武汉大学计算机学院

2009——2010 学年第二学期

《操作系统》试卷 A 参考答案

一．单项选择题（20 分，每题 1 分）

1．B　2．C　3．C　4．D　5．A　6．D　7．B　8．A
9．C　10．C　11．B　12．D　13．D　14．A　15．C
16．D　17．C　18．B　19．A　20．B

二、填空题（20 分，每空 1 分）

1. 存储器管理、设备管理、文件管理
2. 程序、数据、进程控制块
3. 进程同步、互斥
4. 抢占方式，非抢占方式
5. 合法、1054
6. 内部碎片、外部碎片
7. 控制器控制表、系统设备表、通道控制表
8. 连续分配、链接分配、索引分配

三、判断题（10 分，各 2 分）

对、错、错、对、错

四、 按照最短寻道时间优先算法，柱面的访问次序是：

15、16、13、9、20、24、29

最短寻道时间优先算法的柱面移动数为：1+3+4+11+4+5=28。（6分）

按照电梯调度算法，柱面的访问次序是：

15、16、20、24、29、13、9

电梯调度算法的柱面移动数为：1+4+4+5+16+4=34。（6分）

五、

（1）采用先来先服务调度算法时，5个任务在系统中完成时间及周转时间如下表所示。

作业	到达时间	运行时间	开始时间	完成时间	周转时间
A	1	8	1	9	8
B	3	6	9	15	12
C	4	3	15	18	14
D	5	5	18	23	18
E	7	10	23	33	26

根据表中的计算结果，5个进程的平均周转时间T为：

T=(8+12+14+18+26)/5=15.6min　（6 分）

（2）采用优先级调度算法时，5个任务在系统中的完成时间及周转时间如下表所示。

作业	到达时间	运行时间	优先级	开始时间	完成时间	周转时间
A	1	8	3	1	20	19
B	3	6	5	3	9	6
C	4	3	2	20	23	19
D	5	5	4	9	14	9
E	7	10	1	23	33	26

它们的平均周转时间为：

T=(19+6+19+9+26)/5= 15.8min　　6 分

六、状态安全结论（3 分）、安全序列（2 分）、检测过程（2 分）、无法分配（2 分）原因（1 分）

Need 值为：A　2　0　0　0

B　0　0　0　0

C　4　6　2　0

D　5　7　0　0

E　0　0　2　1

（1）利用安全性算法对此时刻的资源分配情况进行分析，可得到如下表所示的安全性检测情况。从中可以看出，存在安全序列 A、B、C、D、E，故该系统状态安全。

资源情况 / 进程	Work	Need	Allocation	Work+Allocation	Finish
A	2 6 2 1	2 0 0 0	3 6 2 0	5 12 4 1	true
B	5 12 4 1	0 0 0 0	1 0 2 0	6 12 6 1	true
C	6 12 6 1	4 6 2 0	1 0 4 0	7 12 10 1	true
D	7 12 10 1	5 7 0 0	0 0 0 1	7 12 10 2	true
E	7 12 10 2	0 0 2 1	5 3 4 1	12 15 14 3	true

（2）进程 D 提出申请（2，6，0，1），按银行家算法进行检查：

- $Request_D$（2，6，0，1）　≮ $Need_D$（5，7，0，0）
- 故申请不合法，此时系统不能将资源分配给 D。

七、10 分

```
semaphore  f1=f2=f3=f4=0;   1 分
    main()
    {  cobegin
       S1();    S2();    S3();    S4();    S5();     1 分
      coend  }
    S1()                                    下面共 8 分
    {    执行代码;
         v(f1);
```

```
    v(f1);
}
S2()
{   执行代码;
    v(f2);
}
S3()
{   p(f1);
    执行代码;
    v(f3);
}
S4()
{   p(f1);
    p(f2);
    执行代码;
    v(f4);
    }
S5()
{   p(f3);
    p(f4);
    执行代码;
    }
```

八、同步描述如下：6 分

解：

```
Semaphore  load=2;   //定义初值 1 分
Semaphore north=1;
Semaphore south=1;
main()
    {  cobegin
        tosouth();
tonorth();     1 分
       coend  }

tosouth()     2 分
{
    P(load);
    P(north);
    走过桥北半段到桥中央;
    V(north);
    P(south)
    走过桥南半段;
    V(south);
    V(load);
}

tonorth()              2 分
{
    P(load);
    P(south);
    走过桥南半段到桥中央;
    V(south);
    P(north)
    走过桥北半段;
    V(north);
    V(load);
}
```