操作系统 第 2 章

2-9.

(1) x<=3　运行顺序为 Px，P3，P5，P6，P9

T=(x+(x+3)+(x+3+5)+(x+3+5+6)+(x+3+5+6+9))/5=x+9.6

(2) 3<x<=5　运行顺序为 P3，Px，P5，P6，P9

T=(3+(3+x)+(3+x+5)+(3+x+5+6)+(3+x+5+6+9))/5=0.8x+10.2

(3) 5<x<=6　T=0.6x+11.2

(4) 6<x<=9　T=0.4x+12.4

(5) 9<x　T=0.2x+14.2

2-12. 计算采用 FCFS、SJN、RHN 的平均周转时间和平均带权周转时间：

作业号	提交时刻	估计运行时间	开始运行时刻			完成时刻		
			FCFS	SJN	RHN	FCFS	SJN	RHN
1	8.00	2.0	8.0	8.0	8.0	10.0	10.0	10.0
2	9.00	1.2	10.0	10.8	10.5	11.2	12.0	11.7
3	9.50	0.5	11.2	10.0	10.0	11.7	10.5	10.5
4	10.20	0.3	11.7	10.5	11.7	12.0	10.8	12.0
		2.05/3.307		1.65/1.875		1.875/2.8125		

1) FCFS　作业运行顺序：1，2，3，4

各作业的周转时间 Ti 和平均周转时间 T：

T1=10.0-8.00=2.0　　T2=11.2-9.00=2.2

T3=11.7-9.5=2.2　　T4=12.0-10.2=1.8

T=(T1+T2+T3+T4)/4=(2.0+2.2+2.2+1.8)/4=8.2/4=2.05

各个作业的平均带权周转时间 W 计算如下：

W=(2/2+2.2/1.2+2.2/0.5+1.8/0.3)=(1+1.83+4.4+6)/4=3.307

2) SJN　作业运行顺序：1，3，4，2

T1=10.0-8.00=2.0　　T2=12-9.00=3

T3=10.5-9.5=1.0　　T4=10.8-10.2=0.6

T=(T1+T2+T3+T4)/4=(2.0+3.0+1.0+0.6)/4=6.6/4=1.65

各个作业的平均带权周转时间 W 计算如下：

W=(2/2+3/1.2+1/0.5+0.6/0.3)/4=1.875

3) HRN　作业运行顺序：1，3，2，4

先选择作业 1　从 8.00-------10.00。当作业 1 完成时，究竟选谁运行，只有

通过计算,选择响应比高者运行:

作业 2 的响应比=((10-9.0) +1.2)/1.2=1.83

作业 3 的响应比=((10-9.5)+0.5) /0.5=2.0

作业 4 还未到，只能选作业 3 运行。

作业 3 运行到 10.5 结束，再计算剩余的作业 2 和 4:

作业 2 的响应比=（(10.5-9.0）+1.2）/1.2=2.25

作业 4 的响应比=((10.5-10.2)+0.3) /0.3=2　选作业 2 运行。

作业 2 到 11.7 完成。最后运行作业 4。运行到 12.0,全部结束。

各个作业的周转时间计算如下:

t1=2　t2=11.7-9=2.7　t3=10.5-9.5=1　t4=12-10.2=1.8

各个作业的平均周转时间计算如下:

T==(2+2.7+1+1.8)/4=1.875

各个作业的平均带权周转时间计算如下:

W=(2/2+2.7/1.2+1/0.5+1.8/0.3)/4=2.8125

2-13. 已知作业 A,B,C,D,E 需要的运行时间分别为 10,6,2,4,8 分钟，优先级分别为 3,5,2,1,4。

(1) 轮转法（假定时间片=2 分钟）

作业完成的顺序为 C,D,B,E,A

开始作业轮转一周需 10 分钟,

作业 C 的周转时间：Tc=10 分钟 （6 分）

C 完成后，剩下四个作业，轮转一周需 8 分钟,

作业 D 的周转时间：Td=10+8×(4-2)/2=18 分钟 （16 分）

D 完成后，剩下三个作业，轮转一周需 6 分钟,

作业 B 的周转时间：Tb=18+6×(6-2-2)/2=24 分钟 （22 分）

B 完成后，剩下两个作业，轮转一周需 4 分钟,

作业 E 的周转时间：Te=24+4=28 分钟 （28 分）

E 完成后，只剩下作业 A,

作业 A 的周转时间：Ta=28+2=30 分钟 （30 分）

平均周转时间： T=(10+18+24+28+30)/5=22 分 （20.4 分）

(2) 优先级调度法

作业完成顺序为：B,E,A,C,D

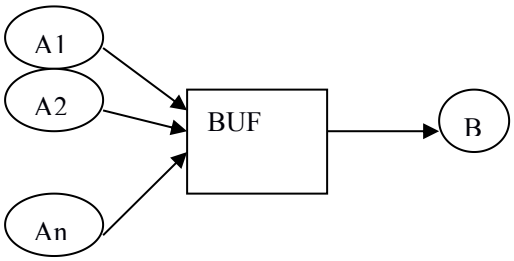
Tb=6 分，Te=6+8=14 分，Ta=14+10=24 分，Tc=24+2=26 分，

Td=26+4=30 分。

平均周转时间： T=(6+14+24+26+30)/5=20 分

第 3 章 习题答案

3-7. 系统中有 n+1 个进程。其中 A1、A2、…、An 分别通过缓冲区向进程 B 发送消息。相互之间的制约关系为：发送进程 A1、A2、…、An 要互斥地向 BUF 中送消息，当接收进程 B 还未将消息接收完之前，任何一个发送不能再送。同样，B 不能重复接收同一个消息。 为此，应设置两个信号量 s1 和 s2。设系统只有容纳一个消息的缓冲区，用信号量 s1 表示，其初值为 1，它用来制约发送进程。信号量 s2 用来制约接收进程，其初值为 0。



现可用 PV 操作描述如下：

进程 A1、…、An 执行的过程为：

```
begin
    准备消息
    P(s1)
    将消息送入 BUF
    V(s2)
end
```

进程 B 执行的过程为：

```
begin
    P(S2)
    从缓冲区 BUF 取消息
    V(s1)
    消耗消息
end
```

若缓冲区容量为 m 个，这个问题就变为生产者和消费者问题。

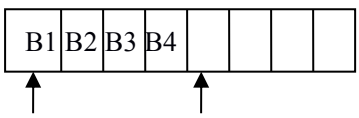
3-8. 首先这里的 IN 和 OUT 分别表示读写指针，而不是信号量。在系统初启时，环行缓冲区为空，此时 IN 和 OUT 都初始化为 0。当并发进程通过环行缓冲区通信时，写进程不断地写，读进程不断地读，使得读写指针不断变化。写进程的速度太快，缓冲区会满；读进程的速度太快，缓冲区会空。

已知循环缓冲区的容量为 100。则

当（IN+1）%100 = OUT 时，说明缓冲区已满。

当 IN = OUT 时，说明缓冲区已空。

初始化时，IN=OUT=0。一段时间以后：



OUT IN

3-9. 为描述阅览室，用一个登记表来记录使用情况。表中共有 100 项。每当有读者进入阅览室时，为了正确地登记，各读者应互斥使用。为此设两个信号量。mutex：互斥信号量，用来制约各读者互斥地进行登记，其初值为 1；empty 同步信号量，用来制约各读者能同时进入阅览室的数量，初值为 100。下面用两个过程描述对表格应执行的动作：

登记过程：

```
begin
  p(empty)
  p(mutex)
  找到一个登记项登记
  v(mutex)
end
```

擦除过程：

```
begin
  p(mutex)
  找到自己的登记项擦除
  v(mutex)
  v(empty)
end
```

为了正确地描述读者的动作，我们可以将读者看成进程。若干读者希望进入阅览室时，调用登记过程，退出阅览室时，调用擦除过程。可见一个程序可对应多个读者。可设的进程数由读者数决定。其动作如下：

```
begin
  调用登记过程
  进入阅览室阅读
  准备退出
  调用擦除过程
end
```

3-12.有 4 个同类资源，3 个进程，每个进程的最大申请为 2，系统不会发生死锁。最不利原则：3 个进程都各自获得了一个资源，都还需申请第二个资源。此时，因系统还有一个剩余资源，所以能满足任一个进程的剩余需求。

3-13. 有 6 个磁带机和 n 个进程。每个进程的最大申请为 2，问 n 取什么值时，系统不会死锁?

