List of regular expression :

[...]: Accept ANY ONE of the character within the square bracket

+: one or more

\*: zero or more

. (dot): ANY ONE character except newline

 $: start-of-line and end-of-line respectively

Task 1

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**

|  |  |  |
| --- | --- | --- |
| **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 known for its open-source nature, allowing for free use, modification, and distribution. It also boasts strong security features, a robust file system, and support for various hardware and software.

Task 3

What is kernal ?

The kernel is the core part of an operating system (like Windows, Linux, or Android). It works like a bridge between your computer’s hardware (like CPU, RAM, etc.) and the software (apps, programs).

Imagine your computer is like a factory.

• The kernel is the manager of the factory.

• It gives orders to the workers (hardware) based on what the users or software (you or your apps) need.

It handles things like

• Managing files and storage

• Handling memory (RAM)

• Talking to devices like keyboards, screens, or printers

• Scheduling tasks and running programs

Why is it important?

Without the kernel, your computer wouldn’t know how to use the hardware. It makes sure everything runs smoothly and safely.

Task 4

What is BASH?

Bourne Again Shell

BASH is a command-line shell used in Linux and other Unix-based systems.

It lets you type commands to do things like:

• Create or delete files

• Run programs

• Manage system settings

Why is it called “Bourne Again Shell”?

• It’s a new version of an older shell called “Bourne Shell” (sh).

• So, it’s a pun: “Bourne Again” → like being “born again”

In simple we can say

BASH is like chatting with your computer using text. Instead of clicking buttons, you type commands to tell it what to do.

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?

In short:

• Use Windows if you want easy setup, games, and popular apps.

• Use Linux if you want more control, better security, or you’re learning programming.

1. Ownership:

• Windows is made by Microsoft (paid software ).

• Linux is open-source (mostly free ), made by many people around the world.

2. Interface:

• Windows: Easy to use, with a graphical interface (lots of buttons and icons).

• Linux: Has a graphical interface too, but many people use the command line (terminal) more often.

3. Customization:

• Windows: You can change settings, but not the core system.

• Linux: You can customize almost everything, even how it looks and works inside.

4. Security:

• Windows: More likely to get viruses (needs antivirus).

• Linux: Safer by design, less chance of viruses.

5. Software:

• Windows: Supports popular software like MS Office, games, etc.

• Linux: Has its own software; some Windows apps may not work easily (but there are alternatives).

6. Used By:

• Windows: Common for personal use (home, office, gaming).

• Linux: Common for programmers, servers, and developers.

Task 6 : What are the basic components of Linux? Describe each in detail with diagrams.?

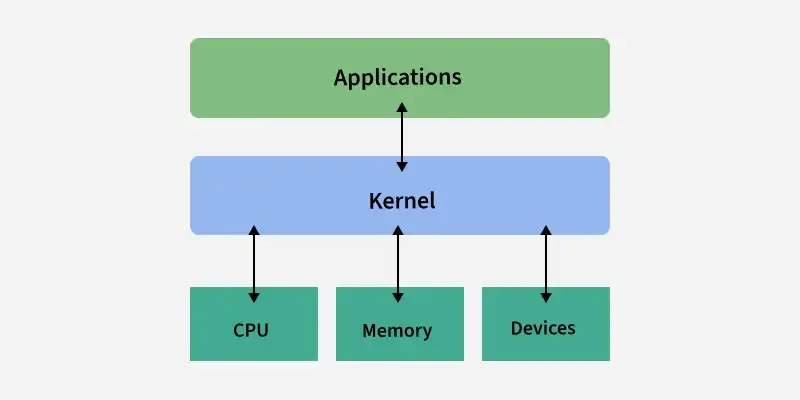
Basic Components of Linux:

1. Kernel

The core part of Linux. It connects software and hardware.

• Manages CPU, memory, devices, files.

• Handles multitasking and system calls.



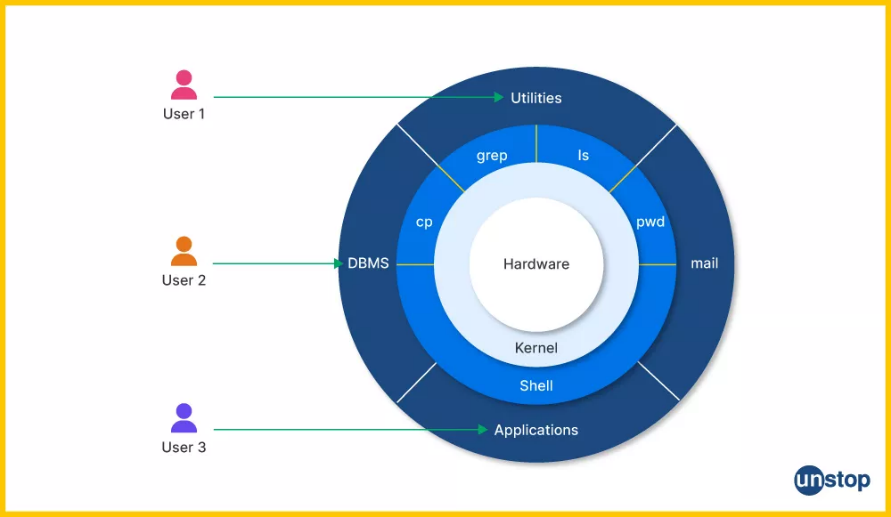
2. Shell

The interface between the user and the kernel.

• Accepts commands from the user.

• Sends them to the kernel to execute.

• BASH is a common shell.

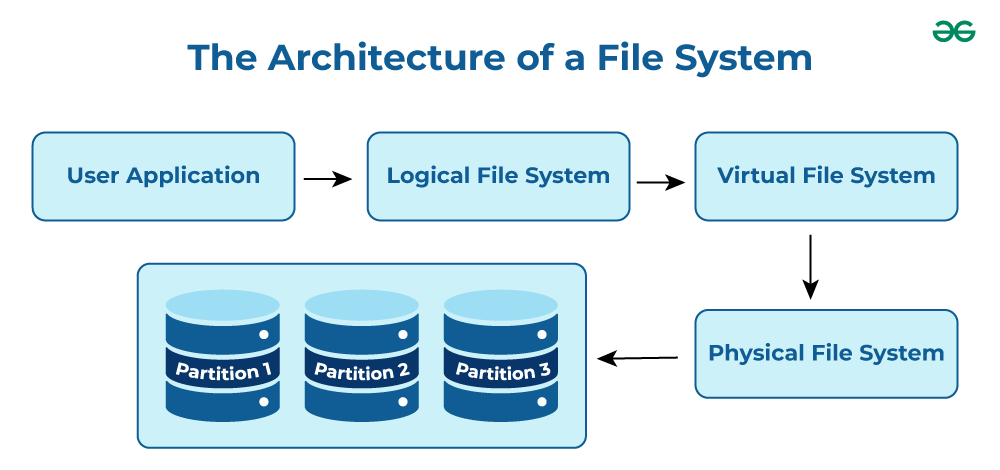


3. File System

Organizes and stores data.

