Ricky Zhang
109311456
Project Instructions

Compile code using gcc projectComplete.c -o project. Execute the program with ./project. The user will be prompted to enter a command. Input one command and at most one pipe. Commands that were given to us to use were fork, waitpid, execvp, exit, pipe, dup, open, close; and were all used in the program. Several global variables needed to work across the multiple methods in the program. Character buffer takes in the input, and the array of character arguments are stored. If the input was cd, it would either chdir to root (“/”), or the directory of the argument. A new process is now created. If any string in the argument array has >, < in it and the argument on the right is not empty, it will create a new file description based on that argument with the access mode given as the second parameter. If pipe was the argument, it will do check() method which will split the strings of the buffer into two different argument arrays, then execute if child or parent. If input is “&”, then program will create a new process group id, redirect input output to “/dev/null” and execute the background process and exit. Argumentcounter gets incremented to inspect the next argument in the array to check for an operation. If whole array has been searched through, then execute the arguments on the left, no operations. If the process is not a child, then wait for the child process to finish.