OS ASSIGNMENT

Process synchronization and deadlocks

1.

What are the necessary conditions for deadlock?Fix the following code to avoid the possible deadlock. What condition did you remove by making your change?

Process A	Process B
acquire(L1) acquire(L2) release(L2) release(L1)	acquire(L2) acquire(L1) release(L1) release(L2)

2.

What's the difference between a process starting another copy of itself and starting another thread? Suppose a program has three threads Thread1, Thread2, and Thread3, and ashared counter variable,count, as shown below:

```
int count = 10;
Semaphore Lock = 1;          // initial value is 1
```

Thread1	Thread2	Thread3

// do something
wait(Lock);
count++;
Signal(Lock);
	// do something
	wait(Lock);
	count—;
	Signal(Lock);
		// do something
		wait(Lock);
		printf("%d", count);
		Signal(Lock);

What are the possible outputs of this program? If there is more than one answer,provide them all. Does this process suffer from a race condition? Justify your answer.

3.

. **Explain deadlock avoidance Apply** deadlock avoidance to the following problem. A restaurant would like to serve four dinner parties, P1 through P4. The restaurant has a total of 8 plates and 12 bowls. Assume that each group of diners will stop eating and wait for the waiter to bring a requested item (plate or bowl) to the table when it is required. Assume that the diners don't mind waiting. The maximum request and current allocation tables are shown as follows:

Maximum Request	Plates	Bowls
P1	7	7
P2	6	10
P3	1	2
P4	2	4

Current Allocation	Plates	Bowls
P1	2	3
P2	3	5
P3	0	1
P4	1	2

i) Determine the Need Matrix for plates and bowls.

Need	Plates	Bowls
P1		
P2		
P3		
P4		

ii) Will the restaurant be able to feed all four parties successfully? Clearly explain your answer – specifically, why no or why/how there is a safe serving order.

iii) Assume a new dinner party, P5, comes to the restaurant at this time. Their maximum needs are 5 plates and 3 bowls. Initially, the waiter brings 2 plates to them. In order to be able to feed all five parties successfully, the restaurant needs more plates.

- Determine the new Need Matrix for plates and bowls.

Need	Plates	Bowls
P1		
P2		
P3		
P4		
P5		

- At least how many plates would the restaurant need to add?
- Show a safe serving sequence.

4.

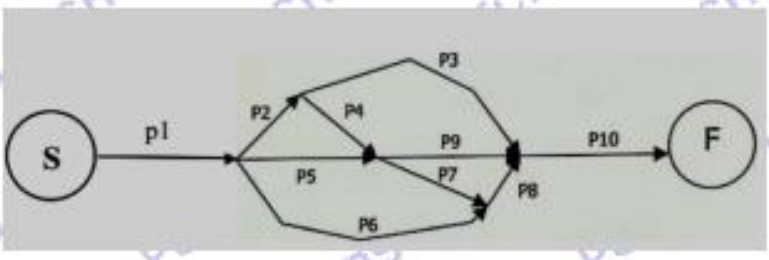
Consider the following resource allocation policy. Requests and releases for resources are allowed at any time. If a request for resources cannot be satisfied because the resources are not available, then we check any process that are blocked, waiting for resources. If they have the desired resource, then these resources are taken away from them and are given to the requesting process. The vector of resources for which the waiting process is waiting is increased to include the resources that were taken away.

i) Can deadlock occur in the above system? Justify with an example.

ii) Can indefinite blocking occur in the above system? Why?

5.

What do you mean by the term synchronization? Explain how semaphore can be used as synchronization tool. Consider the following system flow diagram consisting of 10 processes labeled from p1 to p10. S and F denote the start and end. Write pseudo code for synchronization using semaphore.



6.

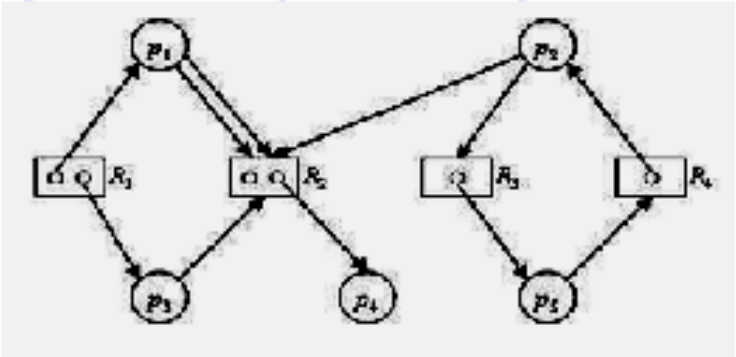
Consider a system consisting of two processes, P0 and P1, each accessing two semaphores, S and Q, set to the value 1.

P0	P1
Wait(S)	Wait(S)
Wait(Q)	Wait(Q)
...	...
...	...
...	...
Signal(S)	Signal(Q)
Signal(S)	Signal(S)

What kind of unwanted situation(s) will happen?

7.

Consider the following resource allocation graph.



Which processes are blocked and which processes are deadlocked? Is the state a deadlock state? If so, show the irreducible sub graph.

8.

