Sean Hinchee

Section G

Project 1

2/8/18

Throughout the project I learned a great deal about how to write a shell. I have never written a shell before, beyond simply interpreting commands. Although I'd consider myself fairly familiar with system calls and process management, I had never dealt with these concepts in such a tightly coupled environment as a shell. I got a great deal of practice in dealing with string splitting and various string processing formats. Furthermore, I had never implemented job tracking and as a result had to implement a list. I found the implementation of an idiomatic list a decent task and challenged myself to make the code as legible and generic as possible.

Adding the syntactical sugar of backgrounding a process and redirecting input to a file was a great learning experience. To accomplish both, I needed to implement a form of operator precedence. That is, & is processed before >. Backgrounding was implemented by searching the last character in an input string and setting a variable, *bg*, if an ampersand is present. The ampersand and the implied space preceding it are thus removed by placing a null character where separating space was expected. Output redirection was handled similarly, with a preceding space to the > character being expected and set to null after a redirection variable was set. Furthermore, in the case of redirection, built in command redirection was permitted as the entire shell's standard output was redirected to the given file (via the dup2 syscall) and reset upon the next shell master loop iteration. To compensate for the mass redirection, the standard error output descriptor was utilized for printing shell messages (other than the prompt) and as such, they would not be redirected when commands are executed.