第 4 章　死锁与饥饿

4-1 什么是死锁？试举出一个生活中发生死锁的例子。

答：死锁是指在一个进程集合中的每个进程，都在等待只能由该组进程中的其他进程才能引发的事件，从而无限期僵持下去的一种局面。如果没有外力作用，它们都将无法再向前推进。因此，发生了死锁的进程陷入了一种永久性的阻塞态。

例如，独木桥问题，两个人从桥的两端同时上桥，造成死锁。

4-2 计算机系统中发生死锁的根本原因是什么？

答：资源有限、进程推进速度不当。

4-3 发生死锁的 4 个必要条件是什么？

答：互斥条件、请求且保持条件、不可抢占条件、循环等待条件。

4-4 解决死锁的常用方法有哪 4 种？

答：死锁预防、死锁避免、死锁检测与接触、死锁忽略。

4-5 死锁预防的基本思想是什么？

答：死锁预防是通过设置某些限制条件，破坏产生死锁的 4 个必要条件的一个或几个条件来预防死锁。

4-6 死锁避免的基本思想是什么？

答：死锁避免是在资源的动态分配过程中，用某种方法防止系统进入不安全状态，避免发生死锁。

4-7 什么是进程的安全序列？何谓系统处于安全状态？

答：安全状态是指系统能按某种进程推进顺序，为每个进程分配所需资源，直至满足每个进程对资源的最大需求，使每个进程都顺利完成。若能找到这样的进程推进顺序，则称其为安全序列。

4-8 设系统中有 5 个进程 P1、P2、P3、P4 和 P5，有 3 种类型的资源 A、B 和 C，其中资源 A 的数量是 17，资源 B 的数量是 5，资源 C 的数量是 20。T0 时刻，系统状态如表 4-9 所示。

表 4-9　习题 4-8 表

进程	资源情况								
	已分配资源数			最大资源需求			可用资源数		
	A	B	C	A	B	C	A	B	C
P1	2	1	2	5	5	9			
P2	4	0	2	5	3	6			
P3	4	0	5	4	0	11			
P4	2	0	4	4	2	5			
P5	3	1	4	4	2	4			

（1）T0 时刻系统是否处于安全状态，为什么？

（2）T0 时刻，进程 P2 提出资源请求[0, 3, 4]，是否实施资源分配，为什么？

（3）T0 时刻，进程 P4 提出资源请求[2, 0, 1]，是否实施资源分配，为什么？

答：系统的可用资源及每个进程已分配资源和尚需的资源情况见表 1 所示。

表 1　T0 时刻系统的状态

进程	资源情况											
	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P1	5	5	9	2	1	2	3	4	7	2	3	3
P2	5	3	6	4	0	2	1	3	4			
P3	4	0	11	4	0	5	0	0	6			
P4	4	2	5	2	0	4	2	2	1			
P5	4	2	4	3	1	4	1	1	0			

（1）T0 时刻的安全性：利用安全性算法对 T0 时刻的资源分配情况进行分析，如表 2 所示。因为，在 T0 时刻存在一个安全序列{P4，P5，P1，P2，P3}，所以，T0 时刻系统处于安全状态。

表 2　T0 时刻的安全性检查

进程	Work			Need			Allocation			Work+Allocation			Finish
	A	B	C	A	B	C	A	B	C	A	B	C	
P4	2	3	3	2	2	1	2	0	4	4	3	7	true
P5	4	3	7	1	1	0	3	1	4	7	4	11	true
P1	7	4	11	3	4	7	2	1	2	9	5	13	true
P2	9	5	13	1	3	4	4	0	2	13	5	15	true
P3	13	5	15	0	0	6	4	0	5	17	5	20	true

（2）进程 P2 发出资源请求 $Request_2(0, 3, 4)$后，系统按银行家算法进行检查：

①$Request_2(0, 3, 4) \leq Need_2(1, 3, 4)$；

②$Request_2(0, 3, 4) \not\leq Available(2, 3, 3)$，即系统没有足够的可用资源供其使用。因此，进程 P2 的资源请求无法满足，故 P2 阻塞。

