

Sinh viên : Huỳnh Trí Nhân

MSSV: 21120302

Bộ môn: Hệ điều hành 21_4

Giáo viên hướng dẫn: TS. Vũ Thị Mỹ Hằng

Bài tập luyện tập 4: Synchronization & Deadlock

Câu 1:

1. Consider an online shopping process consisting of the following steps:

- > makeOrder
- > receiveOrder
- > requireAdditionalInformation
- > provideRequiredInformation
- > verifyReceivedInformation
- > deliverProduct
- > receiveProduct

The following processes Client and Shopping allow performing the above steps:

Client { makeOrder(); provideRequiredInformation(); receiveProduct(); } Shopping { receiveOrder(); requireAdditionalInformation(); verifyReceivedInformation(); deliverProduct(); } }	
---	--

Synchronize processes Client and Shopping Stored by using semaphores to ensure the order of the above shopping process.

Semaphore s_client = 0, s_shopping = 0;

```
Client{
    makeOrder();
    up(s_shopping );
    down(s_client );
    provideRequiredInformation();
    up(s_shopping );
    down(s_client );
    receiveProduct();
}
```

```
Shopping{
    down(s_shopping);
    receiveOrder();
    requireAddtionalInformation();
    up(s_client);
    dow(s_shopping);
    verifyReceivedInformation();
}
```

```

        deliverProduct();
        up(s_client);
    }

```

Câu 2:

2. To form a H₂O molecule, one oxygen atom and two Hydrogen atoms are needed. Let us suppose three processes P1, P2 and P3 allowing to create an Oxygen atom, a Hydrogen, and a H₂O molecule, respectively, as below:

P1 { create1Oxygen(); }	P2 { create1Hydrogen(); }	P3 { create1H2o(); }
-------------------------------	---------------------------------	----------------------------

Write pseudo-code for these three processes and use semaphores to synchronize them so that:

- Different processes P1 and P2 can coexist at the same time in the system.
- Only one process P3 performs its function at a time in the system.
- If one Oxygen atom and two Hydrogen atoms have been produced, all instances of P1 and P2 will be blocked until all existing atoms assembled into H₂O molecule (P3). P3 will be blocked if there are not enough oxygen and hydrogen atoms.

semaphore created_O = 0, created_H = 0, needed_O = 1, needed_H = 2;

```

P1{
    down(needed_O );
    create 1Oxygen();
    up(created_O );
}
P2{
    down(needed_H );
    create1Hydrogen();
    up(created_H );
}
P3{
    down(created_H);
    down(created_H);
    down(created_O);
    create1H2O();
    up(needed_H );
    up(needed_H );
    up(needed_O );
}

```

Câu 4:

4. Consider a system consisting of m resources of the same type that are shared by 4 processes, each of which needs at most 3 resources. What is the minimum value of m that ensures no deadlock?

-Trường hợp deadlock xảy ra khi mỗi process thiếu 1 resources (tức $m=8$). Vì vậy để tránh deadlock thì $m = 9$

Câu 7:

- Consider the following snapshot of a system.

	Allocation					Max					Need (Max - Allocation)				
	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
P1	2	3	3	1	1	4	5	7	1	1					
P2	1	0	2	1	4	2	2	3	1	5					
P3	5	2	2	2	2	5	3	7	2	3					
P4	4	1	0	1	0	5	2	1	2	2					
P5	1	1	3	1	1	4	5	3	1	1					

Available				
R1	R2	R3	R4	R5
3	2	2	1	1

- Fill in the matrix Need.
- Is the system in a safe state **Safe**? Explain your answer.

	Allocation					Max					Need = (Max- Allocation)				
	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
P1	2	3	3	1	1	4	5	7	1	1	2	2	4	0	0
P2	1	0	2	1	4	2	2	3	1	5	1	2	1	0	1
P3	5	2	2	2	2	5	3	7	2	3	0	1	5	0	1
P4	4	1	0	1	0	5	2	1	2	2	1	1	1	1	2
P5	1	1	3	1	1	4	5	3	1	1	3	4	0	0	0
Available															
	R1	R2	R3	R4	R5										
	3	2	2	1	1										

b) Hệ thống an toàn : Do có một kịch bản khiến toàn bộ tiến trình chạy:

P2 -> P1 -> P4 -> P5 -> P3