**Task 1:**

**RegEX Symbols in linux List them down with description**

**Regular expressions**

In Linux, **regular expressions (regex)** are powerful patterns used to search, match, or manipulate text. They are commonly used with tools like grep, sed, awk, and find. Here's a summary of **regular expression symbols and usage in Linux.They’re powerful for searching, filtering, extracting, and manipulating data in shell commands and scripts.**

| **Symbol** | **Description** | **Example** |
| --- | --- | --- |
| **.** | **Any single character (except newline)** | **gr.y matches gray, grey** |
| **^** | **Start of line** | **^Hello matches lines starting with Hello** |
| **$** | **End of line** | **end$ matches lines ending with end** |
| **\*** | **Zero or more of the previous character** | **lo\* matches l, lo, loo** |
| **[]** | **Any one character in the set** | **[aeiou] matches any vowel** |
| **[^]** | **Any one character not in the set** | **[^0-9] matches non-digits** |
| **\** | **Escape special characters** | **\. matches literal dot .** |
| **\{n\}** | **Exactly n repetitions** | **a\{3\} matches aaa** |
| **\{n,\}** | **n or more repetitions** | **a\{2,\} matches aa, aaa,...** |
| **\{n,m\}** | **Between n and m repetitions** | **a\{1,3\} matches a, aa, aaa** |
| **\(...\)** | **Grouping (for sed, not grep)** | **\(ab\)\* matches repeated ab** |
| **+** | **One or more of the previous character** | **lo+ matches lo, loo, etc.** |
| **?** | **Zero or one of the previous character** | **colou?r matches color or colour** |
| **`** | **`** | **Logical OR** |
| **()** | **Group expressions** | **(ab)+ matches ab, abab** |
| **{n}** | **Exactly n repetitions** | **a{3} matches aaa** |
| **{n,}** | **n or more repetitions** | **a{2,} matches aa, aaa, ...** |
| **{n,m}** | **Between n and m repetitions** | **a{1,3} matches a, aa, aaa** |

| **Command** | **Description** |
| --- | --- |
| **grep "^A" file.txt** | **Lines starting with A** |
| **grep "ed$" file.txt** | **Lines ending in ed** |
| **grep "[0-9]" file.txt** | **Lines with any digit** |
| **`grep -E "foo** | **bar"` file.txt** |
| **grep -E "a{2,4}" file.txt** | **Matches aa, aaa, or aaaa** |

**Task 2**

**If you are aware of Linux OS.. can you tell me the feature of Linux.**

Linux is an open-source operating system, meaning its source code is freely available for anyone to view, modify, and distribute. One of its standout features is multitasking, which allows it to run multiple processes at the same time efficiently. It also supports multiuser functionality, enabling several users to access the system simultaneously without conflict. Security is another strong point, with Linux offering robust file permissions, encryption, and built-in security modules like SELinux to protect against threats. Finally, Linux is known for its stability and reliability

**Task 3:**

**What is Kernal ? can you explain about it in your words..**

The **kernel** is the core part of the operating system that directly interacts with hardware.It manages system resources like CPU, memory, and devices.Acts as a bridge between user applications and hardware.Handles processes, memory, device control, and file systems.Without the kernel, the operating system can't function properly.

**Task 4:**

**BASH in Lonux full form and Explanation**

**BASH** stands for **Bourne Again SHell**.

BASH is a popular command-line **shell** and scripting language used in Linux and other Unix-like operating systems. It acts as an interface between the user and the operating system, allowing users to execute commands, run scripts, and automate tasks. BASH extends the original Bourne shell (sh) with additional features like command history, tab completion, improved scripting syntax, and more powerful programming constructs. It is the default shell on most Linux distributions and widely used for system administration and development.

Task 5:

Now that you know Linux is also an Operating System like Windows..

What do you think is the difference between LInux and Windows

1. **Source Code**
   * **Linux** is **open-source**, meaning anyone can view, modify, and distribute its code freely.
   * **Windows** is **proprietary** software developed by Microsoft, with its source code closed to the public.
2. **Cost**
   * **Linux** is generally **free** to use and distribute.
   * **Windows** requires a **paid license** for use.
3. **User Interface**
   * **Windows** offers a user-friendly graphical interface by default, designed for ease of use.
   * **Linux** has multiple desktop environments (like GNOME, KDE), which can be customized but might be less intuitive for beginners.
4. **Software Compatibility**
   * **Windows** supports a wide range of commercial software and games.
   * **Linux** has many free and open-source applications but fewer commercial programs; some Windows apps require compatibility layers like Wine.
5. **Security and Stability**
   * **Linux** is often considered more secure and stable, widely used for servers and critical systems.
   * **Windows** is more prone to viruses and requires regular antivirus protection but is popular for desktops and business applications.

