Chimzim Ogbondah

Professor Mat

Cpr E 308

8 March 2021

Shell project Summary

Through the course of this project, I was able to learn about different Linux system. I got a better understanding of how to change the current working directory using the environment variable “HOME” and relative paths. I also got to utilize process and parent process ID which can be seen through the build in commands and the non-built-in commands. The non-built-in commands utilize how a child process pid (process ID) and the parent pid return different values. Based on that I was able to have the parents wait for the child process to finish or have it run in the background by not waiting. I also got a better understanding of fork(), execvp(), and wait(). While calling fork() I was really able to test and see how two processes are created and then using execvp() to change the processes to any of the commands found in bin by giving the command name and then creating an array for the different arguments with the command. I also understood wait(&status) a lot more because that waits for the child process to finish running or be killed before allowing the parent process to run. Pwd() was also a command I wasn’t to familiar with but got to have a better understanding of how you get the working directory. Perror() is very useful in error handling since it prints out and error message. This saved me time as I didn’t have to printf() for all my error handling. I also got to see how strtok() is used for parsing and input based on a given delimiter. It was tricky at first with getting it to work how I wanted but through trial and error I was able to parse the users input to my liking. Finally, I got to really understand how inputs when starting a program for the command line work. This was something we used in SE 185 but I wasn’t really sure how it worked. Now I understand that the int argc holds the number of arguments while argv holds the input from the command line.

In conclusion I was able to learn about different Linux system calls and how to implement them inside a C program. I also got a better understand of the creation of processes and how to change a process to something else. Finally, I learned about how to run background processes through waiting for the child or not and how to handle different errors that might arise with the user entering a command. (although I couldn’t get my handling of the background processes to work with my shell project)