第 3 章 互斥与同步

8. 设有 N 个进程，共享一个资源 R，但每个时刻只允许一个进程使用 R。算法如下：设置一个整型数组 flag[N]，其每个元素对应表示一个进程对 R 的使用状态，若为 0 表示该进程不在使用 R，为 1 表示该进程要求或正在使用 R，所有元素的初值均为 0。

```
process Pi
{
 ……
 flag[i] = 1;
 for (j=0; j<i; j++)
    do while (flag[j]) ;
 for (j=i+1; j<N; j++)
    do while (flag[j]) ;
 use resource R ;
 flag[i] = 0;
 ……
}
```

试问该算法能否实现上述功能？为什么？若不能，请用 P、V 操作改写上述算法。

答：不能实现，因为标志置为 1 与检测其他进程的标志之间可能被中断。用 P、V 操作改写后为

```
Semaphore s=1;
process Pi
{
 ……
 P(s);
 use resource R ;
 V(s);
 ……
}
```

9. 有三个进程 R、M、P，R 负责从输入设备读入信息并传送给 M，M 将信息加工并传送给 P，P 将打印输出，写出下列条件下的并发进程程序描述。

(1) 一个缓冲区，其容量为 K；

(2) 两个缓冲区，每个缓冲区容量均为 K。

答：

(1) 一个缓冲区。

```
semaphore sr,sm,sp;
int lr, lm, lp;
Datatype B[K];
lr=lm=lp=0;
sr.value=K ;sm.value=0;sp.value=0;
cobegin
repeat R;
repeat M;
repeat P;
```

```
coend
Process R
{
      read data;
      P(sr);
      B[lr]=data;
      lr=(lr+1)%K;
     V(sm);
}
Process M
   {
       P(sm);
       if ( B[lm]== ' ')
             B[lm]= ';';
       lm=(lm+1)%K;
       V(sp);
 }
Process P
  {
       P(sp);
       data = B[lp];
       lp=(lp+1)%K;
       V(sr);
       print data;
    }
```

(2) 两个缓冲区。

方案一：将缓冲区视为 2K，sr 的初始值为 2K，其他与第一问相同。

方案二：第一个缓冲区放入未处理的数据，第二个缓冲区放入处理后的数据。程序如下：

```
semaphore sr,sm1,sm2,sp;
int lr, lm1, lm2,lp;
Datatype B1[K],  B2[K];
Lr=lm1= lm2= lp=0;
sr.value=K ;sm1.value=0; sm2.value= K;  sp.value=0;
cobegin
repeatR;
repeat M;
        repeat P;
coend
Process R
 {
Datatype data ;
      read data;
      P(sr);
      B1[lr]=data;
      lr=(lr+1)%K;
```

```
     V(sm1);
    }
Process M
   {
Datatype data;
    P(sm1);
    data = B1[lm];
   lm1=(lm1+1)%K;
    V(sr);
   加工 data ；
   P(sm2);
   B2[lm2] = data;
   lm2=(lm2+1)%K;
   V(sp);
 }
Process P
  { Datatype data;
     P(sp);
    data = B2[lp];
     lp=(lp+1)%K;
     V(sm2);
    print data;
    }
```

10. 假定一个阅览室最多可以容纳 100 人阅读,读者进入和离开阅览室时，都必须在阅览室门口的一个登记表上注册或注销。假定每次只允许一个人注册或注销，设阅览室内有 100 个座位。

(1) 试问：应编制几个程序和设置几个进程？程序和进程的对应关系如何？

(2) 试用 P、V 操作编写读者进程的同步算法。

答：应编写 1 个程序，设置 *N* 个进程，程序和进程是 1：*N* 的关系。

```
semaphore full, mutex;
PID seat[100];
int i;
 full.value= 100; mutex.value=1;
for (i=0; i<100; i++) seat[i]=-1;
Process reader(PID procid )
{ int i;
  P(full);
  P(mutex);
  i=0;
  while (seat[i] != -1) i++ ;
  seat [i] = procid;
```

```
    V(mutex);
    read at seat i;
    P(mutex);
    seat[i]= -1;
    V(mutex);
    V(full);
}
```

