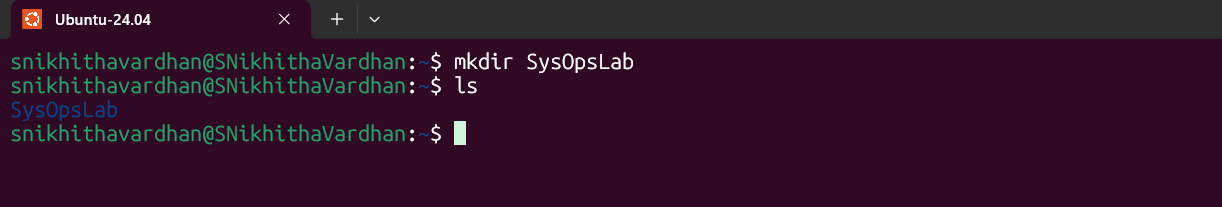
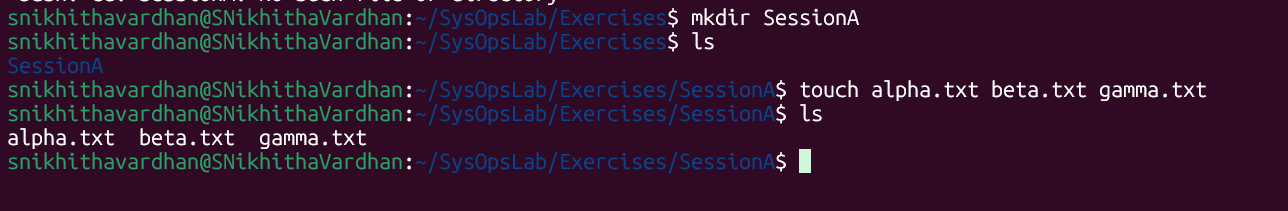
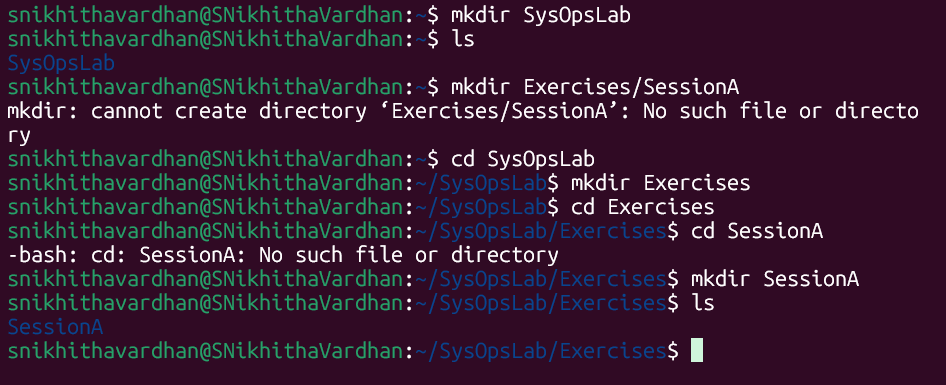
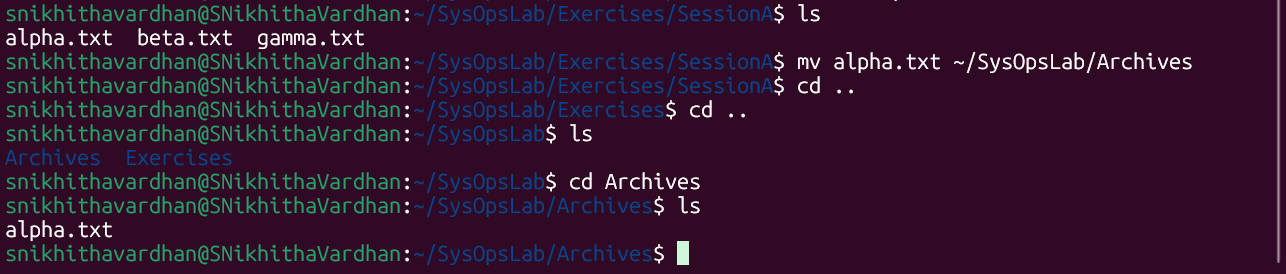
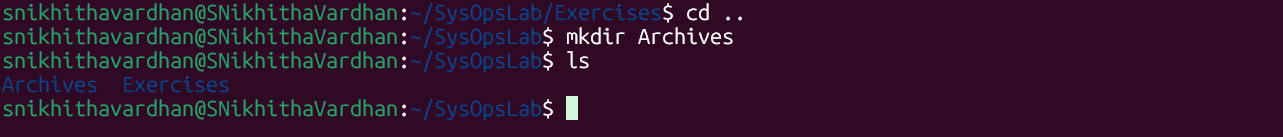
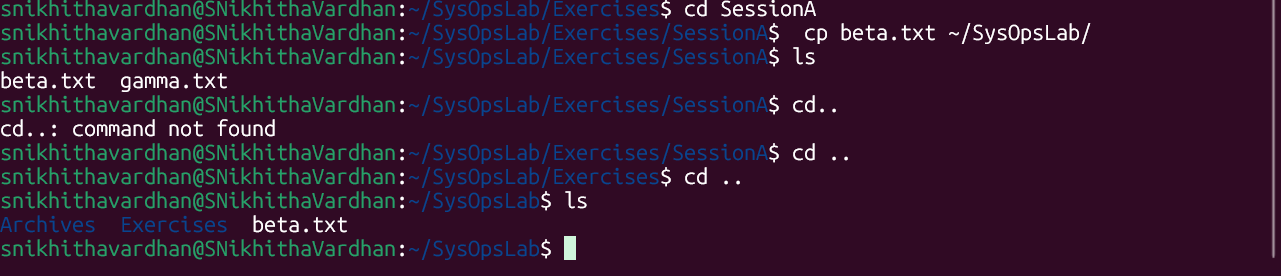
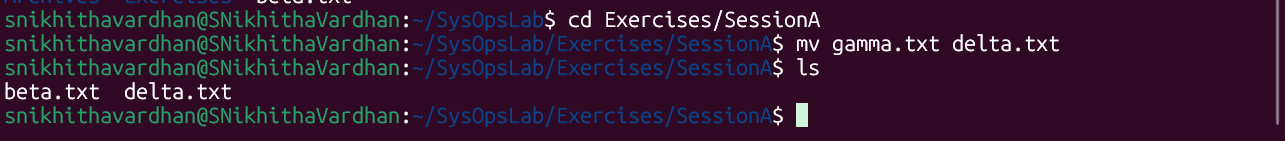
# **241IT077 - TrapShield-Linux Task (1)**

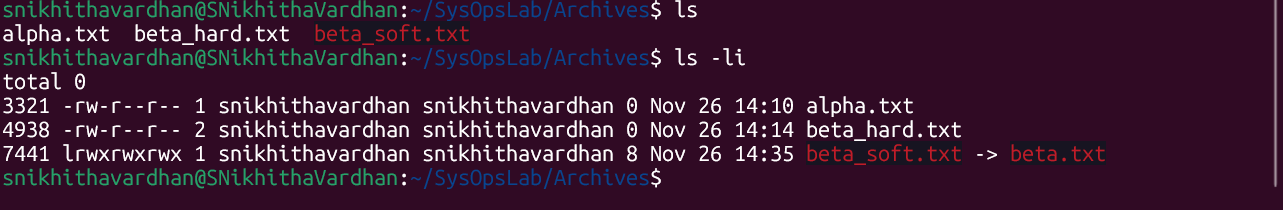
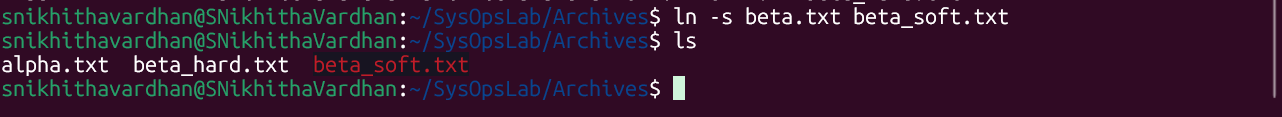
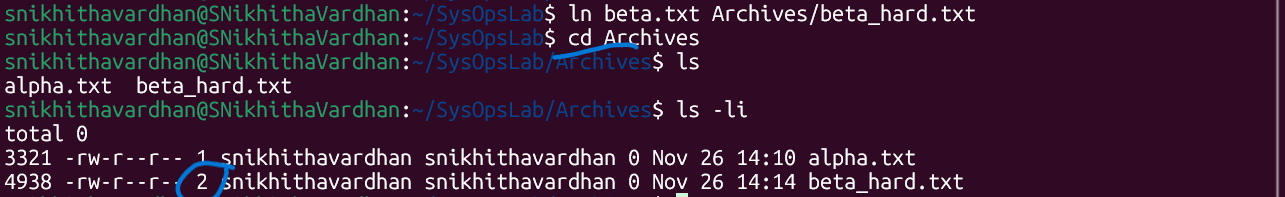
***-Singamaneni Nikhitha Vardhan***

I have tried to do in WSL2 and used the available commands .Below are the screenshots o teh commands and scripts written or each task .

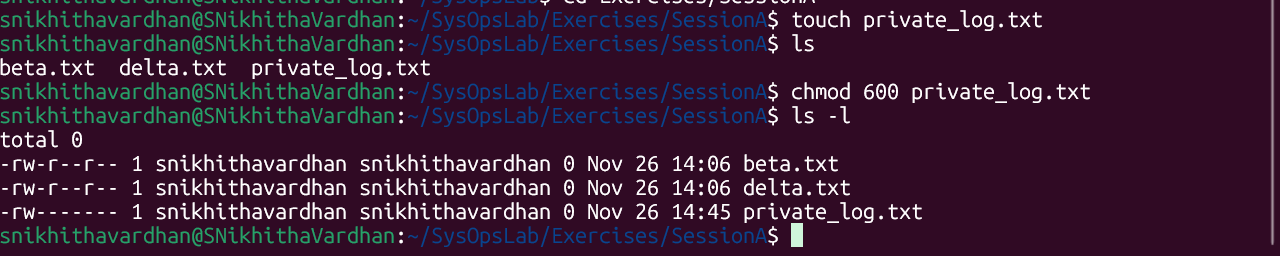
**Section 1: Core Command Operations**

### **1. Directory Navigation & File Handling**

* Create the following directory structure: ~/SysOpsLab/Exercises/SessionA
* Inside SessionA, create three text files: alpha.txt, beta.txt, gamma.txt
* Move alpha.txt into a new folder: ~/SysOpsLab/Archives/
* Copy beta.txt into: ~/SysOpsLab/
* Rename gamma.txt to delta.txt.

**Bonus Challenge:**Create a hard link and a soft link for beta.txt in the Archives directory and verify their types using ls -li.

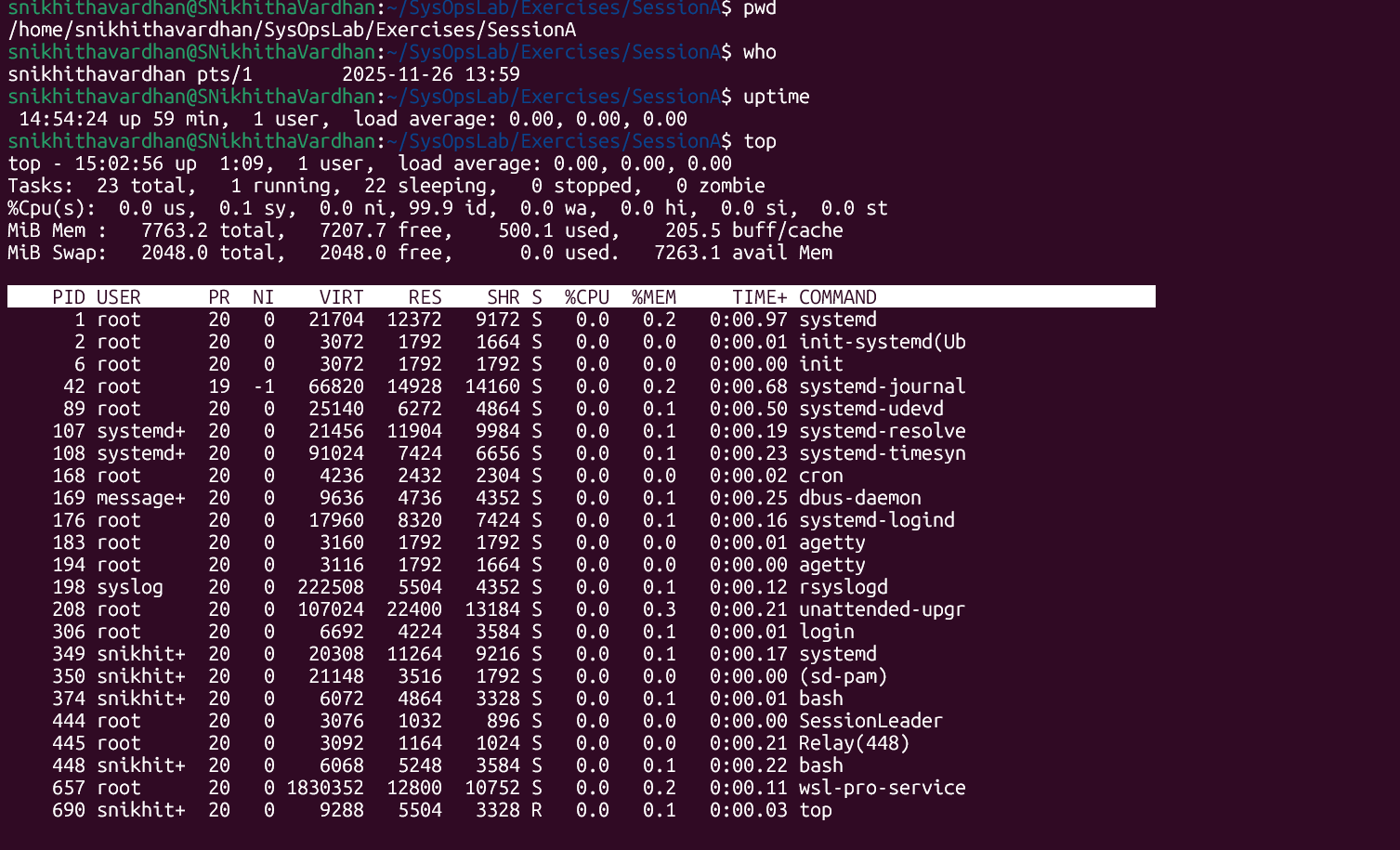
### **2. File Permissions and Ownership**

