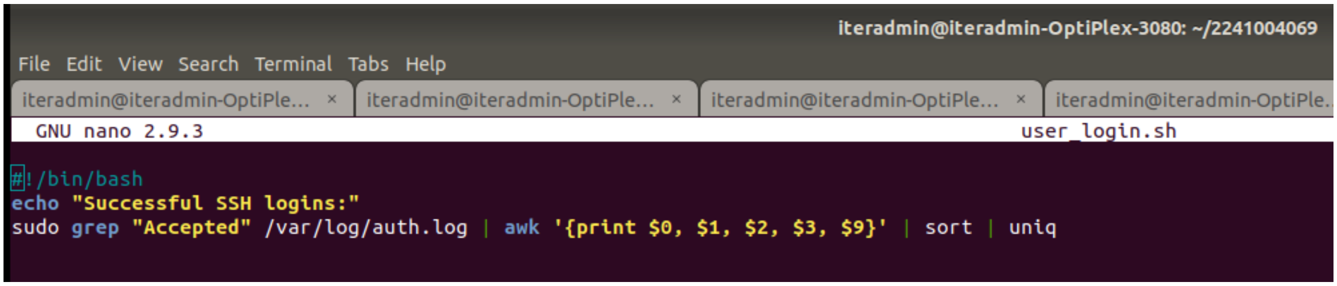
Assignment-1

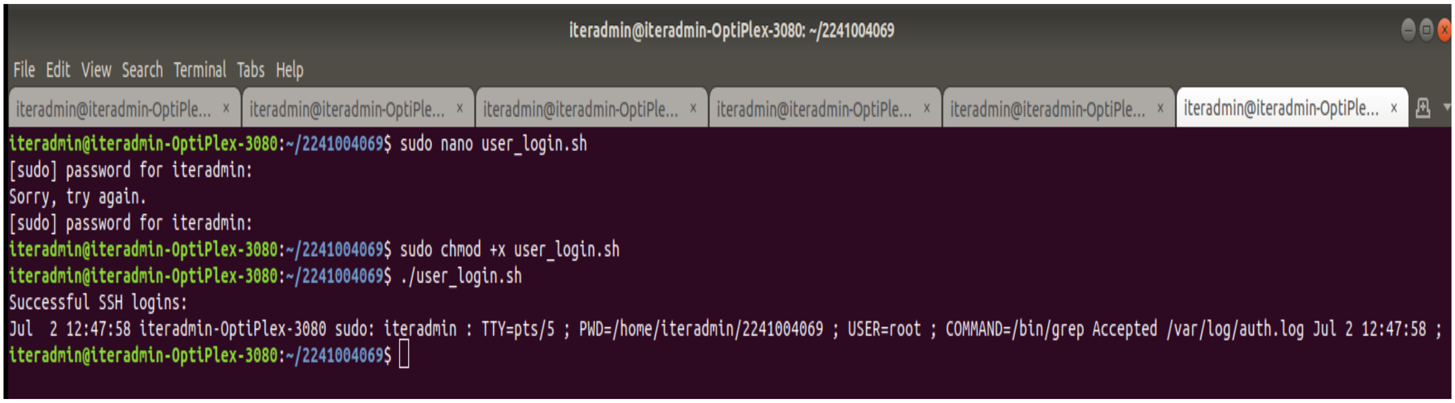
Name- Kiran Das Date:30-June-2025

Q1. Write a script to read the /var/log/auth.log file and extract the list of successful SSH login attempts using grep, awk, or cut. Display the unique usernames and their login timestamps.

Ans:


iteradmin@iteradmin-OptiPlex-3080: ~/2241004069
File Edit View Search Terminal Tabs Help
iteradmin@iteradmin-OptiPle... ×
iteradmin@iteradmin-OptiPle... ×
iteradmin@iteradmin-OptiPle... ×
iteradmin@iteradmin-OptiPle..
GNU nano 2.9.3
user_login.sh

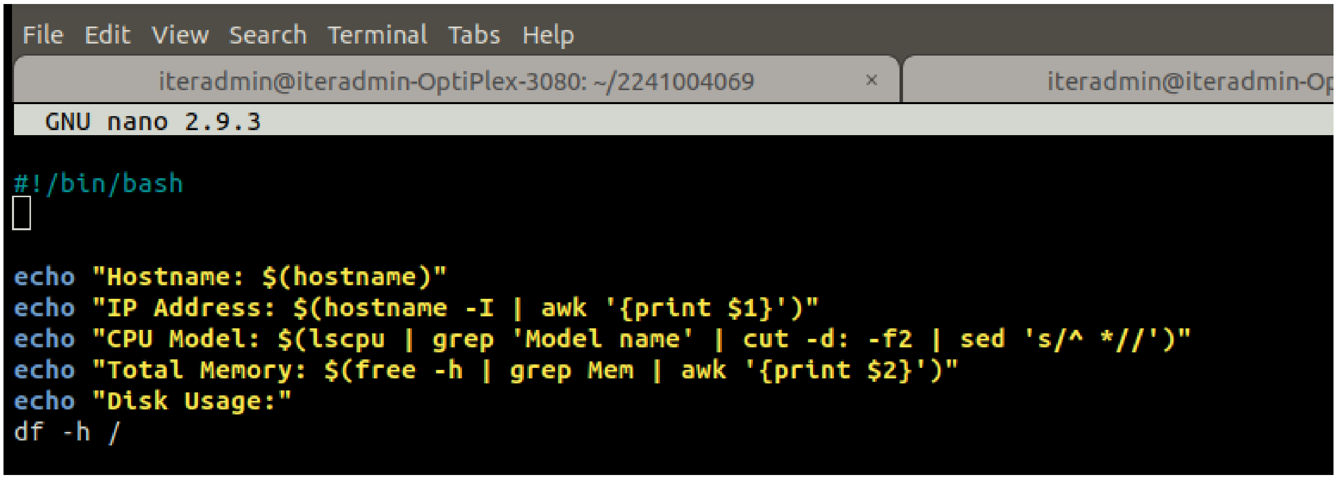
```
#!/bin/bash
echo "Successful SSH logins:"
sudo grep "Accepted" /var/log/auth.log | awk '{print $0, $1, $2, $3, $9}' | sort | uniq
```


iteradmin@iteradmin-OptiPlex-3080: ~/2241004069
File Edit View Search Terminal Tabs Help
iteradmin@iteradmin-OptiPle... ×
iteradmin@iteradmin-OptiPle... ×
iteradmin@iteradmin-OptiPle... ×
iteradmin@iteradmin-OptiPle... ×
iteradmin@iteradmin-OptiPle... ×
iteradmin@iteradmin-OptiPle... ×

