MyShell Project - Brief Description

Introduction

MyShell is a simple command-line shell program written in C that mimics the basic functionality of Unix shells like 'bash' or 'sh'. It allows users to execute common commands, supports output redirection, and includes built-in commands like 'echo'. This project aims to demonstrate the fundamentals of building a shell and provides practical experience with process creation, command parsing, and system calls.

Key Features

1. Basic Command Execution: Users can run standard Unix commands like 'ls', 'cat', and 'pwd' from the MyShell prompt.
2. Built-in 'echo' Command: MyShell includes an implementation of the 'echo' command to print arguments directly from the shell.
3. Output Redirection ('>'): Commands can redirect output to a file using '>', allowing users to save command output to files.
4. Process Management: The shell uses 'fork()' and 'execvp()' to create child processes, enabling the execution of commands in a separate process.
5. Interactive Shell Prompt: Users can interact with the shell in a loop until they type 'exit' to quit.

Usage

- Compile the program:

```
  $ gcc myshell.c -o myshell
```

- Run the shell:

```
  $ ./myshell
```

- Example commands in MyShell:

```
  myshell> echo "Hello World"
  Hello World


  myshell> ls -l > output.txt
  myshell> cat output.txt
```

Technologies Used

- C Programming Language: Core programming language used for creating the shell.
- POSIX System Calls: Utilizes system calls like 'fork()', 'execvp()', and 'dup2()' to manage processes and handle redirection.

Future Improvements

- Input Redirection ('<') and Command Piping ('|'): Add support for more advanced command features.
- More Built-in Commands: Implement commands like 'cd' for changing directories and 'history' for viewing previous commands.
- Background Execution ('&'): Add support for running commands in the background.