

# Experiment 7: Shell Programming, Process and Scheduling

**Name: Biswabandya Mohanty Roll No.: 590029274 Date: 2025-11-1**

## Aim:

- To write shell scripts that demonstrate process management.
- To understand how to schedule processes using `cron` and `at`.
- To monitor running processes and practice job control commands.

## Requirements

- A Linux machine with bash shell.
- Access to process management commands ( `ps` , `top` , `kill` , `jobs` , `fg` , `bg` ).
- Access to scheduling utilities ( `cron` , `at` ).

## Theory

Every program running in Linux is a process identified by a unique process ID (PID). Shell programming allows automation of tasks including spawning and controlling processes. Process management commands like `ps` , `top` , `kill` , `jobs` , `bg` , and `fg` let users monitor and control execution. Scheduling utilities such as `cron` (repeated tasks) and `at` (one-time tasks) allow tasks to run automatically at defined times. Combining scripting with scheduling is a core system administration skill.

## Procedure & Observations

### Exercise 1: Writing a basic shell script

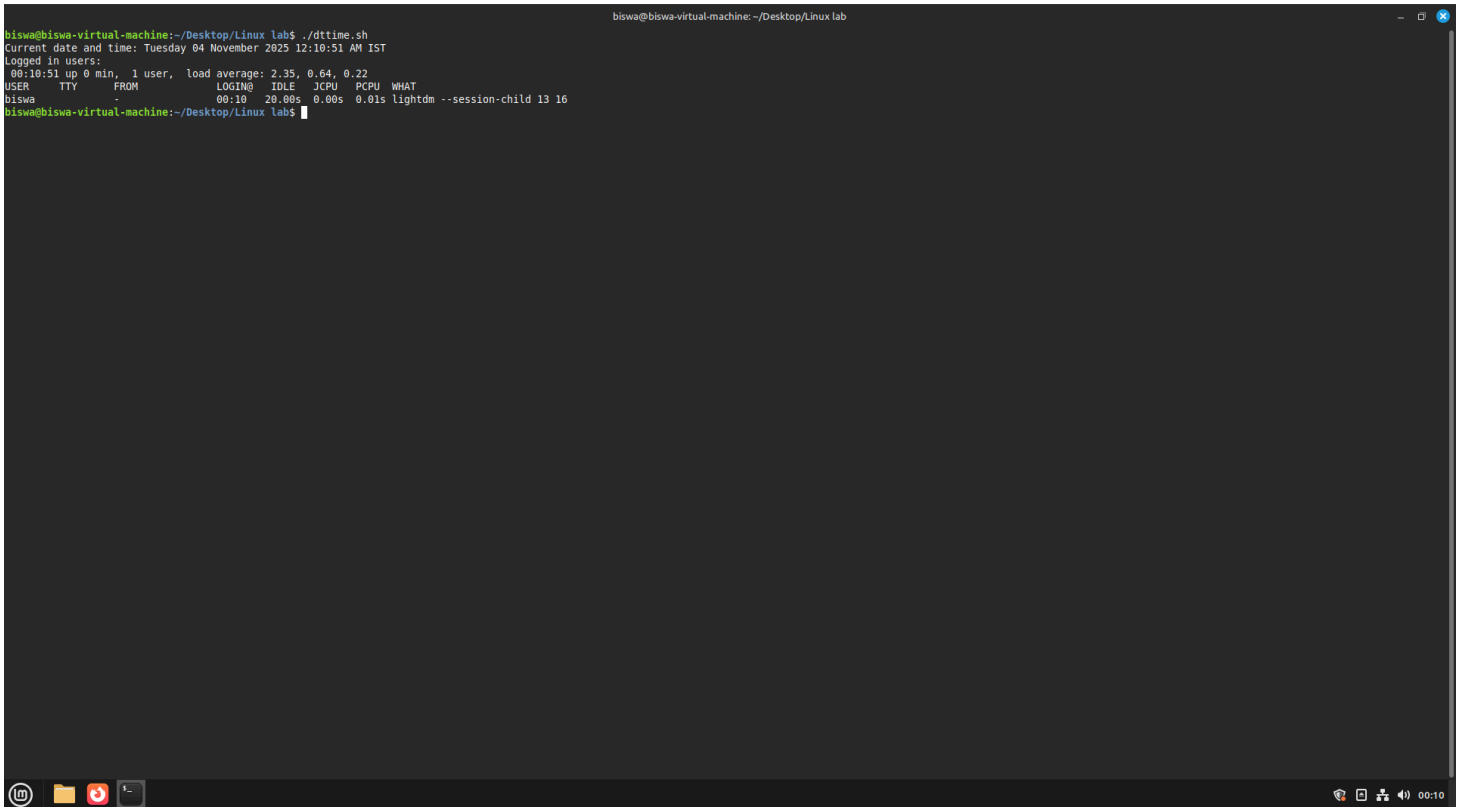
#### Task Statement:

Create a shell script that prints the current date, time, and the list of logged-in users.

