National University of Computer and Emerging Sciences

Subject: Operating Systems **Section:** BSCS-5A & BSCS-5B

Instructions

1. This project is in a group of max **3** members.
2. Implementation of the project will be in C++.
3. You will have maximum marks if you have done the entire task.
4. Submission will not be accepted via email, Facebook or USB flash drive etc.
5. Zip all code files of your Project. It should be uploaded as an individual file in the following format. "RollNoOfGroupMembers_Section.zip".
6. Deadlines should be kept in mind. Because there will be no deadline extension.
7. In case of plagiarism, Zero marks will be awarded to all group members in the project and all assignments.
8. Only one group member will submit the project.

OS Simulator

Project Description

In this project, you have to develop an operating system simulator. Your simulator will be developed using concepts of operating systems which you have studied during the course. Simulator will perform following functionalities:

1. CPU scheduling:

Implement the following CPU Scheduling algorithms and choose parameters (i.e waiting time burst time etc) for CPU Scheduling accordingly.

- Multilevel Queue Scheduling
- Multilevel Feedback Queue Scheduling
- Longest Job First
- Highest Feedback Ratio Next Scheduling

Highest Feedback Ratio Next is one of the most optimal scheduling algorithms. This is a non-preemptive algorithm in which the scheduling is done on the basis of an extra parameter called Feedback Ratio. A Feedback Ratio is calculated for each of the available jobs and the Job with the highest feedback ratio is given priority over the others.

Feedback Ratio is calculated by the given formula:

1. Feedback Ratio = (W+S)/S

   Where: W is Waiting Time, S is Service Time or Burst Time

2. Semaphore

- Teacher Assistant Evaluation Problem

In this problem, You have to achieve synchronization. There will be 1 teacher assistant, N chairs for students who will wait for evaluation, 1 chair for a student who will evaluate his/her assignment.

- If there is no student then the teacher assistant will be busy with his work.
- If the student arrives, the teacher assistant will give an alert signal.
- If the teacher is busy in evaluation of student assignments then the other students have to wait for their turn.

3. Deadlock

- Bankers Algorithm:

Bankers algorithm used to avoid deadlock and allocate resources safely to each process in the computer system. The 'S-State' examines all possible tests or activities before deciding whether the allocation should be allowed to each process. It also helps the operating system to successfully share the resources between all the processes. The banker's algorithm is named because it checks whether a person should be sanctioned a loan amount or not to help the bank system safely simulate allocation resources.

Suppose the number of account holders in a particular bank is 'n', and the total money in a bank is 'T'. If an account holder applies for a loan; first, the bank subtracts the loan amount from full cash and then estimates the cash difference is greater than T to approve the loan amount. These steps are taken because if another person applies for a loan or withdraws some amount from the bank, it helps the bank manage and operate all things without any restriction in the functionality of the banking system.

Similarly, it works in an operating system. When a new process is created in a computer system, the process must provide all types of information to the operating system like upcoming processes, requests for their resources, counting them, and delays. Based on these criteria, the operating system decides which process sequence should be executed or waited so that no deadlock occurs in a system. Therefore, it is also known as deadlock avoidance algorithm or deadlock detection in the operating system.

When working with a banker's algorithm, it requests to know about three things:

- How much each process can request for each resource in the system.
- How much each process is currently holding each resource in a system.It represents the number of each resource currently available in the system.

Following are the important data structures terms applied in the banker's algorithm as follows:

The Banker's Algorithm is the combination of the safety algorithm and the resource request algorithm to control the processes and avoid deadlock in a system:

Safety Algorithm

It is a safety algorithm used to check whether or not a system is in a safe state or follows the safe sequence in a banker's algorithm:

Resource Request Algorithm

A resource request algorithm checks how a system will behave when a process makes each type of resource request in a system as a request matrix.

For better understanding check following links:

- Bankers Algorithm

4. Memory Management

In this section, you are going to implement following page table concepts:

- Hash page table
- Inverted page table

5. Page Replacement

- **Second Chance Algorithm**

In the Second Chance page replacement policy, the candidate pages for removal are considered in a round robin matter, and a page that has been accessed between consecutive considerations will not be replaced. The page replaced is the one that, when considered in a round robin matter, has not been accessed since its last consideration.
It can be implemented by adding a “second chance” bit to each memory frame-every time the frame is considered (due to a reference made to the page inside it), this bit is set to 1, which gives the page a second chance, as when we consider the candidate page for replacement, we replace the first one with this bit set to 0 (while zeroing out bits of the other pages we see in the process).

Thus, a page with the “second chance” bit set to 1 is never replaced during the first consideration and will only be replaced if all the other pages deserve a second chance too!

Using Second chance Algorithm calculate No. of Page faults, page fault probability and page fault percentage.

For better understanding check following links:

- Second Chance Algorithm

Interface of Project

The Interface of the simulator can be console based or Graphical user interface(GUI). Groups which will add GUI will be awarded with bonus marks. Your project will have a proper menu. In which user will select option and Simulator will perform functionality. For GUI, you can use any C++ library.

Hints and Tips

For CPU Scheduling, Don’t create a process using a fork. You have to create a class for processes which have attributes like process ID, Burst Time etc.
For semaphore and deadlock concepts, take input for the required number of threads and for creation of thread you have to use the pthread library.

Good Luck!!!!