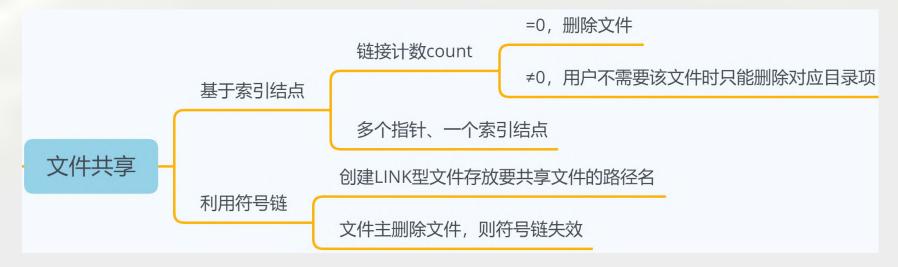
考点三:

文件的物理结构

文件的共享与保护

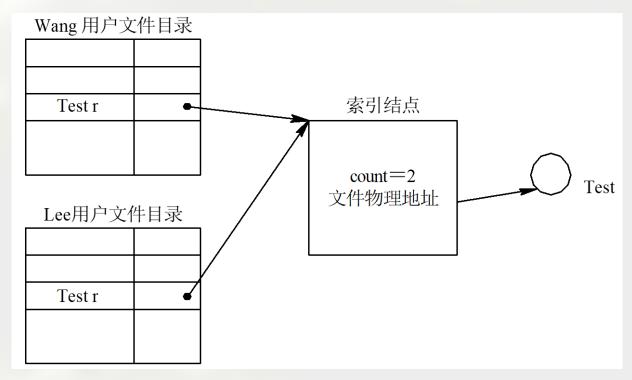




□ 现代操作系统必须提供文件共享手段,允许多个用户/进程共享同一份文件, 系统只需保留该共享文件的一份副本



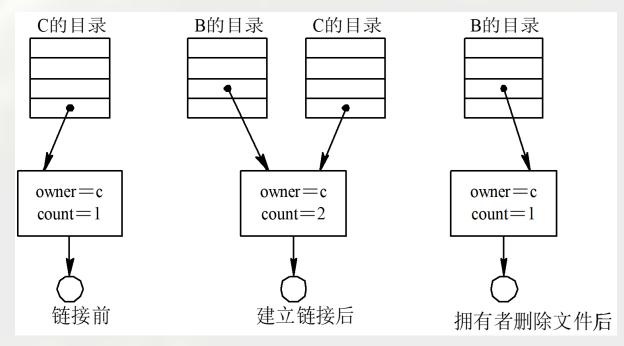
# 基于索引节点的文件共享



□ 索引结点中设置一个链接计数变量count,用于表示链接到本索引结点上 的用户目录项数。



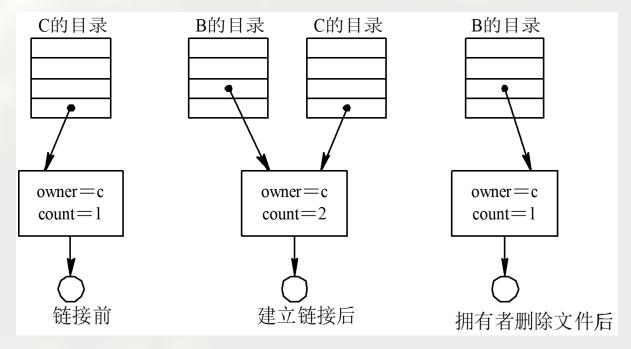
# 基于索引节点的文件共享



□ 若count = 2, 说明此时有两个用户目录项链接到该索引结点上, 或者说是 有两个用户在共享此文件。



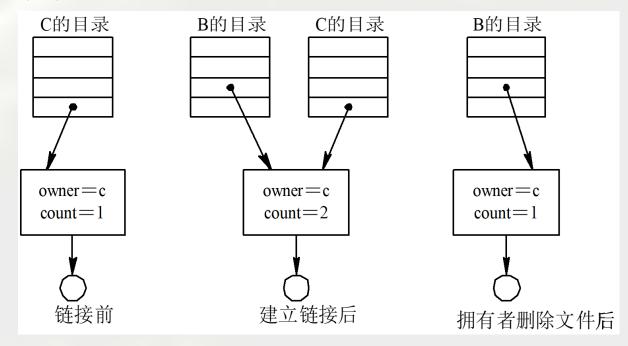
### 基于索引节点的文件共享



□ 若某个用户决定"删除"该文件,则只是要把用户目录中与该文件对应的 目录项删除,且索引结点的count值减1。



# 基于索引节点的文件共享



- □ 若count>0, 说明还有别的用户要使用该文件, 暂时不能把文件数据删除, 否则会导致指针悬空。
- □ 当count = 0时系统负责删除文件。



# 基于符号链接的文件共享





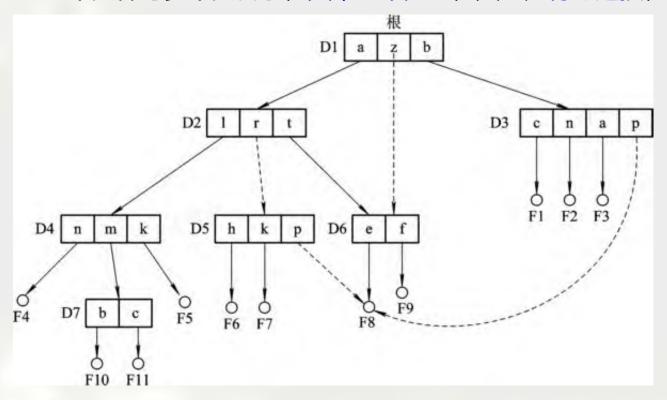
### 基于符号链接的文件共享

- □ 在利用符号链方式实现文件共享时, 只是文件主才拥有指向其索引结点的指针;
- □ 共享该文件的其他用户,则只有该文件的路径名,并不拥有指向其索引结点的 指针。
- □ 就不会发生在文件主删除一共享文件后留下一悬空指针的情况。



# 基于符号链接的文件共享

□ 一个文件可多个父目录,其中一个为主,其它为符号连接,下图的虚线表示链接

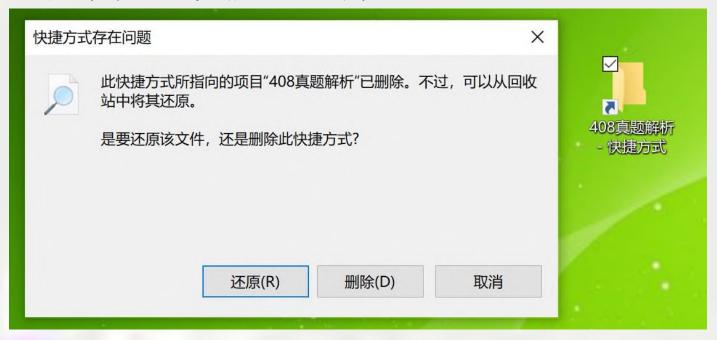


□ count表示共享文件的索引结点的个数;符号链接不改变count的值。



### 基于符号链接的文件共享

□ 当文件的拥有者把一个共享文件删除后, 其他用户试图通过符号链去访问一个 已被删除的共享文件时,会因系统找不到该文件而使访问失败,于是再将符号 链删除,此时不会产生任何影响。





现代操作系统必须提供文件共享手段,允许多个用户/进程共享同一份文件,系统只需保留该共享文件的一份副本

#### □ 影响文件安全性的主要因素

人为因素:人有意或无意的行为,可能会使系统中的数据遭受破坏或丢失

系统因素:系统某部分出现异常导致的数据破坏或丢失,或磁盘出现故障, 影响难以估量

> 自然因素: 随着时间的推移, 存放于磁盘的数据会逐渐消失



文件的保护



保护域机制: 进程只能在保护域内执行操作, 且只能访问具有访问权的对象



# 文件的保护

访问类型

在每个文件的FCB(或索引结点)中增加一个访问控制列表(Access-Control List, ACL) ,该表中记录了各个用户可以对该文件执行哪些操作。

读: 从文件中读数据

写: 向文件中写数据

执行:将文件装入内存并执行

添加:将新信息添加到文件结尾部分

删除: 删除文件, 释放空间

列表清单:列出文件名和文件属性



### 文件的保护

在每个文件的FCB (或索引结点) 中增加一个访问控制列表 (Access-Control List, ACL) ,该表中记录了各个用户可以对该文件执行哪些操作。

用户	读	写	执行	添加	删除	列表 清单
Admin	1	1	1	1	1	1
Owner	1	1	1	0	0	1
guest	0	0	0	0	0	0

1: 表示具有权限

0: 表示没有权限

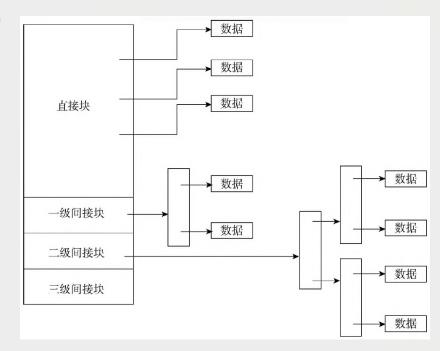
# 【大显身手】

- 1. 某文件系统采用多级索引的方式组织文件的数据存放,假定在文件的i\_node中设有13个地址项,其中直接索引10项,一次间接索引项1项,二次间接索引项1项, 三次间接索引项1项。数据块大小为4 KB,磁盘地址用4B表示,问:
- (1)这个文件系统允许的最大文件长度是多少?

