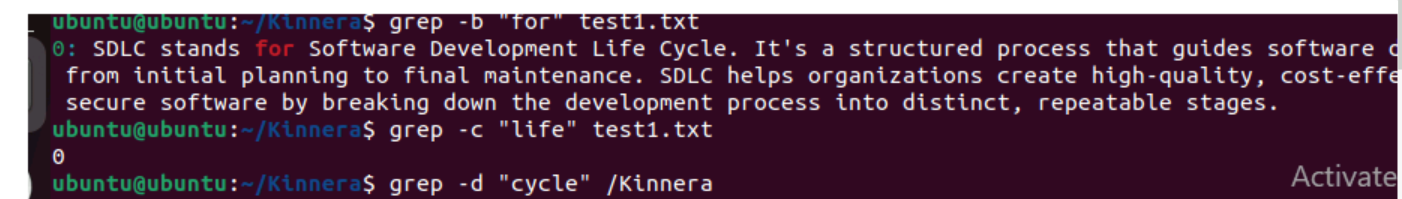
Task 16. Doc 14 Linux Grep commands .. plz work on it..

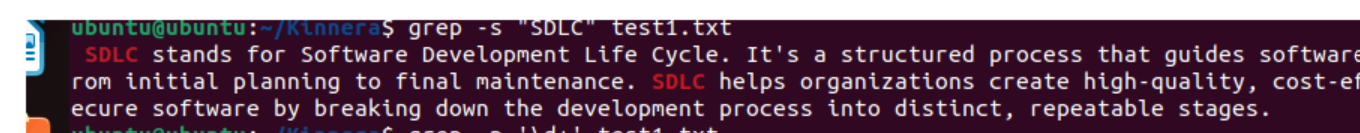
Used :-i,v,c,l,n,r,b,c,s,d,p

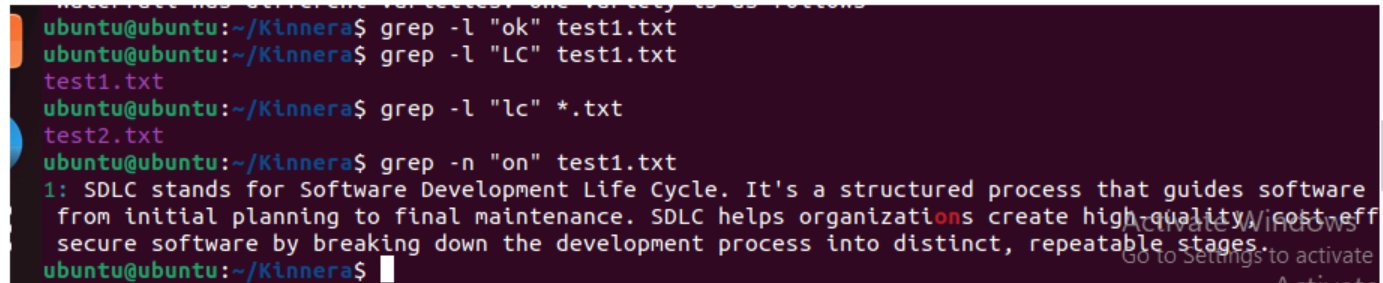