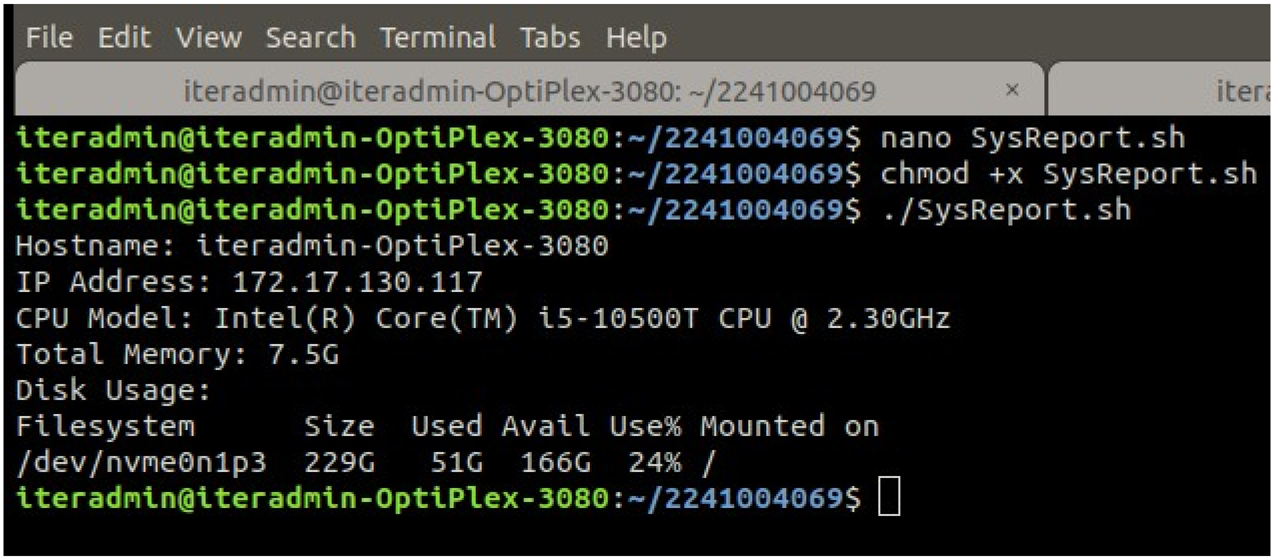
```
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ sudo nano user_login.sh
[sudo] password for iteradmin:
Sorry, try again.
[sudo] password for iteradmin:
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ sudo chmod +x user_login.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ ./user_login.sh
Successful SSH logins:
Jul  2 12:47:58 iteradmin-OptiPlex-3080 sudo: iteradmin : TTY=pts/5 ; PWD=/home/iteradmin/2241004069 ; USER=root ; COMMAND=/bin/grep Accepted /var/log/auth.log Jul 2 12:47:58 ;
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ 
```

Q2. Create a script that gathers and displays system information: hostname, IP address, CPU model, total memory, and disk usage. Use commands like uname, hostname, ifconfig, free, and df.

Ans:


File Edit View Search Terminal Tabs Help
iteradmin@iteradmin-OptiPlex-3080: ~/2241004069
iteradmin@iteradmin-Op
GNU nano 2.9.3

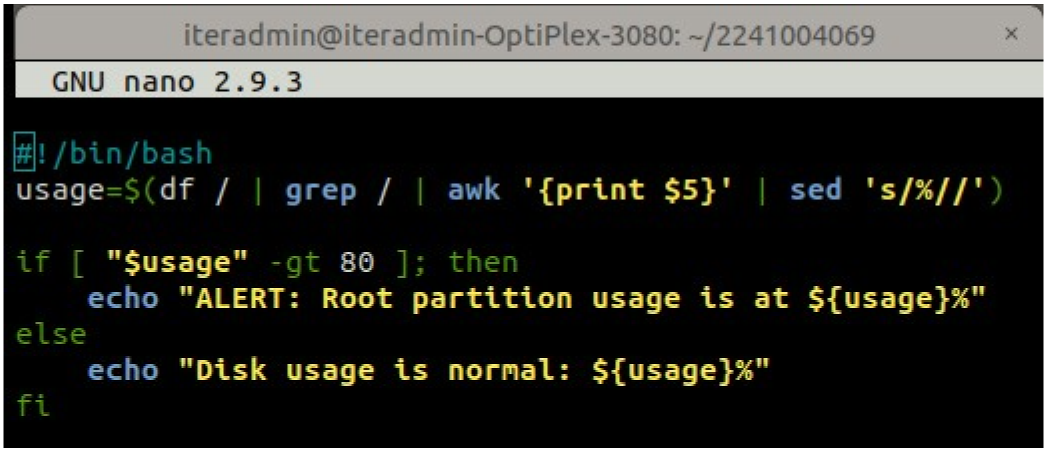
```
#!/bin/bash


echo "Hostname: $(hostname)"
echo "IP Address: $(hostname -I | awk '{print $1}')"
echo "CPU Model: $(lscpu | grep 'Model name' | cut -d: -f2 | sed 's/^ *//')"
echo "Total Memory: $(free -h | grep Mem | awk '{print $2}')"
echo "Disk Usage:"
df -h /
```


File Edit View Search Terminal Tabs Help
iteradmin@iteradmin-OptiPlex-3080: ~/2241004069
itera

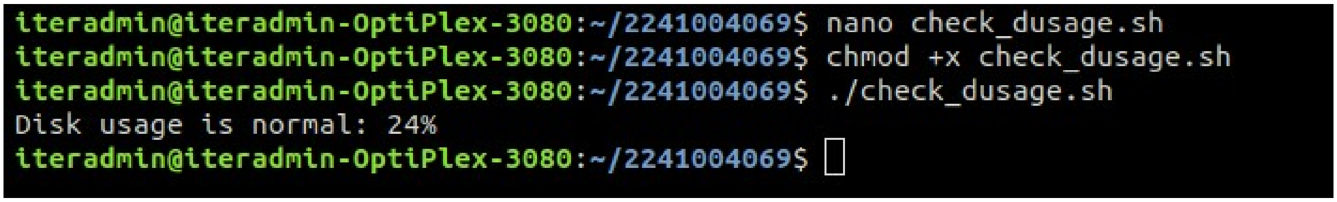
```
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ nano SysReport.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ chmod +x SysReport.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ ./SysReport.sh
Hostname: iteradmin-OptiPlex-3080
IP Address: 172.17.130.117
CPU Model: Intel(R) Core(TM) i5-10500T CPU @ 2.30GHz
Total Memory: 7.5G
Disk Usage:
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p3  229G   51G  166G  24% /
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$
```

Q3. Write a script that checks the disk usage of root (/) partition. If the usage exceeds 80%, print an alert. Use df, awk, and if conditionals.

