

## Assignment 3 – Simple Shell

### Description:

Writing a simple shell by using `fork()` and `execvp()` to handle commands. must take in a line of text and tokenize it to be ready for `execvp()`. Delim by space, tab, newline(whitespaces).Need to handle EOF and blank spaces.

### Approach / What I Did:

First I went to look at how to use `execvp()` at this [website](#) that was posted on slack and discord which helped a lot on what needed to be passed and what it returns. Next to get a line of text, I needed to learn to use `getline()`, where the first argument is the pointer in my case buffer, then the size of the buffer, then `stdin` which is the file descriptor. This ties to the `feof()` where I saw at this [website](#). Using `feof` will help determining when you reach the end of a file so it doesn't go forever. First to handle the input of a prompt, use `argv[1]` the second index to get what the prompt was. If there wasn't any prompt, then default to ">". Next will be making the loop for the shell which will be infinite till `exit()` is called. Next I used the slides for reference on the switch case to handle when the fork returns 0(the child), -1(error), positive number(parent), where the child will do `execvp` and output an error if there is one, error if fork fails and the parent where it shows the child pid, and exit status. I also put a `strcmp` for when "exit" is entered so it can exit the shell. Before that there is a wait till the child exits. Finally at the end, I set the `argList` and `buffer` to null, then free the allocated memory.

### Issues and Resolutions:

The first problem was figuring out how to get input from a line, at first I thought it was in the main, but realized that would not work since `argv` would not be passed again, so I looked up the `getline()` function [here](#). A problem I was stuck on was how to compare the command to the string "exit". It gave me an error where they could not be compared. Someone in the discord suggested using `strcmp` for it and once I looked up how to use it on [geeksforgeeks](#), it helped compare the way I wanted it to.

**Analysis:** N/A

### Screen shot of compilation:

```
File Edit View Search Terminal Help
student@student-VirtualBox:~/csc415-assignment3-simpleshell-kchanw$ make run
gcc -c -o Chan_Kurtis_HW3_main.o Chan_Kurtis_HW3_main.c -g -I.
gcc -o Chan_Kurtis_HW3_main Chan_Kurtis_HW3_main.o -g -I. -l pthread
./Chan_Kurtis_HW3_main "Prompt> "
Prompt> 
```

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

```
student@student-VirtualBox:~/csc415-assignment3-simpleshell-kchanw$ make run < commands.txt
./Chan_Kurtis_HW3_main "Prompt> "
Prompt> exit
make: Nothing to be done for 'commands.txt'.
student@student-VirtualBox:~/csc415-assignment3-simpleshell-kchanw$ make run < commands.txt
./Chan_Kurtis_HW3_main "Prompt> "
Help
Chan_Kurtis_HW3_main Chan_Kurtis_HW3_main.o how README.md
Chan_Kurtis_HW3_main.c commands.txt Makefile
Prompt> Child 10425 , exited with 0
"Hello World"
Prompt> Child 10426 , exited with 0
total 64
drwxrwxr-x 3 student student 4096 Sep 23 20:04 .
drwxr-xr-x 21 student student 4096 Sep 23 15:43 ..
-rwxrwxr-x 1 student student 16160 Sep 23 20:04 Chan_Kurtis_HW3_main
-rw-rw-r-- 1 student student 2832 Sep 23 20:04 Chan_Kurtis_HW3_main.c
-rw-rw-r-- 1 student student 8824 Sep 23 20:04 Chan_Kurtis_HW3_main.o
-rw-rw-r-- 1 student student 42 Sep 22 02:57 commands.txt
drwxrwxr-x 8 student student 4096 Sep 22 02:57 .git
-rw-rw-r-- 1 student student 258 Sep 23 17:52 how
-rw-rw-r-- 1 student student 1864 Sep 22 20:07 Makefile
-rw-rw-r-- 1 student student 5058 Sep 22 02:57 README.md
Prompt> Child 10427 , exited with 0
PID TTY TIME CMD
10372 pts/0 00:00:00 bash
10422 pts/0 00:00:00 make
10423 pts/0 00:00:00 sh
10424 pts/0 00:00:00 Chan_Kurtis_HW3
10428 pts/0 00:00:00 ps
Prompt> Child 10428 , exited with 0
ls: cannot access 'foo': No such file or directory
Prompt> Child 10429 , exited with 512
Prompt> Child 10430 , exited with 139
Prompt>
student@student-VirtualBox:~/csc415-assignment3-simpleshell-kchanw$ 
```