

1. 时间片轮转调度算法: P1 P2 P3 P4 P1 P2 P4 P1 P2 P2

响应比高者优先调度算法: P3 P4 P1 P2

2. 一条河上架设了由若干个桥墩组成的一座桥。若一个桥墩只能站一个人，过河的人只能沿着桥向前走而不能向后退。过河时，只要对岸无人过，就可以过。但不允许河对岸的两个人同时过，以防止出现死锁。请给出两个方向的人顺利过河的同步算法。

【答案】

信号量 **s**: 互斥使用桥，初值为 1

信号量 **scount1**: 对方向 1 上过河人计数器 **count1** 的互斥使用，初值为 1

信号量 **scount2**: 对方向 2 上过河人计数器 **count2** 的互斥使用，初值为 1

信号量 **scount**: 代表桥上过河人的计数信号量，初值为桥墩个数 **N**

变量 **count1**: 方向 1 上过河人计数器

变量 **count2**: 方向 2 上过河人计数器

Semaphore **s, scount1, scount2, scount**;

int **count1, count2**;

s=1; scount1=1; scount2=1; scount=N;

count1=0; count2=0;

void **direct1(int i)**

{

wait(scount1);

if(count1==0)

wait(s);

count1++;

signal(scount1);

wait(scount);

上桥，过桥，下桥；

signal(scount);

wait(scount1);

count1--;

if(count1==0)

signal(s);

signal(scount1);

}

```

void direct2(int i)
{
wait(scount2);
if(count2==0)
    wait(s);
count2++;
signal(scount2);

wait(scount);
上桥，过桥，下桥；
signal(scount);

wait(scount2);
count2--;
if(count2==0)
    signal(s);
signal(scount2);
}

main()
{
    cobegin{
        direct1(1);
        ...
        direct1(n);
        direct2(1);
        ...
        direct2(m);
    }
}

```

3. 有一个仓库，可以存放 A 和 B 两种产品，但要求：（1）每次只能存入一种产品（A 或 B）；（2） $-N < A \text{ 产品数量} - B \text{ 产品数量} < M$ 。其中，N 和 M 是正整数。试用同步算法描述产品 A 与产品 B 的入库过程。

【答案】

A 产品的数量不能比 B 产品的数量少 N 个以上，A 产品的数量不能比 B 产品的数量多 M 个以上。

设置两个信号量来控制 A、B 产品的存放数量，**sa** 表示当前允许 A 产品比 B 产品多入库的数量（当前允许 A 产品入库数量），即在当前库存量和 B 产品不入库的情况下，还可以允许 sa 个 A 产品入库；

sb 表示当前允许 B 产品比 A 产品多入库的数量（当前允许 B 产品入库数量），即在当前库存量和 A 产品不入库的情况下，还可以允许 sb 个 B 产品入库。

初始时，**sa** 为 **M-1**，**sb** 为 **N-1**。当往库中存放入一个 A 产品时，则允许存入 B 产品的数量也增加 1；当往库中存放入一个 B 产品时，则允许存入 A 产品的数量也增加 1。

```
semaphore mutex=1, sa=M-1, sb=N-1;
```

```
process puta()
```

```
{ while(1)
```

```
{ 取一个产品;
```

```
    wait(sa);
```

```
    wait(mutex);
```

```
    将产品入库;
```

```
    signal(mutex);
```

```
    signal(sb);
```

```
}
```

```
}
```

```
process putb()
```

```
{ while(1)
```

```
{ 取一个产品;
```

```
    wait(sb);
```

```
    wait(mutex);
```

```
    将产品入库;
```

```
    signal(mutex);
```

```
    signal(sa);
```

```
}
```

```
}
```

```
main()
```

```
{  cobegin{
```

```
    puta();
```

```
    putb();
```

```
}
```

```
}
```

4. 可访问的内存空间大小为 $2^{32}=4\text{GB}$

物理块的大小等于页的大小，则大小为： $2^{12}=4\text{KB}$

一个作业最多能分的页数为： $2^{32-12}=1\text{M}$ 个

5. (1) 为某文件分配两个盘块的过程如下：

顺序检索位示图，从中找到第一个值为 0 的二进制位，得到其行号 $i1=2$ ，列号 $j1=2$ ；第二个值为 0 的二进制位，得到其行号 $i2=3$ ，列号 $j2=6$ 。

计算出找到的两个空闲块的盘块号分别为：

$$b1=i1\times 16+j1+1=2\times 16+2+1=35$$

$$b1=i2\times 16+j2+1=3\times 16+6+1=55$$

修改位示图，令 $\text{Map}[2, 2]=\text{Map}[3,6]=1$ ，并将对应块 35、55 分配出去。

(2) 释放磁盘的第 300 块时，应进行如下处理：

计算出磁盘第 300 块所对应的二进制位的行号 i 和列号 j ：

$$i=(300-1)/16=18, j=(300-1)\text{Mod }16=11$$

修改位示图，令 $\text{Map}[18,11]=0$ ，表示对应块为空闲块。