• Everything in Linux is a file (even devices).

• It has a tree-like structure starting from / (root).



4. System Libraries

Collections of functions and instructions that programs can use.

• Help software interact with the kernel without directly accessing hardware.

• Example: glibc (GNU C Library).

5. System Utilities

Basic tools and programs that help users manage the system.

• Examples: ls, cp, mv, top, etc.

• Used to handle files, monitor processes, and more.

6. User Interface

Can be:

• Command-Line Interface (CLI) – typing commands (used in terminals).

• Graphical User Interface (GUI) – with windows, icons, buttons (like GNOME or KDE).

Diagram of Linux Components:

Here’s a simple diagram to show how these parts are connected:

+------------------------+

| User Applications |

+------------------------+

|

v

+------------------------+

| Shell (BASH) |

+------------------------+

|

v

+------------------------+

| System Libraries (glibc)|

+------------------------+

|

v

+------------------------+

| Kernel (core) |

| - Memory Mgmt |

| - Process Mgmt |

| - Device Drivers |

+------------------------+

|

v

+------------------------+

| Hardware (CPU, |

| RAM, Disk, etc.) |

+------------------------+

Task 7 : Is it legal to edit Kernal ? when do you think we have to in case?

When would someone edit the kernel?

Most people don’t need to, but some do in special cases, like:

1. To add support for new hardware (like a new keyboard or device)

2. To speed up or improve the system

3. To remove extra stuff for smaller systems (like in robots or smart TVs)

4. To add extra security

5. To learn how Linux works

Task 8 : What is LILO? Explain ?

LILO, short for Linux Loader, is an older bootloader program used in Linux systems

What does LILO do?

• It helps your computer choose and start the operating system.

• You can set it to:

• Boot Linux

• Boot Windows (if you have both installed)

• Boot other OSes

Features of LILO:

• Can boot multiple operating systems

• Simple and fast

• Doesn’t depend on file systems (works directly with disk)

Note:

• LILO is old and not used much today.

• New systems usually use GRUB (a more powerful boot loader).

Task 9:

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

Types of Shells in Linux:

A shell is a program in Linux that lets you interact with the operating system.

• You type commands, and the shell passes them to the system to run.

• It acts like a middleman between you and the Linux system.

There are many shells, but here are the main ones:

Shell Name Description

BASH (Bourne Again Shell) --Most common shell in Linux; user-friendly

SH (Bourne Shell) -The original Unix shell

KSH (Korn Shell) -Faster and supports scripting

CSH (C Shell) - Similar to C programming style

ZSH (Z Shell) - Advanced features, used by many developers

In short:

• A shell is where you type commands in Linux.

• BASH is the most popular one.

• Others like ZSH, KSH, CSH are also used for different needs.

Task 10:

What is swap space?

Swap space is like extra memory on your hard disk.

• Your computer uses RAM to run programs.

• If RAM gets full, Linux uses swap space as a backup memory.

Where is Swap Space?

It can be: • A separate partition on the hard disk

• A swap file inside the file system

Why is it useful?

• Prevents your system from crashing when RAM is full• Helps run big programs • Supports hibernation (stores RAM content temporarily)

Note:

• Swap is slower than RAM because it uses disk

• Too much swapping = slow performance

In short:

Swap space is slow backup memory on your disk that Linux uses when your RAM is full.

Task 11:

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

Mount means connecting a storage device (like a pen drive, hard disk, or partition) to your Linux system so you can use it.

Mount = Make a drive usable

Unmount = Remove it safely

Task 12:

What is chmod command ? how to use it?

chmod means “change mode”.

It is used to change file or folder permissions — like who can read, write, or run a file.

In short:

• chmod changes who can read/write/run a file.

• You can use numbers or letters.

• Example: chmod 777 file.txt → everyone can read, write, run it.

What are permissions?

In Linux, each file has 3 types of users:

• Owner (the one who created it)

• Group (a group of users)

• Others (everyone else)

And 3 types of permissions:

• r = read

• w = write

• x = execute (run)

How to use chmod:

1. Using numbers (easiest way)

Permission Number

read (r) 4

write (w) 2

execute (x) 1

You add the numbers for each user.

Example:

chmod 755 file.txt

Means:

• 7 (4+2+1) = owner can read, write, execute

• 5 (4+0+1) = group can read, execute

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

2. Using letters (alternative way)

chmod u+x file.sh

• u = user (owner)

• g = group

• o = others

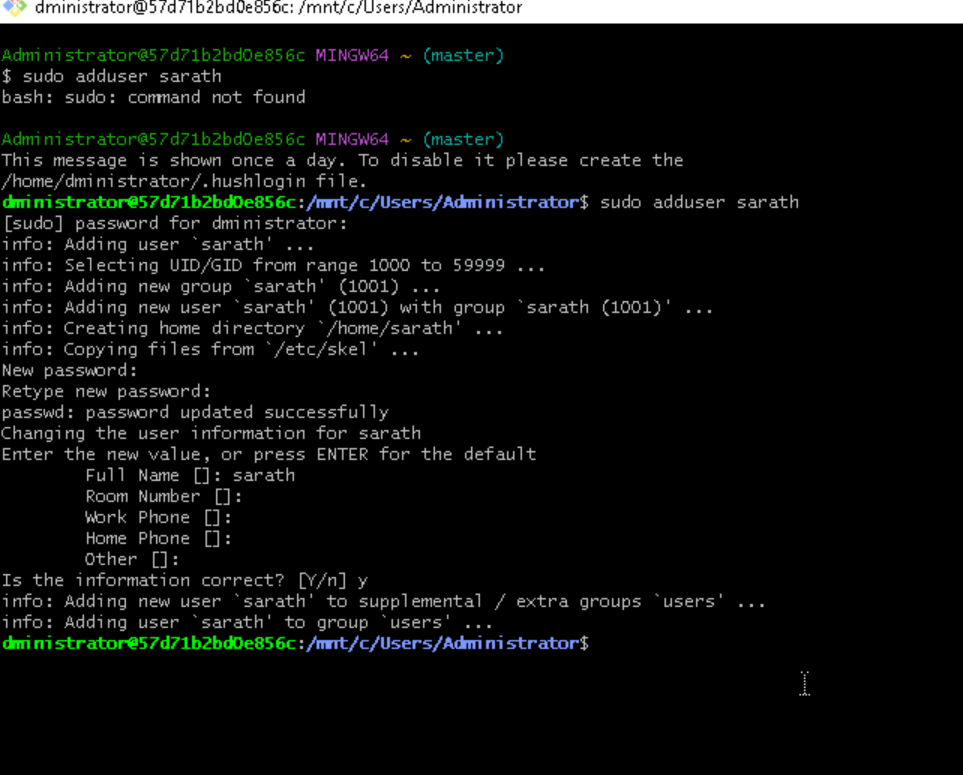
• +x = add execute permission

Task 13:

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

Wsl to enter linux

Sudo adduser – create new user

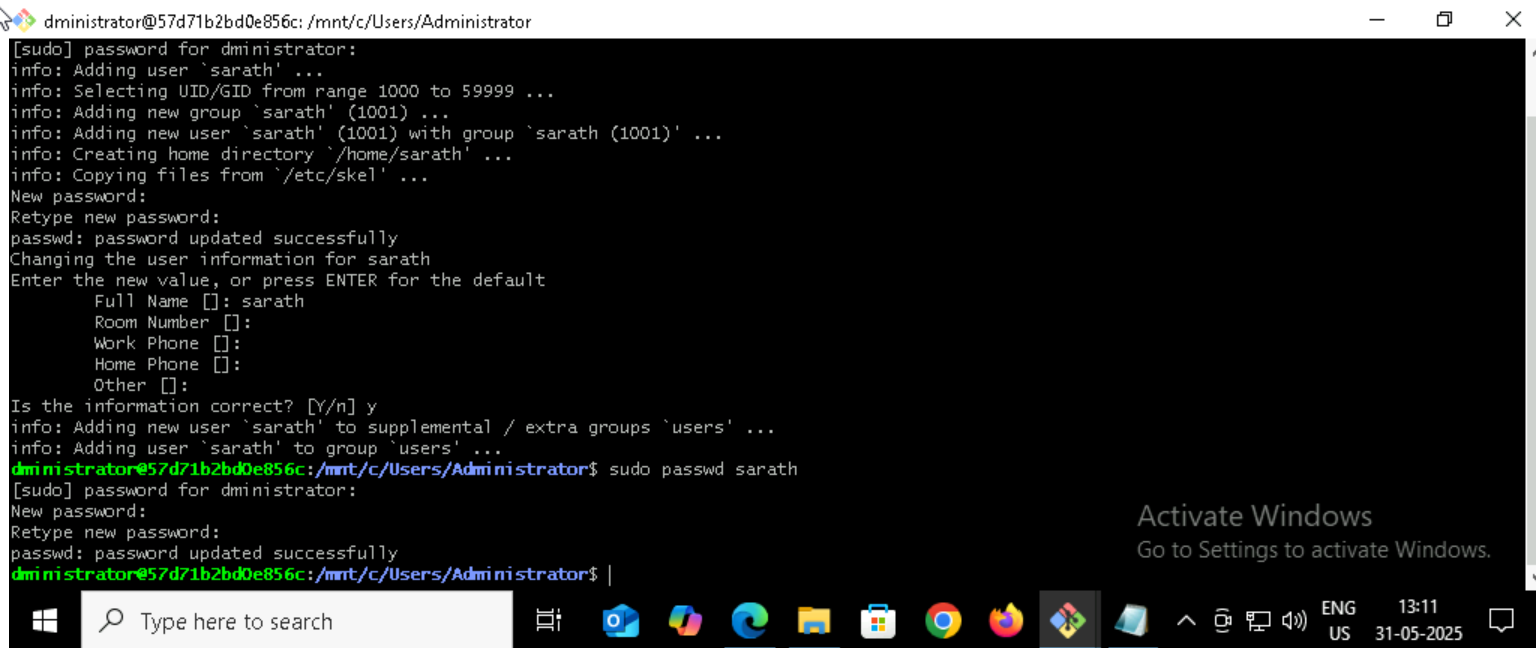


Task 14:

Can you change the password of a user?

How do you do that? Plz share ss

Sudo passwd (username)



Task 15:

What is diff between Process and Thread?

Process vs Thread (in simple words):

Process:

• A program that is running.

• Has its own memory and resources.

• Example: Opening Chrome is a process.

Thread:

• A smaller part inside a process.

• Shares memory with other threads in the same process.

• Example: Chrome loading one tab is a thread, loading another tab is another thread.

• Process = Big program

• Thread = Small tasks inside the program

• A process is a complete program that is running.

• A thread is a smaller part inside a process that does a specific task.

• A process has its own memory.

• Threads share the same memory within a process.

• Processes are heavier and take more time to start.

• Threads are lighter and faster to create.

• If one process crashes, others are usually safe.

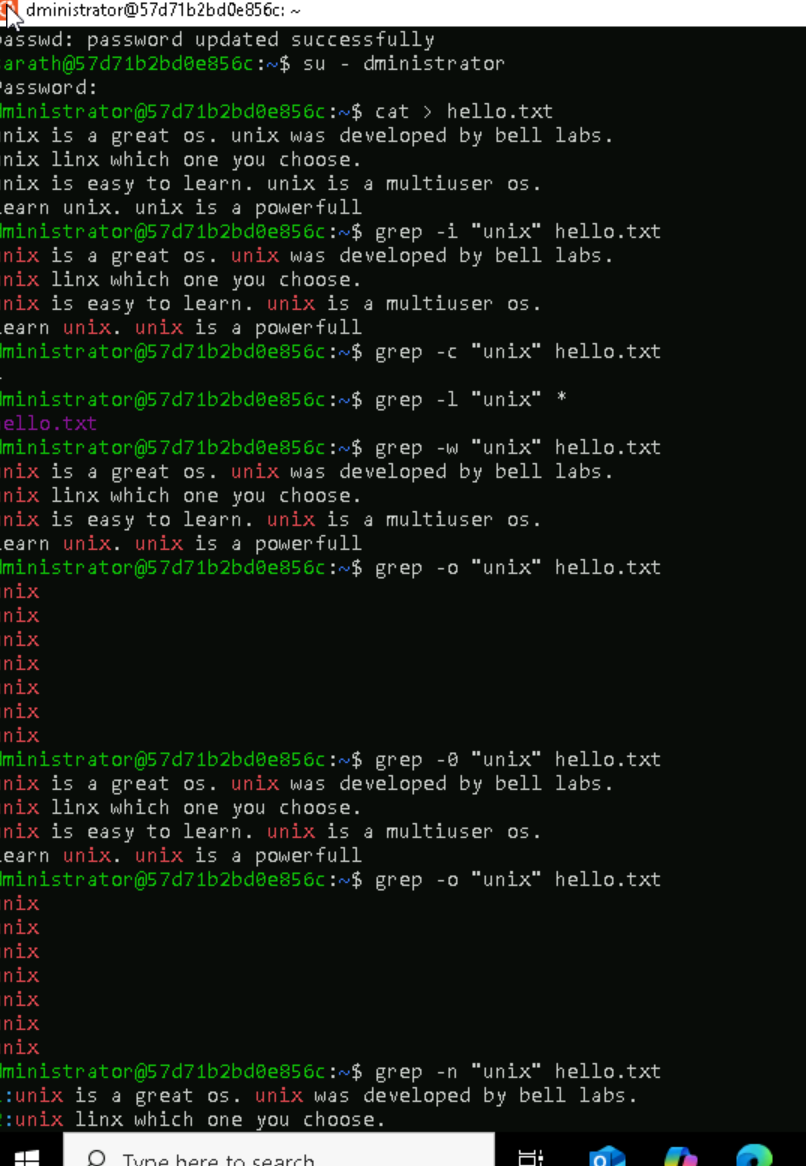
• If one thread crashes, it can affect the whole process.

