

操作系统期中考试试题

1、请各举一个进程的例子和程序的例子。

答：可以同时用 word 打开多个文档，word 就是程序，用 word 打开 n 个文档对应的就是 n 个进程。进程包括程序和数据两部分，word 是一段保存在磁盘上的程序，待打开的文档内容就是数据，调用 word 程序来打开一篇文档就形成一个进程。

2、进程有哪些基本状态？画出状态之间的转换关系图，举例说明引起进程状态转换的事件。

答：进程基本状态包括新建、就绪、运行、阻塞和结束 5 个（图略）。转换关系包括：

新建→就绪 确认
就绪→运行 CPU 调度
运行→就绪 时间片用完、中断
运行→阻塞 等待事件/请求 IO
阻塞→就绪 事件出现/IO 完成
运行→结束 终止

3、设内存中有 P1, P2, P3 三道进程，并按照 P1, P2, P3 的优先级次序运行，其中内部计算和 IO 操作时间由下表给出（CPU 计算和 IO 资源都只能同时由一个进程占用）：

P1: 计算 60ms→IO 80ms→计算 20ms

P2: 计算 120ms→IO 40ms→计算 40ms

P3: 计算 40ms→IO 80ms→计算 40ms

请问完成三道程序并发执行比单道运行可以节省多少时间？

答：

由于每个进程都有三个阶段：计算、IO、计算，我们将这三次计算命名为 A、B、C。

60ms 80ms 20ms 40ms 40ms 40ms 40ms 40ms

P1(A)--> P1(B) --> P1(C)

 P2(A) P2(A) --> P2(B) --> P2(C)

 P3(A) --> P3(B) P3(B) --> P3(C)

最终耗时：60+80+20+40+40+40+40+40=360ms；

全串行执行耗时：160+200+160=520ms；

节约了 520ms-360ms=160ms。

4、有 5 个任务 A, B, C, D, E，它们几乎同时先后达到，预计它们运行的时间为 10, 6, 2, 4, 8 分钟。其优先级分别为 3, 5, 2, 1, 4，这里 5 为最高优先级。对下列每一种调度算法，计算其平均进程周转时间并写出调度序列（进程切换开销可不考虑）。

①先来先服务算法

②高优先级调度算法。（非抢占式）

③时间片轮转调度算法。（时间片 2min）

答：

①先来先服务。调度序列：A -> B -> C -> D -> E $T = (10+16+18+22+30)/5=19.2$

②优先级调度。（非抢占式）调度序列：B -> E -> A -> C -> D $T = (6+14+24+26+30)/5=20$

③时间片轮转调度。（时间片 2min）

第一轮：(A B C D E)

第二轮：(A B D E)

第三轮：(A B E)

第四轮：(A E)

第五轮：(A)

$T = (30 + 22 + 6 + 16 + 28) / 5 = 20.4$

***5、有一个内存中只能装入两道作业的批处理系统，作业调度采用短作业优先算法，进程调度采用以优先级为基础的抢占方式调度算法。有如下表所示的作业序列，表中所列的优先数是指进程调度的优先数，优先数越小优先级越高。

作业名	到达时间	运行时间 (min)	优先数
A	10:00	40	5
B	10:20	30	3
C	10:30	50	4
D	10:50	20	6

①画出 4 个作业的调度和运行情况。

②列出所有作业进入内存的时刻以及结束时刻。

③计算作业的平均周转时间。(从到达时刻算起)

答：

①第一小题画图 (略)

②A、B、C、D 各作业进入内存的时刻分别是 10:00、10:20、11:10、10:50；它们的完成时刻分别是 11:10、10:50、12:00、12:20。

