Description:

This homework demands us to implement a bash terminal such that it executes a pipeline with man and grep commands. To do that, we need to use forks to create new processes.

Command:

The command that I choose is man ls | grep –e -a.

LS command: ls is used to list files. Ls command by itself gives the files of the current directory.

-e option: enables grep to execute special character hyphen “-”.

-a option: enables ls to show hidden files and the files that start with ".".

Almost all the time, I need to check hidden files such as ".git"; therefore, I choose this command.

Program Hierarchy:

In my program, I created a pipe in order to communicate between forks. Then, I create my first fork, which is "MAN". And the output of this fork is connected with the write-end of the pipe. The result of this fork is a manual file of the selected command (which is “ls”), and it is stored in the pipe. Second fork will wait until the execution of the first fork is completed. Then, I create the second fork for the "GREP" command. This fork will take read end of the pipe as an input. It writes the results in a file named "output.txt" instead of the console. The main process will wait for the execution of the second fork. Then it will give the necessary outputs to the console and terminate.

Choice of Variables:

In both of the forks, I created a char array called myargs in order to execute execvp function. Apart from that, I created a pipe called fd in order to communicate between forks and main program.