（3）进程 P4 发出资源请求 $Request_4(2, 0, 1)$后，系统按银行家算法进行检查：

①$Request_4(2, 0, 1) \leq Need_4(2, 2, 1)$；

②$Request_4(2, 0, 1) \leq Available(2, 3, 3)$；

③系统假设满足 P4 的请求，为其分配所需资源，修改后的数据结构如下所示：

$Available = (2, 3, 3) - (2, 0, 1) = (0, 3, 2)$

$Need_4$= (2, 2, 1) - (2, 0, 1) = (0, 2, 0)

$Allocation_4$= (2, 0, 4) + (2, 0, 1) = (4, 0, 5)

④利用安全性算法检测状态是否安全，如表 3 所示。

表 3　P4 申请资源时的安全性检查

进程	Work			Need			Allocation			Work+Allocation			Finish
	A	B	C	A	B	C	A	B	C	A	B	C	
P4	0	3	2	0	2	0	4	0	5	4	3	7	true
P2	4	3	7	1	3	4	4	0	2	8	3	9	true
P3	8	3	9	0	0	6	4	0	5	12	3	14	true
P5	12	3	14	1	1	0	3	1	4	15	4	18	true
P1	15	4	18	3	4	7	2	1	2	17	5	20	true

由安全性检查得知，可以找到一个安全序列{ P4，P2，P3，P5，P1}。因此，系统是安全的。所以，可以将 P4 所请求的资源分配给它。

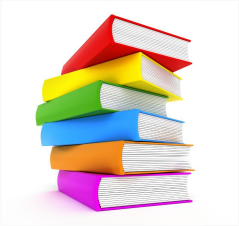
4-9 考虑由 n 个进程共享的具有 m 个同类资源的系统，如果满足下面两个条件：

（1）对 i=1, 2, 3, …, n，进程 Pi 至少需要 1 个资源，最多需要 m 个资源；

（2）在任意时刻，所有进程对资源的需求量之和小于 m+n。

试证明：该系统不会发生死锁。

证明：设每个进程对共享资源的最大需求量为 x(0＜x≤m)，由于每个进程最多申请使用 x 个资源，在最坏的情况下，每个进程都得到了(x-1)个资源并且都需申请最后一个资源。这时，系统剩余资源数为 m-n(x-1)。只要系统还有一个资源可用，就可使其中的一个进程获得所需的全部资源。该进程运行结束后释放出它所占用的资源，其他进程的资源需求也可全部得到满足。因此，当 m-n(x-1)≥1 时，即 x≤(m+n-1)/n 时系统不会发生死锁。进而可得系统中所有进程的最大需求量之和 n·x≤(m+n-1)，即所有进程最大需求量之和小于 m+n 时，系统不会发生死锁。所以，该系统是死锁无关的。题目得证。



信号量练习题

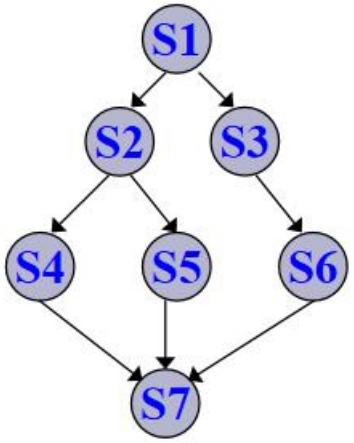
沈晓红　教授
山东财经大学

思考题

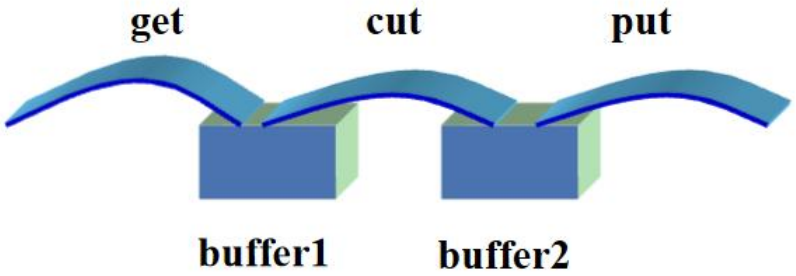
- 1.某单位有200名员工，员工请假需要领导签字，领导在办公室签字，要求每次至多只允许一名员工进入。请用信号量机制实现员工找领导签字要求。
- 2.某单位有200名员工，每个员工都有资格参加员工座谈会，若某次座谈会使用的会议室最多容纳20名员工，请用信号量机制实现员工参加座谈会同步互斥模型。
- 3.某售票厅最多容纳100人购票，请写出购票的同步互斥模型。
- 4.某超市最多容纳100人购物，每位顾客在进入超市时去购物篮区拿一个购物篮，离开超市时再去购物篮区还购物篮，超市的购物篮区每次只允许一个人进入。请用信号量实现。