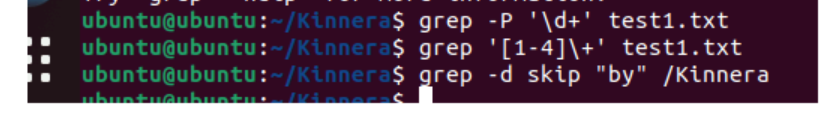




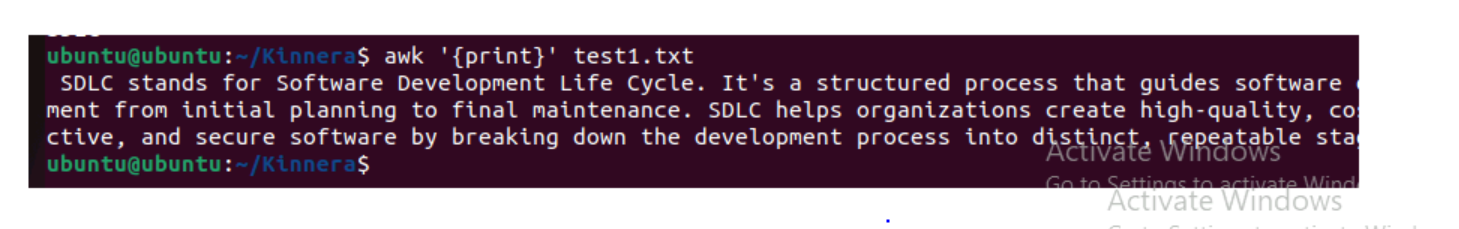


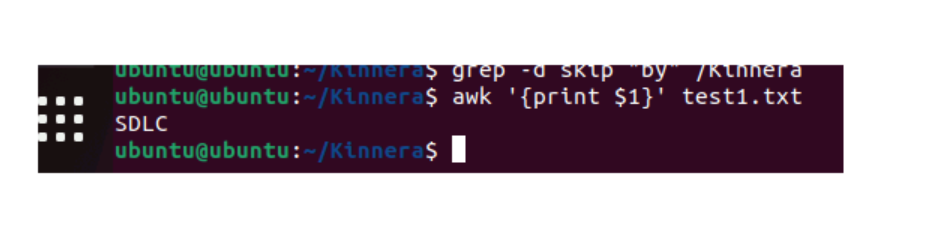
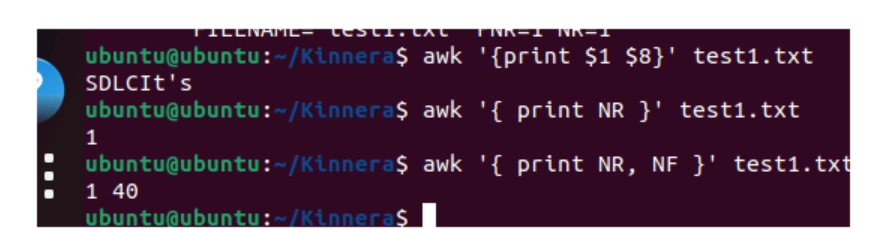
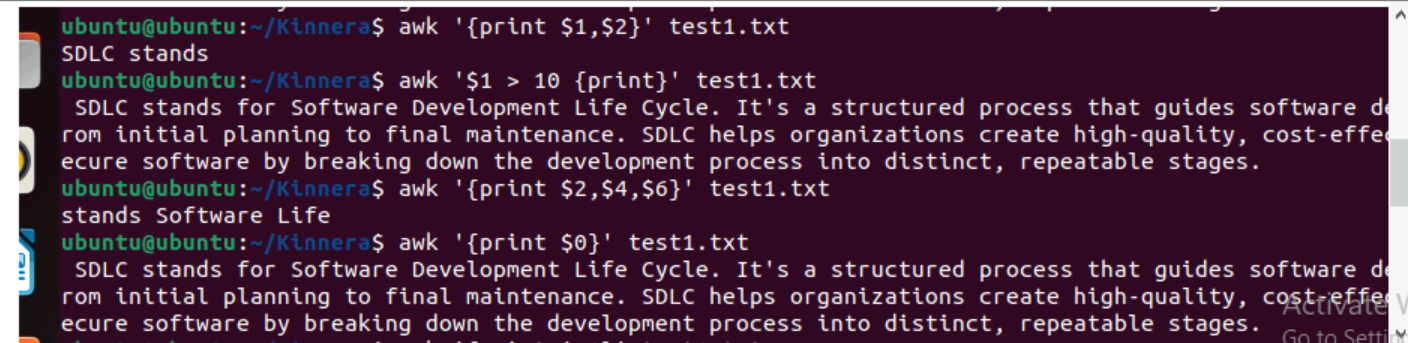