Ans:



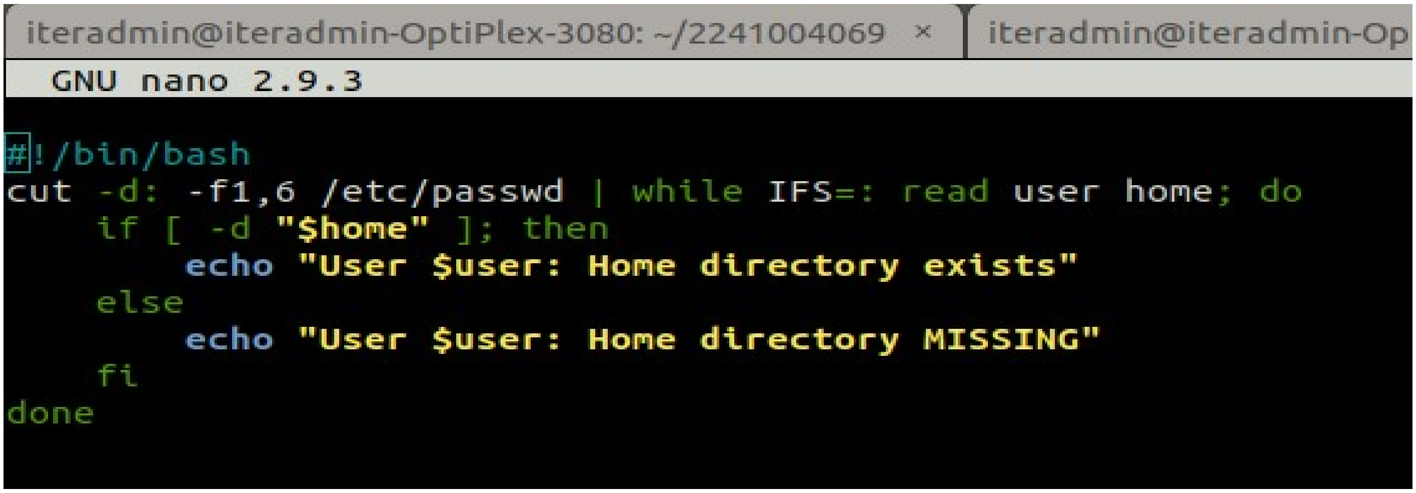
```
#!/bin/bash
usage=$(df / | grep / | awk '{print $5}' | sed 's/%//')

if [ "$usage" -gt 80 ]; then
    echo "ALERT: Root partition usage is at ${usage}%"
else
    echo "Disk usage is normal: ${usage}%"
fi
```



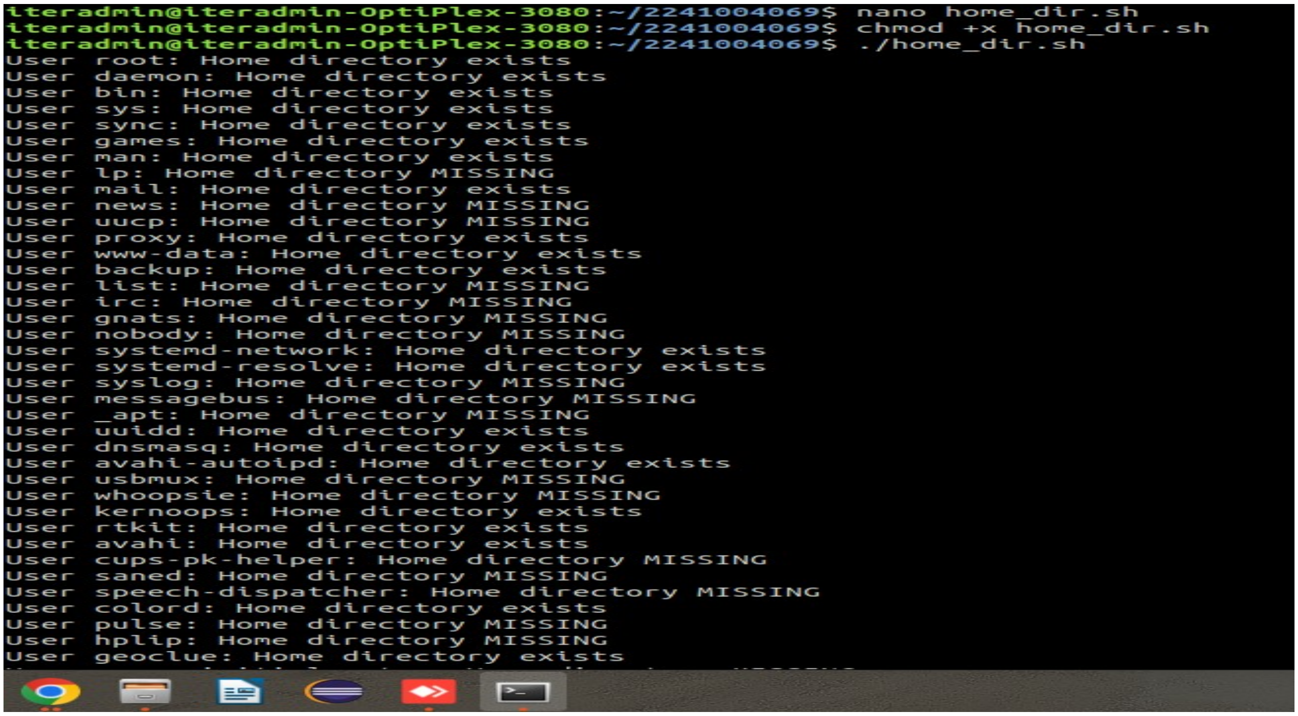
```
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ nano check_dusage.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ chmod +x check_dusage.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ ./check_dusage.sh
Disk usage is normal: 24%
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ 
```

Q4. Loop through all users in /etc/passwd, extract their usernames and home directories using cut, and print whether the directory exists or not using an if check.

