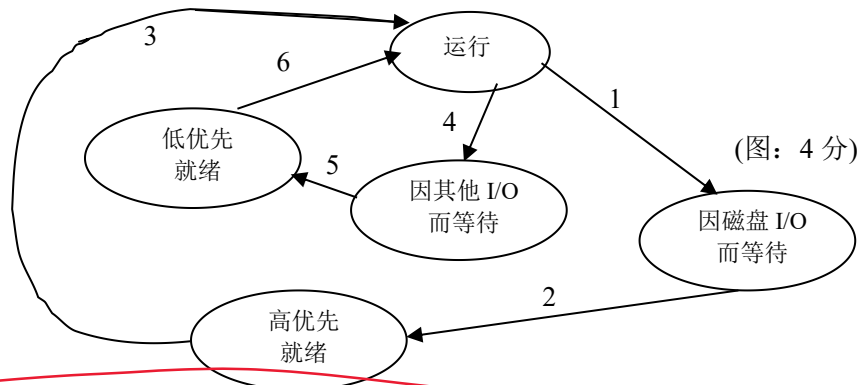


一. 8分
解答:



(图: 4分)

变迁 1: 请求磁盘 I/O 变迁 2: 磁盘 I/O 完成 (1分)

变迁 4: 请求其他 I/O 变迁 5: 其他 I/O 完成 (1分)

变迁 3: 当 CPU 空闲时, 首先从高优先就绪队列选择一个进程去运行 (1分)

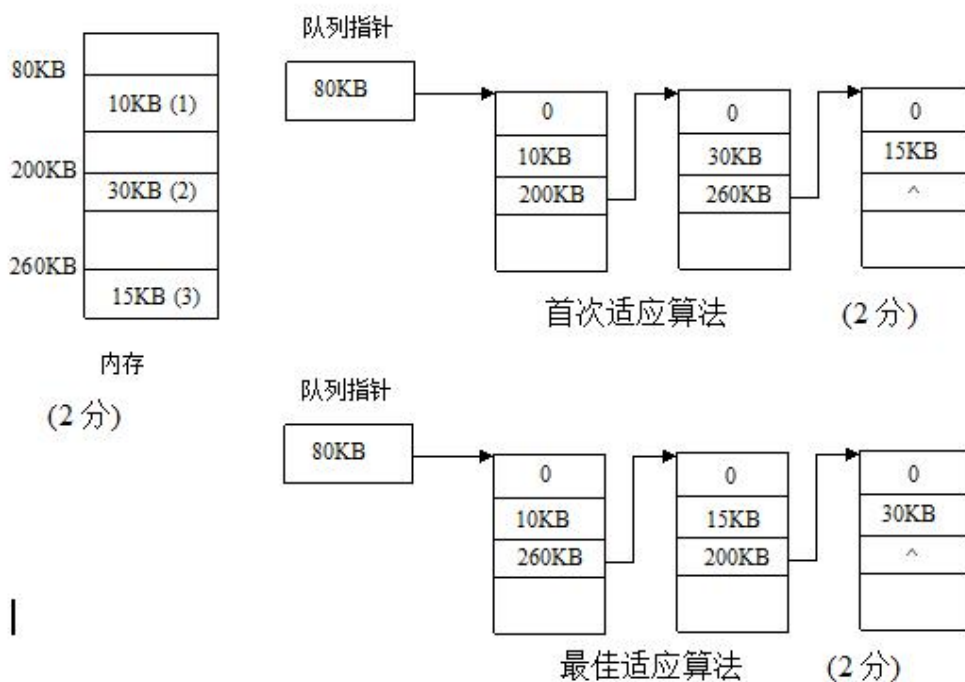
变迁 6: 当 CPU 空闲时, 高优先就绪队列为空, 则从低优先就绪选择一个进程去运行
(1分)

二、 (6分) (1) 不能, 可能发生死锁。

(2) 处于安全状态

P3→P2→P4→P1 或 P3→P4→P2→P1 或 P3→P2→P1→P4

三. 12分, 解答:



(3) 地址区间为(218K,230K), 5000+218K=228232, 各 2 分

(4) 基址 218K, 限长 12K, 0≤逻辑地址<12K, 各 2 分

四、(12 分)

main()

{ s₁: = 0; / P₂ 结束否 / (信号灯初值及意义共 2 分)

s₄: = 0; / P₄ 能否开始/

cobeging (程序结构共 2 分)