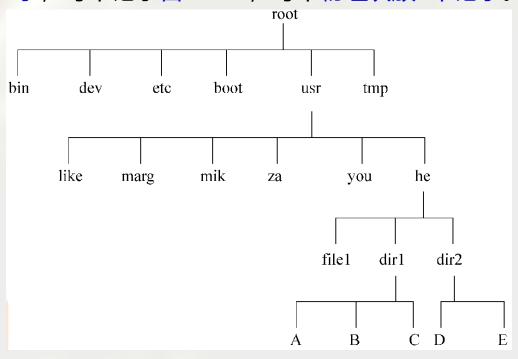
【解析】(1)数据块大小为4KB,磁盘地址为4B,则每个物理块可存储的地址数为4KB/4B=1024。

最大文件的物理块数可达10+1024+1024<sup>2</sup> +1024<sup>3</sup>,每个物理块大小为4 KB, 因此总长度为(10+1024+1024<sup>2</sup>+1024<sup>3</sup>)×4 KB=40 KB+4 MB+4 GB+4 TB, 所以这个文件系统允许的最大文件长度是4 TB+4 GB+4 MB+40 KB,约为4TB。

- 1. 某文件系统采用多级索引的方式组织文件的数据存放,假定在文件的i\_node中设有13个地址项,其中直接索引10项,一次间接索引项1项,二次间接索引项1项,
- 三次间接索引项1项。数据块大小为4 KB, 磁盘地址用4B表示, 问:
- (2)一个2GB大小的文件,在这个文件系统中实际占用多少空间?
- (2)该文件占用2 GB/4 KB=512×1024个数据块。
- □ 直接索引指向10个数据块;
- □ 一级间接索引指向1024个数据块,需1个索引块;
- □ 二级间接索引项指向512×1024-10-1024个数据块,
- □ 共需1+(512×1024-10-1024)/1024=512个索引块。
- □ 索引块所占空间大小为(1+512)×4 KB=2 MB+4 KB。
- □ 另外每个文件使用的i node数据结构占13×4 B=52 B,
- □ 因此该文件实际占用磁盘空间大小为2 GB+2MB+4 KB+52 B。



2. 某个文件系统中,外存为硬盘,物理块大小为512 B,有文件A包含598个记录,每个记录占255 B,每个物理块放2个记录。文件A所在的目录如图所示。



文件目录采用多级树形目录结构,由根目录节点、作为文件目录的中间节点和作为信息文件的树叶组成,每个目录项占127B(包含文件基本信息、存取控制信息及使用信息等),每个物理块放4个目录项,根目录的第一块常驻内存。

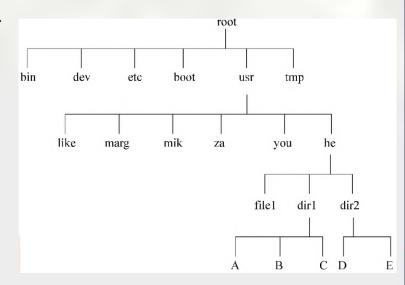
(1)若文件的物理结构采用链式存储方式,链指针地址占2B,那么要将文件A读入内存,至少需要存取几次硬盘?

【解析】(1)由于根目录的第一块常驻内存(即root所指的/bin, /dev, /etc, /boot可直接获得),因此根目录找到文件A需要5次读盘。

由255×2+2=512可知,一个物理块在链式存储结构下可放2个记录及下一个物理块地址,

而文件A共有598个记录,因此读取A的所有记录所需的读盘次数为598/2=299, 所以将文件A读到内存至少需读盘299+5=304(次)。

(598条记录需要598/2=299个数据块,每个数据块含两条记录+指针大小。)

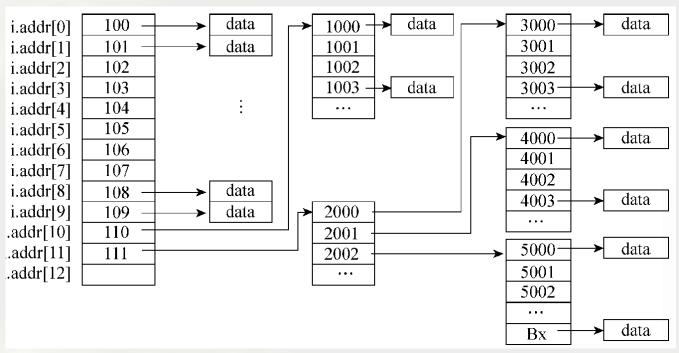


- (2)若文件为连续文件,那么要读文件A的第487个记录,至少要存取几次硬盘?
- (2)当文件为连续文件时,找到文件A同样需要5次读盘,

知道文件A的地址后通过计算,只需一次读盘即可读出第487个记录,

所以至少需要读盘5+1=6(次)。

3. 【广东工业大学 2017】UNIX系统中某记录式文件长度为3200 KB, 其磁盘存储结构如图所示。假设该文件的逻辑记录长度为256 B, 磁盘块的大小为2 KB, 磁盘的逻辑块号占4B, 又设每个索引块中的盘块号是连续的。请回答下列问题:

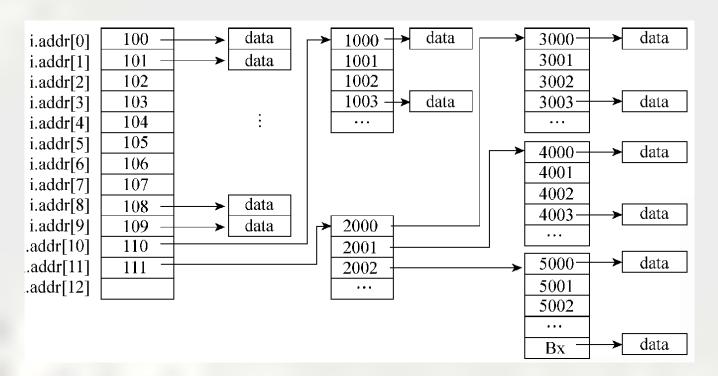


(1) 该文件占用的磁盘块数 (不含索引节点所在的盘块) 是多少? 请给出计算过程。

【解析】(1)文件长度为3200 KB, 磁盘块的大小为2KB, 该文件占有的磁盘块数为3200 KB/2 KB=1600。

(2) Bx是存储该文件的最后一个盘块号, Bx的数值为多少?请说明计算过程。

【解析】(2)因为起始块号为100,且直接地址有10个,其对应的磁盘块号为100~109;又因为一级间接磁盘块数为1000,二级间接磁盘块数为1000×1000=1000000,所以Bx的数值为100+9+1000+1000×1000=1001109。



(3) 假设某用户进程要读取该文件的第5000条记录(文件记录号从1开始编号),写出系统为完成此工作的操作过程。

【解析】(3)因为逻辑记录长度为256 B, 且要读取第5000条记录, 而第5000条记录对应的磁盘块号为5000×256/2 KB=625块,

直接地址有10个磁盘块,

一级地址2 KB/4 B=512块,

因此第5000条记录位于二级地址中,需要访问磁盘3次才可以读取数据。

4. 【浙江工商大学 2017】某文件系统采用混合索引分配方式,如图所示,有10个直接块(每个直接块指向一个数据块),1个一级间接块,1个二级间接块和1个三级间接块,间接块指向的是一个索引块,每个索引块和数据块的大小均为512字节,索引块编号大小为4字节。问:



#### (1)如只使用直接块,文件最大为多少字节?

解析 (1)由题干知,该索引节点共有10个直接块,每个直接块只有一个指向数据块的指针,因此所有的直接块加起来能够表示10个数据块的空间,也就最多能够表示10×512 B=5 KB的文件。

#### (2)在该系统中能存储的文件最大是多少?

#### 【解析】(2)

由(1)可知, 所有的直接块能够表达5 KB的空间;

- 一级间接块的大小为512 B,每个索引表项的大小为4B,故1个一级间接块有512 B/4 B=128 (个)索引表项,而每个索引表项指向一个直接块,就相当于一个数据块,故1个一级间接块能够表示128×512 B=64 KB的空间;
- 1个二级间接块指向128个一级间接块, 1个一级间接块指向128个直接块, 故1个二级间接块能够表示128×128×512 B=8 MB的空间;
- 1个三级间接块指向128个二级间接块, 1个二级间接块指向128个一级间接块, 1个一级间接块指向128个直接块, 故1个三级间接块能够表示128×128×5128 B=1 GB的空间。
- 以上结果相加,得到该索引节点能够访问的地址空间为5 KB+64 KB+8 MB+1 GB≈1 GB。

#### (3)如读取某文件第10MB字节的内容,需要访问磁盘几次?

#### 【解析】

由(2)可知,全部的直接块可以表示的空间大小为5KB,

全部的一级间接块(其实就有一个)能够表示64 KB的内容,

全部的二级间接块(其实也就有一个)能够表示8 MB的内容,

全部的三级间接块 (其实也就有一个) 能够表示1 GB的内容。所有的直接块、一

级间接块和二级间接块加起来能够表示8.064 MB的内容,

所有的直接块、一级间接块、二级间接块和三级间接块加起来能够表示1.008 GB的内容,

显然8.064 MB<10 MB<1.008 GB, 也就是说10 MB的位置正好落在三级间接块中。故而读取数据需要访问4次磁盘。

5.【南京理工大学 2016】某文件系统采用直接索引分配方式为文件分配磁盘空间,即为每个文件分配一个索引块用于记录分配给该文件的所有盘块号,文件目录项中记录该索引块的盘块号。已知硬盘大小为16 TB,盘块的大小为8KB。(1)索引块中块号最少占用多少字节?

【解析】(1)文件系统存储空间共有块数16TB/8 KB=2<sup>31</sup>。表示2<sup>31</sup>个块号,索引表项占(31/8)<sub>向上取整</sub>=4,索引块中块号最少占用4字节。

5.【南京理工大学 2016】某文件系统采用直接索引分配方式为文件分配磁盘空间,即为每个文件分配一个索引块用于记录分配给该文件的所有盘块号,文件目录项中记录该索引块的盘块号。已知硬盘大小为16 TB,盘块的大小为8KB。(2)该文件系统中支持的单个文件的最大长度为多少字节?

