选择题

1-5: C B D C B
6:10: A A D C B
11-15: C A B C A
16-20: C C C D C
21-25: D C A B A

综合题:
1:
（1）时刻 0：P1 到达，P1 运行
时刻 2：P1 还差 4，P2 到达
时刻 6：P1 运行完毕，P2 运行
时刻 8：P2 还差 10，P3 到达
时刻 9：P2 还差 9，P4 到达
时刻 18：P2 运行完毕，P3 运行
时刻 21：P3 运行完毕，P4 运行
时刻 28：P4 运行完毕
调度顺序：P1->P2->P3->P4
平均周转时间为(6+16+13+19)/4，即 13.5

进程	到达时刻	运行时间	开始运行时间	完成时间	周转时间
P1	0	6	0	6	6
P2	2	12	6	18	16
P3	8	3	18	21	13
P4	9	7	21	28	19

（2）时刻 0，P1 到达，开始运行
时刻 2，P2 到达，P1 还差 4，P2 抢占式优先级调度（优先级比 P1 高），P1 就绪，P2 运行
时刻 8，P3 到达，P2 还差 6，P3 抢占式优先级调度（优先级比 P2 高），P2 就绪，P3 运行
时刻 9，P4 到达，P3 还差 2，P4 无法抢占式优先级调度（优先级比 P3 低），P4 就绪，P3 继续运行。
时刻 11，P3 运行完毕，P2 运行
时刻 17，P2 运行完毕，P4 运行
时刻 24，P4 运行完毕，P1 运行
时刻 28，P1 运行完毕
进程调度顺序：P1->P2->P3->P2->P4->P1
平均周转时间:
(28+15+3+15)/4，即 15.25

进程	到达时刻	运行时间	优先级	开始/继续运行	暂停运行	剩余运行时间	完成时间	周周时间
P1	0	6	1	0	2	4		
P2	2	12	4	2	8	6		
P3	8	3	5	8			11	3
P2	2	12	4	11			17	15

P4	9	7	3	17			24	15
P1				24			28	28

（3）会选择抢占式优先级调度算法。因为短进程优先算法完全不考虑进程的优先级，无法保证紧迫型任务得到及时处理，另外短进程优先算法也很难确切的估计出进程的下一次运行时间。抢占式优先级算法总是调度优先级最高的进程先运行，能很好的保证紧迫型任务得到及时处理。

2：信号量 brigde 表示独木桥互斥访问,初值为 1;变量 eastcount 和 westcount 分别表示从东西方上独木桥的行人数量,初值为 0,信号量 eastmutex 和 westmutex 是对变量 eastcount 和 westcount 数据读写的保护，初值为 1，当某个方向上独木桥的人数大于 1 时，直接上桥，当该方向在桥上的人数为 0 时释放 bridge 锁。

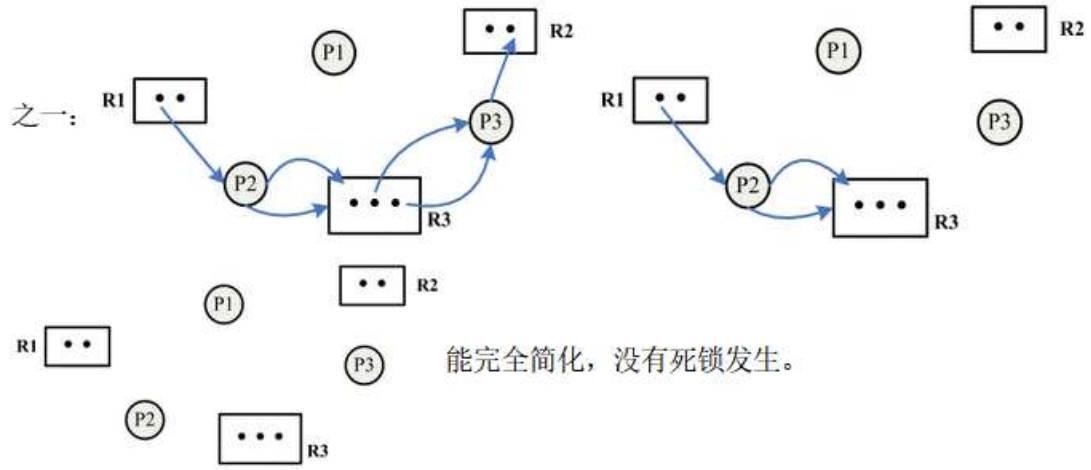
```
bridge = semaphore(1)
eastmutex = semaphore(1)
westmutex = semaphore(1)
var eastcount = 0
var westcount = 0

east:
    while(true):
        P(eastmutex)
        eastcount ++
        if(eastcount == 1):
            P(bridge)
        V(eastmutex)
        上桥
        下桥
        P(eastmutex)
        eastcount --
        if(eastcount == 0):
            V(bridge)
        V(eastmutex)

west:
    while(true):
        P(westmutex)
        westcount ++
        if(westcount == 1):
            P(bridge)
        V(westmutex)
        上桥
        下桥
        P(westmutex)
        westcount --
        if(westcount == 0):
            V(bridge)
        V(westmutex)
```