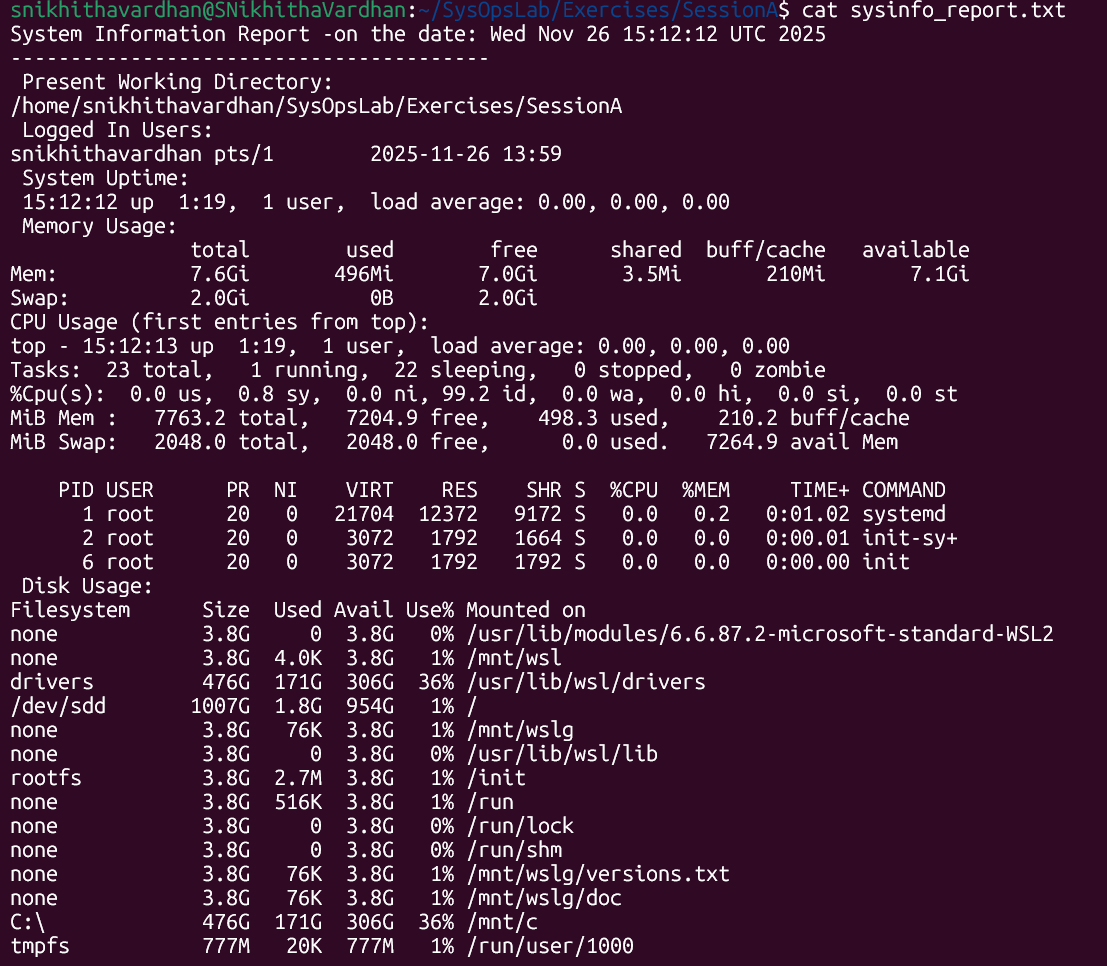
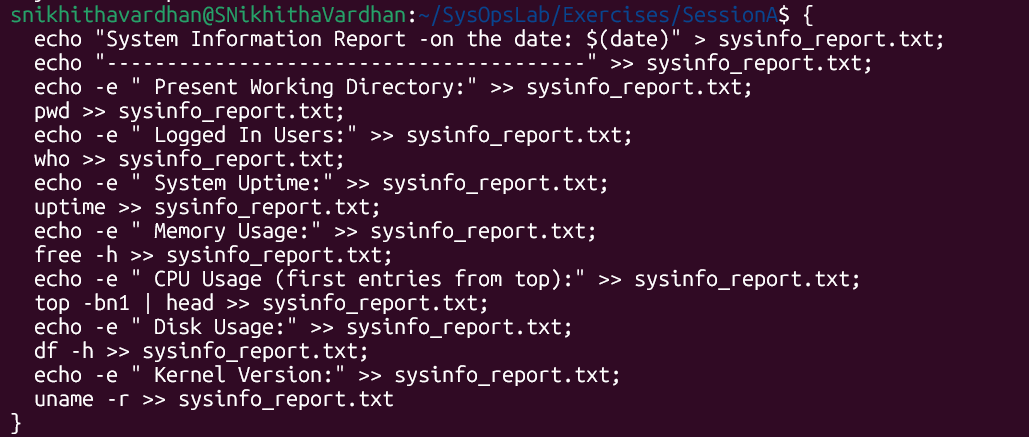
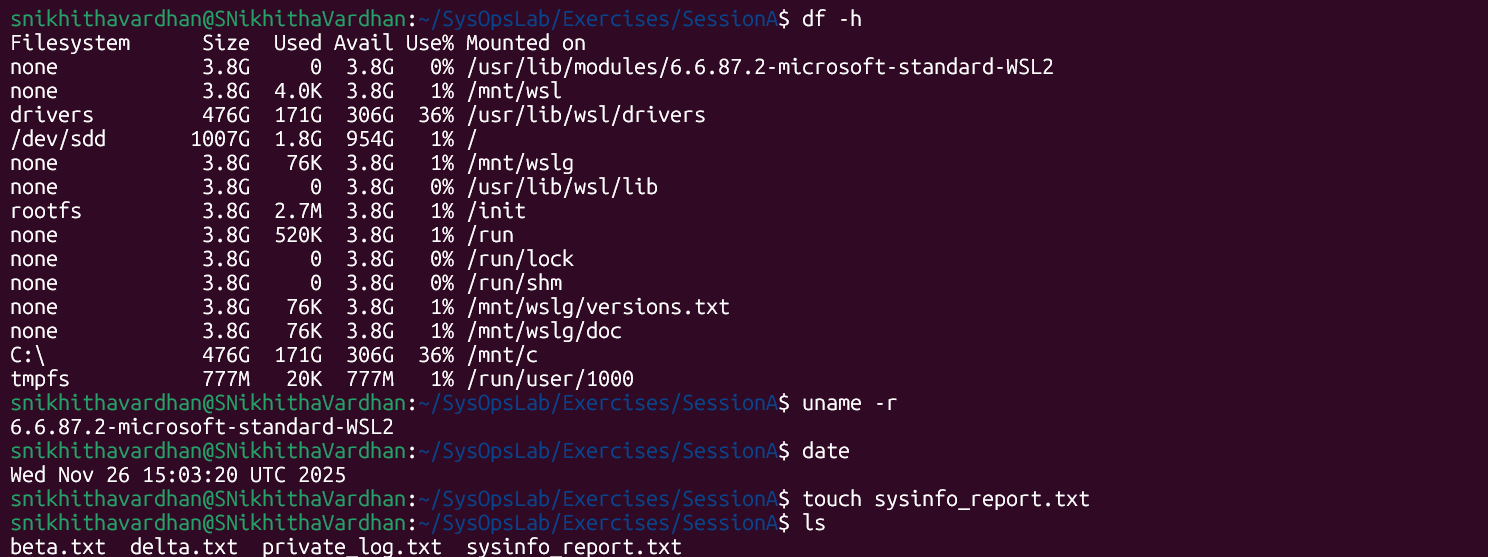
* Create a file named: private\_log.txt
* Set permissions so **only the owner can read and write** it.
* Verify permissions using ls -l.
* Change ownership to another user (if applicable).

**Bonus Challenge:** Apply an **Access Control List (ACL)** to grant another user read-only access.

### **3. System Status & Resource Information**

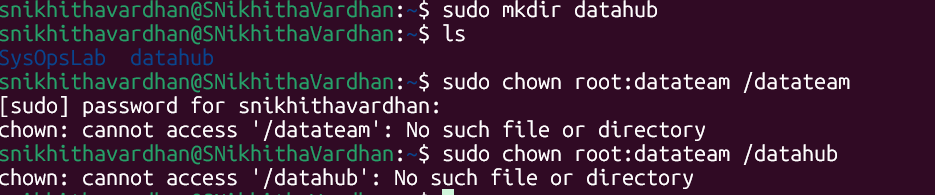
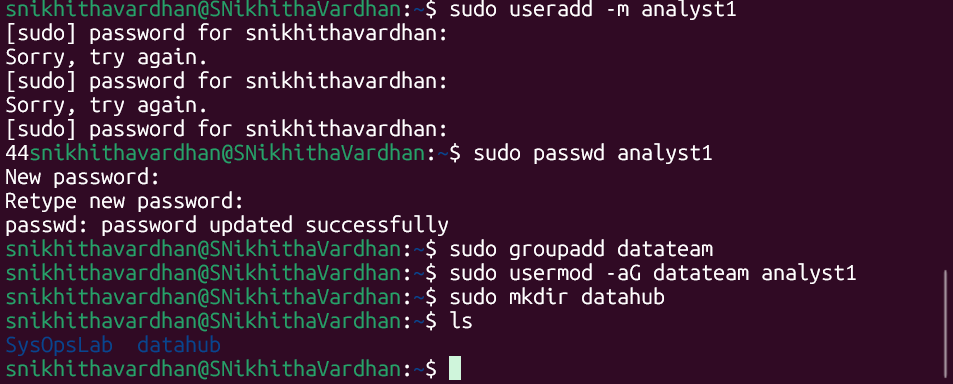
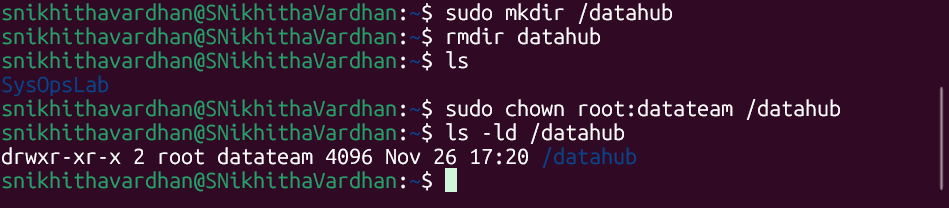
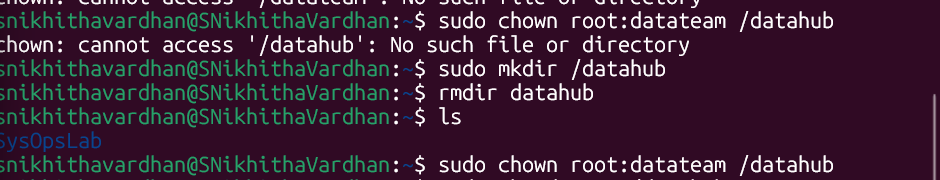
Use Linux utilities to gather:

* Present working directory
* Current logged-in users
* System uptime
* Memory usage summary
* CPU and disk utilization
* Kernel version

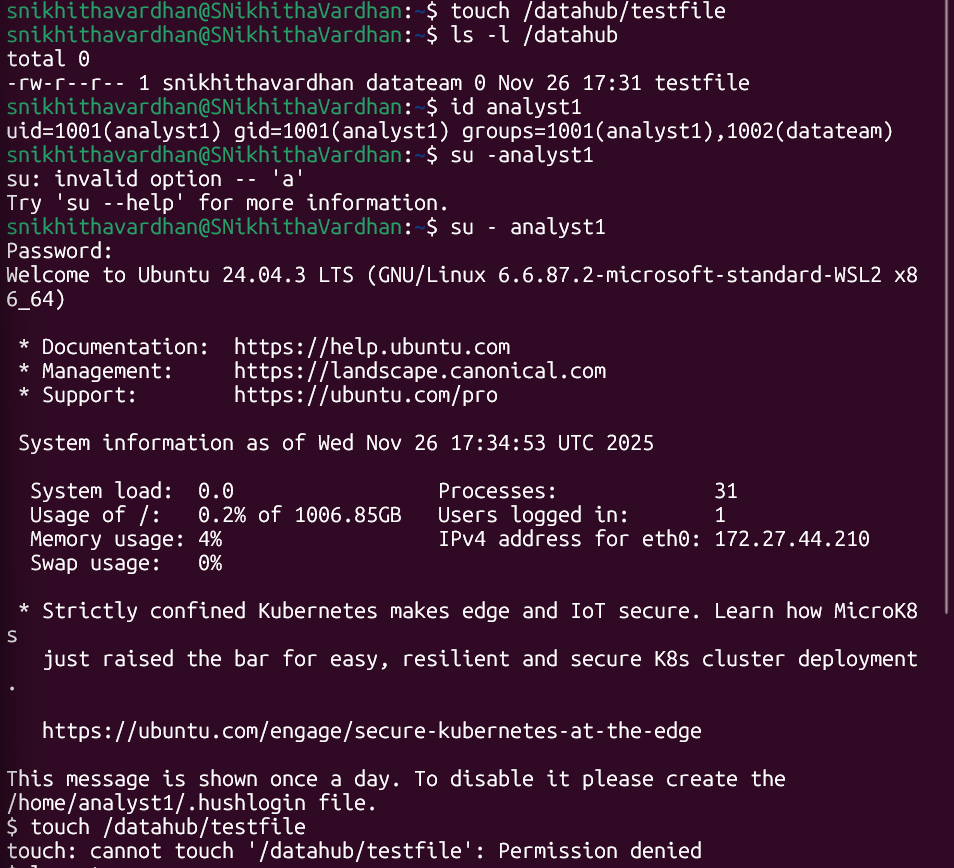
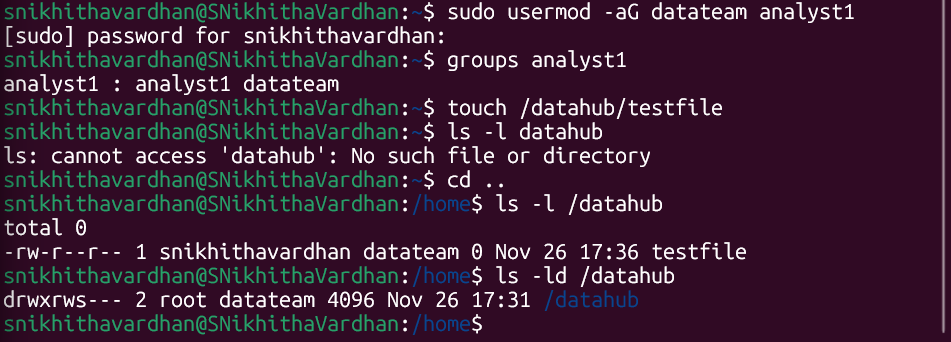
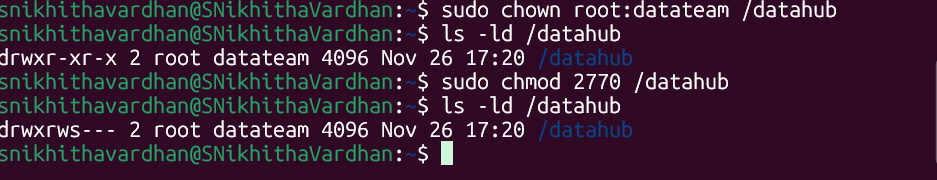
**Bonus Challenge:**Save all the information to sysinfo\_report.txt and include the current date/time in the header.

## **Section 2: User, Group & Permission Management**

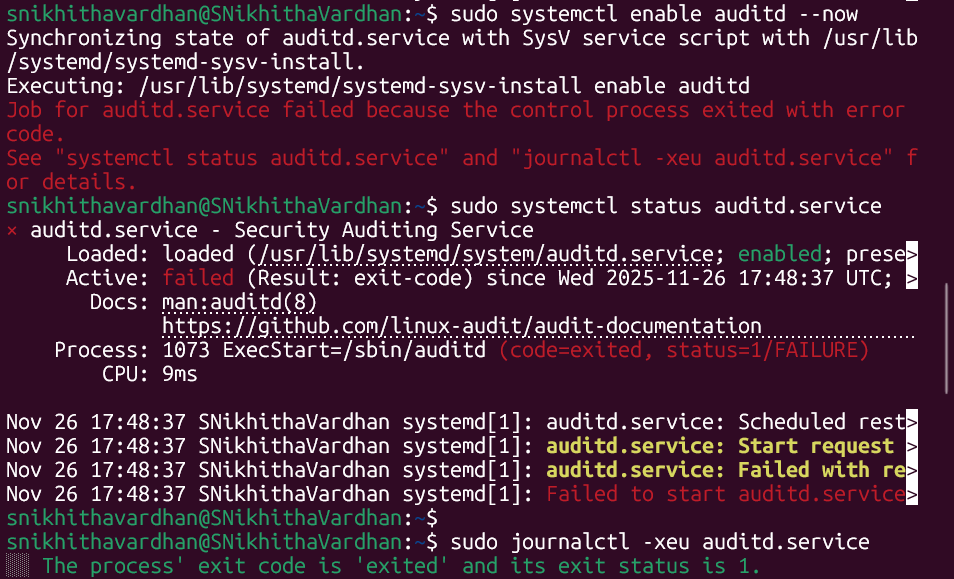
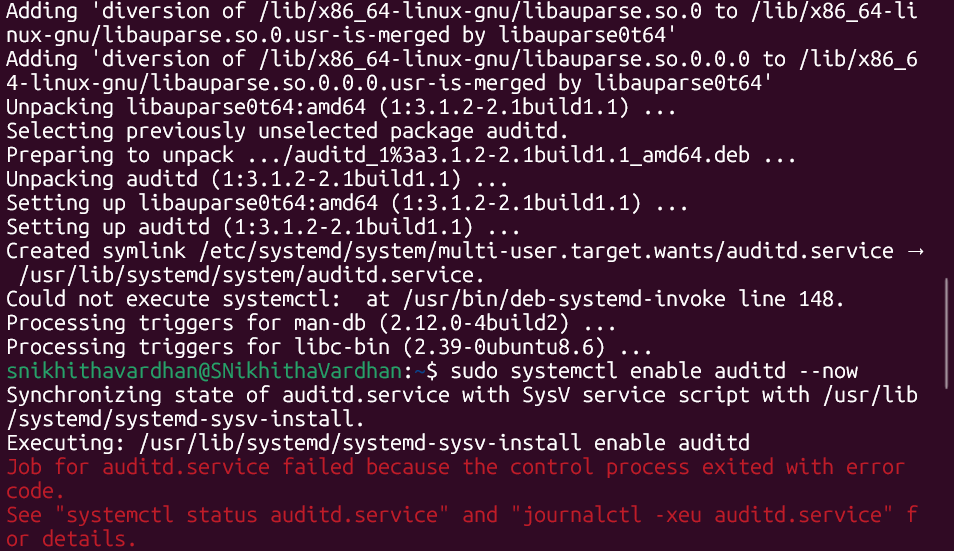
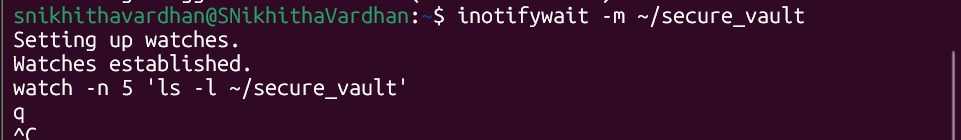
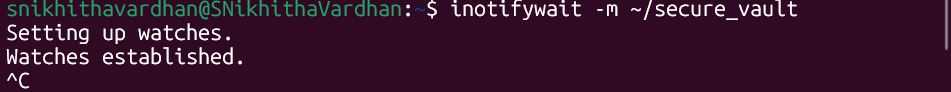
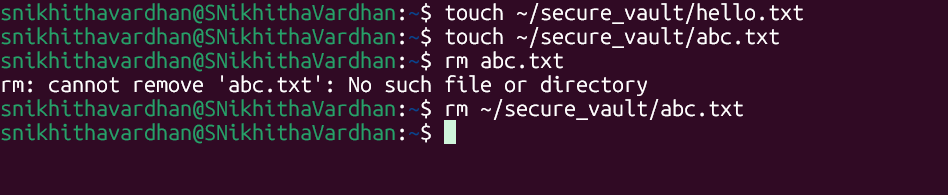
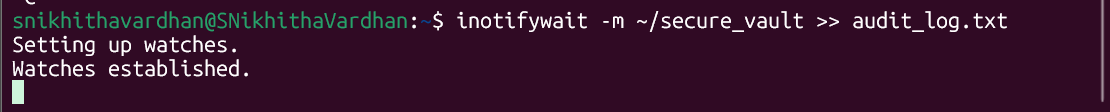
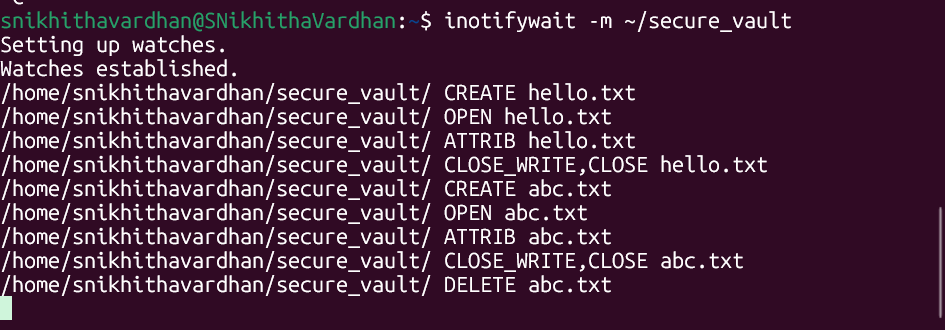
### **4. User and Group Administration**

* Create a new user: analyst1
* Create a new group: datateam
* Add analyst1 to datateam.
* Create a shared directory /datahub that is **accessible only to datateam members**.
* Set **SGID** on the directory so that files created inside inherit the group.

