Task 1:

RegEX Symbols in linux

. Matches any single character (except newlines, normally)

\ Escape a special character (e.g. \. matches a literal dot)

? The preceding character may or may not be present (e.g. /hell?o/ would match hello or helo)

\* Any number of the preceding character is allowed (e.g. .\* will match any single-line string, including an empty string, and gets used a lot)

+ One or more of the preceding character (.+ is the same as .\* except that it won’t match an empty string)

| match the preceding section or the following section (e.g. hello|mad will match “hello” or “mad”)

() group a section together. This can be useful for conditionals ((a|b)), multipliers ((hello)+), or to create groups for substitutions (see below)

{} Specify how many of the preceding character (e.g. a{12} matches 12 “a”s in a row)

[] Match any character in this set. - defines ranges (e.g. [a-z] is any lowercase letter), ^ means “not” (e.g. [^,]+ match any number of non-commas in a row)

^ Beginning of line

$ End of line

Task 2:

What are the imp features of Linux os ?

Linux is a powerful and versatile operating system known for its open-source nature, strong security features, and flexibility. It's widely used for servers, desktops, and embedded systems, offering a customizable environment and excellent performance.

Here are some key features of Linux:

1. Open Source and Free

2. Security

3. Multitasking and Multiuser

4. Portability

5. Customization

6. Stability

7. Powerful Command-Line Interface (CLI)

8. Graphical User Interface (GUI)

9. Active Community Support

10. Resource Efficiency

Task 3:

What is Kernel and can you explain its functions

In Linux (and in general operating systems), a kernel is the core component that acts as a bridge between software and hardware. It manages system resources and allows software (applications and services) to interact safely and efficiently with hardware (CPU, memory, disk, etc.).

What is the Kernel?

The kernel is the lowest level of the operating system.

It runs in memory at all times and controls everything in the system.

It operates in privileged (kernel) mode, unlike user programs that run in user mode.

Main Functions of the Linux Kernel

Process Management:

Creates, schedules, and terminates processes.

Ensures each process gets fair CPU time using scheduling algorithms.

Handles multitasking and process isolation.

Memory Management:

Allocates and deallocates memory to processes.

Implements virtual memory, allowing programs to use more memory than physically available.

Manages paging and swapping.

Device Management:

Uses device drivers to control hardware devices.

Provides a unified interface so software doesn’t need to know hardware details.

Handles input/output operations.

File System Management:

Manages data storage and retrieval using file systems (ext4, XFS, etc.).

Controls access to files with permissions.

Organizes data into files and directories.

System Calls and Security:

Provides system calls (APIs) that user applications use to interact with the system.

Enforces permissions, user privileges, and sandboxing.

Networking:

Manages network interfaces and connections.

Implements protocols (TCP/IP stack).

Supports network firewalls and packet filtering (e.g., via iptables, nftables).

Types of Kernels (FYI)

Monolithic Kernel: Everything (device drivers, file system, etc.) runs in kernel space. Example: Linux kernel.

Microkernel: Only essential functions (IPC, basic scheduling) are in the kernel. Other services run in user space. Example: Minix.

Hybrid Kernel: Mix of both. Example: Windows NT.

When you run a command like ls:

The shell executes the ls binary.

The binary makes system calls (like read()) to access the directory.

The kernel handles the call, talks to the filesystem and disk driver.

Kernel retrieves data, passes it back to the ls command.

You see the directory listing.

Task 4:

What is BASH and full form with explanation.

BASH stands for Bourne Again SHell.

It is a command-line interpreter or shell for Unix/Linux systems. BASH is both a command processor (you type commands, and it runs them) and a scripting language used for writing automation scripts.

Task 5:

What is the difference between linux and windows.

Linux: Linux could be a free and open supply OS supported operating system standards. It provides a programming interface still as a program compatible with operating system primarily based systems and provides giant selection applications. A UNIX operating system additionally contains several developed parts, leading to a 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 packages is that Linux is completely free from price whereas Windows is a marketable package and is expensive. Associate operating system could be a program meant to regulate the pc or computer hardware Associate behave as a treater between user and hardware.

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 is the heart of the system, managing hardware and providing low-level services. System libraries offer pre-built functions for common tasks, and system utilities are programs that manage the system's configuration and operations.

Here's a more detailed breakdown:

Kernel:

The kernel is the foundation of the Linux operating system. It acts as a bridge between hardware and software, managing resources like memory, CPU, and input/output devices. It also handles tasks like memory management, process scheduling, and network communication.

System Libraries:

These libraries provide pre-written functions that allow applications and system utilities to access and utilize the kernel's capabilities. They implement many of the OS functionalities and avoid requiring kernel module access rights.

System Utilities:

These are specialized programs that perform specific tasks to manage the system, such as file management, device configuration, and user administration

Task 7:

Is it legal to edit Kernel?

Yes, it is legal to modify the Linux Kernel, according to the GNU General Public License (GPL). The GPL explicitly grants users the right to change and distribute the software, according to several sources

Task 8:

Explain about LILO.

LILO (Linux Loader) is an older, simpler bootloader for Linux that was once commonly used but has largely been replaced by GRUB. It was used to load the Linux kernel into memory, initiating the boot process. LILO is configured through a file called lilo.conf. While still used by some distributions like Slackware, it's generally not the primary bootloader for most modern Linux systems.