Ans:



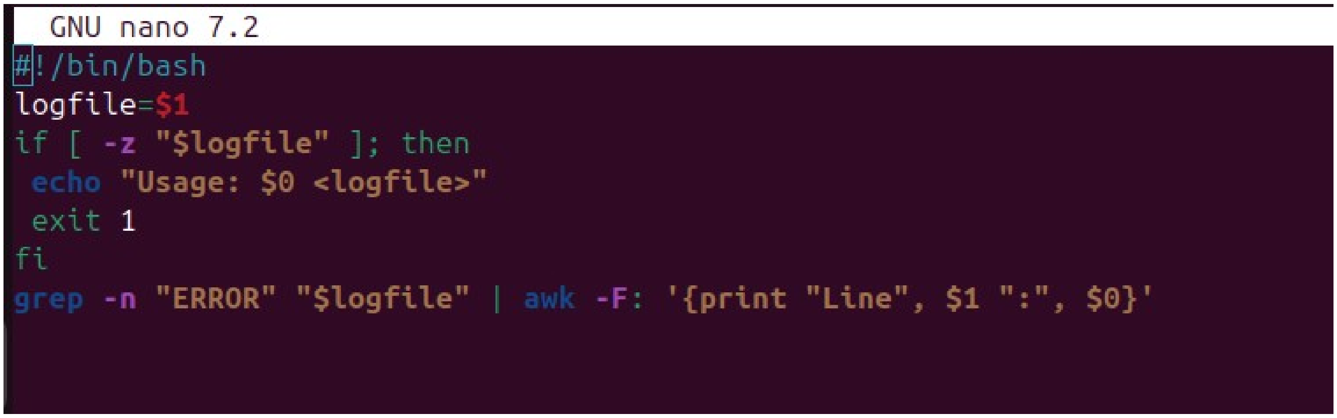
```
#!/bin/bash
cut -d: -f1,6 /etc/passwd | while IFS=: read user home; do
    if [ -d "$home" ]; then
        echo "User $user: Home directory exists"
    else
        echo "User $user: Home directory MISSING"
    fi
done
```



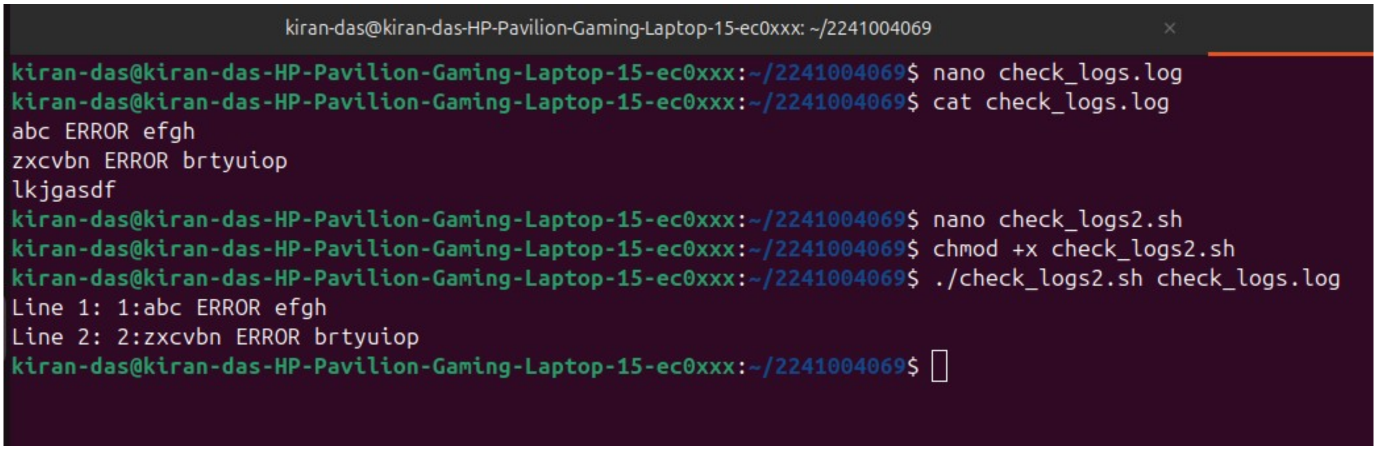
```
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ nano home_dir.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ chmod +x home_dir.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ ./home_dir.sh
User root: Home directory exists
User daemon: Home directory exists
User bin: Home directory exists
User sys: Home directory exists
User sync: Home directory exists
User games: Home directory exists
User man: Home directory exists
User lp: Home directory MISSING
User mail: Home directory exists
User news: Home directory MISSING
User uucp: Home directory MISSING
User proxy: Home directory exists
User www-data: Home directory exists
User backup: Home directory exists
User list: Home directory MISSING
User irc: Home directory MISSING
User gnats: Home directory MISSING
User nobody: Home directory MISSING
User systemd-network: Home directory exists
User systemd-resolve: Home directory exists
User syslog: Home directory MISSING
User messagebus: Home directory MISSING
User _apt: Home directory MISSING
User uuidd: Home directory exists
User dnsmasq: Home directory exists
User avahi-autoipd: Home directory exists
User usbmux: Home directory MISSING
User whoopsie: Home directory MISSING
User kernoops: Home directory exists
User rtkit: Home directory exists
User avahi: Home directory exists
User cups-pk-helper: Home directory MISSING
User saned: Home directory MISSING
User speech-dispatcher: Home directory MISSING
User colord: Home directory exists
User pulse: Home directory MISSING
User hplip: Home directory MISSING
User geoclue: Home directory exists
```

Q5. Using grep and awk, write a script to parse a given log file and list all lines containing the word “ERROR” along with the line number.

