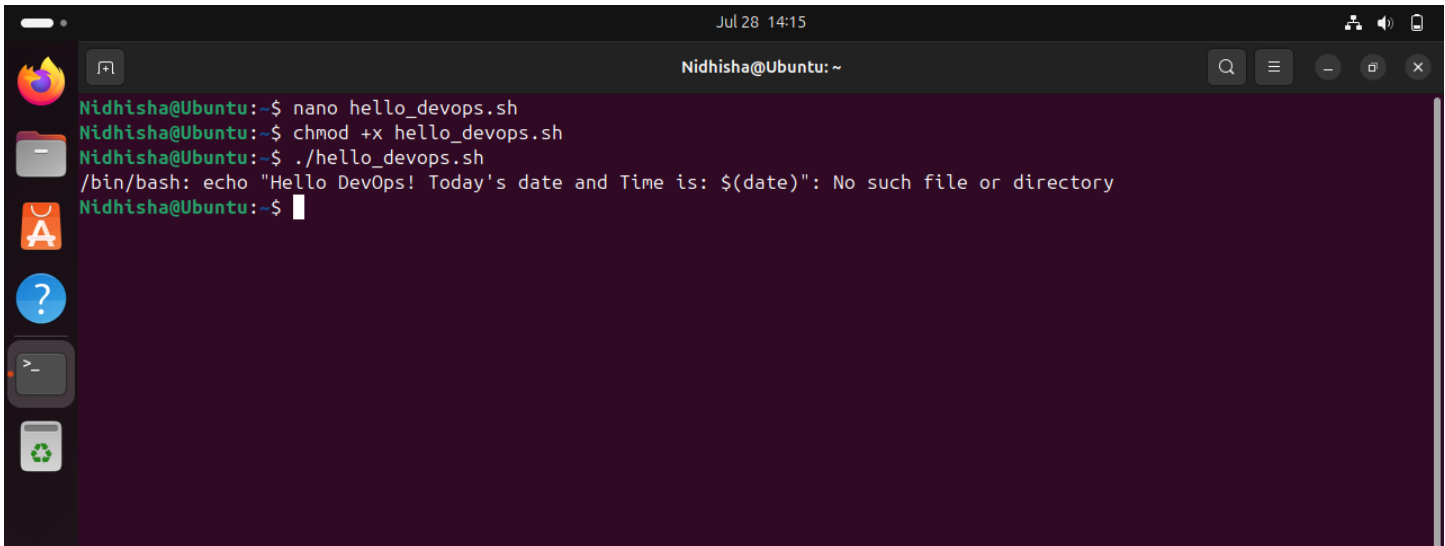


#DAY 2

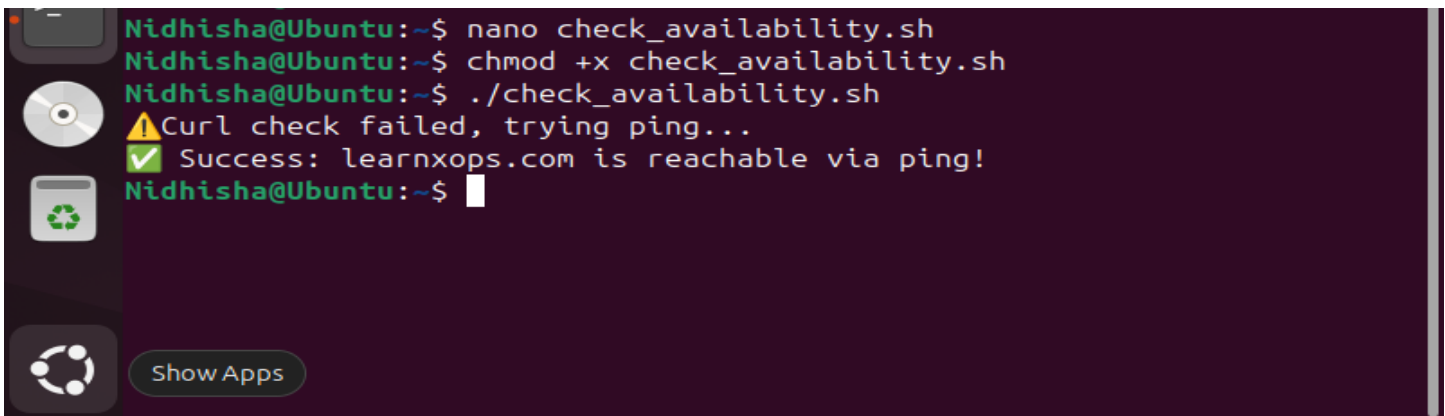
Linux Shell Scripting & Automation

Challenge 1: Write a simple Bash script that prints “Hello DevOps” along with the current date and time.



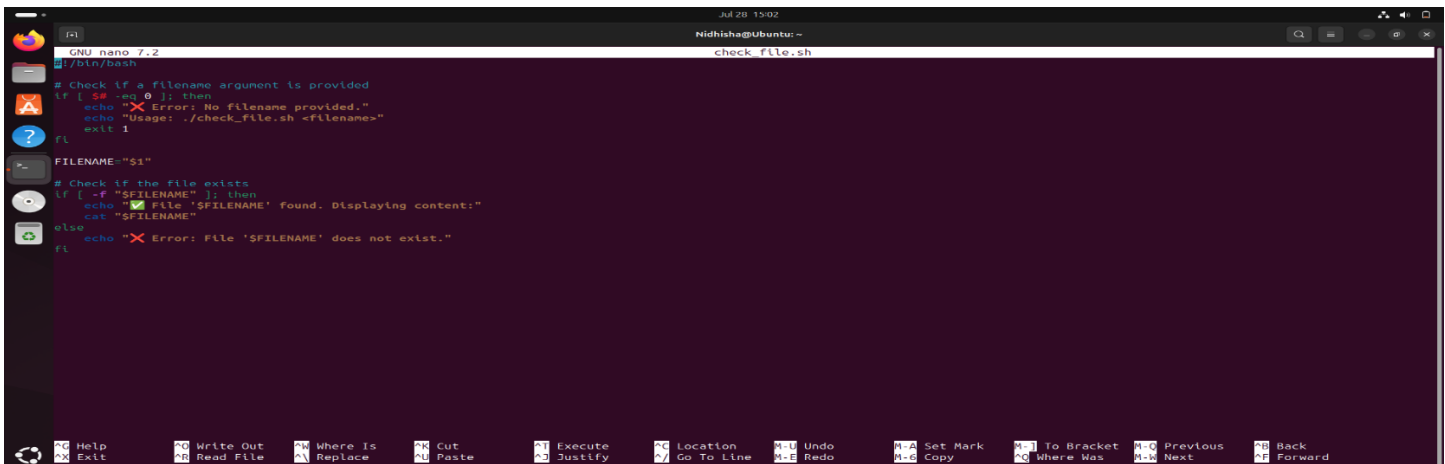
```
Jul 28 14:15
Nidhisha@Ubuntu: ~
Nidhisha@Ubuntu:~$ nano hello_devops.sh
Nidhisha@Ubuntu:~$ chmod +x hello_devops.sh
Nidhisha@Ubuntu:~$ ./hello_devops.sh
/bin/bash: echo "Hello DevOps! Today's date and Time is: $(date)": No such file or directory
Nidhisha@Ubuntu:~$
```

Challenge 2: Create a script that checks if a website (e.g., <https://www.learnxops.com>) is reachable using curl or ping. Print a success or failure message.



```
Nidhisha@Ubuntu:~$ nano check_availability.sh
Nidhisha@Ubuntu:~$ chmod +x check_availability.sh
Nidhisha@Ubuntu:~$ ./check_availability.sh
⚠️ Curl check failed, trying ping...
✅ Success: learnxops.com is reachable via ping!
Nidhisha@Ubuntu:~$
```

Challenge 3: Write a script that takes a filename as an argument, checks if it exists, and prints the content of the file accordingly.