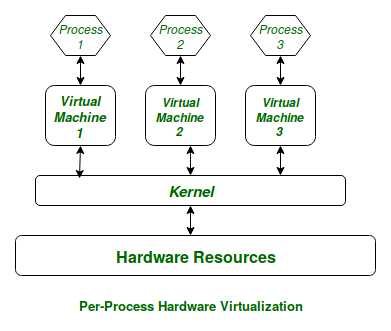
Task 6:

What are the basic components of Linux? Describe each in detail with diagrams.

## **Basic Components of Linux**

Linux OS is composed of several core components that work together to provide the operating system’s functionality:

### **1. Kernel**

* The kernel is the core of the Linux OS.
* It manages system resources like CPU, memory, and device communication.
* It acts as a bridge between hardware and user applications.
* Types of kernels: Monolithic (Linux is monolithic) which means all core functions run in a single large process.
* Responsibilities include process management, memory management, device drivers, and system calls.

### 

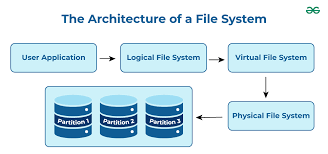
### 

### **2. Shell**

* The shell is the command-line interpreter that takes commands from the user and passes them to the kernel.
* Common Linux shells include **Bash**, **Zsh**, and **Ksh**.
* It also supports scripting to automate tasks.

### **3. File System**

* Linux uses a hierarchical file system structure starting from the root directory /.
* It organizes files and directories and manages data storage on physical devices.
* Supports multiple file system types (ext4, XFS, Btrfs, etc.).
* Everything in Linux is treated as a file, including devices.



### **4. System Libraries**

* These are special programs or functions used by applications to interact with the kernel.
* They provide APIs (Application Programming Interfaces) that simplify complex kernel operations.
* Examples include **glibc**, the GNU C Library, which many Linux programs rely on.

### **5. User Space/Applications**

* This layer includes all user-level programs and applications such as editors, web browsers, and utilities.
* These run outside the kernel space and interact with the kernel via system calls.
* Examples: vim, firefox, gcc, and desktop environments.

**Task 7:**

**Is it legal to edit Kernal ? when do you think we have to in case?**

**Yes, it is legal to edit the Linux kernel** because Linux is released under the **GNU General Public License (GPL) v2**, which explicitly allows anyone to **view, modify, and redistribute** the source code — as long as you follow the license rules.

### **When do we edit the kernel?**

You might need to edit or customize the Linux kernel in cases like:

* **Adding support for new hardware devices** that are not yet supported by the existing kernel.
* **Improving performance** for specific workloads or systems.
* **Fixing bugs or security vulnerabilities** that affect your environment.
* **Adding new features or functionalities** tailored to your needs.
* **Customizing kernel behavior** for embedded systems or specialized devices.
* **Learning and experimentation** if you want to understand OS internals or contribute to Linux development.

**Task 8:**

**What is LILO? Explain**

1. **LILO** stands for **LInux LOader**. It is a **boot loader** for Linux operating systems.
2. LILO is a small program that runs when a computer starts up.
3. Its main job is to load the Linux kernel into memory and start the operating system.
4. It is installed in the **Master Boot Record (MBR)** or a partition’s boot sector.
5. LILO can boot multiple operating systems (Linux, Windows, etc.) by letting the user select which one to start.
6. Although widely used in the past, LILO has largely been replaced by more advanced boot loaders like **GRUB**.

**Task 9:**

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

A **shell** in Linux is a **command-line interface** (CLI) that allows users to interact with the operating system by typing commands.  
 It acts as a **bridge between the user and the Linux kernel** — it takes commands from the user, sends them to the kernel, and displays the output.