19. 若系统有同类资源 *m* 个，被 *n* 个进程共享，问：当 *m*>*n* 和 *m*≤*n* 时，每个进程最多可以请求多少个这类资源，使系统一定不会发生死锁？

答：当 *m*>*n* 时，每个进程最多请求⌈*m*/*n*⌉ 个资源；*m*≤*n* 时，每个进程最多请求 1 个资源。

20. 设系统有某类资源共 12 个，用银行家算法判断下列每个状态是否安全。如果是安全的，说明所有进程是如何能运行完毕的。如果是不安全的，说明为什么可能产生死锁。

状态 A

进程	占有资源数	最大需求
进程 1	2	6
进程 2	4	7
进程 3	5	6
进程 4	0	2

状态 B

进程	占有资源数	最大需求
进程 1	4	8
进程 2	2	6
进程 3	5	7

答：状态 A 是安全的，系统剩余 1 个资源，可以分配给进程 3，进程 3 结束后收回 6 个资源，其他进程可以执行结束。状态 B 是不安全的，系统剩余 1 个资源，但是 3 个进程还需资源数均大于 1，故无法分配。

第 4 章 处理机调度

17. 5 个进程，从 A 到 E，同时到达系统。它们的估计运行时间分别为 15min、9min、3min、6min 和 12min，它们的优先级分别为 6、3、7、9 和 4(值越小，表示优先级越高)。对下面的每种调度算法，确定每个进程的周转时间和所有进程的平均周转时间(忽略进程切换的开销)，并解释是如何得到这个结果的。

(1) 优先级调度。

(2) 先来先服务(按 A、B、C、D、E 的顺序运行)。

(3) 最短进程优先。

答：(1) 优先级调度顺序和周转时间：

B：9

E：9+12=21

A：21+15=36

C：36+3=39

D：39+6=45

平均周转时间：(36+9+39+45+21)/5=30(min)

(2) 先来先服务调度顺序和周转时间：

A：15

B：15+9=24

C：24+3=27

D：27+6=33

E：33+12=45

平均周转时间：(15+24+27+33+45)/5=28.8(min)

(3) 最短进程优先调度顺序和周转时间：

C：3

D：3+6=9

B：9+9=18

E：18+12=30

A：30+15=45

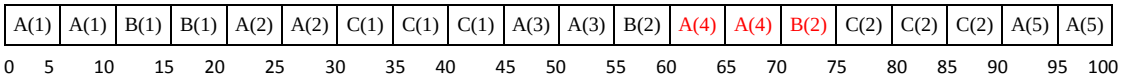
平均周转时间：(45+18+3+9+30)/5=21(min)

18. 有一组周期性的任务(3 个)，题表 4-1 给出了它们的时限要求表，请给出关于这组任务的调度顺序图。

题表 4-1　习题 18 进程时限要求表

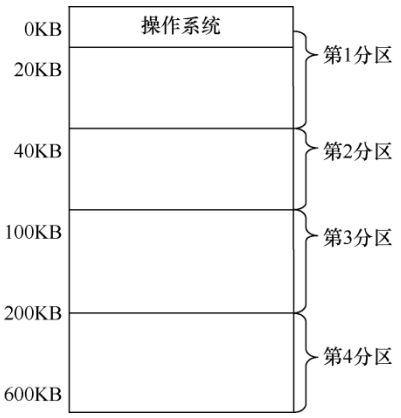
进程	到达时间	执行时间	完成最后期限
A(1)	0	10	20
A(2)	20	10	40
A(3)	40	10	60
A(4)	60	10	80
A(5)	80	10	100
…	…	…	…
B(1)	0	10	50
B(2)	50	10	100
…	…	…	…
C(1)	0	15	50
C(2)	50	15	100
…	…	…	…