③A、B、C、D 的周转时间分别是 70 分钟、30 分钟、90 分钟、90 分钟，故它们的平均周转时间为 70 分钟。

6、3 个进程共享 4 个同样类型的资源，每个进程最大需要 2 个资源，请问该系统是否会因为竞争该资源而死锁？请说明原因。

答：该系统不会因为竞争该类资源而死锁。这是因为，必有一个进程可获得 2 个资源，故能顺利完成，并释放出其所占用的 2 个资源给其他进程使用，让他们也能顺利完成。

7、在银行家算法中：若出现下面的资源分配情况：

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

请回答：

② 该状态是否安全？

②当进程 P2 提出请求 Request(1,2,2,2)后，系统能够将资源分配给它？（指优先分给 P2。。后系统还安不安全）

③如果系统立刻满足 P2 的上述请求，则系统是否立刻进入死锁状态？

答：①利用安全性算法对上面的状态进行分析，找到一个安全序列 {P0, P3, P4, P1, P2}，故系统是安全的。

	Work	Need	Allocation	Work + Allocation	Finish
	A B C D	A B C D	A B C D	A B C D	True
P0	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	True
P3	1 6 5 4	0 6 5 2	0 0 3 2	1 6 8 6	True
P4	1 6 8 6	0 6 5 6	0 0 1 4	1 6 8 10	True
P1	1 6 8 10	1 6 5 0	1 0 0 0	2 6 9 10	True
P2	2 6 9 10	2 3 5 6	1 3 5 4	3 9 14 14	True

②P2 发出请求向量 Request(1,2,2,2)后，按银行算法检查：

Request₂(1,2,2,2) ≤ Need₂(2,3,5,6)

Request₂(1,2,2,2) ≤ Available(1,6,2,2)

系统先假定可为 P2 分配资源，并修改 Available、Allocation₂ 和 Need₂：

Available = (0,4,0,0)

Allocation₂ = (2,5,7,6)

Need₂ = (1,1,3,4)

进行安全性检查：测试对于所有的进程，条件 Need₂ ≤ Available(0,4,0,0)都不成立，即 Available 不能满足任何进程要求，故系统进入不安全状态。因此进程 P2 提出请求 Request(1,2,2,2)，系统不能分配资源给它。

***③系统立即满足进程 P2 的请求(1,2,2,2)后，并没有马上死锁。因为，此时上述进程并没有申请新的资源而进入阻塞状态，只有当上述进程提出新的请求，导致所有未执行完的多个进程因得不到资源而阻塞并形成循环等待链时，系统才进入死锁状态。

7、有 2 个并发进程 P1, P2，其程序代码如下：（没有 P、V 操作？）

```

P1()
{
    X = 1;
    Y = 2;
    if(X > 0)
        Z = X + Y;
    else
        Z = X - Y;
    print Z;
}
P2()
{
    X = -1;
    A = X + 3;
    X = A + X;
    B = A + X;
    C = B * B;
    Print C;
}

```

①可能打印出的 Z 值有哪些？

②可能打印出的 C 值有哪些？（其中 X 为 P1，P2 的共享变量）

答：

① Z 是值有 3, -3, 5, 7。

② C 的值有 9, 25, 81。

8、假设有一个系统有 3 个抽烟者进程和 1 个供应者进程。每个抽烟者不停地卷烟并抽掉它，但是要卷起并抽掉 1 支烟需要 3 种材料：烟草、纸和胶水。三个抽烟者中，第一个拥有烟草，第二个拥有纸，第三个拥有胶水。供应者进程无限制地提供三种材料，供应者每次随机的将两种材料放到桌子上，拥有剩下那种材料的抽烟者卷一根烟并抽掉它，并给供应者一个信号告诉完成了，供应者会重新放两种材料上去，这个过程一直重复。

①试分析题目中的供应者和抽烟者之间存在的关系。

②试用信号量机制完成供应者和 3 个抽烟者的过程。

答：

供应者和 3 个抽烟者之间是同步关系（直接相互制约）。由于供应者无法同时满足两个或以上的抽烟者，因此 3 个抽烟者之间是互斥关系（间接相互制约）。

```
Int Random ;    //存随机数
```

```
Semaphore offer1; //定义信号量对应烟草和纸的组合
```

```
Semaphore offer2; //定义信号量对应烟草和胶水的组合
```

```
Semaphore offer3; //定义信号量对应纸和胶水的组合
```

```
Semaphore finish; //定义信号量表示抽烟是否完成
```

```
供应者:
```

```
While(1)
```

```
{
```

```
Random = 任意一个随机整数;
```

```
Random = random % 3;
```

```
If(Random == 0)
```

```
    V(offer1); //提供烟草和纸
```

```
else if(Random == 1)
```

```
    V(offer2); //提供烟草和胶水
```

```
else
```

```
    V(offer3); //提供纸和胶水
```

```
任意两种材料放在桌子上
```

```
P(finish);
```

```
}
```

```
抽烟者 1(拥有烟草者):
```

```
While(1)
```

```
{
```

```
    P(offer3);
```

```
    拿纸和胶水，卷成烟，抽掉
```

```
    V(finish);
```

```
}
```

```
抽烟者 2(拥有纸者):
```

```
While(1)
```

```
{
```

```
    P(offer2);
```

```
    拿烟草和胶水，卷成烟，抽掉
```

```

        V(finish);
    }
    抽烟者 3(拥有胶水者):
    While(1)
    {
        P(offer1);
        拿烟草和纸, 卷成烟, 抽掉
        V(finish);
    }

