1.Configure smtp in localhost.

Step 1: Install postfix

-> sudo apt update && sudo apt install postfix

Step 2:During installation i was prompted with a configuration screen

I chose "Internet site"

Step 3 : configure postfix

-> sudo nano /etc/postfix/main.cf

Step 4: start and enable the postfix

->sudo systemctl restart postfix

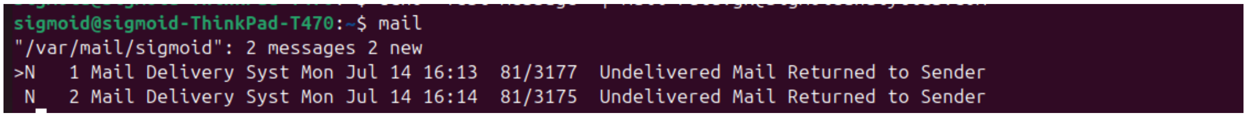
->systemctl enable postfix

Step 5: sending main to test from terminal

->echo "This is a test email body" | mail rutu.gk@sigmoidanalytics.com

Step 6: checking for mail in terminal

-> mail



```
sigmoid@sigmoid-ThinkPad-T470:~$ mail
"/var/mail/sigmoid": 2 messages 2 new
>N   1 Mail Delivery Syst Mon Jul 14 16:13  81/3177  Undelivered Mail Returned to Sender
 N   2 Mail Delivery Syst Mon Jul 14 16:14  81/3175  Undelivered Mail Returned to Sender
```

There was no error in the terminal, but I didn't receive any mail in my Gmail inbox (not even in Spam). Later, I found out that Postfix on localhost doesn't support sending to external emails by default. It needs to be configured to relay through an external SMTP server like Gmail, which I didn't set up.

2.Create a user in your localhost, which should not be able to execute the sudo command.

-> To **create a user in our localhost that should not be able to execute the sudo command follow the below mentioned steps:**

Step 1: Create a new user

->sudo useradd test

Step 2: Set a password

->sudo passwd test

Step 3: Check if the user belongs to sudo group

->groups test

Step 4: Change user from current user to test

->su test

Step 5: To verify that the test user has no sudo privileges run ls command

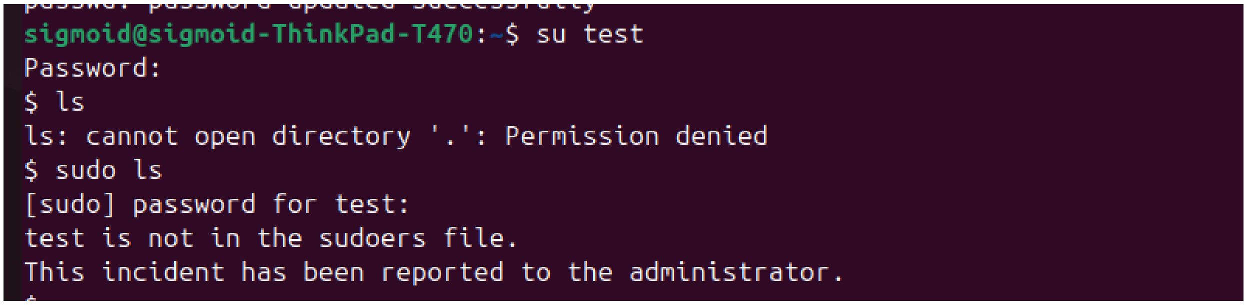
->ls

Step 6: You get a permission denied output that ensuring the test has no access over it.