Ans:



```
  GNU nano 7.2
#!/bin/bash
logfile=$1
if [ -z "$logfile" ]; then
 echo "Usage: $0 <logfile>"
 exit 1
fi
grep -n "ERROR" "$logfile" | awk -F: '{print "Line", $1 ":", $0}'
```

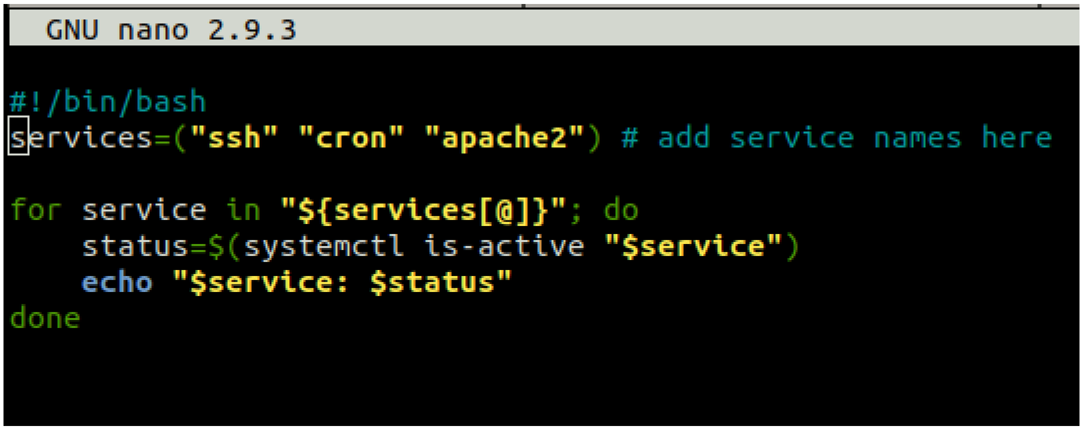


kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx: ~/2241004069

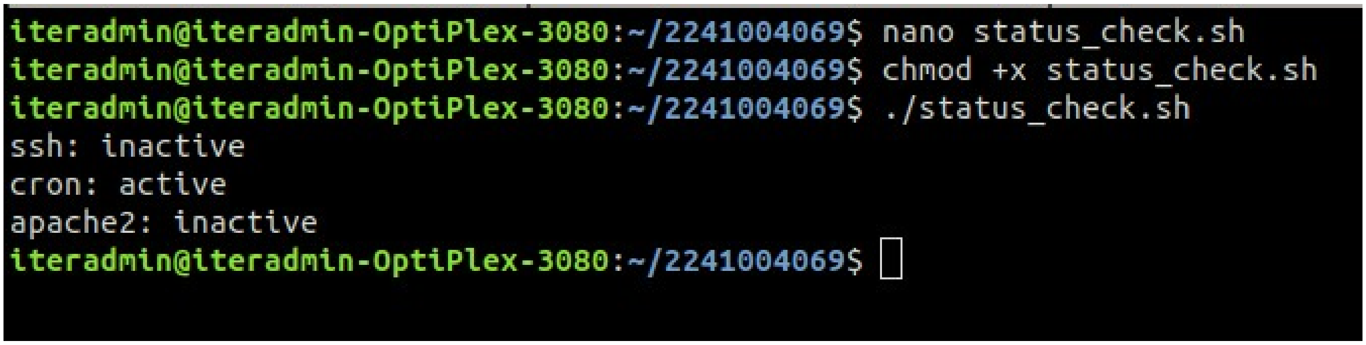
```
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ nano check_logs.log
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ cat check_logs.log
abc ERROR efgh
zxcvbn ERROR brtyuiop
lkjgasdf
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ nano check_logs2.sh
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ chmod +x check_logs2.sh
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ ./check_logs2.sh check_logs.log
Line 1: 1:abc ERROR efgh
Line 2: 2:zxcvbn ERROR brtyuiop
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ 
```

Q6. Create a script that takes a list of service names and checks whether each service is active or inactive using systemctl or service. Use loops and if statements.

Ans:



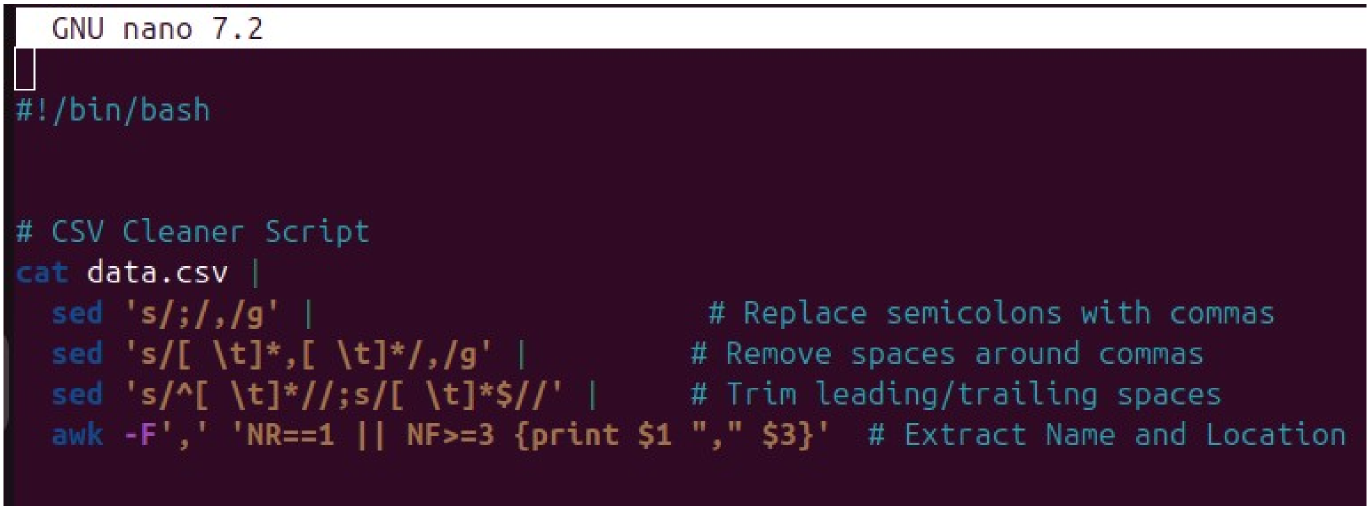
```
  GNU nano 2.9.3

#!/bin/bash
services=("ssh" "cron" "apache2") # add service names here

for service in "${services[@]}"; do
    status=$(systemctl is-active "$service")
    echo "$service: $status"
done
```