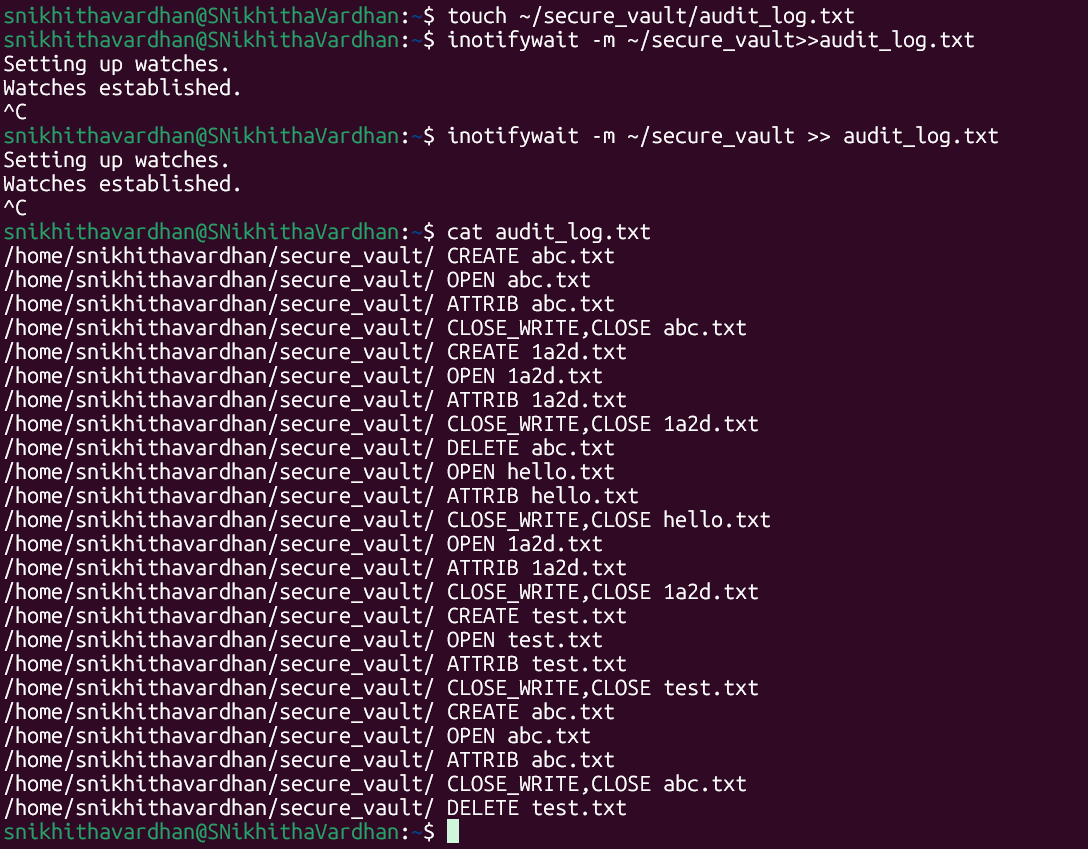
**Bonus Challenge:**Use id and groups commands to verify group membership and permission inheritance.



### **5. File Security and Auditing***(I have used inotifywait as I am using WSL2)*

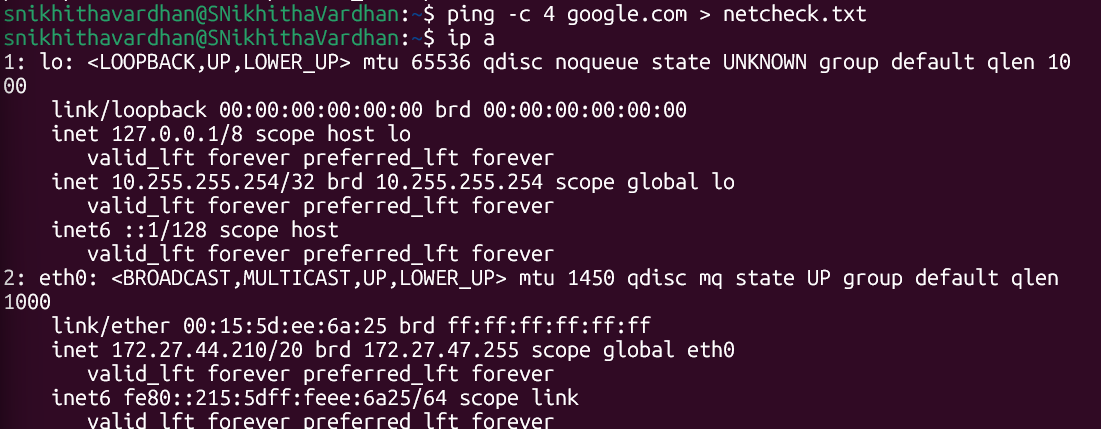
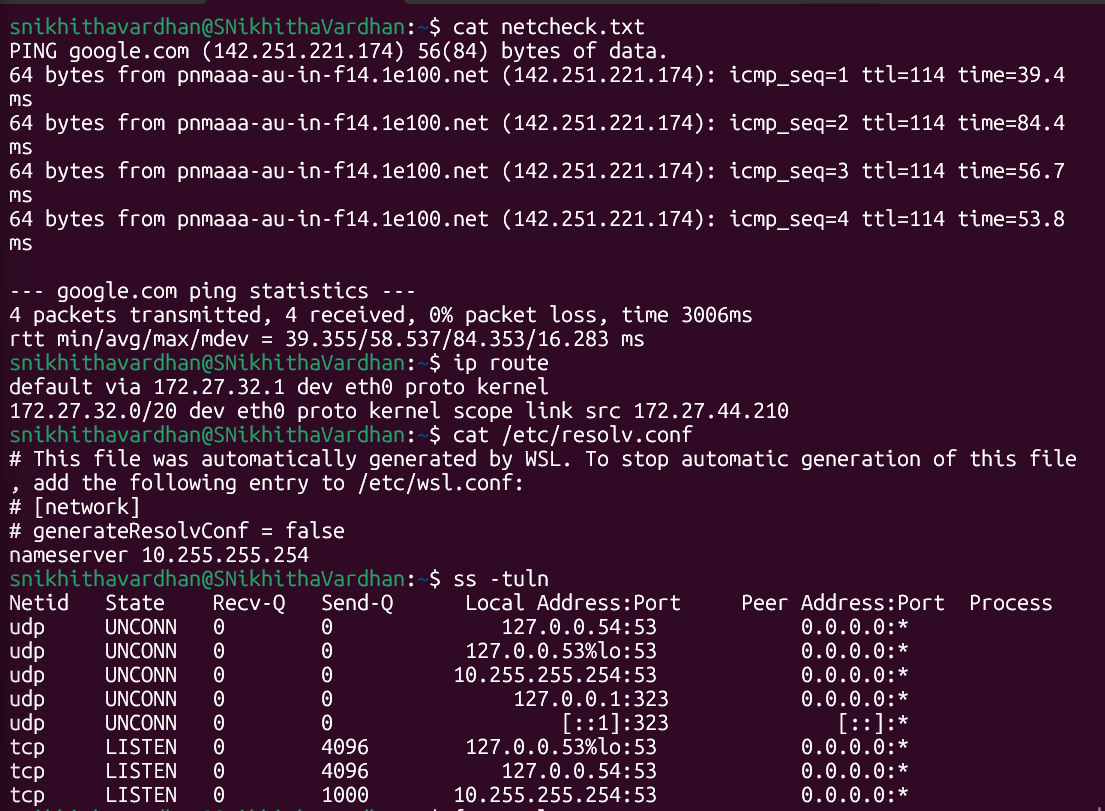
* Create a directory secure\_vault in your home folder.
* Restrict all access except for the owner.
* Enable file change auditing using auditd or an equivalent tool.
* Record and review audit logs for access attempts.

**Bonus Challenge:** Generate a simple audit summary report using ausearch or grep.

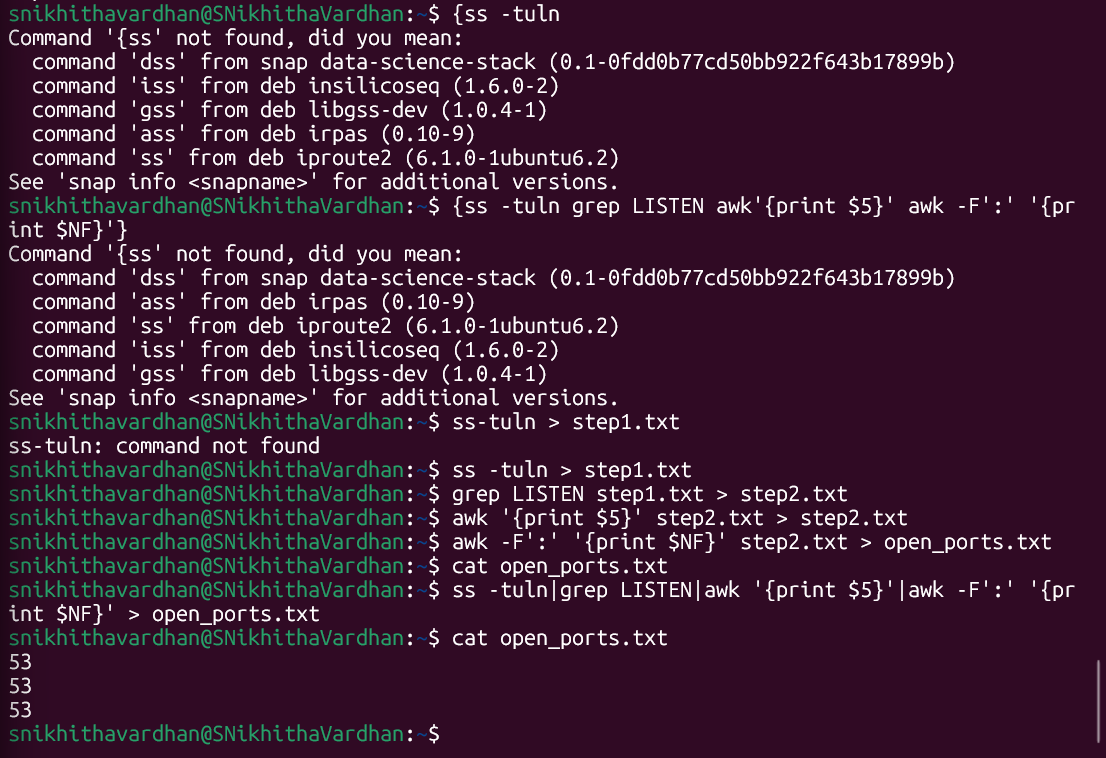
* 

## **Section 3: Networking and Connectivity**

### **6. Network Diagnostics**

* Ping google.com and redirect the output to netcheck.txt.
* Display network interfaces using: ip a
* Show the default gateway and DNS configuration.
* View all active connections and listening ports using: ss -tuln

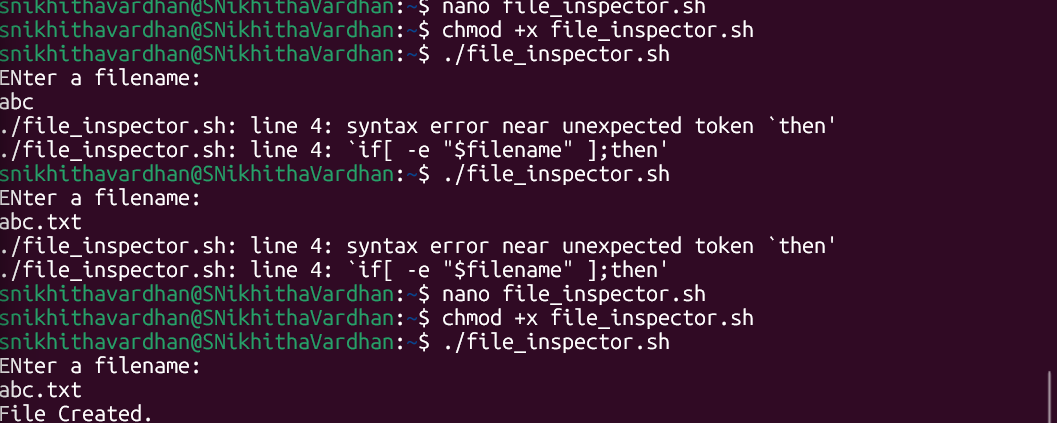
**Bonus Challenge:**Write a one-line command to extract only **open TCP ports** and save them to open\_ports.txt.

* 

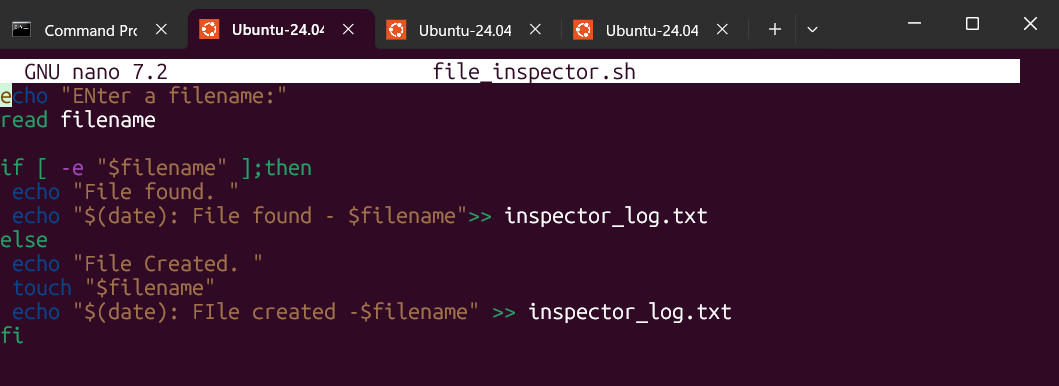
## **Section 4: Shell Scripting and Automation**

### **7. File Checker Script**

Create a shell script file\_inspector.sh that:

1. Prompts the user to enter a filename.
2. Checks if it exists in the current directory.
3. Displays:
   * “ File found.” if it exists.
   * “ File created.” if not found, and creates the file.

