

EXPERIMENT 9

SHELL PROGRAMMING CONTINUED AND SYSTEM PERFORMANCE MONITORING.

ADITYA MISHRA , BATCH-78 , 590029219

Q1: Write a script to find largest file in given directory.

Step-by-Step Process

1. Define the directory

Decide which directory you want to scan.
You can pass it as a script argument (recommended) or hard-code it.

2. Validate the directory

Check whether the provided directory exists.
If it doesn't, exit with an error.

3. Traverse the directory

List all files (ignore subdirectories unless you want recursive search).
You may use:

`os.listdir()` (non-recursive)

`os.walk()` (recursive – scans all subfolders)

4. Compare file sizes

For each file:

Build full path

Use `os.path.getsize(path)` to get file size

Track the largest file encountered

5. Print the result

```
After scanning all files, output:  
Largest file name  
Full pathTTT  
Size in bytes (or human-readable format)
```

Script:

```
#!/bin/bash  
  
# Check if a directory was provided  
if [ -z "$1" ]; then  
    echo "Usage: $0 /path/to/directory"  
    exit 1  
fi  
  
DIR="$1"  
  
# Find the largest file  
largest=$(find "$DIR" -type f -printf "%s %p\n" 2>/dev/null | sort -nr |  
head -1)  
  
if [ -z "$largest" ]; then  
    echo "No files found in the specified directory."  
    exit 1  
fi  
  
# Print result  
size=$(echo "$largest" | awk '{print $1}')  
file=$(echo "$largest" | cut -d' ' -f2-)  
  
echo "-----"  
echo " Largest File in: $DIR"  
echo "-----"  
echo "File: $file"  
echo "Size: $size bytes"  
echo "-----"
```

Output:

```
aditya@aditya-Inspiron-15-3530:~$ mkdir exp9
aditya@aditya-Inspiron-15-3530:~$ cd exp9
aditya@aditya-Inspiron-15-3530:~/exp9$ nano find_largest_file.sh
aditya@aditya-Inspiron-15-3530:~/exp9$ chmod +x find_largest_file.sh
aditya@aditya-Inspiron-15-3530:~/exp9$ ./find_largest_file.sh
Usage: ./find_largest_file.sh /path/to/directory
aditya@aditya-Inspiron-15-3530:~/exp9$ bash find_largest_file.sh
Usage: find_largest_file.sh /path/to/directory
aditya@aditya-Inspiron-15-3530:~/exp9$ ./find_largest_file.sh /home/aditya/linux-lab-assingments
bash: ./find_largest_file.sh /home/aditya/linux-lab-assingments: Not a directory
aditya@aditya-Inspiron-15-3530:~/exp9$ ./find_largest_file.sh /home/aditya/linux-lab-assingments
-----
Largest File in: /home/aditya/linux-lab-assingments
-----
File: /home/aditya/linux-lab-assingments/.git/objects/pack/pack-080fb78859a05086878ccb9d746bcea8a4c4e995.pack
Size: 4149459 bytes
-----
aditya@aditya-Inspiron-15-3530:~/exp9$
```

Q2:Create a script that counts how many .sh files exist in /home/user.

STEP-BY-STEP PROCESS

STEP 1: Open the Terminal

Ctrl + Alt + T

or open it from your application menu.

STEP 2: Create a new script file

Type:

```
nano count_sh_files.sh
```

This opens the nano text editor.

STEP 3: Paste the script inside nano

Copy and paste the following script:

```
#!/bin/bash
```

```
DIRECTORY="/home/user"

# Count .sh files
count=$(find "$DIRECTORY" -type f -name "*.sh" | wc -l)

echo "Number of .sh files in $DIRECTORY: $count"
```

STEP 4: Save the file in nano

Inside nano:

Press CTRL + O → writes the file

Press Enter → confirms the filename

Press CTRL + X → exits nano

Your script is now created.

STEP 5: Make the script executable

Run :

```
chmod +x count_sh_files.sh
```

This allows Linux to execute the script like a program.

STEP 6: Run the script

Now run:

```
./count_sh_files.sh
```

You will see output like:

```
Number of .sh files in /home/user: 1
```

Outout:

```
aaditya@pop-os:~$ nano exp9.sh
aaditya@pop-os:~$ ./exp9.sh
bash: ./exp9.sh: Permission denied
aaditya@pop-os:~$ chmod +x 777 exp9.sh
chmod: cannot access '777': No such file or directory
aaditya@pop-os:~$ chmod 777 exp9.sh
aaditya@pop-os:~$ ./exp9.sh
Error: /home/user does not exist.
aaditya@pop-os:~$ nano exp9.sh
aaditya@pop-os:~$ ./exp9.sh
Number of .sh files in /home/aaditya: 1
aaditya@pop-os:~$ 
```

Q3: Write a script to monitor CPU usage every 10 seconds and log to file

1. Save the script

Create a file:

```
sudo tee /usr/local/bin/monitor_cpu.sh > /dev/null <<'EOF'
(paste the script contents here)
EOF
```

Or save it to your home folder as `monitor_cpu.sh`.