答：为了使系统不发生死锁，应该满足： n=6-1=5

3-14.

证明：假定会死锁，则根据死锁定义，N 个进程之间相互等待，至少需要 N 个单位资源，又系统 M 个资源已分完，故所有进程需求总和大于或等于 M+N，这与题中的所有进程需求总和小于 M+N 矛盾，故假设不成立。因此，在这种情况下不

会死锁。

3-15.

M1:	M2:	M3:	M4:
……	P(s12);	P(s13);	P(s14);
V(s12);	……	……	……
V(s13);	V(s26);	V(s36);	V(s47);
V(s14);	……	V(s38);	……

附加：m 个同类资源，n 个进程，每个进程的对资源的最大需求量：

当 m>n 时，每个进程最多可以请求 $\lceil m/n \rceil$ 个该类资源

当 m=n 时，每个进程最多可以请求 1 个该类资源

当 m<n 时，每个进程最多可以请求 1 个该类资源

(当 m>n 时，每个进程最多可以请求(m+n-1)/n 个该类资源)

3-15

解答：

这是进程之间的同步问题。M2、M3 和 M4 必须在接收到 M1 的消息后才能运行。同理，M6 必须在 M2 和 M3 之后运行，M7 必须在 M4，M5 之后运行，M8 必须在 M3、M7 之后运行。如何保证呢？需设置相应的信号量来保证：S12，S13，S14，用来制约 M2、M3 和 M4 的运行；S26，S36，用来制约 M6 的运行；S47，S57，用来制约 M7 的运行；S38，S78 用来制约 M8 的运行。

各进程的制约关系描述如下。

```
S12,S13,S14,S26,S36,S47,S57,S38,S78:semaphore;
S12:=0;S13:=0;S14:=0;S26:=0;S36:=0;S47:=0;S57:=0;S38:=0;S78:=0;
COBEGIN
   PROCESS M1:              PROCESS M2:
   BEGIN                    BEGIN
       V(S12);                  P(S12);
       V(S13);                  V(S26);
       V(S14);              END
   END

   PROCESS M3:              PROCESS M4:
   BEGIN                    BEGIN
```

```
    P(S13);                    P(S14);
    V(S36);                    V(S47);
    V(S38);                END
END

PROCESS M5:                PROCESS M6:
BEGIN                      BEGIN
    V(S57);                    P(S26);
END                            P(S36);
                           END
PROCESS M7:                PROCESS M8
BEGIN                      BEGIN
    P(S47);                    P(S38);
    P(S57);                    P(S78);
    V(S78);                END
END
COEND
```

3-16. 叉子是临界资源，在一段时间内只允许一个哲学家使用。一个信号量表示一把叉子，五个信号量构成信号量数组，这些信号量的初值为 1。

```
int fork[0]=fork[1]=…=fork[4]=1;
第 i 个哲学家所执行的程序：
do{
      P(mutex);
       P(fork[i]);
       P(fork[(i+1)mod5]);
      V(mutex);
       吃饭
      V(fork[i]);
      V(fork[(i+1)mod5]);
} while(1);
```

3-17.