Shells can also run **scripts** — sequences of commands saved in files — to automate tasks.

| **Shell** | **Full Form** | **Description** |
| --- | --- | --- |
| **Bash** | Bourne Again Shell | Most common shell; powerful scripting features; default in most Linux distros. |
| **Sh** | Bourne Shell | Original Unix shell; simple and widely compatible. |
| **Csh** | C Shell | Syntax similar to the C programming language; supports job control. |
| **Tcsh** | TENEX C Shell | Enhanced version of Csh with auto-completion and command editing. |
| **Ksh** | Korn Shell | Combines features of sh and csh; good for scripting and programming. |
| **Zsh** | Z Shell | Highly customizable, powerful, with plugins and themes support. |
| **Fish** | Friendly Interactive Shell | User-friendly, modern, with features like autosuggestions and syntax highlighting. |

**Task 10:**

**What is swap space?**

**Swap space** is a portion of disk storage used by the Linux operating system as **virtual memory** when the **physical RAM is full**.

* When your system runs out of RAM (Random Access Memory), it moves inactive data from RAM to the **swap space** to free up memory.
* Swap helps prevent system crashes by ensuring that processes still have memory to run.
* It's slower than RAM because it uses the **hard disk** or **SSD**, but it's useful as a backup.
* Swap can exist as a **swap partition** or a **swap file**.
* It’s especially important on systems with **low RAM** or for handling **large applications**.

**Task 11:**

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

In Linux, **mounting** means **making a storage device (like a hard disk, USB, or CD-ROM) accessible** to the system by attaching it to a directory in the Linux file system tree (usually under /mnt or /media).

Once mounted, you can read/write files on that device just like any other directory.

**How to Mount a File System: syntax**

sudo mount /dev/sdb1 /mnt

**How to Unmount a File System:syntax**

sudo umount /mnt

**Task 12:**

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

**chmod** stands for **"change mode"**.

It is used in **Linux/Unix** to **change the permissions** (read, write, execute) of files or directories for the **owner**, **group**, and **others**.

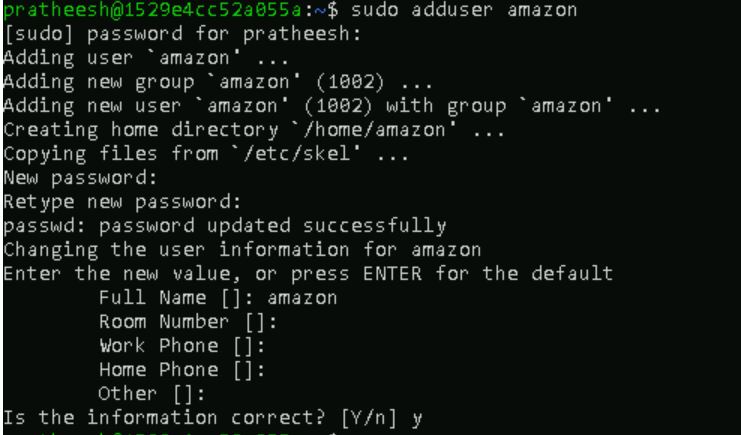
Syntax

chmod 755 file.txt

| **Symbol** | **Meaning** |
| --- | --- |
| r | Read |
| w | Write |
| x | Execute (or open for folders) |
| **Category** | **Who it is** |
| u | User (owner) |
| g | Group |
| o | Others |
| a | All (user + group + others) |

**Task 13:**

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

****

## **Method 1: Using adduser Command (Simpler & Interactive)**

sudo adduser newusername

## **Method 2: Using useradd Command (More Manual)**

sudo useradd -m -s /bin/bash newusername

sudo passwd newusername

**Explanation:**

* -m creates the home directory.
* -s sets the default shell.
* passwd sets the password for the new user.

## **Method 3: Creating User with Custom Home Directory and Shell**

sudo useradd -m -d /custom/home/path -s /bin/zsh newusername

sudo passwd newusername

### **Verify the User**

id newusername

To check how many user

wc -l /etc/passwd

Task 14:

Can you change the password of a user?

How do you do that? Plz share ss

### **For your own account:**

passwd

You’ll be prompted to:

1. Enter your current password.
2. Enter a new password.
3. Confirm the new password.

### **To change another user's password (requires sudo):**

sudo passwd username

**Example:**

sudo passwd john

Task 15:

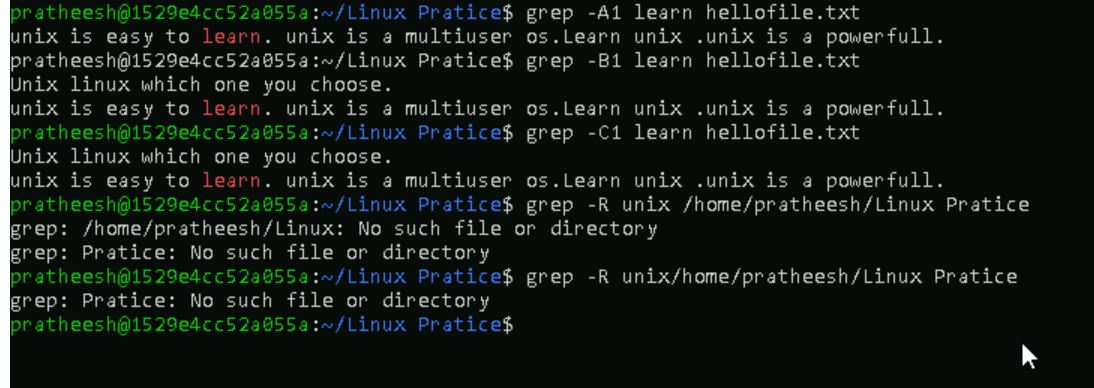
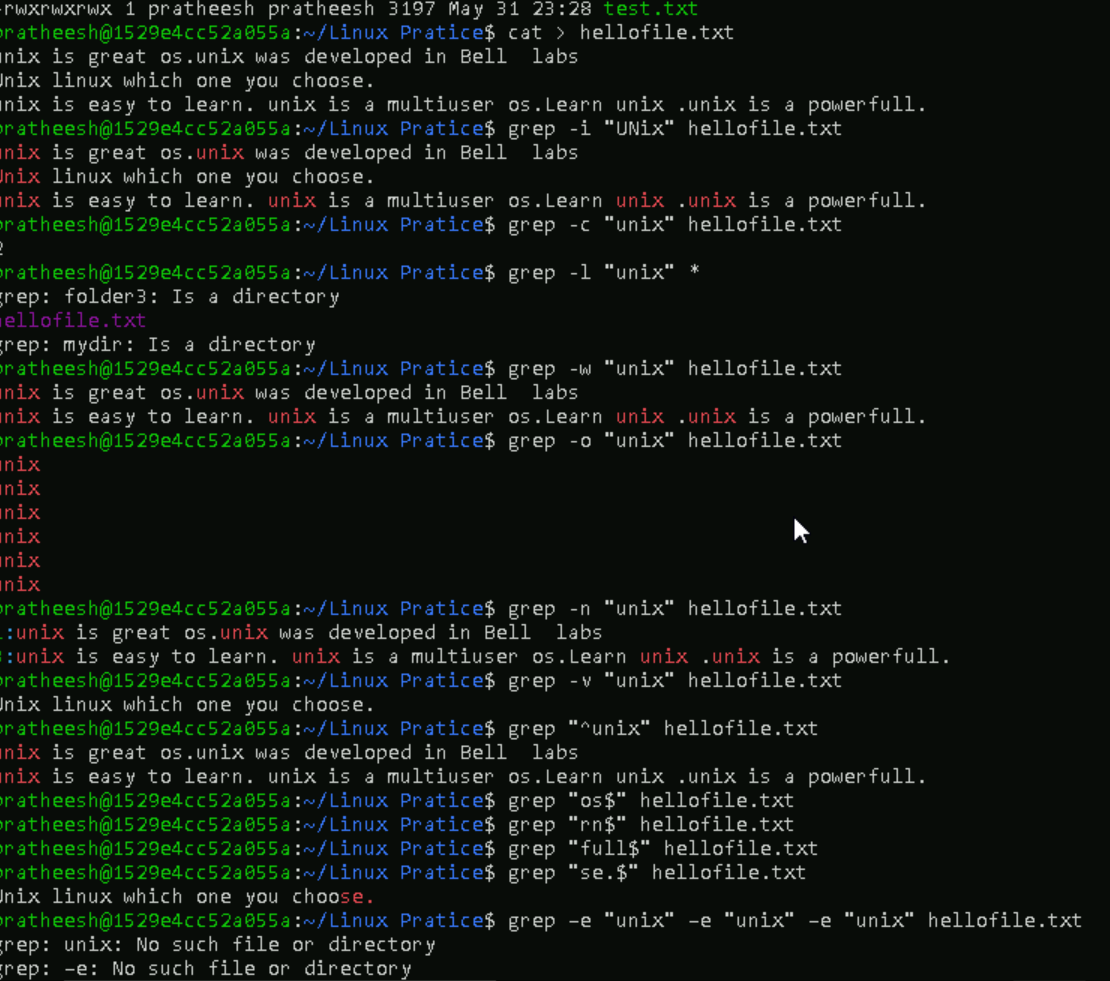
What is diff between Process and Thread?