思考题

- 5.试写出相应的程序来描述右图所示的前趋关系。



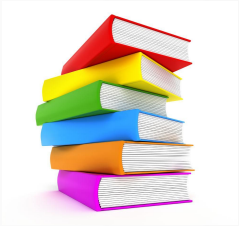
- 6.有三个进程，进程get反复执行从输入设备上读数据存入buffer1;进程cut反复执行将buffer1的内容剪切到缓冲区buffer2，进程put反复执行将buffer2的内容输出在打印机上。三个进程并发执行，协调工作。写出该三个进程并发执行的同步模型。



思考题

- 7.桌上有一只盘子,每次只能放入一个水果.爸爸专向盘中放苹果,妈妈专向盘中放桔子,女儿专等吃盘中的苹果,儿子专等吃盘中的桔子.试用PV操作写出他们能同步的程序。
- 8.用PV操作解决司机和售票员的同步问题。

```
司机进程:                    售票员进程:
while (true)                 while (true)
{                            {
    启动车辆                     关门
    正常驾驶                     售票
    到站停车                     开门
    …                            …
}                            }
```



信号量练习题答案

沈晓红　教授
山东财经大学

思考题

- 1.某单位有200名员工，员工请假需要领导签字，领导在办公室签字，要求每次至多只允许一名员工进入。请用信号量机制实现员工找领导签字要求。

```
semaphore s=1;
employee( )
{           ...
            wait(s);
            找领导请假、签字;
            signal(s);
            ...
}
```

思考题

- 2.某单位有200名员工，每个员工都有资格参加员工座谈会，若某次座谈会使用的会议室最多容纳20名员工，请用信号量机制实现员工参加座谈会同步互斥模型。

```
semaphore s=20;
employee( )
{           ...
            wait(s);
            座谈;
            signal(s);
            ...
}
```

思考题

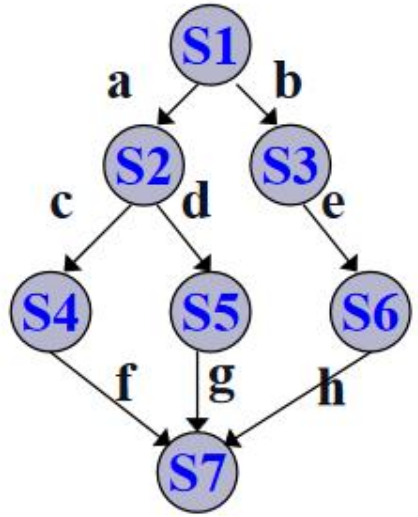
- 3.某售票厅最多容纳100人购票，请写出购票的同步互斥模型。

```
semaphore s=100;
customer( )
{           ...
            wait(s);
            购票;
            signal(s);
            ...
}
```

4.某超市最多容纳100人购物，每位顾客在进入超市时去购物篮区拿一个购物篮，离开超市时再去购物篮区还购物篮，超市的购物篮区每次只允许一个人进入。请用信号量实现。

```
semaphore room=100，basket=1;
customer( )
{           ...
            wait(room);
            wait(basket);
            拿篮子;
            signal(basket);
            购物
            wait(basket);
            还篮子;
            signal(basket);
            signal(room);
            ...
}
```

▶5.试写出相应的程序来描述右图所示的前趋关系。



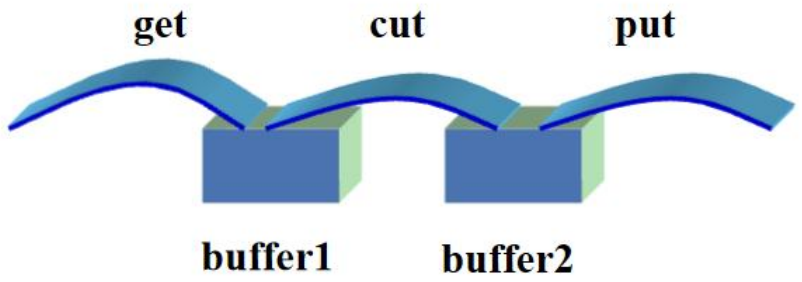

```
semaphore a=0,b=0,c=0,d=0,e=0,f=0,g=0,h=0;

S1( ) { S1;  Signal(a);   Signal(b);  }
S2( ) { Wait(a);   S2;   Signal(c);   Signal(d);  }
S3( ) { Wait(b);   S3;   Signal(e);  }
S4( ) { Wait(c);   S4;   Signal(f);  }
S5( ) { Wait(d);   S5;   Signal(g);  }
S6( ) { Wait(e);   S6;   Signal(h);  }
S7( ) { Wait(f);   Wait(g);   Wait(h);   S7;  }

main()
{  parbegin(S1,S2,S3,S4,S5,S6,S7);  }(main函数部分写不写均可)
```

