Cassey Anene (@02879027)

Elijah Prentice (@02874763)

Dr. Legand Burge

12 October 2020

Simple Shell Project Report

For this project, we completed each part of the project task by task. We used the pair programming method in which we got many Google Meet calls and did the project together. Elijah acted as the driver by sharing his screen, coding, and implementing solutions while Cassey acted as the observer by assisting in coding, looking up resources to use, and implementing test cases.

Firstly, we started on task 1 by adding built-in commands. We started on the most simple command which was exiting the shell. This only took a few minutes to achieve and did not require any extra research. After that, we decided to work on printing and changing the directory. This required some extra work, so we looked up some documentation on changing directories. Next, we had to implement the **echo** command. For this command. We did not have to look up anything extra. We realized that we had to loop through each argument to print out each word in the message after the echo command. Lastly, we had to implement the environment commands. We had never used these commands in our careers before so before we worked on them we decided to see how they worked in our personal terminals. Cassey realized that the **setenv** command is not available on her terminal so she had to use a different command, **export**. Doing this helped us understand a way to implement these commands in our code.

In the second task, we had to add processes and the other commands such as **ls** and **cp**. We first tried to hardcode the functions of these commands but then moved on to using the **execvp** function because it was easier to use. We had to fork a child of the process because running **execvp** exited the program when it was completed.

In the third task, we had to add background processes. For adding a background process we first had to find if the desired process was the executable. If the program was an executable, then the function that we created would recognize the **&** symbol, run the program and then place the program in the background. Implementing this part, was a little tough because it was hard to find resources on learning how to run background processes from our shell.

In the fourth task, we had to do signal handling. This was the hardest task of the project to implement. Many of the solutions that we attempted to implement into our program to exit the foreground process would cause our shell to exit or would cause a segmentation fault. We found an article on signal handling in a user-made shell. We were on the right track with our solutions but the article helped us realize that we had to dig deeper into the concept of signal handling beyond what we have practiced on our other assignments.

In the fifth and final task, we had to kill off long-running processes. We knew how we wanted to kill off the process but we had to figure out where we needed to put the lines of code so that it could work correctly. We did a lot of basic testing of just putting the code in place we saw fit and then it all came together and solved the problem.

In conclusion, we were successful in implementing our own simple shell in Codio and working on this project has helped us gain a better understanding of the concepts we have learned up until this point in our Operating Systems course.

Code Documentation

- https://stackoverflow.com/a/298518
    - This source was used to get the current directory in the terminal.
- https://stackoverflow.com/a/11043336
    - Using the strcpy function to copy strings.
- https://stackoverflow.com/questions/28502305/writing-a-simple-shell-in-c-using-fork-execvp
    - This source was used to learn how to compare the command that we got from the argument to the ones that we have functions for.
- https://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm
    - Helped gain a better understanding of the strtok function.
- https://stackoverflow.com/a/13098645
    - Allowed us to identify an executable file to make a background process.
- https://indradhanush.github.io/blog/writing-a-unix-shell-part-3/
    - This source helped us understand more about signal handling and use it for our specific implementation.
- https://www.geeksforgeeks.org/chdir-in-c-language-with-examples/
    - This source was used to implement the change directory command.
- https://stackoverflow.com/a/12059006
    - This source was used to implement the **env** command. It helped us realize how the already created **environ variable was going to be used.
- https://stackoverflow.com/a/35899248
    - This source was used to help us with killing the process after 10 seconds.

- **https://stackoverflow.com/a/8821378**
    - Helped with concatenating strings with the strcpy function.
- **https://stackoverflow.com/a/26817195**
    - Running an executable from the main program