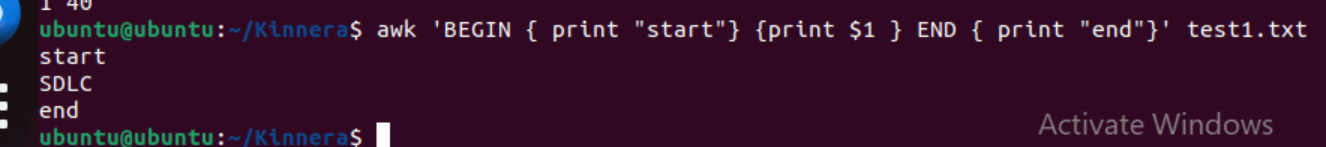




Task 17. AWT commands in doc 15 Linux AWT commands..

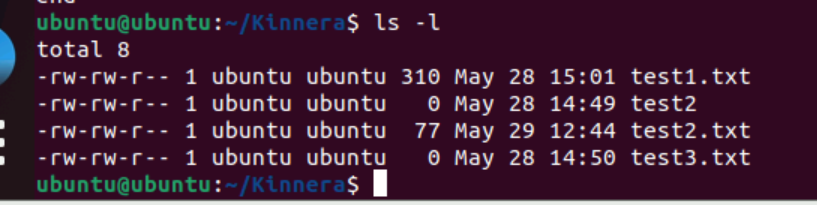






Task 18. How to check file access permission in Linux?

To check file access permissions in Linux, you can use the ls -l command. This command lists the contents of the current directory, including detailed information about each file and directory, including their permissions.



Task 19 . What are the default permissions for a new file ?

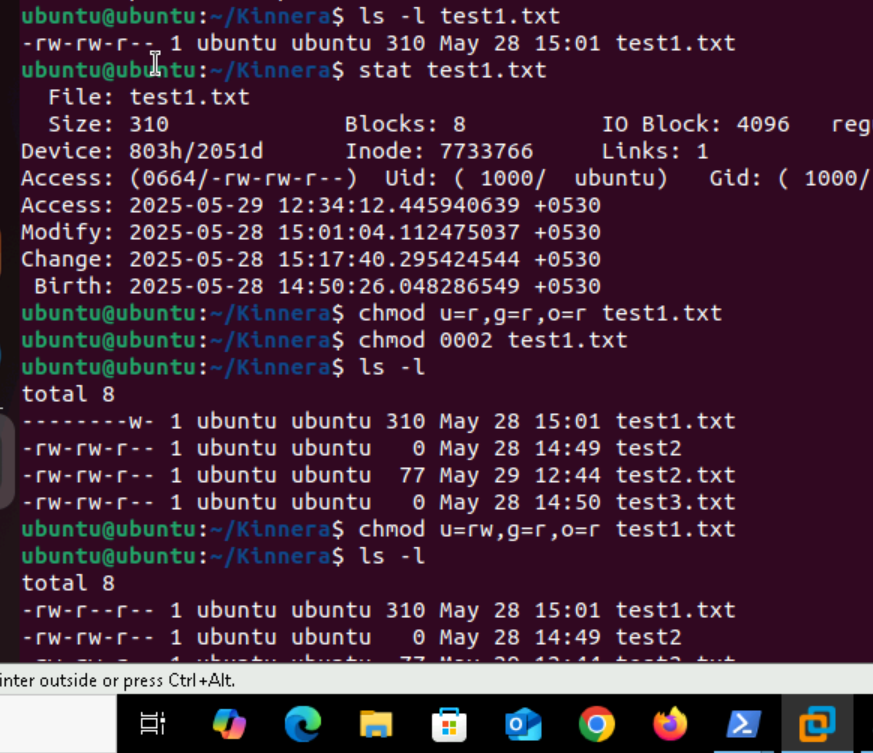
Plz find out for

Owner   → ?