## Command(s):

```
#!/bin/bash
echo "Current date and time: $(date)"
echo "Logged in users:"
w
```

## Output:

A screenshot of a terminal window titled "biswa@biswa-virtual-machine: ~/Desktop/Linux lab". The terminal shows the execution of a script named "dttime.sh". The output includes the current date and time, followed by the output of the "w" command, which displays system statistics and a list of logged-in users. The system statistics show the time as 00:10:51, 0 users, and a load average of 2.35, 0.64, 0.22. The "w" command output shows a single user, "biswa", logged in from "tty" at "00:10" for "20.00s" on "lightdm".

```
biswa@biswa-virtual-machine:~/Desktop/Linux lab$ ./dttime.sh
Current date and time: Tuesday 04 November 2025 12:10:51 AM IST
Logged in users:
00:10:51 up 0 min,  1 user,  load average: 2.35, 0.64, 0.22
USER   TTY      FROM             LOGIN@   IDLE   JCPU   PCPU   WHAT
biswa  -        -                00:10   20.00s  0.00s  0.01s  lightdm --session-child 13 16
biswa@biswa-virtual-machine:~/Desktop/Linux lab$
```

## Exercise 2: Background and foreground processes

### Task Statement:

Run a process in background and bring it to the foreground.

## Command(s):

```
sleep 60 &  
jobs  
fg %1
```

## Output:

```
biswa@biswa-virtual-machine: ~/Desktop/Linux lab$ ls  
"V"      Biswa      dttime.sh  exp2.sh    exp5.sh    exp84.sh  file2.sh  file.txt  greeting2.sh  numeric    reverse.sh  sorting.sh  symbols.sh  while2.sh  
arithmetic.sh  bricctn.sh  escape.sh  exp3.sh    exp82.sh  exp85.sh  file3.sh  for.sh    greeting.sh   palindrome.sh  slayslay    sum.sh      vote.sh     x.sh  
armstrong.sh  countlines.sh  expl.sh    exp4.sh    exp83.sh  fibonacci1.sh  file.sh   gcdlcm.sh  interation.sh  prime.sh    sleep.sh    symbols2.sh  while1.sh  
biswa@biswa-virtual-machine:~/Desktop/Linux lab$ vim background.sh  
biswa@biswa-virtual-machine:~/Desktop/Linux lab$ chmod +x background.sh  
biswa@biswa-virtual-machine:~/Desktop/Linux lab$ ./background.sh  
[1]+  Running                  sleep 60 &  
./background.sh: line 3: fg: no job control  
biswa@biswa-virtual-machine:~/Desktop/Linux lab$
```

## Exercise 3: Killing a process

### Task Statement:

Start a process and terminate it using `kill`.

## Command(s):

```
sleep 300 &  
ps aux | grep sleep  
kill <pid>
```

# Output:

```
biswa@biswa-virtual-machine:~/Desktop/Linux lab$ ls
\
arithmetic.sh  background.sh  countlines.sh  exp1.sh  exp4.sh  exp83.sh  fibonacci.sh  file.sh  gcdlcm.sh  interation.sh  palindrome.sh  slayslay  sum.sh  vote.sh  x.sh
armstrong.sh  brkctn.sh     dttime.sh      exp2.sh  exp5.sh  exp84.sh  file2.sh     file.txt  greeting2.sh  kill.sh       prime.sh     sleep.sh  symbols2.sh  while1.sh
biswa@biswa-virtual-machine:~/Desktop/Linux lab$ vim kill.sh
biswa@biswa-virtual-machine:~/Desktop/Linux lab$ ./kill.sh
biswa      2419  0.0  0.0  8288 1920 ?        S    00:14   0:00 sleep 300
biswa      2550  0.0  0.0  8288 1920 pts/1  S+   00:17   0:00 sleep 300
biswa      2552  0.0  0.1  9144 2176 pts/1  S+   00:17   0:00 grep sleep
./kill.sh: line 4: syntax error near unexpected token `newline'
./kill.sh: line 4: `kill <pid>'
biswa@biswa-virtual-machine:~/Desktop/Linux lab$
```

## Exercise 4: Monitoring processes

### Task Statement:

Use `ps` and `top` to monitor processes.

