操作系统原理作业

## 第1章

**1-2 批处理系统和分时系统各有什么特点？为什么分时系统的响应比较快？**

答 ：在批处理系统中操作人员将作业成批装入计算机并由计算机管理运行，在程 序的运行期间用户不能干预，因此批 处理系统的特点是：用户脱机使用计算机 ，作业成批处理，系统内多道程序并 发执行以及交互能力差。在分时系统中不 同用户通 过各自的终 端以交互 方式共同使 用一台计算 机，计算 机以“分时 ”的方 法轮流为 每个用户服 务。分时 系统的主要 特点是： 多个用户同 时使用计 算机的 同时性，人机问答方 式的交互 性，每个用 户独立使 用计算机的 独占性以 及系统 响应的及 时性。分时系统一般采用时间片 轮转的方法使一台计算机同时为多个终端用户服务，因此分时系统的响应比较快。联机处理

**1-5什么是多道程序设计技术？试述多道程序运行的特征。**

答：多道程序设计技术 是指同时 把多个作业 （程序） 放入内存并 允许它们 交替执行和 共享系统 中的各 类资源；当一道程序因某种原因（如 I/O 请求）而暂停执行时，CPU 立即转去 执行另一道程序。

多道程序运行具有如下特征：

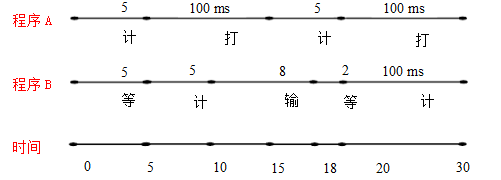
多道计算机内存中同时存放几道相互独立的程序。宏观上并行：同时进入系统的 几道程序都处于运行过程中，它们先后开始了各自的运行但都未运行完毕。 微观上串行：从微观上看内存中的多道程序轮流或分 时地占有处理机，交替执行。

**1-7操作系统的主要特性是什么？为什么会有这样的特性？**

答：并发性，共享性，异步性，虚拟性，这些特性保证了计算机能准确的运行，得出想要的结果。

**1-9**

1. 工作情况如图。



（2）CPU有空闲等待，它发生在100 ms150 ms时间段内，此时间段内程序A与程序B都在进行I/O操作。

（3）程序A无等待现象，程序B在0 ms50 ms时间段与180 ms200 ms时间段内有等待现象。

## 第2章

**2-1 什么是操作系统虚拟机？**

答：在裸机上配置了操作系统程序后就构成了操作系统虚拟机

**2-3 什么是处理机的态？为什么要区分处理机的态？**

答：处理机的态，就是处理机当前处于何种状态，正在执行哪类程序。为了保护操作系统，至少需要区分两种状态：管态和用户态。

**2-5 什么是中断？在计算机系统中为什么要引用中断？**

答：中断是指某个事件发生时，系统终止现行程序的运行、引出处理该事件程序进行处理，处理完毕后返回断点，继续执行。为了实现并发活动，为了实现计算机系统的自动化工作，系统必须具备处理中断的能力。

**2-8中断和俘获有什么不同？**

答：中断指由处理机外部事件引起的中断称为外中断，又称中断。包括I/O中断、外中断。俘获是指由外处理内部事件引起的中断称为俘获。包括访管俘获、程序性中断、机器故障中断。

**2-9 什么是中断响应，其实质是什么？**

答：中断响应是当中央处理机发现已有中断请求时，中止现行程序执行，并自动引出中断处理程序的过程。中断响应的实质是交换指令执行地址和处理器状态，以达到如下目的：（1）保留程序断点及有关信息；（2）自动转入相应的中断处理程序执行。

**2-10 以trap指令为例， 用图说明自陷处理的一般过程？**

PC（指令计数器)

）

自陷处理程序的PC

自陷处理程序的PS

PS(处理机状态字)

SP（处理机堆栈指针）

PC进栈

PS进栈

## 第3章

**3-1 什么是系统生成？**

答：系统生成就是操作系统的生成过程。

**3-2 系统引导的主要任务是什么？**

答：将操作系统的必要部分装入主存并使系统运行，最终处于命令接受状态。

**3-3 处理应用程序分哪几个作业步骤？**

答：①用某种预言编制一个程序，该程序被称为源程序。

②将源程序和初始数据记录在某种输入介质上，一般在终端设备上直接编辑源程序。

③按照一定的要求控制计算机工作，并经过加工，最终计算出结果。

**3-5 用户与操作系统的接口是什么？一个分时系统提供什么接口？一个批处理系统又提供什么接口？**

答 ：用户与 操作系统的 接口是指 操作系统提 供给用户 与计算机打 交道的外 部机制 。 一 个分时系统提供的接口有系统功能调 用和键盘操作命令。 一 个批处理系统提供的接口有系统功能 调用和作业控制语言。

**3-9 假定某系统提供硬件的访管指令（例如形式为“svc n”），为了实现系统调用，系统设计者应做哪些工作？用户又如何请求操作系统服务？**

答：当处理机执行到这一条指令时就发生中断，该中断称为访管中断，它表示正在运行的程序对操作系统的某种需求。借助中断，使机器状态由用户态转为管态。为了使控制能转到用户当前所需的那个例行子程序去，需要指令提供一个地址码。这个地址码表示系统调用的功能号，它是操作系统提供的众多的例行子程序的编号。在访管指令中填入相应的号码，就能使控制转到特定的例行子程序去执行以提供用户当前所需的服务。

## 第4章

**4-3什么是进程？进程与程序的主要区别是什么？**

答：进程是指一个具有一定独立功能的程序关于某个数据集合的一次运行活动。进程是动态的，程序的静态的，进程是程序的动态执行过程。

**4-6进程有哪几个基本状态？试画出进程状态迁移图，并注明发生变迁的原因。**

答：进程有三种基本状态，即运行状态，就绪状态，等待状态。如果系统能为每个进程提供一台处理机，则系统中所有进程都可以同时执行，但实际上处理机的数目总是小于进程数，因此往往只有少数几个进程可真正获得处理机控制权。

**4-9n个并发程序共用一个公共变量Q，写出用信号灯实现n个进程互斥时的程序描述，给出信号灯值的取值范围，并说明每个取值的物理意义。**

答：main( )

{int mutex=1;

int i;

cobegin

for(i=1;i<=n;i++)

P[i]; coend }

P( )

{

...

...

p(mutex);

...

v(mutex);

...