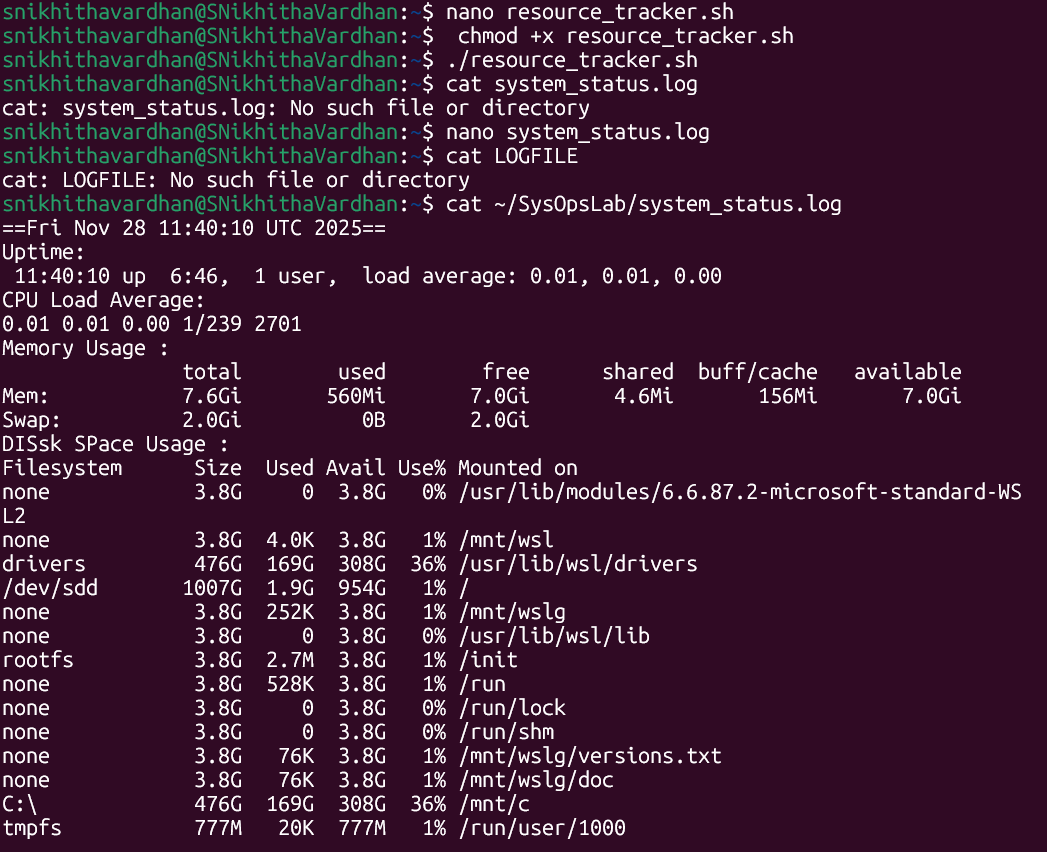
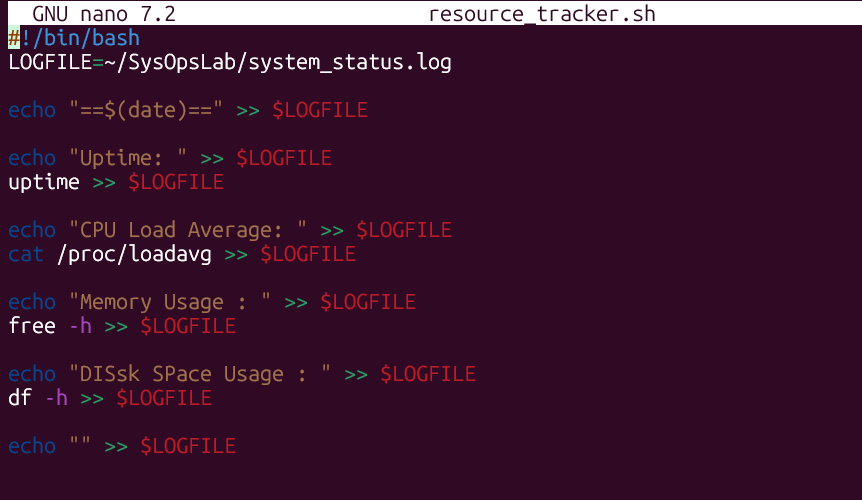
**Bonus Challenge:**Log each action with a timestamp to inspector\_log.txt.

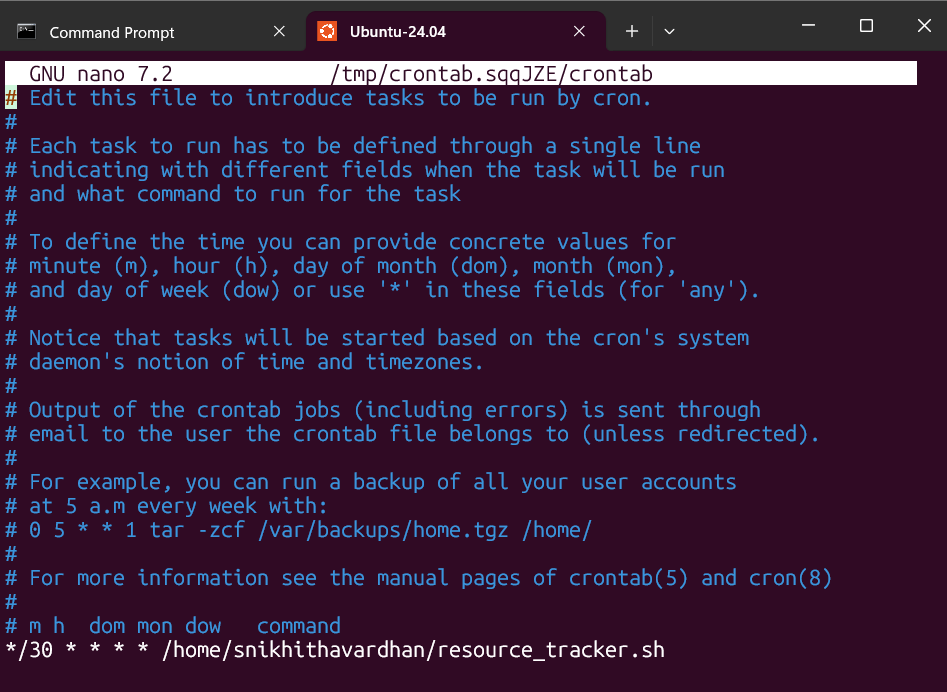


### **8. System Monitoring Script**

Create a script resource\_tracker.sh that records:

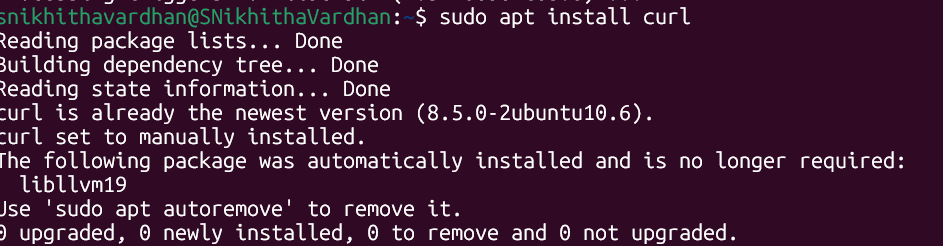
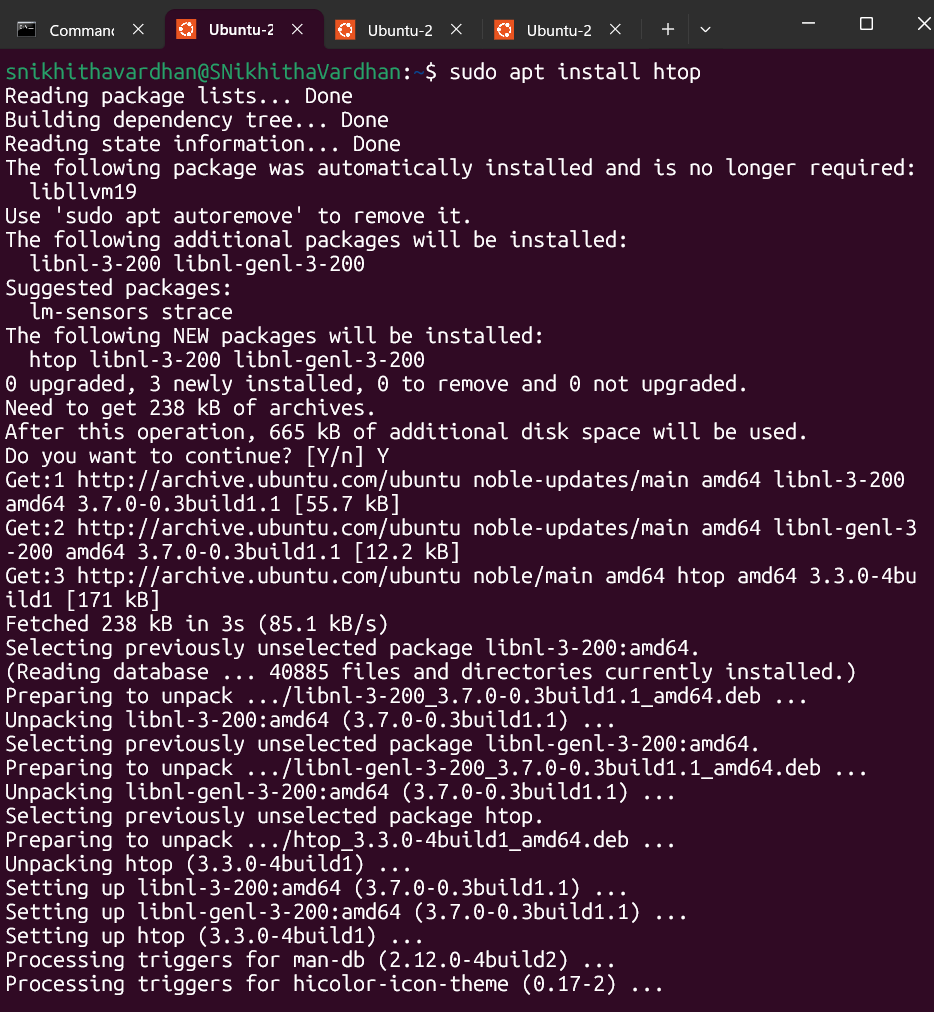
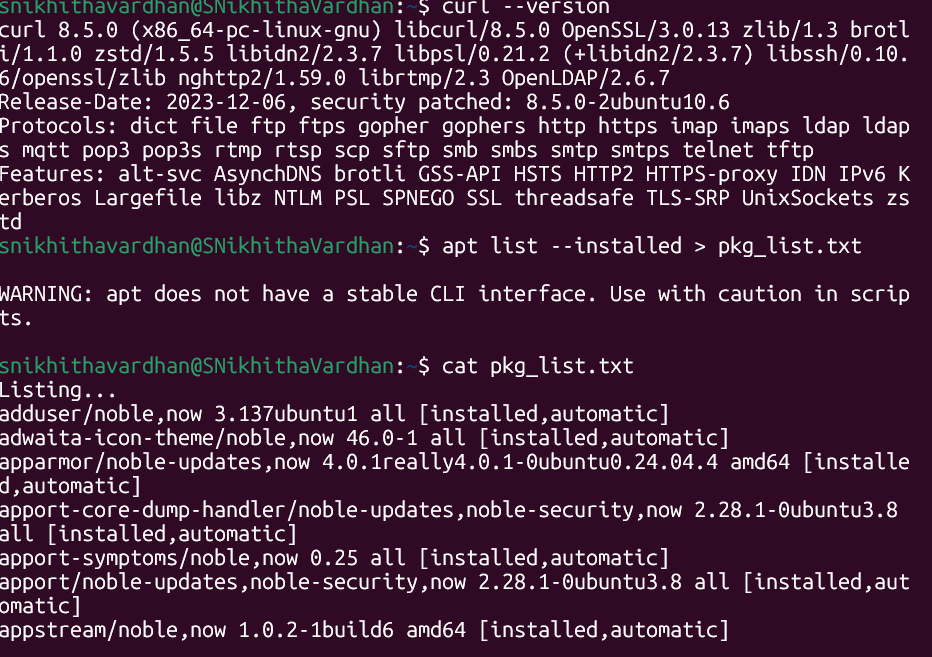
* Uptime
* CPU load average
* Memory usage
* Disk space utilization

