

一、填空题

1. 中断
2. 在裸机上配置操作系统
3. 先来先服务
4. 等待状态
5. 环路条件
6. 解决重名问题
7. 引用位 改变位
8. 虚拟分配
9. 目录表目

二、单选题

1. B
2. A
3. D
4. D
5. A
6. C
7. B
8. D

三、简答题

1. 短作业优先调度算法进行调度

表 1

单位: 小时, 以十进制计

序号	提交时间	运行时间	开始时间	完成时间	周转时间	平均带权周转时间
1	8.00	1.00	8.00	9.00	1.00	1
2	8.20	1.20	9.80	9.8+1.2=11	11-8.2=2.8	2.8/1.2=2.33
3	8.30	0.30	9.00	9.0+0.3=9.3	9.3-8.3=1	1/0.3=3.33
4	8.50	0.50	9.30	9.3+0.5=9.8	9.8-8.5=1.3	1.3/0.5=2.6

平均周转时间 $t = (1 + 2.8 + 1 + 1.3) / 4 = 6.1 / 4 = 1.525$

平均带权周转时间 $w = (1 + 2.33 + 3.33 + 2.6) / 4 = 9.26 / 4 = 2.315$

2. 输入输出控制方式有哪些?

答: ①程序 I/O 方式; ②中断方式; ③DMA 方式; ④通道方式。

3. 答: 页面大小为 4KB $4KB = 2^{12}$ 12 位

允许用户虚地址空间最大为 16 页 $16 = 2^4$ 4 位

允许系统物理主存最多为 512 个主存块 $512 = 2^9$ 9 位

虚地址寄存器位数: $12 + 4 = 16$; 物理地址寄存器位数 $12 + 9 = 21$

四、答: 1. 该系统在时刻 t 的内存分布如图 1(a)所示。

2. 首次适应算法和最佳适应算法在时刻 t 的空闲队列结构如图 1(b)和图 1(c)所示。

3. (1) 首次适应算法:

作业 1 (38KB) 第 1 块 $100 - 38 = 62$

空闲区队列结构如图 2(a)所示。

- (2) 最佳适应算法:

作业 1 (38KB), 队列中的第一个元素 (大小为 36KB), 不能分配;