答：按时限调度算法执行的一种可能调度顺序如下：



第 5 章　内存管理

5. 在固定分区管理中，当有 1K、9K、33K 和 121K 四个进程要求进入系统时，试分析



题图 5-1　主存分配情况

主存空间的分配情况(主存初始状态如题图 5-1 所示)，并说明主存的浪费有多大？

答：1K 进程分配到第 1 分区，9K 进程分配到第 2 分区，33K 进程分配到第 3 分区，121K 进程分配到第 4 分区。分区已分配完，无法接纳新的进程运行。

主存的浪费达到：

$$580–(1+9+33+121)=416(K)$$

6. 在动态分区管理中，当有 1KB、9KB、33KB 和 121KB 四个进程要求进入系统时，试分析主存空间的分配情况。

答：动态分区管理根据进程执行需求逐一将 580KB 的主存分割下来分配给各进程，各进程可紧接着，最终还有 416KB 的空闲区，可接纳新的进程运行。

22. 如果一个作业在执行过程中，按下列的页号依次访问主存：1，2，3，4，2，1，5，6，2，1，2，3，7，6，3，2，1，2，3，6。作业固定占用四个主存页框(块)，试问分别采用 FIFO、LRU、Clock 和 OPT 算法时，各产生多少次缺页中断？并计算相应的缺页中断率，同时写出在这四种调度算法下产生缺页中断时淘汰的页面号和在主存的页面号。

答：共 20 个操作页面，4 个页框。

FIFO 算法：

访问次序	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
主存页号	1	1	1	1	1	1	5	5	5	5	5	3	3	3	3	3	1	1	1	1
		2	2	2	2	2	2	6	6	6	6	6	7	7	7	7	7	7	3	3
			3	3	3	3	3	3	2	2	2	2	2	6	6	6	6	6	6	6
				4	4	4	4	4	4	1	1	1	1	1	1	2	2	2	2	2
淘汰页面							1	2	3	4		5	6	2		1	3		7	

缺页次数：14 次，缺页中断率为 14/20=70%。

LRU 算法：

访问次序	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
主存页号				4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
			3	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3
		2	2	2	3	4	2	1	5	6	6	1	2	3	7	6	3	3	1	2
	1	1	1	1	1	3	4	2	1	5	5	6	1	2	2	7	6	6	6	1
淘汰页面							3	4				5	6	1			7			

缺页次数：10 次，缺页中断率为 10/20=50%。

Clock 算法：

访问次序	1		2		3		4		2		1		5		6		2		1		2		3		7		6		3		2		1		2		3		6	
主存页号	1	1	1	1	1	1	1	1	1	1	1	1	5	1	5	1	5	1	5	1	5	1	3	1	3	1	3	1	3	1	3	1	1	1	1	1	1	1	1	1
		0	2	1	2	1	2	1	2	1	2	1	2	0	6	1	6	1	6	1	6	1	6	0	7	1	7	1	7	1	7	1	7	0	7	0	3	1	3	1
		0		0	3	1	3	1	3	1	3	1	3	0	3	0	2	1	2	1	2	1	2	0	2	0	6	1	6	1	6	1	6	0	6	0	6	0	6	1
		0		0		0	4	1	4	1	4	1	4	0	4	0	4	0	1	1	1	1	1	0	1	0	1	0	1	0	2	1	2	0	2	1	2	1	2	0
淘汰页面													1		2		3		4				5		6		2				1		3				7				

（→表示指针，*表示访问位 R = 1，无*表示访问位 R = 0）

访问次序	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
主存页号	1*	1*	1*	→1*	→1*	→1*	5*	5*	5*	→5*	→5*	3*	3*	3*	3*	→3*	1*	1*	1*	1*
	→	2*	2*	2*	2*	2*	→2	6*	6*	6*	6*	→6	7*	7*	7*	7*	→7	→7	3*	3*
		→	3*	3*	3*	3*	3	→3	2*	2*	2*	2	→2	6*	6*	6*	6	6	→6	→6*
			→	4*	4*	4*	4	4	→4	1*	1*	1	1	→1	→1	2*	2	2*	2*	2*
淘汰							1	2	3	4		5	6	2		1	3		7	