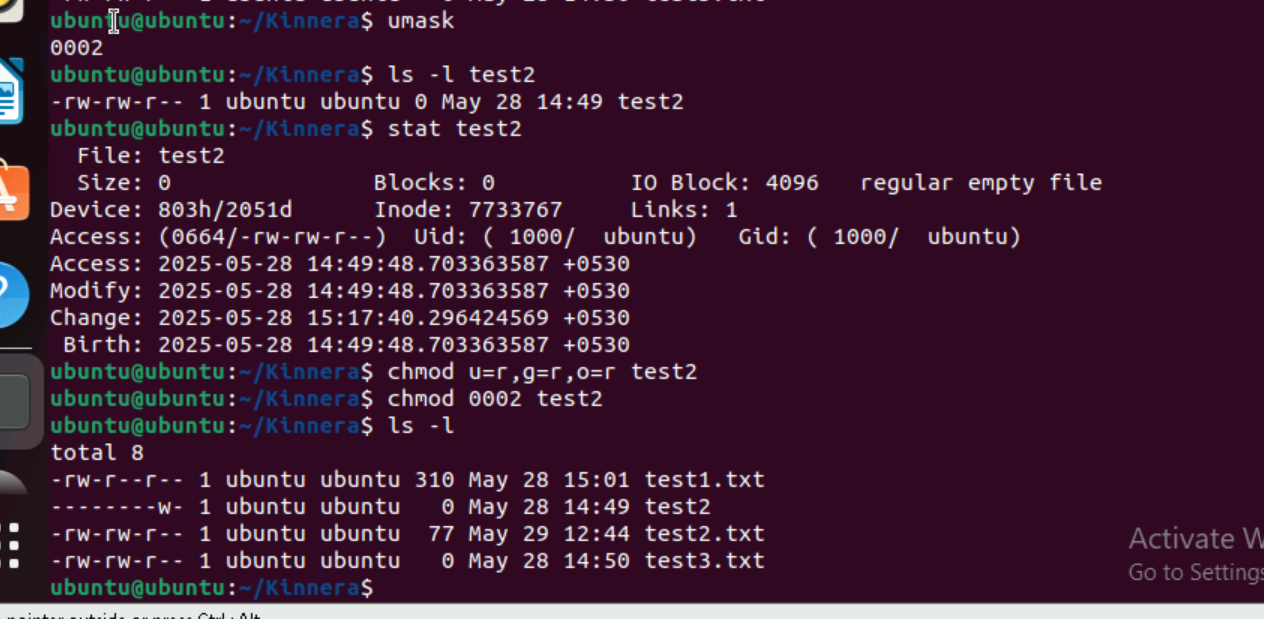
Group → ?

All and others → ?

Juz write no ss req

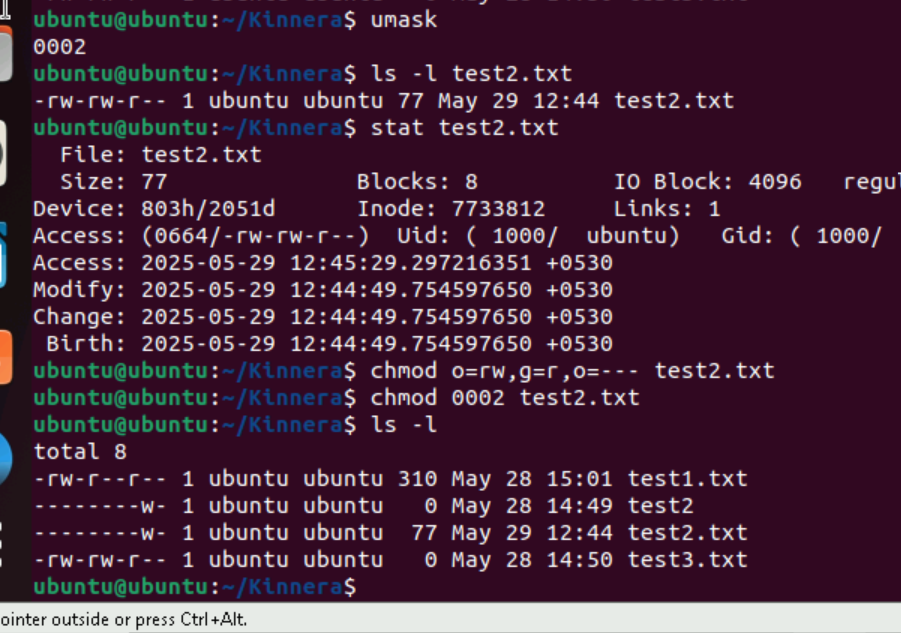


Task 20. What is the command to change the permisssion to read only for the owner, group and all other users