- 6.有三个进程，进程get反复执行从输入设备上读数据存入buffer1;进程cut反复执行将buffer1的内容剪切到缓冲区buffer2，进程put反复执行将buffer2的内容输出在打印机上。三个进程并发执行，协调工作。写出该三个进程并发执行的同步模型。

```
semaphore empty1=1,full1=0,empty2=1,full2=0;
get( )
{           while(true)
            {           从外部读数据；
                        Wait(empty1)；
                        存入buffer1;
                        Signal(full1);
            }
}
 cut( )
{           while(true)
            {           Wait(full1);
                        Wait(empty2);
                        将数据从buffer1剪切到buffer2;
                        Signal(empty1);
                        Signal(full2);
             }
}
put( )
{           while(true)
            {           Wait(full2);
                         从buffer2读数据送入打印机;
                        Signal(empty2);
            }
 }
```



7.桌上有一只盘子,每次只能放入一个水果.爸爸专向盘中放苹果,妈妈专向盘中放桔子,女儿专等吃盘中的苹果,儿子专等吃盘中的桔子.试用PV操作写出他们能同步的程序。

```
semaphor empty=1,apple=0,orange=0;
father( )
{   while(true)
    {   wait(empty);
        向盘中放苹果;
        signal(apple);
    }
 }
mather( )
{   while(true)
    {   wait(empty);
        向盘中放桔子;
        signal(orange);
    }
 }
```

```
daughter( )
{   while(true)
    {   wait(apple);
        吃苹果;
        signal(empty);
    }
 }
son( )
{   while(true)
    {   wait(orange);
        吃桔子;
        signal(empty);
    }
 }
```

8.用PV操作解决司机和售票员的同步问题。

```
semaphore close=0,open=0;
driver( )
{   while(true)
    {   Wait(close)
        启动车辆;
        正常行驶;
        到站停车;
        Signal(stop)
    }
}
```

```
seller( )
{     while(true)
      { 关门
        Signal(close)
        售票
        Wait(stop)
        开门
      }
}
```

```
司机进程:
while (true)
{
    启动车辆
    正常驾驶
    到站停车
    …
}
```

```
售票员进程:
while (true)
{
    关门
    售票
    开门
    …
}
```

练习

- 已知某分页虚拟内存系统的用户空间共32个页面，主存容量为64K，页面大小为1K。对一个4页大的作业，其0、1号页分别被分配到主存的2、4物理块中。求出逻辑地址1023、2500、4500、064CH、0E4CH对应的物理地址。
- 用户空间32个页面：逻辑地址的有效位是15位
- 主存容量64K：物理地址16位

根据题意，页表为：

页号	物理块号
0	2
1	4

因为页长为1K，所以地址变换过程如下：

1)逻辑地址1023

逻辑地址1023分页后，页号为0，页内地址为1023。

查页表，0号页存放于2号物理块，所以物理地址为 2×1024+1023=3071。

2)逻辑地址2500

逻辑地址2500分页后，页号为2，页内地址为452。

查页表，2号页未调入内存，故产生缺页中断。

3)逻辑地址4500

逻辑地址4500分页后，页号为4，页内地址为404。

因为该作业长度为4个页，没有4号页，故产生越界中断。

页号	物理块号
0	2
1	4

4) 页长1K，所以高位表示页号，低10位表示页内地址。

064CH： 0000 0110 0100 1100

页号为1，页内地址为二进制数10 0100 1100

查页表，1号页存于4号物理块，故物理地址为：

0001 0010 0100 1100 ，即124CH

5) 页长1K，所以高位表示页号，低10位表示页内地址。

0E4CH： 0000 1110 0100 1100

页号为3，未越界，查页表，该页未调入内存，故产生缺页中断。

- 已知进程的虚拟地址为20位二进制地址，其中高8位为段号，低12位为段内相对地址。

1)一个作业最多可以有多少段?每段的最大长度为多少字节?

