Day 6 - 28th May 2025

Employee ID: 201933938

Login ID: iamasif

Name: Shaik Asif

Task 1:

**RegEX Symbols in linux :**

A regular expression (often abbreviated as regex or regexp) is a sequence of characters that defines a search pattern. It’s used for matching, searching, and replacing text based on specific patterns.

**List them down with description:**

**.** = Any character except newline

^ = starting of a string

$ = end of string

\* = 0 or more of previous character

+ = 1 or more previous characters

? = 0 or 1 of previous character

[ ] Matches any one in the brackets

\d = Any digit (0–9)

\w = Any word character (letters, digits, \_)

5 min 9.41 to

Task 2:

**What are the imp features of Linux os ?**

#### 1.Open Source

* Source code is freely available to view, modify, and distribute.
* Encourages innovation and transparency.

#### 2. Multitasking

* Can run multiple processes simultaneously without affecting performance.
* Example: Downloading a file, editing a document, and playing music at the same time.

#### 3. Multiuser Support

* Multiple users can log in and work at the same time without interfering with each other.

#### 4. Portability

* Runs on a wide range of hardware: from supercomputers to smartphones to IoT devices.

#### 5. Security

* Strong user permissions, file access control, and features like:  
  + Firewall (iptables/ufw)
  + SELinux/AppArmor
* Linux is less targeted by malware compared to other OSes.

#### 6. Stability and Reliability

* Rarely crashes; can run for years without rebooting.
* Widely used in mission-critical systems like servers and banking infrastructure.

#### 7. Shell/Command Line Interface

* Provides powerful command-line tools for automation, scripting, and system control.

#### 8. Support for Programming Languages

* Supports nearly all major languages: C, C++, Python, Java, Ruby, Go, Perl, etc.

#### 9. Package Management Systems

* Tools like apt, yum, dnf, and pacman make software installation and updates easy.

#### 10. Filesystem Support

* Supports many file systems: ext4, xfs, btrfs, NTFS, FAT32, ZFS, etc.

6 min 9.48 to 9.54

Task 3:

**WHAT IS Kernal and can you explain its functions**

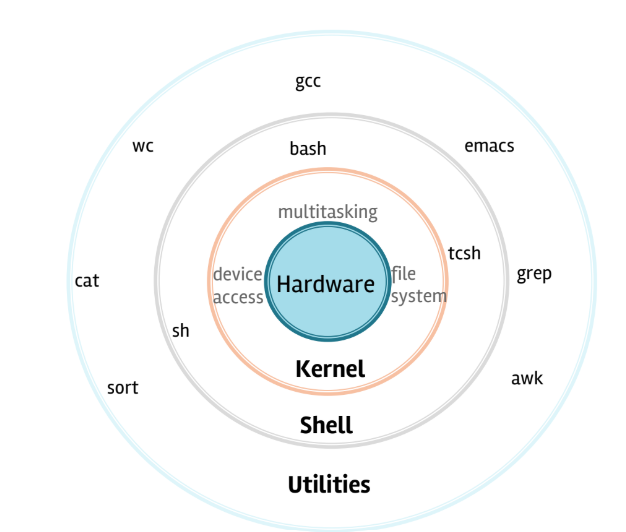
"The kernel is the core part of the operating system. It works like a bridge between the hardware and software.

When we run a program, the program doesn't talk to the hardware directly — it talks to the kernel. The kernel then sends instructions to the hardware.

It handles important things like:

* Managing the CPU and running programs (called processes),
* Controlling the RAM (memory),
* Talking to devices like keyboards, disks, and printers,
* Handling files and security.

So, without the kernel, your operating system and applications wouldn't know how to work with your computer’s hardware."



5 min 9.55 to 10.00

Oral also for rating

Task 4:

**What is BASH? Full form with explaination.**

5min 10.00 to 10.05

**BASH - Bourne Again SHell**

"BASH stands for Bourne Again SHell. It's a command-line interface used in Linux and Unix systems.

When we type commands into the terminal, BASH reads them, processes them, and tells the computer what to do.

It's like a translator between you and the system. You can use it to:

* Run programs
* Manage files and folders
* Automate tasks using scripts

BASH is very powerful. It saves time and helps users and system administrators control the system efficiently, especially without using a mouse or graphical interface."

Task 5:

**What is the diffrenece between window and linux**

Windows and Linux are two different types of computer operating systems.

* Windows is made by Microsoft, and most people use it on their home computers. It’s easy to use and works well with many popular programs and games. But it can sometimes get slow and get viruses more easily.
* Linux is free and open for anyone to use and change. It’s very secure and fast, but sometimes it needs you to type commands instead of clicking buttons. People often use Linux for servers and programming because it’s very powerful.

In short:  
 Windows is user-friendly and popular for everyday use. Linux is free, secure, and great for advanced users and servers.

5 min 10.20 to 10.25

Task 6:

**Define the basic components of Linux**

Kernel

* The core part of Linux.
* It connects the software and hardware.
* Manages memory, processes, devices, and files.

Shell

* A command-line interface between the user and the system.
* It lets you type commands to control the system.
* Example: BASH (Bourne Again SHell).

File System

* Organizes how data is stored and accessed on your disk.
* Everything in Linux is treated like a file (including devices and processes).

System Libraries

* Special programs that help software talk to the kernel.
* They provide reusable functions (like printing or opening files).

System Utilities

* Basic tools and commands used to manage the system.
* Examples: ls (list files), cp (copy files), top (show running processes).

User Interface

* Can be CLI (Command Line Interface) or GUI (Graphical User Interface).
* CLI is for typing commands, GUI is for clicking icons and menus.

10.25 o 10.30

Task 7:

**Is it legal to edit Kernal?**

Yes, it is legal to edit the Linux kernel because it is open source and released under the GNU General Public License (GPL).