【解析】(2)由于索引块中块号占4B,于是共2<sup>32</sup>个盘块,且每个盘块大小为8 KB,于是最大的单个文件为2<sup>32</sup>×8 KB。

- 5. 【南京理工大学 2016】某文件系统采用直接索引分配方式为文件分配磁盘空间,即为每个文件分配一个索引块用于记录分配给该文件的所有盘块号,文件目录项中记录该索引块的盘块号。已知硬盘大小为16 TB,盘块的大小为8KB。
- (3)若文件A索引块中前4个盘块号依次为67,32,90,100,则文件A的字节偏移量20000对应的物理盘块号和块内位移量分别为多少?
- (3)int(20000 B/8 KB)=2, 20000 B%8 KB=3616, 文件A的字节偏移量20000 对应的物理盘块号为90, 块内位移量为3616。

谢谢大家

考点四:

磁盘管理和组织

# 考点框架



磁盘结构



磁盘调度



磁盘存储空间管理

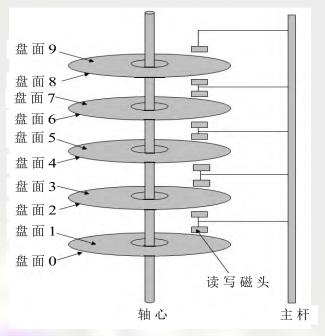
考点四: 磁盘管理和组织

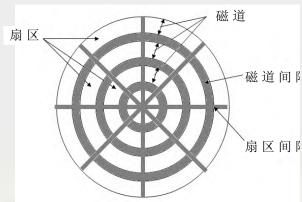
磁盘结构



# 磁盘的结构

磁盘设备可包括一个或多个物理盘片,每个磁盘片分一个或两个存储面(Surface), 每个盘面上有若干个磁道(Track),磁道之间留有必要的间隙(Gap)。为使处理简 单起见, 在每条磁道上可存储相同数目的二进制位,







### 磁盘的地址结构

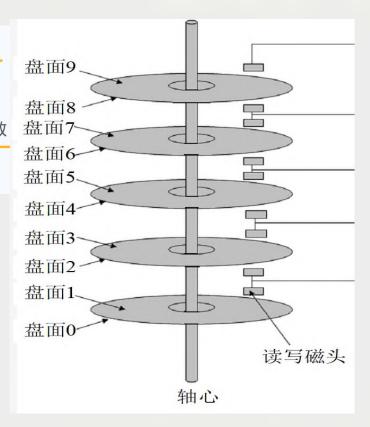
=柱面(磁道)号-盘面号-扇区号 均从0开始

磁盘地址

盘面号= (蔟号%每个柱面的簇数) /每个磁道的簇数

扇区号=扇区地址%每个磁道的扇区数

柱面号=簇号/每个柱面的簇数

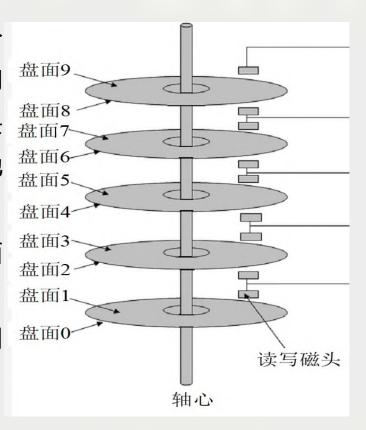




### 磁盘的地址结构

可用(柱面号,盘面号,扇区号)来定位任意一个"磁盘块"。在"文件的物理结构"小节中,我们经常提到文件数据存放在外存中的几号块,这个块号就可以转换成(柱面号,盘面号,扇区号)的地址形式。可根据该地址读取一个"块"

- ①根据"柱面号"移动磁臂,让磁头指向指定柱面
- ②激活指定盘面对应的磁头;
- ③磁盘旋转的过程中,指定的扇区会从磁头下面划过,这样就完成了对指定扇区的读/写。





# 磁盘的地址结构

如何将一个内存中的簇号来转化为磁盘上的物理地址:将簇号转化为磁盘物理地 址的过程由磁盘驱动程序完成。假设内存的簇号为b,则:

- □ 柱面号: [簇号/每个柱面的簇数]。
- □ 分析: 给出的簇号可以当做物理块的总数, 所求柱面号就是判断簇号第几个盘片 上,则需除以每个盘片的总物理块数,结果是向上取整,因为多出来的小数代表 需要多用一块盘片来装。



## 磁盘的地址结构

如何将一个内存中的簇号来转化为磁盘上的物理地址:将簇号转化为磁盘物理地 址的过程由磁盘驱动程序完成。假设内存的簇号为b,则:

- □ 磁头号: | (簇号%每个柱面的簇数)/每个磁道的簇数|。
- □ 分析: 一个磁道代表一个圆环, 也就是一圈, 然后一圈上的物理块的个数是根据 分的扇区个数来决定的。给出的簇号同样当成物理块的总数,所求的磁头号可以 认为簇号在一个柱面的哪一个磁道上,则需除以一个柱面的所有物理块取余(得 到他剩下的物理块),然后再除以一个磁道的物理块数,来判断簇号在哪一个圆 环上,结果依旧向下取整。



## 磁盘的地址结构

如何将一个内存中的簇号来转化为磁盘上的物理地址:将簇号转化为磁盘物理地 址的过程由磁盘驱动程序完成。假设内存的簇号为b,则:

- □ 扇区号: 扇区地址(簇号)%每个磁道的扇区数。
- □ 分析: 所求扇区号可以认为簇号在一个柱面的一个磁道的哪一个扇形的区域上。 由于我们之前得到了柱面号和磁头号。我们就确定了所求簇号在哪一个盘片的哪 一个磁道上(也就是哪一个圆环上),现在需要求出扇区号来得到他最后的位置, 则可以用簇号(所求物理块总数)对一个磁道上的所有物理块总数取余。位置,则 可以用簇号(所求物理块总数)对一个磁道上的所有物理块总数取余。这样我们就 确定了一个簇号在磁盘上的具体位置了。



磁盘访问时间

#### □ 寻道时间Ts

这是指把磁臂(磁头)移动到指定磁道上所经历的时间。该时间是启动磁臂的时 间s与磁头移动n条磁道所花费的时间之和, 即Ts=m×n+s

- □ 旋转延迟时间TT 这是指定扇区移动到磁头下面所经历的时间。
- □ 传输时间Tt

这是指把数据从磁盘读出或向磁盘写入数据所经历的时间。 Tt的大小与每次 所读/写的字节数b和旋转速度有关。



# 磁盘访问时间

寻道时间Ts=m\*n+s

旋转延迟时间Tr

磁盘访问的时间

传输时间Tt=b/rN

平均读取时间:1/r

总时间Ta=Ts+1/2r+b/rN



- 1. 【重庆理工大学 2013】访问磁盘的时间不包括()。
  - A. 寻道时间 B. CPU调度时间
  - C. 读写时间 D. 旋转等待时间
- B【解析】磁盘执行一次输入/输出操作所需花费的时间,由寻道时间、旋转等待时间和读写时间三个部分组成。
- 其中寻道时间是指磁头在移动臂带动下移动到指定柱面所需的时间;
- 旋转等待时间是指指定扇区旋转到磁头位置所需的时间;
- 读写时间是指由指定的磁头把磁道上的信息读到主存储器或把主存储器中的信息 写到磁道上所需的时间。因此选择B。

- 2. 设磁盘的转速为3000r/min, 盘面划分成10个扇区,则读取一个扇区的时间为()。
  - A. 20 ms B. 5 ms C. 2 ms D. 1 ms
- C【解析】访问每条磁道的时间为60/3000s=0.02s=20(ms),即磁盘旋转一圈的时间为20ms,每个盘面分成10个扇区,故读取一个扇区的时间为20/10 ms=2(ms)。因此选择C。

3. 某磁盘的转速为10000转/分,平均寻道时间是6ms,磁盘传输速率是20

MB/s, 磁盘控制器延迟为0.2 ms, 读取一个4KB的扇区所需的平均时间约为()。

- A. 9 ms B. 9.4 ms
- C. 12 ms D. 12.4 ms

B【解析】磁盘旋转一圈的时间是6 ms, 那么旋转延迟是3 ms。

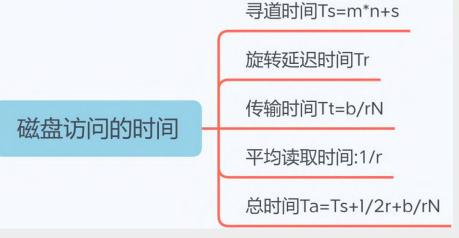
读取4KB扇区信息时间是4KB/20MB/s=0.2(ms),

平均寻道时间是6 ms,

所以平均时间是6+3+0.2+0.2=9.4(ms)。因此选择B。

磁盘调度





当多个进程请求磁盘访问时, 磁盘调度应使磁盘的平均寻道时间最短

最短寻道时间优先 (SSTF)

说明: 要求访问的磁道和当前磁头所在的磁道距

离最近,以使每次的寻道时间最短

扫描算法不仅考虑到欲访问的磁道与当前磁道间的距离,更优先考虑的是磁道当前的移动方向

扫描算法 (SCAN)

联想电梯的运行

可防止低优先级进程出现"饥饿"的现象

算法规定磁头单向移动,例如,只是自里向外移动, 当磁头移到最外的磁道并访问后,磁头立即返回到最 里的欲访问磁道,亦即将最小磁道号紧接着最大磁道

循环扫描算法 (CSCAN)

号构成循环,进行循环扫描

N步SCAN算法是将磁盘请求队列分成若干个长度为N的子

NStepScan算法

队列,磁盘调度将按FCFS算法依次这些子队列。

FSCAN算法

是Nstepscan算法的简化,将磁盘请求队列分成两个子队列

#### 磁盘的调度算法



先来先服务FCFS(First-Come, First Served)

(从 100 号磁道开始)		
被访问的下	移动距离	
一个磁道号	(磁道数)	
55	45	
58	3	
39	19	
18	21	
90	72	
160	70	
150	10	
38	112	
184	146	
平均寻道长度: 55.3		

