

Git link :

<https://github.com/srikanthyadav007/Leightweight-Unix-Shell>

Lightweight UNIX Shell in C

In this project, I developed a command-line shell in C that can execute standard Linux commands, manage background processes, and handle multiple commands in both sequential and parallel modes. The project was divided into several key components:

Part A: Basic Shell Functionality

I created a shell that can execute commands like `ls`, `cat`, and `echo` using the `fork()`, `exec()`, and `wait()` system calls. The shell continuously processes user input, tokenizes commands, and handles errors when commands are incorrect.

Part B: Background Process Handling

The shell was extended to support background execution, allowing commands followed by `&` to run in the background while the shell remains responsive. It also manages and cleans up background processes to prevent the creation of zombie processes.

Part C: Exit Command

An exit command was implemented to allow the user to gracefully terminate the shell, ensuring all resources are cleaned up and any running background processes are stopped.

Part D: Ctrl+C Signal Handling

I added a custom signal handler to manage SIGINT (Ctrl+C) signals, allowing the shell to continue running while only the active foreground process is terminated.

Part E: Serial and Parallel Command Execution

The shell was further enhanced to execute multiple commands in sequence (&&) or concurrently (&&&). It effectively manages and cleans up parallel processes before returning control to the command prompt.

Conclusion

This assignment provided a deep understanding of process management, signal handling, and user input parsing in Linux. By implementing a simple shell, I gained practical experience with system calls such as `fork()`, `exec()`, `wait()`, and `kill()`. The shell was designed to handle both foreground and background processes, manage multiple commands, and gracefully handle errors and signals. This implementation forms the foundation for more complex shell functionalities in the future.