# Simple Shell

## Description

The \*\*Simple Shell\*\* project is a custom implementation of a command-line interpreter, similar to the Unix shell (e.g., `sh`, `bash`). It provides an environment where users can input and execute commands, including built-in commands and system commands, and manage environment variables.

Requirements

General

* Allowed editors: vi, vim, emacs
* All your files will be compiled on Ubuntu 20.04 LTS using gcc, using the options -Wall -Werror -Wextra -pedantic -std=gnu89
* All your files should end with a new line
* A README.md file, at the root of the folder of the project is mandatory
* Your code should use the Betty style. It will be checked using [betty-style.pl](https://github.com/hs-hq/Betty/blob/master/betty-style.pl) and [betty-doc.pl](https://github.com/hs-hq/Betty/blob/master/betty-doc.pl)
* Your shell should not have any memory leaks
* No more than 5 functions per file
* All your header files should be include guarded
* Use system calls only when you need to ([why?](https://intranet.hbtn.io/rltoken/rp53OodD6JzhS5Cv4DHkxQ))

List of allowed functions and system calls+

* all functions from string.h
* access (man 2 access)
* chdir (man 2 chdir)
* close (man 2 close)
* closedir (man 3 closedir)
* execve (man 2 execve)
* exit (man 3 exit)
* \_exit (man 2 \_exit)
* fflush (man 3 fflush)
* fork (man 2 fork)
* free (man 3 free)
* getcwd (man 3 getcwd)
* getline (man 3 getline)
* getpid (man 2 getpid)
* isatty (man 3 isatty)
* kill (man 2 kill)
* malloc (man 3 malloc)
* open (man 2 open)
* opendir (man 3 opendir)
* perror (man 3 perror)
* printf (man 3 printf)
* fprintf (man 3 fprintf)
* vfprintf (man 3 vfprintf)
* sprintf (man 3 sprintf)
* putchar (man 3 putchar)
* read (man 2 read)
* readdir (man 3 readdir)
* signal (man 2 signal)
* stat (\_\_xstat) (man 2 stat)
* lstat (\_\_lxstat) (man 2 lstat)
* fstat (\_\_fxstat) (man 2 fstat)
* strtok (man 3 strtok)
* wait (man 2 wait)
* waitpid (man 2 waitpid)
* wait3 (man 2 wait3)
* wait4 (man 2 wait4)
* write (man 2 write)

---

## Features

* Displays a prompt for user input.
* Reads user commands from standard input.
* Tokenizes the input command for parsing.
* Searches for executable command paths using the PATH environment variable.
* Executes commands with or without arguments.
* Handles built-in commands like exit and env.
* Implements signal handling to prevent interruptions (e.g., CTRL+C).
* Manages common errors during command execution.

##files

| File | Description |
| --- | --- |
| main.c | Entry point of the program. Initializes the shell and runs the main loop. |
| print\_prompt.c | Handles the display of the shell prompt. |
| read\_line.c | Reads user input from the command line. |
| tokensize.c | Tokenizes the user input to extract arguments. |
| find\_command\_path.c | Searches for the full path of commands based on the PATH environment variable. |
| execute\_command.c | Executes user-entered commands. |
| builtin.c | Implements built-in commands such as exit and env. |
| \_getenv.c | Provides a function to access environment variables. |
| signal.c | Handles signals to ensure the program is not interrupted by signals like SIGINT. |
| shell.h | Header file containing function prototypes, constants, and necessary inclusions for the project. |
| man\_1\_simple\_shell | Manual page describing the functionality and usage of the shell. |

---

### Prerequisites

- GCC Compiler

- Linux-based system (or WSL for Windows)

### Installation

1. Clone the repository:

```bash

git clone holbertonschool-simple\_shell

```

2. Navigate to the project directory:

```bash

cd holbertonschool-simple\_shell

```

3. Compile the shell:

```bash

gcc -Wall -Werror -Wextra -pedantic -std=gnu89 \*.c -o hsh

---

## Flowchart

```

---

## Authors

- \*\*Madjiguéne Elodie Mbaye\*\*

- \*\*Mame Penda Sadio\*\*

---

## License

This project is licensed under the MIT License

---

## Acknowledgments

- \*\*Holberton School\*\*: For guidance and resources.

- \*\*Unix Community\*\*: For the inspiration behind the project.