

# CSc 332 - Operating Systems

Lab – Spring 2021

## Task 4 - System Calls Summary

March 19, 2020

**Max Points:** 30 **Due:** April 08, 2021 11:59 PM

**(Max Points:** 20 **Due:** April 15, 2021 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`.

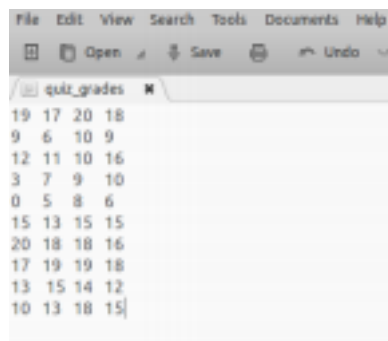
Write a special simple command interpreter that takes command and its arguments. This interpreter is a program where the main process creates a child process to execute the command using `exec()` family functions. After executing the command, it asks for a new command input (i.e., parent wait for child).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

### PART 2 Average Grade Calculator

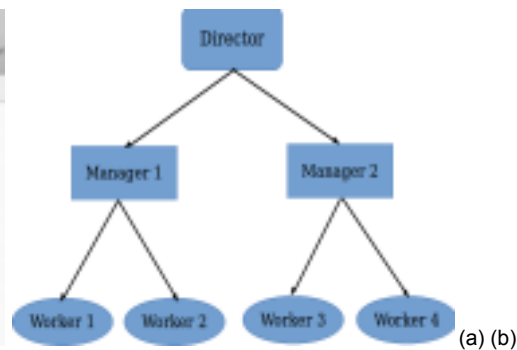
There are 10 students enrolled in a course. The course covers  $x$  number of chapters from a textbook ( $x > 1$ ). In each chapter  $y$  number of homework(s) are assigned ( $y \geq 1$ ). The average grade for each homework in all the chapters needs to be found out.

To solve this, write a program that has the main process as *Director* process, which reads a file containing grades of all homework 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:



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



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().

**Note:**

*There are a number of ways to implement this task, but only one way to start: you need to create a file like the illustration but containing a column for each homework for each chapter. Let's set a minimum of 2 homeworks per chapter and 5 chapters. That gives you a 2-d array of 10 columns and 10 rows. The rows represent each students' grade on a homework, and the columns represent homework assignments, where all you need to know is that there are 2 per chapter.*

*The process for creating workers is analogous. The Manager parent either gives the worker child the path of the file to open and an offset at which to read its values, or else passes the actual values to be averaged as a list of parameters.*

*It is essential that you provide me a written explanation of your process and how you implemented the programs. You also need to provide me with the input file with the original homework grades, and an output file with the averages. Ideally, you would also provide me with a log of execution so that we can see how much parallelism is achieved by dividing the work up among managers and workers.*

**Submission Instructions**

- *Upload to the folder with both the part 1 and part 2 files to the shared folder.*

\*\*\*\*\*