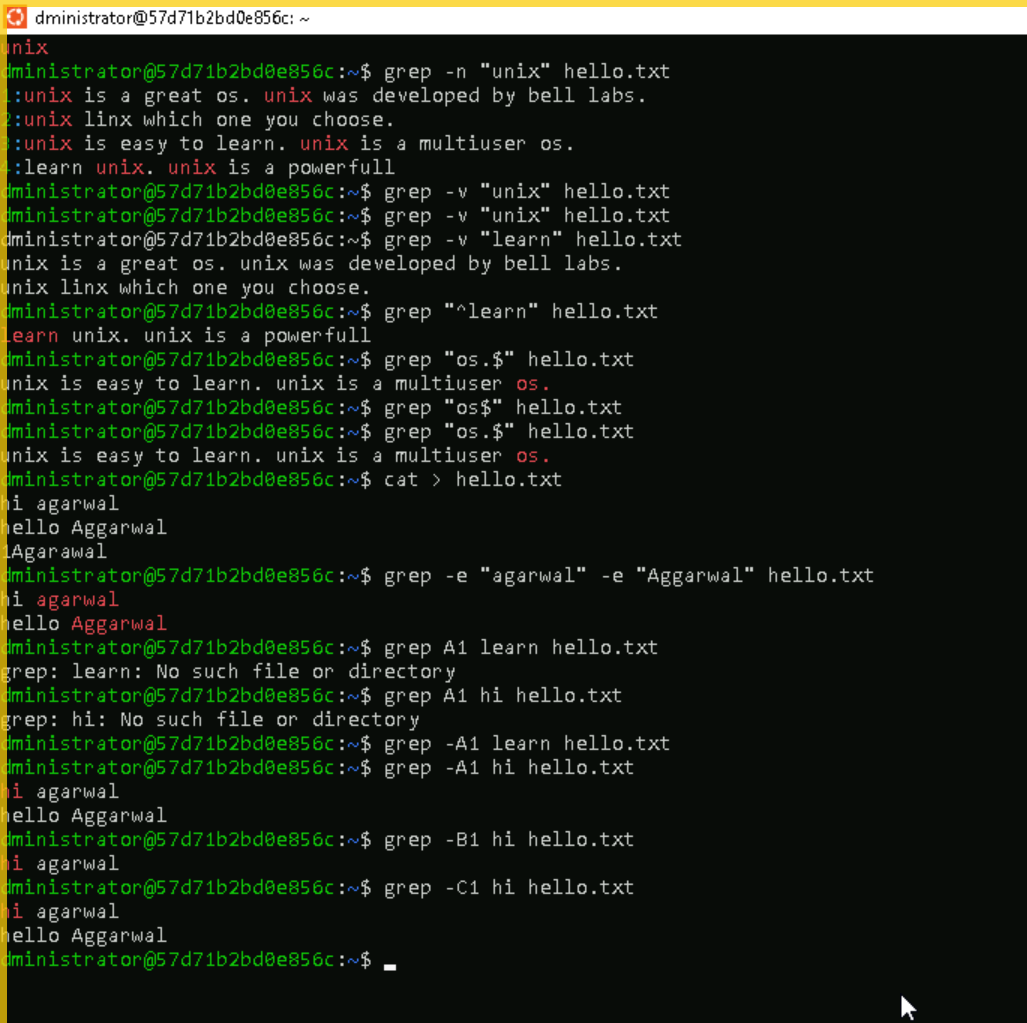
• Example: Opening Chrome is a process. Each tab opened in Chrome is a thread.

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>





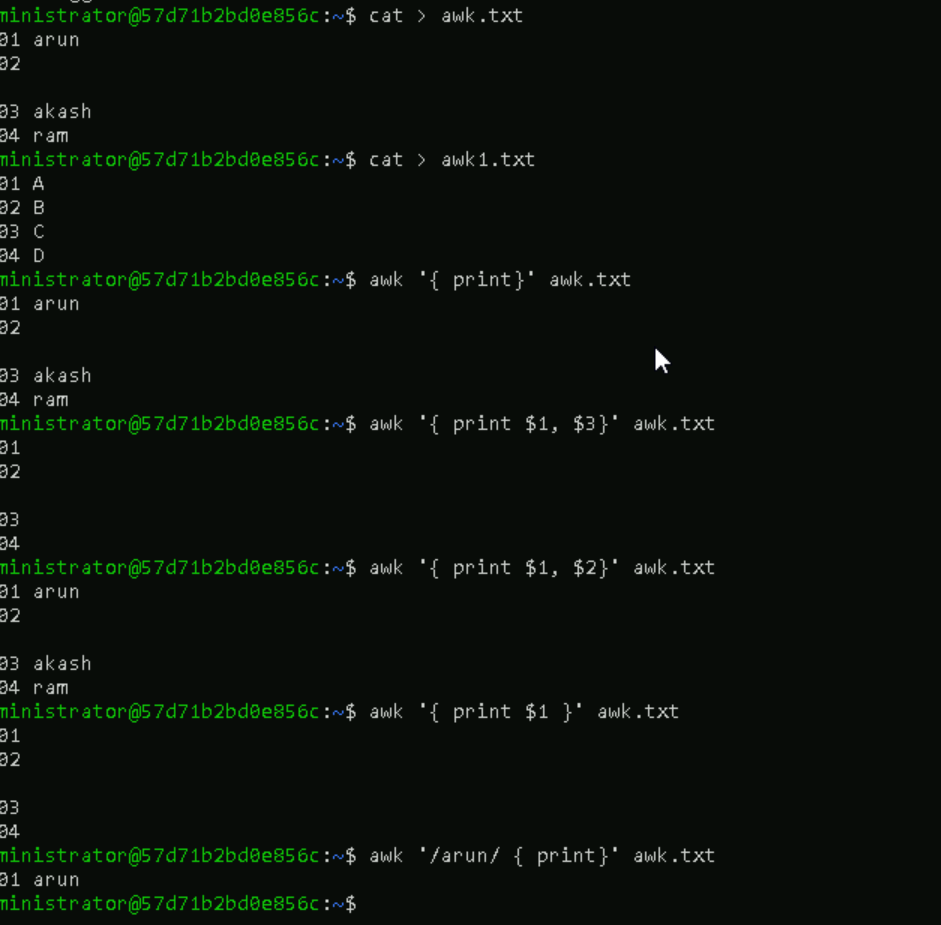
Task 17:

AWK commands in doc 15 Linux AWK commands.

