

CSC 33200 (L) - Operating Systems – Spring 2023

Lab 5: System Calls Summary

Date: 03/24/2023

PART 1 Simple Command Interpreter

DUE: 04/13/2023

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 (parent waits for child). The interpreter program will get terminated when the user enters **exit**.

Example:

./interpreter

command: pwd

⇒ output

command: ls -la

⇒ output

command: date

⇒ output

command: ls -lr /foldername

⇒ output

command: tail /etc/passwd | grep username

⇒ output

command: ps -u username | grep firefox

⇒ output

command: kill pid

⇒ output

command: exit

⇒ terminates the program.

Marks: 15

PART 2 Average Grade Calculator

DUE: 04/13/2023

There are **n** ($n > 1$) **students** enrolled in a course. The course covers **x** number of chapters from a textbook ($x > 1$).

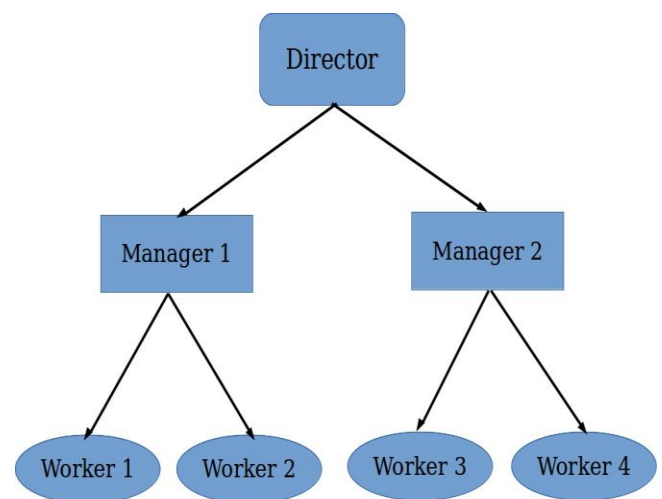
In each chapter **y** number of homeworks are assigned ($y \geq 1$). The average grade for each homework in all the chapters need to be found out.

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 marks of **n** students 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** and **n**. For example, the input text file and the process tree for $x = 2$ and $y = 2$ and $n = 10$ will look like the following:

	X1Y1	X1Y2	X2Y1	X2Y2
Student1	19	17	20	18
Student2	9	6	10	9
Student3	12	11	10	6
Student4	3	7	9	10
Student5	0	5	8	6
Student6	15	13	15	15
Student7	20	18	18	16
Student8	17	19	19	18
Student9	13	15	14	12
Student10	10	13	18	15
Output:	avg	avg	avg	avg

```
File Edit View Search Tools Documents Help
+ Open Save Undo
quiz_grades x
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
```



Finally, Worker process will write their avg in a shared memory and Director process will print out the avg of each homework from that shared memory.

For example, in the input above, we have 4 worker process- they will connect to a shared memory which is an array of size 4- containing the values of the avg for each homework. Once the avg is calculated, 4 worker process will terminate, then 2 managers will terminate and finally Director process will print out the avg for each homework and terminate. Your output should be in this format:

Avg of Homework1 : Avg1

Avg of Homework2 : Avg2

Avg of Homework3: Avg3

Avg of Homework4: Avg4

Note that, Input file will only contain the marks for each student for each homework in the quiz_grades.txt format.

Director process will need to read the file and determine the values of X, Y and N from the file and create appropriate number of Manager process. Manager process will then create appropriate number of Child process.

Please make sure you print the output in a proper order. And make sure your code can handle any input regarding the values of X, Y and N. The above example is for example only. Generalize your code to handle different values of X, Y and N.

Marks: 15

Submission Instructions

- All the programs MUST be clearly indented and internally documented
- Do not include executables in the zip file.
- Make sure your programs compile and run without any errors
- Save all your programs with meaningful names and zip into a single folder as: Lab5_[your last name here].zip (e.g., Lab5_Xyz.zip)
- Email your code with the subject line, "Lab5-CSC33200(L)–Class#G-*lastname*" (e.g., Lab5 - CSC33200(L)-Class #G-Xyz)
- Email: sdebnath@ccny.cuny.edu