National University of Computer and Emerging Sciences

Subject: Operating Systems Section: BSCS-4A & BSCS-4B

Instructions

1. This project is in a group of max 2 members.
2. Implementation of the project will be in C++ or 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. You have to submit a single .c or .cpp file ,do not zip your file.
6. Deadlines should be kept in mind. Because there will be no deadline extension.
7. Plagiarism will be checked by your TA with your own section, cross section,with last year submissions and also with online resources e;g chatgpt ,copilot, chegg, github etc.. In case of plagiarism, Zero marks will be awarded to all group members in the project and also in 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

You have to implement 3 queues; each queue will implement a different scheduling algorithm.

First queue will implement priority scheduling, 2nd will implement RR and third queue will be implemented using SRJF(shortest remaining job first).

- Multilevel Feedback Queue Scheduling

A process can move between the various queues; aging can be

implemented this way
Multilevel-feedback-queue scheduler defined by the following
parameters:
number of queues(3)
scheduling algorithms for each queue
method used to determine when to upgrade a process
method used to determine when to demote a process
method used to determine which queue a process will enter
when that process needs service

Similarly for MLFQ you will again implement 3 queues, each queue will implement a different scheduling algorithm.
First queue will implement FCFS, 2nd will implement SJF and third queue will be implemented using RR (shortest remaining job first).

- Longest Job First
- Lowest Feedback Ratio Next Scheduling

Lowest Feedback Ratio Next is one of the 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 lowest 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

Implement Dining Philosopher Problem.

Design and implement a solution for the dining philosopher's problem using semaphores. In this problem, a group of philosophers sit at a round table and alternate between thinking and eating. However, there are only a limited number of chopsticks available for the philosophers to use as utensils. To eat, a philosopher must pick up both chopsticks adjacent to them. If a philosopher cannot pick up both chopsticks, they must wait until both are available. Use semaphores to ensure that each philosopher is able to eat without creating a deadlock or starvation scenario.(for more information see the slides).

3. Deadlock

- Bankers Algorithm:

Banker’s 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.

4. Memory Management

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

- Two-Level Page-Table Scheme (see slides for more detail)
- Hashed Page Table with TLB

5. Page Replacement

- Implement Second Chance Algorithm using circular link list (for more information see slides).

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

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.

For Queries:

Contact: **Mr. Hamza Majeed,** Email: l202074@lhr.nu.edu.pk

Good Luck!!!!