P₁ (); P₂ (); P₃ (); P₄ ();

coend

} (同步 8 分)

P₁ ()

P₂ ()

P₃ ()

P₄ ()

{

{

{

{

⋮

⋮

P (s₁);

P (s₄);

⋮

P (s₄);

V (s₄);

V (s₁);

V (s₄);

⋮

}

}

}

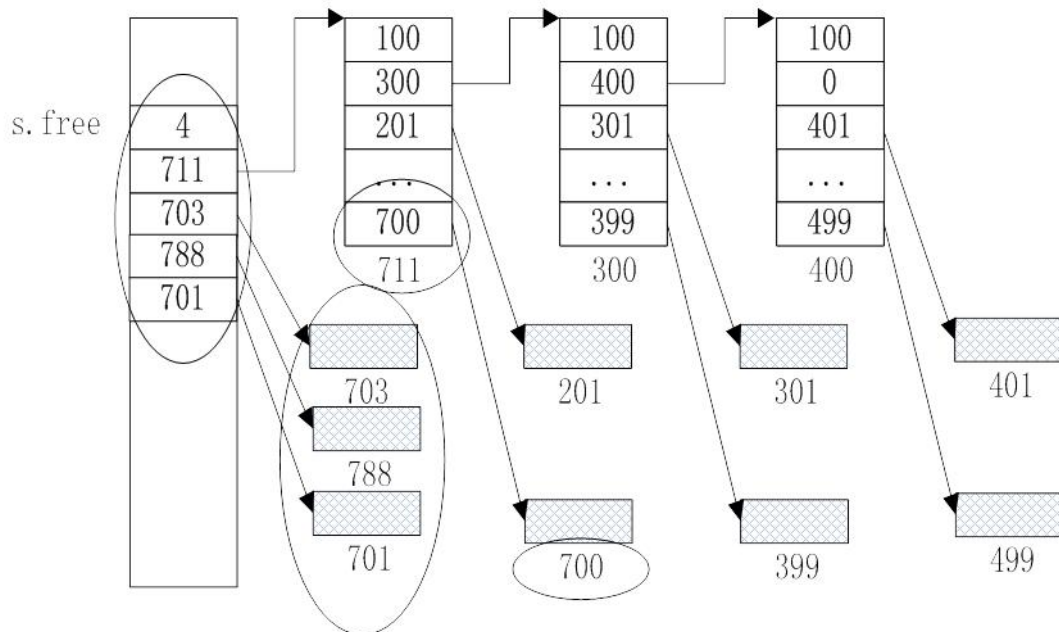
}

五、(12 分) (1) 299+2=301 块

(2) 100、200、299

(3)

空闲盘块号栈



六. (12分)

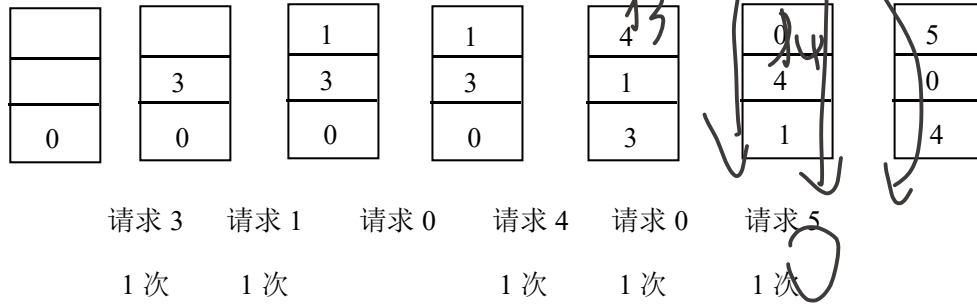
解答:

1. 对于以上的存储器引用序列, 给出其页面走向。

0, 3, 1, 0, 4, 0, 5

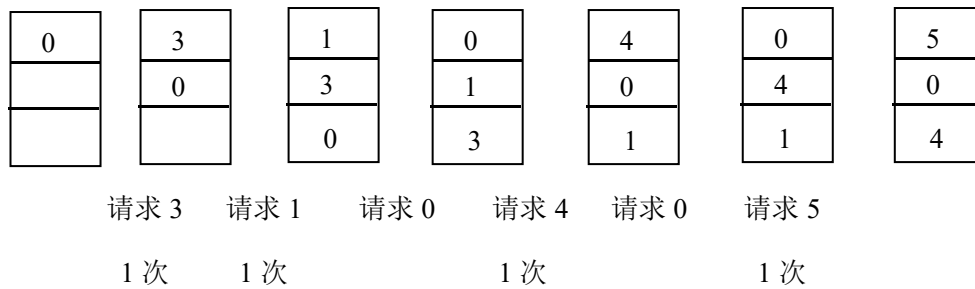
2.