2 min 10.31 to 10.33

Task 8:

**how many of you have gone through techadamy Linux plz raise ur hand**

**16 pax done out of 29**

**Can you explain LILO**

LILO stands for Linux Loader.

It is a boot loader for Linux operating systems.

A boot loader is a small program that runs when you turn on your computer. Its job is to load the operating system (like Linux) from the hard drive into the computer’s memory.

10.34 to 10.40 6 min

Task 9:

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

### What is a Shell?

A shell is a program that lets users interact with the operating system — mainly through commands.

* You type a command (like ls or mkdir), and the shell sends it to the system to be done.
* It's like a translator between you and the computer.
* It can also run scripts to automate tasks.

There are **several types of shells**, each with its own features. Here are the most common ones:

#### **1. BASH (Bourne Again Shell)**

* Most widely used shell in Linux.
* User-friendly, powerful scripting features.
* **Default shell** in many Linux distributions.
* Command prompt usually looks like: $

**2. SH (Bourne Shell)**

* The **original Unix shell**.
* Basic scripting support, not as advanced as BASH.
* Mostly used in **older systems or scripts**.

#### **3. CSH (C Shell)**

* Syntax looks more like the **C programming language**.
* Good for programming-style scripts.
* Prompt: %

#### **4. TCSH (Tenex C Shell)**

* An **improved version of CSH**.
* Adds features like **command history** and **auto-complete**.

#### **5. KSH (Korn Shell)**

* Combines features of SH and CSH.
* Good performance and scripting capabilities.
* Often used in enterprise environments.

#### **6. ZSH (Z Shell)**

* Very **powerful and customizable**.
* Supports **auto-completion**, **syntax highlighting**, plugins, and themes.
* Popular among developers using tools like **Oh My Zsh**.

10.41 to 10.51 10 min

Task 10:

**What is Swap space ?**

Swap space is a part of your computer’s storage (like hard disk or SSD) that is used as extra memory when your system runs out of RAM (Random Access Memory).

2 min 10.52 to 10.54

Task 11:

**What is Mount ? how do you mount and unmount file system in Linux?**

In Linux, "mounting" means attaching a storage device (like a USB drive, hard disk, CD, etc.) to the Linux file system, so that you can access its files.

Think of it like plugging in a pen drive, and the system needs to "connect" it to a folder so you can open and use the files.

Or

In Linux, mounting means connecting a storage device to the system so we can use its files. We mount it to a folder, and when we’re done, we unmount it to safely disconnect it

Mounting:

* sudo fdisk -l = This lists all devices.
* sudo mkdir /mnt/mydrive = Create a mount point
* sudo mount /dev/sdb1 /mnt/mydrive = Mount the device:
* cd /mnt/mydrive = Go to the folder and see your files
* Ls

Unmounting:

* sudo umount /mnt/mydrive

10 min

Plz raise hand once done till here.. So that we can go for oral..

Oral - random pick only few.. By the Facilitator/trainer.

Task 12:

**What is chmod command ? how to use it?**

chmod stands for "change mode".  
 It is used to change the permissions of files and directories in Linux.

Every file or folder has permissions that control:

* Who can read it (r)
* Who can write (edit) it (w)
* Who can execute (run) it (x)

## Types of Users in Permissions

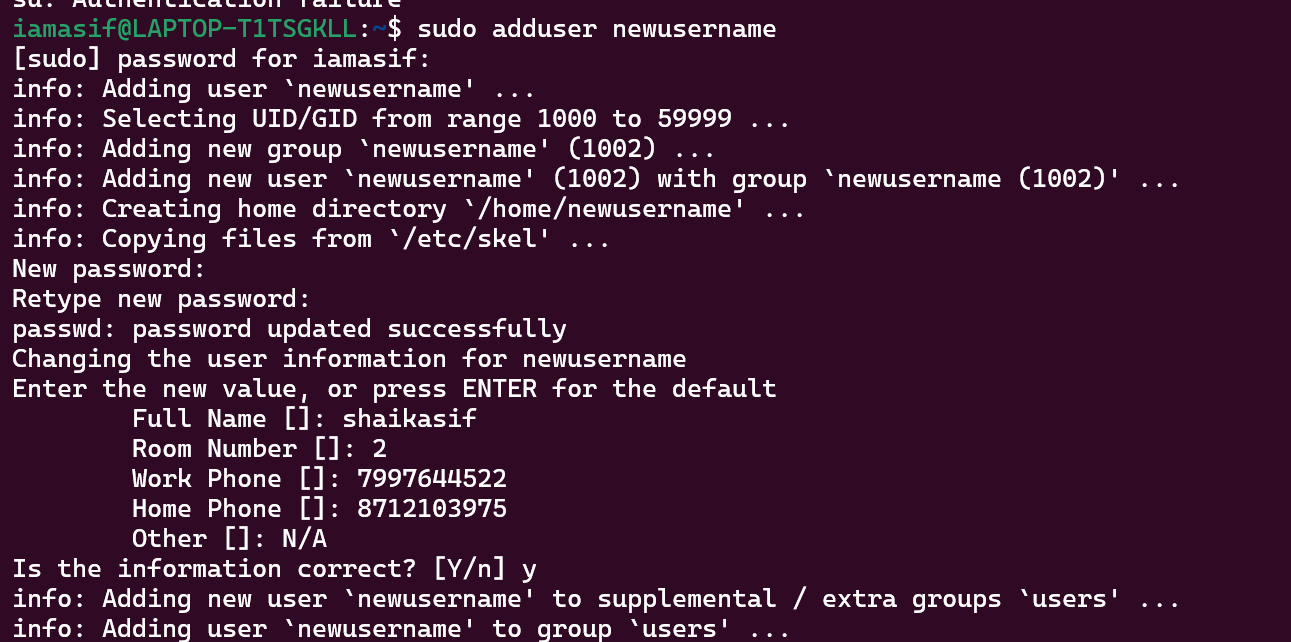
1. **u** = user (owner)
2. **g** = group
3. **o** = others
4. **a** = all (user + group + others)

