

Prakhar Yadav
2019H1030032G
4 September 2019

Assignment 2

Advanced OS Shell implementation - PSHELL

Introduction

Custom shell implementation in C, having various built-in commands and ability to execute the commands which are supported by Linux.

Running

To compile the program:

```
>> gcc psh3.c -lreadline -o psh3
```

On the command prompt enter the object:

```
>>psh3
```

To enter the shell:

```
>>START
```

To exit the shell at this point:

```
>>END
```

upon entering the shell use the "help" command to navigate through the shell.

upon exiting the shell using "exit" command, end the program with END command, or re-enter the shell using START

Modules

Core modules

- void psh_init() : initialise shell by calling required modules
- char *psh_read() : read the input from user
- char **psh_parse(char *) : tokenise the input from user and separate into words
- int psh_launch(char **) : module to run commands that are not built-in

Function Declarations of builtin shell commands:

- int psh_cd(char **args) : implementation of change directory command
- int psh_help(char **args) : implementation of help command
- int psh_exit(char **args) : implementation of exit command
- int psh_num_builtins() : function returns number of built-in commands
- int psh_pwd(char **) : implementation of present working directory command
- int psh_man(char **) : implementation of man command
- int psh_history(char **) : implementation of history command
- int psh_run(char **) : module responsible to run all the commands that are passed by the user.

Function declarations for implementing pipe

- void psh_parseSpace(char *, char **) : to parse the input if pipe is found
- int psh_parsePipe(char *str, char **stripped) : to separate commands before and after pipe
- void psh_execArgsPiped(char **, char **) : to execute commands if pipe is found in input
- int psh_processString(char *, char **, char **) : checks for presence of pipe and calls necessary functions for parsing and executing the pipe commands

Built With

- Linux
- C
- VS Code

Authors

Prakhar Yadav - 2019H1030032G - [h20190032@goa.bits-pilani.ac.in]

Acknowledgments

- *GeeksForGeeks* [<https://www.geeksforgeeks.org/making-linux-shell-c/>]
- *Stephen Brennan* [<https://brennan.io/2015/01/16/write-a-shell-in-c/>]

Special mention

I would like to thank the following for their valuable advises and constant support during the development of the shell:

- Dhairya Parikh [2019H1030906G]
- Aakash Mehta [2019H1030024G]
- Parth Shah [2019H1030563G]
- Nishi Singh [2019H1030108G]