| **Feature** | **Process** | **Thread** |
| --- | --- | --- |
| **Definition** | A program in execution. | A lightweight unit of a process. |
| **Memory** | Each process has its own memory space. | Threads share the same memory space. |
| **Isolation** | Processes are independent; crashing one doesn't affect others. | Threads are interdependent; error in one can affect others. |
| **Communication** | Inter-process communication (IPC) is complex. | Thread communication is easier (shared memory). |
| **Creation Time** | More time and system resources to create. | Faster to create and manage. |
| **Examples** | Opening Chrome and VLC as separate apps. | Opening multiple tabs in Chrome (each tab = thread). |

Task 16:

Doc 14 Linux Grep commands in docs to study folder .. plz work on it..

<https://docs.google.com/document/d/1ZV5l14lKa6SZc8IJD-HxogqMmTuRzMOV/edit?tab=t.0>

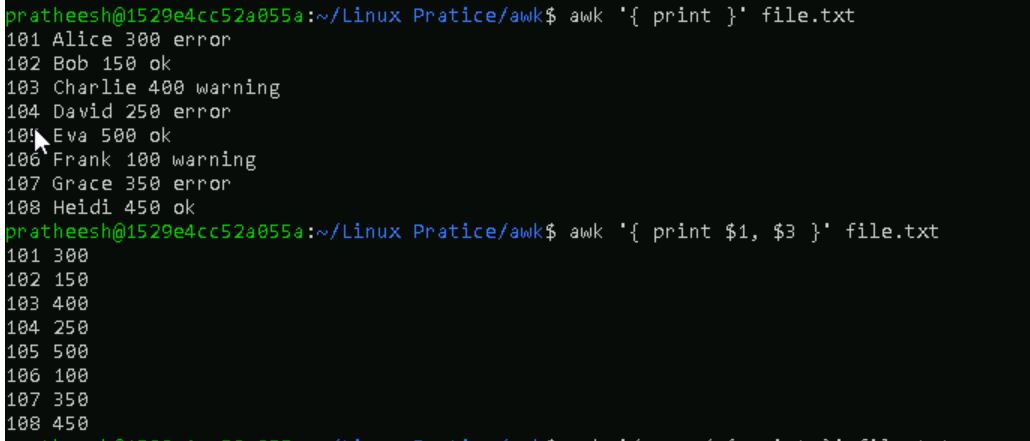


Task 17:

Task 17 AWK commands in doc 15 Linux AWK commands.

Example 1: Print Contents of a File

Example 2: Print Specific Columns of a File

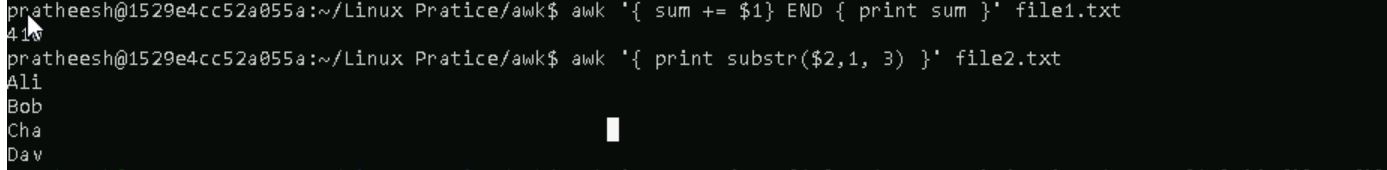


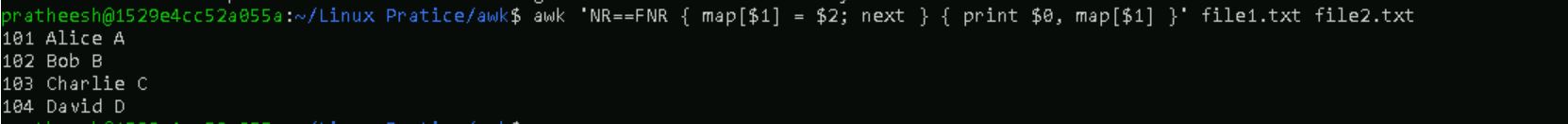
Example 3: Filter Lines Based on a Condition

