

CRON

- to edit file use cron -e
- sudo crontab -u root -e(to edit roots crontab in particular)
- sudo crontab -u root -l (shows which crontab was installed)

```
#!/bin/bash
PATH="/home/t0315000"
/usr/bin/touch $PATH/testfile
echo "Cron job sample file" > $PATH/testfile
```

Task 1: Write a script to monitor a directory for new files and log changes.

```
t0315000@thales:~$ nano monitor_directory.sh
t0315000@thales:~$ chmod +x monitor_directory.sh
t0315000@thales:~$ ./monitor_directory.sh
Monitoring directory: /home/t0315000
Logging changes to: /home/t0315000/log_file.log
Setting up watches. Beware: since -r was given, this may take a while!
Watches established.

^Z
[2]+  Stopped                  ./monitor_directory.sh
t0315000@thales:~$ cat /home/t0315000/log_file.log
cat: /home/t0315000/log_file.log: No such file or directory
t0315000@thales:~$ ./monitor_directory.sh
Monitoring directory: /home/t0315000
Logging changes to: /home/t0315000/log_file.log
Setting up watches. Beware: since -r was given, this may take a while!
Watches established.

^C
t0315000@thales:~$ ./monitor_directory.sh
Monitoring directory: /home/t0315000
Logging changes to: /home/t0315000/log_file.log
Setting up watches. Beware: since -r was given, this may take a while!
Watches established.

^C
t0315000@thales:~$ touch why.txt
t0315000@thales:~$ jobs
[1]-  Stopped                  cat
[2]+  Stopped                  ./monitor_directory.sh
t0315000@thales:~$ nano monitor_directory.sh
```

AM

Approach : using inotifywait

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

```
# Directory to monitor
MONITOR_DIR="/home/t0315000"
# Log file to store changes
LOG_FILE="/home/t0315000/log_file.log"

# Check if the directory exists
if [ ! -d "$MONITOR_DIR" ]; then
    echo "Directory $MONITOR_DIR does not exist."
```

```

        exit 1
    fi

    # Start monitoring
    echo "Monitoring directory: $MONITOR_DIR"
    echo "Logging changes to: $LOG_FILE"

    # Use inotifywait to monitor the directory for new files
    inotifywait -m -r -e create "$MONITOR_DIR" | while read path action file; do
        echo "$(date +%Y-%m-%d %H:%M:%S) - New file created: $file in $path" >> "$LOG_FILE"
    done

./monitor_directory.sh
cat /path/to/log_file.log
nohup ./monitor_directory.sh &

```

OUTPUT

```

t0315000@thales:~$ cat /home/t0315000/log_file.log
2025-02-11 06:12:41 - New file created: log_file.log in /home/t0315000/
2025-02-11 06:12:41 - New file created: log_file.log in /home/t0315000/
2025-02-11 06:12:41 - New file created: log_file.log in /home/t0315000/
2025-02-11 06:12:41 - New file created: log_file.log in /home/t0315000/
t0315000@thales:~$ cat /home/t0315000/log_file.log
2025-02-11 06:12:41 - New file created: log_file.log in /home/t0315000/
2025-02-11 06:12:41 - New file created: log_file.log in /home/t0315000/
2025-02-11 06:12:41 - New file created: log_file.log in /home/t0315000/
2025-02-11 06:12:41 - New file created: log_file.log in /home/t0315000/
2025-02-11 06:14:05 - New file created: why.txt in /home/t0315000/
2025-02-11 06:14:05 - New file created: why.txt in /home/t0315000/
2025-02-11 06:14:05 - New file created: why.txt in /home/t0315000/
t0315000@thales:~$ █

```

Task 2: Write a script to continuously ping a server and log the response.

```

#!/bin/bash

# Server to ping
SERVER="10.113.53.85" # Replace with the server you want to ping
# Log file to store the ping responses
LOG_FILE="/home/t0315000/ping_log.log" # Specify the path to your log file

# Log the start time
echo "Pinging $SERVER. Logging to $LOG_FILE."
echo "Start time: $(date)" >> "$LOG_FILE"

# Continuous ping loop
while true; do
    # Get the current date/time for logging
    TIMESTAMP=$(date +%Y-%m-%d %H:%M:%S)
    # Ping the server
    RESPONSE=$(ping -c 1 "$SERVER")
    # Check if the ping command was successful
    if [ $? -eq 0 ]; then
        # Log successful response
        echo "$TIMESTAMP - SUCCESS: $RESPONSE" >> "$LOG_FILE"
    fi
done

```

```

else
    # Log failed response
    echo "$TIMESTAMP - FAIL: Unable to reach $SERVER" >> "$LOG_FILE"
fi
# Wait for a specified duration before the next ping (e.g., 1 second)
sleep 1
done

```

```

t0315000@thales:~$ nano ping_server.sh
t0315000@thales:~$ nano ping_server.sh
t0315000@thales:~$ touch ping_log.log
t0315000@thales:~$ chmod +x ping_server.sh
t0315000@thales:~$ ./ping_server.sh
Pinging 10.113.53.85. Logging to /home/t0315000/ping_log.log.
^C
t0315000@thales:~$ ./ping_server.sh
Pinging 10.113.53.85. Logging to /home/t0315000/ping_log.log.
^C
t0315000@thales:~$ nano ping_server.sh
t0315000@thales:~$ less /home/t0315000/ping_log.log
t0315000@thales:~$ cat /home/t0315000/ping_log.log
Start time: Tue Feb 11 06:46:01 IST 2025
2025-02-11 06:46:01 - SUCCESS: PING 10.113.53.85 (10.113.53.85) 56(84) bytes of data.
64 bytes from 10.113.53.85: icmp_seq=1 ttl=64 time=0.035 ms

--- 10.113.53.85 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.035/0.035/0.035/0.000 ms
2025-02-11 06:46:02 - SUCCESS: PING 10.113.53.85 (10.113.53.85) 56(84) bytes of data.
64 bytes from 10.113.53.85: icmp_seq=1 ttl=64 time=0.039 ms

--- 10.113.53.85 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.039/0.039/0.039/0.000 ms
2025-02-11 06:46:03 - SUCCESS: PING 10.113.53.85 (10.113.53.85) 56(84) bytes of data.
64 bytes from 10.113.53.85: icmp_seq=1 ttl=64 time=0.039 ms

--- 10.113.53.85 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.039/0.039/0.039/0.000 ms
2025-02-11 06:46:04 - SUCCESS: PING 10.113.53.85 (10.113.53.85) 56(84) bytes of data.
64 bytes from 10.113.53.85: icmp_seq=1 ttl=64 time=0.040 ms

--- 10.113.53.85 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.040/0.040/0.040/0.000 ms
2025-02-11 06:46:05 - SUCCESS: PING 10.113.53.85 (10.113.53.85) 56(84) bytes of data.
64 bytes from 10.113.53.85: icmp_seq=1 ttl=64 time=0.039 ms

```