National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Operating Systems**	**Code:**	**CL205**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**120 minutes**	**T. Marks:**	**40**
Date:	**Thursday 25-10-2018**	**Weight**	**30**
Section:	**A, B, C**	**Page(s):**	**2**
Exam:	**Lab midterm**		

Instructions/Notes:

- Discussion with other students is not allowed.
- Use of the internet, notes, codes, lab manuals, and flash drives is strictly prohibited.
- See **MAN page** for a fucntion's prototype, its header(s) and explanation.
- Plagiarism will result in **F** grade in lab.
- Submission path: Section-X (here X will be your section A or B or C)
  \\sandata\xeon\Fall 2018\Shakeel Zafar\MID OS\Section-X\Q1 or Q2
- Code must be **intended**, failure to comply will cause ZERO in marks.

Question #1: **(20 marks)**

Write a program that computes results of teacher's evaluations. For this purpose, you will receive at least 2 filenames from command line. First filename will be student.txt and second filename – the teachers file - will be tec1.txt. There can be an arbitrary number of teacher's filenames, N, named as tec1.txt, tec2.txt, tec3.txt, tec4.txt and so on. Assume every file exists.
Possible list of commands for running your program can be:

```
./mid_question student.txt tec1.txt
./mid_question student.txt tec1.txt tec2.tx
./mid_question student.txt tec1.txt tec2.tx tec3.tx tec4.txt tec5.txt
```

In student.txt you have students count in first line and in next lines you have student roll numbers and his/her cgpa separated by space.
In teacher's files you have information about teachers. These file names are tec1.txt, tec2.txt, tec3.txt, tec4.txt and so on. In these files at first line you have teacher's name and in next line you have count of reviews. Then in the remaining lines you have student's roll number along with their reviews (good, bad and excellent) space seprated. For example tec1.txt is:

tec1.txt

```
ABC
4
16L6890 good
16L1234 bad
15L4444 bad
14L5555 excellent
```

student.txt

```
5
14L5555 2.55
15L4444 3.55
16L6890 2.90
16L1234 3.10
16L8901 3.40
```

Assume that every roll no in teachers file exists in student.txt. Now your task is to write a program in which for every teacher you need to create a children process. Each process will read student.txt and tecX.txt (1≤X≤N). Nth teacher is processed by Nth child. For every teacher you need to compute average CGPA for good reviews, average CGPA for bad reviews and average CGPA for excellent reviews. For example, tec1.txt result will be:

Sample Output

```
Teacher name: ABC
Average cpga of students for good: 2.90
Average cpga of students for bad: 3.325
Average cpga of students for excellent: 2.55
```

Save result of every teacher in a shared memory. Parent process will wait for every child. When a child terminates. Parent will read data from shared memory and display the result of a teacher on screen and then go for next child.

<u>Question #2:</u> **(20marks)**

1. You have to write a function that copies the contents of files in the current working directory to (target) directory using thread(s). The name and extension of the original file and the copy file would be the same.

```
void * copyfile (void * arg);
```

2. The function receives argument which consists:
   a. The name of the file to copy
   b. Absolute path of the file's copy in the new(target) directory – the file to be created.
3. The thread/function returns the number of bytes written to the new file.
4. In case of a nonexistent target directory, create a child (fork) to execute "mkdir" with proper arguments in main (). So, the target directory will be created. Now, use this directory to as target.
5. A file "input.txt" contains the names of all the files in the current working directory. Hint: To generate input.txt:
   a. In the terminal, navigate to the working directory.
   b. Run "ls > input.txt". Executing "ls > input.txt" would create a file input.txt in the working directory and write the output of the ls command in input.txt instead of stdout.
6. You have to create as many threads as the number of files listed in input.txt to copy all the files. (Copy input.txt too.)
7. Your program has only one command line argument which is the absolute path of the new(target) directory.

Notes:
You are not allowed to use the "cp" command.
You can assume that the working directory contains no sub-directories.

Sample Output

```
~/Desktop/test$ ./a.out /home/oracle/Desktop/test2/
Creating Directory:/home/oracle/Desktop/test2/
Created file:/home/oracle/Desktop/test2/1input.txt and wrote 13 bytes
in total.
Created file:/home/oracle/Desktop/test2/ainput.txt and wrote 22 bytes
in total.
Created file:/home/oracle/Desktop/test2/a.out and wrote 13600 bytes in
total.
Created file:/home/oracle/Desktop/test2/code.c and wrote 1532 bytes in
total.
Created file:/home/oracle/Desktop/test2/code.cpp and wrote 2026 bytes
in total.
Created file:/home/oracle/Desktop/test2/input.txt and wrote 65 bytes in
total.
Created file:/home/oracle/Desktop/test2/output.txt and wrote 0 bytes in
total.
~/Desktop/test$ ls /home/oracle/Desktop/test2/
1input.txt  ainput.txt  a.out  code.c  code.cpp  input.txt  output.txt
```