先进先出页面置换算法



共 5 次

LRU 页面置换算法



共 4 次

七. (12分) 解答:

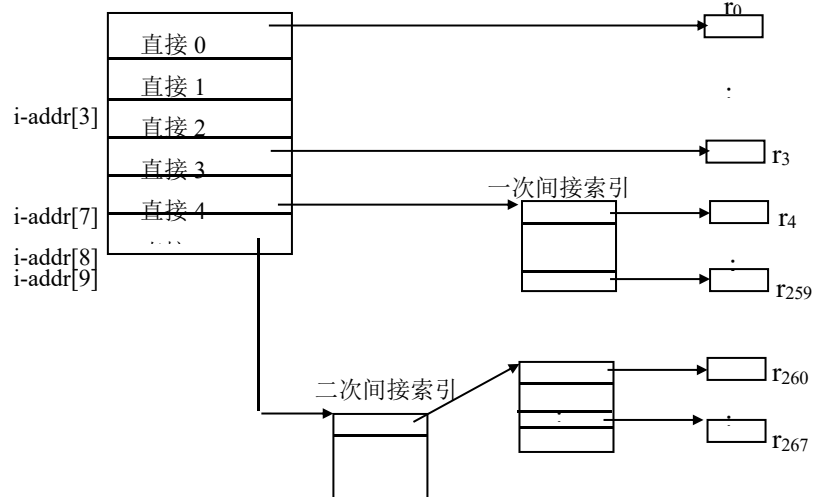
1. 该文件系统能构造的最大的文件

$$(4 + 256 + 256^2 + 256^3) * 512B = 16843012 * 512 = 8623622144B$$

(4分)

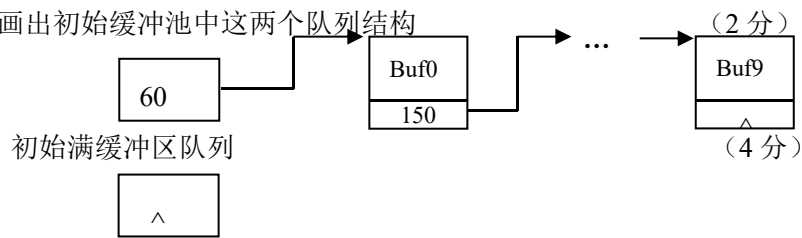
2. 文件 file 有 268 个记录 (大小为 512 字节), 试用图画出该文件的索引结构。

(8分)



八. (12 分) 解答:

1. 画出初始缓冲池中这两个队列结构



2.

main()

```
{  
    int s1=0; /* 表示满 buf 的数目 */  
    int s2=10; /* 表示空闲 buf 数目 */  
    mutex1=1; /* 满 buf 互斥信号灯 */  
    mutex2=1; /* 空闲 buf 互斥信号灯 */
```

cobegin

cp1(); cp2();

iop1(); iop2();

coend

}

计算进程对应的程序 cp()

```
{
```

while(计算未完成)

```
{
```

得到一个计算结果;

p(s2);

p(mutex2);

从空闲 buf 队列中摘下首元素;

v(mutex2);

将数送到该空闲缓冲区中;

p(mutex1);

将该 buf 插入满 buf 队列尾;

v(mutex1);

v(s1);

```
}
```

while(打印未完成)

```
{
```

p(s1);

p(mutex1);

从满 buf 队列中摘下首元素;

v(mutex1);

将该缓冲区中的数据输出打印;

p(mutex2);

将该 buf 插入空闲 buf 队列尾;

v(mutex2);

v(s2);

```
}
```

评分: 信号灯意义、初值: 4 分

程序结构与同步描述: 6 分

九. (6 分) SSTF: 访问次序 54-->66-->68-->38-->15-->99-->123-->125-->184

总磁道数: 12+ 2+ 30+ 23+ 84+ 24+ 2+ 59 = 236

电梯: 53-38-15-66-68-99-123-125-184

总磁道数: 16+23+51+2+32+24+2+59=209

十. (8 分) fork() pid1==0 exit() fork() pid2==0 exit() wait() wait()