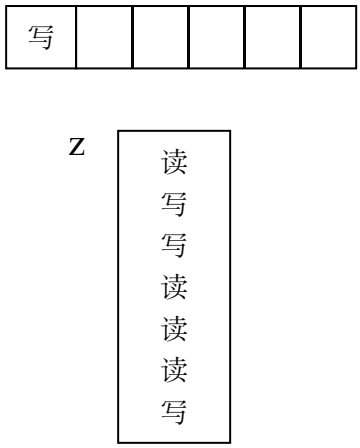
（1）公平竞争（无写者时，读者仍遵循多个读者可以同时读）

rmutex 互斥共享 readcount；rwmutex 读写互斥，写写互斥；

读写进程在 z 上排队。

```
int rmutex=1,rwmutex=1,readcount=0;
```

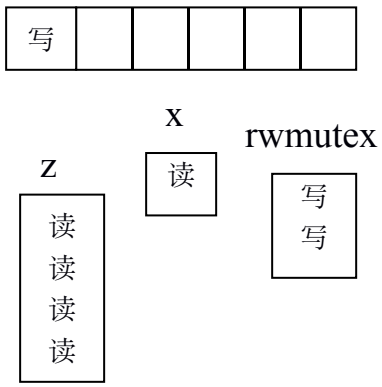
```
reader:
begin
    p(z);     //读写进程在 z 上排队。
    p(rmutex);
    if(readcount=0) then p(rwmutex);
    end if
    ++readcount;
    v(rmutex);
    v(z);     //无写者时，多个读者可以同时读.
    read data;
    p(rmutex);
    --readcount;
    if(readcount=0 then v(rwmutex);
    end if;
    v(rmutex);
    …
end
writer:
begin
    p(z);     //读写进程在 z 上排队。
    p(rwmutex);
    write data;
    v(rwmutex);
    v(z);
    …
end
```



（2）写者优先

```
int readcount,writecount;
semaphore rmutex=1,wmutex=1,rwmutex=1,z=1,x=1;
reader:
//当来了一个写进程时，通过 p(x)禁止其后读进程读，直到写进程写完为止。
    while(1){
        p(z);     //其他读进程在 z 上排队
        p(x);     //一个读进程与一个写进程在 x 上竞争
        p(rmutex);   //读进程互斥访问 readcount
```

```
        ++readcount;
        if(readcount==1) p(rwmutex);
        v(rmutex);
        v(x);
        v(z);
        read data;   //临界区
        p(rmutex);
        --readcount;
        if(readcount==0) v(rwmutex);
        v(rmutex);
    }
```




```
Writer:
    while(1){
        p(wmutex);          //写进程互斥访问 writecount
        ++writecount;
        if(writecount==1) p(x);   //一个写进程与一个读进程在 x 上竞争
        v(wmutex);
        p(rwmutex);         //其他写进程在 rwmutex 上排队
        write data;   //临界区
        v(rwmutex);
        p(wmutex);
        --writecount;
        if(writecount==0) v(x); //写进程都写完时，通过 v(x)允许读进程读
        v(wmutex);
    }
```

附加题:

读者优先，规定仅允许 5 个进程同时读，怎样修改程序?

解：增加一个资源信号量 s，初值为 5。

```
  int s=5;
Reader:
begin
```

```
        P(rmutex);
        readcount=readcount+1;
        if(readcount==1)then P(rwmutex);
        V(rmutex);
        P(s);
        read_file();
        V(s);
        P(rmutex);
        readcount=readcount-1;
        if(readcount==0)then V(rwmutex);
        V(rmutex);
end

writer:
begin
    p(rwmutex);
    write data;
    v(rwmutex);
    …
end
```

3-18.

```
    int s1=0,   s2=n;
    顾客进程:                          理发师进程:
        P(s2);                             P(s1);
        V(s1);                             给顾客理发
        坐椅子等理发                       V(s2)
```

3-19. （2）和（4）会发生死锁。

3-20.

	P1/剩余	P2/剩余	P3/剩余	系统剩余
1	3/5			7
2		2/4		5
3			4（不安全）	
4	5/3			3
5		2（不安全）		
6	(5+3)/0			0(8)

7			4/3	4
8		(2+2)/2		2
9				

(1) P1 占有 5 个资源，剩余 3 个资源请求。
P2 占有 2 个资源，剩余 4 个资源请求。
P3 占有 0 个资源，剩余 7 个资源请求。
系统剩余 3 个资源。

(2) P1 的请求最先满足。进程完成序列：P1，P2，P3。

3-21.

