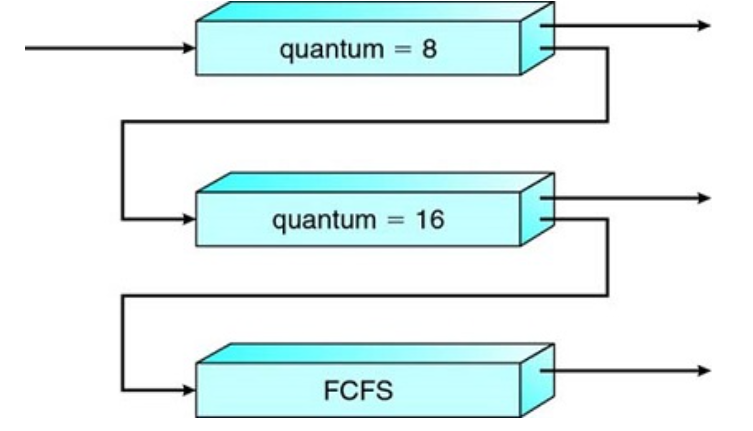
**Assignment #1**

**CPSC 351 Fall 2021**

**IPC and Multilevel feedback queue scheduler**

Implement a C++ program use the producer–consumer model with shared memory. The producer (example PPT 3.18) establishes a shared-memory object and writes to shared memory, and the consumer (example PPT 3.19) reads from shared memory. The data input is used by the consumer process for via multilevel feedback queue scheduler as described below.



An entering process is put in queue 0. A process in queue 0 is given a time quantum of 8 milliseconds. If it does not finish within this time, it is moved to the tail of queue 1. If queue 0 is empty, the process at the head of queue 1 is given a quantum of 16 milliseconds. If it does not complete, it is preempted and is put into queue 2. Processes in queue 2 are run on an FCFS basis but are run only when queues 0 and 1 are empty. For example, Q0 has 1 process, Q1 has 1 process, and Q2 has 0 process. Then, the process in Q0 should be executed, and then a process in Q1 is executed. Finally, Q2 will get CPU resource. Thus, all long processes automatically sink to queue 2 and are served in FCFS order with any CPU cycles left over from queues 0 and 1.

In general, a multilevel feedback queue scheduler is defined by the following parameters:

The number of queues

The scheduling algorithm for each queue

The method used to determine when to upgrade a process to a higher-priority queue

The method used to determine when to demote a process to a lower-priority queue

The method used to determine which queue a process will enter when that process needs service.

**Input example:**

P1 (ID), 0 (AT), 40 (BT), P2 (ID), 2 (AT), 10 (BT)

**Output example:**

P1 at Queue 0 it is executed for 8

P1 at Queue 1 it is executed for 16

P1 at Queue 2 it is executed for 16

P1 is interrupted 2 times, completes on queue #2, and TAT for P1 is 50

P2 at Queue 0 it is executed for 8

P2 at Queue 1 it is executed for 2

P1 at Queue 2 it is executed for 0

P2 is interrupted 1 time and completes on queue #, TAT for P2 is 34

**Submission guidelines:**

You must make sure that the code compiles and runs correctly. Copy and paste your code and include screenshots of output of your program. Fill out 3 tables: Team members, your roles and your evaluation. This assignment must be completed using C++ on Linux. Work in groups of 3-4. Place all your files under one PDF file with a unique name example - AssignmentWinningTeam.

**Grading guidelines:**

The program compiles: 10

Dynamic data/variables and data allocations/deallocations: 15

Producer: 15

Consumer:15

Correct and efficient/re-usable user-defined function: 35

Documentation (code comments, include testing screenshots and the project info files): 5

Makefile 5

Late submissions shall be penalized 1 point per hour until the assignment closing date as listed on Canvas.

**Academic Honesty:** All forms of cheating shall be treated with utmost seriousness. You may discuss the problems with other students in your group only; however, you must write your OWN codes and solutions. The project requirements are unique to your course, and it absolutely should not be posted online for the rest of the world to solve it. Copying an assignment from another student/group or allowing another student/group to copy your work may lead to an automatic F for this course. If you have any questions about whether an act of collaboration may be treated as academic dishonesty, please consult the instructor before you collaborate. Details posted at <http://www.fullerton.edu/senate/documents/PDF/300/UPS300-021.pdf>.

Ground Rules

(Meeting schedule/locations, attendance expectations, agenda, assignment completion, communication methods, etc.) You may modify the rules as needed and as you see necessary for your team.

▪ Participation and input are expected from all team members. All opinions will be considered and equally valued.

▪ Team members will notify the manager in advance if they are not going to be able to attend a scheduled meeting.

▪ Team members should check email and/or other communication channels at least once a day to stay on top of things.

▪ All team members will be held accountable for their portions of the projects and are expected to complete them in a timely manner and doing the best job they can.

▪ Notify team of emergencies that may result in not being able to meet deadlines or meetings. The rest of the team will do their best to pitch in on the team assignment.

▪ Assist fellow team members when they are in need.

▪ Maintain a positive atmosphere by respecting other members’ suggestions, using constructive criticism, and encouragement.

▪ No plagiarism. Every member must be responsible to avoid/prevent plagiarism.

**Team Members**

|  |
| --- |
| Name/ Student Email |
|  |
|  |
|  |
|  |

**Team Roles** (Define roles of each member for this project implementation)

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | Good example: Wrote all functionally for consumer process data receiving PID/AT/BT and parsing.  Bad example: did coding and testing |
|  | |  |
| ***How to score your own and peer performance.*** | Refer to the Grading guidelines listed in this document and apply the same point’s deductions (if any), otherwise apply a full score. If someone is not contributing to your group work, it’s a good idea to save a digital evidence (screenshots, logs). |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Team Members> | John Doe | Mary Poppins |  |  | **Total** | **Optional Comments on Your Evaluation on Team** |
|  |
| **Evaluators** |
| John Doe (fill out your row) | 100 | 100 |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Total** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **Average** |  |  |  |  |  |  |
| **Percent** |  |  |  |  |  |  |
| **Initials** |  |  |  |  |  |  |
| **Optional Comments on Your Score Earned from Team** |  |  |  |  |  |  |