缺页次数：14 次，缺页中断率为 14/20=70%。

OPT 算法：

访问次序	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
主存页号	1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7	1	1	1	1
		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
				4	4	4	5	6	6	6	6	6	6	6	6	6	6	6	6	6
淘汰页面							4	5					1				7			

缺页次数：8，缺页中断率为 8/20=40%。

23. 假设一个进程固定分配 5 个页框，该进程的各页的装载时间、最近访问时间、R 位和 M 位信息如表 5-9 所示。

表 5-9　习题 23 进程信息

页号	装载时间	最近访问时间	R	M
0	125	280	1	0
1	225	260	0	1
2	150	270	0	0
3	100	290	1	1

当页号 4 要访问时，请问：

(1) FIFO 页面置换算法应置换哪个页面？

(2) LRU 页面置换算法应置换哪个页面？

(3) NRU 页面置换算法应置换哪个页面？

(4) 第二次机会页面置换算法应置换哪个页面？

答：

(1) FIFO 算法淘汰最先进入内存的页，由表 5-9 所示可知，第 3 页最先进入内存(装入时间最小)，故 FIFO 算法将淘汰第 3 页。

(2) LRU 算法淘汰最近最久未用的页。根据表 5-9 所示，最近最久未使用的页(上次引用时间最小)是第 1 页，故 LRU 算法淘汰第 1 页。

(3) NRU 算法是从那些最近一个时期内未被访问过的页中任选一页淘汰。根据表 5-9 所示，只有第 2 页的 R 和 M 位均为 0，故第 2 页是最近一个时间内未被访问过的页，所以 NRU 算法将淘汰第 2 页。

(4) 第二次机会算法是淘汰一个自上一次对它检查以来没有被访问过的页。根据表 5-9 所示可知，自上一次对它检查以来第 2 页未被访问过且没被修改(R 和 M 均为 0)，故第二次机会算法淘汰第 2 页。

24. 假设虚拟地址页访问流包含了一个重复访问的长的页面访问序列，该序列偶尔会出现随机的页面访问，如访问序列为：0，1，…，511，428，0，1，…，511，202，0，1，…，包含了一个重复的访问页面访问序列 0，1，…，511 并伴随着随机访问页面 428，202。请思考回答以下问题：

(1) 为什么标准的 FIFO、LRU、Clock 页面置换算法对于给定页框数小于序列长度的页面分配时其处理负载不是很好？

(2) 如果分给进程可达到 500 个页框，请给出一个比 FIFO、LRU 或 Clock 执行性能更好的页面置换方案。

答：

(1) 因为标准的 FIFO、LRU、Clock 页面置换算法是针对进程访问内存的时间远离性设计置换的。当对给定页框数小于序列长度的页面进行分配调度时，重复访问的页面未能被很好地保留在内存中，而被置换了，缺页中断频率较高。故这种情况下 LRU、Clock、FIFO 算法性能未达到理想效果。

(2) 一个可能的更好的页面置换方案是：对于给定的 m 个页框，始终保留先前访问的 m–1 个页面在前 m–1 页框中，缺页中断时不进行置换，而仅仅置换第 m 个页框所对应的页面。这样就能更好地保留重复访问的页面在内存中，降低缺页中断率。

第 6 章 文件管理

1. What would happen if the bitmap or free list containing the information about free disk blocks was completely lost due to a crash? Is there any way to recover from this disaster, or is it bye-bye disk? Discuss your answer for the FAT-16 file system.

答：

1）若 bitmap or free list 信息丢失了，系统将无法准确掌握空闲磁盘块的情况，从而无法进行新的文件存储空间的分配。

2）有方法可使系统从该损坏中恢复。

3） 对于 FAT-16 file system，系统可在内存开辟出 FAT 的存储空间并清 0，然后从根目录开始，逐一检查每一个文件和每一个子目录中的文件所占用的相对块号，按链接的形式填入内存的 FAT 中。最后用内存的 FAT 更新磁盘上的 FAT 区域。

2. A UNIX file system has 8-KB blocks and 4-byte disk address. Assume that there are 12 direct block pointers, and a singly, doubly and triply pointer in each i-node, then

1) What is the maximum file size supported by this system?