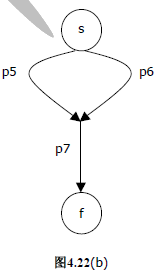
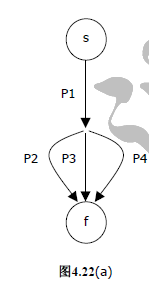
}

若mutex=1，表示没有进程进入临界区；

若mutex=0，表示有一个进程进入临界区；

若mutex=-1，表示一个进程进入临界区，另一个进程等待进入。

**4-11在如图4.33所示的进程流程图中，有五个进程合作完成某一任务。说明这五个进程之间的同步关系，并用P,V操作实现之，要求写出程序描述。**



答：main( )

{

int s2,s3,s4,s5=0; /\*表示进程p2，p3，p4,p5能否执行\*/

cobegin

P1( );

P2( );

P3( );

P4( );

P5( );

coend

}

P1( ) P2( ) P3( ) P4( ) P5( )

{ ... { p(s2); { p(s3); { p(s4); { p(s5);

... ... ... ... ...

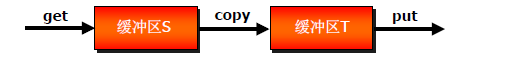
v(s2); ... .. ... ...

v(s3); } v(s5); v(s5); }

v(s4); } }

}

**4-12如图4.34所示，get,copy,put三个进程共用两个缓冲区s,t。Get进程负责不断地把输入记录送入缓冲区s中，copy进程负责从缓冲区s中取出记录复制到缓冲区t中，而put进程负责把记录从缓冲区t中取出打印。试用P,V操作实现这三个进程之间的同步，并写出程序描述。**



答：main( )

{

int s,t=0; /\*表示进程get,copy，put能否执行\*/

cobegin

Get( );

Copy( );

Put( );

coend

}

Get( ) Copy( ) Put( )

{ p(s); { p(t); { ...

.... .... v(t);

} v(s); }

}

**4-13 什么是进程的互斥，什么是进程的同步？同步和互斥这两个概念有什么联系与区别？**

答：互斥：多个操作不能在同一时间，同一临界区里执行。

同步：并发进程在一些关键点上可能需要互相等待与互通消息

互斥可以说是同步的特殊情况，通过锁和信号灯机制可以实现同步与互斥。

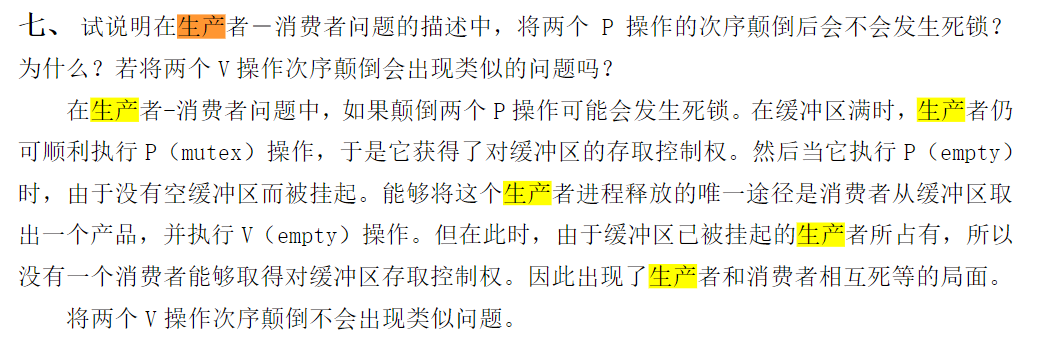
**4-18 用于进程控制的原语主要有哪几个？**

答：主要有：进程创建，进程撤销，进程阻塞，进程唤醒。

**4-19什么是线程，线程与进程有什么区别？**

答：线程是进程的进一步划分，线程也称为轻量进程。

## 第5章



**5-5, p进程共享m个同类资源，每一个资源在任一时刻只能供一个进程使用，每一进程对任一资源都只能使用一有限时间，使用完便立即释放，并且每个进程对该类资源的最大需求量小于该类资源的数目。设所有进程对资源的最大需求数目之和小于p+m.试证：在该系统中不会发生死锁。**

证：假设每个进程最多请求Xi(1<i<p)个资源，则根据题意有，

X1+X2+...+Xp-1+Xp<p+m

X1+X2+...+Xp-1+Xp - p<m

(X1-1)+(X2-1)+...+(Xp-1-1)+(Xp-1)<m

(X1-1)+(X2-1)+...+(Xp-1-1)+(Xp-1)+1<m+1

(X1-1)+(X2-1)+...+(Xp-1-1)+(Xp-1)+1<m

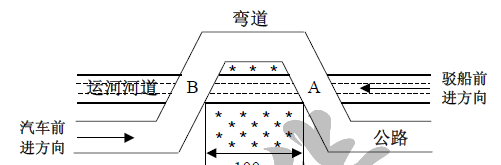
这说明在最坏情况下，每个进程均还差一个资源，而此时系统中还有一个没被分配的可用资源。将它分配给任何一个进程，都可以使该得到全部资源的 进程运行结束而释放其占有的资源，并将释放的资源分配给其它的进程，使其 它进程都能运行结束，系统不会发生死锁。 证毕。

**5-6 图5.9表示一带闸门的运河，其上有两家吊桥。吊桥坐落在一条公路上，为使该公路避开一块沼泽地而令其横跨运河两次。运河和公路的交通都是单向的。运河上的基本运输由驳船担负。在一般驳船接近吊桥A时就拉汽笛警告，若吊桥上无车辆，吊桥就吊起，直到驳船尾部通过此桥为止。对吊桥B也按同样次序处理。**

**一艘典型驳船的长度为200m,当它在河上航行时是否会产生死锁？若会，其理由是什么？**

**如何能克服一个可能的死锁？请提出一个防止死锁的办法。**

**如何利用信号灯的P,V操作，实现车辆和驳船的同步？**



（1）驳船长 200 米，当驳船通过了A桥，其船头到达B桥，请求B桥吊起，而此时它的尾部仍占据 A 桥。若这个时候 B 桥上及 B 桥到 A 桥之间的公路上都被汽车占据，而汽车又要求通过 A 桥。这样驳船和汽车都无法前进，形成死锁的局面。

（2）可以有以下两种方法：

c资源的静态分配。即进程把它所需要的所有资源在运行前提前申请，系统把它所需要的全部资源一次性都分配给它。也就是说，这时把 A 桥和 B 桥看成一个资源。打破了产生死锁的四个必要条件之一的部分分配条件。

d可以规定资源按序申请和分配，从而破坏了死锁的循环等待条件，防止死锁的发生。规定如下：B 桥的序号小于 A 桥的序号，驳船和汽车都必须先申请序号小的资源 B 桥，申请得到满足后，再申请序号大的资源 A 桥。

（3）算法如下：

c设置两个互斥信号量 mutexa，mutexb，用来实现驳船和汽车对 A 桥和对 B 桥的互斥使用；设置一个共享变量 count，用来记录当前占用 A 桥和 B 桥的汽车数并设置互斥信号量 mutex，用来实现汽车对共享变量 count 的互斥访问。

Main( ){

int mutexa, mutexb, mutex, count

mutexa=1;

mutexb=1;

mutex=1;

count=0;

cobegin

bargei; //i=1,2,..,m

carj; //j=1,2,..,n

coend

}

bargei(){

.....