Example 4: Sum Values in a Column



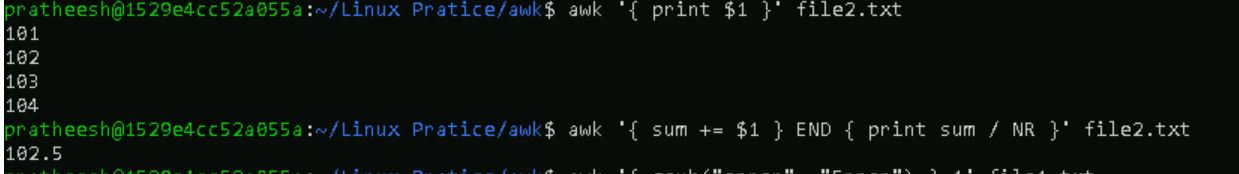
Example 5: Extract Substring from a Column



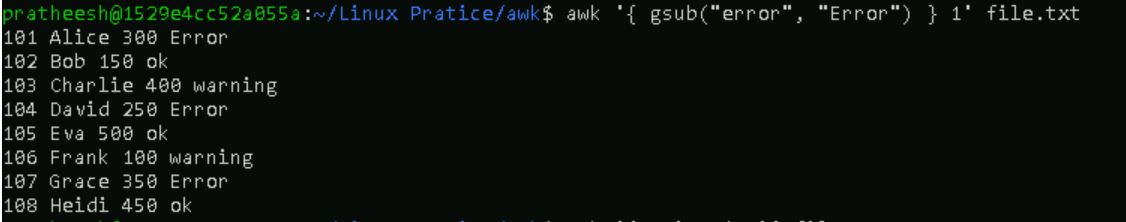
Example 6: Join Two Files Based on a Common Columnmn