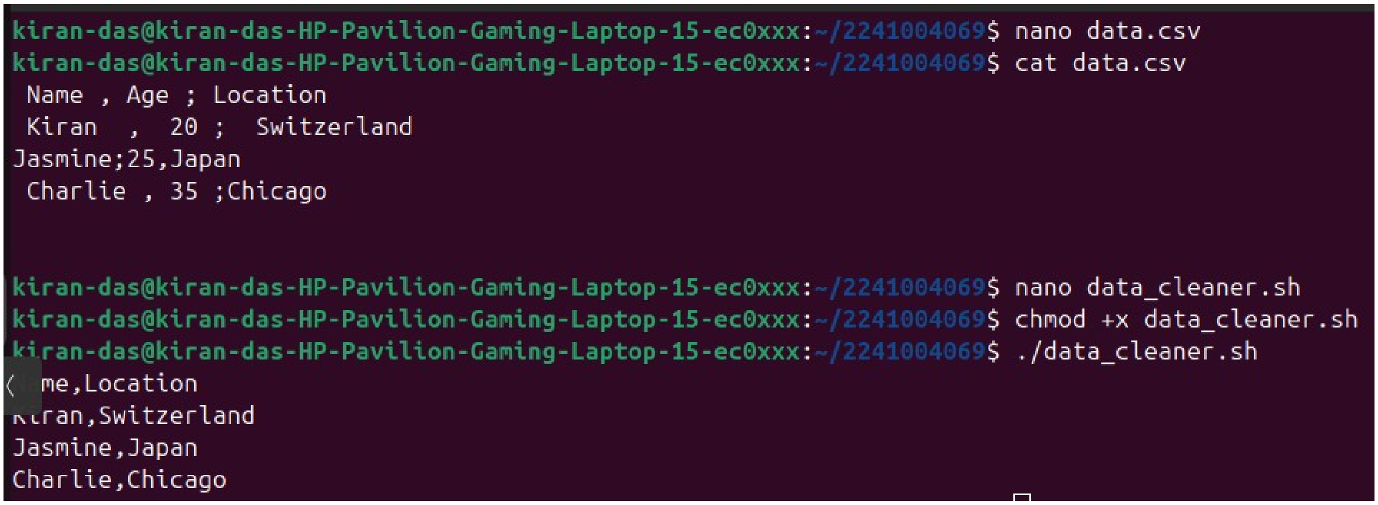
```
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ nano status_check.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ chmod +x status_check.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ ./status_check.sh
ssh: inactive
cron: active
apache2: inactive
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$
```

Q7. Given a CSV file with extra spaces and inconsistent delimiters, use awk, sed, and cut to clean it and print only selected columns neatly.

Ans:



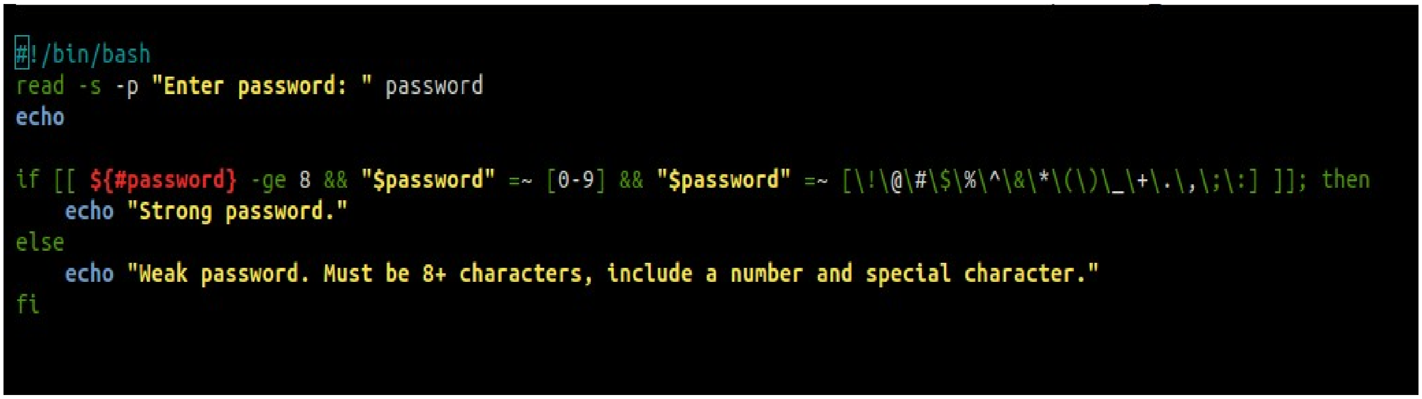
```
  GNU nano 7.2

#!/bin/bash


# CSV Cleaner Script
cat data.csv |
  sed 's/;/,/g' |                         # Replace semicolons with commas
  sed 's/[ \t]*,[ \t]*/,/g' |            # Remove spaces around commas
  sed 's/^[ \t]*//;s/[ \t]*$//' |        # Trim leading/trailing spaces
  awk -F',' 'NR==1 || NF>=3 {print $1 "," $3}'  # Extract Name and Location
```



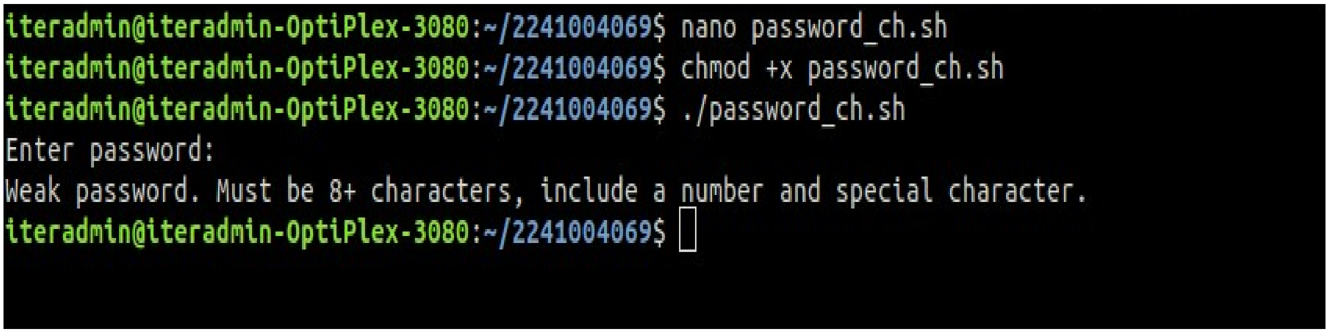
```
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ nano data.csv
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ cat data.csv
 Name , Age ; Location
 Kiran  ,  20 ;  Switzerland
Jasmine;25,Japan
 Charlie , 35 ;Chicago


kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ nano data_cleaner.sh
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ chmod +x data_cleaner.sh
kiran-das@kiran-das-HP-Pavilion-Gaming-Laptop-15-ec0xxx:~/2241004069$ ./data_cleaner.sh
me,Location
Kiran,Switzerland
Jasmine,Japan
Charlie,Chicago
```

Q8. Accept a string as input and use a combination of regex (grep or [[... =~]]) to validate that it’s at least 8 characters, contains one number, and one special character.

