

Simple UNIX-shell

To implement a simple POSIX/Unix shell in C++.

Author: Neeraj Sahasrabudhe

A simple shell implemented in C++ that supports basic command execution, argument parsing, and handling of quoted strings. This project demonstrates the fundamental concepts of process creation and management using system calls like `fork`, `execvp`, and `waitpid`. Error handling for arguments with improper quotes has also been demonstrated.

Primary Features

- **Command Execution:** Executes standard Unix commands by creating child processes.
- **Argument Parsing:** Parses command-line input into individual arguments, including support for quoted strings.
- **Error Handling:** Handles common errors such as command not found, mismatched quotes, and invalid usage of built-in commands.
- **Custom Prompt:** Displays a custom shell prompt (\$) to accept user input.

Getting Started

Prerequisites

- **C++ Compiler:** C++ compiler installed (e.g., `g++`).
- **Linux Environment:** This project is designed to run in a Unix-like environment such as Linux.

Running the Shell

1. Compile the source code:

```
g++ -o simple_shell simple_shell.cpp
```

2. Run the shell:

```
./simple_shell
```

Usage

Once the shell is running, you can type any valid Unix command and press Enter to execute it. For example:

```
$ ls -l
$ pwd
$ echo "This is Maverick"
```

```
$ cd /path/to/any/directory
$ exit
```

The shell exit and returns control when `exit()` is called or if it reaches the EOF.

Testing

Tests similar to those shown in the challenge were executed. Error messages with specific error codes are thrown on `stderr`. A screenshot of commands executed and the respective results displayed by the shell has been included in the submission files.