OS ASSIGNMENT BY GROUP-71

GROUP MEMBERS CONTRIBUTION:

1) Ayush Singhal: Implemented different commands of the execvp family and implemented various other functions.

2) Sanjeet Kumar Patel: Implemented pipe part and different functions in the code.

This assignment was a combined effort from both of us where we both have contributed to our best.

- **The code represents a simple shell program that continuously accepts and executes user commands. It supports single commands and piped commands. Key features include:**
- **Command Parsing: User input is tokenized into commands and arguments, stored in a linked list for history tracking.**
- **Signal Handling: The program handles the SIGINT signal (Ctrl+C) by printing the command history and associated process details.**
- **Execution: The executeCommand() function forks a child process to execute a single command, tracking process ID, start time, and execution duration.**
- **Pipe Handling: The handlePipes() function manages piped commands, creating child processes for each part of the pipeline, connecting their input and output as needed.**
- **History Command: Typing "history" displays the list of executed commands.**
- **Memory Management: The program ensures proper memory allocation and deallocation for command structures.**
- **Overall, the code provides basic shell functionality with command history and pipe support, making it a simple yet functional shell implementation.**

******** We have done both the bonus part ********

We will also submit our test case files along with our main file.

***** Our code couldn’t implement cd command because cd command is not provided by the execvp system call.**

GITHUB LINK:

LINK