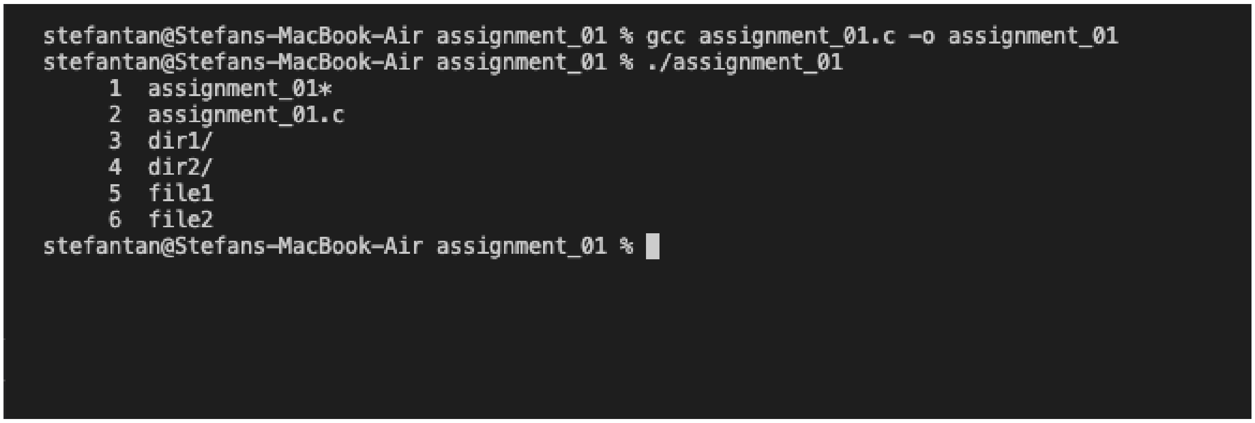
Stefan Tan

CSC 332 Section H

Professor Zaid Al-Mashhadani

Assignment #1 Report

Screenshot of the Output:



```
stefantan@Stefans-MacBook-Air assignment_01 % gcc assignment_01.c -o assignment_01
stefantan@Stefans-MacBook-Air assignment_01 % ./assignment_01
     1  assignment_01*
     2  assignment_01.c
     3  dir1/
     4  dir2/
     5  file1
     6  file2
stefantan@Stefans-MacBook-Air assignment_01 % 
```

Questions:

1. What form of exec() did you use? Why?

   The form of exec() that I used was the execlp() function. I decided to use the execl* function because the arguments that will be passed are already known. The arguments will be “ls -F” and “nl”, and they won’t be changed so there is no need to construct a vector of arguments for execv* functions. As the commands “ls -F” and “nl” both are available in the PATH, I chose to use execlp() as it uses the PATH environment variable. This makes it unnecessary to type out the full path of the command.

2. How many times did you use fork? Why?

   I used fork() 2 times. As the assignment requires us to create one parent process alongside two child processes, I just needed to ensure that only the parent gets forked twice so that the parent produces two child processes. To do that, I had to use the fork() function twice in the parent process' code.

3. How many pipes did this assignment require? Why?

   This assignment requires only one pipe. This is due to the fact that the pipe is only needed to communicate between the two child processes. One of the child processes will write to one end of the pipe while the other child process will read from the other end of the pipe. Thus, only one pipe is required.

4. What form of wait() did you use? How many times?

   The form of wait() that I used was waitpid() as I am able to specify both of the child processes that the parent process will wait for. As there are two child processes that the parent process has to wait for, I used waitpid() 2 times, one waitpid() for each child process.