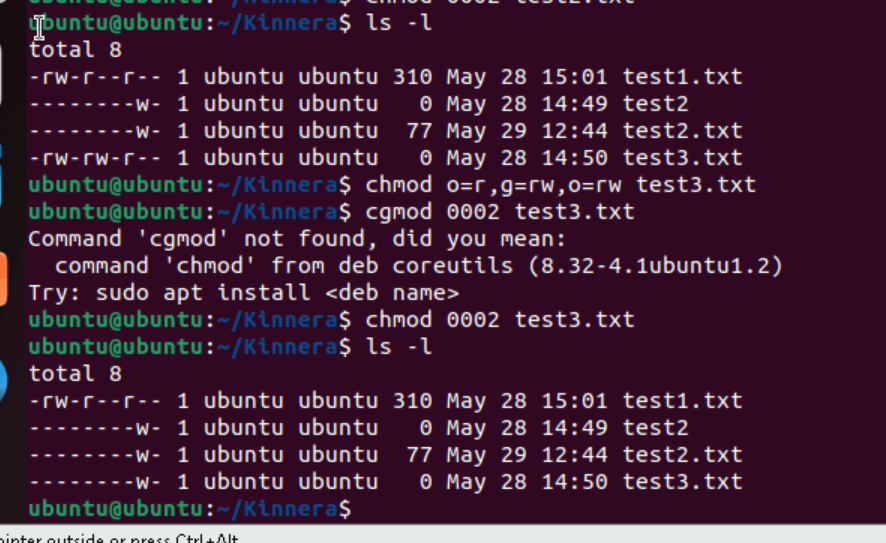


Task21. Can you change the file permissions to match the following:

* owner: Read and Write
* group: Read
* other: no permissions (None)



Task22. What was the command for changing the file permissions to -rw-r-----?



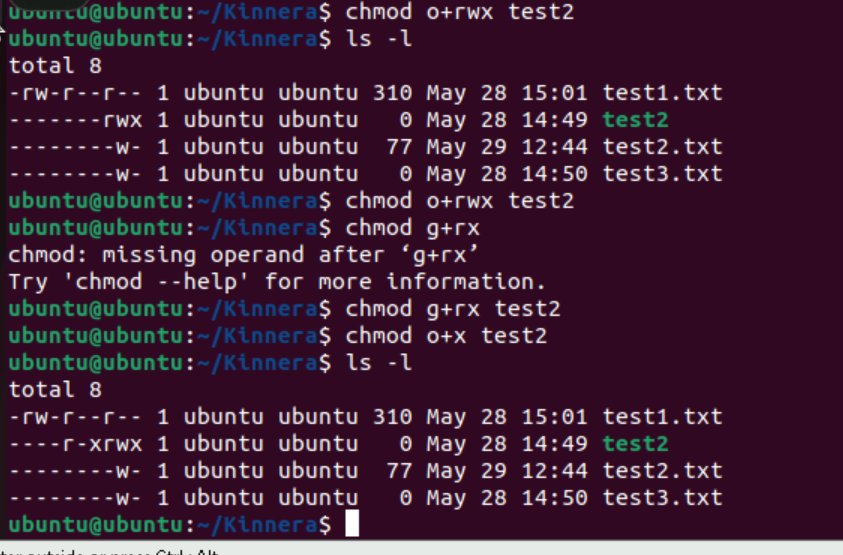
Task 23. Change chmod.exercises permissions to -rwxr-x--x

Change the file permissions to match the following:

owner: Read, Write and Execute

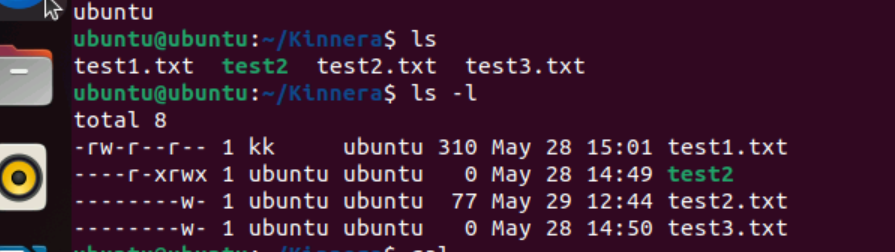
group: Read and Execute

other: Execute



Task25. Guys what will this command do?

