**BRAC UNIVERSITY**

**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| Examination: Final  Duration: 2 Hours | Semester: Spring 2022  Full Marks: 40 |

CSE 321: Operating Systems

Answer the following questions.

Figures in the right margin indicate marks.

|  |  |  |  |
| --- | --- | --- | --- |
| 1.  **CO5** | 1. **Explain** Race conditions with an example. **Mention** how we protect the system from this phenomenon. 2. **Explain** with a code example how a careless ordering of semaphore operations can lead to a deadlock situation among two processes. 3. Suppose, you have to design an online consultation system for the teachers and students of your university. There are certain constraints that you have to keep in mind while designing the system -   i. the teachers can set their status whether they are available to give consultation or not. A teacher will set him unavailable after the giveConsultation() function.  ii. the students will enter a voice channel if the teacher is available for consultation.  iii. the students will wait if one student is in consultation with the teacher  Now, you have to **design** the teacher and student function using semaphores so that synchronization can be achieved among them maintaining the constraints mentioned above. You can use the following code template given below and complete it. **Mention the initial semaphore values before writing the functions.**   |  | | --- | | //initialize the semaphore values here  teacher(){  // write semaphore code here  *giveConsultation();*  // write semaphore code here  }  student(){  // write semaphore code here  *takeConsultation();*  // write semaphore code here  } | | [2+1]  [2]  [5] |
| 2.  **CO5** | 1. **Explain** how Banker's algorithm can help to find the processes that are causing a deadlock in a system. 2. **Describe** some strategies for deadlock prevention that can break the hold-and-wait condition. 3. Suppose, in an office, we have a set of resource types, R = {R1, R2, R3} and a set of processes, P = {P1, P2, P3, P4}. R1, R2, and R3 have 4, 2, and 2 instances respectively.    1. P1 is holding 2 instances of R1,    2. P2 requests 1 instance of R3    3. P3 requests 2 instances of R2    4. P2 requests 1 instance of R1    5. P2 is holding 1 instance of R2    6. P3 is holding 1 instance of R3   **Construct** a resource allocation graph for the above scenario and **identify** whether there is a deadlock or not.   1. Consider the following snapshot of a system:  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | **Allocation** | | | |  | **Max** | | | | |  |  | **A** | **B** | **C** | **D** |  | **A** | **B** | **C** | **D** | | **P1** |  | 0 | 0 | 1 | 2 |  | 0 | 0 | 2 | 3 | | **P2** |  | 1 | 0 | 0 | 0 |  | 1 | 2 | 2 | 0 | | **P3** |  | 1 | 3 | 5 | 4 |  | 2 | 3 | 5 | 6 | | **P4** |  | 0 | 0 | 0 | 1 |  | 2 | 2 | 0 | 1 | |  |  |  |  |  |  |  |  |  |  |  | |  |  | **Available** | | | |  |  |  |  |  | |  |  | 1 | 2 | 2 | 0 |  |  |  |  |  |   i. Is the system in a safe state?  ii. Can P3’s request (1 0 0 0) be safely granted immediately?  iii. If P3’s request is granted immediately, does the system enter a deadlock? | [2]  [2]  [3]  [3]  [2]  [3] |
| 3.  **CO6** | 1. **Explain** the disadvantage of using Contiguous allocation and how Paging is more beneficial than Contiguous allocation. 2. **Explain** how the operating system’s behavior and hardware mechanism for logical to physical address translation ensure that one process cannot access the memory allocated for another process. 3. At a particular time, the snapshot of the Main memory is given below for dynamic partition. Gray portions of the memory are occupied space.      * 1. **Apply** worst-fit and best-fit algorithms to allocate processes with the space requirement of P1(26k), P2(30k), P3(15k), P4(20k), and P5(6k).   2. **Explain** which algorithm makes the most effective use of memory?  1. Suppose, in a system, there are two processes - P1 (16 bytes) and P2 (12 bytes) with a page size of 4 bytes. The main memory size of the system is 32 bytes. Page tables of both processes are given below.   5  0  7  2  0  1  2  3  **f**  **p**  6  1  4  0  1  2  **f**  **p**  Page Table  of P1  Page Table  of P2  Find the corresponding physical address of the following logical addresses -  i. address 1011 of P1  ii. address 0100 of P1  iii. address 0111 of P2  iv. address 1010 of P2 | [3]  [2]  [4]  [2]  [4] |
|  |  |  |