



National University of Computer & Emerging Sciences, Karachi
Spring 2020 CS-Department



Final Examination

23th June 2020, 09:20 AM – 12:30 PM

| | |
|--|------------------------------------|
| Course Code: CL 220 | Course Name: Operating Systems Lab |
| Instructor Name / Names: Safia Baloch, Rahemeen Khan | |
| Student Roll No: | Section: |

Instructions:

- Don't return the question paper in the end.
- Read the question completely before answering it.

Time: 220 minutes.

Max Marks:

INSTRUCTIONS: (10 Minutes directory creation, 20 minutes submission): 30 minutes

When you're ready to start your final exam, do the following:

1. Create a **main** directory called your roll number and section "k122310_4A". In it, you should have two things: a subdirectory called "Questions", and a README.TXT file. The subdirectory "Questions" should, of course, contain the exam question solutions as Q1.c, Q2.c etc. Inside each Question file you must comment actual question text and each program must print your roll number first"
2. Run the command "ls" inside the main directory and take a screenshot of output, save the Screenshot inside the main directory. [Screenshot by default stores in pictures directory so you can find it there].
3. The README.TXT file should contain your name, student number, and nu email address, as well as instructions if you want to specify about your any programs before its execution.
4. Once you are done with your final exam now compress your exam folder (k122310_4A) [Compress option can be found by clicking right key on folder].
5. **Submit it on google classroom you were invited for finals. "GOOD LUCK"**

Instructions have some marks for following

1. Shell Programming (5%): (20 minutes)

Write a program which takes these strings as command line arguments, "I LOVE PAKISTAN". Your shell program converts all string letters small Except the first one, for Example, "I Love Pakistan".

Then convert all the strings into small letters (i love pakistan) and then store the output into a file named your_roll_num.txt, and then display the output from reading that file.

→ Take a Screenshot of output and paste it in the question directory. "Don't crop the picture"

2. Process (10%) (40 minutes)

Given sample code **Q2.c** in attachments to use accordingly for the following questions:

a. Write a C program which will fork a child process

- i. Parent Process will wait for the child process to complete.
- ii. Child process will execute a shell script by passing 2 arguments.
- iii. Shell Script would perform addition, subtraction, multiplication and division of that number.

→ Take a screenshot of output and paste it in the questions directory. "Don't crop the picture"

b. The program will fork four child process, each terminating in a different way: (Marks 5)

- i. The Wise Son: sleeps for 1 second and then exits.
- ii. The Simple Son: calls exec to display mac address and exits.
- iii. The Wicked Son: sleeps for seven seconds and executes date command after waking up.
- iv. The Son Who Doesn't Know How to Ask Questions: Prints the id of its grandparent.

→ Take a screenshot of output and paste it in the question directory. "Don't crop the picture"

3. Thread + Semaphore (15%): (40 Minutes)

Create a program of 'ArraySummation', using pthread maintaining synchronization(semaphore) such that each thread needs to spend 5 seconds in the critical section i.e. set an alarm for it and then the thread exits the critical section using signal. Take an array of 25 elements. Check the time difference with using different numbers of threads such as.

| Number of Threads | Two Threads | Three Threads | Four Threads |
|-------------------|-------------|---------------|--------------|
| Time Taken | | | |

You can write your answer in txt file with time readings.

→ Take a screenshot of the output and paste it in the question directory. "Don't crop the picture".

4. OpenMP(15%): (30 Minutes)

- a. You must have studied a linked list in a data structure course, Now it's time to convert them into a parallel one. You are given a program of linked lists in attachment **Q4a.c** which computes fibonacci sequence. Now use openMP for parallel traversing of linked lists.

You can make little changes according to your logic and also can introduce new data structures if needed, but can't change the whole program.

→ Take a screenshot of the output and paste it in the question directory. "Don't crop the picture"

- b. **This Question is open Book**, means you must write the answer in **your own words explaining why** parallel traversing of a linked list is difficult?

→ For answer: type your answer in a text file(keep it in the main directory) OR write your answer in a paper explaining the reason(Take a clear picture of it and past it in folder)

5. Remember your project (5%): (20 Minutes)

You are supposed to write at least 10 points about the sequence of your project workflow. Your answer must not match with your project team members. Only it can match the semantics of it.

Write your answer in a txt file named **your_roll_number_proj.txt**