2. Make it executable

```
chmod +x /usr/local/bin/monitor_cpu.sh
```

3. Run it

Run in foreground (useful to see output while debugging):

```
/usr/local/bin/monitor_cpu.sh
```

Run in background:

```
/usr/local/bin/monitor_cpu.sh &
```

Run with an alternate log file (no sudo required if you write to a writable path):

```
LOGFILE="$HOME/cpu.log" /usr/local/bin/monitor_cpu.sh &
```

4: View the log

Tail live:

```
tail -f /var/log/cpu_monitor.log
```

Show last 50 lines:

```
tail -n 50 /var/log/cpu_monitor.log
```

If you used a custom path:

```
tail -f ~/cpu.log
```

5.Run as a service (recommended for long-running monitoring)

Create a systemd unit file /etc/systemd/system/monitor_cpu.service:

```
[Unit]
Description=CPU monitor (every 10s)

[Service]
ExecStart=/usr/local/bin/monitor_cpu.sh
Restart=always
User=root
# Or set User=someuser and ensure LOGFILE is writable by that user

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

Reload and enable:

```
sudo systemctl daemon-reload
sudo systemctl enable --now monitor_cpu.service
```

Check status and logs:

```
sudo systemctl status monitor_cpu.service
sudo journalctl -u monitor_cpu.service -f
```

Stop the script

If background process: find PID and kill:

```
pgrep -f monitor_cpu.sh kill
```

If systemd service:

```
sudo systemctl stop monitor_cpu.service
```

Script:

```
#!/bin/bash

LOGFILE="/var/log/cpu_usage.log"

touch "$LOGFILE"

echo "Starting CPU usage monitoring..." echo "Logging CPU usage to $LOGFILE" echo "Press Ctrl+C to stop."

while true do timestamp=$(date +"%Y-%m-%d %H:%M:%S")

cpu_usage=$(top -bn1 | grep "Cpu(s)" | awk '{print 100 - $8 "%"}')

echo "$timestamp - CPU Usage: $cpu_usage" >> "$LOGFILE" sleep 10

done
```

```
### Output:  
![alt text](<Screenshot from 2025-11-21 00-55-05.png>)
```

```
### Q4: Create a script that adds a new user and sets default permissions for their home directory.
```

```
#### Step 1: Decide the username
```

```
Pick the username you want to create, e.g.:
```

```
```newuser```
```

```
Step 2: Open terminal with sudo/root access
```

```
Run:
```

```
```sudo -i```
```

```
(or prefix commands with sudo)
```

```
#### Step 3: Create the script file
```

Create a file called add_user.sh:

```
```sudo nano add_user.sh```
```

#### Step 4: Paste the script below

Paste this complete script:

```
```#!/bin/bash
```

Script to add a user and set permissions on their home directory

```
if [ -z "$1" ]; then
    echo "Usage: $0 <username>"
    exit 1
fi
```

```
USERNAME="$1"
```

```
HOMEDIR="/home/$USERNAME"
```

```
PERMISSIONS="700" # Change to 755 if you want more open access
```

1. Create the user (home directory auto-created)

```
useradd -m "$USERNAME"
```

2. Set password (interactive)

```
echo "Set password for user $USERNAME:"
```

```
passwd "$USERNAME"
```

3. Set default permissions for the home directory

```
chmod "$PERMISSIONS" "$HOMEDIR"
```

4. Confirm

```
echo "User $USERNAME created."
```

```
echo "Home directory permissions set to $PERMISSIONS."
```

```
echo "Location: $HOMEDIR"
```

Step 5: Save and exit

Press:

CTRL + O, Enter, CTRL + X

Step 6: Make the script executable

```
sudo chmod +x add_user.sh
```

Step 7: Run the script

Example:

```
sudo ./add_user.sh newuser
```

Step 8: Set password when prompted

You'll be prompted to enter a password.

Step 9: Verify the new user

```
id newuser
```

Step 10: Verify home directory permissions

```
ls -ld /home/newuser
```

You should see something like:

```
drwx----- 3 newuser newuser 4096 Nov 21 22:30 /home/newuser
```

Script:

```
#!/bin/bash

if [ "$(id -u)" -ne 0 ]; then
    echo "Error: This script must be run as root."
    exit 1
fi

if [ -z "$1" ]; then
    echo "Usage: $0 <username>"
    exit 1
fi

USERNAME="$1"

if id "$USERNAME" &>/dev/null; then
    echo "Error: User '$USERNAME' already exists."
    exit 1
fi

useradd -m "$USERNAME"

if [ $? -ne 0 ]; then
    echo "Failed to create user '$USERNAME' ."
    exit 1
fi

chmod 700 /home/"$USERNAME"

echo "Set password for $USERNAME:"
passwd "$USERNAME"

echo "User '$USERNAME' created successfully."
```

```
echo "Home directory: /home/$USERNAME"
echo "Permissions set to 700 (rwx-----)
```

Output:

The screenshot shows a terminal window titled 'root@pop-os:~'. The session starts with a command to create a user 'starky', which fails because the command is not found. The user then becomes root and creates a shell script 'add_user.sh' using nano, gives it execute permissions with chmod, runs it to create the user 'starky', sets a password for the user, and updates the password with passwd. The user 'starky' is created successfully, and the terminal shows the home directory permissions are set to 700. Finally, the user lists the contents of the '/home/starky' directory, which contains a single file named 'starky'.

```
aaditya@pop-os:~$ starky
starky: command not found
aaditya@pop-os:~$ sudo -i
[sudo] password for aaditya:
root@pop-os:~# touch add_user.sh
root@pop-os:~# nano add_user.sh
root@pop-os:~# sudo chmod +x add_user.sh
root@pop-os:~# sudo ./add_user.sh starky
Set password for user starky:
New password:
Retype new password:
passwd: password updated successfully
User starky created.
Home directory permissions set to 700.
Location: /home/starky
root@pop-os:~# ls -ld /home/starky
drwx----- 2 starky starky 4096 Nov 21 01:17 /home/starky
root@pop-os:~#
```