Example: chmod u+x file.txt

5 min

Task 13:

**Can you add a new user account? Crate a new user in different ways and paste ss**

****

5 min

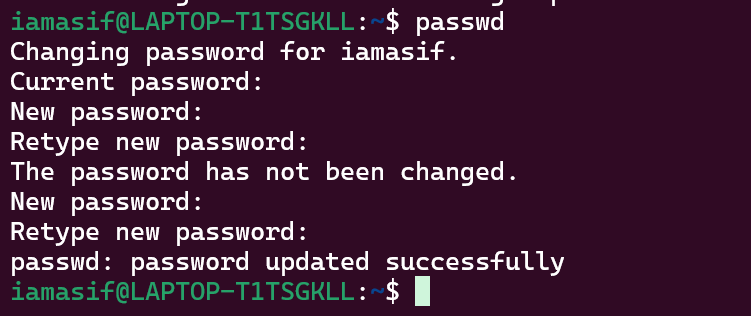
Task 14:

**Can you change the password of a user?**

**How do you do that? Plz share ss**

Yes we can change

5 min



Task 15:

**What is diff between Process and Thread?**

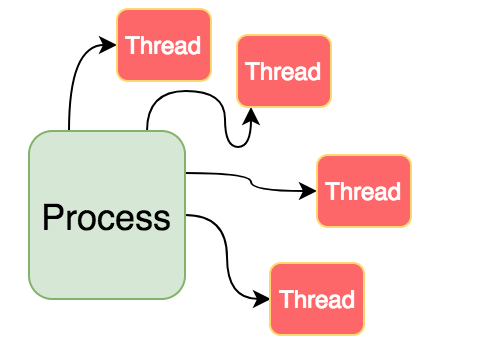
### What is a Process?

* A process is like a program that your computer is running.
* It works all by itself and has its own space to keep things like data and instructions.
* For example, when you open your web browser, it becomes a process.

### What is a Thread?

* A thread is a small part inside a process that does work.
* Threads inside the same process share the same space, so they can easily talk to each other.
* For example, your web browser uses many threads to do different jobs like loading a page, playing videos, or checking for updates —all at the same time.

### Easy way to remember:

* Process = a whole program running (like a big machine).
* Thread = a small worker inside that program doing a task.
* 

Plz explain

8 min

In the mean time others plz keep a file ready with some content in it for Grep command..

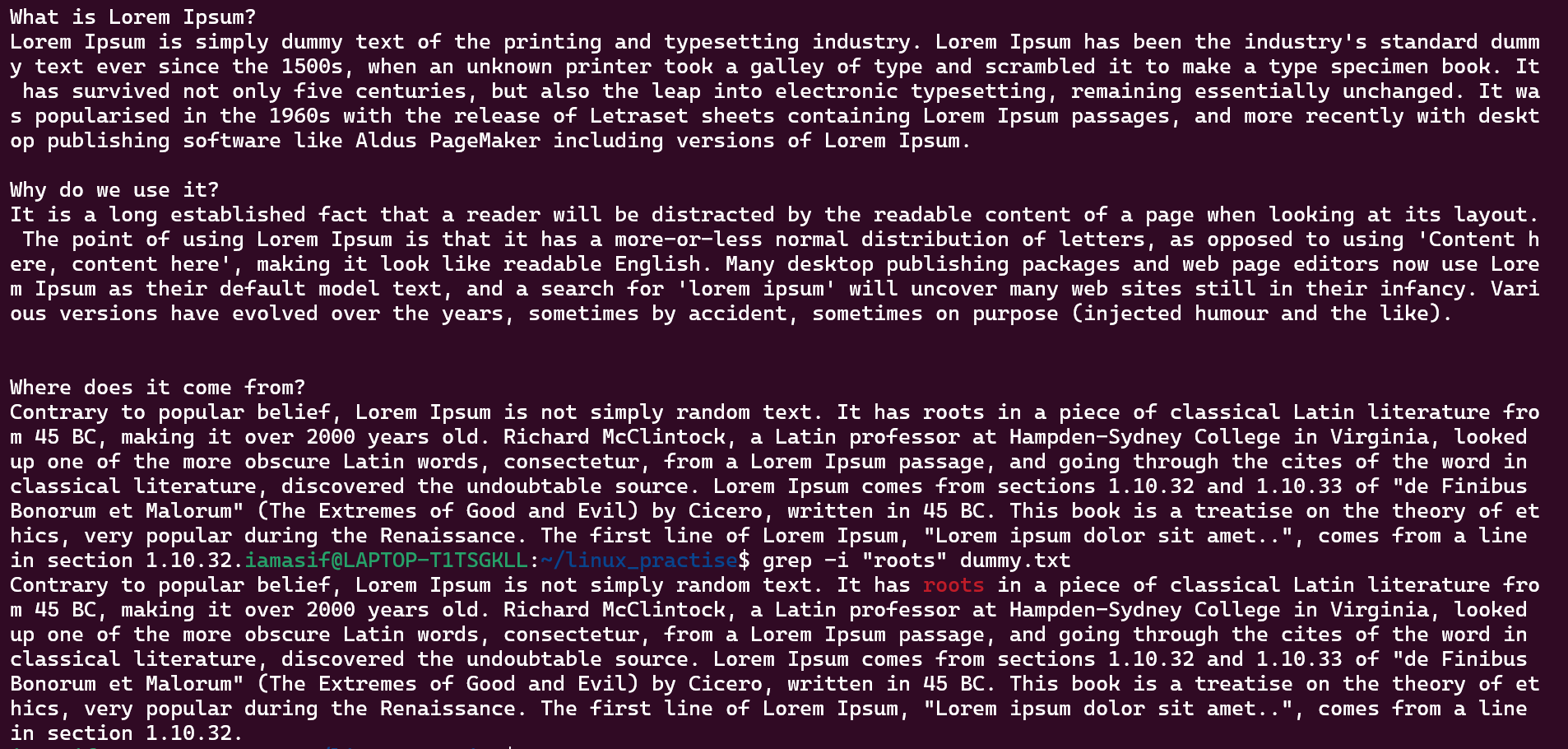
Task 16:

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

## **1. Case insensitive search**

The -i option enables to search for a string case insensitively in the given file. It matches the words like "UNIX", "Unix", "unix".