- 根据进程请求访问磁盘的先后次序进行调度
- > 公平、简单,进程请求都能依次得到处理
- ▶ 适合请求磁盘I/O进程数较少场合



最短寻道时间优先SSTF(Shortest Seek Time First)

(从 100 号磁道开始)		
被访问的下	移动距离	
一个磁道号	(磁道数)	
90	10	
58	32	
55	3	
39	16	
38	1	
18	20	
150	132	
160	10	
184	24	
平均寻道长度: 27.5		

- 算法选择要求访问的磁道与当前磁头所在 磁道距离最近的进程,使本次操作寻道时 问最短。
- □ 可能导致某些进程发生"饥饿"。



### SCAN算法

(从 100<sup>#</sup> 磁道开始,向磁道号增加方向 访问)

	<del>,</del>	
被访问的下	移动距离	
一个磁道号	(磁道数)	
150	50	
160	10	
184	24	
90	94	
58	32	
55	3	
39	16	
38	1	
18	20	

平均寻道长度: 27.8

- □ 称电梯调度算法。
- □ 不仅考虑欲访问的磁道与当前磁道的距 离,更优先考虑磁头的当前移动方向。
- □ 优点:算法既能获得较好的寻道性能, 又能防止进程饥饿,广泛应用于大、中、 小型机和网络中的磁盘调度。



### 循环扫描(CSCAN)算法

(从 100<sup>#</sup> 磁道开始, 向磁道号增加方向 访问)

被访问的下	移动距离	
一个磁道号	(磁道数)	
150	50	
160	10	
184	24	
18	166	
38	. 20	
39	1	
55	16	
58	3	
90	32	

平均寻道长度: 27.5

- □ 不仅考虑欲访问的磁道与当前磁道的距离,更优先考虑磁头的当前移动方向。
- □ 规定只有磁头朝某个特定方向移动时才处理磁道访问请求,而返回时直接快速移动至起始端而不处理任何请求。



## N-Step-SCAN算法

- □ 在SSTF、 SCAN及CSCAN几种调度算法中, 都可能出现磁臂停留在某处不 动的情况,例如,有一个或几个进程对某一磁道有较高的访问频率,即这 个(些)进程反复请求对某一磁道的I/O操作,从而垄断了整个磁盘设备。
- □ 这一现象称为"磁臂粘着"(Armstickiness)。



### N-Step-SCAN算法

- □ N步SCAN算法是将磁盘请求队列分成若干个长度为N的子队列,磁盘调度 将按FCFS算法依次处理这些子队列。 而每处理一个队列时又是按SCAN算 法,对一个队列处理完后,再处理其他队列。 当正在处理某子队列时,如 果又出现新的磁盘I/O请求,便将新请求进程放入其他队列,这样就可避 免出现粘着现象。
- □ 当N值取得很大时,会使N步扫描法的性能接近于SCAN算法的性能;
- □ 当N=1时,N步SCAN算法便蜕化为FCFS算法。



- □ FSCAN算法实质上是N步SCAN算法的简化,即FSCAN只将磁盘请求队列 分成两个子队列。
- □ 一个是由当前所有请求磁盘I/O的进程形成的队列,由磁盘调度按SCAN算法进行处理。
- □ 在扫描期间,将新出现的所有请求磁盘I/O的进程, 放入另一个等待处理的 请求队列;
- □ 所有的新请求都将被推迟到下一次扫描时处理。

【政哥点拨】

- 1. 驱动调度算法中, () 算法可能会随时改变移动臂的运动方向。
  - A. 电梯调度 B. 最短寻道时间优先 C. 扫描 D. 单向扫描
- B【解析】最短寻道时间优先(SSTF)算法是以查找距离磁头最短(也就是查找时 间最短)的请求作为下一次服务对象,因此其可能会随时改变移动臂的运动方向。 扫描算法可细分为电梯调度(SCAN)算法和循环扫描(CSCAN)算法。电梯调度 (SCAN)算法是在磁头前进方向上查找最短寻道时间的请求, 若前进方向上没有请 求(即处理完最高/低编号柱面请求后),则掉转方向。SCAN算法很大程度上消 除了SSTF算法的不公平性,但仍有利于中间磁道的请求。循环扫描(CSCAN)算法 是对SCAN算法的改进,它总是按同一方向移动磁头,当处理完最高编号的柱面 请求后,不是掉转方向,而是把磁头移动到最低编号的柱面请求处,然后按同一 方向继续向上移动。这种算法彻底消除了对两端磁道请求的不公平性。因此选择 B.

2. 当前磁盘读写位于柱面号20,此时有多个磁盘请求以下列柱面号的顺序送到磁盘驱动器: 10,22,2,40,6,38。在寻道时,移动一个柱面需要6ms,按照先来先服务算法和电梯算法(方向从0到40)计算所需的总寻道时间。

【解析】(1)先来先服务算法: 寻道的次序为20,10,22,2,40,6,38。总的寻道时间为(10+12+20+38+34+32)×6=876(ms)。

(2)电梯算法(方向从0到40): 寻道的次序为20,22,38,40,10,6,2。总的寻道时间为 (2+16+2+30+4+4)×6=348(ms)。

磁盘存储空间管理



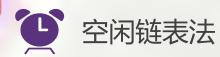
- □ 空闲表法属于连续分配方式,它与内存的动态分配方式雷同,它为每个文件 分配一块连续的存储空间,即系统也为外存上的所有空闲区建立一张空闲表,
- □ 每个空闲区对应于一个空闲表项,其中包括表项序号、该空闲区的第一个盘块号、该区的空闲盘块数等信息。
- □ 将所有空闲区按其起始盘块号递增的次序排列。

序号	第一个空闲盘号	空闲盘块数
1	2	4
2	9	3
3	15	5
4	-	1



#### 空闲链表法

- □ 空闲链表法是将所有空闲盘区拉成一条空闲链。根据构成链所用基本元素的不同,可把链表分成两种形式: 空闲盘块链和空闲盘区链。
- (1) 空闲盘块链。这是将磁盘上的所有空闲空间,以盘块为单位拉成一条链。当用户因创建文件而请求分配存储空间时,系统从链首开始,依次摘下适当数目的空闲盘块分配给用户。当用户因删除文件而释放存储空间时,系统将回收的盘块依次插入空闲盘块链的末尾。这种方法的优点是用于分配和回收一个盘块的过程非常简单,但在为一个文件分配盘块时,可能要重复操作多次。



- □ 空闲链表法是将所有空闲盘区拉成一条空闲链。根据构成链所用基本元素的不同,可把链表分成两种形式: 空闲盘块链和空闲盘区链。
- (2) 空闲盘区链。这是将磁盘上的所有空闲盘区(每个盘区可包含若干个盘块) 拉成一条链。在每个盘区上除含有用于指示下一个空闲盘区的指针外,还应有能指明本盘区大小(盘块数)的信息。分配盘区的方法与内存的动态分区分配类似,通常采用首次适应算法。在回收盘区时,同样也要将回收区与相邻接的空闲盘区相合并。在采用首次适应算法时,为了提高对空闲盘区的检索速度,可以采用显式链接方法,亦即,在内存中为空闲盘区建立一张链表



- □ 位示图是利用二进制的一位来表示磁盘中一个盘块的使用情况。
- □ 当其值为 "0" 时,表示对应的盘块空闲;
- □ 为 "1" 时,表示已分配。(用一位的两种状态来标志空闲和已分配两种情 况。)
- □ 磁盘上的所有盘块都有一个二进制位与之对应,这样,由所有盘块所对应的 位构成一个集合, 称为位示图。



□ 通常可用 m×n 个位数来构成位示图,并使 m×n 等于磁盘的总块数,

```
6 7 8 9 10 11 12 13 14 15 16
16
```



- (1) 盘块的分配
- ①顺序扫描位示图,从中找出一个或一组其值为"0"的二进制位("0"表示空闲时)。
- ②将所找到的一个或一组二进制位, 转换成与之相应的盘块号。假定找到的其值为 "0" 的二进制位,位于位示的第i行、第j列,则其相应的盘块号应按下式

计算: b=n(i-1)+j

式中,n代表每行的位数。

③修改位示图, 令map [i,j] =1。



## 位示图法

- (2) 盘块的回收
- ①将回收盘块的盘块号转换成位示图中的行号和列号。 转换公式:

$$i=(b-1)DIV n + 1$$
  
 $j=(b-1)MOD n + 1$ 

②修改位示图。 令map [i,j] =0。

【政哥点拨】

#### 位示图可用于()。

- A. 文件目录的查找 B. 磁盘空间的管理
- C. 主存空间的管理 D. 文件的保密
- B【解析】位示图方法是空闲块管理方法,用于管理磁盘空间。



- 1. 下列关于磁盘的说法中, 正确的是()。
- I. 每个磁盘只有两个盘面, 但为了方便读取, 一般只使用其中一个盘面进行存取
- Ⅱ.磁道是磁盘的最小物理存储单位
- Ⅲ. 磁盘在使用之前必须进行格式化,即把一个磁盘分成扇区以便磁盘控制器能读和写
- IV. 同一个磁道的柱状空间称为柱面,每个柱面上分为若干个扇区
  - A. I和Ⅲ B. 只有Ⅲ C. Ⅱ和IV D. 只有IV
- B【解析】一个磁盘包括一个或多个盘片,每个盘片又有两个存储面,每个存储面可以划分为多个磁道,不同盘片及盘面上同一个磁道的柱状空间又称为柱面,每个磁道上分为若干个扇区。