Task 9:

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

In Linux and other Unix-like operating systems, a shell is a command-line interface (CLI) program that allows users to interact with the operating system.

The shell:

Accepts user input (commands)

Passes them to the operating system (kernel) for execution

Displays the result/output

The shell can also run scripts, which are text files containing a series of shell commands.

There are many different types of shells. The most common ones include:

| Shell Name | Command to Start | Description |
| --- | --- | --- |
| Bourne Shell | sh | The original Unix shell; simple and fast; basis for other shells |
| Bash (Bourne Again Shell) | bash | Most widely used shell in Linux; supports scripting, history, auto-completion |
| C Shell | csh | Syntax similar to the C programming language |
| Korn Shell | ksh | Combines features of Bourne and C shells; good for scripting |
| Z Shell | zsh | Advanced shell with powerful features and customization |
| Fish (Friendly Interactive Shell) | fish | User-friendly shell with autosuggestions and modern syntax |
| Dash (Debian Almquist Shell) | dash | Lightweight, fast shell; used for system scripts (e.g., /bin/sh on Ubuntu) |

Task 10:

What is swap space?

Swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM.

Task 11:

Mounting in Linux means making a file system accessible at a certain point in the directory tree.

Linux treats everything as a single hierarchical directory structure starting at the root /.

Devices like hard drives, USB drives, or partitions need to be attached ("mounted") to this directory structure before you can use the files on them.

The directory where you attach the device is called a mount point (e.g., /mnt/usb or /media/cdrom).

Create a mount point (if it doesn't exist):

sudo mkdir /mnt/mydrive

Mount the device:

sudo mount /dev/sdX1 /mnt/mydrive

Unmount the device:

sudo umount /mnt/mydrive or sudo umount /dev/sdX1

Checking Mounted File Systems

mount or df -h

Task 12:

What is chmod command and how to use it ?

The chmod command in Linux stands for "change mode". It is used to change file or directory permissions.

In Linux, every file and directory has permissions that control who can:

Read it (r)

Write to it (w)

Execute it (x)

These permissions are defined for three types of users:

User (u) – The owner of the file

Group (g) – Members of the group assigned to the file

Others (o) – Everyone else

Check:

ls -l

How to use:

chmod [options] mode filename

#### **1. Symbolic Mode (using letters)**

| **Symbol** | **Meaning** |
| --- | --- |
| u | user (owner) |
| g | group |
| o | others |
| a | all (u+g+o) |
| + | add permission |
| - | remove permission |
| = | set exact value |

chmod u+x file.sh # Add execute permission for the user

chmod g-w file.txt # Remove write permission from group

chmod o=r file.log # Set read-only for others

#### **2. Numeric Mode (using numbers)**

Each permission is represented by a number:

| **Permission** | **Value** |
| --- | --- |
| r | 4 |
| w | 2 |
| x | 1 |

chmod 755 script.sh

Breakdown:

7 (4+2+1) = user: read, write, execute

5 (4+0+1) = group: read, execute

5 (4+0+1) = others: read, execute

bash

Copy

Edit

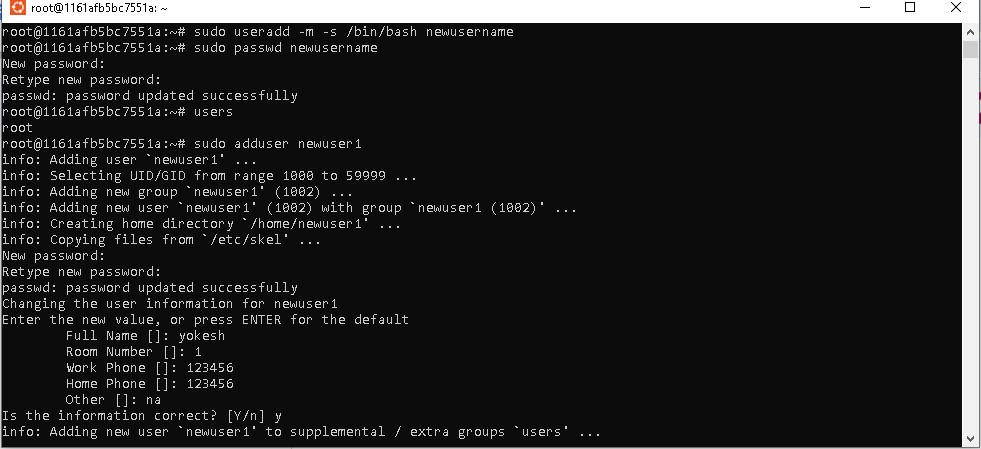
chmod 644 file.txt

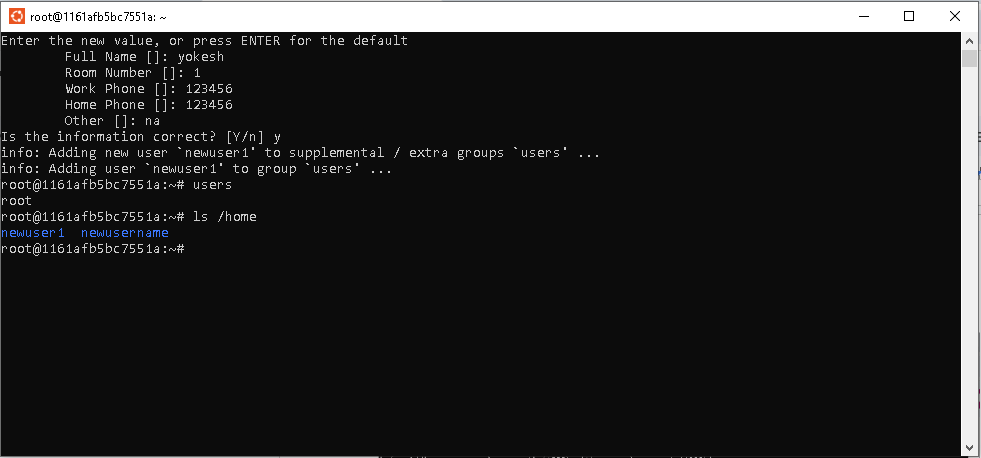
user: read, write

group: read

others: read

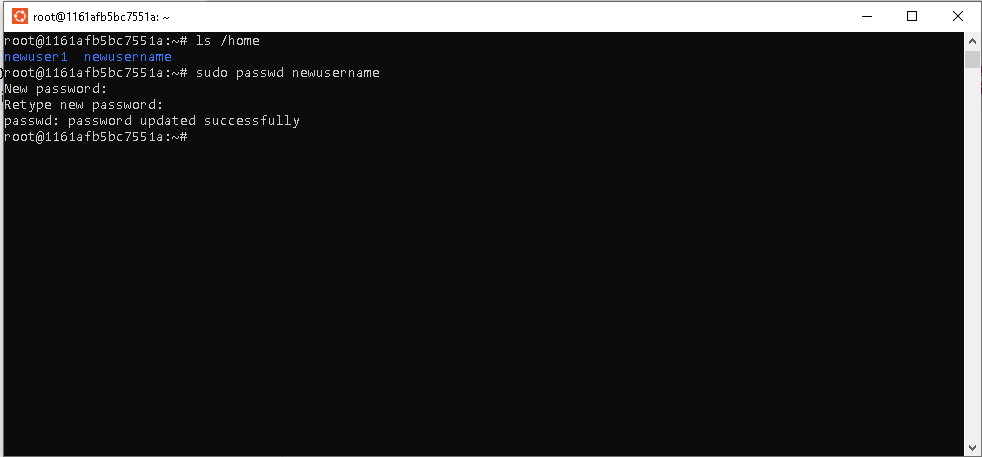
Task 13:





Task 14:

Change password



Task 15:

What is the difference between process and threads.

Process

A process is a computer program under execution. Linux is running many processes at any given time. We can monitor them on the terminal using the ps command or on the System Monitor UI. For instance, let’s see an example of using the ps command to view all the processes running on the machine

As we run new commands/applications or the old commands complete, we can see the number of processes grow and shrink dynamically. Linux processes are isolated and do not interrupt each other’s execution.

With a PID, we can identify any process in Linux. Internally, the kernel uniquely allocates this number and releases it for reuse after the process exits. We can see PID as the second column in the output of the above ps command.

Since many processes are running at any given time in Linux, they have to share the CPU. The process of switching between two executing processes on the CPU is called process context switching. Process context switching is expensive because the kernel has to save old registers and load current registers, memory maps, and other resources.