(1)

最大需求矩阵：　　分配矩阵：　　剩余请求矩阵：

$$\text{Max} = \begin{pmatrix} 0&0&1&2 \\ 1&7&5&0 \\ 2&3&5&6 \\ 0&6&5&2 \\ 0&6&5&6 \end{pmatrix} \quad \text{Allocation} = \begin{pmatrix} 0&0&1&2 \\ 1&0&0&0 \\ 1&3&5&4 \\ 0&6&3&2 \\ 0&0&1&4 \end{pmatrix} \quad \text{Need} = \begin{pmatrix} 0&0&0&0 \\ 0&7&5&0 \\ 1&0&0&2 \\ 0&0&2&0 \\ 0&6&4&2 \end{pmatrix}$$

剩余资源向量：Available=(1 5 0 2)

(2) 当前系统是安全的。

判断系统是否安全，只要检查系统剩余资源向量能否对各进程的剩余请求向量找到一个进程完成序列，当按照这个序列为各进程分配资源时，各进程都能成功完成。若能找到，则系统是安全的，否则，为不安全。

先找到 p0，因为 p0 已满足最大资源请求，它可以完成，释放其占有的资源，使系统剩余资源向量为（1 5 1 4）

之后，系统剩余资源向量（1 5 1 4），可满足进程 p2，使 p2 可以完成，释放其占有的资源，使系统剩余资源向量为（2 8 6 8）。

之后无论选哪一个进程都可成功完成。

故找到的进程完成序列可为：p0,p2,p4,p3,p1；或 p0,p2,p3,p1,p4 等，故系统是安全的。

(3) 因系统剩余可用向量为 (1502)，p2 的剩余请求向量为 (1002)，即 (1502)>(1002)。故，当 p2 提出（1001）请求时，能满足。进程完成序列：p0,p2,p4,p3,p1

第 4 章　习题答案

4-14 内存有如下顺序排列的空闲块：10K，40K，20K，18K，7K，9K，12K 和 15K，有如下的请求序列：12K，10K，9K。

（1）若采用首次适应法：

- 12K 的请求：将分配 40K 的空闲块，　40K 变为剩余的（40-12）K=28K，空闲队列变为：10K，28K，20K，18K，7K，9K，12K 和 15K；
- 10K 的请求：将分配 10K 的空闲块，空闲队列变为：28K，20K，18K，7K，9K，12K 和 15K；
- 9K 的请求：将分配 28K 的空闲块，空闲队列变为：（28-9）=18K，20K，18K，7K，9K，12K 和 15K；

（2）若采用最佳适应法：

- 12K 的请求：将分配 12K 的空闲块，空闲队列变为：10K，40K，20K，18K，7K，9K 和 15K；
- 10K 的请求：将分配 10K 的空闲块，空闲队列变为：40K，20K，18K，7K，9K，12K 和 15K；
- 9K 的请求：将分配 9K 的空闲块，空闲队列变为：　40K，20K，18K，7K，　12K 和 15K；

（3）若采用最坏适应法：

- 12K 的请求，将分配 40K 的空闲块，空闲队列变为：10K，28K，20K，18K，7K，9K 和 15K；
- 10K 的请求：将分配 28K 的空闲块，空闲队列变为：　20K，18K，7K，9K，12K 和 15K；
- 9K 的请求：将分配 20K 的空闲块，空闲队列变为：10K，18K，11K，　18K，7K，　12K 和 15K。

4-15　有如下图所示的页表中的虚地址与物理地址之间的关系，即该进程分得 6 个内存块。页的大小为 4096。给出对应下面虚地址的物理地址：(1)20；　(2) 5100；(3) 8300；　(4) 47000.