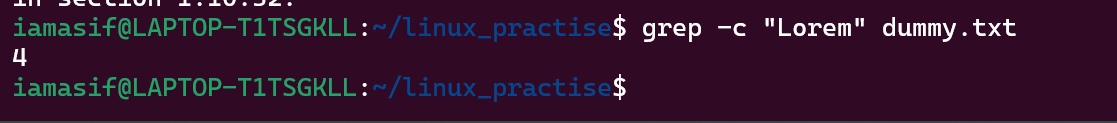
grep -i "UNix" hellofile.txt



## **2. Displaying the Count of Number of Matches Using grep**

We can find the number of lines that matches the given string/pattern

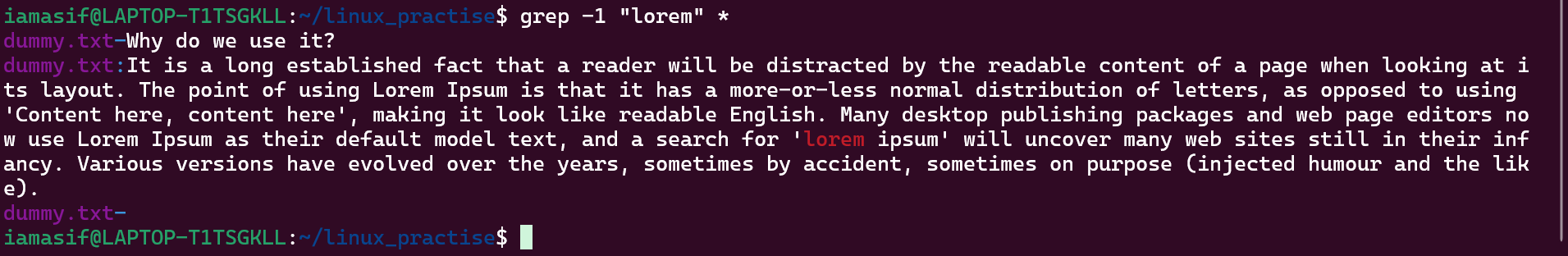
grep -c "unix" hellofile.txt



## **3. Display the File Names that Matches the Pattern Using grep**

We can just display the files that contains the given string/pattern.

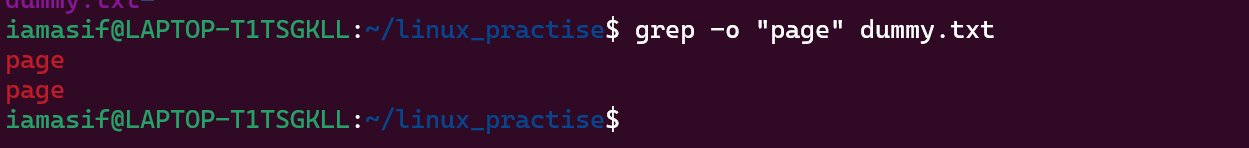
grep -l "unix" \*



## **5. Displaying only the matched pattern Using grep**

By default, grep displays the entire line which has the matched string. We can make the grep to display only the matched string by using the -o option.

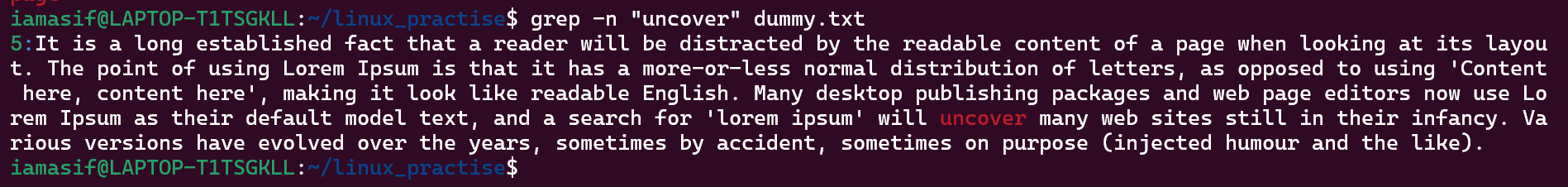
grep -o "unix" hellofile.txt



## **6. Show Line Number While Displaying the Output Using grep -n**

To show the line number of file with the line matched.

