

操作系统同步与死锁作业

1. 仓库中可以存放产品 A 和产品 B，容量无限；但是仓库中产品 A 的数量 M 和产品 B 的数量 N 之间必须满足： $M-N < 5$ 以及 $N-M < 10$ ，请用信号量实现产品 A 的生产者进程和产品 B 的生产者进程。

Use S to present the operator of putting in products, use A to present the number product A can be put in, use B to present the number product B can be put in.

Semaphore S=1, A=4, B=9;

Aproducer()

```
{
    wait(A);
    wait(S);
    putinA();
    signal(S);
    signal(B);
}
```

Bproducer()

```
{
    wait(B);
    wait(S);
    putinB();
    signal(S);
    signal(A);
}
```

2. Consider the following snapshot of a system. There are no current outstanding queued unsatisfied requests.

Available			
R1	R2	R3	R4
2	1	0	0

Process	Current allocation				Maximum demand				Still need			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
P1	0	0	1	2	0	0	1	2				
P2	2	0	0	0	2	7	5	0				
P3	0	0	3	4	6	6	5	6				
P4	2	3	5	4	4	3	5	6				
P5	0	3	3	2	0	6	5	2				

a. Is this system currently in a safe or unsafe state? Why?

Safe, use (x x x x) to present the number of R1,R2,R3,R4, as below:

P1 needs (0 0 0 0), P1 can finish, available becomes (2 1 0 0)+(0 0 1 2)=(2 1 1 2)

P4 needs (2 0 0 2), P4 can finish, available becomes (2 1 1 2)+(2 3 5 4)=(4 4 6 6)
P5 needs (0 3 2 0), P5 can finish, available becomes (4 4 6 6)+(0 3 3 2)=(4 7 9 8)
P2 needs (0 7 5 0), P2 can finish, available becomes (4 7 9 8)+(2 0 0 0)=(6 7 9 8)
P3 needs (6 6 2 2), P3 can finish, so all can finish.

- b. Is this system currently be deadlocked? Why or why not?

No, because it can run by order as P1->P4->P5->P2->P3.

- c. If a request from P3 arrives for (0, 1, 0, 0), can that request be safely granted immediately? In what state (deadlocked, safe, unsafe) would immediately granting that request leave the system?

No, this system can't find an order to finish all processes.

It should be in safe state after granting the request.

3. Consider the following snapshot of a system:

		Allocation					Max					Available			
		A	B	C	D		A	B	C	D		A	B	C	D
P0		0	0	1	2		0	0	1	2		1	5	2	0
P1		1	0	0	0		1	7	5	0					
P2		1	3	5	4		2	3	5	6					
P3		0	6	3	2		0	6	5	2					
P4		0	0	1	4		0	6	5	6					

Answer the following questions using the banker's algorithm:

- a. What is the content of the matrix Need?

matrix Need=Max-Allocation:

	Need			
	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	1	0	0	2
P3	0	0	2	0
P4	0	6	4	2

- b. Is the system in a safe state?

Safe:

P0 needs (0 0 0 0), P0 can finish, available becomes (1 5 2 0)+(0 0 1 2)=(1 5 3 2)

P2 needs (1 0 0 2), P2 can finish, available becomes (1 5 3 2)+(1 3 5 4)=(2 8 8 6)

P3 needs (0 0 2 0), P3 can finish, available becomes (2 8 8 6)+(0 6 3 2)=(2 14 11 8)

P4 needs (0 6 4 2), P4 can finish, available becomes (2 14 11 8)+(0 0 1 4)=(2 14 12 12)

P1 needs (0 7 5 0), P1 can finish, so all can finish.

- c. If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately?

Yes, after granting the request, available becomes (1 1 0 0)

P0 needs (0 0 0 0), P0 can finish, available becomes (1 1 0 0)+(0 0 1 2)=(1 1 1 2)

P2 needs (1 0 0 2), P2 can finish, available becomes (1 1 1 2)+(1 3 5 4)=(2 4 6 6)

P3 needs (0 0 2 0), P3 can finish, available becomes (2 4 6 6)+(0 6 3 2)=(2 10 9 8)

P1 needs (0 3 3 0), P1 can finish, available becomes (2 10 9 8)+(1 4 2 0)=(3 14 11 8)

P4 needs (0 6 4 2), P4 can finish, so all can finish.

Order is P0->P2->P3->P1->P4.