Example 1: Print Contents of a File

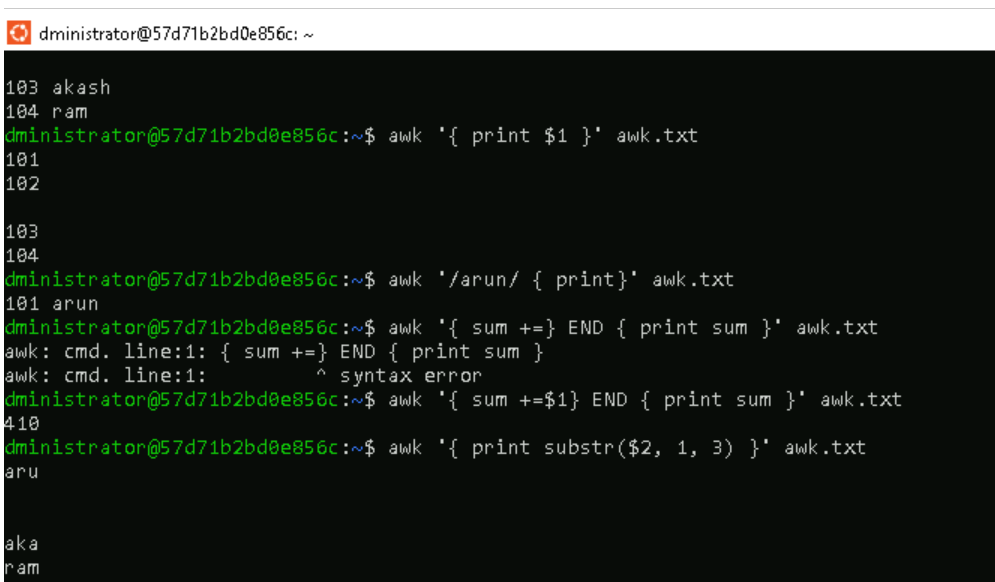
Example 2: Print Specific Columns of a File

