

## Assignment 3 – Simple Shell

### Description:

The goal of this assignment is to understand the use of concepts such as processes and the use of fork, exec, and wait functions, understanding error handling, Reading user input, etc. We will build a simpleshell

### Approach:

I have to research and make sure I understand the learning objectives of the assignments. That includes:

- Understanding processes and the use of fork, exec, and wait functions
- Understanding error handling
- Reading user input
- Tokenizing input
- Getting user input and processing end-of-file condition
- Understanding and using Vectors (arrays of pointers)
- Comprehensive use of redirection of standard input and output and the use of pipes

I also believe i would have to draw the underlying structure of my shell

### Sources:

<https://www.youtube.com/watch?v=GFdrztZUt8U>  
<https://stackoverflow.com/questions/8389033/implementation-of-multiple-pipes-in-c>  
<https://www.geeksforgeeks.org/command-line-arguments-in-c-cpp/>  
<https://man7.org/linux/man-pages/man2/dup.2.html>  
<https://www.youtube.com/watch?v=cex9XrZCU14&list=PLfqABt5AS4FkW5mOn2Tn9ZZLLDwA3kZUY>

### Issues and Resolutions:

My initial structure consists of building all the programs in the main function. I decided to change my structure and have it more organized to break down the different concepts of the problem. I structured my code like this: The biggest problem I found is to get the right structure and make sure it is logical. So a lot of time has been spent to refine the structure.

### Main function:

contains the input from the command line arguments and serves as the entry point into the shell. it will use a loop to continuously read commands from the user. The main function basically uses all the other functions I made from the main.

**Parse function:**

This function will take a command line string and an array of pointers. it will iterate through the command line and replace all the white spaces with null characters and update the value of argv in order.

**NumPipes Function:**

This function will accept an array of strings and will count the occurrences of the pipe character. This will serve the purpose of determining whether or not a command involves a pipe. This will help us know how many pipes we have in our input and determine potentially how many times we would have to loop after the first pipe.

**ExecuteFunctionPipes:**

function is responsible for executing commands with pipes. It finds the position of the pipe character in the argument array, separates the arguments before and after the pipe, creates two child processes, establishes a pipe between them, and redirects standard input/output.

**ExecuteFunction:**

Function is responsible for executing commands without pipes. I believe it would be easier this way as the structure is clearer and more organized. It creates a child process using fork(), and in the child process, it executes the specified command using execvp. The parent process waits for the child to complete its execution before proceeding, ensuring proper synchronization between parent and child processes.

Understanding of when to close fd[2]  
and use of dup2  
understanding of how to deal with multiple pipes.

(check sources)

**Analysis:****Screenshot of compilation:**

```
student@student: ~/Documents/csc415-assignment3-simple...
student@student:~/Documents/csc415-assignment3-simpleshell-Aymane-Arfaoui$ make
make: 'arfaoui_aymane_HW3_main' is up to date.
student@student:~/Documents/csc415-assignment3-simpleshell-Aymane-Arfaoui$
```

Screen shot(s) of the execution of the program:

```
student@student:~/Documents/csc415-assignment3-simpleshell-Aymane-Arfaoui$ make run < commands.txt
./arfaoui_aymane_HW3_main "Prompt> "
Prompt>
Child PID: 0
Return Status: 65535
arfaoui_aymane_HW3_main  arfaoui_aymane_HW3_main.o  Makefile  README.md
arfaoui_aymane_HW3_main.c  commands.txt  prompts
Prompt> Prompt>
Child PID: 0
Return Status: 65535
"Hello World"
Prompt> Prompt> Prompt>
Child PID: 0
Return Status: 65535
total 76
drwxrwxr-x 3 student student 4096 Feb 29 21:57 .
drwxr-xr-x 6 student student 4096 Feb 29 20:18 ..
-rwxrwxr-x 1 student student 18752 Feb 29 21:57 arfaoui_aymane_HW3_main
-rw-rw-r-- 1 student student 5317 Feb 29 21:57 arfaoui_aymane_HW3_main.c
-rw-rw-r-- 1 student student 15112 Feb 29 21:57 arfaoui_aymane_HW3_main.o
-rw-rw-r-- 1 student student 64 Feb 29 19:24 commands.txt
drwxrwxr-x 8 student student 4096 Feb 22 14:43 .git
-rw-rw-r-- 1 student student 1867 Feb 22 14:52 Makefile
-rw-rw-r-- 1 student student 56 Feb 22 14:53 prompts
-rw-rw-r-- 1 student student 7638 Feb 21 13:11 README.md
Prompt> Prompt> Prompt> Prompt>
Child PID: 0
Return Status: 65535
  PID TTY          TIME CMD
 178445 pts/0    00:00:00 bash
 178682 pts/0    00:00:00 make
 178683 pts/0    00:00:00 sh
 178684 pts/0    00:00:00 arfaoui_aymane_
 178688 pts/0    00:00:00 ps
Prompt> Prompt> Prompt> Prompt> Prompt>
Child PID: 0
Return Status: -1176693968
Prompt> Prompt> Prompt> Prompt> Prompt>
Child PID: 0
Return Status: -1176693968
64
Prompt> Prompt> Prompt> Prompt> Prompt> Prompt>
Child PID: 0
Return Status: 65535
ls: cannot access 'foo': No such file or directory
Prompt> Prompt> Prompt> Prompt> Prompt> End of file reached. Exiting.
student@student:~/Documents/csc415-assignment3-simpleshell-Aymane-Arfaoui$
```