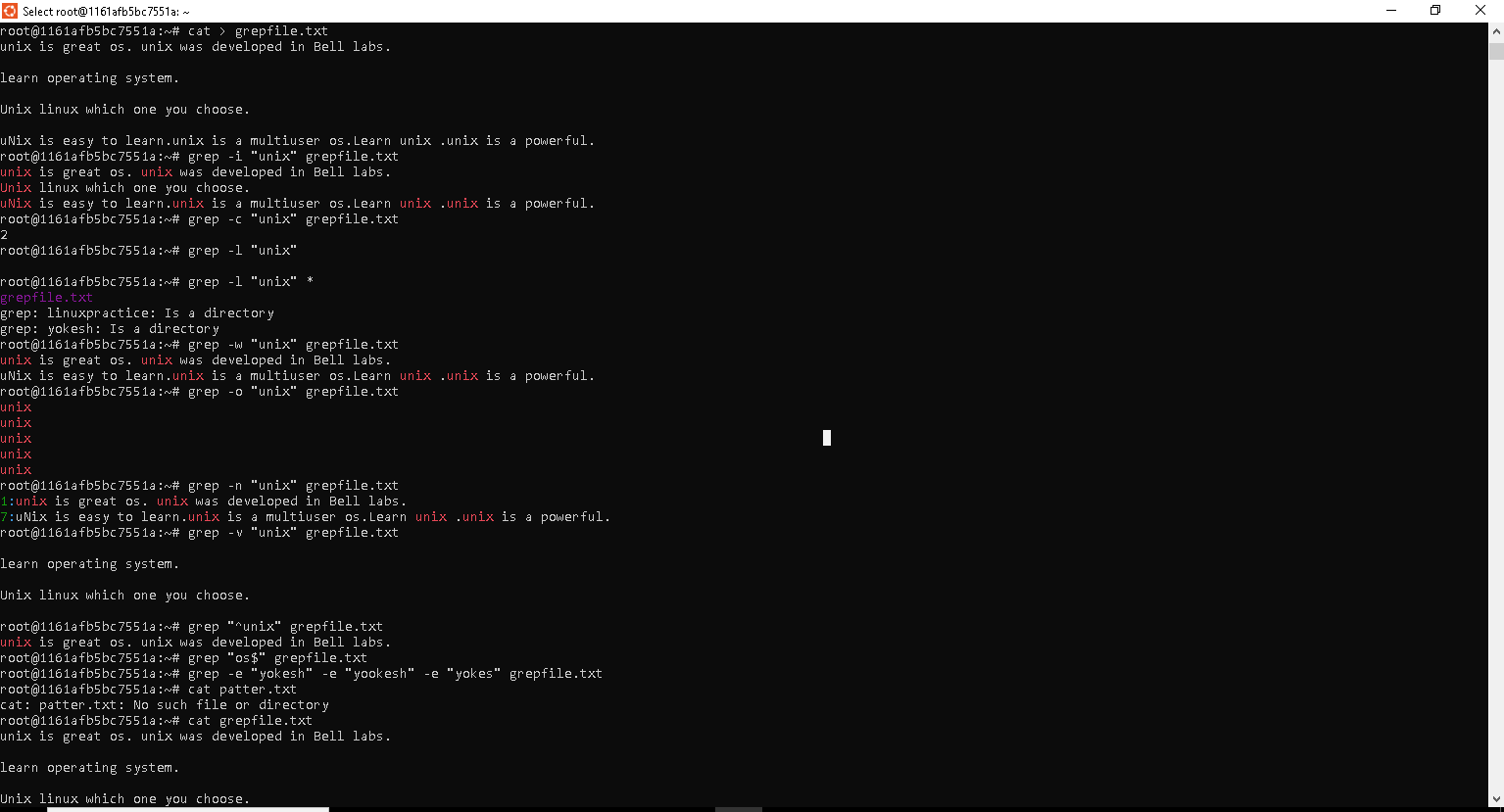
## Thread

A thread is a lightweight process. A process can do more than one unit of work concurrently by creating one or more threads. These threads, being lightweight, can be spawned quickly.

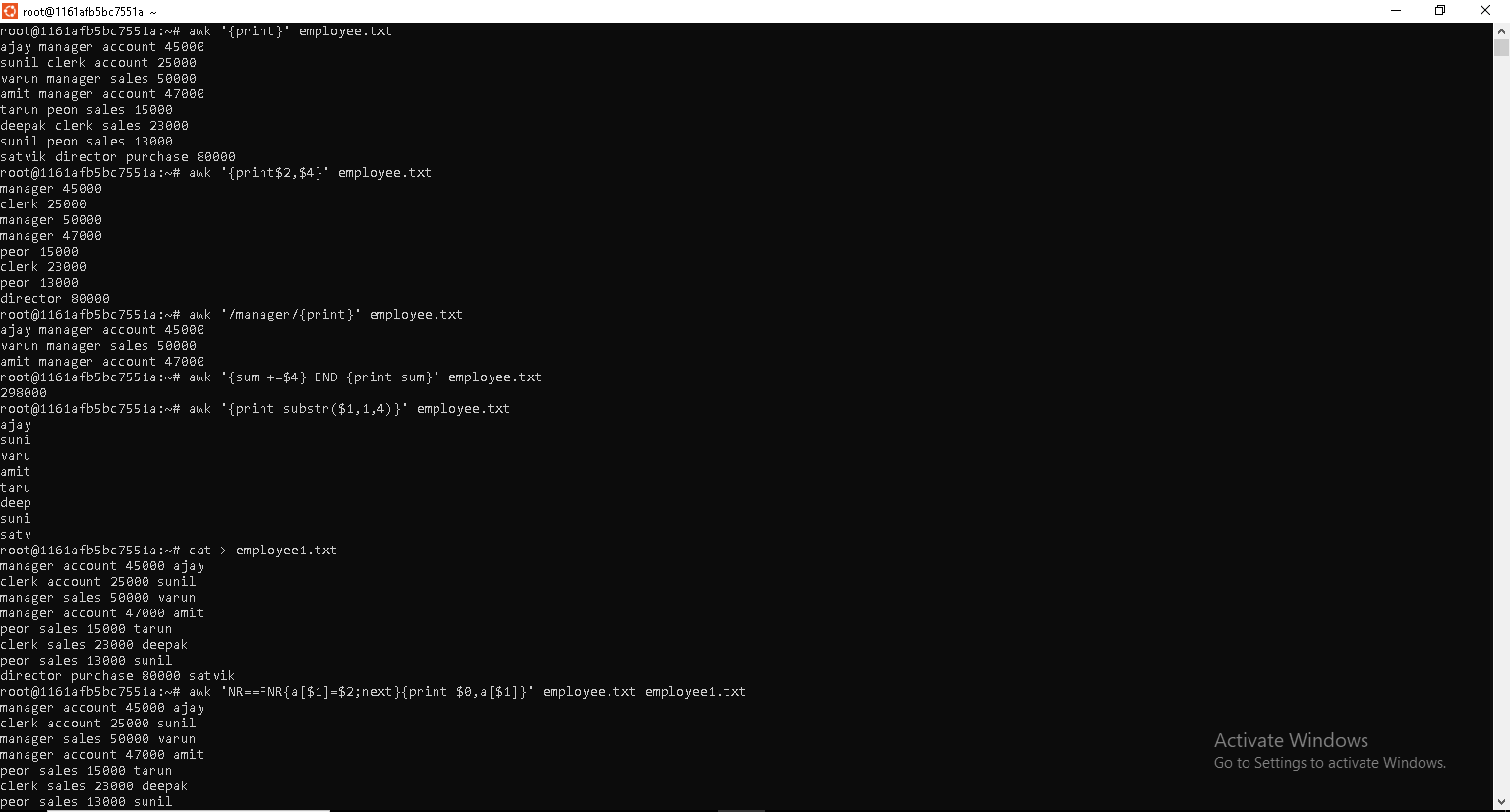
Let’s see an example and identify the process and its thread in Linux using the *ps -eLf* command. We’re interested in PID, LWP, and NLWP attributes:

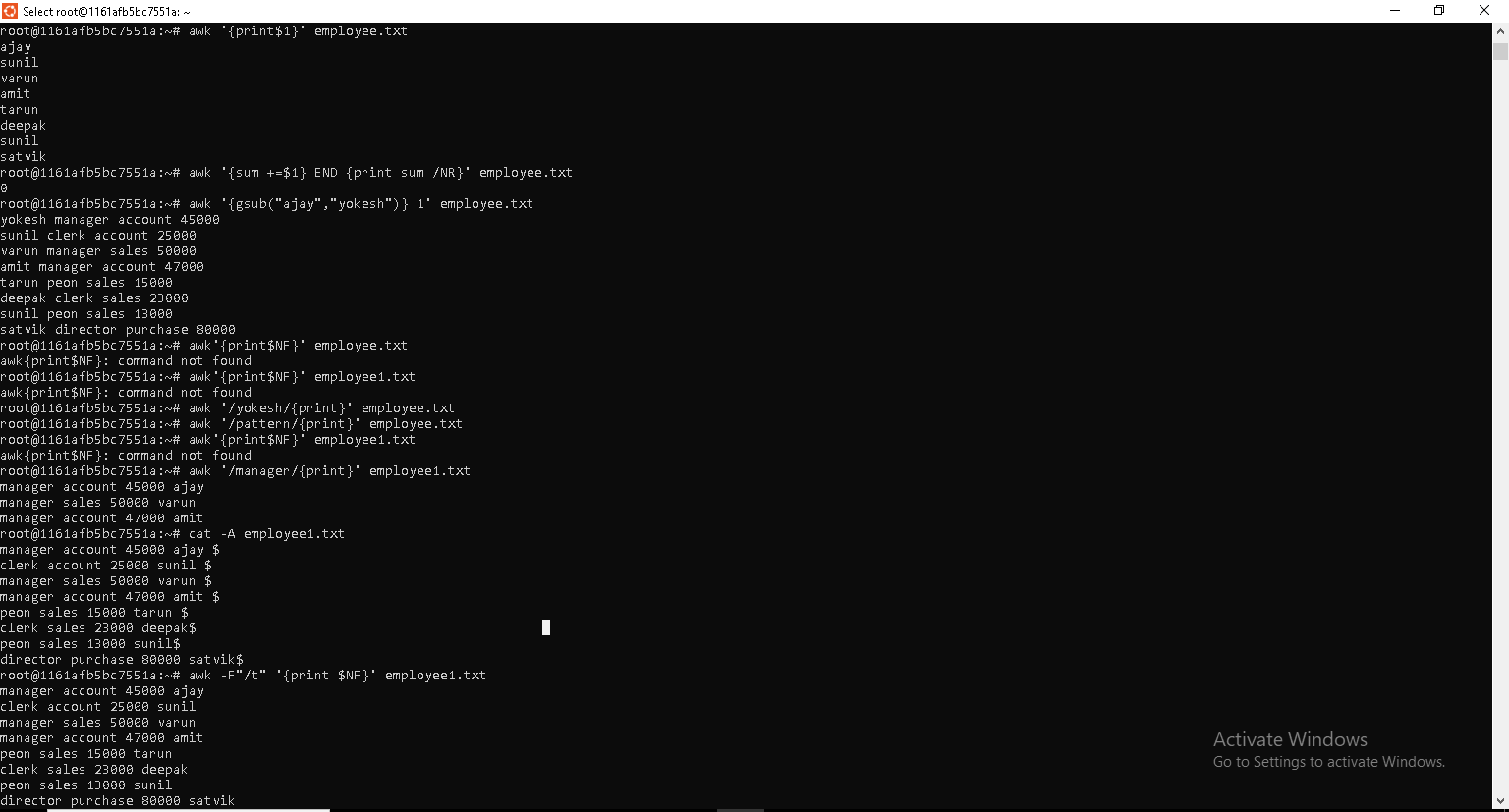
* PID: Unique process identifier
* LWP: Unique thread identifier inside a process
* NLWP: Number of threads for a given process