P(mutexb);

P(mutexa);

吊起 B 桥;

吊起 A 桥;

驳船通过 A 桥；

放下 A 桥;

驳船通过 B 桥;

放下 B 桥;

V(mutexa);

V(mutexb);

.....

}

carj(){

......

P(mutex);

count++;

if(count==1)

{

P(mutexb);

P(mutexa);

}

V(mutex);

汽车通过 B 桥;

汽车通过 AB 段公路;

汽车通过 A 桥；

P(mutex);

count--;

if(count==0)

{

V(mutexb);

V(mutexa);

}

V(mutex);

.....

}

d设置两个互斥信号量mutexa，mutexb，用来实现驳船和汽车对A桥和对B桥的互斥使用；设置两个共享变量counta和countb，分别用来记录A桥和B桥上的汽车数并设置互斥信号量mutex1和mutex2，用来实现汽车对共享变量counta和countb的互斥访问。

Main( ){

int mutexa, mutexb, mutex1, mutex2, counta, countb;

mutexa=1;

mutexb=1;

mutex1=mutex2=1;

counta=countb=0;

cobegin

bargei; //i=1,2,...,m

carj; //j=1,2,...,n

Coend

}

bargei(){

.....

P(mutexb);

吊起 B 桥;

P(mutexa);

吊起 A 桥;

驳船通过 A 桥；

放下 A 桥;

V(mutexa);

驳船通过 B 桥;

放下 B 桥;

V(mutexb);

....

}

carj(){

....

P(mutex2);

countb++;

if(countb==1)

P(mutexb);

V(mutex2);

汽车通过 B 桥;

P(mutex2);

countb--;

if(countb==0)

V(mutexb);

V(mutex2);

汽车通过 AB 段公路;

P(mutex1);

counta++;

if(counta==1)

P(mutexa);

V(mutex1);

汽车通过 A 桥；

P(mutex1);

counta--;

if(counta==0)

V(mutexa);

V(mutex1);

.....

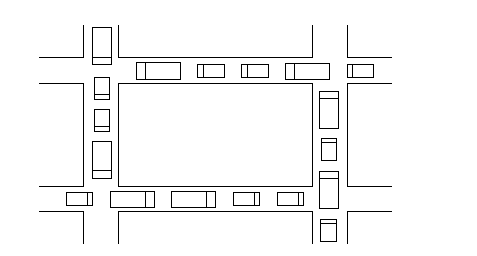
}

**5-7 讨论下图描述的交通死锁的例子（设个方向上的汽车是单线的，直线行驶）：**

**(1)对于死锁的四个必要条件中的哪些条件在此例中是适用的？**

**(2)提出一个简单的原则，它能避免死锁。**

**(3)若用计算机实现交通的自动管理，请用信号灯上的V,P操作来实现各方向上汽车行驶的同步。**



答：（1）路口是共享资源。

c 互斥条件：路口必须互斥使用，即汽车对它所需要的路口是排他性控制的。

d 不剥夺条件：汽车一旦占用了路口，除非自己让出路口，别人无权剥夺。

e 部分分配条件：每个方向的车队都占有一个路口，同时因申请新路口而等待。

f 环路等待条件：占有路口的车都在等待其它车占有的路口，循环等待。

（2）可以在每个路口设置红绿灯进行控制：绿灯亮时，南北方向的车可以通行，东西方向的车禁止通行；当红灯亮时，东西方向的车可以通行，而南北方向的车禁止通行。

## 第6章

**6-2 某系统的进程状态变迁图如图6.9所示（设该系统的进程调度方式为非剥夺式），请说明：**

1. **一个进程发生变迁3的原因是什么？发生变迁2、变迁4的原因又是什么？**

答：变迁3：请求I/O设备 ，变迁2：超时间片，变迁4：I/O完成。

**（2）下述因果变迁是否会发生，如果有可能的话，在什么情况下发生？**

**（a）2->1；（b）3->2；（c）4->5；（d）4->2；（e）3->5**

答：<1><5>不能发生，<2>在该进程位于就绪队列首部时<3>在I/O完成后<4>在某进程剥夺处理机时

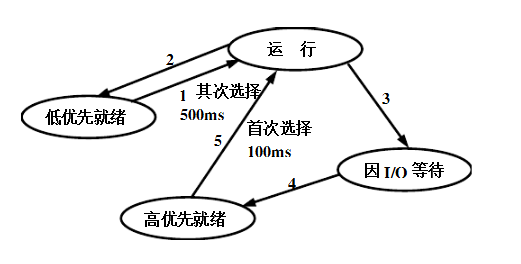


图6.9

**6-3 某系统的设计目标是优先照顾I/O量大的进程，试画出满足此设计目标的进程状态变迁图。**



**6-4 作业调度和进程调度的任务格式什么？它们有什么联系？**

答：作业调度又称为宏观调度，其任务是对提交给系统的、存放在赋存设备上的大量作业，以一定的策略进行挑选，分配主存等必要的资源，建立作业对应的进程，使其投入运行。而进程调度的对象是进程，其任务是在进入主存的所有进程中，确定哪个进程在什么时候获得处理机，使用多长时间等。两者的联系：作业调度使该作业对应的进程具备使用处理剂的权利，而进入主存的进程，分别在什么时候真正获得处理机，这是由处理机的进程调度来决定。