Step 7: Try sudo ls command also

-> sudo ls

Step 8: The output displays that the test is not in sudoers file.



```
passwd: password updated successfully
sigmoid@sigmoid-ThinkPad-T470:~$ su test
Password:
$ ls
ls: cannot open directory '.': Permission denied
$ sudo ls
[sudo] password for test:
test is not in the sudoers file.
This incident has been reported to the administrator.
```

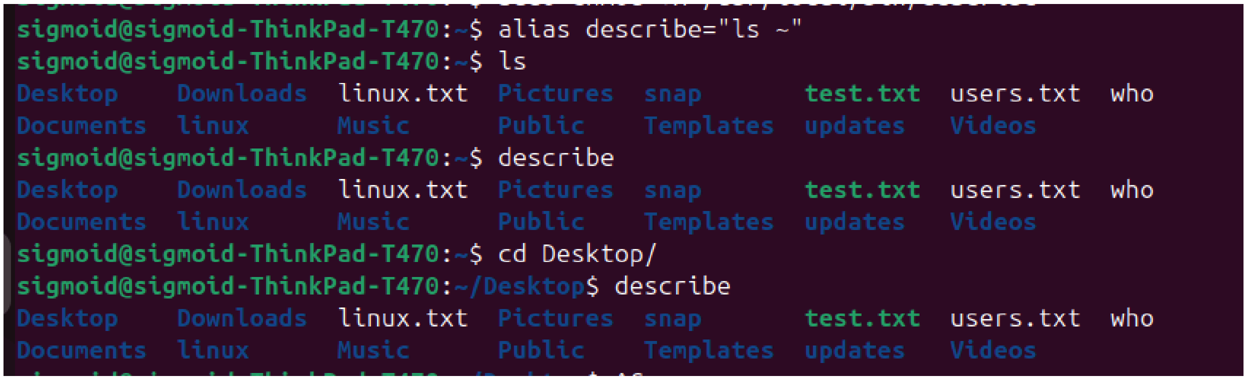
3.Configure your system in such a way that when a user type and executes a describe command from anywhere of the system it must list all the files and folders of the user's current directory.

Ex:- $ describe

$ content1 content2

Content3 content 4

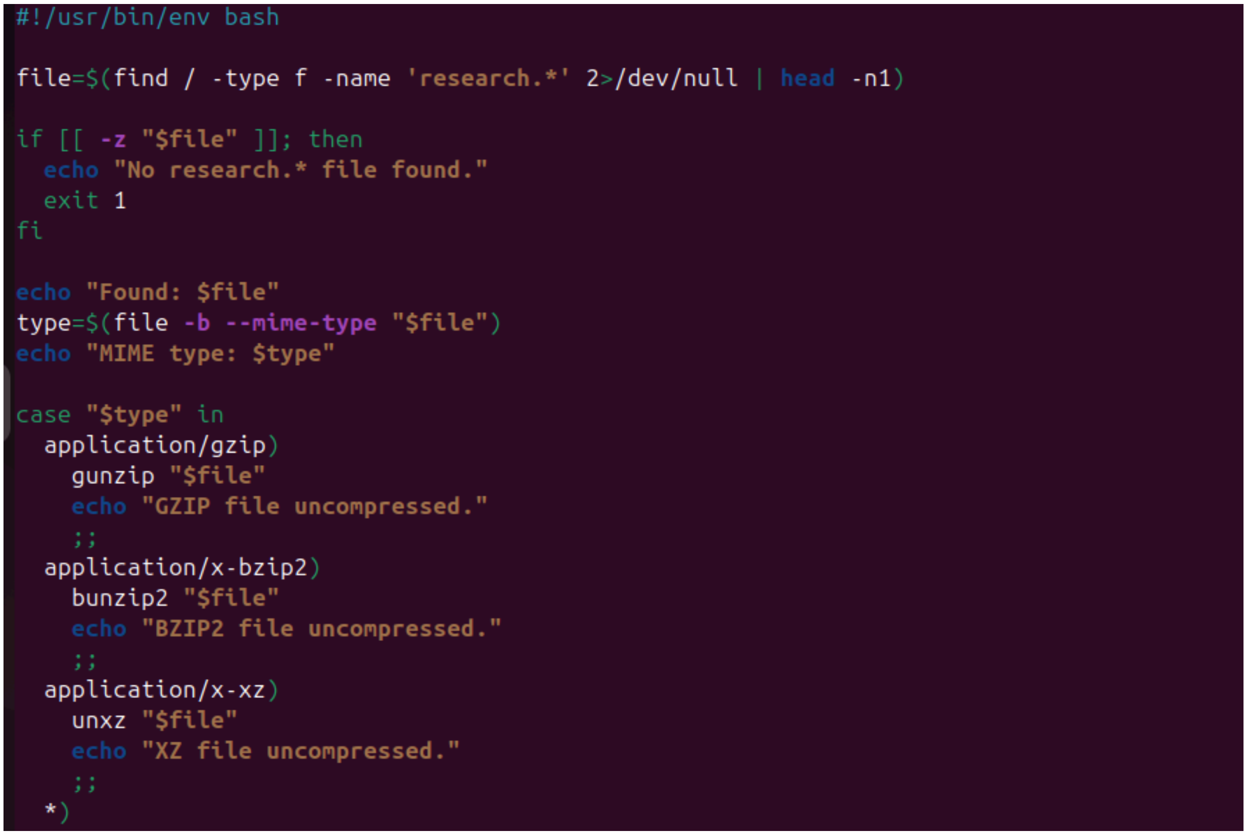
Step 1:alias ls as describe
->alias describe="ls ~"
Step 2:Run describe to check if it shows home directory contents.
->describe
Step 3: Now change the directory to any of the option say Desktop .
->cd Desktop/
Step 4: In the Desktop directory give describe to view all files and folders of the current directory.



```
sigmoid@sigmoid-ThinkPad-T470:~$ alias describe="ls ~"
sigmoid@sigmoid-ThinkPad-T470:~$ ls
Desktop    Downloads  linux.txt  Pictures  snap       test.txt  users.txt  who
Documents  linux      Music      Public    Templates  updates   Videos
sigmoid@sigmoid-ThinkPad-T470:~$ describe
Desktop    Downloads  linux.txt  Pictures  snap       test.txt  users.txt  who
Documents  linux      Music      Public    Templates  updates   Videos
sigmoid@sigmoid-ThinkPad-T470:~$ cd Desktop/
sigmoid@sigmoid-ThinkPad-T470:~/Desktop$ describe
Desktop    Downloads  linux.txt  Pictures  snap       test.txt  users.txt  who
Documents  linux      Music      Public    Templates  updates   Videos
```

