Mos Kullathon
Github: mosguinz

ID: 921425216
CSC415 Operating Systems

Assignment 3 – Simple Shell with Pipes

Description:

The purpose of this assignment is to develop a program that simulates a shell environment, complete with the capability to execute commands and handle piping between commands.

Approach:

As mentioned in class, I will be approaching this assignment in two parts: first, dealing with emulating the simpler behavior of a shell i.e., printing the prompts, executing the command, and displaying the output on the terminal; the second part is to expand the functionality of the shell by supporting the pipe operator.

The shell part (without the pipes): the flow of the shell for each command should be as follows:

- First, display the prompt to the user e.g., “Prompt >.”
- Then, read in the user input.
  - Split the input on the spaces.
  - Place the argument in an array, with the last element containing “NULL.”
- If it’s “exit,” just exit the shell.
- Create a child process using `fork()`.
- Then, in the child process, pass the array to `execvp()` to execute the command.
- In the parent, wait for the process to finish and print the process ID.

Of course, the above needs to be repeated each time after a command is executed.

Since we will be dealing with user input for each command, we do not know ahead of time how many arguments we need to accommodate for. As such, we will need to dynamically allocate the array for the arguments.

Although we know ahead of time the maximum length (177 here), we *could* theoretically limit the number of arguments to be half the maximum length. However, we could potentially be allocating too much memory when not needed.

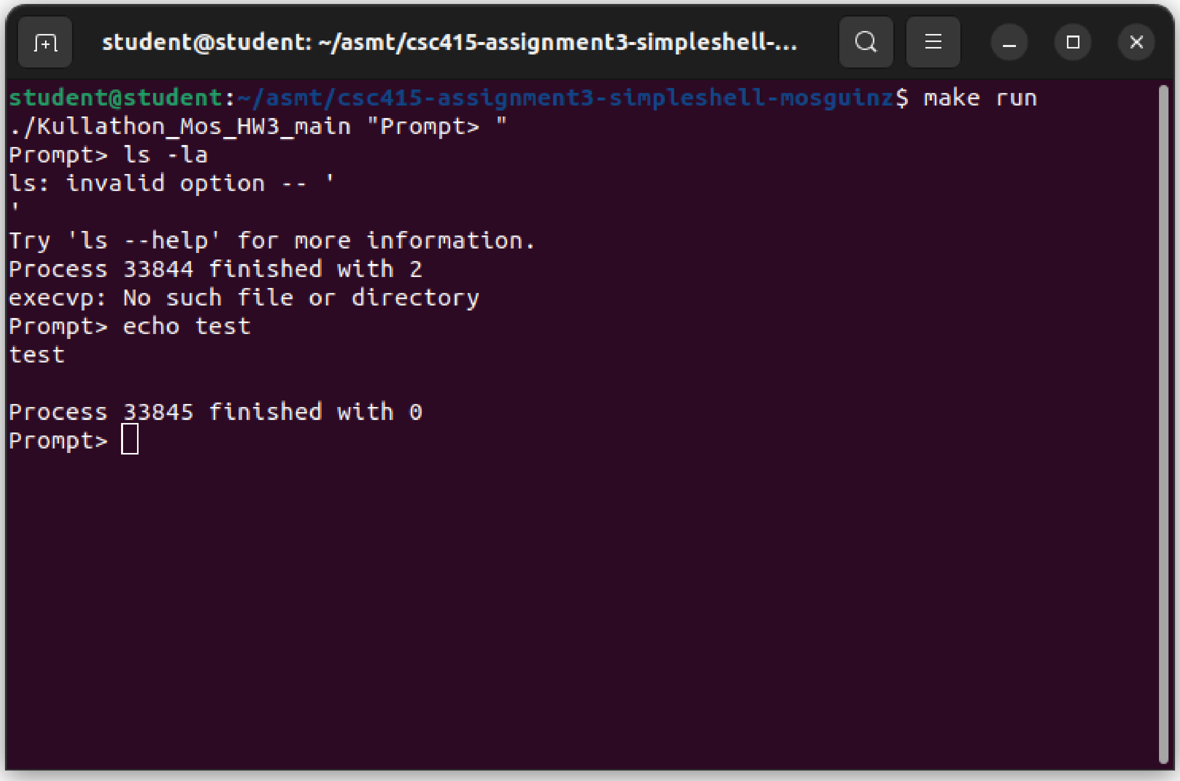
Now, the hard part. To accommodate for pipes, we would still need to perform the steps above to execute each of the commands. But in addition to tokenizing the commands, we would need to do it for the entire input, splitting on the pipes. If there are pipes, we would also need to:

- Create a pipe with two file descriptors.
- Then, create a child process using `fork()`.
- Then, in the child process, duplicate the write end of the pipe to the stdout.

- Then, execute the command using `execvp()`.
- In the parent, wait for the process to finish and print the process ID.
- Then, tokenize the command again to be passed to `execvp()`.
- Before executing the command, we again redirect the write end to stdin.
- And repeat for each pipes…

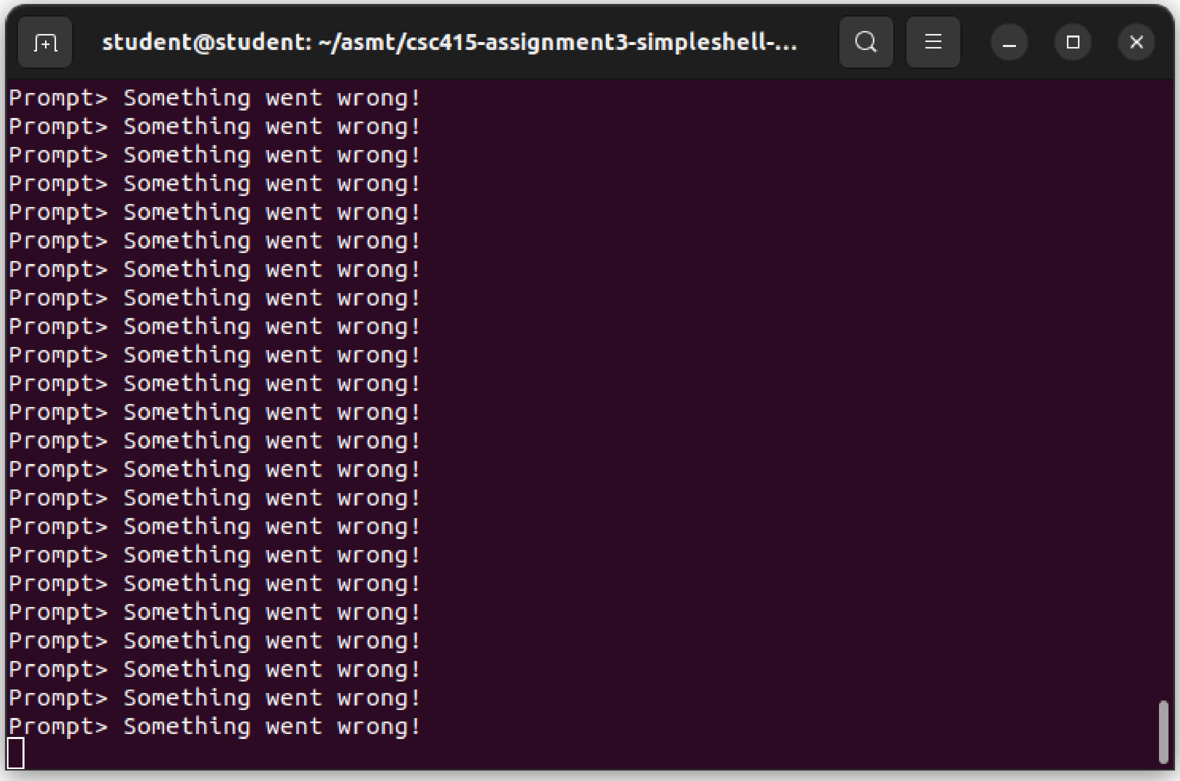
Issues and Resolutions:

First issue: when entering each command, the following prompt would get skipped due to the newline being read on the next prompt. I fixed this by removing the newline character at the end of each line when tokenizing the command.