What is Semaphore? In the following code, three processes produce output using the routine putc and synchronize using two semaphores L and R.
Answer the following with explanations.

semaphore L = 3, R = 0;

/* Process 1 */	/* Process 2 */	/* Process 3 */

L1:
wait(L)
putc('C');
signal(R);
goto L1;
	L2:
	wait(R)
	putc('A');
	putc('B');
	signal(R)
	goto L2;
		L3:
		wait(R)
		putc('D');
		goto L3;

a) How many D's are printed when this set of processes runs?

b) What is the smallest number of A's that might be printed when this set of processes runs?

c) Is CABABDDCABCABD a possible output sequence when this set of processes runs?

d) Is CABACDBCABDD a possible output sequence when this set of processes runs?

9.

What is a semaphore? Explain how semaphore can be used as synchronization tool. Consider the following 3-process concurrent program which uses semaphores S1, S2, and S3. The semaphore operation, which are sometimes called "wait" and "signal", are denoted here with the classical notation of "P" and "V".

Process 1	**Process 2**	**Process 3**
L1:P(S3);	**L2:P(S1);**	**L3:P(S2);**
print("T");	**print("U");**	**print("B");**
V(S2);	**V(S3);**	**V(S1);**
goto L1;	**gotoL2**	**goto L3;**

i) Are there initial values that can be given to the semaphores so that the processes cooperate to print the string BUTBUTBUTBU? If so, give the initial values (tell which value is to be used for which semaphore) and explain how the string is printed.

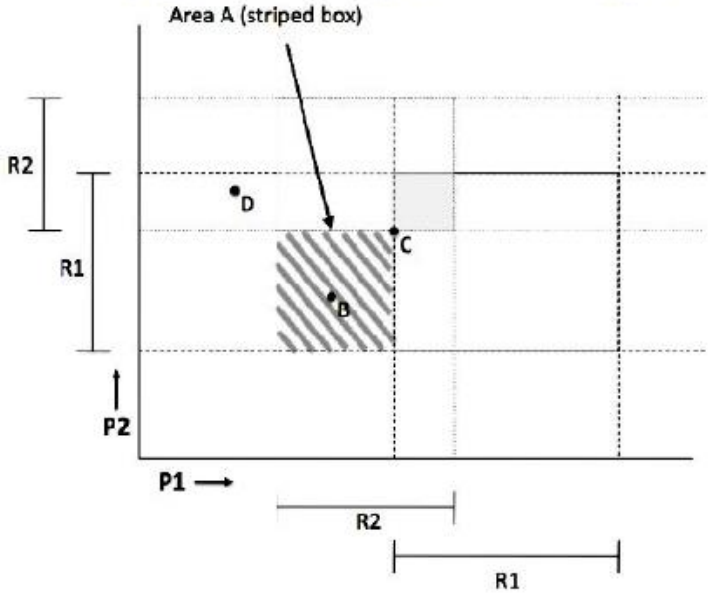
ii) Suppose the initial values are S1=0, S2=0, S3=0. Is it possible for the processes to cooperate to produce a string that begins with BBTTUTT? Explain your answer.

10.

Consider a system with 10 units of a single resource. There are four processes in the system, named [A,B,C,D], each of which have a maximum resource requirement that can be specified by the vector [6,5,4,7] units, where each element indicates the maximum needs of the corresponding process. If the processes currently hold [1,2,2,4] resource units, is the system in a safe state (i.e., is deadlock possible given the maximum needs of each process)?

11.

Consider the following resource trajectory graph for a system with two processes (P1 and P2) and two resources (R1 and R2)



i) What does the area identified as Area A indicate?

ii) Draw three resource allocation diagrams (the ones with circles, squares and arrows), one for each of the points indicated above as B, C, and D.

iii) If you drew the same resource allocation graphs for both B and C, explain why. If you drew different resource allocation graphs for points B and C, explain why. What does that tell you about the usefulness of using resource allocation graphs to avoid deadlock?

iv) In what way is the banker's algorithm "pessimistic"?

v) Write, in C, two very simple, non-looping programs that use shared binary semaphores, such that deadlock among all three could occur if nothing was done to prevent it. Assume the programs don't use any other resource that could cause deadlock. Give a scenario of execution in which your two programs actually deadlock.

vi) Assume that each program has to declare, when it starts, which semaphores it will be using. In what way would the banker's algorithm, avoids the deadlock in the scenario you gave in your previous answer?

12.

You have just been hired by "PSG Nature Club" to help them out with the chemical reaction to form water, which they do not seem to be able to get right due to synchronization problems. The trick is to get two H atoms and one O atom all together at the same time. The atoms are threads. Each H atom invokes a procedure hReady when it is ready to react and each O atom invokes a procedure oReady when it is ready. For this problem, you are to write the code for hReady and oReady. The procedures must delay until there are at least two H atoms and one O atom present, and then one of the threads must call the procedure makeWater (which just prints out a message that water was made). After the makeWater call, two instances if hReady and one instance of oReady should return. Your solution should avoid starvation and busy-waiting.

i) Specify the correctness constraints

ii) Write the pseudo code implementation of hReady, oReady and makeWater using semaphores.