3：(1)进程的最大资源需求数减去当前进程已获得的资源数就是进程仍需的资源数。此时各个进程的仍需资源数向量为
P1:(0,0,0,0)
P2:(0,7,5,0)
P3:(6,6,2,2)
P4:(2,0,0,2)
P5:(0,3,2,0)

而系统的可用资源向量为(2,1,0,0),这时存在如下进程执行序列,可以使进程顺利执行完毕,所以该状态是安全的。
进程 可用资源数
P1 完成后: (2,1,1,2)
P4 完成后: (4,4,6,6)
P5 完成后: (4,7,9,8)
P2 完成后: (6,7,9,8)
P3 完成后: (6,7,12,12)
(2)在 P3 发出资源请求(0,1,0,0)后,假设系统把资源分配给 P3,则各进程已分配资源数为
P1:(0,0,1,2)
P2:(2,0,0,0)
P3:(0,1,3,4)
P4:(2,3,5,4)
P5:(0,3,3,2)
这时系统可用资源数为(2,0,0,0),各个进程仍需资源向量为
P1:(0,0,0,2)
P2:(0,7,5,0)
P3:(6,5,2,2)
P4:(2,0,0,2)
P5:(0,3,2,0)
满足资源需求的进程执行序列为
进程 可用资源数
P1 完成后: (2,0,1,2)
P4 完成后: (4,3,6,6)
P5 完成后: (4,6,9,8)
此时可用资源已不能满足 P2 或 P3 的需求,即此时系统状态是不安全的,系统将拒绝资源请求。
4:



5：
内存块大小=页面大小=4KB=2^12B,4GB 内存共被分为 2^32/2^12=2^20 个内存块。内存号至少要用 20bit 来表示，至少 3B 来表示块号。

6：1）由于索引表占用一个大小为 512B 的磁盘，所以该文件系统的索引表可以管理 512/3=170 个表项，而每一个表项对应一个物理块，
因此该文件系统可以支持的最大文件为： 170*512B=87040B=85K
能管理的最大磁盘空间：224*512B
2）若采用二级索引，则是：170*170*512B=7225KB
3）若采用三级索引，则是：170*170*170*512B=2456500KB=2398.93M

7：（1）虚拟地址 02A5=0000001010100101，低 11 位为页内地址，高位为页号，所以页号是 0，查页表得块号为 2，即 10B，所以物理地址是：0001001010100101，即 12A5H。

（2）物理地址 251D=0010010100011101，高位为块号，块号是 4，查页表，对应的页号是 2，所以虚拟地址是：0001010100011101，即 151DH。

（3）虚拟地址 2A3D=0010101000111101，页号为 5，不在内存，发生缺页中断，系统将把 5 号页读入内存，然后进行写操作；第二次读取这个地址的数据时，页面在内存，所以正常执行。

8：引入索引节点前：

目录文件需要占用磁盘块数：=（64*256）/512=32（块）

平均启动磁盘次数：=（32+1）/2=16.5 次

引入索引节点后：

符号目录项长度=（8+2）*256/512=5（块）

平均启动磁盘次数：=（5+1）/2+1=4 次