```

9、理发店有一位理发师、一把理发椅和 n 把供等候理发的顾客坐的椅子。如果没有顾客，理发师便在理发椅上睡觉，一个顾客到来时，顾客必须叫醒理发师，如果理发师正在理发时又有顾客到来，则如果有空椅子可坐，就坐下来等待，否则就离开。试用 P、V 操作实现理发师和顾客的同步，并给出信号量的定义和初始值。

答：

```

int waiting = 0; //等候理发的顾客数，初值为 0，进来一个顾客加 1，顾客理发时减 1
int charis = n; //为顾客准备的椅子数

semaphore customers = 0, barbers = 0, mutex = 1; //customers 用来记录等待理发师的
顾客数，并用作阻塞理发师进程，初值为 0; barbers 记录正在等候顾客的理发师数，并用作
阻塞顾客进程，初值为 0; mutex 用于互斥，初值为 1

```

```

barber()           //理发师进程
{
    while (1)
    {
        P(customers); //理完一人还有顾客吗，若无顾客理发师睡眠
        P(mutex);     //进程互斥
        waiting = waiting - 1; //等候顾客数少一个
        V(barbers);   //理发师去给一个顾客理发
        V(mutex);     //开放临界区
        cut_hair();   //正在理发
    }
}

customers()
{
    while (1)
    {
        P(mutex); //进程互斥
        if (waiting < charis) //如果有空的椅子就找到椅子坐下
        {
            waiting = waiting + 1; //等待顾客数加 1
            V(customers);           //唤醒理发师
            V(mutex);               //开放临界区
            P(barbers);             //无理发师，顾客坐着
            get_haircut();           //一个顾客坐下等待理发
        }
    }
}

```

```

    }
    else
        V(mutex);          //人满离开
    }
}

```

10、假设一个录像厅有 0, 1 和 2 三种不同的录像片由观众选择放映。录像厅的放映规则为:

(1) 任意时刻最多只能放映一部录像片, 正在放映的录像片是自动循环放映的, 最后一名观众主动离开时结束当前录像片的放映。

(2) 选择当前放映录像片的观众可以立即进入, 允许同时有多名观众选择同一录像片观看, 同时观看的人数不受限制。

(3) 等待观看其他录像片的观众可以按到达顺序排队, 当一种新的录像片开始放映时, 所有等待观看此录像片的观众可以依次进入录像厅同时观看。用一个进程代表一个观众。要求用信号量方法 P, V 操作写出同步活动的程序, 并给出信号量的定义和初始值。

答:

电影院一次只能放映一部影片, 希望观看的观众可能有不同的爱好, 但每次只能满足部分观众的需求, 即希望观看另外两部影片的用户只能等待。分别为三部影片设置三个信号量 S0, S1, S2, 初值为 1, 1, 1。电影院一次只能放一部影片, 因此需要互斥使用。由于观看影片的观众有多个, 因此必须分别设置三个计数器, 初值都是 0, 用来统计观众个数。当然计数器是个共享变量, 需要互斥使用。

```

semaphore s = 1, s0 = 1, s1 = 1, s2 = 1;
int count0 = 0, count1 = 0, count2 = 0;
videoshow0() //看第一部影片的观众
{
    P(s0);
    count0 = count0 + 1;
    if (count0 == 1)
    {
        P(s);
    }
    V(s0);
    看影片
    P(s0);
    count0 = count0 - 1;
    if (count0 == 0)
    {
        V(s);
    }
    V(s0);
}
videoshow1() //看第二部影片的观众
{

```

```

    P(s1);
    count1 = count1 + 1;
    if (count1 == 1)
    {
        P(s);
    }
    V(s1);
    看影片
    P(s1);
    count1 = count1 - 1;
    if (count1 == 0)
    {
        V(s);
    }
    V(s1);
}

videoshow2() //看第三部影片的观众
{
    P(s2);
    count2 = count2 + 1;
    if (count2 == 1)
    {
        P(s);
    }
    V(s2);
    看影片
    P(s2);
    count2 = count2 - 1;
    if (count2 == 0)
    {
        V(s);
    }
    V(s2);
}

```