Operating Systems
CSC2133 (Practical)
Level II - Semester II

Department of Computer Science
Faculty of Science
University of Ruhuna

This course unit designed to do the relevant practical in the Course CSC2123 – Operating Systems.

Practical 02 is designed to be familiar with the advance use of a Linux system.

---

1. Prerequisite - assignment 01 completion.
2. Login to Linux system with your common user ID **bcs2.**
3. Start a shell / terminal.
4. See the contents of your current directory.
5. Make a directory called **MYTEST** in your home directory.
6. Change to **MYTEST** & make a file called **me.txt** with your personal details ($ **vi me.txt**).
7. Make a directory called **ME** in **MYTEST**.
8. Move **me.txt** to the **ME** directory.
9. Copy **me.txt** to your **home** directory.
10. Remove **me.txt** from your **home** directory.
11. Try to copy **me.txt** to with another name in to your home directory.
12. Delete the **MYTEST** directory and its entire content using a single command (use man page)
13. Play with **uname** command ($ **uname**).(use man page)
14. Try to identify your home directory path ($ **echo $HOME**).
15. How many disk partitions are on your system ($ **df**).
16. Tyr to get disk partitions information in human readable format.
17. Identify directories in the system where executable files can be found ($ echo **$PATH**).
18. Identify path of commands such as ls, wc & httpd ($ **which ls**).
19. Identify locations of the binary, source, and manual page files for **pwd** & **sendmail** ($ **whereis pwd**).
20. Using command prompt go to the **tmp** directory in **/var** ($ **cd /var/tmp**).
21. Now go to **share** in **/usr**
22. Change to the **/proc** directory and try followings
    - **I.** What CPU system is running on? ($ less **cpuinfo).**
    - **II.** How much RAM does it currently use? ($ less **meminfo).**
    - **III.** How much swap space do you have?
    - **IV.** How many hours has the system been running? ($ **uptime).**
23. Change to the **/etc** directory and try followings
    - **I.** Which release are you running? ($ **cat issue).**
    - II. Which shells are on the system ($ **cat shells**).
    - III. Which is the shell, you are using? ($ **echo $SHELL**).
    - IV. Switch to several deferent shells & exit.
24. What is the difference between **cat** and **less** command?
25. Observe processes running and system load on your system ($ **top**)
26. Get other terminal and Give **xtrem** command in shell see what happened. ($ **xterm**) {Which shell you can use, parent one or child one}
27. Send another **xterm** to background ($ **xterm &**).
    {Which shell you can use, parent one or child one}
28. Observe jobs in the current shell. **($** j**obs**).

29. Send another **xterm** to background & Observe jobs in the current shell.
    Get another **xterm** to foreground and suspend the **xterm** ($ **xterm**), ($ **Ctrl+Z**).
    {Which shell you can use, parent one or child one}
30. Observe jobs in the current shell. ($ **jobs**).
31. Get suspended job to foreground and again suspend ($ **fg ….**).
32. Get suspended job to foreground and send to background. ($ **bg ….**).
33. Kill one job from suspend jobs. ($ **kill %….**)
34. Start two **xclock** programs as one in foreground & one in background.
35. Suspend foreground clock. After two minutes see the different of clocks.
36. Send suspended clock to background.
37. Stop clocks using the kill command.
38. Run the **xcalc** directly in the background, so that the prompt of the issuing terminal is released.
39. Run the command **find /**, what effect does it have on system load & stop it.
40. How long does it take to execute ls in the current directory & home ($ **time ls**)
41. List current shell and processes ($ **ps**).
42. Display all processes (use man page).
43. Try to understand **ps -e** (or **ps -ef**) output.
44. Display a tree of processes & identify xsession ($ **pstree**).
45. Start two terminals and identify your terminal processes.
46. Get an xterm & display, identify how processes tree changes.
47. Get an xterm using previous xterm & display, identify how processes tree changes.
48. Try to understand the usage of **ipcs** command.
49. List all **ipc** facilities.
50. List all semaphores using **ipcs.** (-s)
51. displays current usage for all the IPC facility (-u)
52. List process id which accessed the corresponding ipc facility very recently.(-m, -p)
53. List common process signal ($ **kill -l**)
54. What does **kill -9 -1** do?
55. Start two terminals and identify both terminal connected to standard input file ($ **tty**).
56. Send a "Hello terminal" massage from 1st to 2nd ($ **write ……. ……..**), ($ **Ctrl+C**).
57. Create an **at** job (Schedule) to display system time on current terminal after two minutes
    ($ **at <time>**, then enter, then type **echo date > /dev/<your terminal>** ,then enter, then Ctrl+D) wait 2 minutes.
58. Create an **at** schedule to display your name on current terminal in specific time.
59. Create a file called **command.sh** with list of your favorite 5 Linux commands and save.
60. Grant executable permissions to **command.sh** file. ($ **chmod +x …….**).
61. Test **command.sh.** ($ **./…….**).
62. Create an **at** schedule to automate **command.sh.**
63. Create a **crontab** to display date of the system on the terminal after two minutes.
    ($ **crontab –e**)
    File format

```
minute / hour / day of the month / month of the year / day of the week/ command
      12  4  *     *    * date > /dev/pts/0
```

64. With **crontab** try to follow 50 to 53 steps.
65. Modify your **crontab** output direct to a file called mycron.txt.
66. Try to automate steps 4 to 10 using **crontab**
67. Try to understand usage of **cut** command.
68. Try to display only all user names in /etc/passwd file using cut command.
69. Try to understand usage of **dmesg** command.
70. Try to read startup messages to be viewed one screen at a time. (pipeline with less)
71. Try to see content of /var/log/dmesg.
72. Try to view Ethernet link status. (grep eth)

73. Try to understand usage of **grep** command.
74. Try to show how much physical memory (RAM) is available on the system using dmesg and grep combination. (-i memory)
75. Try to clear all contents in startup messages.
76. Try to start several Firefox instances.
77. Find out PIDs of Firefox (ps aux | grep firefox).
78. Apply **killall** to Firefox.
79. Try to find out current run level of the system ($**who -r**)
80. Try to change default run level of the system to text mode editing **/etc/etc/inittab** and reboot the system.