4.Users can put a compressed file at any path of the linux file system. The name of the file will be research and the extension will be of compression type, example for gzip type extension will be .gz. You have to find the file and check the compression type and uncompress it.

Step 1. Created a Script:
->sudo nano /usr/local/bin/unpack-research
Script Content:



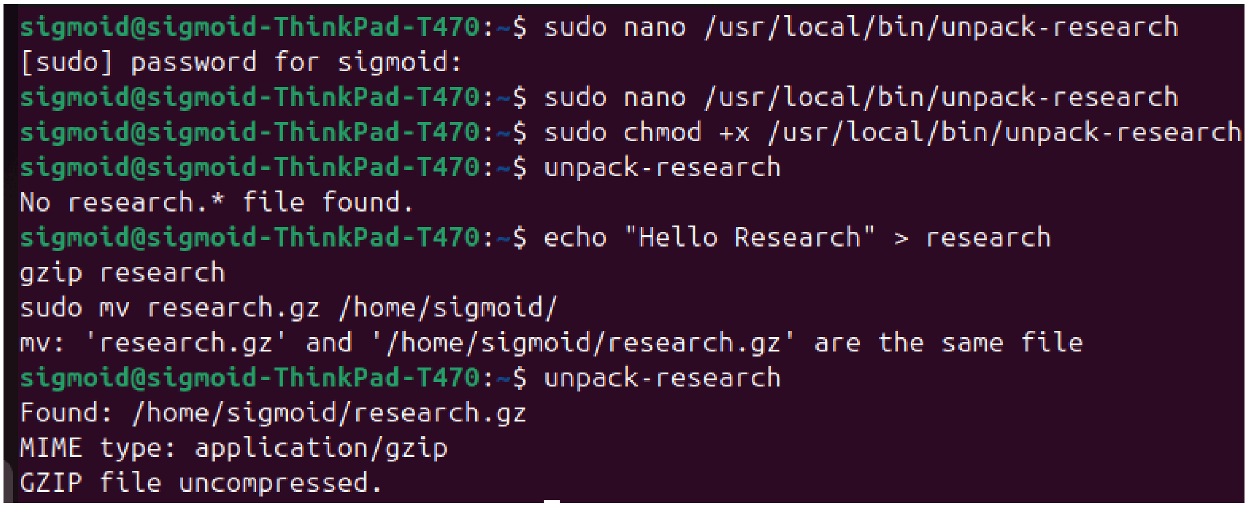
```
#!/usr/bin/env bash

file=$(find / -type f -name 'research.*' 2>/dev/null | head -n1)

if [[ -z "$file" ]]; then
  echo "No research.* file found."
  exit 1
fi

echo "Found: $file"
type=$(file -b --mime-type "$file")
echo "MIME type: $type"

case "$type" in
  application/gzip)
    gunzip "$file"
    echo "GZIP file uncompressed."
    ;;
  application/x-bzip2)
    bunzip2 "$file"
    echo "BZIP2 file uncompressed."
    ;;
  application/x-xz)
    unxz "$file"
    echo "XZ file uncompressed."
    ;;
  *)
```