Example 3: Filter Lines Based on a Condition

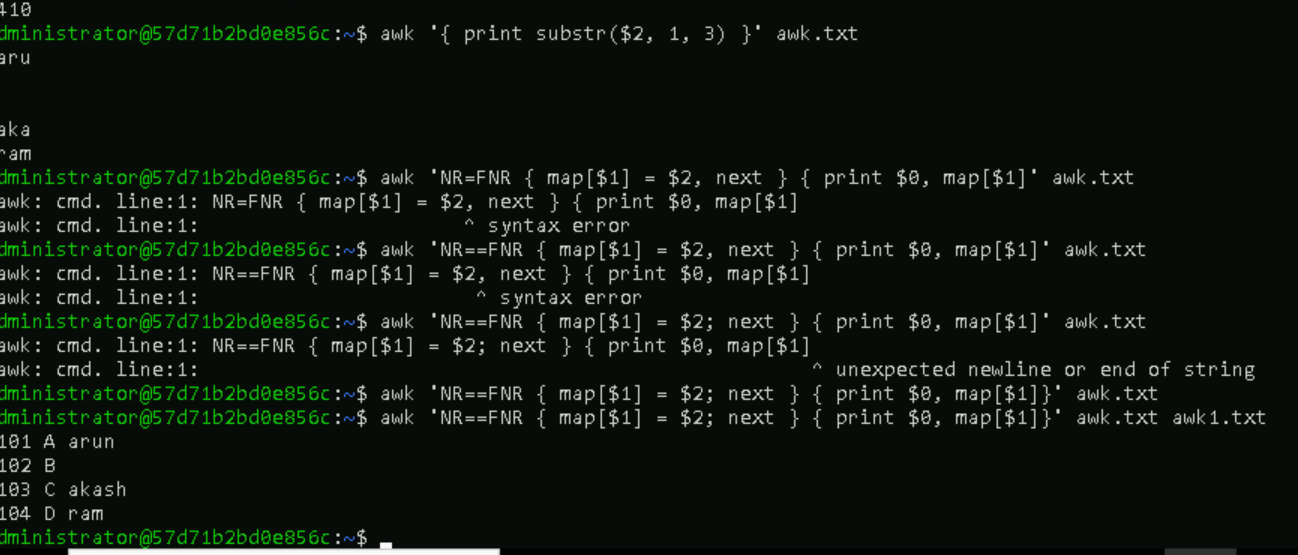


Example 4: Sum Values in a Column

Example 5: Extract Substring from a Column

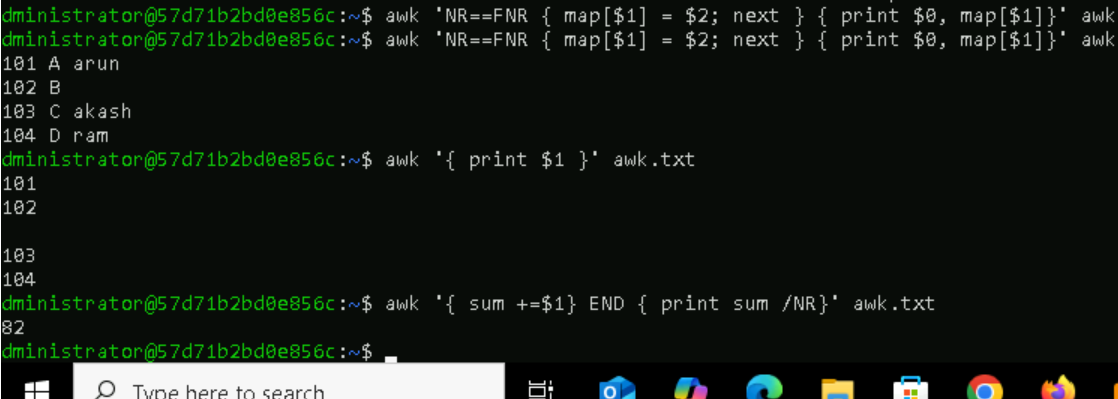


Example 6: Join Two Files Based on a Common Column



Example 7: Extract First Column of a File

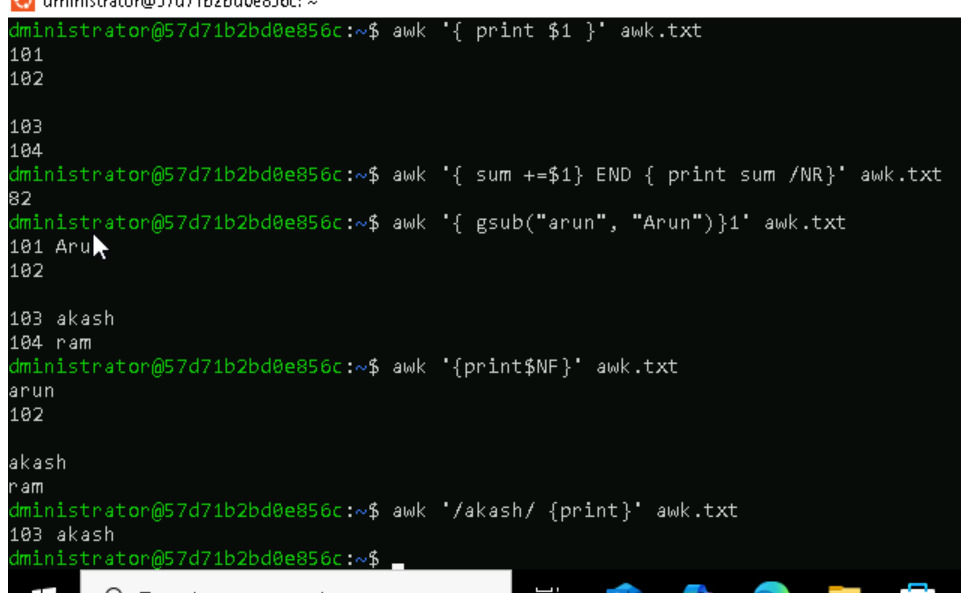
Example 8: Calculate Average of a Column