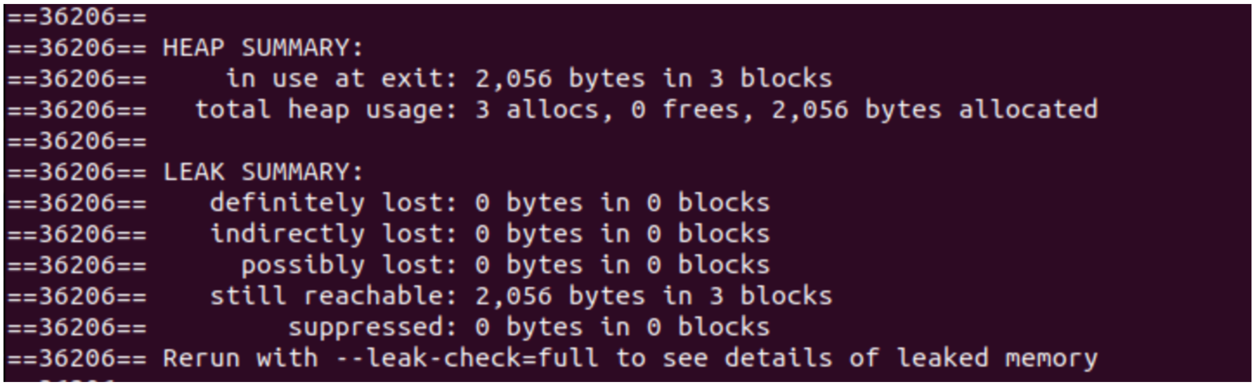
```
student@student:~/asmt/csc415-assignment3-simpleshell-mosguinz$ make run
./Kullathon_Mos_HW3_main "Prompt> "
Prompt> ls -la
ls: invalid option -- '
'
Try 'ls --help' for more information.
Process 33844 finished with 2
execvp: No such file or directory
Prompt> echo test
test

Process 33845 finished with 0
Prompt> 
```

Issue two: Terminal spamming when reading from file or hitting Ctrl+D. This was caused by not properly checking for the end of the file. I simply fixed this by checking the return value of `fgets` and exiting the shell.



```
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
Prompt> Something went wrong!
```

Issue three: while this didn't cause the program to crash, when inspecting the program with `make vrun`, there were multiple points where memory weren't being freed when the command would fail. Evidently, I did not free the array that held the tokens when the `execvp` would fail. I fixed this by painstakingly combing through the code and finding places that I needed to invoke `free`.