**6-6 在单道披处理系统中，有下列四个作业分别用先来先服务调度算法和最短作业优先算法进行调度，哪一种算法调度性能好写？请按表6.5的格式，分别用两张表正确填补表中为填写的各项。**

先来先服务调度算法

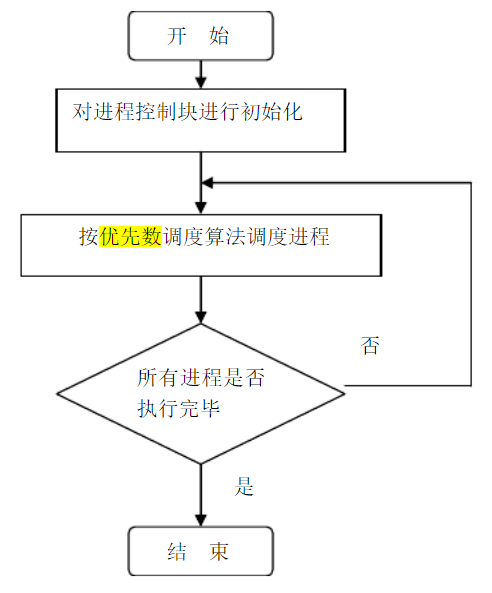
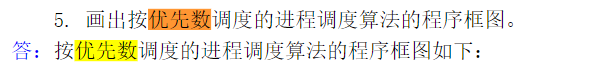
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 作业 | 进入系统时间 | 执行时间 | 开始时间 | 完成时间 | 周转时间 | 带权周转时间 |
| 1 | 10.00 | 2.00 | 10.00 | 12.00 | 2.00 | 1 |
| 2 | 10.10 | 1.00 | 12.00 | 13.00 | 2.9 | 2.9 |
| 3 | 10.25 | 0.25 | 13.00 | 13.25 | 3.00 | 12 |
| 4 | 9.50 | 0.20 | 9.50 | 9.70 | 0.20 | 1 |
| 平均周转时间 t=2.03 h  平均带权周转时间 w=3.98 h | | | | | | |

最短作业优先调度算法

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 作业 | 进入系统时间 | 执行时间 | 开始时间 | 完成时间 | 周转时间 | 带权周转时间 |
| 1 | 10.00 | 2.00 | 10.00 | 12.00 | 2.00 | 1 |
| 2 | 10.10 | 1.00 | 12.25 | 13.25 | 3.15 | 3.15 |
| 3 | 10.25 | 0.25 | 12.00 | 12.25 | 2.00 | 8 |
| 4 | 9.50 | 0.20 | 9.50 | 9.70 | 0.20 | 1 |
| 平均周转时间 t=1.84 h  平均带权周转时间 w=3.29 h | | | | | | |

**6-7 线程调度的主要任务是什么？**

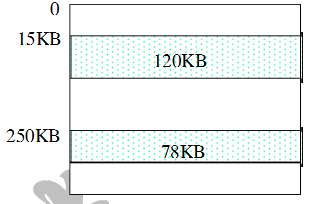
答：为了提高计算机并行处理能力。



## 第7章

**7-7 如图7.43所示，主存中有两个空白区。现有这样一个作业序列：作业1要求50KB，作业2要求60KB，作业3要求70KB。若用首次适应算法和最佳适应算法来处理这个作业序列，试问哪一种算法可以分配得下，为什么？**

**图7.43**



答：用首次适应法首先把120KB的空白区分配50KB的空间给作业1，分割后还剩70KB的空白区，再将其分配给作业2，剩下10KB的空白区。起始地址为250KB的空白区（78KB）可以满足作业3的需求，分割后还剩8KB的空白区。因此首次适应法可以吞吐此作业序列。

用最佳适应法，则先分配78KB的空白区给作业1，还剩28KB的空白区，不能满足作业2的需求，因此分配120KB的空白区给作业2，还剩60KB的空白区。此时系统中有大小为28KB和60KB的两个空白区，它们均不能满足作业3的需求。