2) Assuming no information other than that the file i-node is already in main memory, how many disk accesses are required to access the byte in position 13,423,965?

答：

（1）1 个磁盘块可容纳的磁盘地址数：8KB / 4B = 2K

一到三级间接索引下的文件最大尺寸：

12*8K + 2K*8K + 2K*2K*8K + 2K*2K*2K*8K = 64T 32G 16M 96KB

（2）13423965 / (8*1024) = 1638 余 5460

所以该地址的逻辑块号是 1638。

12 < 1638 < 12+2048，故该地址在一级间接索引中。

由于 i-node 已在内存，因此需要 2 次访问磁盘块：

一次访问一级间接索引块，一次访问含有该位置字节的数据块。

3. Given a disk-block size of 4KB and block-pointer address value of 4 bytes, what is the largest file size (in bytes) that can be addressed using 10 direct address and one indirect block?

答：1 个磁盘块可以容纳的磁盘地址数：4KB / 4B = 1K

10 个直接地址和一级间接索引下的文件最大尺寸：

10*4K + 1K*4K = 4M 40KB

4. The following figure is a file allocation table in main memory, please write out: which disk blocks does file A use? Which disk blocks does file B use? Which disk blocks does file C use?

0	9	
1	14	
2	10	
3	11	
4	3	← File A starts here
5		
6	7	← File B starts here
7	2	
8	0	← File C starts here
9	1	
10	12	
11	13	
12	-1	
13	-1	
14	-1	
15		

答：File A uses disk blocks 4, 3, 11 and 13;

File B uses disk blocks 6, 7, 2, 10 and 12;

File C uses disk blocks 8, 0, 9, 1 and 14.

5 In FreeBSD system, the size of a disk-block is 4 KB, and the address of a block is represented

by 8 bytes. If an i-node contain 12 direct entries, one single indirect entry, one double indirect entry, and one triple indirect entry, then

1) What is the largest file size (in bytes) supported by FreeBSD ?
2) Which logic block is the logical address 4239361 in ?

答：

（1）1 个磁盘块可以容纳的磁盘地址数：4KB / 8B = 0.5K.

文件的最大尺寸 = 12*4K+0.5K*4K+0.5K*0.5K*4K+0.5K*0.5K*0.5K*4K = 513G 2M 48KB（0.5T 1G 2M 48KB）

（2）磁盘块大小为 4KB = 4 * 1024B = 4096B

4239361 / 4096 = 1035 余 1，所以该地址的逻辑块号是 1035。

第 7 章 I/O 管理

16. 假定某活动头磁盘有 200 个磁道，编号为 0~199，磁头已经完成对 125 号磁道的访问，并向磁道号增大方向移动，则对于下列请求序列：86、147、91、177、94、150、102、175、130。求在下列调度策略下词头的移动顺序及其移动量(以磁道数计)。(1)FCFS；(2)SSTF；(3)SCAN；(4)CSCAN。

答：

(1) FCFS

访问顺序：125->86->147->91->177->94->150->102->175->130

则移到磁道长度为

$$(125-86)+(147-86)+(147-91)+(177-91)+(147-94)+(150-94)+(150-102)+(175-102)$$
$$+(175-130)=547$$

(2) SSTF

访问顺序：125->130->147->150->175->177->102->94->91->86

则移到磁道长度为

$$(177-125)+(177-86)=143$$

(3) SCAN

访问顺序：125->130->147->150->175->177->102->94->91->86

则移到磁道长度为

$$(177-125)+(177-86)=143$$

(3) C-SCAN

访问顺序：125->130->147->150->175->177->86->91->94->102

则移到磁道长度为

$$(177-125)+(177-86)+(102-86)=159$$