2)若段表如下，对逻辑地址[0,100]、[1,650]、[2,30]、[3,100]进行地址变换。

段号	段长	主存起始地址	是否在主存
0	200	2048	是
1	300	2800	是
2	100	—	否

1) 2^8 , 2^{12}

2)逻辑地址[0,100]重定位后物理地址为2048+100=2148

[1,650]重定位时，段内地址越界，产生越界中断

[2,30]重定位时该段未调入内存，产生缺段中断

[3,100]重定位时段号非法，产生越界中断

第 5 章　处理机调度

5-1 处理机调度一般可分为哪三个层次？其中哪一级调度必不可少？

答：处理机调度按照层次可以分为三级：高级调度、中级调度和低级调度。其中，低级调度必不可少。

5-2 高级调度与低级调度的主要任务是什么？为什么要引入中级调度？

答：高级调度负责从后备队列中选择多个作业调入内存，为它们创建进程并分配必要的资源，然后链接到就绪队列上。

低级调度负责按照某种调度算法，从内存的就绪队列中选择一个就绪进程获得 CPU，并分派程序执行进程切换的具体操作。

引入中级调度的主要目的是为了缓解内存压力，提高内存利用率和系统吞吐量，使那些暂时不能运行的进程不再占用内存资源，将它们调至外存等待，把进程状态改为挂起状态。

5-3 何谓静态优先级和动态优先级？确定静态优先级的依据是什么？

答：静态优先级是在创建进程时确定的，且在进程的整个生命周期内保持不变。动态优先级是指在创建进程时确定的优先级，可以随进程的推进或进程等待时间的增加而改变，以便获得更好的调度性能。

通常，根据进程类型、进程对资源的需求、用户要求等确定一个进程的静态优先级。

5-4 试比较 FCFS、SPF、HRRN 三种调度算法。

答：FCFS 调度算法是按照作业或进程到达系统的先后次序进行调度。SPF 调度算法是优先选择短进程投入运行。也即 FCFS 调度算法只考虑了每个作业的等待时间而未考虑执行时间，SPF 调度算法只考虑了执行时间而未考虑等待时间。

HRRN 调度算法同时考虑每个作业的等待时间和要求服务时间，从中选出响应比最高的作业投入执行。（1）若等待时间相同，则要求服务时间越短，其响应比越高，因而 HRRN 调度算法有利于短作业。（2）若要求服务时间相同，则等待时间越长，其响应比越高，此时 HRRN 调度算法等价于 FCFS 调度算法。（3）对于长作业，响应比可以随着等待时间的增加而相应地提高，使长进程获得 CPU 的可能性逐步提高，从而避免了饥饿现象。因此，HRRN 调度算法既照顾了短作业，又不会使长作业的等待时间过长，有效提高了调度的公平性。

5-5 为什么说 FB 调度算法能较好地满足各方面用户的需要？

答：对终端型作业用户而言，由于他们所提交的大多属于交互型作业，作业通常比较短小，系统只要能使这些作业在第 1 级队列所规定的时间片内完成，便可使终端型作业用户感到满意；对于短批处理作业用户而言，他们的作业开始时像终端型作业一样，如果仅在第 1 级队列中执行一个时间片即可完成，便可以获得与终端型作业一样的响应时间，对于稍长的作业，通常也只需要在第 2 级队列和第 3 级队列中各执行一个时间片即可完成，其周转时间仍然较短；对于长批处理作业用户而言，它们的长作业将依次在第 1，2，…，直到第 n 级队列中运行，

然后再按时间片轮转方式运行，用户不必担心其作业长期得不到处理。故 FB 调度算法能较好地满足各种类型用户的需要。

5-6 什么是 EDF 调度算法？并举例说明。

答：EDF 调度算法是最早截止时间优先调度算法。该算法是根据任务的开始截止时间确定的任务优先级调度算法。截止时间越早则优先级越高。该算法要求在系统中保持一个实时任务就绪队列，该队列按各任务截止时间的先后排序。例如，任务 A 的开始截止时间和所需的运行时间分别为 200ms 和 100ms，任务 B 的开始截止时间和所需的运行时间分别为 500ms 和 250ms。按照 EDF 调度算法，因为任务 A 的开始截止时间早于任务 B 的开始截止时间，故调度任务 A 先执行。