0	1	2	3	4	5	6	7
2	1	6	0	4	3	x	x

解：

（1）虚地址 20 变为页号 0 和页内偏移 20

由页号查页表得 0 页对应内存块号为 2 ，可计算得

物理地址=块号*页的大小+页内偏移=2*4096+20=8212

（2）虚地址 5100 变为页号 1 和页内偏移 1004（5100/4096）

由页号查页表得 1 页对应内存块号为 1 ， 可计算得

物理地址=块号*页的大小+页内偏移=1*4096+1004=5100

(3) 虚地址 8300 变为页号 2 和页内偏移 108

由页号查页表得 2 页对应内存块号为 6 ， 可计算得

物理地址=块号*页的大小+页内偏移=6*4096+108=24684

(4) 虚地址 47000 变为页号 11 和页内偏移 1944

11＞7 页号越界

4-16杆一个作业在执行过程中，按如下顺序依次访问各页，作业分得四个主存块，问分别采用 FIFO、LRU 和 OPT 算法时，要产生多少次缺页中断?设进程开始运行时，主存没有页面。

页访问串顺序为：0 1 7 2 3 2 7 1 0 3 2 5 1 7

(1) FIFO

0 1 7 2 3 2 7 1 0 3 2 5 1 7

0	1	7	2	3	3	3	3	0	0	0	5	1	7
0	1	7	**2**	2	2	2	3	3	3	0	5	1
	0	1	7	7	7	7	**2**	2	2	3	0	5
		0	1	1	1	1	7	7	7	2	3	0
F	F	F	F	F	S	S	S	F	S	S	F	F	F

采用 FIFO 淘汰算法，产生 9 次缺页中断。

(2) LRU

0 1 7 2 3 2 7 1 0 3 2 5 1 7

0	**1**	**7**	**2**	**3**	2	7	1	**0**	**3**	**2**	**5**	**1**	**7**
0	1	7	2	3	2	7	1	0	3	2	5	1
	0	1	7	7	3	2	7	1	0	3	2	5
		0	1	1	1	3	2	7	1	0	3	2
F	F	F	F	F	S	S	S	F	F	F	F	F	F

采用 LRU 算法时，产生 11 次缺页中断。

4-17 考虑如图所示的段表，给出如下所示的逻辑地址所对应的物理地址。

段始址	段的长度
219	600
2300	14
92	100
1326	580
1954	96

(1) 0，430 → 219+430=649
(2) 1，10 → 2300+10=2310
(3) 2，500 → 500＞100 段内地址越界
(4) 3，400 → 1326+400=1726
(5) 4，112 → 112＞96 段内地址越界

4-18 一台计算机含有 65536 字节的存储空间，这一空间被分成许多长度为 4096 字节的页。有一程序，其代码段为 32768 字节，数据段为 16386 字节，栈段为 15870 字节。试问该机器的主存空间适合这个作业吗?如果每页改成 512 字节，适合吗?
答:

(1)

存储空间每块为 4096 个字节，共可分成 16 块。

程序代码段占 32768/4096=8 块，数据段占 16386/4096=5 块，栈段占 15870/4096=4 块，合计为 8+5+4=17 块，故该机器的主存空间不适合这个作业。

(2)

当存储空间每块为 512 个字节，共可分成 128 块。

程序代码段占 32768/512=64 块，数据段占 16386/512=33 块，栈段占 15870/512=31 块，合计为 64+33+31=128 块，故该机器的主存空间是适合这个作业的。

4-19 逻辑地址中，用 9 位表示页号，用 10 位表示页内地址。

4-20 (1) 缺页中断 50 次； 5000 次
(2) 缺页中断 100 次； 10000 次

4-21 0.9×(0.75×1+0.25×8)+0.10×(8+5000+8)+8

4-23 8192/4=2048
64=7+11+11+11+11+13 5 级页表

第 5 章 文件系统

5-9.杆 文件存贮空间管理可采用成组自由块链表或位示图。若一磁盘有 B 个盘块，

其中有 F 个自由块。若盘块号用 D 位表示。试给出使用自由块链表比使用位示图占用更少的空间的条件。当 D 为 16 时，给出满足条件的自由空间占整个空间的百分比。