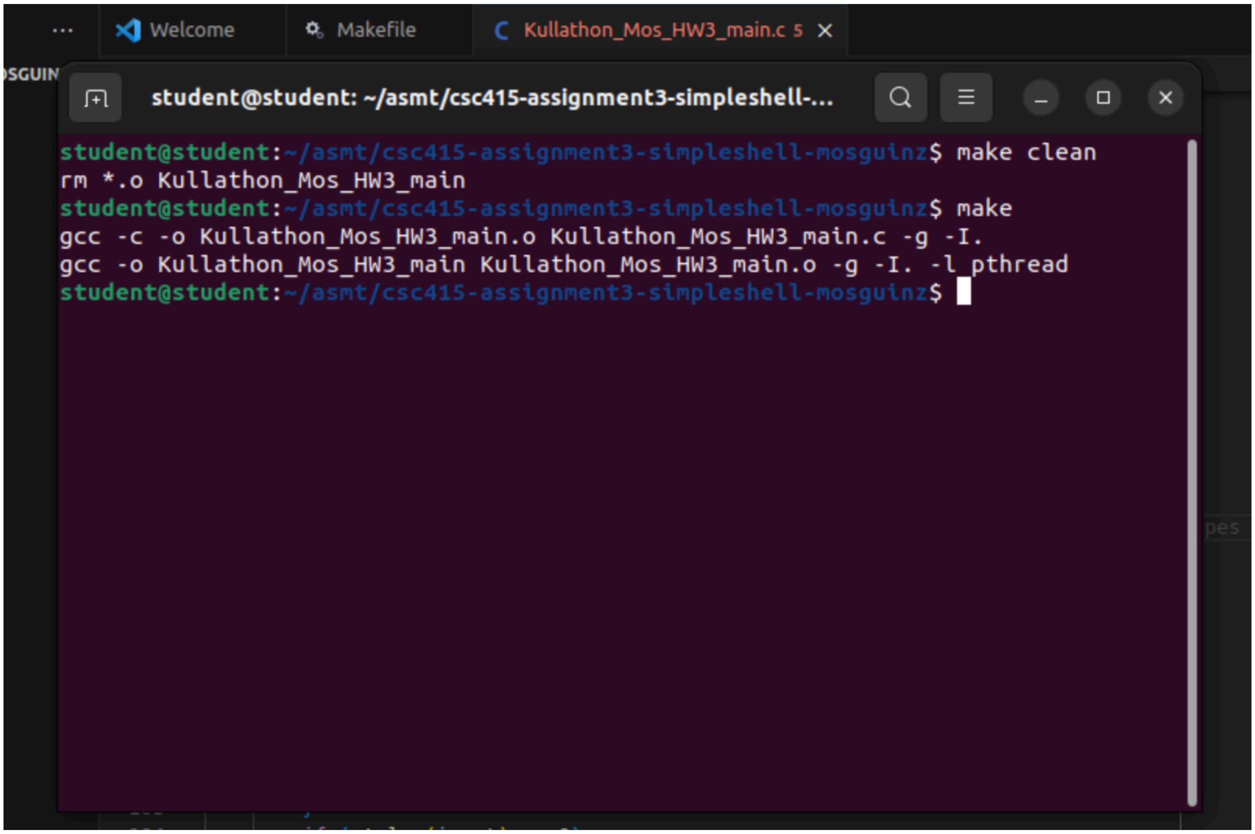
```
==36206==
==36206== HEAP SUMMARY:
==36206==     in use at exit: 2,056 bytes in 3 blocks
==36206==   total heap usage: 3 allocs, 0 frees, 2,056 bytes allocated
==36206==
==36206== LEAK SUMMARY:
==36206==    definitely lost: 0 bytes in 0 blocks
==36206==    indirectly lost: 0 bytes in 0 blocks
==36206==      possibly lost: 0 bytes in 0 blocks
==36206==    still reachable: 2,056 bytes in 3 blocks
==36206==         suppressed: 0 bytes in 0 blocks
==36206== Rerun with --leak-check=full to see details of leaked memory
```

Issue four: invoking commands with more than one pipe would not work. I didn't quite fix this issue as I ran out of time. Piping more than three commands would not work. Initially, including more than two pipes will crash the programs. I *partially* fixed the problem by ensuring that the program would not crash by closing unused pipes and ensuring that the next commands are being read.