扇区是磁盘的最小物理存储单位。

为了在磁盘上存储数据,必须对磁盘进行格式化,格式化就是把一个磁盘分成扇 区以便磁盘控制器能读和写。因此选择B。 2. 磁盘存储器由()组成。

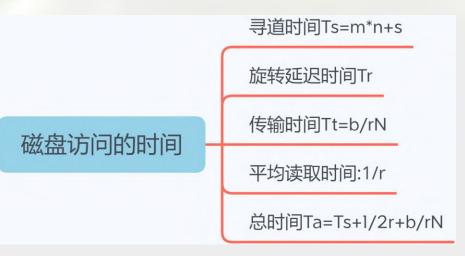
I. 磁盘 Ⅱ. 通道 Ⅲ. 驱动器结构 IV. 控制器 V.缓冲区

A. I、 I 和 V B. I、 I 和 I V C. I、 II 和 I V D. I、 I V和 V

C【解析】磁盘存储器主要由磁盘、磁盘驱动器和磁盘控制器三部分组成。因此选择C。

- 3. 磁盘的读写单位是()。
  - A. 块 B. 扇区 C. 簇 D. 字节
- B【解析】磁盘读写的基本单位是扇区。因此选择B。

- 【重庆理工大学 2014】磁盘调度的策略主要是为了优化()。
  - A. 交换时间
- B. 寻道时间 C. 旋转延迟时间
- D. 传输时间



B【解析】当多个进程请求磁盘访问时,磁盘调度应使磁盘的平均寻道时间最短

- 5. 机房新购买一台外存储器——磁盘,其中共m个柱面,每个柱面有n个磁道,每个磁道有p个存储块(即扇区),每个存储块有q字节。假定将这台磁盘机安装在字节处理器中,并采用位示图方式管理空间的分配和回收,那么系统需要划出()字节来建立这张位示图。
  - A. mnp B. mnpq C. mnp/8 D. mnpq/8
- C【解析】磁盘容量为mnp块,字节处理器,一个字节占8位,则位示图的大小为mnp/8。因此选择C。

- 6. 一个磁盘的转数为7200 r/min,每个磁道有160个扇区,每扇区有512 B,那么理想情况下,其数据传输是()。
  - A. 7200×160 KB/s B. 7200 KB/s
  - C. 9600 KB/s D. 19200 KB/s
- C【解析】磁盘的转速为7200r/min=120r/s, 转一圈经过160个扇区, 每个扇区 有512B, 所以数据传输率为120×160×512/1024=9600(KB/s)。因此选择C。

- 7. 【中国计量大学 2018】下列算法中用于磁盘调度的是( )。
  - A. 最近最少使用LRU算法 B. FIFO算法
  - C. 时间片轮转法RR D. 循环扫描算法(CSCAN)
- D【解析】最近最少使用LRU算法属于内存页面置换算法,

FIFO算法和时间片轮转法RR属于进程调度算法,

磁盘调度算法中常见的有先来先服务算法、最短寻道时间优先算法、扫描算法、

循环扫描算法等。因此选择D。

8. 设某文件为链接文件,它由5个逻辑记录组成,每个逻辑记录的大小与磁盘块的大小相等,均为512 B,并依次存放在50,121,75,80,63号磁盘块上。若要存取文件的第1569逻辑字节处的信息,则应该访问()号磁盘块。

A. 3 B. 80 C. 75 D. 63

B【解析】由于1569=512×3+33,因此要访问的字节位于第4个磁盘块上,对应的盘块号为80。因此选择B。

9. 【中国计量大学 2020】假设磁头当前位于100道,现有一个磁道访问请求序列为48,22,88,120,95,150,168,采用最短寻道时间优先算法(SSTF算法)得到的磁道访问序列是()。

A. 120,150,168,22,48,88,95 B. 120,150,168,95,88,48,22

C. 95,88,120,150,168,48,22 D. 48,22,88,120,95,150,168

C【解析】最短寻道时间优先算法选择调度处理的磁道是与当前磁头所在磁道 距离最近的磁道,100最近的是95。因此选择C。

即顺序是: 95,88,120,150,168,48,22

- 10. 若用8个字(字长为32位)组成的位示图管理内存,假定用户归还一个块号为100的内存块时,它对应位示图的位置为()(统一由0开始编号)。
  - A. 字号为3, 位号为5 B. 字号为4, 位号为4
  - C. 字号为3, 位号为4 D. 字号为4, 位号为5
  - C【解析】字号: |100/32|=3, 位号: 100 mod 32=4。因此选择C。

11.【南京工业大学 2015】现有容量为20GB的磁盘分区,磁盘空间以簇 (cluster)为单位进行分配,簇的大小为16KB,若采用位图法管理该分区的空闲 空间,即用一位(bit)标识一个簇是否被分配,则存放该位图所需簇的个数为( )。 A. 10 B. 160 C. 10 K D. 160 K

A【解析】簇的总数为20GB/16KB=1.25(M),用一位标志一个簇是否被分配,则整个磁盘共需要1.25Mb,即需要1.25Mb/8=160(KB),则共需要160KB/16 KB=10(个)。因此选择A。

- 12. 【中国计量大学 2017】 位示图用于( )。
- A. 页面置换 B. 磁盘空间管理 C. 文件目录查找 D. 磁盘驱动调度 B【解析】位示图可以用来管理内存和磁盘空间,每一位代表内存或磁盘的一个最小分配单元,如页面或磁盘块(文件块),当其值为"1"时表示已被占用,当其值为"0"时表示闲置。因此选择B。

- 13. 下列关于磁盘移臂调度算法的叙述,错误的是()。
- A. 移臂调度的目的是减少寻道时间,包括先来先服务、最短寻道时间优先、电 梯调度和循环扫描算法等
- B. 先来先服务算法是最简单且花费时间最短的调度算法
- C. 先来先服务算法不考虑访问的物理位置,只按到来的先后顺序进行调度
- D. 最短寻道时间优先和先来先服务调度算法会随时改变磁头的移动方向
- B【解析】先来先服务(FCFS)算法根据磁道访问请求到来的先后顺序完成请求。 如果有大量进程竞争使用磁盘,请求访问的磁道很分散,则FCFS在性能上很差, 寻道时间就会很长。因此选择B。

- 14. 下列算法中,用于磁盘臂调度的是()。
  - A. 时间片轮转法 B. 优先级高者优先算法
  - C. 最短寻道时间优先 D. LRU算法
- C【解析】最短寻道时间优先算法总是优先满足距离磁头当前位置最近的访问请求,磁盘的磁臂会不断改变方向。因此选择C。

- 15. 在以下磁盘调度算法中,可能出现饥饿现象的是()。
  - A. 电梯调度算法 B. 最短寻道时间优先算法
  - C. 循环扫描算法 D. 先来先服务算法
- B【解析】最短寻道时间算法的本质是利用贪心算法来实现,假设当前磁道在某一位置,接下来处理的是距离当前磁道最近的磁道号,处理完成之后再处理离这个磁道号最近的磁道号,直到所有的磁道号都服务完了程序结束。
- □ 优点是性能优于FCFS算法,
- □ 缺点是会使距离当前磁道较远的磁道号长期得不到服务,也就是"饥饿"现象,因为要求访问的服务的序列号是动态产生的,即各个应用程序可能不断地提出访问不同的磁道号的请求。就近处理,远的可能长时间得不到服务,就会"饿"。因此选择B。

16. 【南京理工大学 2015】磁盘请求以10,22,20,2,40,7,38柱面次序到达磁盘驱动器。寻道时磁头每移动一个柱面要6ms,磁盘调度算法采用SSTF,则满足上述磁盘请求所需的寻道时间为(当前磁头在8号磁道)()。

A. 360 ms B. 264 ms C. 300 ms D. 828 ms

C【解析】SSTF算法处理次序为8,7,10,2,20,22,38,40,则移动的磁道数为1+3+8+18+2+16+2=50,寻道时磁头每移动一个柱面要6 ms,则总寻道时间为50×6 ms=300(ms)。因此选择C。

- 17. 【南京工业大学 2015】对磁盘进行移臂调度时,既考虑了减少寻道时间,
- 又不频繁改变移动臂的移动方向的调度算法是()。
  - A. 先来先服务算法 B. 最短寻道时间优先算法
  - C. 电梯调度算法 D. 优先级高者优先算法
- C【解析】磁盘调度需要考虑公平性和高效性,电梯调度既考虑了减少寻道时间, 又不频繁改变移动臂的移动方向。因此选择C。

18. 设一个磁道访问请求序列为55,58,39,18,90,160,150,38,184, 磁头的起始位置为100, 若采用SSTF(最短寻道时间优先)算法,则磁头移动()个磁道。

A. 55 B. 184 C. 200 D. 248

D【解析】对于SSTF算法, 寻道序列应为100,90,58,55,39,38,18,150,160,184; 移动磁道次数分别为10,32,3,16,1,20,132,10,24, 故磁头移动总次数为248。

另外也可以画草图来解答,从100寻道到18需要82次,从18到184需要184-18=166(次), 共移动166+82=248(次)。

- 19. 若干个等待访问磁盘者依次要访问的磁道为20,44,40,4,80,12,76,当前磁头位于40号柱面,若用最短寻道时间优先磁盘调度算法,则访问序列为();若用扫描算法(当前磁头移动的方向为磁道递增),则访问序列为()。
  - A. 20,44,40,4,80,12,76 B. 40,44,20,12,4,76,80
  - C. 40,44,76,80,20,12,4 D. 40,44,76,80,4,12,20
- B, C【解析】最短寻道时间优先算法,磁盘总是把磁头移向离当前请求最近的磁道。故而,当磁头在40号柱面时,其访问序列为40,44,20,12,4,76,80。因此选择B。

采用扫描算法,当前磁头移动方向为磁道递增方向。要特别注意,扫描算法 (电梯调度算法)把磁头移动到请求的最远端,又从该最远端移回。故而,当前磁头位于40号柱面时,其访问序列应该是40,44,76,80,20,12,4。因此选择C。