解：一磁盘有 B 个盘块，用位图表示要使用 B 位

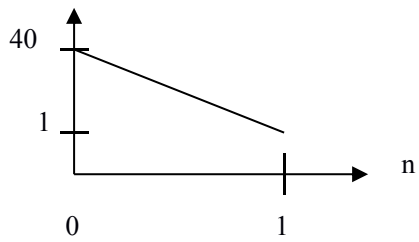
现有 F 个自由块，若表示一个盘块需用 D 位。则采用链表接连 F 个盘块，需要 F 个链指针，共占 F*D 位。使用自由块链表比使用位示图占用更少的空间的条件是 F*D<B。

当 D=16 时，满足条件的自由空间占整个空间的百分比为

F/B<1/16=6.25%。

5-10.杆 文件系统的执行速度依赖于缓冲池中找到盘块的比率。假设盘块从缓冲池读出用 1 毫秒，从盘上读出用 40 毫秒。从缓冲池找到盘块的比率为 n，请给出一个公式计算读盘块的平均时间，并画出 n 从 0 到 1.0 的函数图像。

解：读一个盘块的平均时间=n+40(1-n)=40-39n



5-13.

1574/256=6……38， 因此，要访问文件的第 7 个记录内的 38B 处。每个块放两个记录，所要访问的字节处在第 4 个逻辑块内，其对应的物理块号为 4，应访问 4 号块内的第 38 个字节。要访问 4 次磁盘才能将该字节的内容读出。

5-14. 共需要 4 次磁盘操作

5-15. 1GB=2^{30} ，16KB=2^{14} ， $2^{30}/2^{14}=2^{16}$ ，每个磁盘块号需要 2 个字节表示，即 2B。

(1) 10KB<16KB， 所以，只占用 1 个磁盘块。

(2) 1089KB/16KB=69 需一个索引块和 69 个数据块，共 70 个盘块。

(3) 129MB/16KB=8256 ， 16KB/2B=8K（个索引项）<8256

所以，需 2 个一级索引表和一个 1 个二级索引表，8256 个数据块。

共需 8259 个磁盘块。

第 6 章　设备管理

6-6. 下列工作各是在四层 I/O 软件的哪一层上实现的?囍

(1) 对于读磁盘，计算柱面、磁头和扇区囍（设备驱动）

(2) 维持最近所用块而设的高速缓冲囍（独立于设备的软件层）

(3) 向设备寄存器写命令囍（设备驱动）

(4) 查看是否允许用户使用设备囍（独立于设备的软件层）

(5) 为了打印，把二进制整数转换成 ASCII 码囍（用户进程）

6-13. 假设移动头磁盘有 200 个磁道(从 0 号到 199 号)。目前正在处理 143 号磁道上的请求，而刚刚处理结束的请求是 125 号，如果下面给出的是按到达时间的先后排成的等待服务队列：86，147，91，177，94，150，102，75，130。那么，用下列各种磁盘调度算法来满足这些请求所需的总磁头移动量是多少?囍

(1) FCFS：125 →143--86--147--91--177--94--150--102--75--130
满足这些请求所需的总磁头移动量=（143-86）+（147-86）+（147-91）+（177-91）+（177-94）+（150-94）+（150-102）+（102-75）+（130-75）
=57+61+56+86+83+56+48+27+55=524
(2) SSTF：125 →143--147--150--130--102--94--91--86--75--177
满足这些请求所需的总磁头移动量
=(150-143)+(150-75)+(177-75)=7+75+102=182
(3)SCAN：125 →143--147--150--177--199--130--102--94--91--86--75
满足这些请求所需的总磁头移动量=(199-143)+(199-75)=56+124=180
(5)C-SCAN：125　143--147--150--177--199--0--75--86--91-94—102 --130
满足这些请求所需的总磁头移动量=(199-143)+(199-0)+(130-0)=56+199+130=385
(4)C-LOOK：125 →143--147--150--177--75--86--91--94--102--130
满足这些请求所需的总磁头移动量
=(177-143)+(177-75)+(130-75)=33+102+55=190