

CSE325: Operating System Laboratory Assignment

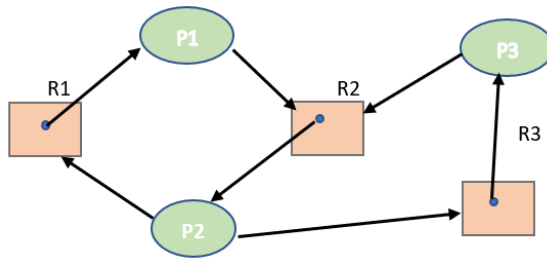
Last date of submission is: 07th April 2020

Instructions to be followed

1. The assignment is to be done on individual basis (no groups)
2. All the questions are compulsory.
3. The assignment submission mode is **Online** only. Student has to upload the assignment on or before the last date on UMS only. No submission via e-mail or pen-drive or any media will be accepted.
4. Non-submission of assignment on UMS till the last date will result in **ZERO** marks.
5. The student is supposed to solve the assignment on his/her own. If it is discovered at any stage that the student has used unfair means like copying from peers or copy pasting the code taken from internet etc. **ZERO** marks will be awarded to the student.
6. Attach the screenshot of the output of your program along with code.
7. Solve questions which is specified for your group.

Questions for G1 students

1. Consider the following resource allocation graph.



Write a C program to demonstrate the deadlock scenario given in the graph and also write a solution to remove deadlock.

2. Write a C program to show how semaphores are used to solve Reader Writers problem of synchronization.
3. A parent process creates a child process. The child process after its creation will send a message "Hello parent, this is child process" to its parent through pipe. Once the message is received by the parent, the parent will execute and print "This is Parent process".
4. Write a C program to avoid inconsistency that might occur when a multithreaded system is used to transfer money from one account to another account. Write a program to avoid such situation using locks.

Questions for G2 students

1. Write a C program to show how semaphores are used to solve Dining Philosopher's problem of synchronization.
2. Consider the following scenario



Write a program to demonstrate the above scenario in which the parent will send a message to child "Hello". After the receiving the message the child will execute and send the acknowledgment message to the parent "Message received".

3. Write a program to show how race condition can be avoided using semaphores.

4. Consider the following resource allocation graph:

P = {P1, P2, P3}

R = {R1, R2, R3}

E = {R1 --> P1, P1 --> R2, R2 --> P2, P2 --> R3,

R3 --> P3, P3 --> R1}

resource type R1 has one instance

resource type R2 has one instance

resource type R3 has one instance

Write a C program to demonstrate the above deadlock scenario. How it can be avoided?