队列中的第二个元素 (大小为 60KB) $60 - 38 = 22$, 队列重新排列。

空闲区队列结构如图 2(b)所示。

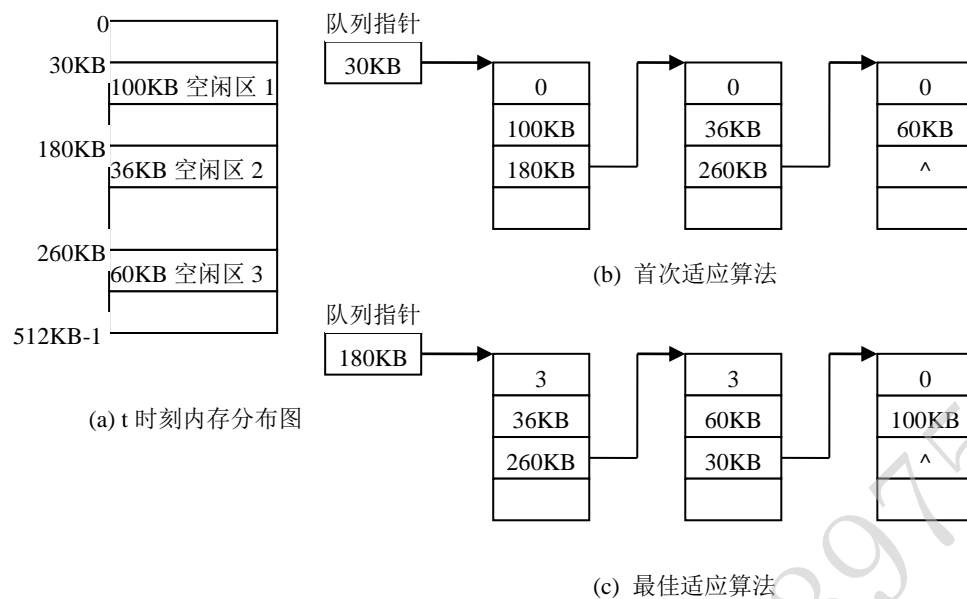


图 1 第四题 1, 2 的解答

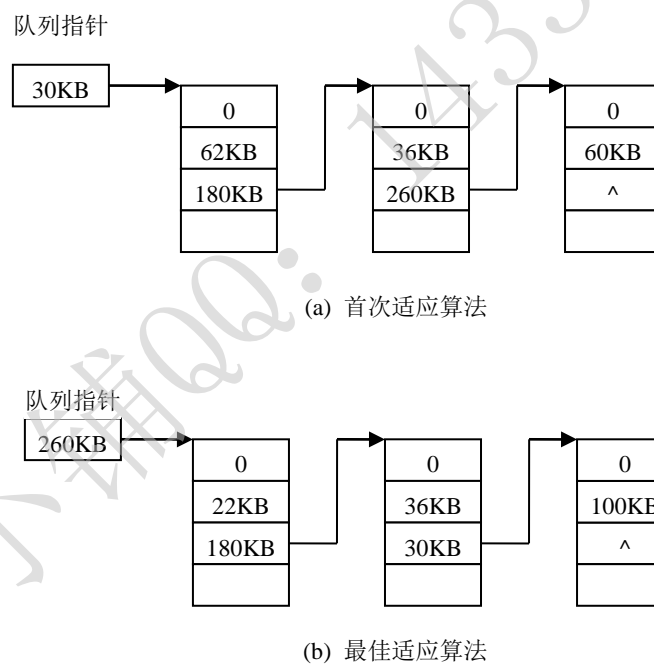


图 2 第四题 3 的解答

五、答：调度用进程状态变迁图如图 3 所示。

当 CPU 空闲时，首先从高优先就绪队列中选择队首元素去运行；若高优先就绪队列为空，则从中优先就绪队列中选择队首元素去运行；若中优先就绪队列为空，则从低优先就绪队列中选择队首元素去运行。

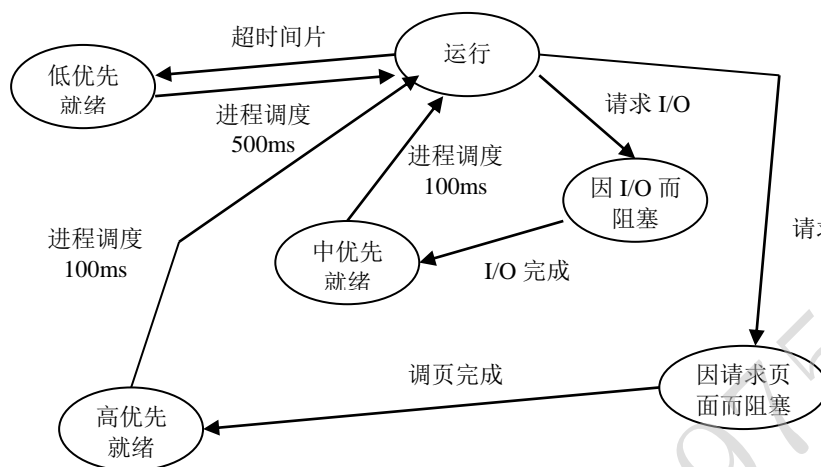


图 3 调度用进程状态变迁图

六、答:

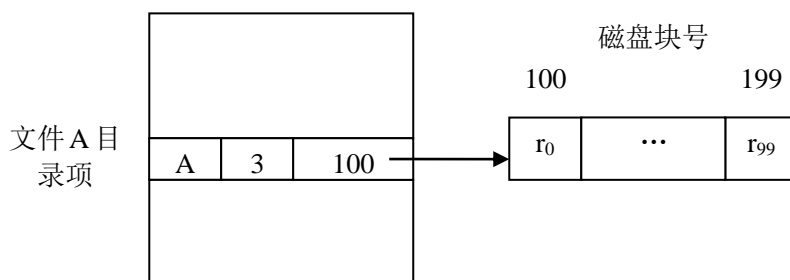
```
main(){
    s1 = 1; /* buf1 有无空位置*/
    s2 = 0; /* buf1 有无数据*/
    t1 = 1; /* buf2 有无空位置*/
    t2 = 0; /* buf2 有无数据*/
    cobegin
        p1(); /* 写进程*/
        p2(); /* 读进程*/
    coend
}
```

```
p1(){
    while(输入未完成){
        P(s1);
        数据放入 buf1 中;
        V(s2);
        P(t1);
        数据放入 buf2 中;
        V(t2);
    }
}
```

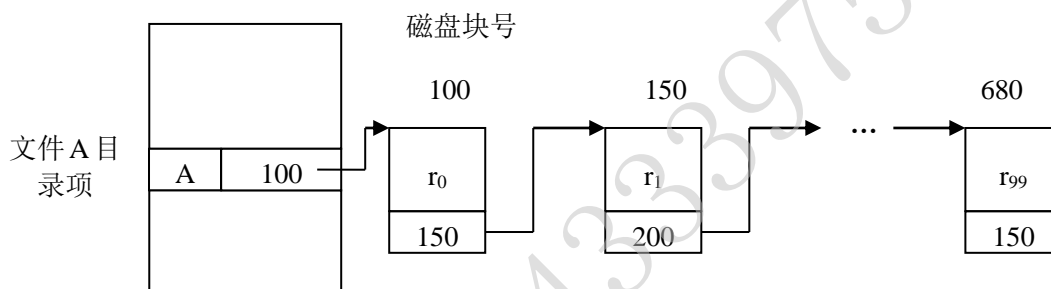
```
p2(){
    while(输出未完成){
        P(s2);
        数据从 buf1 中取出;
        V(s1);
        P(t2);
        数据从 buf2 中取出;
        V(t1);
    }
}
```

七、答:

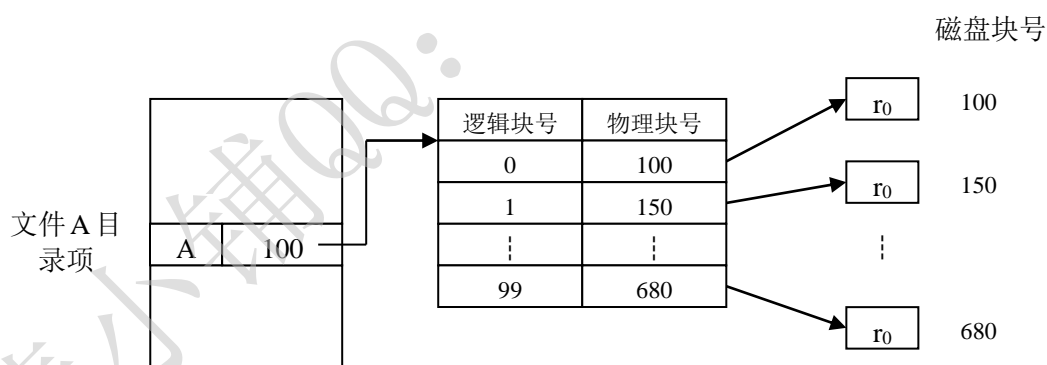
1. (1)连续文件



(2)串联文件



(3) 索引文件



2. (1) 连续文件

经过计算确定 r_7 记录的物理块号:

相对块号 $b = 1 \times r / \text{size} = 512 \times 7 / 512 = 7$;

r_0 所在的物理块号由文件目录项查得为 100,

r_7 的块号 $= 100 + 7 = 107$ 可直接读 107 块, 读 1 次

(2) 串联文件 读入 r_0 、 r_1 、...、 r_7 共读 8 次

(3) 索引文件 读入索引表 1 次,
查 r_7 所在物理块, 读该物理块 共读 2 次