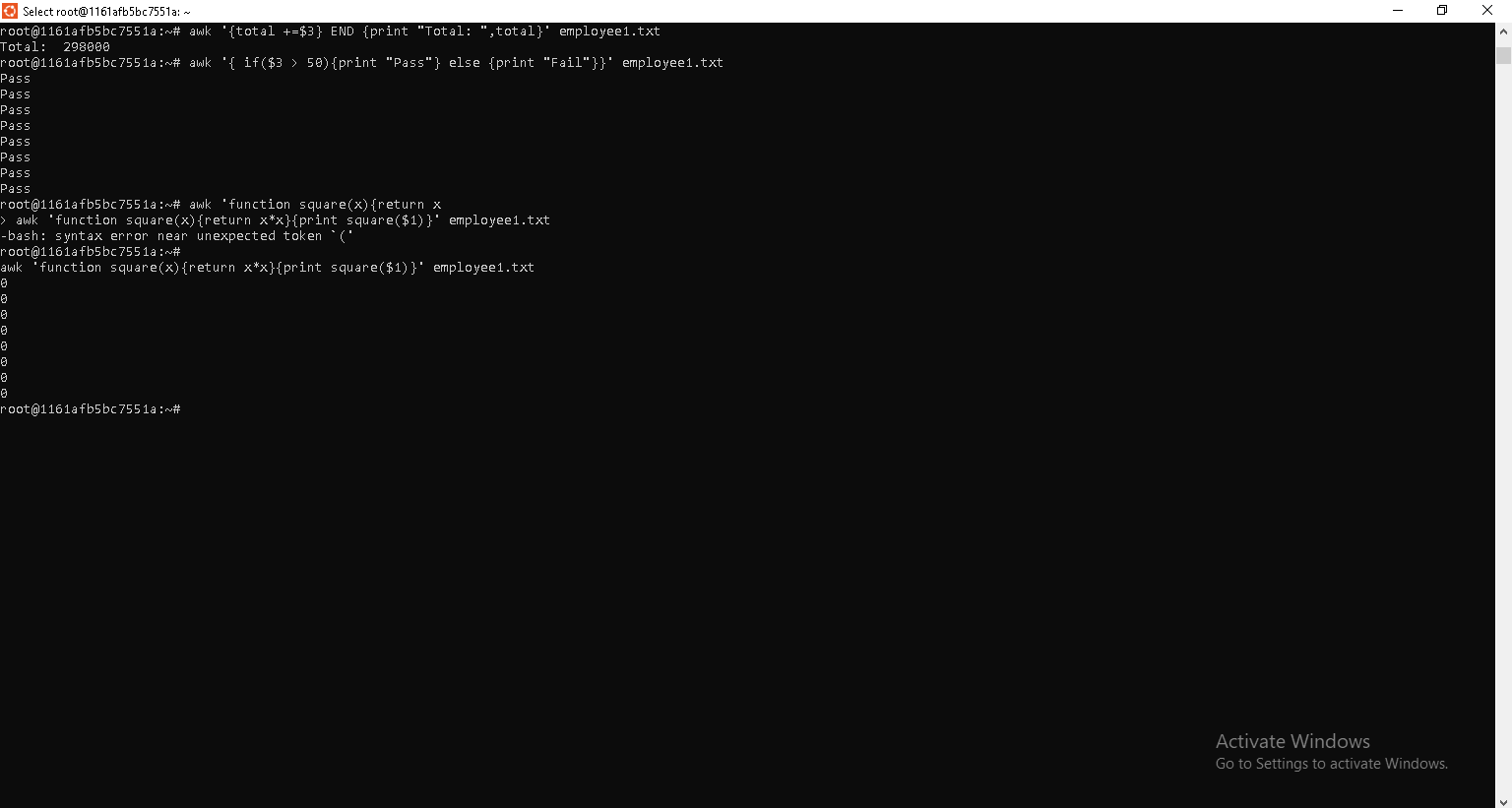
Task 16: grep command



Task 17: awk commands







Task 19:

What are the default permissions for a new file ?

When you create a new file in Linux, it is assigned default permissions based on two factors:

This means:

User (owner): read & write

Group: read & write

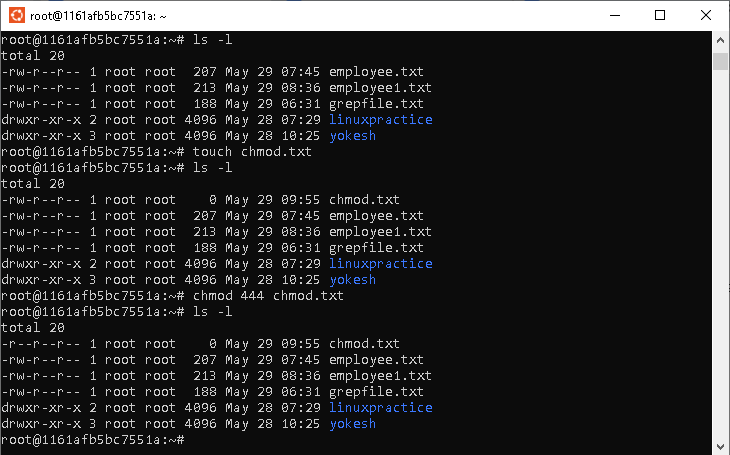
Others: read & write

**An instance of a running program.**

Whenever you run a command or open an application, Linux creates a **process** for that task. The process includes the program code, memory, system resources, and other information the OS needs to manage it.

