THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA

SS BE-III (CSE) (2022-2026)

CSE1603: Operating Systems

Lab Assignments

April 2025 Semester

Marks : 50

LAB 1 :

Objective : How is Shell Scripting useful to you as a Professional Software Engineer?

ii) Try to answer the following points :

- how it can help automate my tasks
- simplifying my work flows
- managing system operations

ii) Showcase 5 real world Case Studies on using Shell Scripts.

LAB 2 :

Objective : Write the following shell scripts in order to learn their syntax.

1. Write a script to print "Hello, World!".
2. Create a script to check if a number is even or odd.
3. Write a simple calculator script for addition, subtraction, multiplication, and division.
4. Check if a given name is a file or a directory.
5. Write a script to display the current date and time.
6. Create a script to check if a string is a palindrome.
7. Print numbers from 1 to N based on user input.
8. Count and display the number of files in the current directory.
9. Write a script to display the system uptime.
10. Create a backup script for a directory provided by the user.

THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA

SS BE-III (CSE) (2022-2026)

CSE1603: Operating Systems

Lab Assignments

April 2025 Semester

Marks : 50

LAB 3 :

Objective : Continue learning syntax of shell scripts via advanced demo programs

Write Shell Scripts for the following tasks.

1. Write a shell script to print the addition of two numbers.
2. Print a given number in reverse order.
3. Read 'n' from the user and print the Fibonacci sequence until 'n'.
4. Say "Hello to a user and greet them based on the time of the day (Good Morning / Eve etc.)
5. Write a script for printing all file related information in present working directory (size, permissions etc.)
6. Print the length of each and every string using arrays.
7. Display the longest and the shortest usernames on a system.
8. Generate five random 8 character passwords having alpha-numeric characters.
9. Display the names of all file systems which have less than 10% free space available.
10. Write a script to search whether a user exists on the system or not.
11. Display the current date in words.
12. Display the current active time of a user that has been logged in.
13. Print the total number of lines in a C program.
14. Find whether a C file contains the printf() method or not.
15. Find whether a C program uses void main () or int main().

LAB 4 and 5 :

Objective : Understanding and testing various linux commands to see theoretical Operating System concepts in live action

A. Process Management

1. ps - View running processes.
2. top or htop - Monitor system processes in real-time.
3. kill - Terminate a process using its PID.
4. jobs - List active background jobs.
5. fg and bg - Move jobs between foreground and background.

THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA

SS BE-III (CSE) (2022-2026)

CSE1603: Operating Systems

Lab Assignments

April 2025 Semester

Marks : 50

6. nice and renice - Adjust process priorities.
7. strace - Trace system calls of a process.
8. lsof - List open files for a process.

Lab Exercises:

1. Use ps to find and terminate a process.
2. Start a long-running process (e.g., sleep 1000), then move it to the backgroun and bring back to the foreground.
3. Change the priority of a process using nice or renice.

B. Memory Management Commands

1. free - Display system memory usage.
2. vmstat - Report memory, CPU, and I/O statistics.
3. uptime - Show system uptime and load average.
4. dmesg - Kernel ring buffer messages, including memory-related logs.
5. Advanced Tasks:
6. cat /proc/meminfo - View detailed memory information.
7. cat /proc/<PID>/status - Inspect memory usage of a specific process.
8. watch free -h - Continuously monitor memory usage.

Lab Exercises:

1. Compare output from free and /proc/meminfo.
2. Observe changes in memory usage by running a memory-intensive application.

C. Input/Output Management Commands

1. iostat - Monitor I/O device usage.
2. df - Display disk space usage.
3. du - Show disk usage of files and directories.
4. lsblk - List information about block devices.
5. Advanced Tasks:
6. iotop - Monitor I/O usage by processes.
7. udevadm - Manage device events.
8. dd - Perform low-level data copying.

THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA

SS BE-III (CSE) (2022-2026)

CSE1603: Operating Systems

Lab Assignments

April 2025 Semester

Marks : 50

D. Device Management:

1. ls /dev - View device files.
2. mount and umount - Manage mounting of filesystems.
3. blkid - Get UUID of storage devices.