因此最佳适应法不能吞吐此作业序列。

**7-8 已知主存有256KB容量，其中OS占用低址20KB，可以有这样一个作业序列：**

**作业1    要求    80KB**

**作业2    要求    16KB**

**作业3    要求    140KB**

**作业1    完成**

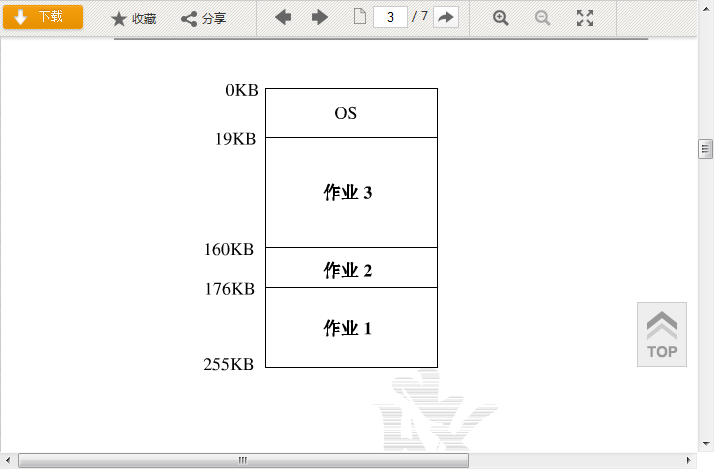
**作业3    完成**

**作业4    要求    80KB**

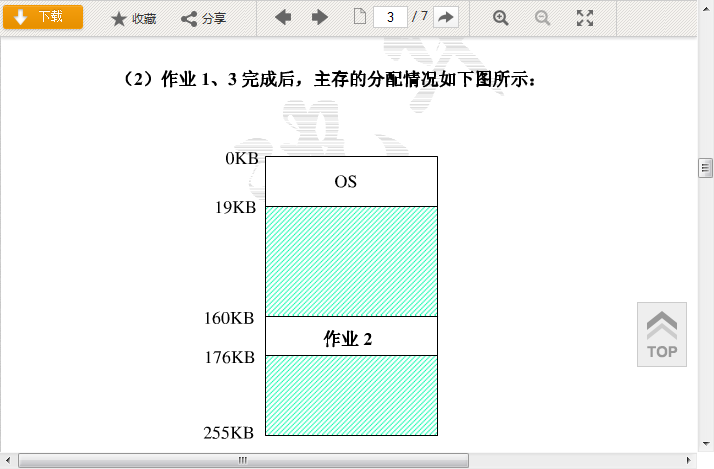
**作业5    要求    120KB**

**试用首次适应算法和最佳适应算法分别处理上述作业序列（在存储分配时，从空白区高址处分割作为已分配区），并完成以下各步：**

1. **画出作业1、2、3进入主存后，主存的分配情况。**



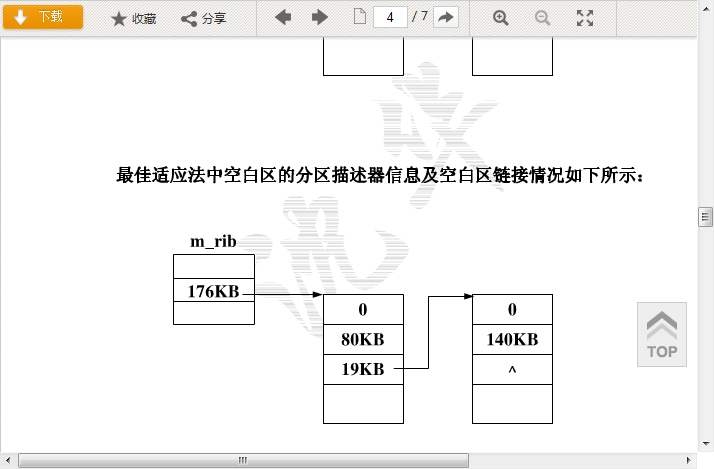
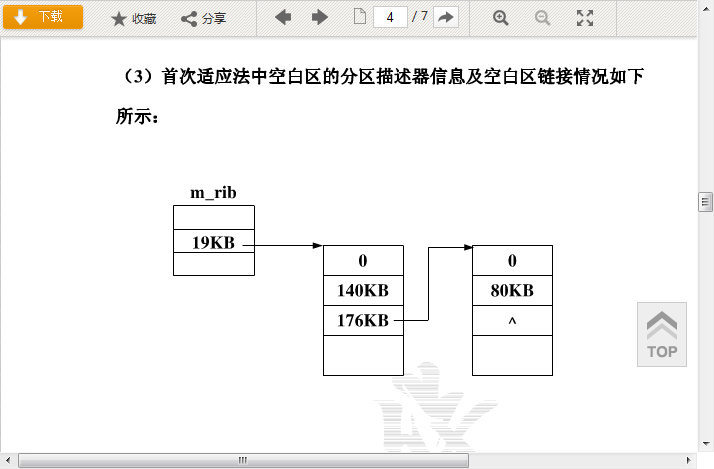
1. **作业1、3完成后，画出主存分配情况。**



**（3） 画出两种算法中空白区的分区描述器信息（假定分区描述器所**

**需占用的字节数已包含在作业所要求的主存容量中）及空白区链接情况。**

首次适应： 最佳适应法：



1. **哪种算法对该作业序列而言是合适的？**

答：最佳适应法是合适的

**7-10 已知主存容量为64KB，某一作业A的地址空间如图7.40所示，它的4个页面（页面大小为1KB）0、1、2、3被分配到主存的2、4、6、7块中，要求并回答**

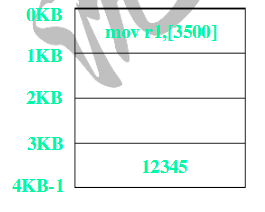
1. **画出作业A的页面映射表。**



**（2）当200号单元处有一条指令“mov r1,[3500]”执行时，如何进行正确的地址变换，以使3500处的内容12345装入r1中，要求用图画出地址变换过程并给出最终的物理地址。**

答：因为每页大小为1KB=1024字节，而3500=3\*1024+428，可知逻辑地址3500对应的页号为3，页内地址为428.根据页号检索页表可知对应的物理块号为7，所以物理地址为：7\*1024+428=7596

**图7.44**



## 第8章

**8-1什么是“设备独立性”？引入这一概念有什么好处？**

答：所谓设备独立性是指，用户在编制程序时所使用的设备与实际使用的设备无关，也就是在用户程序中仅使用逻辑设备名。引入设备独立性，可使应用程序独立于物理设备。独立性可使用户程序独立于某一特定的物理设备。此时，进程只需用逻辑设备名去请求使用某类设备。当系统中有多台该类设备时，系统可将其中的任一台设备分配给请求进程，而不必局限于某一指定设备。这样，可以显著地提高资源的利用率和可适应性。独立性还可以使用户程序独立于设备类型。例如，在进行输出时，既可以利用显示终端进行输出，也可以利用打印机进行输出。有了这种适应性，就可以很方便地实现输出重定向。类似地可以实现输入重定向。

**8-4什么是缓冲？引入缓冲的原因是什么？**

答：缓冲是两种不同速度的设备之间传输信息时平滑传输过程的常用手段。引入了缓冲技术的原因有：

（**1**）为了进一步缓和 **CPU** 和 **I/O** 设备之间速度不匹配的矛盾。

（**2**）为了减少中断次数和 **CPU** 的中断处理时间。如果没有缓冲，慢速 **I/O** 设备每传送一个字节就要产生一个中断，**CPU** 必须处理该中断。如果采用了缓冲，则慢速 **I/O** 设备将缓冲区填满时，才向 **CPU**发出中断，从而减少了中断次数和 **CPU** 的中断处理时间。

（**3**）为了解决 **DMA** 或通道方式下数据传输的瓶颈问题。**DMA**或通道方式都适用于成批数据传输，在无缓冲的情况下，慢速 **I/O** 设备只能一个字节一个字节的传送信息，这造成了 **DMA** 或通道方式数据传输的瓶颈。缓冲区的设置适应了 **DMA** 或通道方式的成批数据传输方式，解决了数据传输的瓶颈问题。

**8-5常用的缓冲技术有哪些？**

答：常用的缓冲技术有双缓冲、环形缓冲和缓冲池。

**8-8什么是独占设备？对独占设备如何分配？**

答：独占设备是指在一段时间内只允许一个用户进程访问的设备。系统一旦把这类设备分配给某进程后，便由该进程独占直到使用完后释放。多数低速 **I/O** 设备都属于独占设备。如打印机就是典型的独占设备。独占设备应采用独占分配方式，即将一个独占设备分配给某进程后便一直由它独占，直到该进程完成或释放该设备时，系统才能将该设备分配给其他进程。