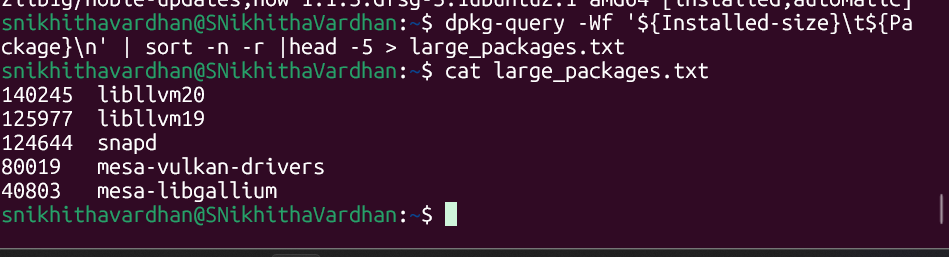
Append the logs to: ~/SysOpsLab/system\_status.log

**Bonus Challenge:** Automate the script with a **cron job** to run every 30 minutes.

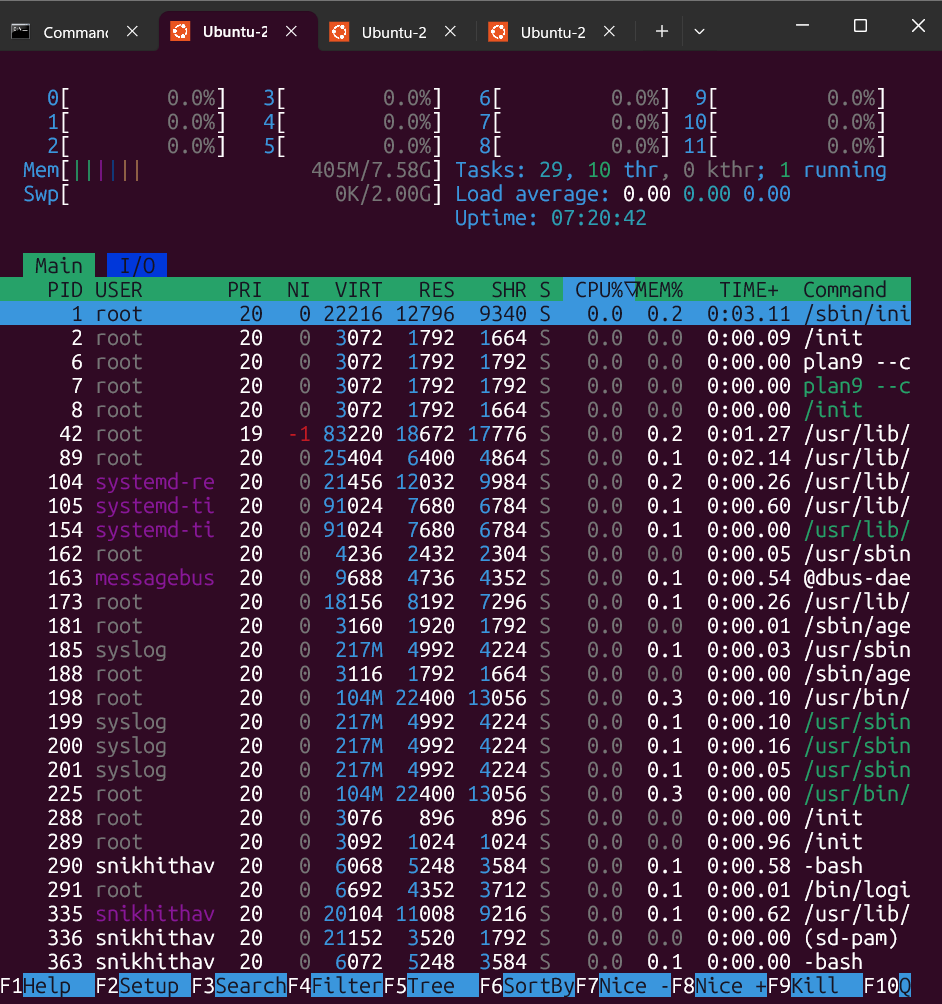
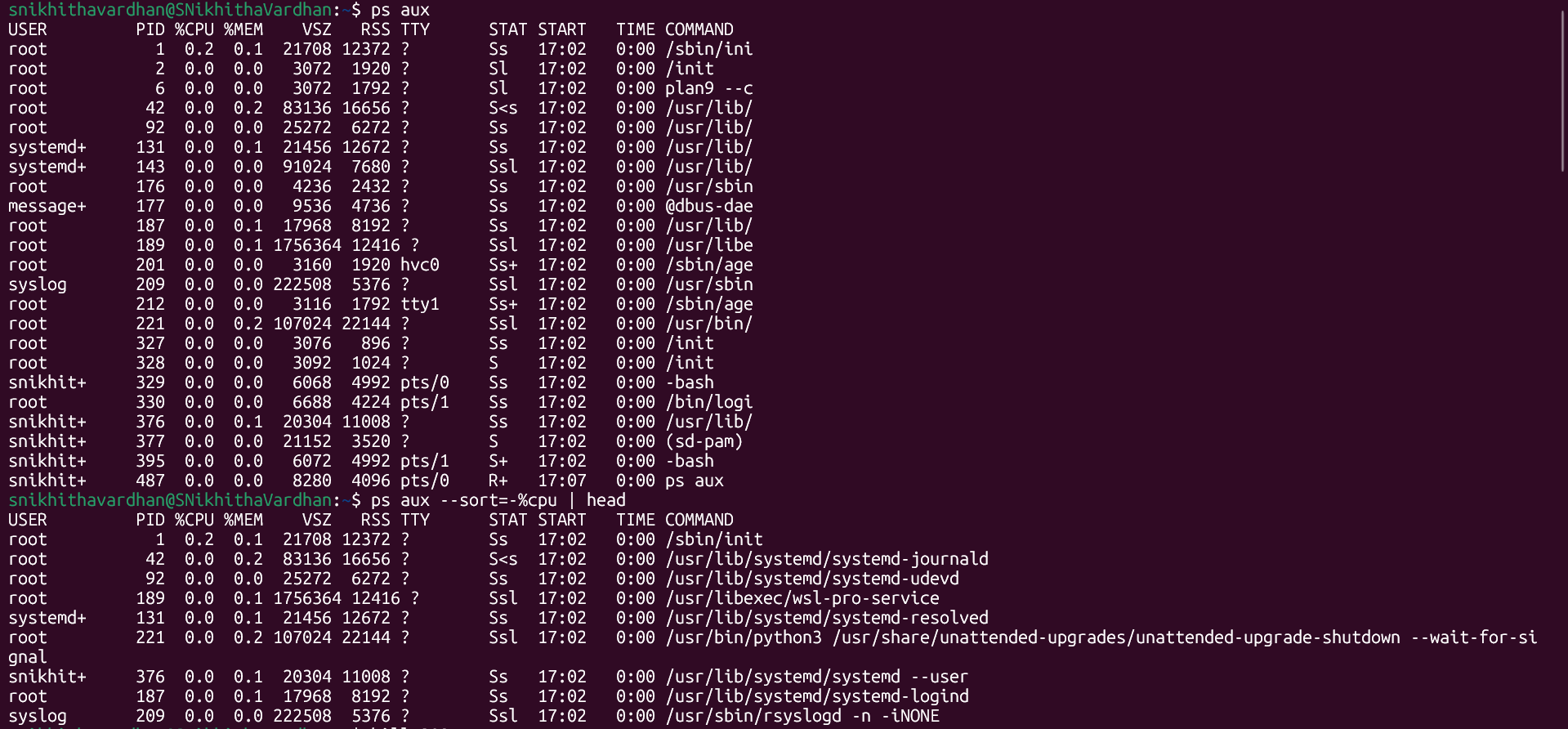
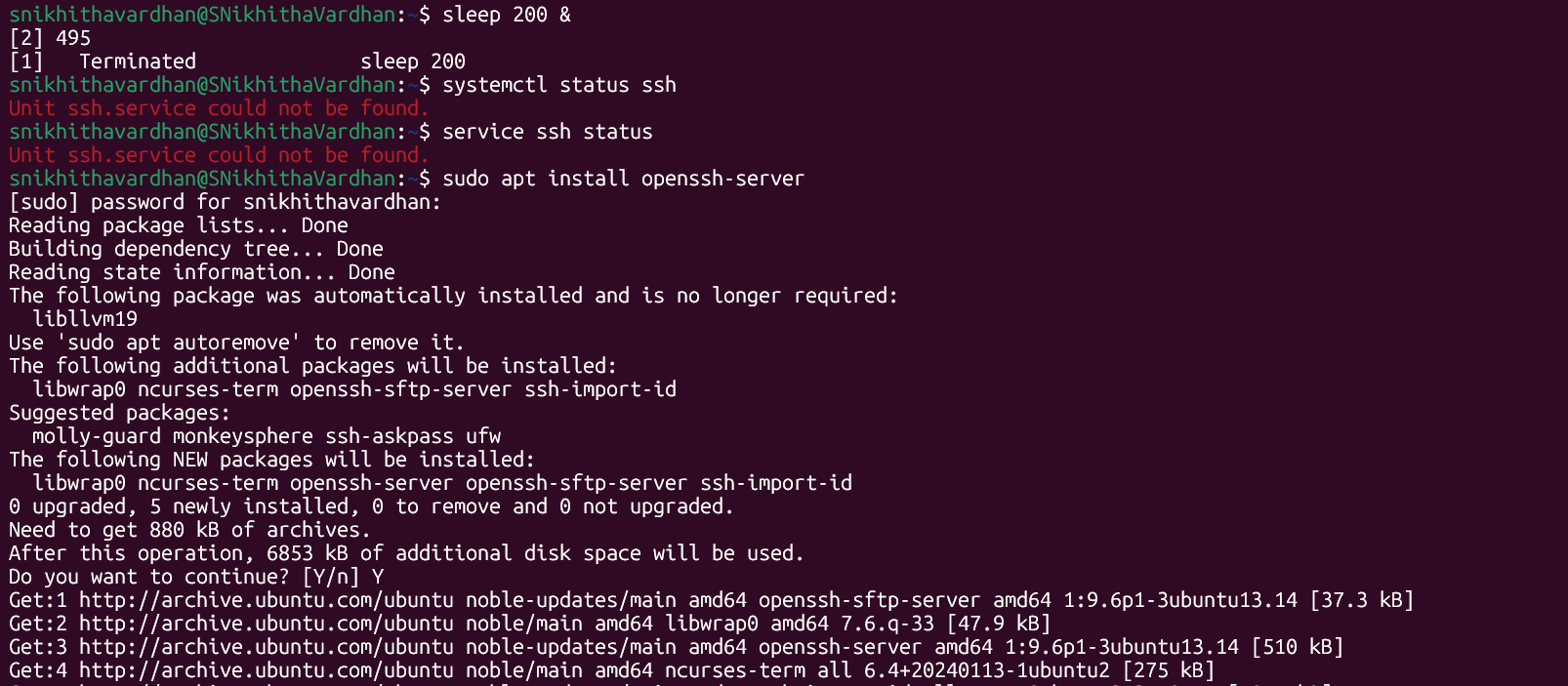
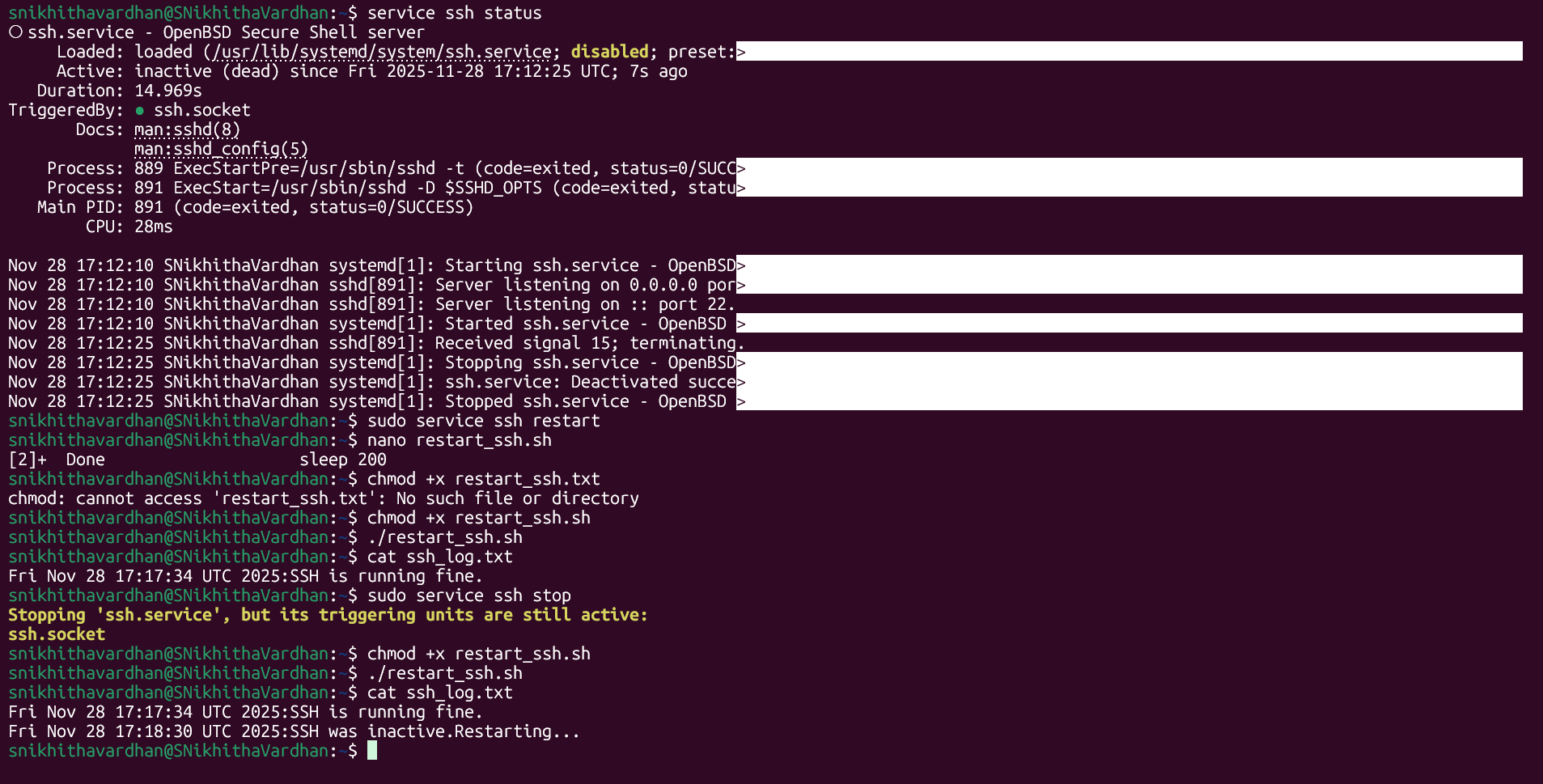
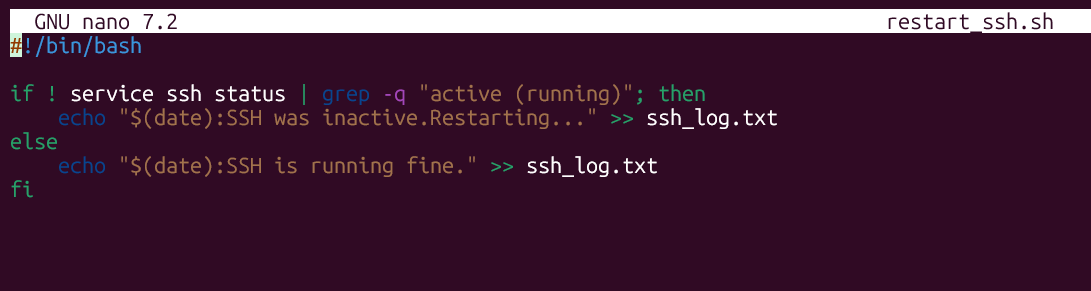
## **Section 5: Software and Process Management**

