# COEN -283

**Operating Systems**

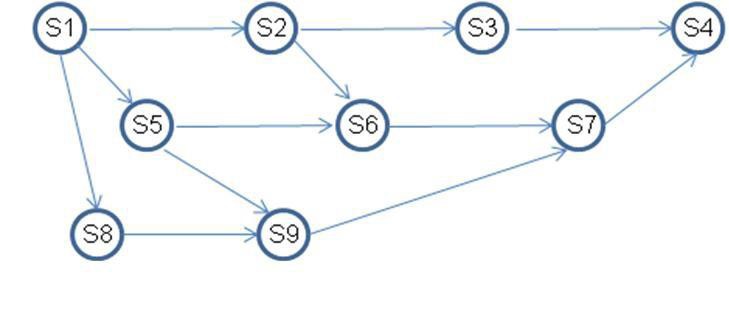
**Winter 2017**

**Assignment 2 - IPC and Scheduling**

## Ques 1: (20 points)

Assume you are given the following wait-graph that represents the relationship between multiple threads (s1,s2,s3,…). An arrow from one thread (Sy) to another (Sx) means that thread Sx must finish its computation before Sy starts. (For example: S1 has to wait for S2,S5,S8 to finish, S2 has to wait for s3,s6 to finish and so on.)

Use semaphores to enforce this relationship specified by the graph. Be sure to show the initial values and the locations of the semaphore operations. You will be marked based on finding the best solution with minimum number of semaphores.



## Ques 2(10 points):

A counting semaphore was initialized to 10. Then 6 P operations and 4 V operations were completed on this semaphore. The resulting value of the semaphore is

(a) 0 (b) 8    (c) 10     (d) 12

Give brief explanation to support your answer.

## Ques 3: (10 points)

What is the meaning of the term busy waiting? What other kinds of waiting are there in an operating system? Can busy waiting be avoided altogether? Explain your answer.

**Ques 4. (10 Points)**

## Suppose we replace the wait () and signal () operations of monitors with a single construct await(B), where B is a general Boolean expression that causes the process executing it to wait until B become true. Explain why, in general, this construct cannot be implemented efficiently?

## Ques 5:

**(20 points)** The following pair of processes share a common variable X:

|  |  |  |
| --- | --- | --- |
|  | **Process A** | **Process B** |
| **L1:** | int Y | int Z |
| **L2:** | Y = X\*2 | Z = X+1 |
| **L3:** | X = Y | X = Z |

X is set to 5 before either process begins execution. As usual, statements within a process are executed sequentially, but since no assumptions can be made regarding each process’s speed of execution, statements in either process may execute in any order with respect to statements in the other process.

1. How many different values of X are possible after both processes finish executing?
2. suppose the programs are modified as follows to use a shared binary semaphore T:

|  |  |  |
| --- | --- | --- |
|  | **Process A** | **Process B** |
| L1: | int Y | int Z |
| L2: | Y = X\*2 | P(T) |
| L3: | X = Y | Z = X+1 |
| L4: | V(T) | X = Z |

T is set to 0 before either process begins execution or, as before, X is set to 5. Now, how many different values of X are possible after both processes finish executing?

**Programming Question**

**Ques 6: (30 points)**

Solve the dining philosopher’s problem using monitors instead of semaphores.