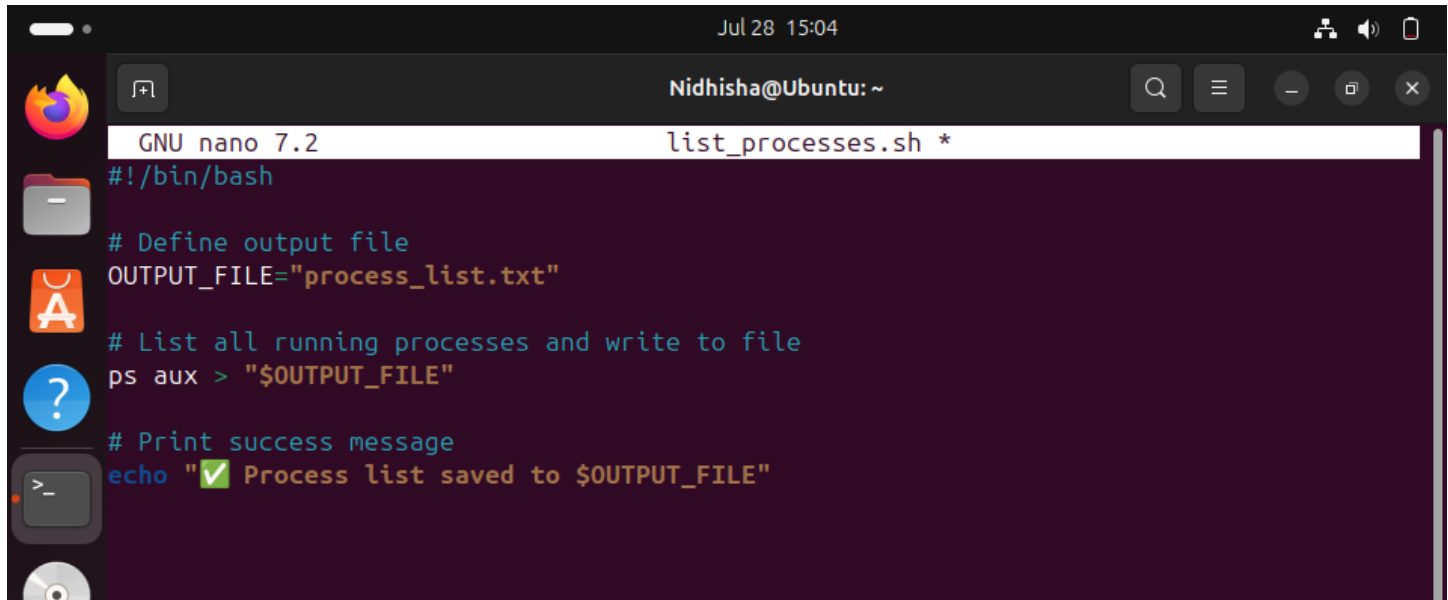


```
Jul 28 15:02
Nidhisha@Ubuntu: ~
GNU nano 7.2
# Check if a filename argument is provided
if [ $# -eq 0 ]; then
    echo "❌ Error: No filename provided."
    echo "Usage: ./check_file.sh <filenames>"
    exit 1
fi

FILENAME="$1"

# Check if the file exists
if [ -f "$FILENAME" ]; then
    echo "✅ File '$FILENAME' found. Displaying content:"
    cat "$FILENAME"
else
    echo "❌ Error: File '$FILENAME' does not exist."
fi
```

Challenge 4: Create a script that lists all running processes and writes the output to a file named `process_list.txt`.