### **9. Package Operations**

* Update and upgrade all system packages.
* Install curl or htop and verify installation.
* List all installed packages and save to: pkg\_list.txt

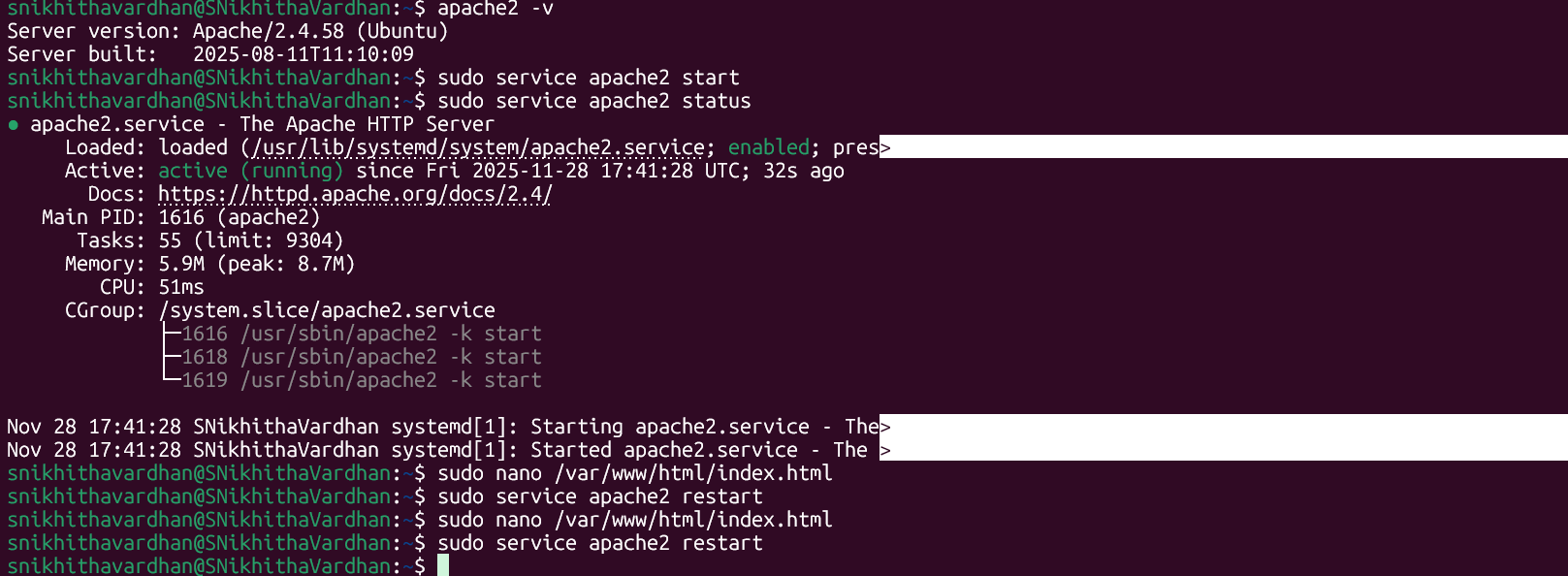
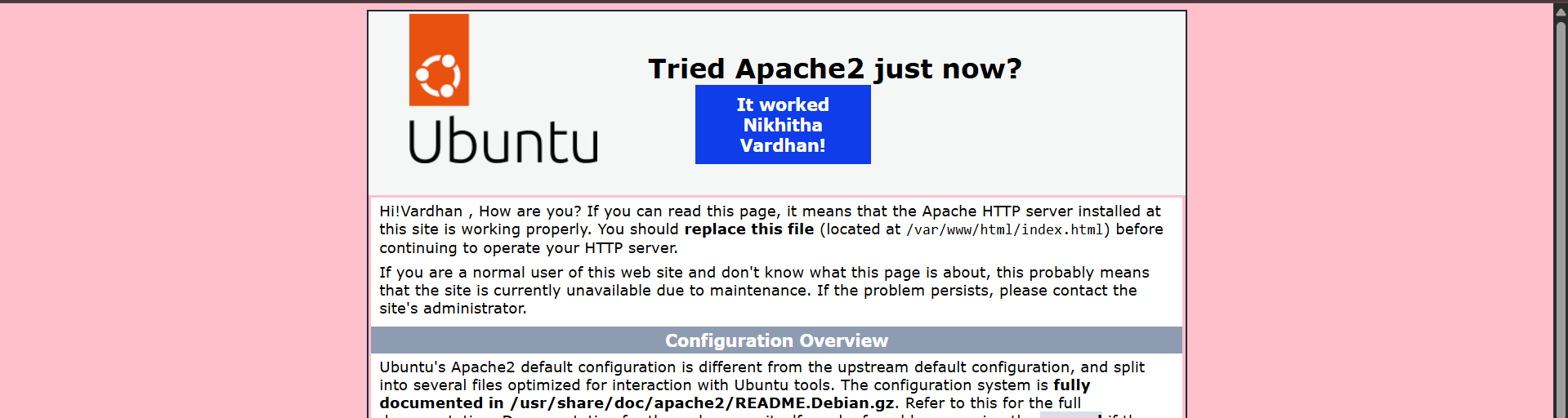
**Bonus Challenge:**Find and list the 5 largest installed packages (by size) into large\_packages.txt.

### **10. Process and Service Control**

* List all running processes.
* Identify top CPU-consuming processes.
* Stop and restart a background process using its PID.
* Check the status of a system service (e.g., ssh, cron).
* 
* **Bonus Challenge:**Use a script to automatically restart a service if it’s inactive.

## **Section 6: Web Hosting and Security**

### **11. Web Server Setup**

* Install and start **Apache** or **Nginx**.
* Verify it runs by visiting http://localhost or your server IP.
* Replace the default index page with a custom HTML message.
* Document each step taken.

**Bonus Challenge:** Enable a basic **firewall rule** (using ufw or firewalld) to allow only HTTP and SSH traffic.

## 

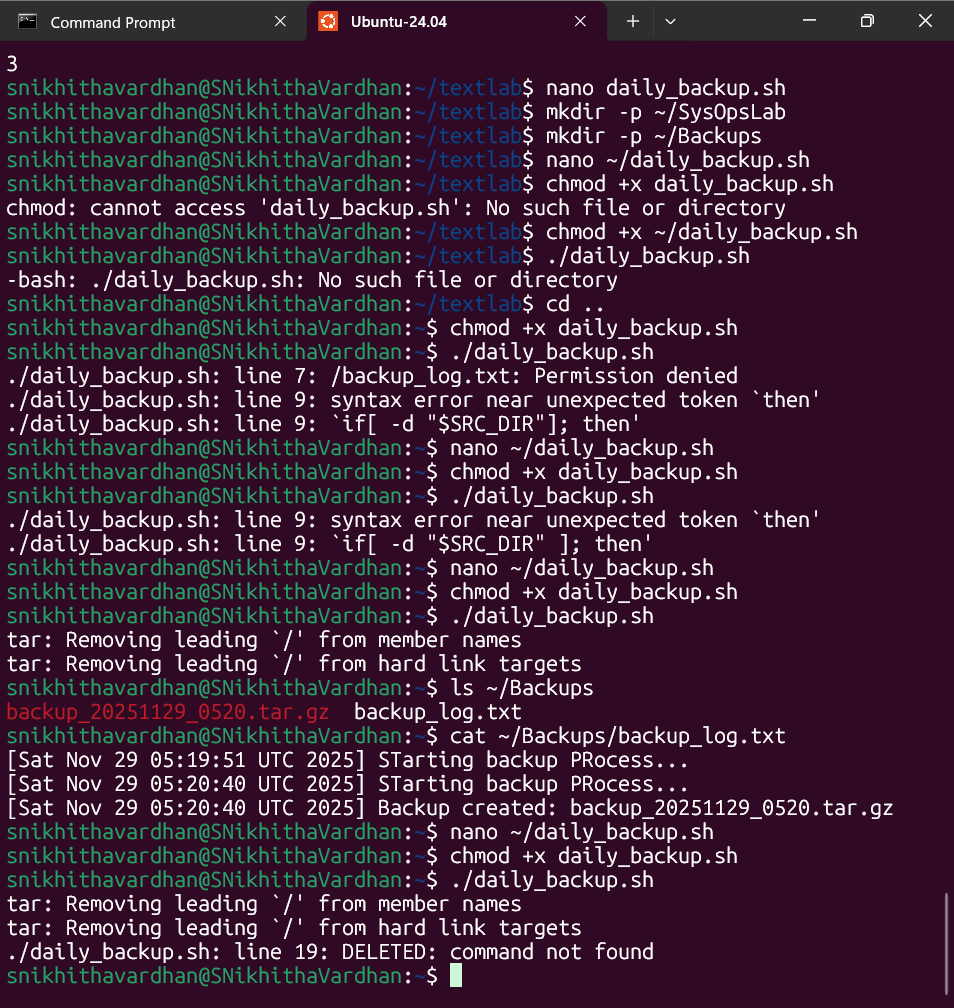
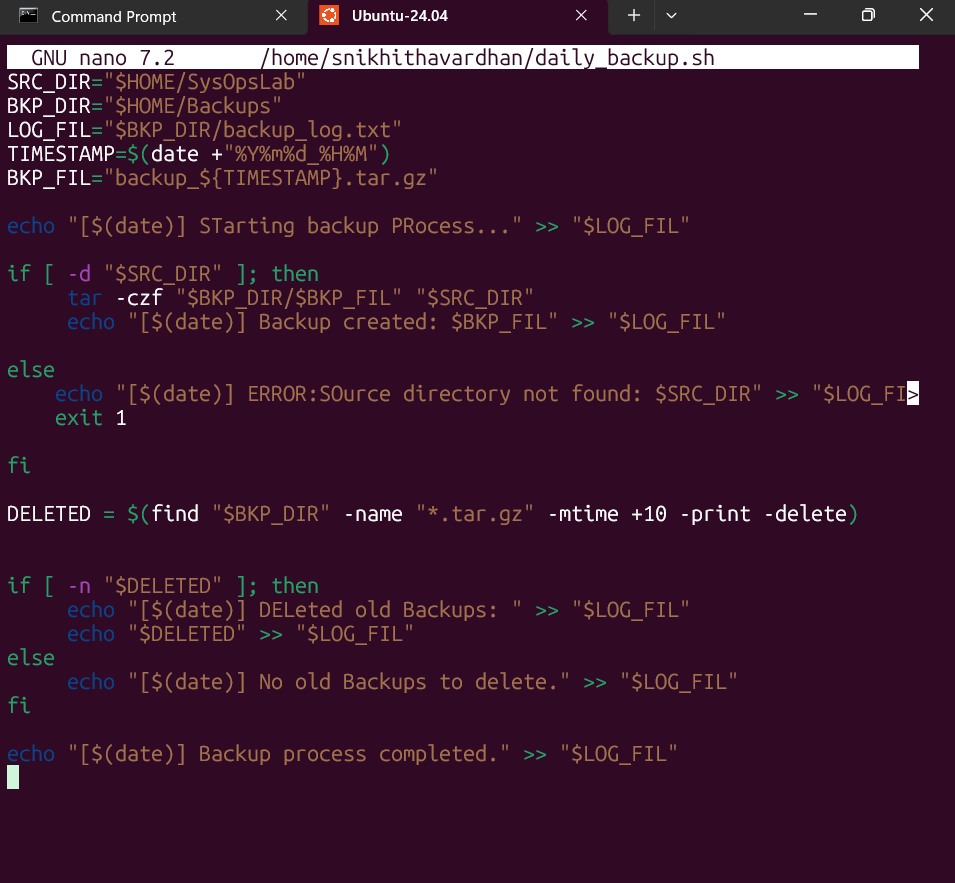
## 

