

CSc 332 - Operating Systems

Lab – Spring 2019

Instructor: Prajwal Khatiwada, email: prajwal.ccnny@gmail.com

Task 4 - System Calls Summary

March 8, 2019

Max Points: 30 **Due:** March 22, 2019 11:59 PM

(**Max Points:** 20 **Due:** March 29, 2019 11:59 PM)

PART 1 Simple Command Interpreter

Recall: In Task 3, we worked with `exec()` system calls for specific commands such as `date`, and `ls`.

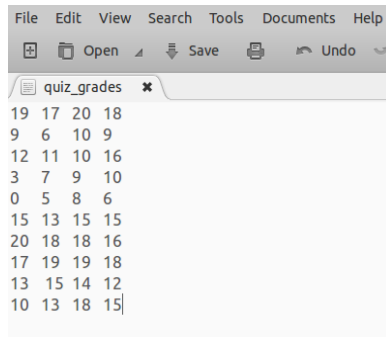
Write a special simple command interpreter that takes a command and its arguments. This interpreter is a program where the main process creates a child process to execute the command using `exec()` family functions. After executing the command, it asks for a new command input (i.e., parent wait for child). The interpreter program will get terminated when the user enters `quit`.

PART 2 Average Grade Calculator

There are 10 students enrolled in a course. The course covers x number of chapters from a textbook ($x > 1$). In each chapter y number of homework(s) are assigned ($y \geq 1$). The average grade for each homework in all the chapters need to be found out.

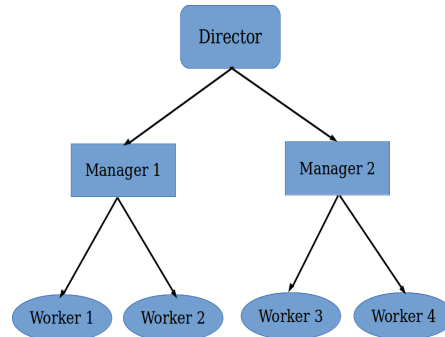
To solve this, write program which has the main process as *Director* process, which reads a file containing grades of all homeworks of all chapters and creates x number of *Manager* processes. Each *Manager* process will take care of solving a chapter. Each manager process will create y number of *Worker* process and pass one homework to each of them and they calculate and print the average.

The input file should contain the data according to the value of x and y . For example, the input text file and the process tree for $x = 2$ and $y = 2$ will look like the following:



```
File Edit View Search Tools Documents Help
Open Save Undo
quiz_grades
19 17 20 18
9 6 10 9
12 11 10 16
3 7 9 10
0 5 8 6
15 13 15 15
20 18 18 16
17 19 19 18
13 15 14 12
10 13 18 15
```

(a)



(b)

The Director process is responsible for opening and closing the input text file. It stores the values in a two dimensional integer array with 10 rows. You may need to use the following C functions (in addition to the necessary file & process management system calls): `fopen()`, `fscanf()`, `fseek()`, `fclose()`.

Submission Instructions

- Save your programs in a single folder and zip as: *task4_firstname_lastname.zip*. Make sure your programs compile and run without any errors.
- Email your code with subject line "Task 4 – CSc 332 (L) – firstname *lastname*"