Ans:



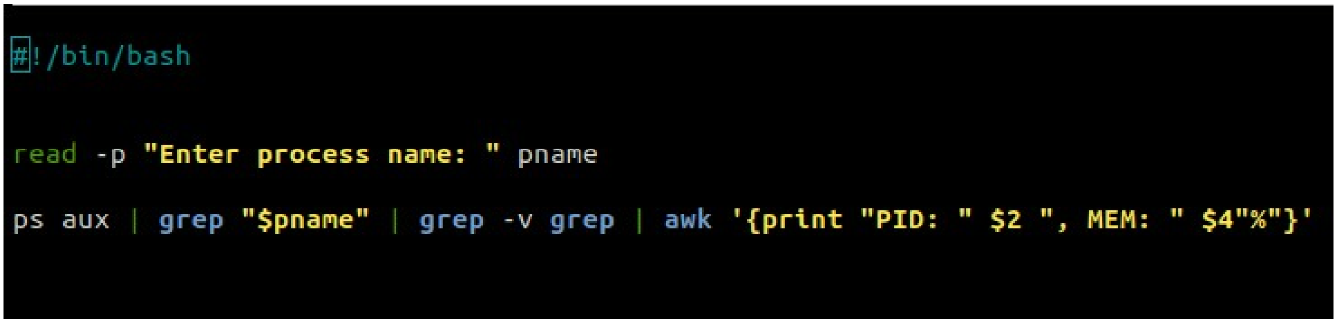
```
#!/bin/bash
read -s -p "Enter password: " password
echo

if [[ ${#password} -ge 8 && "$password" =~ [0-9] && "$password" =~ [\!\@\#\$\%\^\&\*\(\)\_\+\.\,\;\:] ]]; then
    echo "Strong password."
else
    echo "Weak password. Must be 8+ characters, include a number and special character."
fi
```



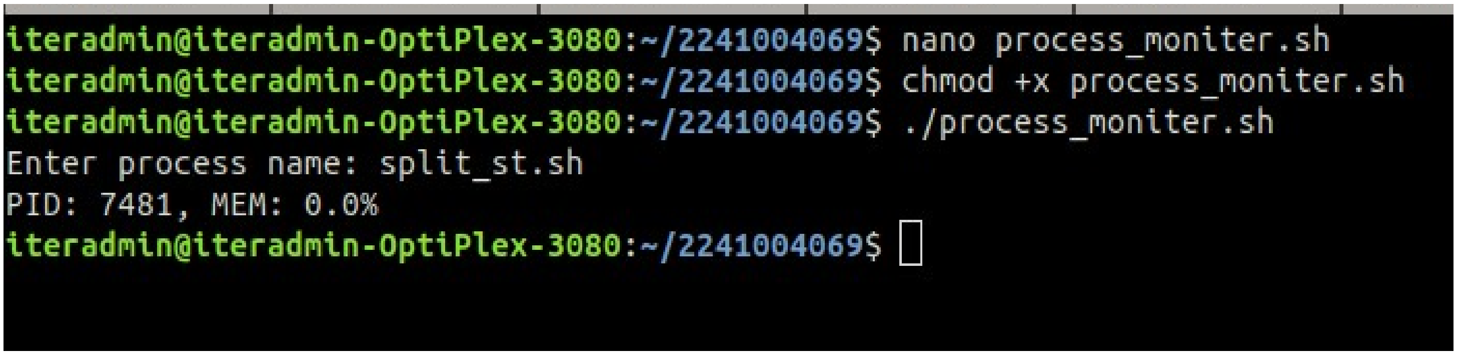
```
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ nano password_ch.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ chmod +x password_ch.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ ./password_ch.sh
Enter password:
Weak password. Must be 8+ characters, include a number and special character.
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ 
```

Q9. Create a script that accepts a process name as input, checks if it is running using ps and grep, and displays its PID and memory usage.

Ans:



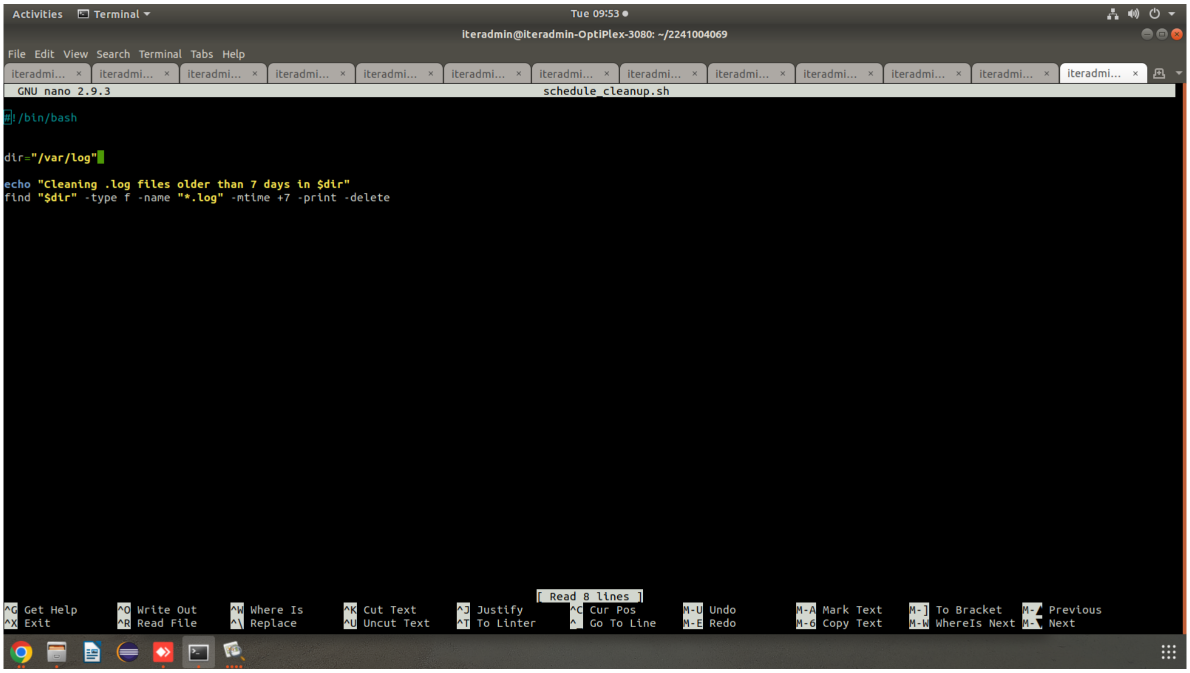
```
#!/bin/bash

read -p "Enter process name: " pname

ps aux | grep "$pname" | grep -v grep | awk '{print "PID: " $2 ", MEM: " $4"%"}'
```