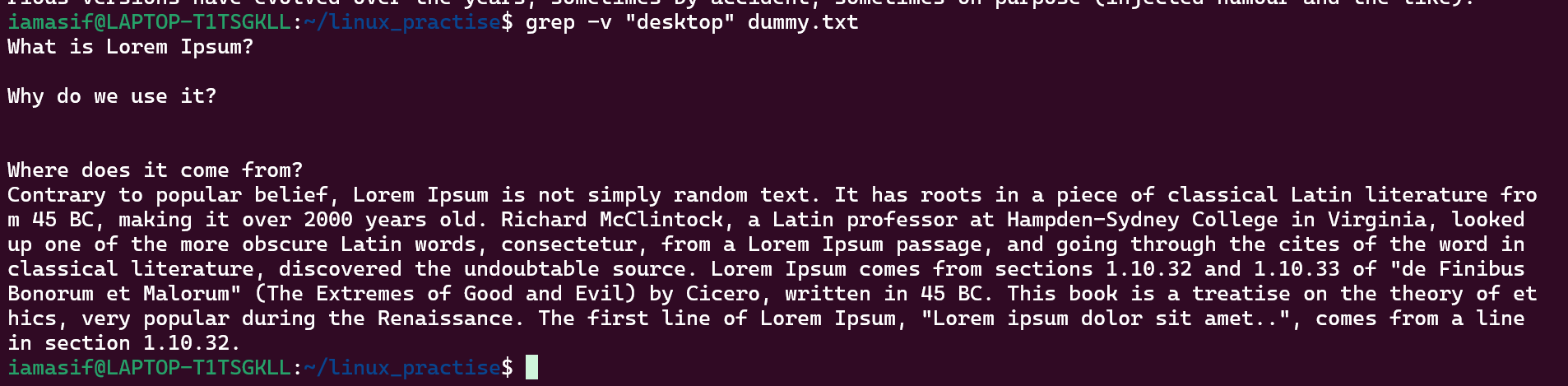
grep -n "unix" hellofile.txt



## **7. Inverting the Pattern Match Using grep**

You can display the lines that are not matched with the specified search string pattern using the -v option.

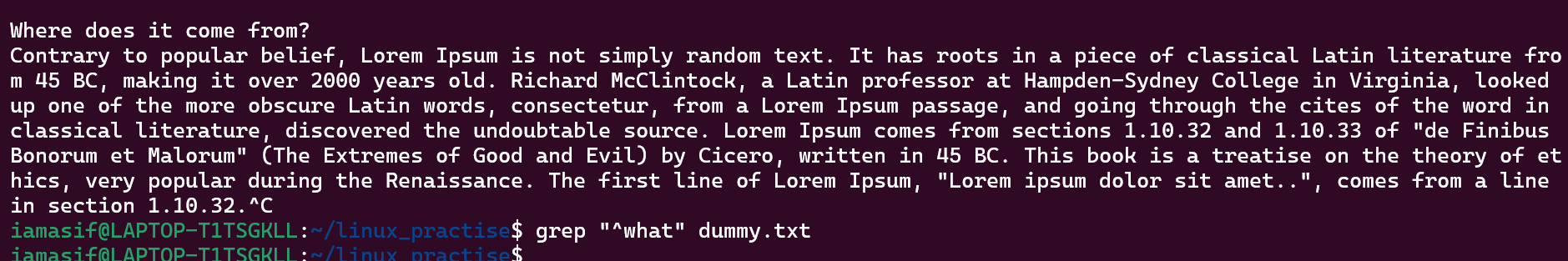
grep -v "unix" hellofile.txt



## **8. Matching the Lines that Start with a String Using grep**

The ^ regular expression pattern specifies the start of a line. This can be used in grep to match the lines which start with the given string or pattern.

grep "^unix" hellofile.txt



## 

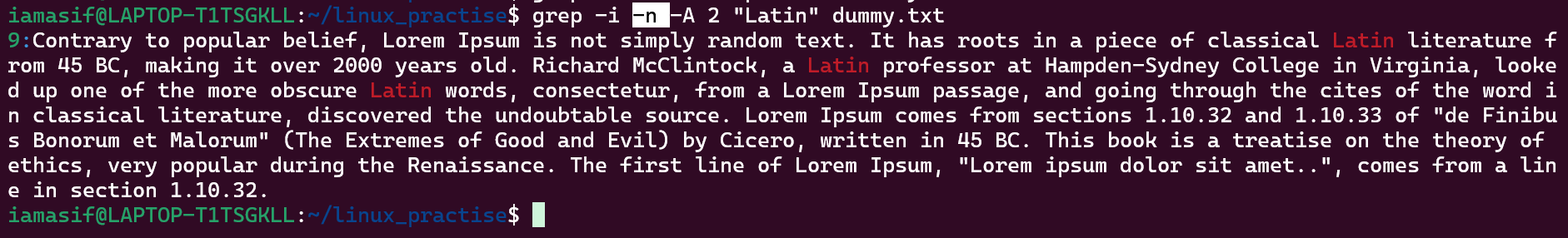
| **-c** |
| --- |
| **-h** |
| **-i** |
| **-l** |
| **-n** |
| **-v** |
| **-e exp** |
| **-f file** |
| **-E** |
| **-w** |
| **-o** |
| **-A n** |

## 

| **-B n** |
| --- |
| **-C n** |

### 

**MULTIPLE COMMANDS USAGE**

****

12.03 to 12.13

**Done with the commands**

Plz check if Doc 15 Linux awt commands doc is uploaded ..

**Done with the commands**

Good job guys its the initials of the creators AWT.. not Abstrat Windows Toolkit which is in Java

Plz remember.. If interviewer asks AWT full form .. plz answer according to the context..

Task 17:

**AWK commands in doc 15 Linux AWK commands..**

**Done with commands**

13.30 to 14.30 lunch break

10 done out of 29

14.31 to 14.36 to 14.45→ for AWK commands

This is the link to open ODs files in linux.. You need to download the compatibles

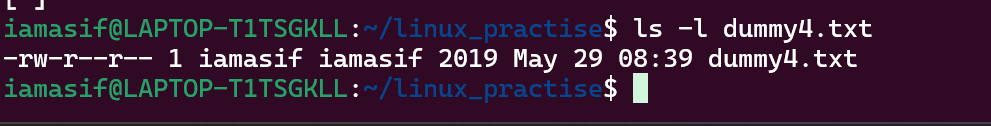
<https://www.adobe.com/uk/acrobat/resources/document-files/open-doc/ods.html>

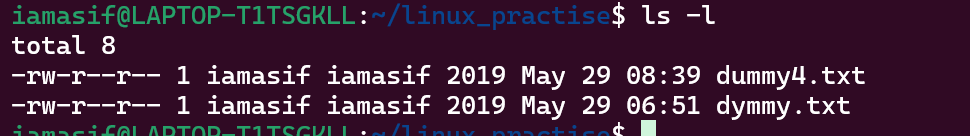
Task 18:

How to check file access permission in Linux?