Example 9: Replace a String in a File

Example 10: Display Last Field of a File

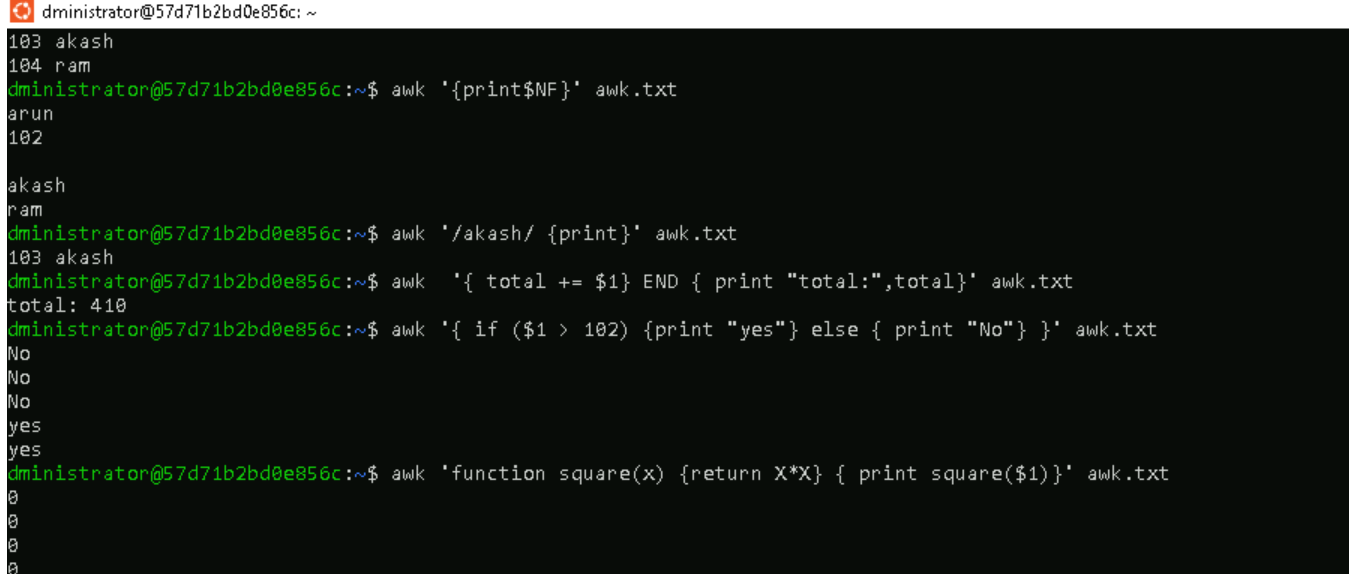
Example 11: Using Regular Expressions



Example 12: Using Variables

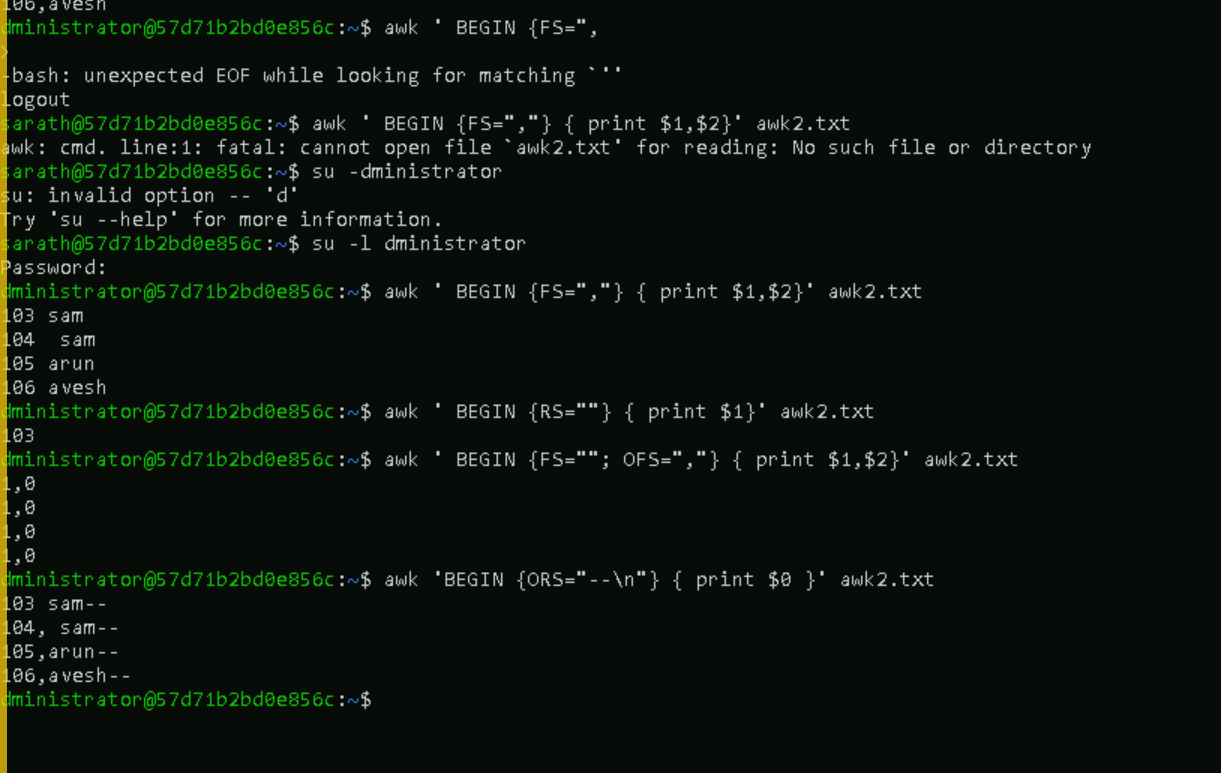
Example 13: Using Control Statements

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.

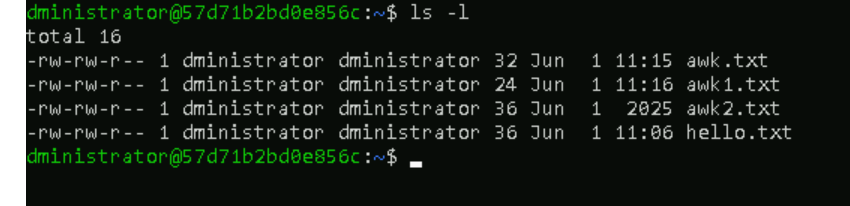


Task 18:

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

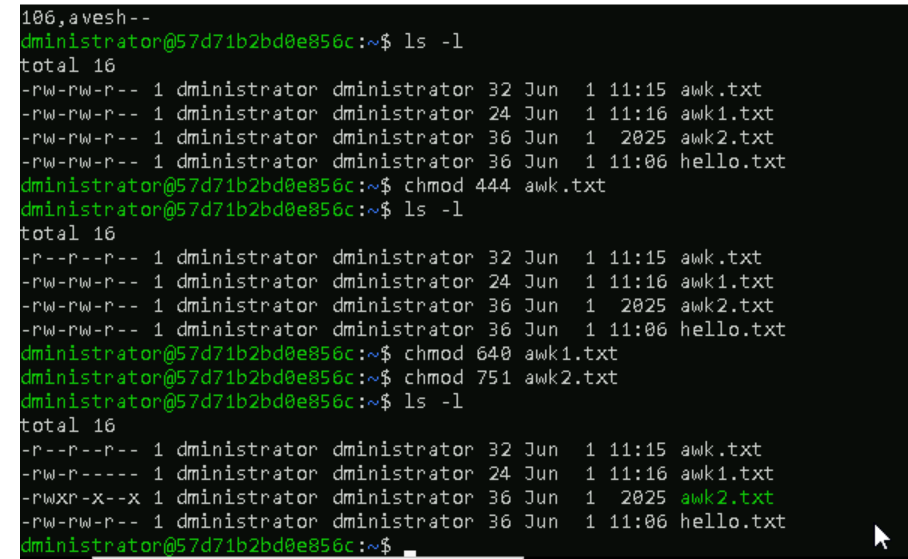
group: Read and Execute

other: Execute

Task 24:

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

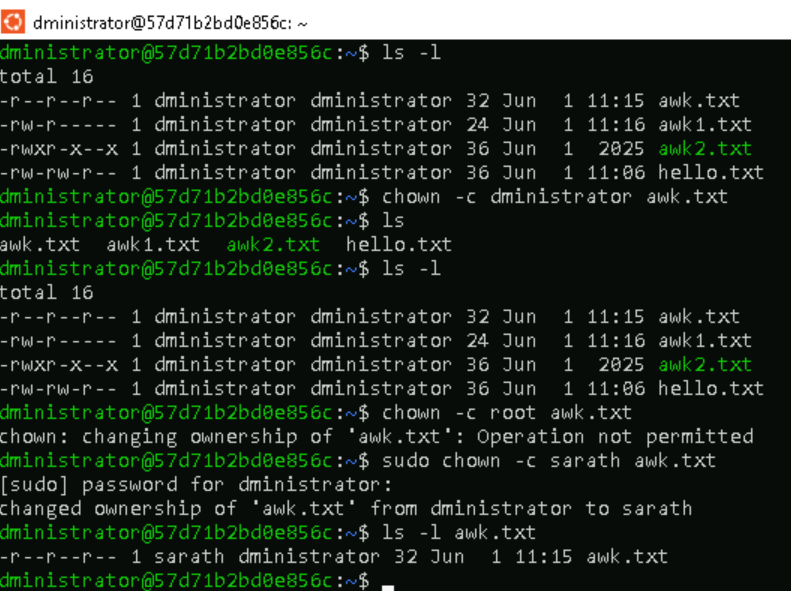
Hint : use chmod 751 filename



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

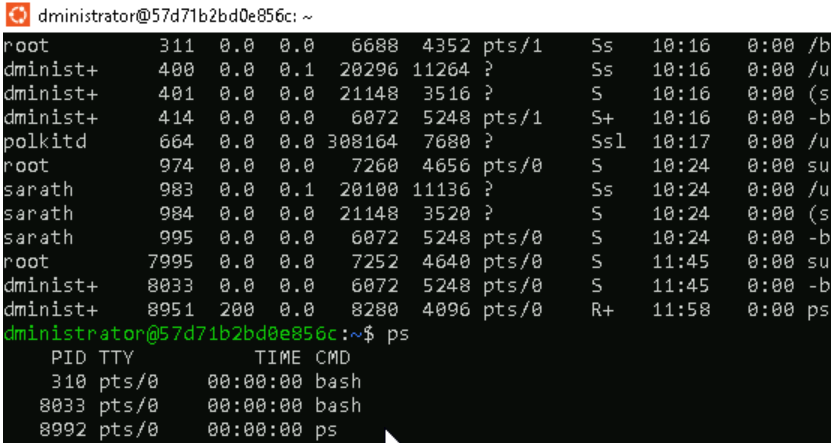
**See All Running Processes**

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:

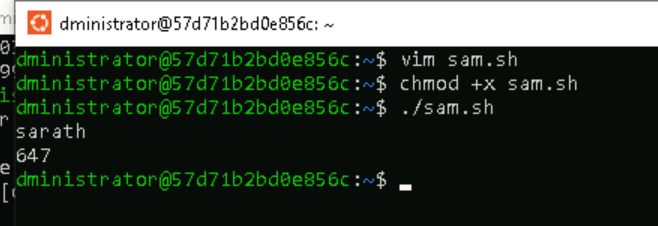
Name =  “prasunamba”