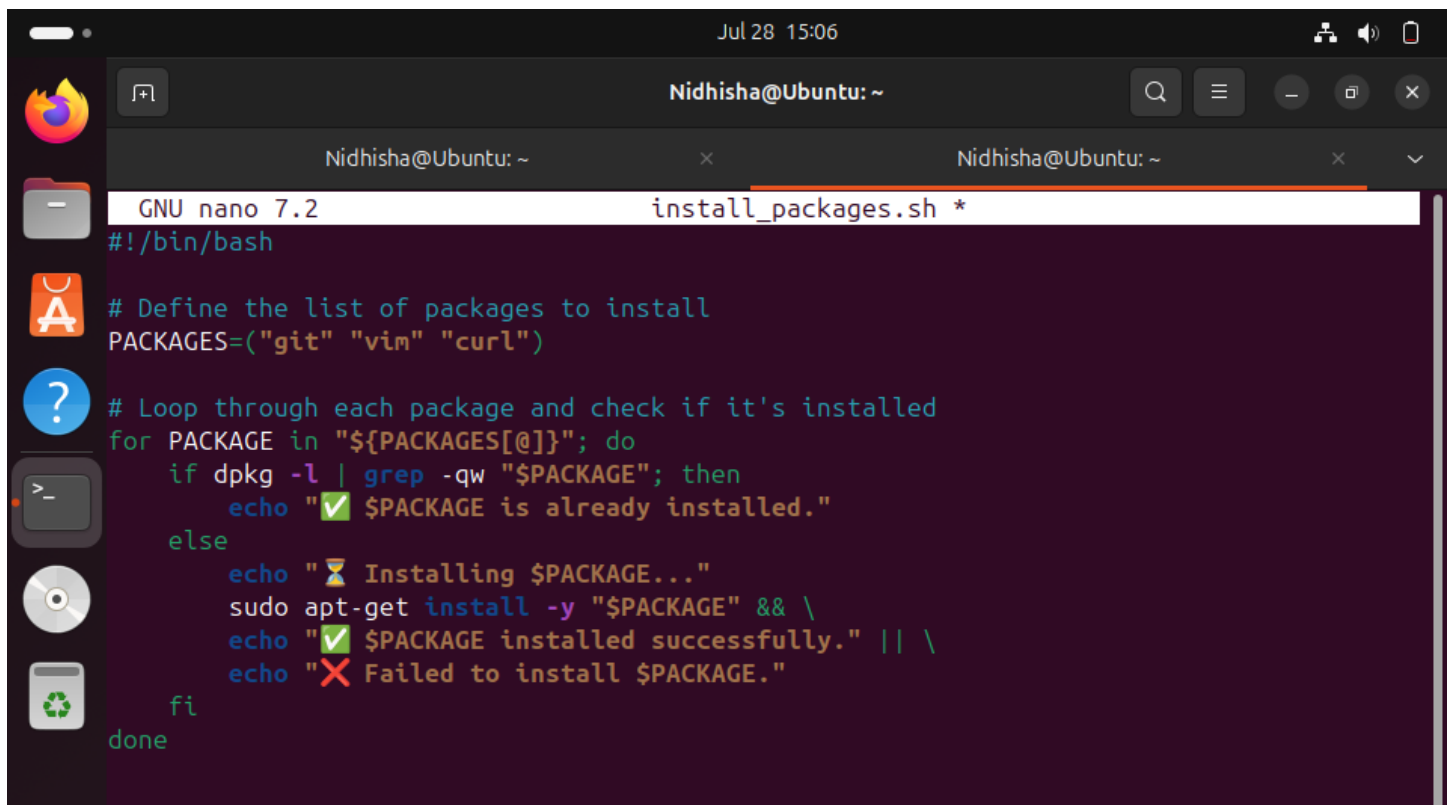
```
GNU nano 7.2 list_processes.sh *
#!/bin/bash

# Define output file
OUTPUT_FILE="process_list.txt"

# List all running processes and write to file
ps aux > "$OUTPUT_FILE"

# Print success message
echo "✅ Process list saved to $OUTPUT_FILE"
```