**8-9什么是共享设备？对共享设备如何分配？**

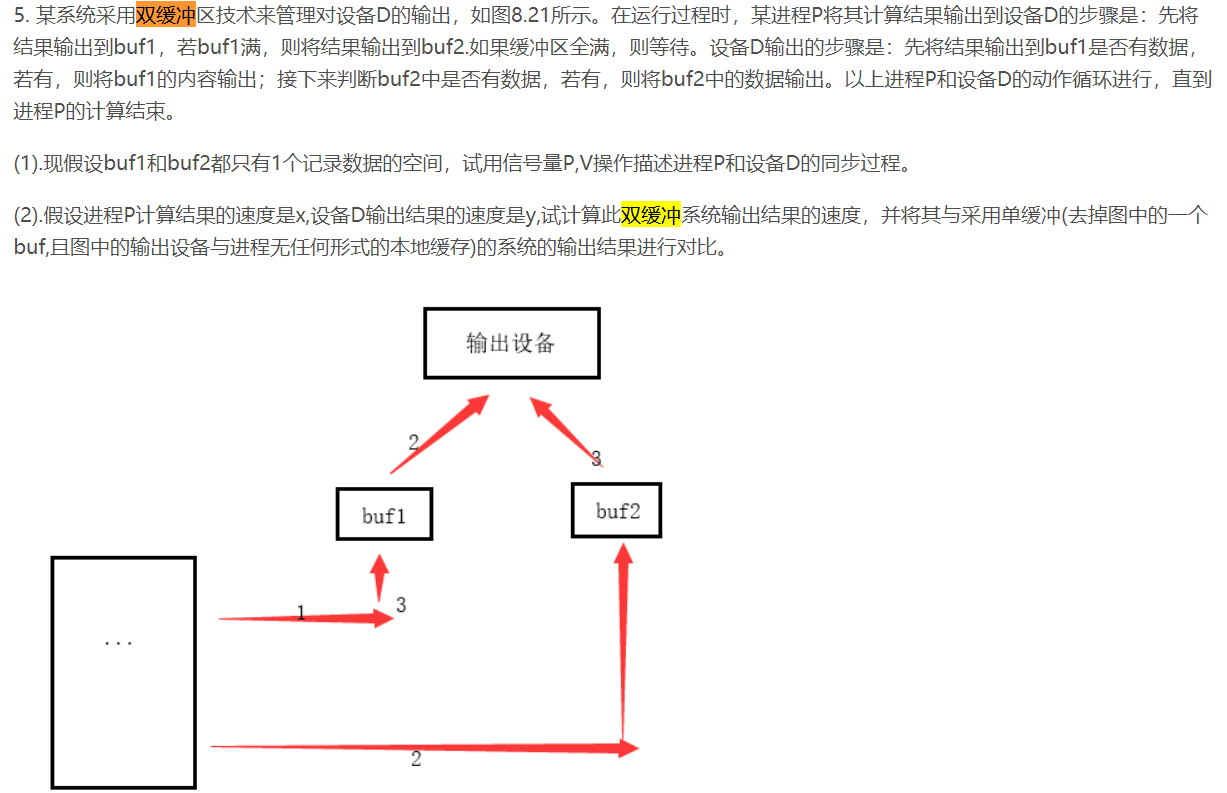
答：共享进程是指在一段时间内允许多个进程同时访问的设备。如磁盘就是典型的共享设备，若干个进程可以交替地从磁盘上读写信息。对共享设备可将其同时分配给多个进程使用。共享分配方式显著提高了设备的利用率，但对设备的访问需进行合理的调度。

**8-10什么是虚拟设备技术？什么是虚拟设备？如何进行虚拟分配？**

答：所谓虚拟设备技术，是在一类物理设备上模拟另一个物理设备的技术，是将独占设备转换为共享设备的技术。目前最广泛流行的虚拟设备技术是 **SPOOLing** 技术。

虚拟设备是指通过虚拟技术将一台独占设备变换为若干台逻辑设备，供若干个用户进程使用，通常把这种经过虚拟技术处理后的设备成为虚拟设备。引入虚拟设备的目的是为了克服独占设备所具有的速度较慢、资源的利用率较低的缺点，以提高设备的利用率。

虚拟分配是针对虚拟设备而言的。当进程申请独占设备时，由系统分配给它共享设备（如磁盘）上的一部分存储空间；当进程要与设备交换信息（以输出为例）时，系统就将要交换的信息存放到这部分存储空间中；在适当的时候，系统再将存储空间中的信息传送到独占设备上。



**int empty1=1, empty2=1, full1=0, full2=0;**

**//其中empty1=1表示buf1有位置, full1=0表示buf1没有数据, 2同理**

**int mutex1=1, mutex2=1;**

**//mutex=1, mutex2=1表示每个缓冲区只有一个资源可使用，并且现在均处于未使用的状态**

**int main(){**

**int empty1=1, empty2=1, full1=0, full2=0;**

**int mutex1=1, mutex2=1;**

**cobegin**

**P();D();**

**coend**

**return 0;**

**}**

**P(){**

**while(true){**

**if(empty1 >= 1)**

**{**

**...**

**P(empty1);//buf1是否有位置**

**P(mutex1);**

**向buf1放入数据**

**V(mutex1);//结束放置**

**V(full1);//buf1现有数据**

**...**

**}**

**else if(empty2 >= 1)**

**{**

**...**

**P(empty2);//buf2是否有位置**

**P(mutex2);**

**向buf2放入数据**

**V(mutex2);**

**V(full2);//buf2现有数据**

**...**

**}**

**}**

**}**

**//D 写错了**

**D(){**

**while(true){**

**if(full1 >= 1)**

**{**

**...**

**P(empty1);//判断buf1是否有数据**

**P(mutex1);**

**取出buf1数据**

**V(mutex1);**

**P(full1);**

**...**

**}**

**else**

**{**

**...**

**P(empty2);**

**P(mutex2);**

**取出buf2数据**

**V(mutex2);**

**P(full2;)**

**...**

**}**

**}**

**}**



## 第9章

**9-2什么是文件系统？其主要功能是什么？**

答：文件系统是操作系统用于明确磁盘或分区上的文件的方法和数据结构；即在磁盘上组织文件的方法。也指用于存储文件的磁盘或分区，或文件系统种类。操作系统中负责管理和存储文件信息的软件机构称为文件管理系统，简称文件系统。文件系统是对文件存储器空间进行组织和分配，负责文件存储并对存入的文件进行保护和检索的系统。具体地说，它负责为用户建立文件，存入、读出、修改、转储文件，控制文件的存取，当用户不再使用时撤销文件等