Task 20:

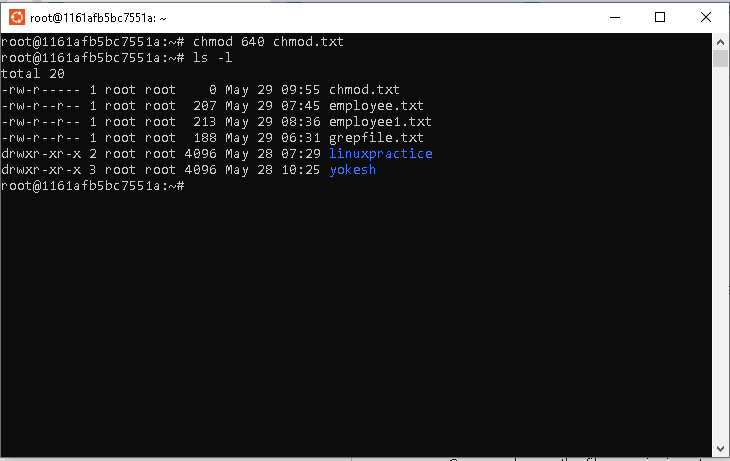
Chmod



Tosk 21:

Can you change the file permissions to match the following:

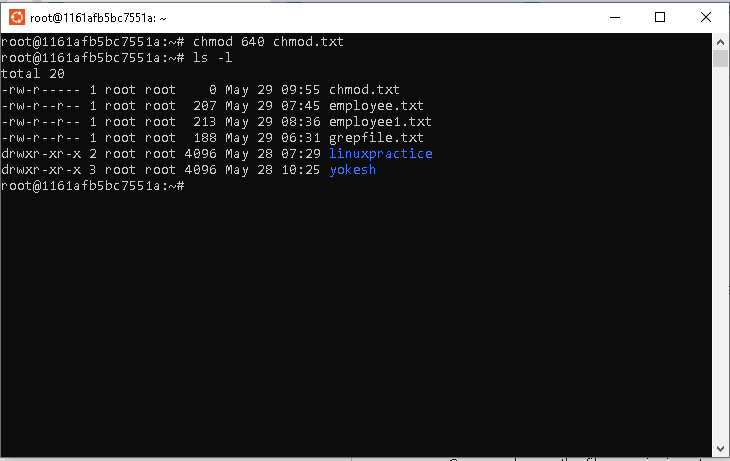
* owner: Read and Write
* group: Read
* other: no permissions (None)



Task 22:

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

Ans chmod 640 chmod.txt



Task 23:

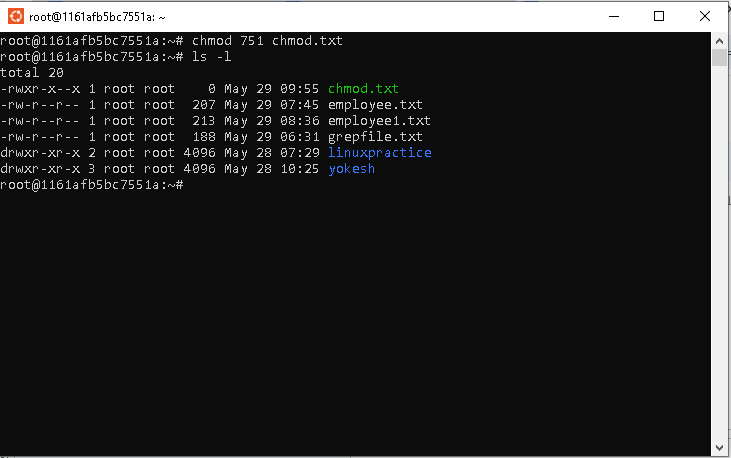
Change chmod.exercises permissions to -rwxr-x--x

Change the file permissions to match the following:

owner: Read, Write and Execute

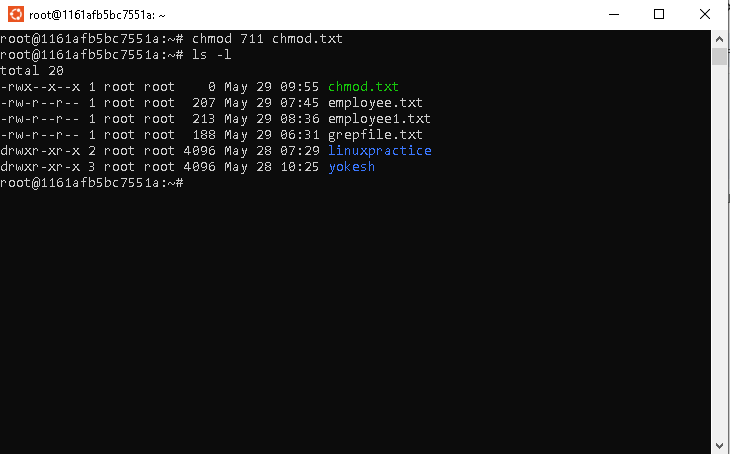
group: Read and Execute

other: Execute

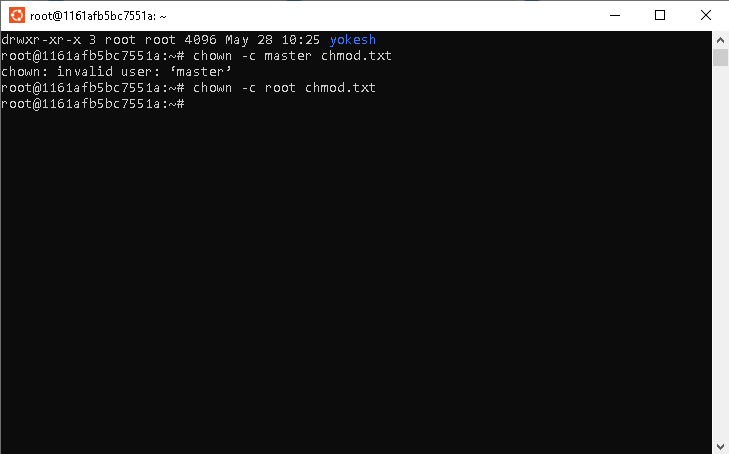


Task 24:

What was the command for changing the file permissions to -rwxr-x--x



Task 25:

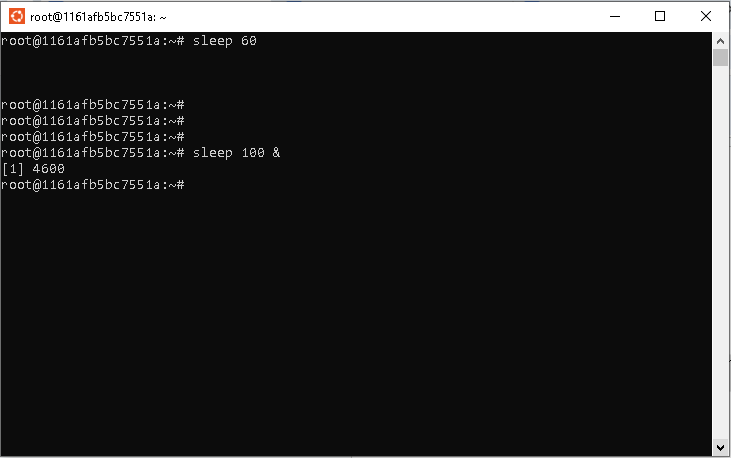


Task 26:

An instance of a running program.

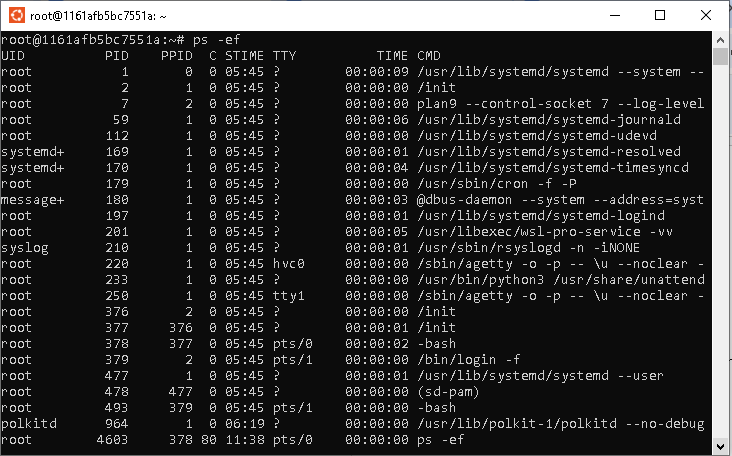
Whenever you run a command or open an application, Linux creates a process for that task. The process includes the program code, memory, system resources, and other information the OS needs to manage it.

Foreground process and background process:

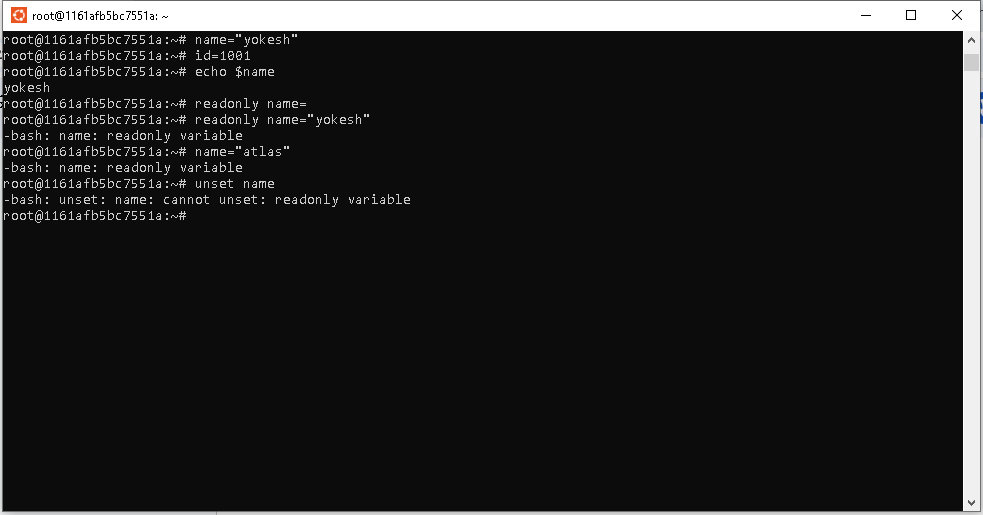


Can you list all the running process. What ps -ef will do.

It will list all the running processes in a full format, include PID,UID, command etc.



Task 30,31,32



Task 33,34,35