## 【大显身手】

- 1. 【西北大学 2015】假定有一个磁盘组共有199个柱面,每个柱面有16个磁道,每个磁道被划分成8个扇区,柱面、磁道和扇区的编号均从0开始。现有一个700个逻辑记录的文件,逻辑记录大小与扇区大小相同,其编号从0开始。该文件以顺序结构的形式,从磁盘的1号柱面、5磁道、0扇区开始存放,试问: (1)磁盘的盘块编号与柱面号、磁道号和扇区号有什么关系?
- 【解析】(1)已知磁盘块物理地址,则磁盘块号:块号=柱面号×(磁头数×扇区数)+磁道号×扇区数+扇区号=柱面号×(16×8)+磁道号×8+扇区号。

## 已知块号,则磁盘驱动用的三地址:

**柱面号** = [(块号/(磁头数×扇区数))/扇区数] = [块号/(16×8)];

磁道号 =  $\lfloor ($ 块号mod(磁头数×扇区数 $))/扇区数 \rfloor = \lfloor$ 块号% $(16\times8))/8 \rfloor$ ;

扇区号=(块号 mod (磁头数×扇区数) mod 扇区数= (块号%(16×8)%8。

- 1. 【西北大学 2015】假定有一个磁盘组共有199个柱面,每个柱面有16个磁道,每个磁道被划分成8个扇区,柱面、磁道和扇区的编号均从0开始。现有一个700个逻辑记录的文件,逻辑记录大小与扇区大小相同,其编号从0开始。该文件以顺序结构的形式,从磁盘的1号柱面、5磁道、0扇区开始存放,试问:
- (2)该文件的第380个逻辑记录应存放在哪个柱面的第几磁道的第几个扇区?
- (2)由于逻辑记录大小与扇区大小相同,该文件从磁盘的1号柱面,5号磁道,0号扇区开始存放,因此1号柱面还有 11 个磁道可以存放记录,共可以存放 11 x 8=88 个记录, 2号柱面可以存放 16 x 8=128个记录,
- 3 号柱面可以存放 16 x 8=128 个记录,
- 那么4号柱面还需要存放 380-88-128-128=36个,前个磁道可以存放 4\*8=32 个记录,第5个磁道存放4个记录,因此,文件第 380 个逻辑记录存放在磁盘的4 号柱面(第5个磁道)、4 号磁头(第5个磁道)的 3号扇区(第4个扇区)。

- 1. 【西北大学 2015】假定有一个磁盘组共有199个柱面,每个柱面有16个磁道,每个磁道被划分成8个扇区,柱面、磁道和扇区的编号均从0开始。现有一个700个逻辑记录的文件,逻辑记录大小与扇区大小相同,其编号从0开始。该文件以顺序结构的形式,从磁盘的1号柱面、5磁道、0扇区开始存放,试问:
- (3)2柱面的第1磁道的第7个扇区中存放了该文件的第几个逻辑记录?
- (3)由于1号柱面还可以存放88个记录,而2号柱面的0号磁道可以存放8个记录,则1号磁道的6号(第七个)扇区存放的是第88+8+7=103个逻辑记录。

2.【广东工业大学 2019】若干个等待访问磁盘者依次要访问的柱面为20,44,40,4,80,12,76,假设每移动一个柱面需要的时间为3ms。移动臂当前位于40号柱面,请按下列算法分别计算完成上述访问总共花费的寻道时间。(1) FCFS算法。

【解析】(1) FCFS算法就是每次都找先来的,初始位于40,访问序列就是40,20,44,4,80,12,76,一共移动了20+24+40+76+68+64=292个柱面,所以总共花费的寻道时间就是292×3=876(ms)。

- 2. 【广东工业大学 2019】若干个等待访问磁盘者依次要访问的柱面为 20,44,40,4,80,12,76,假设每移动一个柱面需要的时间为3ms。移动臂当前位于 40号柱面,请按下列算法分别计算完成上述访问总共花费的寻道时间。
- (2) 最短寻道时间优先算法。
- (2)最短寻道时间优先(SSTF)算法就是每次都找最近的,初始位于40,访问序列就是40,44,20,12,4,76,80,一共移动了4+24+8+8+72+4=120个柱面,所以总共花费的寻道时间就是120×3=360(ms)。

- 2. 【广东工业大学 2019】若干个等待访问磁盘者依次要访问的柱面为 20,44,40,4,80,12,76,假设每移动一个柱面需要的时间为3ms。移动臂当前位于 40号柱面,请按下列算法分别计算完成上述访问总共花费的寻道时间。
- (3) 我们现在需要研究一下这两种算法哪一种更好,请说明至少两种评价指标,并且设计一个研究方案,可以合理地回答这个问题。
- ①从寻道时间看, SSTF算法的平均寻道时间要明显短于FCFS算法;
- ②从饥饿角度看,SSTF算法会导致饥饿,而FCFS算法不会导致饥饿。

3. 【吉林大学 2017】某磁盘共有500个磁道,编号0~499,磁头当前位于102 磁道时顺序到来如下磁道访问请求:50,90,100,105,125,305。试用下列算法分别计算磁头跨越磁道总数。

## (1)先来先服务算法FCFS。

【解析】(1)先来先服务(FCFS)算法就是每次都找先来的,初始位于102,每次都找先来的,访问序列就是102,50,90,100,105,125,305,磁头跨越磁道总数为52+40+10+5+20+180=307。

- 3. 【吉林大学 2017】某磁盘共有500个磁道,编号0~499,磁头当前位于102 磁道时顺序到来如下磁道访问请求:50,90,100,105,125,305。试用下列算法分别计算磁头跨越磁道总数。
- (2)最短寻道时间优先算法(SSTF)。
- (2)最短寻道时间优先(SSTF)算法就是每次都找最近的,初始位于102,每次都找最近的,访问序列就是102,100,105,90,125,50,305,磁头跨越磁道总数为2+5+15+35+75+255=387。

- 3. 【吉林大学 2017】某磁盘共有500个磁道,编号0~499,磁头当前位于102 磁道时顺序到来如下磁道访问请求:50,90,100,105,125,305。试用下列算法分别计算磁头跨越磁道总数。
- (3)比较二种算法结果有何启示?
- (3)先来先服务(FCFS)算法根据磁道访问请求到来的先后顺序完成请求。

优点:公平,简单;如果请求访问的磁道比较集中的话,算法性能很好。

缺点: 很难优化寻道时间; 如果有大量进程竞争使用磁盘, 请求访问的磁道很分

散,则FCFS在性能上很差,寻道时间就会很长。

最短寻道时间优先(SSTF)算法总是优先满足距离磁头当前位置最近的访问请求。

优点: 性能较好, 平均寻道时间短。

缺点:可能产生"饥饿"现象。

- 4. 【南京理工大学 2016】某磁盘有1000个柱面(0~999),设磁头当前位置是756,磁头正向0磁道运动。按照FCFS排列的磁盘请求队列为:811,348,153,968,407,500。
- (1)计算SCAN算法完成上述磁盘请求中磁头运动的磁道数。

【解析】(1) SCAN算法处理次序为756,500,407,348,153,811,968,磁盘请求中磁 头运动的磁道数为256+93+59+195+658+157=1418。

- 4. 【南京理工大学 2016】某磁盘有1000个柱面(0~999),设磁头当前位置是756,磁头正向0磁道运动。按照FCFS排列的磁盘请求队列为:811,348,153,968,407,500。
- (2)设某单面磁盘转速为每分钟12000转,每个磁道有100个扇区,相邻磁道间的平均移动时间为1ms,且对请求队列中的每个磁道需读取1个随机分布的扇区,计算完成这些磁盘请求的寻道时间。
- (2)寻道时间为1418×1 ms=1418(ms)。

- 4. 【南京理工大学 2016】某磁盘有1000个柱面(0~999),设磁头当前位置是756,磁头正向0磁道运动。按照FCFS排列的磁盘请求队列为:811,348,153,968,407,500。
- (3)计算完成这些磁盘请求的访问时间。
- (3)每分钟12000转,转一圈时间是0.005 s,同时因为是100个扇区,通过1个扇区的时间是0.00005 s,读取6个扇区读取时间:读一圈加上通过一圈半径时间,注意这个扇区在对面的1/2,则读取时间为
- $(0.005 \times 0.5 + 0.00005) \times 6 = 0.0153(s) = 15.3 \text{ ms},$
- 故寻道时间加上读取时间是1418 ms+15.3 ms=1433.3 ms。

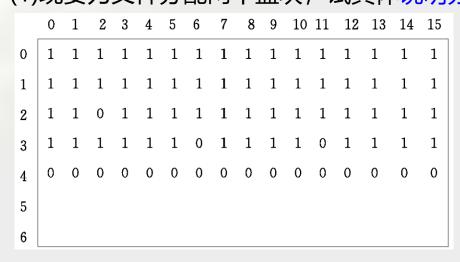
- 4. 【南京理工大学 2016】某磁盘有1000个柱面(0~999),设磁头当前位置是756,磁头正向0磁道运动。按照FCFS排列的磁盘请求队列为:811,348,153,968,407,500。
- (4)简述SCAN算法的优缺点。
- (4)电梯调度算法(SCAN算法)对于先后到达的磁盘访问请求,首先选择移臂方向, 磁臂在该方向上移动的过程中依次处理途经的各个访问请求,直到该方向上再无 请求时,改变移臂方向,依次处理相反方向上遇到的各个请求。如果同一柱面上 有多个请求,还需进行旋转优化。

优点: 性能较好, 平均寻道时间较短, 不会产生饥饿现象。

缺点:只有到达最边上的磁道时才能改变磁头移动方向; SCAN算法对于各个位

置磁道的响应频率不平均。

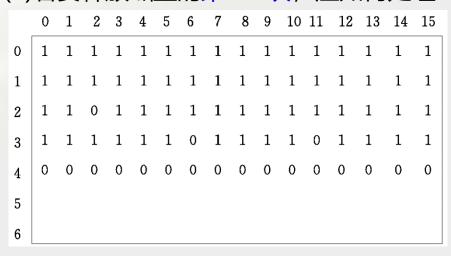
5.【安徽师范大学 2019】有一计算机系统利用如图所示的位示图(行号、列号都从0开始编号)来管理空闲盘块。如果盘块从1开始编号,每个盘块的大小为1 KB。(1)现要为文件分配两个盘块,试具体说明分配过程。



【解析】(1)①顺序检索位示图,从中找到第一个值为0的二进制位,得到其行号i=2,列号j=2。

- ②计算出位所对应的盘块号: b=n×i+j+1=16×2+2+1=35。
- ③修改位示图,令map[2,2]=1,并将对应块35分配给文件。按照同样的方式,可找到第3行、第6列的值为0的位,转换为盘块号55,将位的值修改为1,并将55号盘块分配给文件。

- 5.【安徽师范大学 2019】有一计算机系统利用如图所示的位示图(行号、列号都从0开始编号)来管理空闲盘块。如果盘块从1开始编号,每个盘块的大小为1 KB。
- (2)若要释放磁盘的第300块,应如何处理?