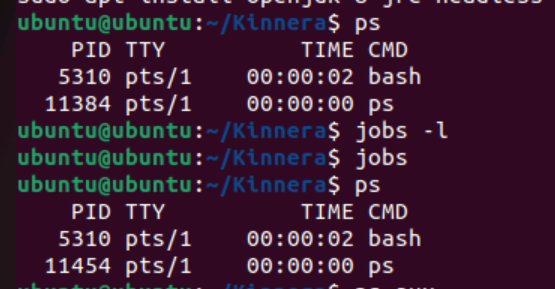
chown -c master file1.txt



Task 26. Can you define what is  a process

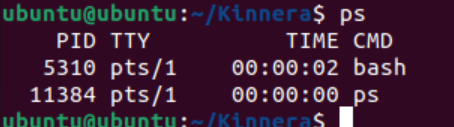
In Linux, a process is a program that is currently running or executing. It's essentially a running instance of a program, and any command you execute in Linux starts a new process. Each process has a unique identifier called a Process ID (PID).

Task27. What is command to check foreground process and background process



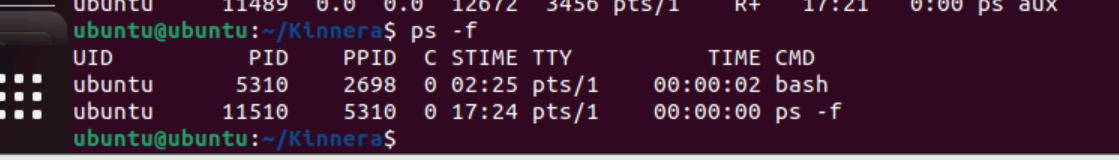
Task 28.Can you list all the running processes?

ps, ps aux

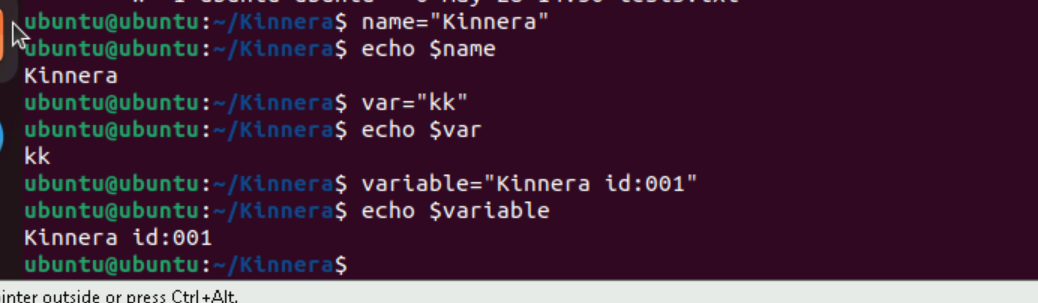
t

Task 29 . What will ps -f command do ? plz try n check .. ss required.

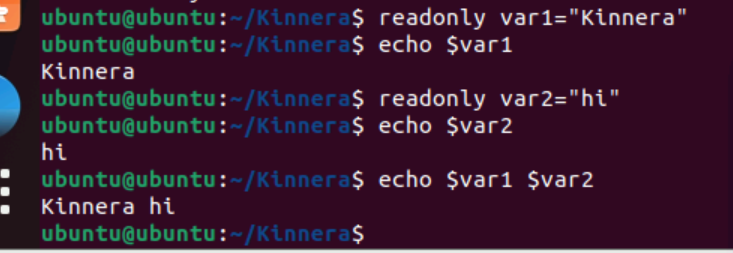
It’s used to display detailed information about processes in a full format listing.