Id  = 10001

And check

Echo $Name

Chek the output



Task 31:

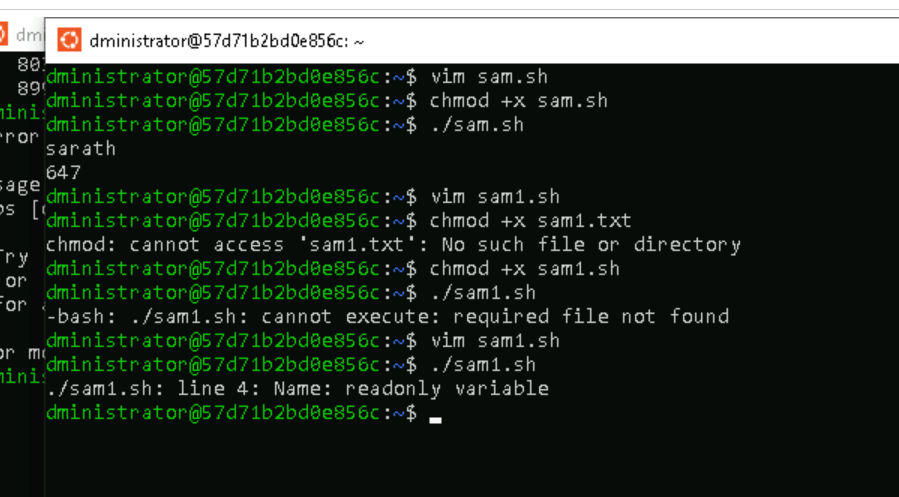
Can you make the above name variable read only..

Ex:

Name = “Prasunamba”

Readonly Name

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



Task 32:

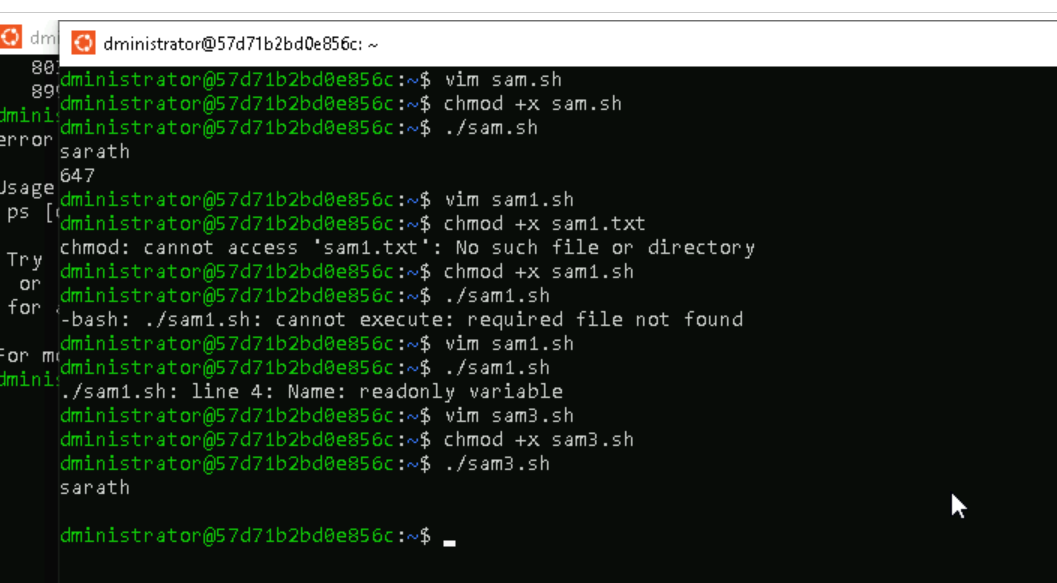
Now will unset or delete the variables

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"

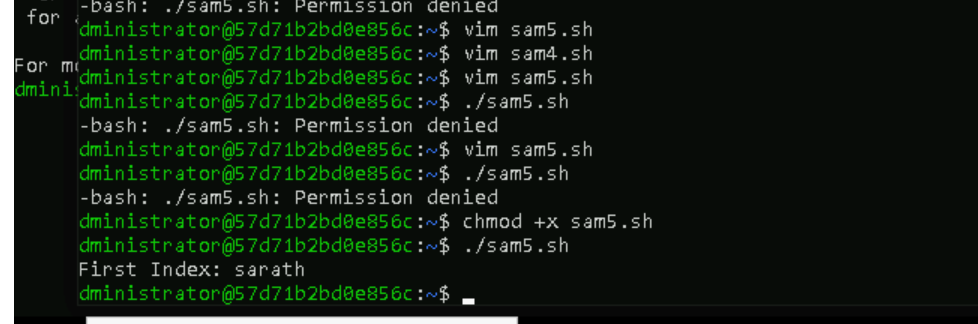
NAME[1]="Sita"

NAME[2]="Tina"

NAME[3]="Veena"

NAME[4]="Tim"

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

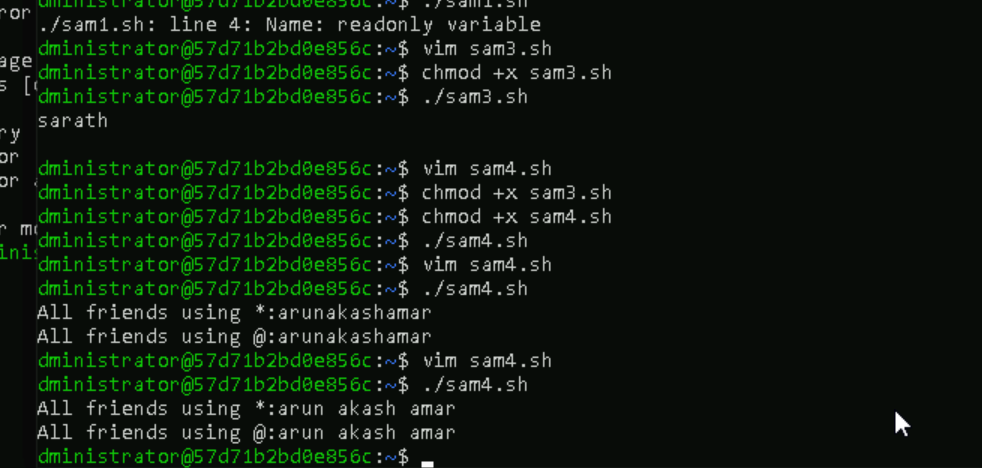


Task 34:

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