Step 2:Made the Script Executable
->sudo chmod +x /usr/local/bin/unpack-research

Step 3:Ran the Script
->unpack-research

Step 4:Created a dummy compressed file:
->echo "Hello Research" > research
gzip research
sudo mv research.gz /home/sigmoid/

Output Terminal:



```
sigmoid@sigmoid-ThinkPad-T470:~$ sudo nano /usr/local/bin/unpack-research
[sudo] password for sigmoid:
sigmoid@sigmoid-ThinkPad-T470:~$ sudo nano /usr/local/bin/unpack-research
sigmoid@sigmoid-ThinkPad-T470:~$ sudo chmod +x /usr/local/bin/unpack-research
sigmoid@sigmoid-ThinkPad-T470:~$ unpack-research
No research.* file found.
sigmoid@sigmoid-ThinkPad-T470:~$ echo "Hello Research" > research
gzip research
sudo mv research.gz /home/sigmoid/
mv: 'research.gz' and '/home/sigmoid/research.gz' are the same file
sigmoid@sigmoid-ThinkPad-T470:~$ unpack-research
Found: /home/sigmoid/research.gz
MIME type: application/gzip
GZIP file uncompressed.
```

5.Configure your system in such a way that any user of your system creates a file then there should not be permission to do any activity in that file.

Step 1: Created a Special Folder Called `/restricted`
Since I didn't want to apply this restriction system-wide, I created a specific directory where the restriction would apply:
->sudo mkdir /restricted

Step 2: Gave Ownership of the Folder to the User `rutu`

I wanted the user `rutu` to be able to create files in `/restricted`, so I changed the folder ownership:

->sudo chown supriya2 /restricted

Now, the user can enter the folder and create files inside it.