Hint use:

Ls -l





Task 19:

**What are the default permissions for a new file ?**

When a new file is created in Linux, the default permissions are determined by:

* The system default (666 for files: read & write for everyone), and
* The user's umask (which subtracts permissions).

Plz find out for

Owner → ?

Group → ?

All and others → ?

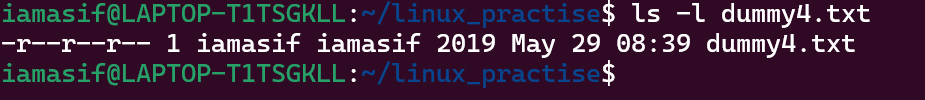
* Owner = rw- = can read and write
* Group = r- - = can read only
* Others = - - = can read only

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

HInt: chmod 444 filename



Task 21:

Can you change the file permissions to match the following:

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

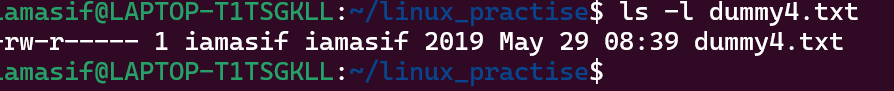
Yes! To set the file permissions as:

* **Owner**: Read and Write → rw-
* **Group**: Read only → r--
* **Others**: No permission → —

Task 22:

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

Hint : use chmod 640 filename



Task 23:

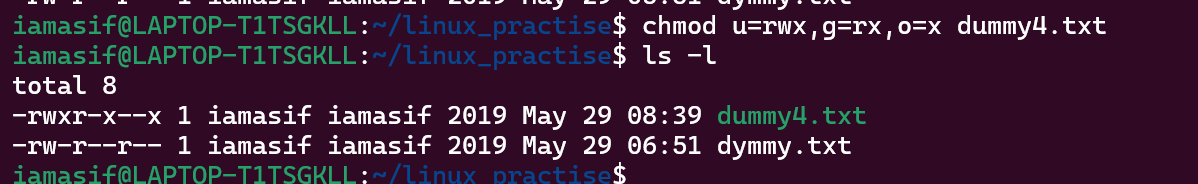
Change chmod.exercises permissions to -rwxr-x--x

Change the file permissions to match the following:

owner: Read, Write and Execute

group: Read and Execute

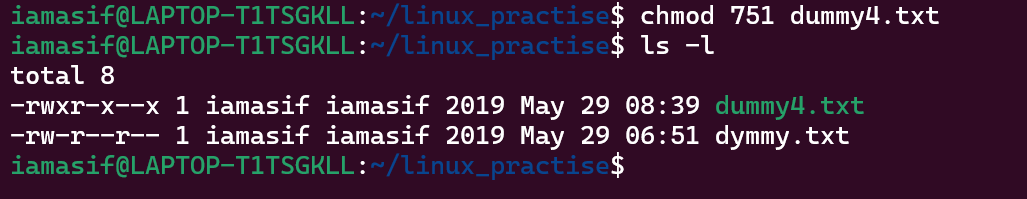
other: Execute



Task 24:

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

Hint : use chmod 751 filename



Task 25:

Guys what will this command do?

chown -c master file1.txt

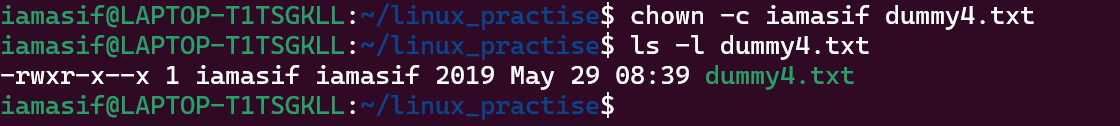
Changes the owner of file1.txt to master, and prints a message only if a change actually happens.

chown: Changes the ownership of a file.

-c: Verbose mode – it tells you what changed.

master: The new owner of the file.

file1.txt: The target file.



Task 26:

**Can you define what is a process**

A process is any program or command that is being executed by the operating system.

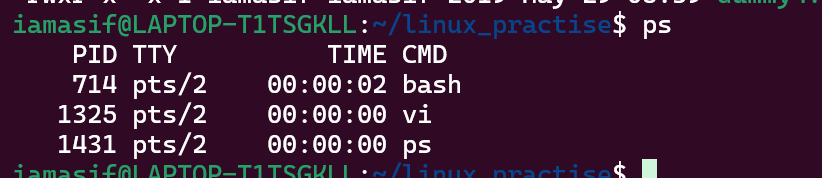
Task 27:

What is command to check foreground process and background process

Ps is the command

Task 28:

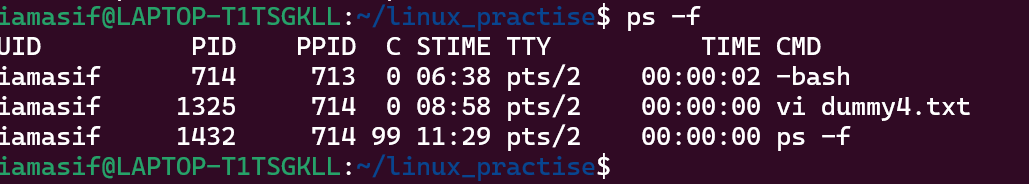
Can you list all the running processes?