Challenge 5: Write a script that installs multiple packages at once (e.g., git, vim, curl). The script should check if each package is already installed before attempting installation.



```
GNU nano 7.2 install_packages.sh *
#!/bin/bash

# Define the list of packages to install
PACKAGES=("git" "vim" "curl")

# Loop through each package and check if it's installed
for PACKAGE in "${PACKAGES[@]}; do
    if dpkg -l | grep -qw "$PACKAGE"; then
        echo "✅ $PACKAGE is already installed."
    else
        echo "⌚ Installing $PACKAGE..."
        sudo apt-get install -y "$PACKAGE" && \
        echo "✅ $PACKAGE installed successfully." || \
        echo "❌ Failed to install $PACKAGE."
    fi
done
```

Challenge 6: Create a script that monitors CPU and memory usage every 5 seconds and logs the results to a file.

```
Jul 28 15:08
Nidhisha@Ubuntu: ~
GNU nano 7.2 monitor_resources.sh *
#!/bin/bash

# Define the log file
LOG_FILE="resource_usage.log"

echo "Monitoring CPU and Memory usage... Logs will be saved in $LOG_FILE"
echo "Timestamp | CPU (%) | Memory (%)" > "$LOG_FILE"

# Infinite loop to log system usage every 5 seconds
while true; do
    TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

    # Get CPU usage
    CPU_USAGE=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}')

    # Get Memory usage
    MEM_USAGE=$(free | awk '/Mem/ {printf "%.2f", $3/$2 * 100}')

    # Write data to the log file
    echo "$TIMESTAMP | $CPU_USAGE | $MEM_USAGE" >> "$LOG_FILE"

    # Wait for 5 seconds
    sleep 5
done
```

Challenge 7: Write a script that automatically deletes log files older than 7 days from /var/log.

```
Jul 28 15:11
Nidhisha@Ubuntu: ~
Nidhisha@Ubuntu: ~
Nidhisha@Ubuntu:~$ nano clean_old_logs.sh
Nidhisha@Ubuntu:~$ chmod +x clean_old_logs.sh
chmod: cannot access 'clean_old_logs.sh': No such file or directory
Nidhisha@Ubuntu:~$ chmod +x clean_old_logs.sh
Nidhisha@Ubuntu:~$ sudo ./clean_old_logs.sh
[sudo] password for Nidhisha:
✔ Deleted log files older than 7 days from /var/log.
Nidhisha@Ubuntu:~$
```

Challenge 8: Automate user account creation – Write a script that takes the username as an argument, checks, if the user exists, gives the message “user already exists” else creates a new user, adds it to a “devops” group, and sets up a default home directory.

```
Nidhisha@Ubuntu:~$ nano create_user.sh
Nidhisha@Ubuntu:~$ chmod +x create_user.sh
Nidhisha@Ubuntu:~$ sudo ./create_user.sh devops_user
✔ User 'devops_user' already exists.
Nidhisha@Ubuntu:~$ sudo ./create_user.sh cloud_user
⏱ Creating group 'devops'...
⏱ Creating user 'cloud_user'...
passwd: password changed.
✔ User 'cloud_user' created successfully and added to group 'devops'.
ℹ Default password: ChangeMe123 (User must change it on first login)
Nidhisha@Ubuntu:~$
```

Challenge 9: Use awk or sed in a script to process a log file and extract only error messages.

```
Nidhisha@Ubuntu:~$ nano extract_errors.sh
Nidhisha@Ubuntu:~$ chmod +x extract_errors.sh
Nidhisha@Ubuntu:~$ ./extract_errors.sh
awk: cannot open "/var/log/syslog" (Permission denied)
✓ Extracted error messages saved to 'error_messages.log'.
```

Challenge 10: Set up a cron job that runs a script to back up (zip/tar) a directory daily.

```
Nidhisha@Ubuntu:~$ nano backup.sh
Nidhisha@Ubuntu:~$ chmod
chmod: missing operand
Try 'chmod --help' for more information.
Nidhisha@Ubuntu:~$ chmod +x backup.sh
Nidhisha@Ubuntu:~$ 0 2 * * * /path/to/backup.sh >> /var/log/backup.log 2>&1
bash: /var/log/backup.log: Permission denied
Nidhisha@Ubuntu:~$
```

