Eric Hsueh
1384955

Assignment 1: a1shell

Objective:

The objective of the project in my eyes is twofold:

1- understand the relationship between parent, child, fork(), and how processes run both in parallel, or wait for each other to finish.

2- understand some of the inner workings of the command line, how linux itself implements some of the command line.

This project is very valuable because it gives us insight into the inner workings of the command line, as well as teaches us the basics of the parent child relationships, and how processes are formed, and their states.

Design Overview:

So my project worked something like this:
Step 1: Set the cpu limit
Step 2: Fork the child process
Step 3: use an if else statement to differentiate between the parent process and child process. Since the return of the pid indicates whether or not the process is the child, the first if else statement separates the two paths.

If it was the child, it ran everything it needed to as a1monitor:
It would grab the current date/time and read from /proc/loadavg/ in order to give the output specified.
If it was the parent, it would start the loop of the a1shell process, where it waits for the user to input commands.

In the a1shell process, it has if else statements to differentiate between functions
The first function is the implementation of “done”, it sends a signal to the child and kills the child, and then it exits the parent process.
The second function implemented is the pwd command, it uses the getcwd() function in order to print out the current directory
The third function implemented is the umask command, it gets the octal values of the constants IRWXU, IRWXG, IRWXO and prints it out to the terminal through the function umask()
The fourth implemented function is the cd “pathname” function, checks that the first 2 characters are “c” and “d” respectively and then takes the path name and imputs it to the chdir() function. However since we would have to check for expansion, There’s also a check to see if the first variable typed is a $, if it is, it passes it to getenv to get the true path name, afterwards, it passes it to chdir() in order to switch to the path.
The last function is the else statement. If none of the above catches the buffer, then it calls the timer, tries its best to pass it to our linux bash, and then sees if it executes. If it does, then it returns the time values as and output and the result. The parent also waits for this child process to complete before moving on.
Those are the 5 major functions implemented for the parent.

Project Status:

I did encounter a lot of difficulties at the start since, I did 201 3 years ago and haven’t touched any C code since then. There was a lot of syntax errors and a lot that I didn’t quite understand when I started. There was a lot of functions I had to search up again (such as atoi) and just general knowledge stuff that I should know. However, after getting the skeleton code down, everything went a lot more smoothly from there on since the a1.pdf is very detailed and pointed us in the right direction to read in the textbook.

Testing and Results:

I tested my implementation after every single function. Once a function was complete, I tested it to its fullest range, trying to make it fail, and trying to make it succeed. Once I was happy with it, I moved on to the next function and repeated the process. After I was

finished the entire project, I ran through every function and made sure I haven't broken anything else in the process.

Acknowledgements:

For this project, I had taken some code from stack overflow and modified, the link is included inside the code.
Also I have taken the very skeleton code of my project from the text book figure 1.7 as it got me started and is the skeleton for my entire project