## 

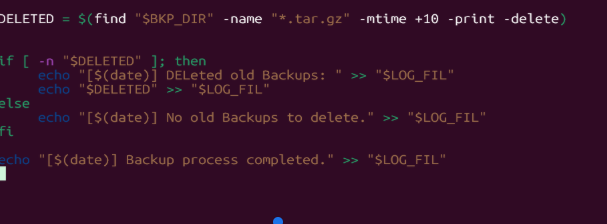
## **Section 7: Backup and Recovery**

### **12. Automated Backup Script**

* Create a script daily\_backup.sh that:
* Compresses the directory: ~/SysOpsLab/
* Saves it to: ~/Backups/
* Uses the filename format: backup\_YYYYMMDD\_HHMM.tar.gz



**Bonus Challenge:** Add logic to delete backup files older than 10 days and log all actions to backup\_log.txt.



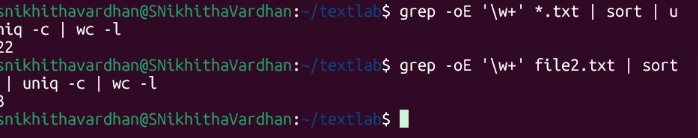
## **Section 8: Text Processing & File Manipulation**

### **13. Text Utilities Practice**

* Use grep, awk, and sed to:
  + Search for a specific word inside multiple files.
  + Replace a string pattern across files.
  + Print the 2nd column of a space-separated file.



**Bonus Challenge:** Combine multiple tools in a single command pipeline to count total unique words in a file.



**These are my notes that i made while learning about linux commands for this task -**

Pwd-print working directory-I can check where I am

Ls-list files: ls -l ,ls-a

Mkdir-make directory

Cd-open folders

Touch,nano - create files

Mv,rm,rmdir,cp-copy,move,remove ,remove dir

-p create parent folders - I can create nested folders at a time

-R recursive listing

Cp textfile name dest path not name

Move : mv file\_name dest\_folder

Cd .. - go to parent folder

snikhithavardhan@SNikhithaVardhan:~/SysOpsLab/Exercises/SessionA$ mv alpha.txt Archives  
This is wrong because it moved my file into archives folder and kept in SessionA so while moving a file keep the correct path not just name of Archives - because i am not in the actual parent folder of Archives

mv oldfolder newfolder : rename a folder

mv "old name.txt" "new name.txt" : for names with spaces

mv oldname.txt newname.txt : just move into a new txt file (has new name right)

The actual data stored is like inode

The file name is a label pointing to taht data

Inode 2202(or some random no.) is where teh actual data is stored

File name->inode 22035->actual content

Link:

A link is simply another name pointing to the SAME inode (data).

Two names pointing to one data , the actual name and the hard link what i create - these both r the ones pointing to the data , even if i accidentally delete one , i still have another label pointing to my inode right

Softlink - like a shortcut just points to the file name - if original si deleted then this also gone

The soft link pointing to a inode - it doesnt contain content so its a diff inode than actual inode

Ln command used to create links

Permissions : rw-r–r– first rw:owner , r : others and r:group

We cans ee only owner has read and write access

600 , 644 , 755 600:owner rw only , 644 rw owner but others read only ,755 all rw+exec

It has got this notation from binary form for read , write , exec each 1 or 0 and then merging numerical forms of owner , group , others

When ls -li

Total 0 means file size 0 as no content

sudo chown anotheruser private\_log.txt ===this is to change ownership to other user

Who/users == who is logged in now

Check hwo long has the system been up

Free -h : check RAM usage(h is human readable presents in GB/MB)

While writing contents in file :echo “ …”> replace contents in file and create if not there

And “...”>> append to the file contents

Everything inside { ... } runs sequentially.  
; separates each command.

Cat filename : can print contents to terminal

‘q’ - to quit anywhere

head -n 20 : shows first 20 lines.  
tail -n 20 : shows last 20 lines.

nano sysinfo\_report.txt : open in txt editor

vim sysinfo\_report.txt scrollable , readable,editable inside file

Sudo - “superuser do” - needed when system wide resources are being modified

sudo useradd -m analyst1 : as soon as this ask password of root not new passwd

If you are working inside your own user space or only checking things, you don’t need root privileges.

Sudo groupadd team\_name :This creates the group that will have access to your shared directory.

Modifying users in group : sudo usermod -aG datateam analyst1

Here aG will append this into team without erasing anything

sudo mkdir /datahub

That slash helps u create in root folder,else will be in my personal space and so i got that error

by now group can see and enter in file but still the ownership accessibility not limited to team yet so change it by   
Root:group

Right now, /datahub probably has default permissions (755), meaning:root → full access  
datateam :read + execute (can enter but cannot create files)  
others : read + execute

sudo usermod -aG datateam snikhithavardhan

Add myself in group

Logout and log in again

Root is not me , I can only modify my personal team —root is like a god who can do everything when am doing sudo its like am temporarily a root

Without SGID →  
 Everyone uploads files, but ownership is random → others can't edit.

With SGID →  
 Everything uploaded automatically belongs to the shared team → everyone can edit.

Drwxrw”s” taht s is like SGID flag

SGID - set group ID It is a special permission you apply to a directory

It means root is main owner but group can also have accessibility

Chown - change ownership

Su - analyst1:change user

Groups analyst1

Ls-l /datahub :Verify file inherits group datateam

ls-ld/datahub : verify directory SGID permissions

Audit tool is to record and report what happens on your system, especially security-related stuff.

What it tracks

* File access: read, write, delete, execute.
* User actions: logins, sudo usage, command executions.
* System events: network connections, process starts/stop

Since I ahvent dual booted my system - I am going with inotify in WSL ,

Inotifywait -m ~/foldername

This has started watching in 1 terminal and i have gone to terminal 2 to create some access activities and have observed those logs

Ctrl+c to exit from watch

Grep - command line tool

Global - regular expression -print

Grep pattern file\_name.txt(helps in searching for thar pattern inside my file)

inotifywait -m ~/secure\_vault >> audit\_log.txt

Redirected to output file so nothing printed here in terminal.

inotifywait -m ~/secure\_vault | tee -a audit\_log.txt

This helps in seeing livea nd also sending to output file

Ping would mean - sending packets to [google.com](http://google.com) and checking connectivity

-c 4 send only 4 requests

Each command does one job well. Pipes connect them to create more powerful workflows.

Taht’s why we use the | symbol

The 5th column in the display of sockets - have IP port address and all , so I want to extract that column and also where socket is listening

Now split it based on : so one yield ahs IP address and other has Port number and nwo the last field - that is the second field need to be printed

Grep LISTEN == can list out the sockets which are listening

Then i am moving to 5 th column as awk separates tehm by spaces default , then split that coloumna gain wit custom separtpr as : so now take last field NF(NF has number of fields right so gives the last field) as it gives me teh port number

awk '{print $5}' step2.txt > step2.txt- this is wrong because the shell empties teh file before readind it so use another ile there

I can do all this using pipes in single steps instead of having tehse many files to be created

Shell Scripting ::

Think of the Shell (like Bash) as the "steering wheel" of your computer's operating system. Usually, you type one command, press Enter, and the computer does one thing.

Shell Scripting is simply writing a list of those commands into a text file (the script) so the computer can execute them all at once, in order.

I hve opened a file with extension has .sh

The inspector\_log.txt file will be in the same directory where you ran the script.

#!/bin/bash- tells teh system to use bash to execute this file

I took a variable LOGFILE where i have stored the path od system log stud.txt , so that if i change name or so in future better and also clean code instead of writing it evry line

A cron job is like a scheduled task in Windows Task Scheduler.

It runs commands automatically at fixed time intervals .Linux uses a program called cron to maintain and execute these schedules.

crontab -e

\*/30 \* \* \* \* /home/snikhithavardhan/resource\_tracker.sh

Each of the 5 places note down as -

Minute field –hourfield–day of month–month–day of week

So I have filled first place and resta lla sa any values

My script of cron writes to system\_status.log

Systemctl status cron—to verify if cron is running or not

Crontab -l to view cron tasks

\*\*cron requires full path always

Updating means refreshing the package index (list of available software).

package management in Linux (APT).

curl stands for Client URL.

It is a command-line tool used to interact with web URLs.  
 You use it mainly to:

* Download files
* Test APIs or web servers
* Send HTTP requests (GET, POST, PUT, DELETE)
* Check network connectivity to web services

### What is htop?

htop is a system monitoring tool, like Task Manager on Windows but inside Linux terminal.

It shows:

* Running processes
* CPU usage
* RAM usage
* Swap usage
* Process tree
* Ability to kill tasks interactively

Dpkg-query can tell me about the package size , version,name status and all -Wf

We lsit out and make it give in a specified format : Installed size and package name on,y to be displayed

Sort -n -r (sort -numeric ordr - reverse(DESC))

Head -5 its like show top 5 results

Creating a dummy process so as to demonstrate the kill

SSH = Secure Shell

It is a tool/service that allows you to remotely connect to a computer securely using a terminal.

Examples of services in real Linux systems:

* ssh → remote login
* cron → scheduled tasks
* network-manager → handles internet
* mysql → database service

Check the status of a system service.

To do that, you need a real service installed. SSH is a simple and safe example.

So i have did shell scripting as to record when last time ssh service is started and if it si running now then shoes - running fine if not then , show last active date and then restart again

Lets become web server admin and see h ow websites run

The browser sends a request using HTTP

To your computer (localhost)

On port 80

Apache receives the request

Apache sends back your webpage (index.html)

The browser displays it

When you installed Apache, it automatically created:

/var/www/html/index.html

This file contains the default Apache welcome page you saw earlier.

That file is the one Apache shows when someone visits http://localhost.

Fire wall - http port 80 and ssh port 22

I go change for my cutomised website tehre only in index.html

* $1 = first column
* $2 = second
* $3 = third

Awk splits text by spaces automatically.

sed -i 's/banana/kiwi/g' \*.txt

Across all txt file i have 22 unique words

Uniq-c for unique and sort o resorting alphabetically(dupliactes appear togethr)

Wc -l == how many unique lines

\*.txt means across all files

‘\w+’

This is the regex pattern.

Basically allow words -oE allows regex and take out matching ones to teh regex patter grep si used ofr search

Tar -czf==compress it to .[tar.gz](http://tar.gz)

Restore backup == tar -xzf backupfile.tar.gz

Dont have absolute paths , cos when restoring it might overwrote taht complete folder tahts why tar gives a warning message

SRC\_DIR="$HOME/SysOpsLab"

Has path of source folder which is to be backed up

BKP\_DIR="$HOME/Backups"

Have teh folder to store backup files

LOG\_FILE="$BKP\_DIR/backup\_log.txt"

== a file inside bvackups folder so as to store the backup data

[ -d ] → “Is this a directory?”

exit 1 stops script and tells system the script failed

$variable means: use the value stored inside the variable

—-X X X—