**Department of Computer Science and Engineering**

**A**

**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)** What is meant by busy waiting in the implementation of a critical section solution, and why is it considered problematic? | [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 | |  | Flag[1]=T  T=0  While False | | Flag[0]=T  T=1  While T |  | |  | CS 1-3 | | While T |  | |  | CS4,  Flag[1]=False  RS1 | | While F  CS 1-2 |  | |  | RS 2,3 | | Rest of it |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | | [3.5] |