Capstone Project

Assignment 1(LinuxOS)	Assignment 2(LinuxOS)	Assignment 3(LSP)	Assignment 4(LSP)	Assignment 5(LinuxOS and LSP)
File Explorer Application **Objective:** Develop a console-based file explorer application in C++ that interfaces with the Linux operating system to manage files and directories. **Day-wise Tasks:** Day 1: Design the application structure and setup the development environment. Start with basic file operations like listing files in a directory. Day 2: Implement file and directory navigation features. Enable the user to move through directories. Day 3: Add file manipulation capabilities (copy, move, delete, create). Day 4: Implement file search functionality within the file explorer. Day 5: Add file permission management features.	**Custom Shell Implementation** **Objective:** Build a simple shell in C++ that can execute commands, manage processes, and handle redirection and piping. **Day-wise Tasks**: Day 1: Plan the shell features and parse user input. Day 2: Implement execution of basic commands through the shell. Day 3: Add support for process management (foreground, background processes). Day 4: Implement piping and redirection features. Day 5: Incorporate job control (listing jobs, bringing jobs to foreground/background).	**System Monitor Tool** **Objective:** Create a system monitor tool in C++ that displays real-time information about system processes, memory usage, and CPU load, similar to the 'top' command. **Day-wise Tasks**: Day 1: Design UI layout and gather system data using system calls. Day 2: Display process list with CPU and memory usage. Day 3: Implement process sorting by CPU and memory usage. Day 4: Add functionality to kill processes. Day 5: Implement real-time update feature to refresh data every few seconds.	**Network File Sharing Server & Client** **Objective:** Develop a networked file sharing application with a server and client architecture, enabling file transfers over sockets. **Day-wise Tasks**: Day 1: Setup server-client socket communication. Day 2: Implement file listing and selection feature on the client side. Day 3: Enable file transfer from server to client. Day 4: Add file upload functionality from client to server. Day 5: Implement security features like authentication and encryption	**Bash Scripting Suite for System Maintenance** Objective: Write a suite of Bash scripts to automate system maintenance tasks such as backup, system updates, and log monitoring. **Day-wise Tasks**: Day 1: Write a script for automated system backups. Day 2: Create a script to perform system updates and clean up. Day 3: Develop a log monitoring script to alert on certain conditions. Day 4: Combine scripts into a maintenance suite with a menu to execute them. Day 5: Test scripts and add error handling and logging functionalities.