Example 7: Extract First Column of a File

Example 8: Calculate Average of a Column



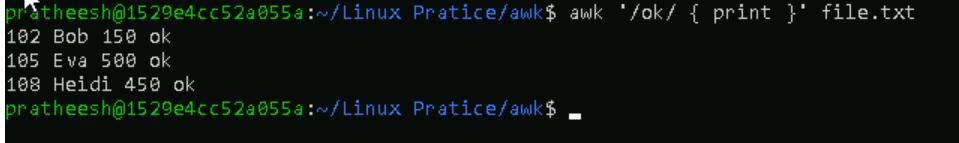
Example 9: Replace a String in a File



Example 10: Display Last Field of a File



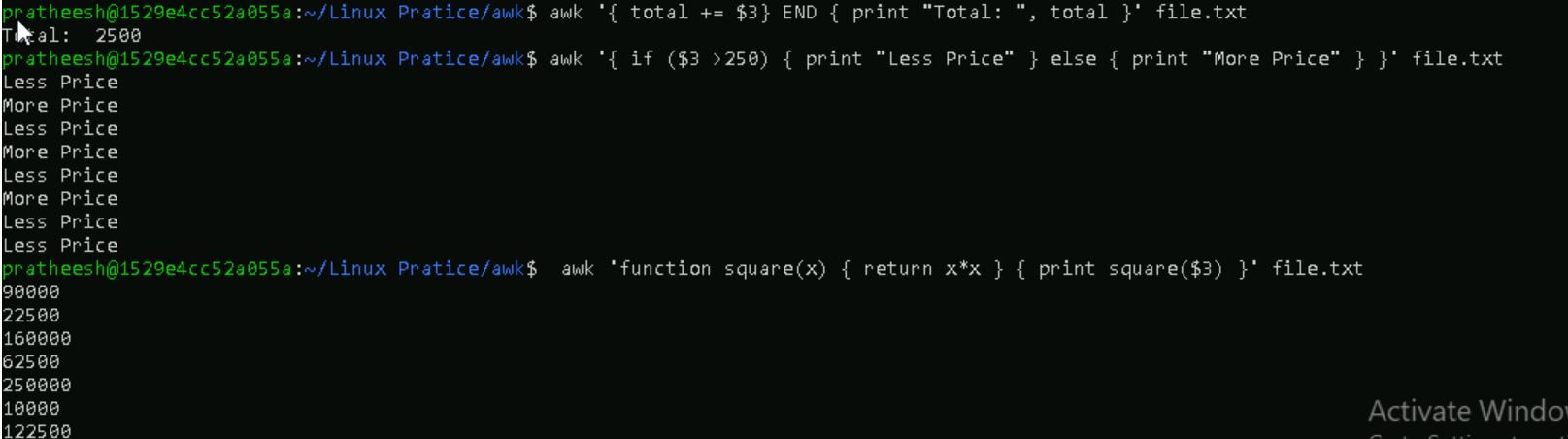
Example 11: Using Regular Expressions



Example 12: Using Variables

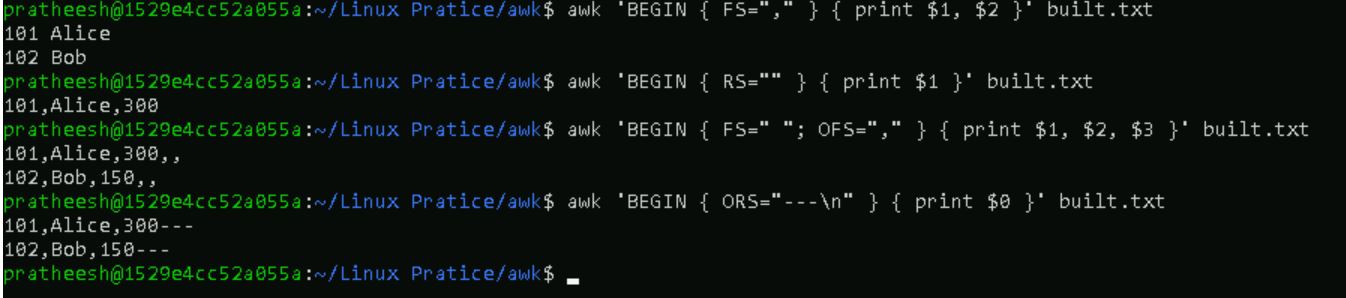
Example 13: Using Control Statements

Example 14: Using Functions



Example 15: Using Built-in Variables

* **FS** ? field separator variable is used to specify delimiter used to separate fields in input file. By default, it is set to whitespace.
* **RS** ? record separator variable is used to specify delimiter used to separate records in input file. By default, it is set to a newline character.
* **OFS** ? output field separator variable is used to specify delimiter used to separate fields in output. By default, it is set to whitespace.
* **ORS** ? output record separator variable is used to specify delimiter used to separate records in output. By default, it is set to a newline character.



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 ?

Plz find out for

Owner → rw- → read, write

Group → r-- → read only

All and others → r-- → read only

Using ls -l

-rw-r--r-- 1 prathesh pratheesh 30 May 1 12:34 testfile.txt

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)

Task 22:

What was the command for changing the 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

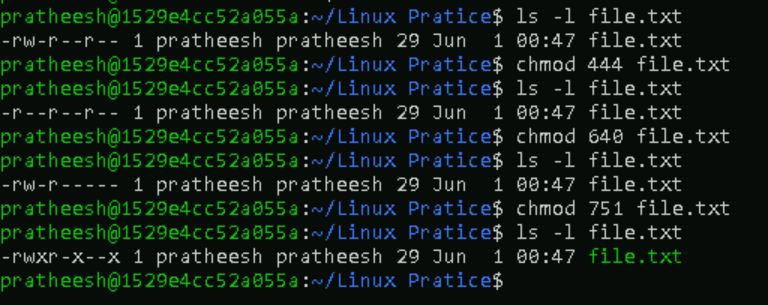
group: Read and Execute

other: Execute

Task 24:

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

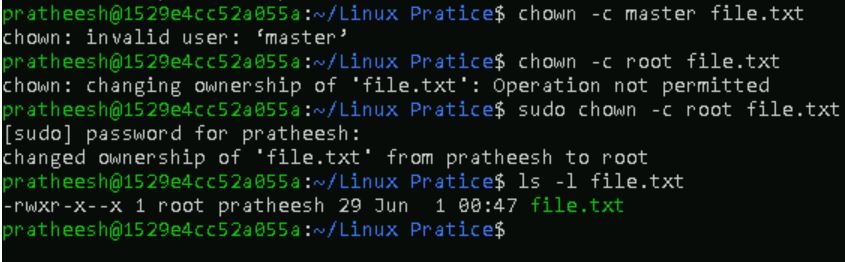
Hint : use chmod 751 filename



ask 25:

Guys what will this command do?

chown -c master file1.txt



Task 26:

Can you define what is a process

A **process** is an instance of a program that is currently running on a computer. It includes the program’s code, its current activity, and the system resources (like CPU time, memory, files) it uses.

* When you run a program, the operating system creates a **process** to execute it.
* A process has a **unique Process ID (PID)**.
* It contains its own **memory space** and **execution context**.
* Multiple processes can run the same program independently.
* Processes can create **child processes**.