Hint use ps

Task 29:

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



Task 30:

Can you createa a variable name with your name in it

Ex:

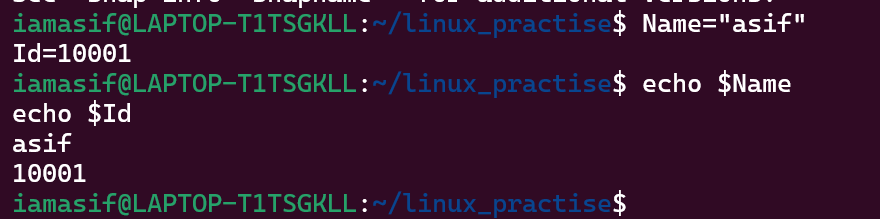
Name = “prasunamba”

Id = 10001

And check

Echo $Name

Chek the output



Task 31:

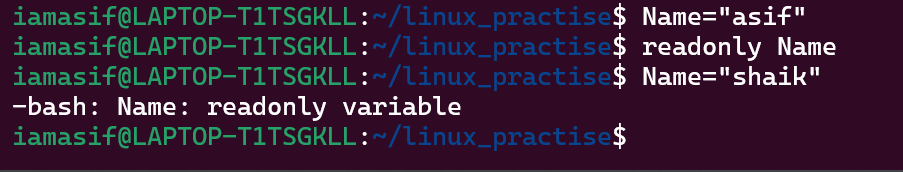
Can you make the above name variable read only..

Ex:

Name = “Prasunamba”

Readonly Name

Name = “Meher” —>what will this display.. Is it saying read only?? Pl check



Task 32:

Now will unset or delete the variables

Use the below command and check

Unset Name

Now check for

echo $Name —> this should not print anything.. Plz try also specify the reason

Read n Know that

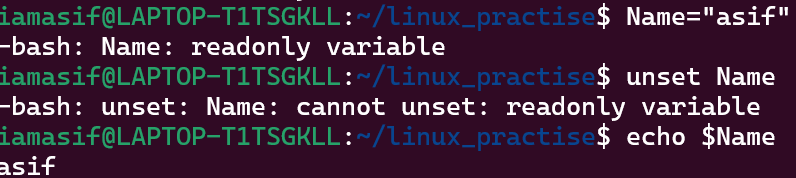
Variable Types

When a shell is running, three main types of variables are present −

Local Variables − A local variable is a variable that is present within the current instance of the shell. It is not available to programs that are started by the shell. They are set at the command prompt.

Environment Variables − An environment variable is available to any child process of the shell. Some programs need environment variables in order to function correctly. Usually, a shell script defines only those environment variables that are needed by the programs that it runs.

Shell Variables − A shell variable is a special variable that is set by the shell and is required by the shell in order to function correctly. Some of these variables are environment variables whereas others are local variables.



Task 33:

CAn u try to add a list of your friends names in an array and try to printout

Ex:

NAME[0]="Ram"

NAME[1]="Sita"

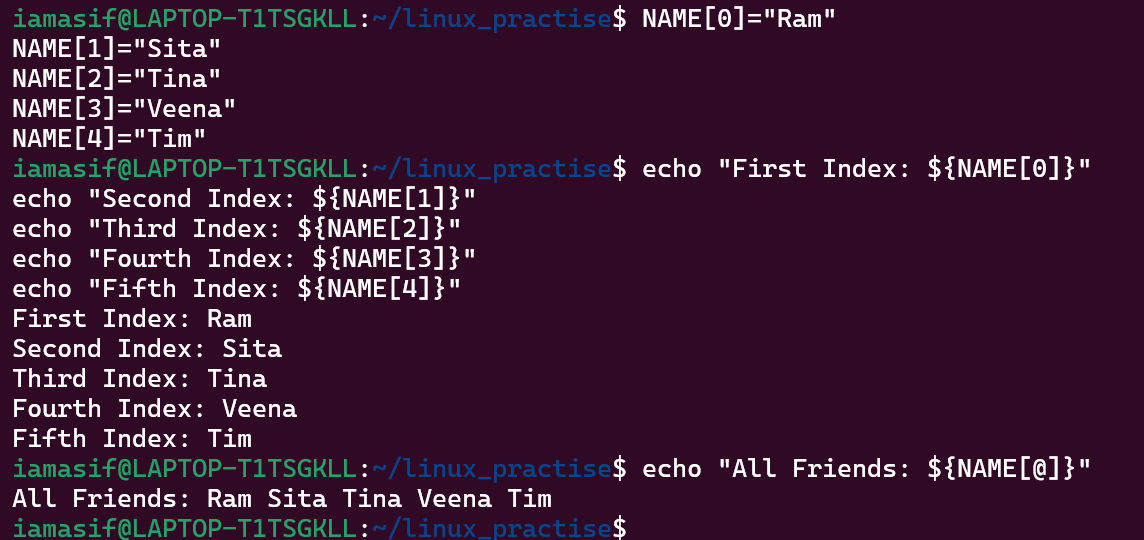
NAME[2]="Tina"

NAME[3]="Veena"

NAME[4]="Tim"

echo "First Index: ${NAME[0]}"

echo "Second Index: ${NAME[1]}"

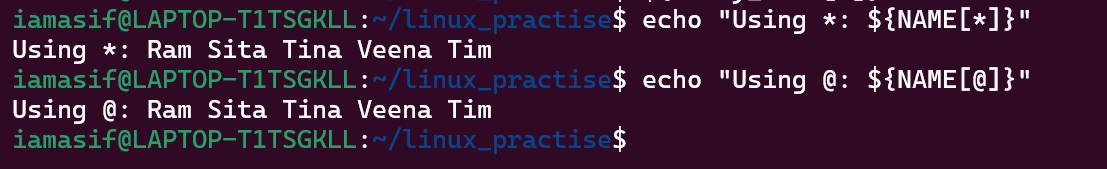


Task 34:

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

Echo “${array\_name[\*]}”

Echo “${array\_name[@]}”



Opetrators 👍

* Arithmetic Operators
* Relational Operators
* Boolean Operators
* String Operators
* File Test Operators

If else

if...fi statement

if...else...fi statement

if...elif...else...fi statement

case...esac statement

The while loop

The for loop

The until loop

The select loop

Plz have an idea about the above

**Done**

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 "

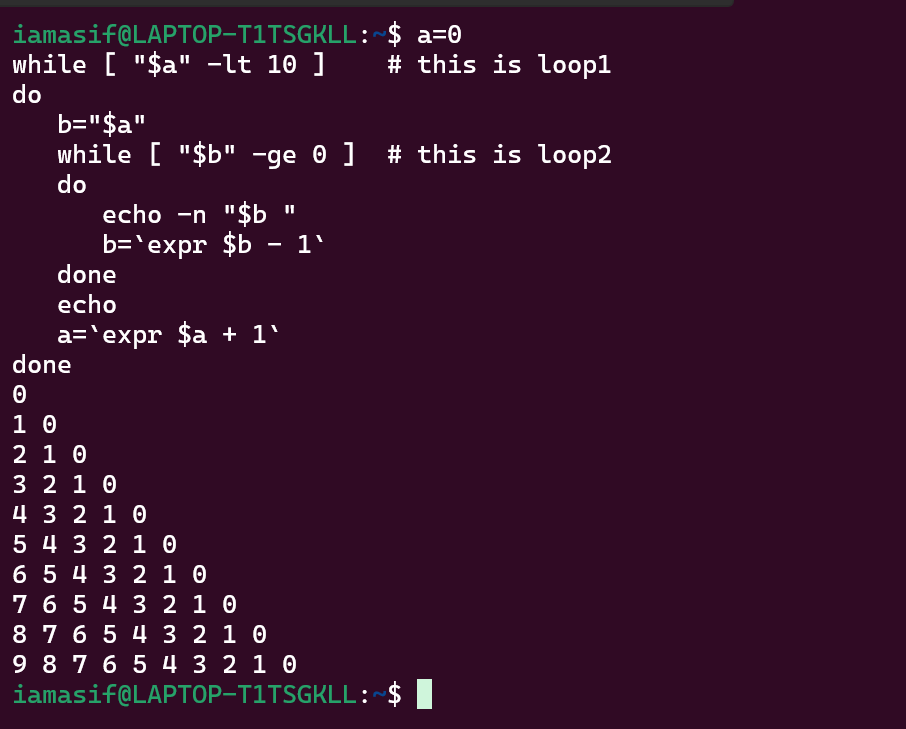
b=`expr $b - 1`

done

echo

a=`expr $a + 1`

Done

****

## Stopping Processes

Ending a process can be done in several different ways. Often, from a console-based command, sending a CTRL &plus; C keystroke (the default interrupt character) will exit the command. This works when the process is running in the foreground mode.

If a process is running in the background, you should get its Job ID using the **ps** command. After that, you can use the **kill** command to kill the process as follows −

$ps -f

UID PID PPID C STIME TTY TIME CMD

amrood 6738 3662 0 10:23:03 pts/6 0:00 first\_one

amrood 6739 3662 0 10:22:54 pts/6 0:00 second\_one

amrood 3662 3657 0 08:10:53 pts/6 0:00 -ksh

amrood 6892 3662 4 10:51:50 pts/6 0:00 ps -f

$kill 6738

Terminated

Here, the **kill** command terminates the **first\_one** process. If a process ignores a regular kill command, you can use **kill -9** followed by the process ID as follows −

$kill -9 6738

Terminated

## Parent and Child Processes

Each unix process has two ID numbers assigned to it: The Process ID (pid) and the Parent process ID (ppid). Each user process in the system has a parent process.

Most of the commands that you run have the shell as their parent. Check the **ps -f** example where this command listed both the process ID and the parent process ID.

## Zombie and Orphan Processes

Normally, when a child process is killed, the parent process is updated via a **SIGCHLD** signal. Then the parent can do some other task or restart a new child as needed. However, sometimes the parent process is killed before its child is killed. In this case, the "parent of all processes," the **init** process, becomes the new PPID (parent process ID). In some cases, these processes are called orphan processes.

When a process is killed, a **ps** listing may still show the process with a **Z** state. This is a zombie or defunct process. The process is dead and not being used. These processes are different from the orphan processes. They have completed execution but still find an entry in the process table.

## Daemon Processes

Daemons are system-related background processes that often run with the permissions of root and services requests from other processes.

A daemon has no controlling terminal. It cannot open **/dev/tty**. If you do a **"ps -ef"** and look at the **tty** field, all daemons will have a **?** for the **tty**.

To be precise, a daemon is a process that runs in the background, usually waiting for something to happen that it is capable of working with. For example, a printer daemon waiting for print commands.

If you have a program that calls for lengthy processing, then its worth to make it a daemon and run it in the background.

## The top Command

The **top** command is a very useful tool for quickly showing processes sorted by various criteria.

It is an interactive diagnostic tool that updates frequently and shows information about physical and virtual memory, CPU usage, load averages, and your busy processes.

Here is the simple syntax to run top command and to see the statistics of CPU utilization by different processes −

$top

## Job ID Versus Process ID

Background and suspended processes are usually manipulated via **job number (job ID)**. This number is different from the process ID and is used because it is shorter.

In addition, a job can consist of multiple processes running in a series or at the same time, in parallel. Using the job ID is easier than tracking individual processes.