Custom Shell with Piping and Background Execution

Rahul Agarwal

September 25, 2024

Introduction

This project implements a simple custom shell in C with the following features:

- Basic command execution
- Support for piping ("|")
- Background execution of processes using '&'
- Signal handling ("Ctrl+C")
- Command history with process ID ("PID") and execution time

Code Overview

The shell program supports the execution of commands, handling background processes, and piping multiple commands. The code maintains a history of executed commands, including their respective PIDs and time taken for execution.

Main Features

- **Piping Support:** Commands separated by pipes ('—') are executed sequentially, with the output of one command passed as input to the next.
- **Background Execution:** Commands followed by '' will run in the background, allowing the user to continue using the shell.
- **Signal Handling:** The shell handles the 'SIGINT' signal (Ctrl+C) by terminating only the child process, not the shell itself.
- **Command History:** Each command executed is stored along with its PID and the time it took to complete.

How to Compile

To compile the shell program, use the following command:

```
gcc -o assign assign2.c
```

How to Run

To run the shell program, use the following command:

```
./assign
```

Example Commands

Here are some examples of how to use the shell:

- Running a command in the background:

```
    sleep 10 &
```

- Piping commands:

```
    ls -l | grep ".c" | sort
```

- Running a command normally:

```
    ls -l
```

- Viewing command history (at exit):

```
    exit
```

License