**9-3文件的逻辑结构有哪2种形式？**

答：一种是无结构的流式文件，是指对文件内信息不再划分单位，它是依次的一串字符流构成的文件。

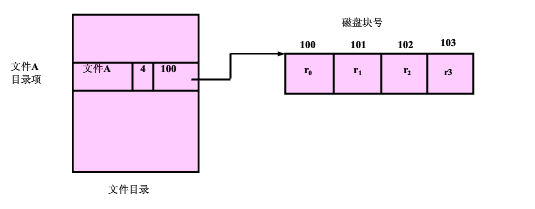
另一种是有结构的记录式文件, 是用户把文件内的信息按逻辑上独立的含义划分信息单位，每个单位称为一个逻辑记录（简称记录）。所有记录通常都是描述一个实体集的，有着相同或不同数目的数据项，记录的长度可分为定长和不定长记录两类

**9-4对文件的存取有哪2种基本方式？各有什么特点？**

答：（1）顺序存取 顺序存取是按照文件的逻辑地址顺序存取

（2）直接存取（随机存取法） 直接存取文件对读或写块的次序没有限制

**9-5设文件A按连续文件构造，并由4个逻辑记录组成（每个逻辑记录的大小与磁盘块大小相等，均为512B）。若第一个逻辑记录存放在第100号磁盘块上，试画出此连续文件的结构**



9-5 设文件A按连续文件构造，并由四个逻辑记录组成 (每个逻辑记录的大小与磁盘块大小相等，均为512B) 。若第一个逻辑记录存放在第100号磁盘块上，试画出此连续文件的结构。

答：连续文件的结构如下图：

磁盘块号

103

102

101

100

文件A 4 100

r0 r1 r2 r3

文件目录

9-6 设文件B按串联文件构造，并由四个逻辑记录组成 (其大小与磁盘块大小相等，均为512B)。这四个逻辑记录分别存放在第100、157、66、67号磁盘块上，回答如下问题。

(1) 画出此串联文件文件的结构，

(2) 若要读文件B第1560字节处的信息，问要访问哪一个磁盘块? 为什么?

(3) 读文件B第1560字节处的信息需要进行多少次I/O操作? 为什么?

(1) 答：此串联文件结构如下图所示。

文件A 100

文件目录

150

157

100

r0

150

66

157

r1

磁盘块号

150

67

66

150

∧

67

r3

r1

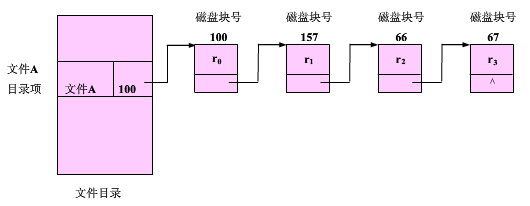
r2

(2) 答：1560/512=3余24，因此文件第1560逻辑字节在r3逻辑块上，该逻辑块被分配在67号磁盘块上。

(3) 答：要访问67号磁盘块，需要先找到文件目录，然后依次访问100、157和66号磁盘块，最后读取67号磁盘块。因此若文件已打开 (文件目录信息已在内存中) 需要4次I/O操作，文件未打开需要5次I/O操作。

**9-6设文件B按串联文件构造，并由4个逻辑记录组成（每个逻辑记录的大小与磁盘块大小相等，均为512B）。这4个逻辑记录分别存放在第100、157、66、67号磁盘块上，回答如下问题：**

1. **画出此串联文件的结构；**



1. **若要读文件B第1560字节处的信息，问要访问哪一个磁盘块？为什么？**

答：1560/512=3 访问逻辑号3，磁盘块67

1. **读文件B第1560字节处的信息需要进行多少次I/O操作？为什么？**

答：4次

**9-9试分别说明一级文件索引结构、二级文件索引结构是如何构造的。**

答：文件目录(file directory）为每个文件设立一个表目。文件目录表目至少要包含文件名、文件内部标识、文件的类型、文件存储地址、文件的长度、访问权限、建立时间和访问时间等内容。

文件目录（或称为文件夹）是由文件目录项组成的。文件目录分为一级目录、二级目录和多级目录。多级目录结构也称为树形结构，在多级目录结构中，每一个磁盘有一个根目录，在根目录中可以包含若干子目录和文件，在子目录中不但可以包含文件，而且还可以包含下一级子目录，这样类推下去就构成了多级目录结构。

**9-17什么是文件共享？试简述文件共享的实现方法。**

答：文件共享是指主动地在网络上（互联网或小的网络）共享自己的计算机文件。一般文件共享使用P2P模式，文件本身存在用户本人的个人电脑上。大多数参加文件共享的人也同时下载其他用户提供的共享文件。有时这两个行动是连在一起的

**9-21常用的文件操作命令有哪些？**

答：最常用的命令列表，下面列出十个在使用linux过程中使用频率最高的命令。

cat，显示文件内容。

cd，改变目录路径。

cp，复制文件。

find，查找文件。

grep，搜索、过滤信息。

ls，列出目录信息。

more，分页显示。

rm，删除文件或目录。

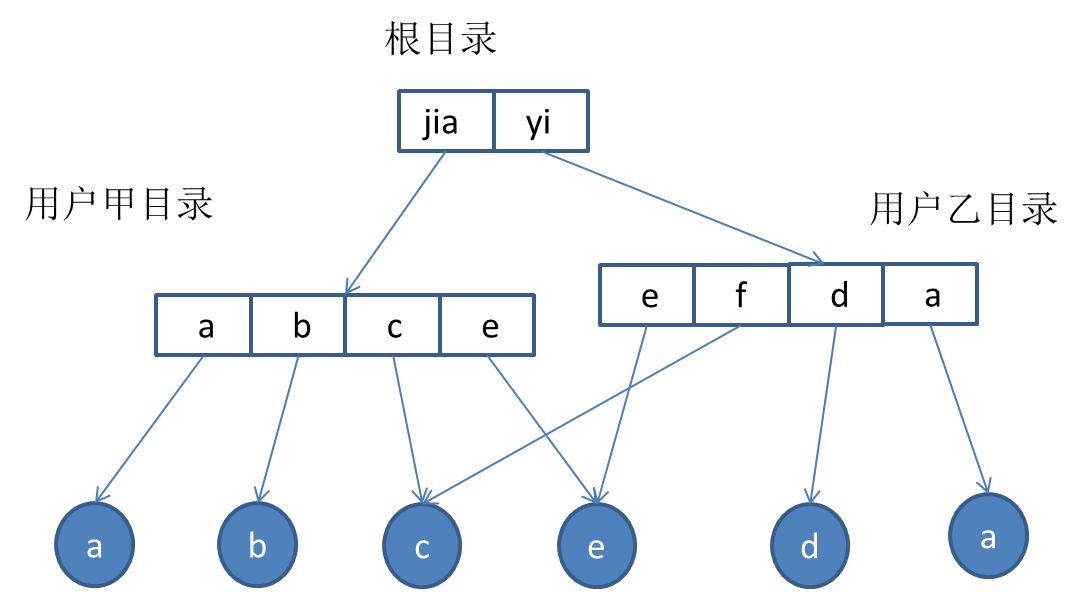
vi，调用vi文本编辑器。

who，显示登录用户信息

9-18 假设两个用户共享一个文件系统，用户甲要用到文件a、b、c、e，用户乙要用到文件a、d、e、f。已知：用户甲的文件a与用户乙的文件a实际上不是同一文件；用户甲的文件c与用户乙的文件f实际上是同一文件；甲、乙两用户的文件e是同一文件。试拟定一个文件组织方案，使得甲、乙两用户能共享该文件系统而不致造成混乱。

