Operating Systems **Assignment 1** **Zaid Al-Mashhadani**

One of the philosophies behind Unix is the motto do one thing and do it well. In this spirit, each shell command usually has a very specific purpose. More complicated commands can be constructed by combining simpler commands in a pipeline such that the output of one command becomes the input to another command. The standard shell syntax for pipelines is to list multiple commands, separated by vertical bars `|` (the pipe character).

Your task is to create a system with one parent process and two child processes where the children communicate using a pipe.

In the below example the output from the `ls -F` command is piped to input of the `nl` command.

Use fork(), one form of exec() functions, so that the first child will perform `ls -F` and pass the output to the second child using one direction pipe, so the second one can perform `nl` on the list of current directory contents. Later the second child process will print to the screen the result (see example below). The parent process must wait for its both children.

```
1       file1*
2       file2*
3       dir1/
4       dir2/
```

Submit the following:

I. Your complete code with all comments
II. Screenshot of the output (suggesting to create few files and directories within the assignment local directory)
III. Assignment report which will answer the following questions:
   1. What form of exec() function you used? Why?
   2. How many times you used fork? Why?
   3. How many pipes this assignment required? Why?
   4. What form of wait() you used? How many times?