Task 27:

What is command to check foreground process and background process**Check Background Processes**

Use: jobs

**Send Process to Background**

Use : bg

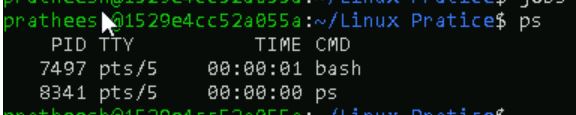
**See All Running Processes**

Use : ps aux

Task 28:

Can you list all the running processes?

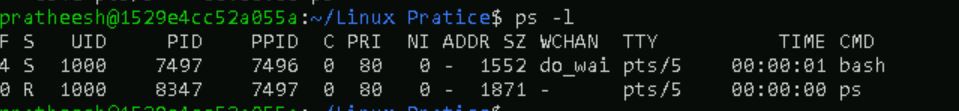
Hint use ps



Task 29:

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

check the status of active processes on a system, as well as display technical information about the processes



Task 30:

Can you create a variable name with your name in it

Ex:

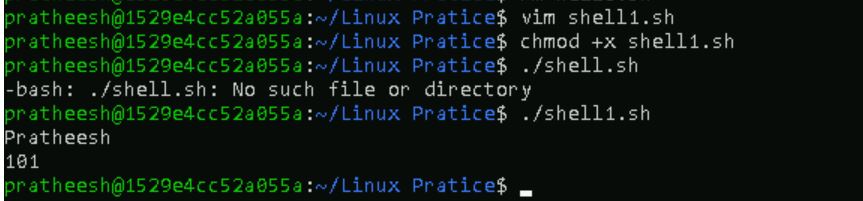
Name = “prasunamba”

Id = 10001

And check

Echo $Name

Chek the output



Task 31:

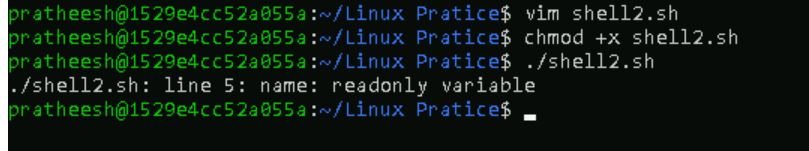
Can you make the above name variable read only..

Ex:

Name = “Prasunamba”

Readonly Name

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



Task 32:

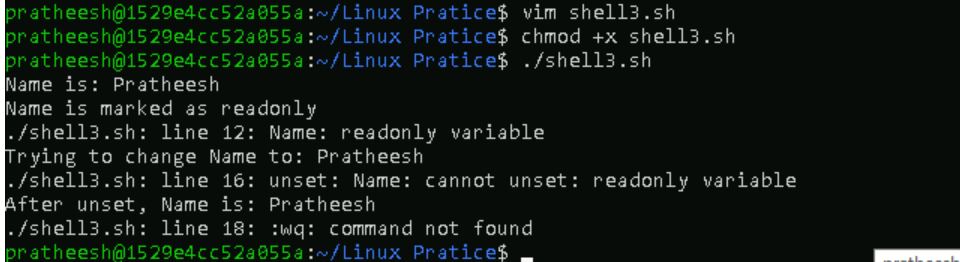
Now will unset or delete the variables

Use the below command and check

Unset Name

Now check for

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



Task 33:

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

Ex:

NAME[0]="Ram"

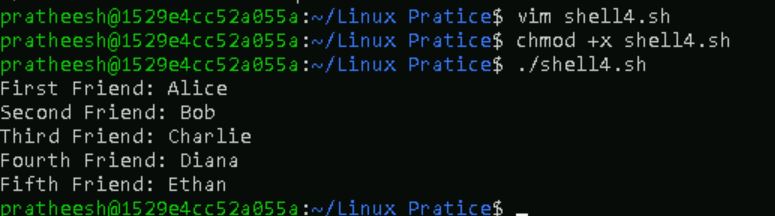
NAME[1]="Sita"

NAME[2]="Tina"

NAME[3]="Veena"

NAME[4]="Tim"

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

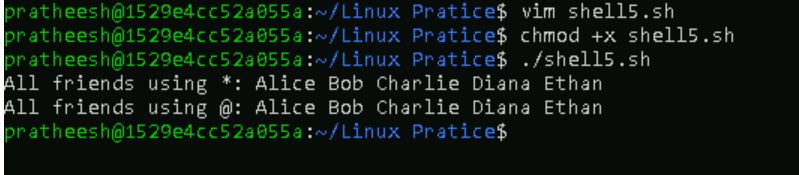


Task 34:

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

Echo “${array\_name[\*]}”

Echo “${array\_name[@]}”



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