- (2)要释放磁盘的第300块,应进行如下处理:
- ①计算出磁盘第300块所对应的二进制位的行号i和列号j: i = [(300-1)/16]=18, j=(300-1)%16=11。
- ②修改位示图, 令map[18, 11]=0, 表示对应块为空闲块。

6. 【西北大学 2017】假定有一个磁盘系统有200个柱面,每个柱面有8个磁道,每个磁道有16个扇区,每个扇区的大小为8KB。柱面、磁道和扇区的编号均从0开始。现有一个1000个逻辑记录(编号从0开始)的文件,一个扇区恰好放1个逻辑记录。该文件以顺序结构的形式,从磁盘的1号柱面、5磁道、0扇区开始存放,试问:

(1)该磁盘系统容量多大?

【解析】(1)磁盘容量计算公式为n×t×s×b。n为保存数据的总盘面数,t为每面磁道数,s为每道的扇区数,b为每个扇区存储的字节数,则磁盘系统容量为200×8×16×8 KB=200(MB)。

- 6. 【西北大学 2017】假定有一个磁盘系统有200个柱面,每个柱面有8个磁道,每个磁道有16个扇区,每个扇区的大小为8KB。柱面、磁道和扇区的编号均从0开始。现有一个1000个逻辑记录(编号从0开始)的文件,一个扇区恰好放1个逻辑记录。该文件以顺序结构的形式,从磁盘的1号柱面、5磁道、0扇区开始存放,试问:
- (2)磁盘的盘块号N与柱面号i、磁道号j和扇区号k之间的关系是什么(即由i, j, k 如何算出N; 反之通过N如何得到i, j, k)?
- (2)已知磁盘块物理地址,则磁盘块号:块号 N=柱面号 i×(磁头数×扇区数)+磁道号 j×扇区数+扇区号 k。

已知块号,则磁盘驱动用的三地址:

柱面号 i=[块号 N/(磁头数×扇区数)]=[块号 N/(8×16)];

磁道号 j=L(块号 N mod(磁头数×扇区数))/扇区数」=L(块号 N%(8×16))/16 」;

扇区号 k=(块号  $N \mod($ 磁头数 $\times$ 扇区数 $))\mod$ 扇区数=(块号 $\%(8\times16))\%16$ 。

- 6. 【西北大学 2017】假定有一个磁盘系统有200个柱面,每个柱面有8个磁道,每个磁道有16个扇区,每个扇区的大小为8KB。柱面、磁道和扇区的编号均从0开始。现有一个1000个逻辑记录(编号从0开始)的文件,一个扇区恰好放1个逻辑记录。该文件以顺序结构的形式,从磁盘的1号柱面、5磁道、0扇区开始存放,试问:
- (3)该文件的第480个逻辑记录应存放在哪个柱面、哪个磁道、哪个扇区?
- (3)由于逻辑记录大小与扇区大小相同,该文件从磁盘的1号柱面,5号磁道,0号扇区开始存放,因此1号柱面还有3个磁道可以存放记录,共可以存放 3\*16=48个记录,
- 2号柱面可以存放8\*16=128 个记录,
- 3号柱面可以存放 8\*16=128 个记录,
- 4号柱面可以存放 8\*16=128个记录,
- 那么5号柱面还需要存放 480-48-128-128-128=48个,3个磁道可以存放 16X3=48 个记录, 因此,文件第 480 个逻辑记录存放在磁盘的 5 号柱面、2号磁头(第3个磁道)的 15 号扇区(第 16 个扇区)。

- 6. 【西北大学 2017】假定有一个磁盘系统有200个柱面,每个柱面有8个磁道,每个磁道有16个扇区,每个扇区的大小为8KB。柱面、磁道和扇区的编号均从0开始。现有一个1000个逻辑记录(编号从0开始)的文件,一个扇区恰好放1个逻辑记录。该文件以顺序结构的形式,从磁盘的1号柱面、5磁道、0扇区开始存放,试问:
- (4)第2柱面的第2磁道的第5个扇区中存放了该文件的第几个逻辑记录?
- (4)1号柱面还可以存放 48个记录,
- 2号柱面的0号磁道和1号磁道共可以存放 16 \* 2=32 个记录,
- 2号磁道还可以存放5个记录,则第2号磁道的第5个扇区存放的是
- 第 48+32+5=85 个逻辑记录。

7. 【南京理工大学 2017】在一个磁盘上,有1000个柱面,从0~999。假定当前磁头在磁道345上,并且磁头正在向磁道0移动。按照FCFS顺序排列的请求队列中包含了如下磁道上的请求: 186,900,692,620,120,400。分别求出系统采用FCFS、最短寻道时间优先和SCAN磁盘调度算法满足上述磁盘请求,磁头运动经过的磁道序列及总磁道数。

【解析】FCFS算法磁头运动经过的磁道序列为345,186,900,692,620,120,400,总磁道数为159+714+208+72+500+280=1933。

最短寻道时间优先算法磁头运动经过的磁道序列为345,400,620,692,900,186,120,总磁道数为55+220+72+208+714+66=1335。

SCAN算法磁头运动经过的磁道序列为345, 186, 120, 400, 620, 692, 900, 总磁道数为159+66+280+220+72+208=1005。

谢谢大家

考点五:

磁盘格式化



#### 提高磁盘 I/O 速度途径

- 磁盘高速缓存:内存中为磁盘盘块设置的缓冲区,保存某些盘块的副本。指利用内存中的存储空间,来暂存从磁盘中读出的一系列盘块中的信息。因此,这里的高速缓存是一组在逻辑上属于磁盘,而物理上是驻留在内存中的盘块。高速缓存在内存中可分成两种形式。
- 第一种是在内存中开辟一个单独的存储空间来作为磁盘高速缓存,其大小是固定的,不会受应用程序多少的影响;
- 第二种是把所有未利用的内存空间变为一个缓冲池,供请求分页系统和磁盘I/O时(作为磁盘高速缓存)共享。



提高磁盘 I/O 速度途径

提前读: 采用顺序访问方式时, 可以通过预读将磁盘数据先读入高速缓存, 减 少启动磁盘机会



## 提高磁盘 I/O 速度途径

- 延迟写: 应写回磁盘的缓冲区数据先不写回, 而挂在空闲缓冲区队列末尾
- 优化物理块的分布:同一文件的盘块尽量位于同一磁道或相邻磁道,以减少磁
  - 头平均移动距离
- 虚拟盘:通过内存空间模拟磁盘。
- 廉价磁盘冗余阵列 RAID: 通过资源重复解决系统瓶颈问题



# 提高磁盘可靠性的途径

- □ 磁盘容错技术: 增加冗余磁盘设备, 防止系统因素造成的文件不安全性
- □ **后备系统**: 利用后备系统 (磁带、移动硬盘等) 存储暂时不需要但还有用的数

据, 防止自然因素造成的文件不安全性



磁盘低级格式化

在清空某磁盘所有文件的时候,都会使用到磁盘格式化的功能,**磁盘的格式化**操作就是把现有的磁盘或分区里的所有文件删除,从而初始化硬盘。

- 一般格式化分为快速格式化、高级格式化、低级格式化,
- 一般常用快速格式化和高级格式化这两项,而低级格式化在特殊情况才会使用。



磁盘低级格式化

低级格式化就是将空白的磁盘划分出柱面和磁道,再将磁道划分为若干个扇区,每个扇区又划分出标识部分id、间隔区gap和数据区data等。

低级格式化是高级格式化之前的一件工作,而且低级格式化只能针对一块硬盘而不能支持单独的某一个分区。

每块硬盘在出厂时,已由硬盘生产商进行低级格式化,因此通常使用者无需再进行低级格式化操作。

低级格式化是一种损耗性操作,其对硬盘寿命有一定的负面影响。



磁盘低级格式化

低级格式化过程中对硬盘的操作:

1)对扇区清零和重写校验值。低格过程中将每个扇区的所有字节全部置零,并将每个扇区的校验值也写回初始值,这样可以将部分缺陷纠正过来。譬如,由于扇区数据与该扇区的校验值不对应,通常就被报告为校验错误(ECC Error)。如果并非由于磁介质损伤,清零后就很有可能将扇区数据与该扇区的校验值重新对应起来,而达到"修复"该扇区的功效。这是每种低格工具和每种硬盘的低格过程最基本的操作内容,同时这也是为什么通过低格能"修复大量坏道"的基本原因。另外,DM中的Zero Fill(清零)操作与IBM DFT工具中的Erase操作,也有同样的功效。



## 磁盘低级格式化

2) 对扇区的标识信息重写。在多年以前使用的老式硬盘,需要在低格过程中重写每个扇区的标识(ID)信息和某些保留磁道的其他一些信息,当时低格工具都必须有这样的功能。



#### 磁盘低级格式化

3) 对扇区进行读写检查,并尝试替换缺陷扇区。有些低格工具会对每个扇区进行读写检查,如果发现在读过程或写过程出错,就认为该扇区为缺陷扇区。然后,调用通用的自动替换扇区(Automatic reallocation sector)指令,尝试对该扇区进行替换,也可以达到"修复"的功效。



