# COMSATS UNIVERSITY ISLAMABAD
## DEPARTMENT OF COMPUTER SCIENCE
## Terminal Examination, Spring 2022

Class: - BSCS-4, BSSE-4
Subject: - Operating Systems (CSC322)
Dated: - June 27, 2022

Marks: - 50
Time: - 3 Hours

Name: _____ Reg # _____

**Question 1** **CLO- 2** **Marks [7+3+6+4=20]**

**Part 1:** Suppose, in an OS multilevel-feedback queue scheduling is implemented. There are three queues to hold the ready processes. In Q1, RR scheduling algorithm is used to schedule processes (quantum is 6, assigned on FCFS basis). While in Q2 and Q3, priority-based scheduling and SJF preemptive scheduling is used respectively. When a process arrives, it is kept in Q1. If a process in Q1, does not complete in one quantum then it is moved to Q2; if its priority is less 10 otherwise it is sent to Q3. When the Q1 gets empty, processes from Q2 are executed and then Q3 processes are executed. Considering this scheduling system, draw a Gantt chart showing the execution order of the following processes. **(7)**

| Process ID | Priority | Burst Time | Arrival Time |
|---|---|---|---|
| P0 | 12 | 4 | 0 |
| P1 | 8 | 24 | 3 |
| P2 | 16 | 15 | 1 |
| P3 | 24 | 14 | 6 |
| P4 | 6 | 18 | 2 |
| P5 | 4 | 12 | 4 |
| P6 | 20 | 14 | 5 |
| P7 | 1 | 12 | 7 |

**Note:** Smaller priority number means higher priority

**Part 2:** Consider the methods used by processes P1 and P2 for accessing their critical sections whenever needed, as given below. The initial values of shared Boolean variables S1 and S2 are randomly assigned.

| *Method Used by P1* | *Method Used by P2* |
|---|---|
| `while (S1 == S2) ;` | `while (S1 != S2) ;` |
| `Critical Section` | `Critical Section` |
| `S1 = S2;` | `S2 = not (S1);` |

Keeping in mind the methods used by both processes, figure out why process p1 cannot re-enter its critical section immediately after executing once in its critical section. **(3)**

**Part 3:** Give brief answer to the following questions **(6)**

a) What is the necessary condition to implement deadlock avoidance algorithm? **(2)**
b) How can we ensure that hold & wait condition never gets true? **(2)**

c) What are the common approaches to recover from a deadlock sate? **(2)**

**Part 4:** Consider the following resource-allocation state

| Process | Allocation | Request | Total Resources |
|---|---|---|---|
| P1 | 1,0,1,1,0 | 0,1,0,0,1 | 2,1,1,2,1 |
| P2 | 1,1,0,0,0 | 0,0,1,0,1 | |
| P3 | 0,0,0,1,0 | 0,0,0,0,1 | |
| P4 | 0,0,0,0,0 | 1,0,1,0,1 | |

Find whether the system is in deadlock state or not? If yes, then which processes are in deadlock? Also suggest a recovery scheme to get out of the deadlock situation. **(4)**

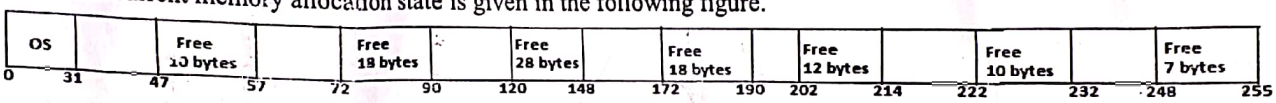
## Question 2

**CLO- 3** **Marks [8+7+5=20]**

**Part 1:** Give brief answer to the following questions **(8)**

a) Why logical and physical addresses are same in compile-time and load-time address binding? **(2)**
b) Does dynamic partitioning allocation scheme suffer from internal fragmentation? (Explain) **(2)**
c) Differentiate paging, demand paging and pure-demand paging? **(2)**
d) What is Belady's Anomaly? **(2)**

**Part 2:** Suppose in an OS segmentation is used for memory allocation. Total memory installed on the system is 256 bytes. The current memory allocation state is given in the following figure.




Suppose a new process $P_{10}$ arrives. Its total size is 60 bytes with segments as follows: $S_0$:11b, $S_1$:26b, $S_2$:16b, $S_3$:13b and $S_4$:12b.

a) Allocate the memory to $P_{10}$ using Best-Fit algorithm and write the segment table. **(3)**
b) Convert the following logical addresses to physical addresses: (0,8), (3,11), (4, 14) and (5,3) **(2)**
c) Suppose $P_{10}$ contains one more segment $S_5$:18b then show that external fragmentation exists. **(2)**

**Part 3:** Suppose pure-demand paging is implemented in an OS. A process P1; consisting of 10 pages access the pages in the following sequence

0, 4, 2, 0, 3, 4, 2, 5, 7, 4, 2, 5, 7, 9, 6, 1, 8, 5, 3, 4

If second-chance algorithm is used for page-replacement, then count the total number of page faults (assume quota of frames is 4). **(5)**

## Question 3

**CLO- 4** **Marks [5+5=10]**

**Part 1:** Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4,999. The drive is currently serving a request at cylinder 2,150, and the previous request was at cylinder 1,805. The queue of pending requests, in FIFO order, is:

2069, 1980, 2296, 1864, 1544, 2339, 356, 1523, 4965, 3681

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms? **(5)**

a. SSTF b. SCAN

**Part 2:** What are the different categories of OS security violation? Explain the security violation methods **(5)**

*The End*