答：如下图所示。用户甲的主目录名为jia，有四个文件，文件名为a、b、c、e。

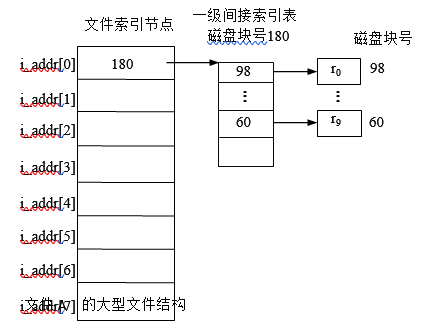
用户乙的主目录名为yi，有四个文件，文件名为a、d、e、f。



9-27设某文件A有10个逻辑块，另一文件B有500个逻辑块，试分别用UNIX 7版本的索引结构画出这两个文件的索引结构图。

答：文件A有10个逻辑块：按UNIX 7版本的索引结构则要构造大型文件结构。这时数组i\_addr[ ]用作一级间接索。

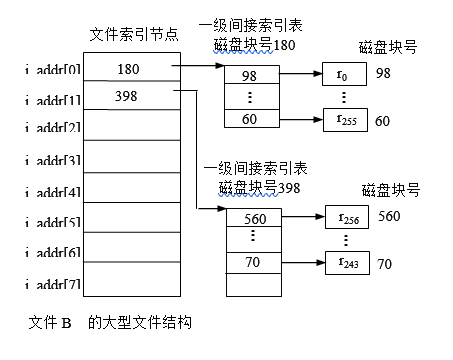
文件A的文件索引结构如下图所示 ( 磁盘块号随意设置 ) 。



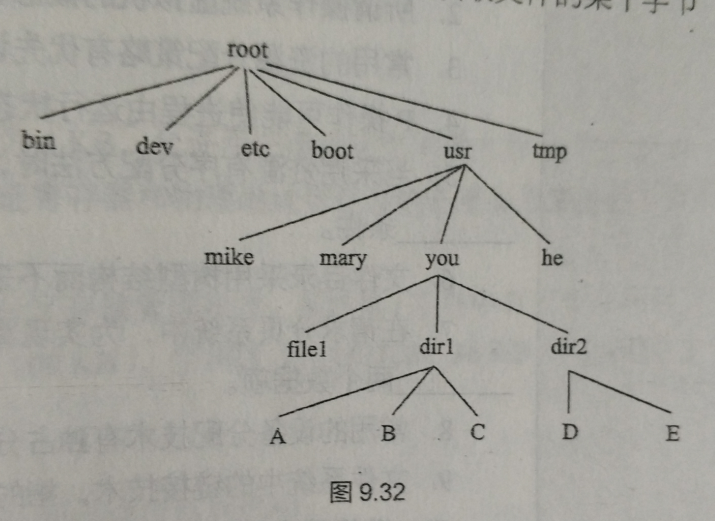
文件A 的大型文件结构

文件B有500个逻辑块：按UNIX 7版本的索引结构则要构造大型文件结构。这时数组i\_addr[ ]用作一级间接索。

500**/**256 = 1，余244 文件B的文件索引结构如下图所示 ( 磁盘块号随意设置 ) 。



9.38某文件系统中，辅存为硬盘，物理块大小为512B。有文件A，包含590个逻辑记录，每个记录占255B，每个物理块存放2个逻辑记录。文件A所在的目录如图9.32所示，此树型目录结构由根目录节点、作为目录文件的中间节点和作为信息文件的叶节点组成。每个目录项占127B，每个物理块放4个目录项，根目录的第一块常驻主存。回答如下问题：



(1)若文件采用串连结构，链接字占2B，那么，要将A读入主存，至少要存取几次硬盘?为什么?

(2) 若文件采用连续结构，那么，要将A的第480号记录读入主存，至少要存取几次硬盘?为什么?

答：（1）第一次读根目录的第二块， 获得usr的地址；

第二次读usr目录的第一块，获得you的地址；

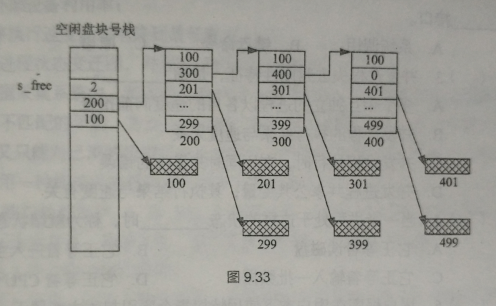
第三次读you目录的第一块，获得dir1的地址；

第四次读dir1目录的第一块，获得A的地址；

第五次开始遍历串联的A的逻辑记录，590个逻辑记录占用295个磁盘块，需要存取295次。

（2）连续结构可随机访问，第四次获得A的地址后就可计算得到A的第480号记录的地址，然后直接第五次访问该记录。

9-39某系统采用成组链接法来管理系统盘的空闲存储空间， 目前，磁盘的状态如图9.33所示。回答如下问题:



(1)该磁盘中目前还有多少个空闲盘块?

(2)系统需要给文件F分配3个磁盘块，试给出将被分配出去的磁盘块号。

(3)接着(在创建文件F之后),系统要删除另一个文件，并回收它所占的5个盘块，它们的盘块号依次为700、711、703、788、701, 试给出回收后的盘块链接情况。

答：

（1）301个空闲盘块。

（2）100，200，299。

（3）s\_free=4, s\_free[0]=711, s\_free[1]=703 s\_free[2]=788 s\_free[3]=701, 原299所在位置存放盘块号700。