Operating Systems - Lab Assignment 1

Deadline: Nov. 13, 2022

Part 1

The purpose of this part is to work with `fork`, `exec`, and `wait` to create new processes and use `pipe` to communicate between parent/child processes. You should implement a code to illustrate the following command: `ls / | wc -l`. This command prints out the number of files in the root path: `ls /` shows the files/directories in the root path, and its output will be piped through `|` to `wc -l`, which counts the number of lines.

Hint1: Use `fork` to make a child process. Then, the child process executes `ls /`, passing the result (i.e., the list files/directories) through a pipe to the parent process. The parent executes `wc -l` to print out the number of lines for the list passed by the child.

Hint2: You can use `dup2` to redirect the output of the `exec` to the input descriptor made by `pipe`.

Part 2

In this part, you will work with message queues. You need to implement two processes, such that the first process reads the content of a text file and passes it to the second process through a message queue. Upon receipt of the file content, the second process should count and print out the number of words in the file.

How to submit the assignment

You should zip all your source codes and upload them to Canvas. Please use the following format to name your file: `os_assignment<num>_<group_name>.zip`
For example, if your group name number is group5, you should submit:
`os_assignment1_group5.zip`