Adam Sunderman

Section B

Project 1

In this project my task was to design a shell in which a user can run a few build in operations as well as commands utilizing the execvp() function. The two major obstacles of designing the shell were parsing user input accurately and managing child processes efficiently (including printing the child information at the right time). To parse user input I decided to look for the built in commands first. For the ones that didn’t require any arguments I simply checked if the input was exactly equal to the built in command. For built in commands with arguments I did a little extra parsing within the ‘if’ statement. If the input could not be recognized my code will simply attempt to parse it as a command, since incorrect input will cause execvp() to fail anyways. For command parsing I used spaces as delimeters and filled an args[] array for execvp() input.

For handling child processes I used a method from the previous lab. I created an integer value named child and set it equal to the return value of fork(). That way if child was equal to 0 I knew the child process was running. This allowed me to make sure the correct process was executing the correct code. Overall, this lab was very good experience in implementing the several types of instructions we learned in the last lab. It was a very practical application of these concepts resulting in a deep learning experience.