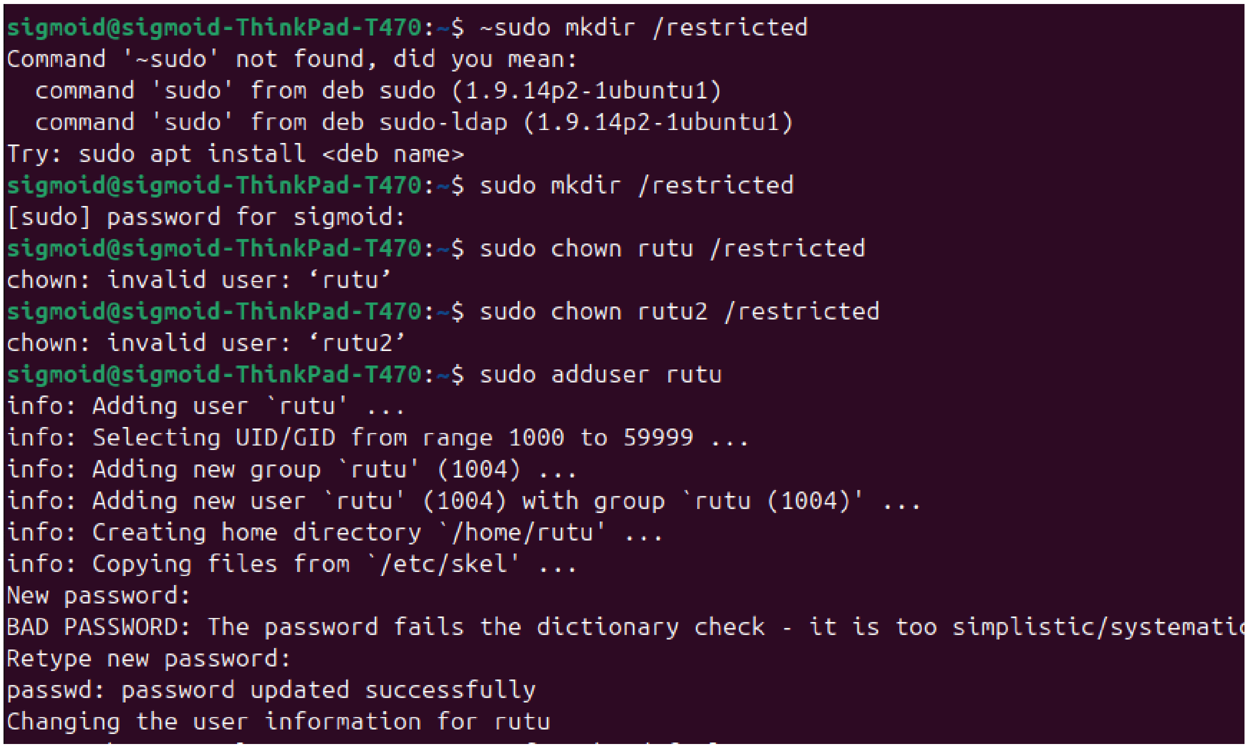
Step 3: Used ACL to Block All Permissions for the File Creator (Without Using `chmod`)
To make sure that any file created inside `/restricted` will have **no permissions** for the creator (user), I used **Access Control Lists (ACL)**:
->sudo setfacl -d -m u::0 /restricted

Step 4: Switched to User and Tested It
Switched to user `rutu`:
->su - supriya2
->cd /restricted

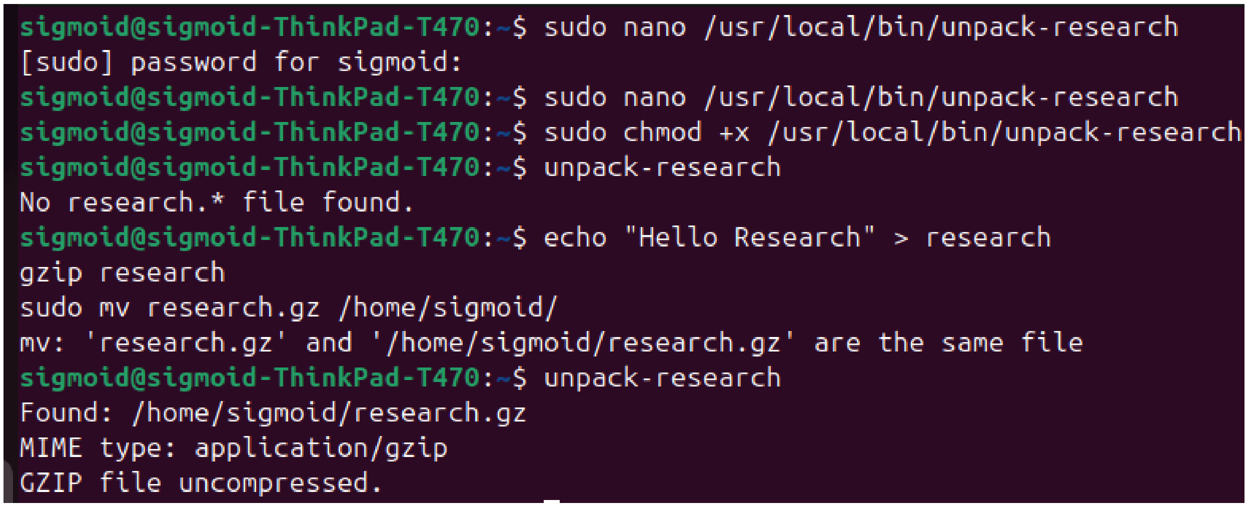
Then created a file:
->touch secret.txt
Checked its permissions:
->ls -l