Lab Exercises:

1. Use df and du to analyze disk space usage.
2. Run iotop and observe I/O usage when copying a large file.
3. Mount a USB drive and inspect its filesystem.

E. File System Management

1. ls, cd, pwd - Navigate file systems.
2. touch, mkdir, rm - Create and delete files/directories.
3. cp, mv - Copy and move files.
4. stat - Display detailed information about a file.
5. chmod, chown - Modify file permissions and ownership.
6. ln - Create hard and symbolic links.
7. find and locate - Search for files and directories.
8. tar, gzip, zip - Archive and compress files.

F. File System Inspection:

1. fsck - Check and repair file systems.
2. df -T - Display file system type.
3. mount - View all mounted file systems.

Lab Exercises:

1. Search for files modified within the last 7 days using find.
2. Create a tarball of a directory and extract it.

G. System Monitoring

1. uptime - Show system uptime and load averages.
2. w - Display who is logged in and what they are doing.
3. vmstat - View system performance metrics.
4. sar - Historical system monitoring (requires sysstat package).

THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA
SS BE-III (CSE) (2022-2026)
CSE1603: Operating Systems

Lab Assignments
April 2025 Semester

Marks : 50

5. sysctl - Modify kernel parameters at runtime.
6. sar - Analyze resource usage over time.

Lab Exercises:

1. Use uptime to monitor load averages at different times.
2. Analyze CPU and I/O usage with vmstat.
3. Modify kernel parameters using sysctl.

H. Networking

1. ping - Test network connectivity.
2. ifconfig or ip addr - Display network interface details.
3. netstat or ss - View active connections.
4. traceroute - Trace network paths.
5. curl or wget - Fetch content from a URL.
6. iptables - Configure firewall rules.

Lab Exercises:

1. Test network connectivity to a server using ping and traceroute.
2. Use netstat or ss to view open ports and connections.
3. Fetch the content of a webpage using curl.

I. Advanced Shell Scripting

1. Write a shell script to monitor disk usage and send alerts.
2. Log system resource usage to a file.
3. Write a script to find the 5 largest files in a directory.
4. Automate backup of a directory using tar in a script.
5. Write a script to monitor and log CPU usage

Lab 6 and 7 :

1. Basic Process Creation

Objective: Understanding process creation using `fork()` in C.

Task:

- Write a C program that creates a child process using **`fork()`**.

THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA
SS BE-III (CSE) (2022-2026)
CSE1603: Operating Systems

Lab Assignments
April 2025 Semester

Marks : 50

- Print the process IDs **(`PID`)** of both parent and child.
- Use **`getpid()`** and **`getppid()`** to display process details.

Expected Output:

- Display parent and child PIDs.
- Identify whether execution happens in the child or parent process.

---

2. Process Synchronization Using `wait()`

Objective: Understanding how a parent waits for a child process to finish execution.

Task:

- Modify the previous program to make the parent wait for the child to complete execution using **`wait()`**
- Print messages before and after the child process exits.

Expected Output:

- The parent should wait for the child to finish before printing its final message.

---

3. Process Execution using `exec()`

Objective: Executing a new program within a child process.

Task:

- Create a C program where the child process replaces itself with another program (e.g., **`/bin/ls`**) using **`execlp().`**
- The parent process waits for the child to complete execution.

Expected Output:

- The child process successfully replaces itself with the `ls` command output.

---

4. Inter-Process Communication using Pipes

Objective: Using pipes for communication between processes.

Task:

- Create a parent and child process.
- The parent sends a message to the child process via a pipe.
- The child reads the message from the pipe and prints it.

Expected Output:

- The child successfully reads and displays the message sent by the parent.

---

5. Process Scheduling Simulation

Objective: Implementing a basic Round Robin scheduling algorithm.

Task:

- Simulate Round Robin scheduling for multiple processes with different burst times.
- Assume a time quantum and display process execution order.

Expected Output:

- A sequence of time slices showing process execution, demonstrating context switching.