Analysis:

N/A

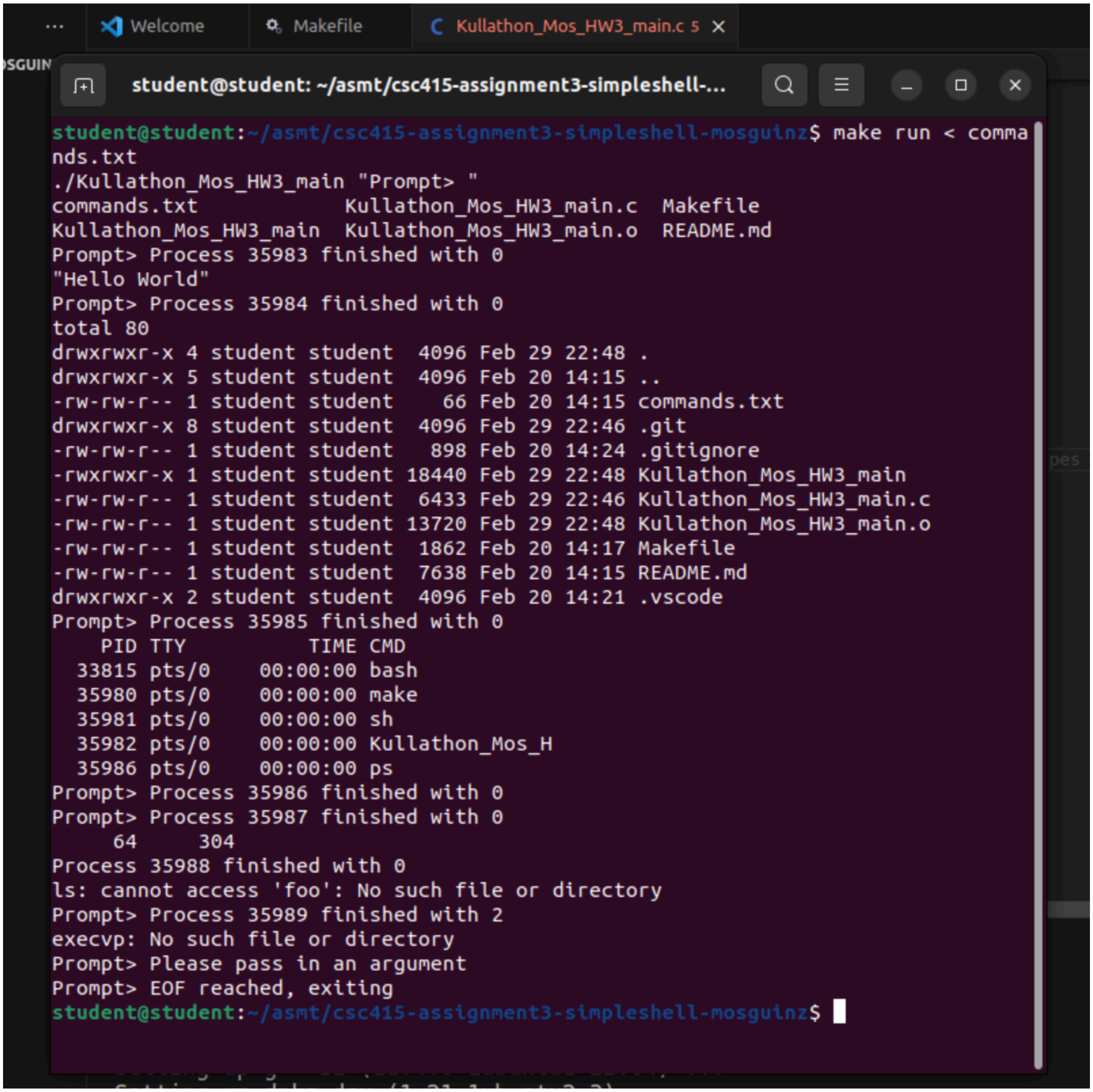
Screen shot of compilation:


Welcome
Makefile
Kullathon_Mos_HW3_main.c 5
student@student: ~/asmt/csc415-assignment3-simpleshell-...

```
student@student:~/asmt/csc415-assignment3-simpleshell-mosguinz$ make clean
rm *.o Kullathon_Mos_HW3_main
student@student:~/asmt/csc415-assignment3-simpleshell-mosguinz$ make
gcc -c -o Kullathon_Mos_HW3_main.o Kullathon_Mos_HW3_main.c -g -I.
gcc -o Kullathon_Mos_HW3_main Kullathon_Mos_HW3_main.o -g -I. -l pthread
student@student:~/asmt/csc415-assignment3-simpleshell-mosguinz$ 
```