Step 5: Tried Reading and Writing the File — Both Failed!
I tested by trying to read the file:
->cat secret.txt

But got the error:
->cat: secret.txt: Permission denied
This confirmed that the ACL rule worked — the user who created the file **could not access it** themselves.



```
sigmoid@sigmoid-ThinkPad-T470:~$ ~sudo mkdir /restricted
Command '~sudo' not found, did you mean:
  command 'sudo' from deb sudo (1.9.14p2-1ubuntu1)
  command 'sudo' from deb sudo-ldap (1.9.14p2-1ubuntu1)
Try: sudo apt install <deb name>
sigmoid@sigmoid-ThinkPad-T470:~$ sudo mkdir /restricted
[sudo] password for sigmoid:
sigmoid@sigmoid-ThinkPad-T470:~$ sudo chown rutu /restricted
chown: invalid user: ‘rutu’
sigmoid@sigmoid-ThinkPad-T470:~$ sudo chown rutu2 /restricted
chown: invalid user: ‘rutu2’
sigmoid@sigmoid-ThinkPad-T470:~$ sudo adduser rutu
info: Adding user `rutu' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `rutu' (1004) ...
info: Adding new user `rutu' (1004) with group `rutu (1004)' ...
info: Creating home directory `/home/rutu' ...
info: Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
passwd: password updated successfully
Changing the user information for rutu
```



```
sigmoid@sigmoid-ThinkPad-T470:~$ sudo nano /usr/local/bin/unpack-research
[sudo] password for sigmoid:
sigmoid@sigmoid-ThinkPad-T470:~$ sudo nano /usr/local/bin/unpack-research
sigmoid@sigmoid-ThinkPad-T470:~$ sudo chmod +x /usr/local/bin/unpack-research
sigmoid@sigmoid-ThinkPad-T470:~$ unpack-research
No research.* file found.
sigmoid@sigmoid-ThinkPad-T470:~$ echo "Hello Research" > research
gzip research
sudo mv research.gz /home/sigmoid/
mv: 'research.gz' and '/home/sigmoid/research.gz' are the same file
sigmoid@sigmoid-ThinkPad-T470:~$ unpack-research
Found: /home/sigmoid/research.gz
MIME type: application/gzip
GZIP file uncompressed.
```

6.Create a service with the name showtime , after starting the service, every minute it should print the current time in a file in the user home directory.

->1. Created the Script

Created a script that writes the current time to a log file every 60 seconds:

->sudo nano /usr/local/bin/showtime.sh

Script Content:

```
#!/bin/bash
while true; do
  date >> /home/$USER/showtime.log
  sleep 60
done
```

Made the script executable:

sudo chmod +x /usr/local/bin/showtime.sh

2.Created the systemd Service File

Created a custom service file:

->sudo nano /etc/systemd/system/showtime.service

Service File Content:

```
[Unit]
```

```
Description=Showtime Service - Log current time every minute
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/showtime.sh
Restart=always
User=sigmoid

[Install]
WantedBy=multi-user.target
```

3. Reloaded Systemd and Started the Service

Reloaded systemd to recognize the new service:

```
->sudo systemctl daemon-reload
```

Enabled it to start on boot:

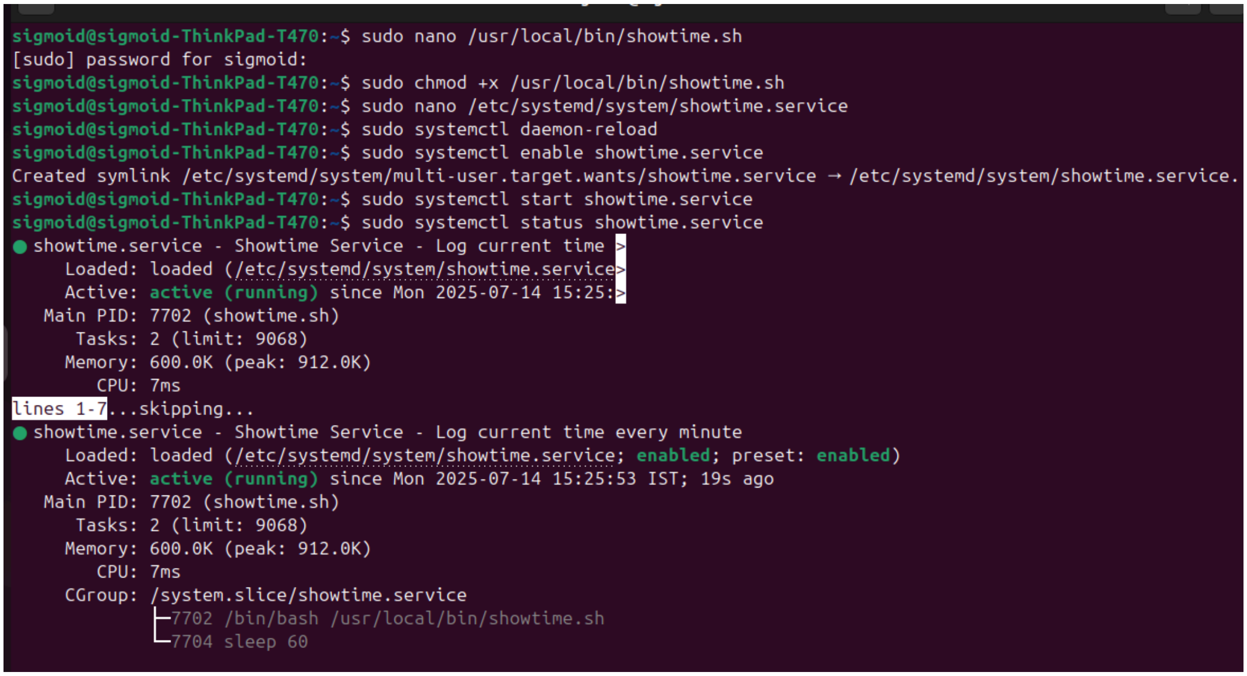
```
->sudo systemctl enable showtime.service
```

Started the service:

```
->sudo systemctl start showtime.service
```

Checked service status:

```
->sudo systemctl status showtime.service
```



```
sigmoid@sigmoid-ThinkPad-T470:~$ sudo nano /usr/local/bin/showtime.sh
[sudo] password for sigmoid:
sigmoid@sigmoid-ThinkPad-T470:~$ sudo chmod +x /usr/local/bin/showtime.sh
sigmoid@sigmoid-ThinkPad-T470:~$ sudo nano /etc/systemd/system/showtime.service
sigmoid@sigmoid-ThinkPad-T470:~$ sudo systemctl daemon-reload
sigmoid@sigmoid-ThinkPad-T470:~$ sudo systemctl enable showtime.service
Created symlink /etc/systemd/system/multi-user.target.wants/showtime.service → /etc/systemd/system/showtime.service.
sigmoid@sigmoid-ThinkPad-T470:~$ sudo systemctl start showtime.service
sigmoid@sigmoid-ThinkPad-T470:~$ sudo systemctl status showtime.service
● showtime.service - Showtime Service - Log current time >
     Loaded: loaded (/etc/systemd/system/showtime.service>
     Active: active (running) since Mon 2025-07-14 15:25:>
   Main PID: 7702 (showtime.sh)
      Tasks: 2 (limit: 9068)
     Memory: 600.0K (peak: 912.0K)
        CPU: 7ms
lines 1-7...skipping...
● showtime.service - Showtime Service - Log current time every minute
     Loaded: loaded (/etc/systemd/system/showtime.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-07-14 15:25:53 IST; 19s ago
   Main PID: 7702 (showtime.sh)
      Tasks: 2 (limit: 9068)
     Memory: 600.0K (peak: 912.0K)
        CPU: 7ms
     CGroup: /system.slice/showtime.service
             ├─7702 /bin/bash /usr/local/bin/showtime.sh
             └─7704 sleep 60
```