BTLT 4 – SYNCHRONIZATION AND DEADLOCK

1. Semaphore s12 = 0, s23 = 0, s34 = 0, s45 = 0

Client {	**Shopping** {
1-makeOrder(); Up(s12); Down(s23); **3-**provideRequiredInformation(); Up(s34); Down(s45); **5-**receiveProduct(); }	Down(s12); **2-**receiveOrder(); **2-**requireAdditionalInformation(); Up(23); Down(s34); **4-**verifyReceivedInformation(); **4-**deliverProduct(); Up(s45); }

2. Semaphore sH=2, sO=1, s=0;

P1 {	P2 {	P3 {

Down(sO);
create1Oxygen();
Up(s);
}
	Down(sH);
	create1Hydrogen();
	Up(s);
	}
		Down(s);
		Down(s);
		Down(s);
		create1H2o();
		Up(sO);
		Up(sH);
		Up(sH);
		}

3. Trường hợp xấu nhất mỗi tiến trình sở hữu 2 resource: 2(resource) x 2(processes) = 4 ➔ Deadlock

4. 3 x 1 = m ➔ có deadlock ➔ m tối thiểu 3 + 1 = 4 ➔ ko deadlock

5. a. n = 6 ➔ deadlock; b. n – 1 = 5 ➔ ko deadlock

6. 2(P1)+ 0(P2) + 1(P3) + 1(P4) = 4 + 1= n ➔ ko deadlock

7. Consider the following snapshot of a system.

	Allocation					Max					Need				
	A	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**
P1	1	3	1	3	2	4	5	2	3	2	3	2	1	0	0
P2	2	0	0	3	1	3	2	3	4	2	1	2	3	1	1
P3	4	1	0	0	0	4	1	1	1	2	0	0	1	1	2
P4	0	0	0	1	1	1	2	1	2	2	1	2	1	1	1
P5	1	2	3	0	0	2	2	3	1	1	1	0	0	1	1

Available					
A	B	C	D	E	
1	2	2	2	1	
2	4	5	3	2	P4, P5: ok
8	5	5	6	3	P2, P3: ok
					P4, P5, P3, P2, P1
0	0	1	2	1	d.==> no-safe

a. Fill in the matrix Need.
b. Is the system in a safe state Safe? Explain your answer. P4, P5, P2, P3, P1
c. If P1 requests for (1, 2, 3, 4, 5), can the request be granted? Explain your answer.
Ko vì tài nguyên hiện tại ko đủ.
d. If P1 requests for (1, 2, 1, 0, 0), can the request be granted? Explain your answer.
Tài nguyên thoả mãn ➔ Thử cấp phát nếu tìm thấy một chuỗi an toàn thì OK.