Ian Altoveros

CS433 Operating Systems

Professor Zhang

October 16th, 2020

Programming Assignment 2

I attempted to recreate the program in the book titled, “Unix Shell”. This program is designed as

a mock interface of a shell that will take in user commands and afterwards will execute the

command as inputted. When the program is first ran, it will mock what a Unix shell interface

will show. The first line will be outputted osh >, which is waiting for the command from the user.

Here the user can input any Unix Shell commands, for example, If I input the command “ls”, it

will list the current items in the folder since the ls command is designed to do so. Another

command the user can input is ps -ael, which will display the current list of processes in the

system that’s running. Lastly, another command you can use in Unix would be more <FILE

NAME HERE>, which basically shows the first part of the file you listed.

Now getting into the code, there were several steps in accomplishing this program. In the main

function we must designate a command array which holds an array of characters, and we use that

array to check what command is being used when the user inputs something. After you have the

user input a command it will take the input and call a fork() which creates a new process

called a child process that will concurrently run alongside the parent process. But if there are no

processes being currently ran then the child will create one by calling the function execvp(), which

processes the command that was inputted. There are many more commands that the execvp() can

process, the unix shell is very versatile in what it can possibly do.