Contribution

Both contributed equally

Github Repository

[Sanchay117/OS-Shell: Assignment 2 Of Operating Systems Course (github.com)](https://github.com/Sanchay117/OS-Shell)

SimpleShell Implementation

This program implements a custom shell that supports command execution with pipe functionality, using child processes. Here's a brief explanation of key parts:

1. **Command Input**: The read\_user\_input() function reads commands from the user using getline(). The shell also removes leading and trailing spaces from the commands.

2. **Command Splitting**: The split\_command() function breaks commands into smaller parts, accounting for spaces and pipes (|).

3. **Piping**: The shell handles commands connected by pipes, redirecting outputs from one process to another with pipe() and dup2().

4. **History**: The shell maintains a history of commands, stored in the history array.

5. **Execution**: Child processes are created with fork(), and commands are executed via execvp().

6. When pressing Ctrl+C, the system displays the process ID , the exact time the process was executed, and the total execution time up to that point.

7. **Removing Extra Spaces(Leading/Trailing):**This function removes leading and trailing spaces from a string and also reduces multiple spaces between words to a single space.

Commands not Supported

**1. cd:** Since our shell is very simple and static changing directories presents a problem

**2.exit:**We are using control+C to exit so this won’t work also since we are technically running a C program not a shell typing exit won’t work

**3.redirection:** Something like echo Hello > output.txt won’t work as we have only implemented pipes for ‘|’ not for ‘>’.

**4.environment variables:** export VAR=123 won’t work as in our simple shell implementation, the environment variables are not properly managed or updated.

**5.complex commands/scripts:** if statements and for loops also not supported in our implementation as it is outside the scope of this assignment.