5-7 什么是 LLF 调度算法？并举例说明。

答：LLF 调度算法是最低松弛度优先调度算法。该算法是根据任务的紧急（或松弛）程度，来确定任务的优先级。任务的紧急程度越高，为该任务所赋予的优先级就越高，以使之优先执行。任务 A 的完成截止时间和所需的运行时间分别为 400ms 和 250ms，则该任务的松弛度为 150ms；任务 B 的完成截止时间和所需的运行时间分别为 300ms 和 100ms，则该任务的松弛度为 200ms。根据 LLF 调度算法，系统优先调度任务 A 执行。

5-8 假设 4 个作业到达系统的时间及所需服务时间如表 5-8 所示。

表 5-8 习题 5-8 表

作业	到达时刻/ms	所需服务时间/ms
J1	0	20
J2	5	15
J3	10	5
J4	15	10

写出采用 FCFS、SPF、HRRN 调度算法时，各作业的执行顺序、周转时间、带权周转时间以及作业流的平均周转时间和平均带权周转时间。

答：（1）采用 FCFS 算法时，先到达系统的作业先调度执行，所以调度顺序为 J1、J2、J3、J4。

作业名	到达时间/ms	服务时间/ms	开始执行时间/ms	完成时间/ms	周转时间/ms	带权周转时间
J1	0	20	0	20	20	1
J2	5	15	20	35	30	2
J3	10	5	35	40	30	6
J4	15	10	40	50	35	3.5
平均值					28.75	3.125

（2）采用 SJF 算法时，优先调度到达的短作业，所以调度顺序为 J1、J3、J4、J2。

作业名	到达时间/ms	服务时间/ms	开始执行时间/ms	完成时间/ms	周转时间/ms	带权周转时间
J1	0	20	0	20	20	1
J2	5	15	35	50	45	3
J3	10	5	20	25	15	3
J4	15	10	25	35	20	2
平均值					25	2.25

（3）采用 HRRN 算法时，优先调度高响应比作业。

首先调度 J1。J1 的周转时间为 20ms，带权周转时间为 1。

然后，因为 R2=(35-5)/15=2， R3=(25-10)/5=3， R4=(30-15)/10=1.5，所以再调度 J3。J3 的周转时间为 25-10=15ms，带权周转时间为 15/5=3。

之后，因为 R2=(40-5)/15=2.33， R4=(35-15)/10=2，所以调度 J2 执行。J2 的周转时间为 40-5=35ms，带权周转时间为 35/5=7。

最后调度 J4 执行。J4 的周转时间为 50-15＝35ms，带权周转时间为 35/10=3.5。

综上，调度顺序为 J1、J3、J2、J4。

平均周转时间为(20+15+35+35)/4=26.25ms，平均带权周转时间为(1+3+7+3.5)/4=3.625。

5-9 假设系统几乎同时有四个进程 P1、P2、P3 和 P4 同时到达，进程的优先级及所需服务时间如表 5-9 所示。

表 5-9 习题 5-9 表

进程	优先级	所需服务时间/ms
P1	2	4
P2	5	3
P3	4	5
P4	3	2

考虑如下调度算法下，各进程的调度顺序和周转时间。

（1）最高优先级调度算法；

（2）RR 调度算法（q=1）。

答：（1）最高优先级调度算法时，调度顺序为 P2、P3、P4、P1。

P2 的周转时间为 3ms；

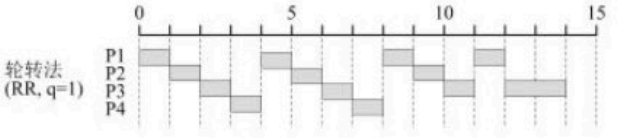
P3 的周转时间为 3+5=8ms；

P4 的周转时间为 3+5+2=10ms；

P1 的周转时间为 3+5+2+4=14ms；

平均周转时间为（3+8+10+14）/4=8.75ms

（2）采用 RR 轮转法（q=1）时，调度顺序如下图所示。



P1 的周转时间为 12ms；

P2 的周转时间为 10ms；

P3 的周转时间为 14ms；

P4 的周转时间为 8ms；

平均周转时间为（12+10+14+8）/4=11ms