### Command(s):

```
ps aux | head -5
top
```

# Output:

```
top - 08:21:46 up 11 min, 1 user, load average: 0.49, 0.33, 0.22
Tasks: 287 total, 1 running, 286 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.2 us, 0.5 sy, 0.0 ni, 97.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1920.0 total, 193.8 free, 1175.6 used, 766.9 buff/cache
MiB Swap: 6046.0 total, 6046.0 free, 0.0 used, 744.4 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR     S    %CPU  %MEM     time+ COMMAND
 1748 biswa    20   0 3869156 311232 178604 S    3.0 15.8 1:02.01 cinnamon
 1226 root       20   0 502084 186860 108400 S    1.0 9.5 0:11.94 Xorg
 2289 biswa    20   0 566048 59456 36096 S    0.7 3.0 0:01.67 mintreport-tray
 1803 biswa    20   0 246488 36776 29608 S    0.3 1.9 0:00.92 vmtoolsd
 2051 biswa    20   0 1081972 80556 49108 S    0.3 4.1 0:01.23 nemo
 2524 biswa    20   0 545208 46040 33664 S    0.3 2.3 0:00.89 gnome-terminal-
2636 biswa    20   0 14568 5760 3584 R    0.3 0.3 0:00.12 top
   1 root       20   0 22380 13540 9444 S    0.0 0.7 0:01.06 systemd
   2 root       20   0 0 0 0 S    0.0 0.0 0:00.01 kthreadd
   3 root       20   0 0 0 0 S    0.0 0.0 0:00.00 pool_workqueue_release
   4 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-rcu_g
   5 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-rcu_p
   6 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-slub
   7 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-netns
   9 root       20   0 0 0 0 I    0.0 0.0 0:00.02 kworker/0:1-cgroup_destroy
  10 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/0:0H-events_highpri
  11 root       20   0 0 0 0 I    0.0 0.0 0:00.00 kworker/u256:0-ext4-rsv-conversion
  12 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-mm_pe
  13 root       20   0 0 0 0 I    0.0 0.0 0:00.00 rcu_tasks_kthreadd
  14 root       20   0 0 0 0 I    0.0 0.0 0:00.00 rcu_tasks_rude_kthreadd
  15 root       20   0 0 0 0 I    0.0 0.0 0:00.00 rcu_tasks_trace_kthreadd
  16 root       20   0 0 0 0 S    0.5 0.0 0:00.01 ksoftirqd/0
  17 root       20   0 0 0 0 I    0.0 0.0 0:00.07 rcu_preempt
  18 root       rt  0 0 0 0 S    0.5 0.0 0:00.00 migration/0
  19 root      -51  0 0 0 0 S    0.0 0.0 0:00.00 idle_inject/0
  20 root       20   0 0 0 0 S    0.5 0.0 0:00.00 cpuhp/0
  21 root       20   0 0 0 0 S    0.5 0.0 0:00.00 cpuhp/1
  22 root      -51  0 0 0 0 S    0.5 0.0 0:00.00 idle_inject/1
  23 root       rt  0 0 0 0 S    0.5 0.0 0:00.18 migration/1
  24 root       20   0 0 0 0 S    0.0 0.0 0:00.01 ksoftirqd/1
  25 root       20   0 0 0 0 I    0.0 0.0 0:00.00 kworker/1:0-events
  26 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/1:0H-events_highpri
  29 root       20   0 0 0 0 S    0.5 0.0 0:00.00 kdevtmpfs
  30 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-inet_
  31 root       20   0 0 0 0 S    0.5 0.0 0:00.00 kauditd
  34 root       20   0 0 0 0 S    0.5 0.0 0:00.00 khungtaskd
  35 root       20   0 0 0 0 S    0.5 0.0 0:00.00 oom_reaper
  36 root       20   0 0 0 0 I    0.0 0.0 0:00.32 kworker/u256:1-events_power_efficient
  37 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-write
  38 root       20   0 0 0 0 S    0.0 0.0 0:00.01 kcompactd0
  39 root       25   5 0 0 0 S    0.0 0.0 0:00.00 ksmd
  42 root       39 19 0 0 0 S    0.5 0.0 0:00.00 khugepaged
  43 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-kinte
  44 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-kbloc
  45 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-blkcg
  46 root      -51  0 0 0 0 S    0.0 0.0 0:00.00 irq/9-acpi
  47 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-tpm_d
  48 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-ata_s
  49 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-md
  50 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-md_b1
  51 root       0 -20 0 0 0 I    0.0 0.0 0:00.00 kworker/R-edac-
```

## Exercise 5: Using cron for scheduling

### Task Statement:

Schedule a script to run every day at 7:00 AM using cron .

### Command(s):

```
crontab -e
```

```
0 7 * * * /home/retr0/myscript.sh
```

# Exercise 6: Using `at` for one-time scheduling

## Task Statement:

Schedule a script to run once at a specified time using `at` .

## Command(s):

```
echo "/home/user/myscript.sh" | at 08:30
atq
```

## Result

- Learned to create and run shell scripts.
- Managed processes using `background`, `foreground`, and `kill` commands.
- Monitored processes with `ps` and `top` .
- Scheduled recurring tasks with `cron` and one-time tasks with `at` .

## Challenges Faced & Learning Outcomes

- Challenge 1: Remembering the `crontab` time format. Solved by using online `crontab` generators and practice.
- Challenge 2: Ensuring `atd` service is running for `at` command. Fixed by starting the service with `systemctl start atd` .

## Learning:

- Gained hands-on knowledge of process creation and termination.
- Learned job control and scheduling using `cron` and `at` .

## Conclusion

This experiment provided practical experience with shell scripting, process management, and scheduling. These are critical skills for system administrators to automate and control Linux environments effectively.