第3章 处理机 作业

1、进程A、B、C、D、E需要的执行时间为10、6、20、7、3分钟，其中进程A要求必须在20分钟内完成，求5个进程的最短平均等待时间

EBADC （0+3+9+19+26）/5=11.4

2、在批处理系统中有以下5道作业，它们提交的时刻和作业长度由下表给出。计算在单道程序环境下，采用先来先服务调度算法和短作业优先算法的平均周转时间，并指出它们的调度顺序。

周转时间：

1: 8:45-8:00=45

2： 10:45-8:20=2:25=145

3：9:15-8:40=35

4：9:45-8:50=55

5：9:25-9:00=25

平均转换时间：(45+145+35+55+25)/5=81

调度顺序：1,3,5,4,2

|  |  |  |
| --- | --- | --- |
| 作业号 | 提交时刻 | 作业长度（分钟） |
| 1 | 8:00 | 45 |
| 2 | 8:20 | 60 |
| 3 | 8:40 | 30 |
| 4 | 8:50 | 20 |
| 5 | 9:00 | 10 |

3、单CPU实时系统中有3个周期任务，其中任务A每15秒执行一次，每次执行需要4秒；任务B每18秒执行一次，每次执行需要8秒；任务C每20秒执行一次，每次执行需要6秒。问是否存在一种合理的调度策略保证3个周期任务的正常执行？如果存在，需给出一种具体的调度策略；如果不存在，需给出理由。

单处理机 4/15+8/18+6/20>1所以不存在调度策略

4、实时系统中有2个周期任务。第一个任务每隔m1秒需要进行n1次运算；第二个任务每隔m2秒需要进行n2次运算。现有2种CPU可供选择，第一种CPU每秒能运算r1次运算，价格为c1；第二种CPU每秒能运算r2次，价格为c2。问如何配置最省钱？用C++代码实现。

两个周期平均每秒运算次数：n1/m1+n2/m2;

设k=L/r1

I为第一种CPU的数量

第二种CPU的数量为：(L-（r1\*i）)/r2

成本c=c1\*i+c2\*j

5、分析以下代码存在的问题

当A进行到Signal(resource\_2);时B进程可以进行Wait(resource\_2);

Wait(resource\_1);

这是AB都在等待出现死锁

Semaphore resource\_1 = 1, resource\_2 = 1;

process\_A()

{

Wait(resource\_1);

Wait(resource\_2);

Use\_both\_resources();

Signal(resource\_2);

Signal(resource\_1);

}

process\_B()

{

Wait(resource\_2);

Wait(resource\_1);

Use\_both\_resources();

Signal(resource\_1);

Signal(resource\_2);

}

6、P0请求Request0(0,1,0)，系统是否安全？

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 进程/资源 | Max | Allocation | Need | Available |
| A B C | A B C | A B C | A B C |
| P0 | 7 5 3 | 0 1 0 020 | 7 4 3  730 | 2 3 0  2 20 |
| P1 | 3 2 2 | 3 0 2 | 0 2 0 |  |
| P2 | 9 0 2 | 3 0 2 | 6 0 0 |  |
| P3 | 2 2 2 | 2 1 1 | 0 1 1 |  |
| P4 | 4 3 3 | 0 0 2 | 4 3 1 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **进程\资源** | **Work** | **Allocation** | **Need** | **Work+ Allocation** | **Finish** |
| A B C | A B C | A B C | A B C |
| **P1** | 2 2 0 | 3 0 2 | 0 2 0 | 5 2 2 | true |
| **P3** | 5 2 2 | 2 1 1 | 0 1 1 | 7 3 3 | true |
| **P4** | 7 3 3 | 0 0 2 | 4 3 1 | 7 3 5 | true |
| **P2** | 7 3 5 | 3 0 2 | 6 0 0 | 10 3 7 | true |
| **P0** | 10 3 7 | 0 2 0 | 7 4 3 | 10 5 7 | true |

资源总量：10 5 7 finish都为true

安全

7、银行家算法中，若某时刻资源分配情况如下图所示，问：资源总量为多少？当前时刻是否处于安全状态？若安全，说明原因并写出一种进程完成顺序；若不安全，说明原因并提出解决方案。 资源总量：11 4 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 进程/资源 | Max | Allocation | Need | Available |
| A B C | A B C | A B C | A B C |
| P0 | 8 3 3 | 5 1 2 | 3 2 1 | 2 0 3 |
| P1 | 1 3 1 | 1 1 1 | 0 2 0 |  |
| P2 | 3 1 1 | 2 1 0 | 1 0 1 |  |
| P3 | 2 3 4 | 0 0 4 | 2 3 0 |  |
| P4 | 6 1 1 | 1 1 0 | 5 0 1 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **进程\资源** | **Work** | **Allocation** | **Need** | **Work+ Allocation** | **Finish** |
| A B C | A B C | A B C | A B C |
| **P2** | 2 0 3 | 2 1 0 | 1 0 1 | 4 1 3 | true |
|  |  |  |  |  | true |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

执行到p2没有need小于413 所以不安全

8、用C++代码实现银行家算法

9、用C++代码实现付出代价最小的死锁解除算法