# COMSATS University Islamabad, Lahore Campus

☐ **Sessional-1** ☐ **Sessional-II** ☑ **Terminal Examination FALL 2020**

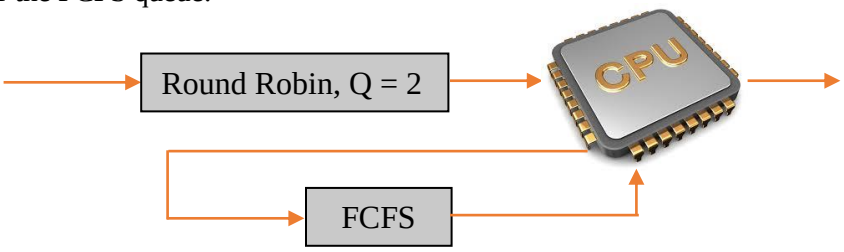
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Course Title: | Operating Systems | | | Course Code: | CSC322 | Credit Hours: | 3 |
| Course Instructor/s: | Dr. Hasan Jamal | | | Programme Name: | BS Computer Science | | |
| Semester: | | Batch: | | Section: | | Date: | 04/01/2021 |
| **Time Allowed:** | **3 Hour** | | | **Maximum Marks:** | | | **70** |
| Student's Name: | | | | Reg. No. | | | |

**Important Instructions / Guidelines:**

- Submit your solution on Google Classroom in PDF format. No other format will be accepted.
- Show all your work, as partial credits will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Please be neat.
- In case of late submission, one mark will be deducted for each minute over the submission deadline
- Any solution found to be copied would strictly result in zero marks**.**

**Question 1:** **[Marks: 10]**

Imagine we have a multi-level feedback queue with 2 queues. The highest priority queue is a ***RR*** scheduler with a quantum of 2. The second priority queue runs as a ***FCFS*** queue, as shown in figure below. Processes start in the ***RR*** queue and are demoted to the FCFS queue if they exceed their quantum. Processes in the ***RR*** queue are always prioritized over the FCFS queue.




| Process | Arrival Time | CPU Burst |
|---|---|---|
| 1 | 0 | 4 |
| 2 | 3 | 4 |
| 3 | 4 | 3 |
| 4 | 9 | 2 |
| 5 | 10 | 3 |

The table above shows a list of processes along with their arrival times and CPU burst times. Calculate the average ***waiting time*** by using the multi-level feedback queue and draw the ***Gantt chart***.
**Note:** A process can only be preempted if it exceeds its quantum.

**Question 2:** **[Marks: 10]**

Given below is the pseudocode to synchronize processes P1, P2, P3, P4, P5, and P6 by using three semaphores X, Y and Z that are initialized as follow: X=0, Y=0, Z= –2. The Operating System can schedule the processes which are ready to execute in any order. List down at least 5 possible orders of execution in which the processes can run.

| P1 | P2 | P3 | P4 | P5 | P6 |
|---|---|---|---|---|---|
| Wait(X)<br>Print P1<br>Signal(Y)<br>Signal(Y) | Wait(Y)<br>Print P2<br>Signal(Z) | Print P3<br>Signal(X) | Wait(Y)<br>Print P4 | Print P5<br>Signal(Z)<br>Signal(Z) | Wait(Z)<br>Print P6 |

**Question 3:** **[Marks: 10]**

Assume that a system distinguishes between four types of resources (A, B, C, and D), and the following is an example of how those resources could be distributed. Identify whether the system is in safe state? If yes, what will be the sequence?

**Note:** To obtain full marks, after each step show the available system resources and the resources allocated.

**Available system resources:**

| A | B | C | D |
|---|---|---|---|
| 3 | 3 | 4 | 3 |

**Processes (currently allocated resources):**

| | A | B | C | D |
|---|---|---|---|---|
| **P1** | 1 | 1 | 2 | 1 |
| **P2** | 1 | 0 | 3 | 3 |
| **P3** | 1 | 1 | 2 | 0 |
| **P4** | 3 | 0 | 4 | 2 |

**Processes (maximum resources):**

| | A | B | C | D |
|---|---|---|---|---|
| **P1** | 5 | 2 | 5 | 6 |
| **P2** | 3 | 1 | 3 | 3 |
| **P3** | 1 | 2 | 6 | 2 |
| **P4** | 5 | 6 | 5 | 6 |

**Question 4:** **[Marks: 10]**

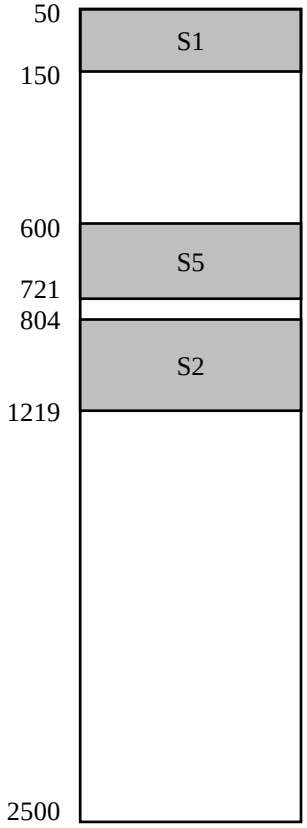
Given below is the incomplete snapshot, at time $T_0$, of a programs' segment table and the physical memory. The physical memory allocated to this program is from 50 to 2050.

(i) Update the segment table using the information obtained from the physical memory figure.

(ii) Update the physical memory figure using the information obtained from the segment table.

(iii) Update the segment table and physical memory figure with the segments 6 and 7 having size of 350 and 133, respectively.

| Seg # | Base | Length |
|---|---|---|
| S0 | 1219 | 600 |
| S1 | | |
| S2 | | |
| S3 | 167 | 355 |
| S4 | 1952 | 96 |
| S5 | | |
| S6 | | |
| S7 | | |



**Question 5:** **[Marks: 15 + 5 = 20]**

(a) Consider the following string of page references:

7, 4, 0, 10, 8, 7, 9, 4, 9, 6, 8, 2, 8, 0, 5, 2, 8, 8, 3, 4

Imagine you have four frames. Calculate the page faults for each of the following algorithms. You must show the contents of the four frames after every page fault.

(i) Optimal
(ii) LRU
(iii) FIFO

(b) Would the page faults generally increase or decrease for the three algorithms mentioned in part (a), if number of frames is increased from four to five? Explain your answer in less than three lines. (No calculations required).

**Question 6:** **[Marks: 10]**

Consider a memory system that generates 16-bits addresses and the frame size is 16-byte. At time $T_0$ the status of the page table is given below.

| Page # | Page Table Entry |
|---|---|
| 976 | 23 |
| 955 | 19 |
| 131 | 22 |
| 1038 | 45 |
| 658 | 14 |
| 335 | 36 |

a) What is the maximum number of entries within a single page/frame?

b) What is the maximum number of page table entries in the memory system?

c) Using the given page table, determine the physical addresses for the following logical addresses. Write down the physical addresses in binary and decimal format.

(i) 0000100000111010
(ii) 0001010011111101
(iii) 0011000100100101
(iv) 0100000011101011