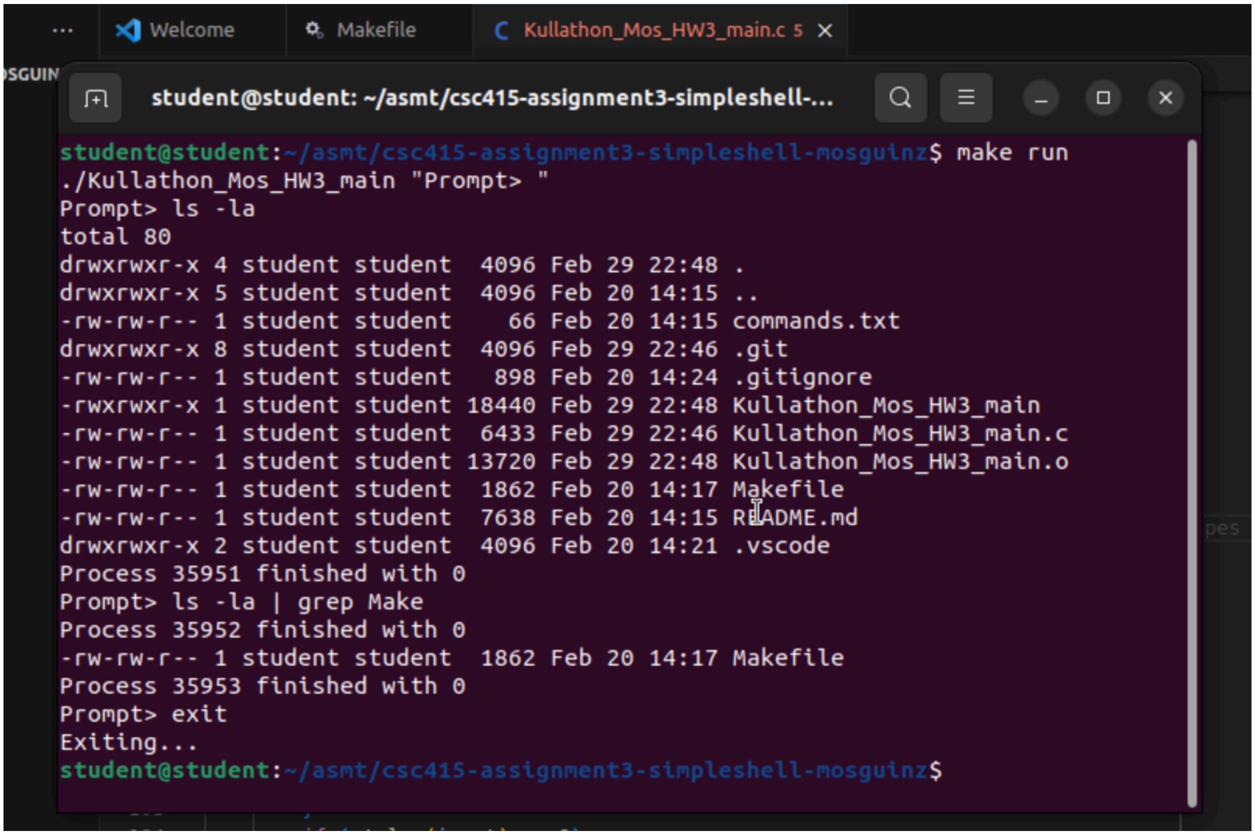
Screen shot(s) of the execution of the program:

Using the commands.txt file:


Welcome
Makefile
Kullathon_Mos_HW3_main.c 5
student@student: ~/asmt/csc415-assignment3-simpleshell-...

```
student@student:~/asmt/csc415-assignment3-simpleshell-mosguinz$ make run < comma
nds.txt
./Kullathon_Mos_HW3_main "Prompt> "
commands.txt            Kullathon_Mos_HW3_main.c  Makefile
Kullathon_Mos_HW3_main  Kullathon_Mos_HW3_main.o  README.md
Prompt> Process 35983 finished with 0
"Hello World"
Prompt> Process 35984 finished with 0
total 80
drwxrwxr-x 4 student student  4096 Feb 29 22:48 .
drwxrwxr-x 5 student student  4096 Feb 20 14:15 ..
-rw-rw-r-- 1 student student    66 Feb 20 14:15 commands.txt
drwxrwxr-x 8 student student  4096 Feb 29 22:46 .git
-rw-rw-r-- 1 student student   898 Feb 20 14:24 .gitignore
-rwxrwxr-x 1 student student 18440 Feb 29 22:48 Kullathon_Mos_HW3_main
-rw-rw-r-- 1 student student  6433 Feb 29 22:46 Kullathon_Mos_HW3_main.c
-rw-rw-r-- 1 student student 13720 Feb 29 22:48 Kullathon_Mos_HW3_main.o
-rw-rw-r-- 1 student student  1862 Feb 20 14:17 Makefile
-rw-rw-r-- 1 student student  7638 Feb 20 14:15 README.md
drwxrwxr-x 2 student student  4096 Feb 20 14:21 .vscode
Prompt> Process 35985 finished with 0
    PID TTY          TIME CMD
  33815 pts/0    00:00:00 bash
  35980 pts/0    00:00:00 make
  35981 pts/0    00:00:00 sh
  35982 pts/0    00:00:00 Kullathon_Mos_H
  35986 pts/0    00:00:00 ps
Prompt> Process 35986 finished with 0
Prompt> Process 35987 finished with 0
     64     304
Process 35988 finished with 0
ls: cannot access 'foo': No such file or directory
Prompt> Process 35989 finished with 2
execvp: No such file or directory
Prompt> Please pass in an argument
Prompt> EOF reached, exiting
student@student:~/asmt/csc415-assignment3-simpleshell-mosguinz$
```

On a typical run:


Welcome
Makefile
Kullathon_Mos_HW3_main.c 5
student@student: ~/asmt/csc415-assignment3-simpleshell-...

```
student@student:~/asmt/csc415-assignment3-simpleshell-mosguinz$ make run
./Kullathon_Mos_HW3_main "Prompt> "
Prompt> ls -la
total 80
drwxrwxr-x 4 student student  4096 Feb 29 22:48 .
drwxrwxr-x 5 student student  4096 Feb 20 14:15 ..
-rw-rw-r-- 1 student student    66 Feb 20 14:15 commands.txt
drwxrwxr-x 8 student student  4096 Feb 29 22:46 .git
-rw-rw-r-- 1 student student   898 Feb 20 14:24 .gitignore
-rwxrwxr-x 1 student student 18440 Feb 29 22:48 Kullathon_Mos_HW3_main
-rw-rw-r-- 1 student student  6433 Feb 29 22:46 Kullathon_Mos_HW3_main.c
-rw-rw-r-- 1 student student 13720 Feb 29 22:48 Kullathon_Mos_HW3_main.o
-rw-rw-r-- 1 student student  1862 Feb 20 14:17 Makefile
-rw-rw-r-- 1 student student  7638 Feb 20 14:15 README.md
drwxrwxr-x 2 student student  4096 Feb 20 14:21 .vscode
Process 35951 finished with 0
Prompt> ls -la | grep Make
Process 35952 finished with 0
-rw-rw-r-- 1 student student  1862 Feb 20 14:17 Makefile
Process 35953 finished with 0
Prompt> exit
Exiting...
student@student:~/asmt/csc415-assignment3-simpleshell-mosguinz$
```