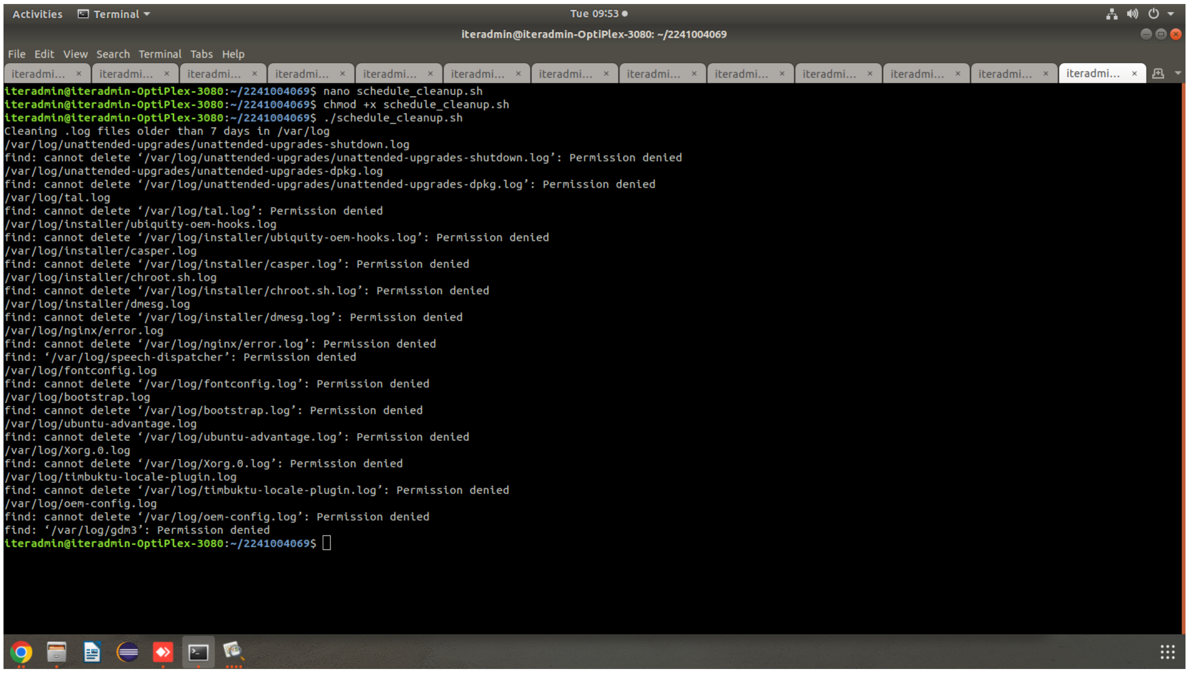
```
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ nano process_moniter.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ chmod +x process_moniter.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ ./process_moniter.sh
Enter process name: split_st.sh
PID: 7481, MEM: 0.0%
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ 
```

Q10. Write a script that deletes .log files older than 7 days from a given directory. Use find, conditionals, and optionally log what files were removed.

Ans:



```
#!/bin/bash


dir="/var/log"

echo "Cleaning .log files older than 7 days in $dir"
find "$dir" -type f -name "*.log" -mtime +7 -print -delete
```


Activities
Terminal
Tue 09:53
iteradmin@iteradmin-OptiPlex-3080: ~/2241004069
File Edit View Search Terminal Tabs Help
iteradmi…

```
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ nano schedule_cleanup.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ chmod +x schedule_cleanup.sh
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ ./schedule_cleanup.sh
Cleaning .log files older than 7 days in /var/log
/var/log/unattended-upgrades/unattended-upgrades-shutdown.log
find: cannot delete ‘/var/log/unattended-upgrades/unattended-upgrades-shutdown.log’: Permission denied
/var/log/unattended-upgrades/unattended-upgrades-dpkg.log
find: cannot delete ‘/var/log/unattended-upgrades/unattended-upgrades-dpkg.log’: Permission denied
/var/log/tal.log
find: cannot delete ‘/var/log/tal.log’: Permission denied
/var/log/installer/ubiquity-oem-hooks.log
find: cannot delete ‘/var/log/installer/ubiquity-oem-hooks.log’: Permission denied
/var/log/installer/casper.log
find: cannot delete ‘/var/log/installer/casper.log’: Permission denied
/var/log/installer/chroot.sh.log
find: cannot delete ‘/var/log/installer/chroot.sh.log’: Permission denied
/var/log/installer/dmesg.log
find: cannot delete ‘/var/log/installer/dmesg.log’: Permission denied
/var/log/nginx/error.log
find: cannot delete ‘/var/log/nginx/error.log’: Permission denied
find: ‘/var/log/speech-dispatcher’: Permission denied
/var/log/fontconfig.log
find: cannot delete ‘/var/log/fontconfig.log’: Permission denied
/var/log/bootstrap.log
find: cannot delete ‘/var/log/bootstrap.log’: Permission denied
/var/log/ubuntu-advantage.log
find: cannot delete ‘/var/log/ubuntu-advantage.log’: Permission denied
/var/log/Xorg.0.log
find: cannot delete ‘/var/log/Xorg.0.log’: Permission denied
/var/log/timbuktu-locale-plugin.log
find: cannot delete ‘/var/log/timbuktu-locale-plugin.log’: Permission denied
/var/log/oem-config.log
find: cannot delete ‘/var/log/oem-config.log’: Permission denied
find: ‘/var/log/gdm3’: Permission denied
iteradmin@iteradmin-OptiPlex-3080:~/2241004069$ 
```