**Department of Computer Science and Engineering**

**B**

**Final Examination Summer 2023**

**CSE 321: Operating Systems**

|  |  |
| --- | --- |
| **Duration:** 1 Hour 50 Minutes | **Total Marks:** 40 |

Answer the following questions.

Figures in the right margin indicate marks.

|  |  |  |
| --- | --- | --- |
| **1.**  **CO4** | **a)** Imagine a computer lab with multiple computers, and equipped with printers. Students use these computers for various tasks and may want to print documents. However, there are only a limited number of printers available. How do you solve this issue using semaphore? Your answer should have the steps associated in solving the given scenario. | [3] |
|  | **b)** How do semaphores and mutexes differ in their implementation. | [1.5] |
|  | **c)** For Peterson’s problem below conditions will be applied.   * There are two processes: P1 and P2. P2 gets to execute first. * Each Statement takes 4ms to execute. * Context Switch will occur after 12ms. * Critical section contains 4 statements. * Remainder section contains 3 statements. * For P1: i=0 and j=1 * For P2: i=1 and j=0 * turn=0 * flag[0] = FALSE, flag[1] = TRUE   **The structure of process Pi in Peterson’s solution:**   |  | | --- | | **do{**  **flag[i] = true;**  **turn = j;**  **while(flag[j] == true && turn == 1){**  **//busy wait**  **}**  **//critical section**  **flag[i] = false;**  **//remainder section**  **}while(true);** | |  |
|  | **Complete** the table below for processes P1 and P2 using **Peterson’s solution.**   |  |  | | --- | --- | | Process 1: i = 0, j = 1 | Process 2: i = 1, j = 0 | |  |  | |  |  | |  |  | | [3.5] |
| **2.**  **CO4** | **a)** Consider the following snapshot of a system:   |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | **Allocation** | | |  | **Max** | | |  | **Available** | | | |  |  | **A** | **B** | **C** |  | **A** | **B** | **C** |  | **A** | **B** | **C** | | **P0** |  | 5 | 2 | 2 |  | 9 | 9 | 8 |  | 7 | 10 | 5 | | **P1** |  | 3 | 2 | 0 |  | 9 | 9 | 10 |  |  |  |  | | **P2** |  | 5 | 3 | 3 |  | 6 | 8 | 5 |  |  |  |  | | **P3** |  | 3 | 0 | 0 |  | 6 | 7 | 9 |  |  |  |  | | **P4** |  | 4 | 2 | 1 |  | 5 | 7 | 6 |  |  |  |  | |  |
|  | 1. **Calculate** the Need Matrix. Is this system in a **safe state**? If yes, then find the safe sequence using Banker’s Safety algorithm otherwise, provide the necessary explanation. | [1+3] |
|  | 1. What happens if the process **P4** requests at this moment for **(0, 3, 1)?** Whether Banker’s algorithm grants the request or not? | [5] |
|  | **b)** Suppose, in a workplace, we have a set of resource types, R = {R1, R2, R3, R4} and a set of processes, P = {P1, P2, P3, P4, P5}. **R1, R2, R3, and R4** have **3, 1, 4, and 2** instances respectively.   * P1 is holding 2 instances of R1 * P2 is holding 1 instance of R3 * P3 is holding 1 instance of R4 * P5 requests 2 instances of R3 * P4 is holding 1 instance of R4 * P3 requests 1 instance of R2 * P2 requests 1 instance of R1 * P2 is holding 1 instance of R2 * P1 is requesting 1 instance of R4 * P3 is holding 1 instance of R3 * P4 is holding 1 instance of R3. |  |
|  | **Construct** a resource allocation graph for the above scenario and **identify the cycle (if any) and decide** whether there is a deadlock or not. | [4] |
| **3.**  **CO5** | **a)** Given variable size memory (dynamic) partitions of **10 MB, 16 MB, 45 MB, and 49 MB** (in order, top to bottom), **apply** best-fit and worst-fit algorithms to place processes with the space requirement of **5 MB, 15 MB, 10 MB, 5 MB, 10 MB, 20 MB, 25 MB, and 15 MB** (in order). Which algorithm makes the most effective use of memory? | [4+1] |
|  | **b)** How cache memory can be used to design an efficient paging hardware. Your answer should have the required diagram with the necessary justification. | [3] |
|  | **c)** Compare the following systems in terms of performance:  1.  A system with a hit ratio of 72%, associative lookup time of 24ns, and memory access time of 56ns.  2. A system with a hit ratio of 65%, associative lookup time of 6ns, and memory access time of 133ns. | [3] |
| **4.**  **CO5** | **a)** Suppose there are two processes **P1 (32 B)** and **P2 (40 B)** with a page size of **8 B**. The main memory size is **96 B.** The page table for P1 and P2 are given below:   |  |  | | --- | --- | | Page# | Frame# | | 0 | 5 | | 1 | 8 | | 2 | 11 | | 3 | 1 | | 4 | 6 |   Page table: P2   |  |  | | --- | --- | | Page# | Frame# | | 0 | 3 | | 1 | 10 | | 2 | 2 | | 3 | 0 |   Page table: P1 |  |
|  | **Find** the corresponding physical addresses of the following logical addresses:   1. Address 000001 of P1 2. Address 001001 of P2 3. Address 101001 of P2 | [3] |
|  | **b)** Consider a computer with a main memory that has 5 frames and page reference string of 0-7 page **[6, 5, 3, 6, 4, 5, 0, 5, 5, 1, 1, 5, 0, 5, 4, 6, 3, 5, 1, 5]**. The page reference string represents the order in which the pages are accessed by a program. **Apply LRU** & **OPT** algorithm to **simulate** the page replacement that occurs when the main memory can hold at most 5 pages at a time. **Record** the number of **page faults** and compare the result. **Mention** which algorithm performs better in this scenario. | [4+1] |
|  |  |  |
|  |  |  |