Task 30. Can you createa  a variable name with your name in it

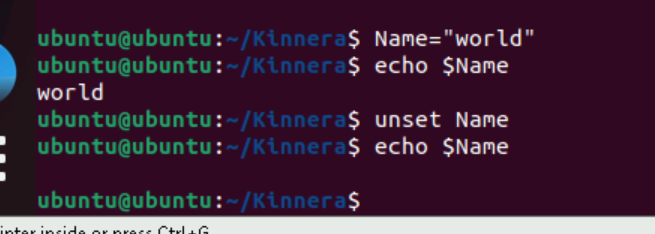


Task 31. Can you make the above name variable read only..



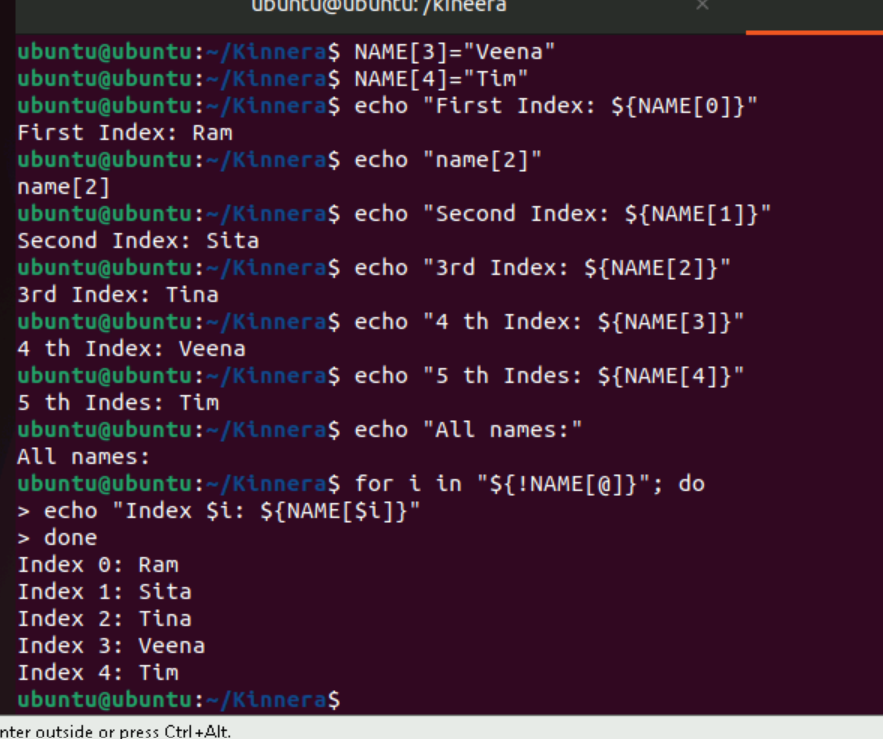
Task 32. Now will unset or delete the variables

When we use unset command it wont give any output.

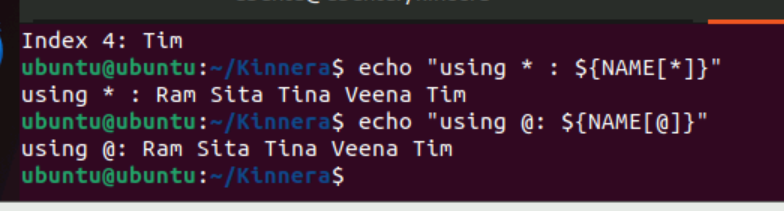


Task 33. Can u try to add a list of your friends names in an array and try to printout

First 2 lines of code unable to take screen shoot please check that



Task 34. Can you print all the list at once in an array.. Try the below cmds and check



Task 35. Plz let me know whats the output of the below snippet:

a=0

while [ "$a" -lt 10 ]    # this is loop1

do

   b="$a"

   while [ "$b" -ge 0 ]  # this is loop2

   do

      echo -n "$b "

      b=`expr $b - 1`

   done

   echo