#### 磁盘低级格式化

4) 对所有物理扇区进行重新编号。编号的依据是P-list中的记录及区段分配参数(该参数决定各个磁道划分的扇区数),经过编号后,每个扇区都分配到一个特定的标识信息(ID)。编号时,会自动跳过P-list中所记录的缺陷扇区,使用户无法访问到那些缺陷扇区(用户不必在乎永远用不到的地方的好坏)。如果这个过程半途而废,有可能导致部分甚至所有扇区被报告为标识不对(Sector ID not found, IDNF)。要特别注意的是,这个编号过程是根据真正的物理参数来进行的,如果某些低格工具按逻辑参数(以 16heads 63sector为最典型)来进行低格,是不可能进行这样的操作。



#### 磁盘低级格式化

5) 写磁道伺服信息,对所有磁道进行重新编号。有些硬盘允许将每个磁道的伺服信息重写,并给磁道重新赋予一个编号。编号依据P-list或TS记录来跳过缺陷磁道 (defect track),使用户无法访问(即永远不必使用)这些缺陷磁道。这个操作也是根据真正的物理参数来进行。



#### 磁盘低级格式化

6) 写状态参数,并修改特定参数。有些硬盘会有一个状态参数,记录着低格过程是否正常结束,如果不是正常结束低格,会导致整个硬盘拒绝读写操作,这个参数以富士通IDE硬盘和希捷SCSI硬盘为典型。有些硬盘还可能根据低格过程的记录改写某些参数。



磁盘逻辑格式化

高级格式化是根据一定的分区格式对磁盘的进行标记,生成引导区信息、初始化空间分配表表、标注逻辑坏道、校验数据等,使操作系统能够对磁盘进行读写,所以高级格式化的方法因操作系统的不同而不同,或者说因分区格式的不同而不同。高级格式化只是对磁盘进行寻常的读写操作,对硬盘并没有不利的影响。



#### 磁盘逻辑格式化

格式化又称逻辑格式化,是指根据用户选定的文件系统(例如FAT12、FAT16、FAT32、exFAT、NTFS、EXT2、EXT3等),在硬盘的特定区域写入特定数据,以达到初始化硬盘或硬盘分区、清除原硬盘或硬盘分区中所有数据的操作。

高级格式化操作包括:对主引导记录中分区表相应区域的重写;根据用户选定的文件系统,在硬盘分区中划出一块用于存放文件分配表、目录表等用于文件管理的硬盘空间,以便用户管理硬盘分区中的文件。

谢谢大家

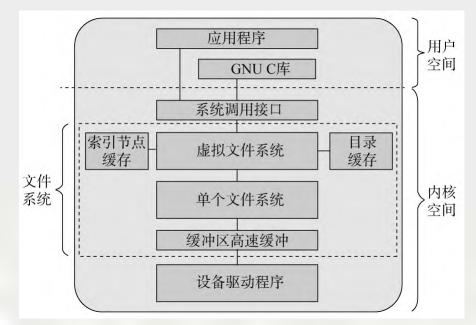
考点六:

文件管理系统和VFS

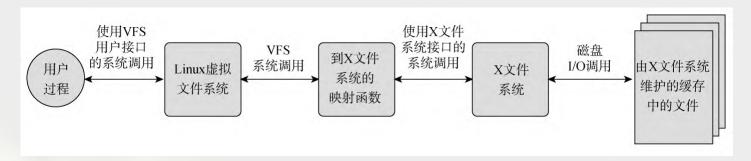
Linux包含一个强有力的通用文件处理机制,这种机制利用虚拟文件系统 (Virtual File System,VFS)来支持大量的文件管理系统和文件结构。VFS向用户 进程提供一个简单且统一的文件系统接口。VPS定义了一个能代表任何文件系统 的通用特征和行为的通用文件模型。

VFS认为文件是计算机大容量存储器上的对象。这些计算机大容量存储器具有共同的特征,它与目标文件系统或底层的处理器硬件无关。文件有一个符号名,以便在一个文件系统的特定目录下能唯一地标识该文件。同时,文件有一个所有者、对未授权访问或修改的保护和其他系列属性。文件可被创建、读写或删除。任何文件系统都需要一个映射模块,以便将实际文件系统的特征转换为虚拟文件

系统所期望的特征。



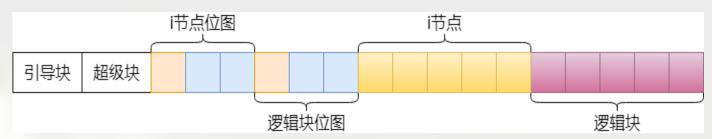
VFS在 Linux内核中的作用如图所示。当进程发起一个面向文件的系统调用时,内核用VPS中的一个函数。该函数处理完与具体文件系统无关的操作后,调用目标文件系统中的相应函数。这个调用通过一个将VFS的调用转换到目标文件系统调用的映射函数来实现。VFS与任何具体的文件系统无关。因此映射函数的实现是文件系统在Linux上的实现的一部分。目标文件系统将文件系统请求转换到面向设备的指令。



VFS是面向对象的方案。因为VPS 不是用面向对象语言(如C++和 Java)实现的,而是用C语言实现的,因此VFS的对象简单实现为C语言的结构。每个对象都包含数据和函数指针。这些函数指针指向操作这些数据的文件系统的实现函数。

#### VFS主要的4个对象如下:

- ➤ 超级块对象(superblock object):表示一个已挂载的特定文件系统。
- > 索引节点对象 (inode object):表示一个特定的文件。
- ▶ 目录项对象 (dentry object):表示一个特定的目录项。
- > 文件对象 (file object):表示一个与进程相关的已打开文件。



1) 超级块:一个超级块对应一个文件系统(已经安装的文件系统类型如ext2,此处是实际的文件系统哦,不是VFS)。之前我们已经说了文件系统用于管理这些文件的数据格式和操作之类的,系统文件有系统文件自己的文件系统,同时对于不同的磁盘分区也有可以是不同的文件系统。那么一个超级块对于一个独立的文件系统。保存文件系统的类型、大小、状态等等。

2) 索引节点inode:保存的其实是实际的数据的一些信息,这些信息称为"元数据"(也就是对文件属性的描述)。例如:文件大小,设备标识符,用户标识符,用户组标识符,文件模式,扩展属性,文件读取或修改的时间戳,链接数量,指向存储该内容的磁盘区块的指针,文件分类等等。(注意数据分成:元数据+数据本身)同时注意:inode有两种,一种是VFS的inode,一种是具体文件系统的inode。前者在内存中,后者在磁盘中。所以每次其实是将磁盘中的inode调进填充内存中的inode,这样才是算使用了磁盘文件inode。

注意inode怎样生成的:每个inode节点的大小,一般是128字节或256字节。inode节点的总数,在格式化时就给定(现代OS可以动态变化),一般每2KB就设置一个inode。一般文件系统中很少有文件小于2KB的,所以预定按照2KB分,一般inode是用不完的。所以inode在文件系统安装的时候会有一个默认数量,后期会根据实际的需要发生变化。

注意inode号: inode号是唯一的,表示不同的文件。其实在Linux内部的时候,访问文件都是通过inode号来进行的,所谓文件名仅仅是给用户容易使用的。当我们打开一个文件的时候,首先,系统找到这个文件名对应的inode号; 然后,通过inode号,得到inode信息,最后,由inode找到文件数据所在的block,现在可以处理文件数据了。

inode和文件的关系: 当创建一个文件的时候,就给文件分配了一个inode。一个inode只对应一个实际文件,一个文件也会只有一个inode。inodes最大数量就是文件的最大数量。

3) 目录项: 目录项是描述文件的逻辑属性,只存在于内存中,并没有实际对应的磁盘上的描述,更确切的说是存在于内存的目录项缓存,为了提高查找性能而设计。注意不管是文件夹还是最终的文件,都是属于目录项,所有的目录项在一起构成一颗庞大的目录树。例如: open一个文件/home/xxx/yyy.txt,那么/、home、xxx、yyy.txt都是一个目录项,VFS在查找的时候,根据一层一层的目录项找到对应的每个目录项的inode,那么沿着目录项进行操作就可以找到最终的文件。

注意:目录也是一种文件(所以也存在对应的inode)。打开目录,实际上就是打开目录文件。

4) 文件对象:注意文件对象描述的是进程已经打开的文件。因为一个文件可以被多个进程打开,所以一个文件可以存在多个文件对象。但是由于文件是唯一的,那么inode就是唯一的,目录项也是定的!

进程其实是通过文件描述符来操作文件的,注意每个文件都有一个32位的数字来表示下一个读写的字节位置,这个数字叫做文件位置。一般情况下打开文件后,打开位置都是从0开始,除非一些特殊情况。Linux用file结构体来保存打开的文件的位置,所以file称为打开的文件描述。这个需要好好理解一下! file结构形成一个双链表,称为系统打开文件表。

谢谢大家

考点七:

SSD

#### 考点七: SSD

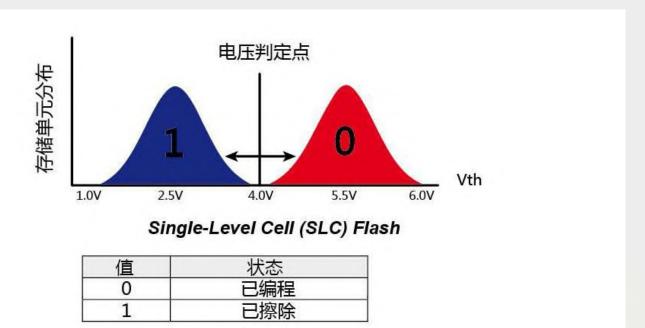
SSD (Solid State Drive),即固态硬盘,是一种以半导体闪存(NAND FLASH)为介质的存储设备。SSD组成:SSD主要部件为控制器和存储芯片,SSD硬件包括主控,闪存 ,缓存芯片DRAM(可选),PCB(电源芯片,电阻,电容等),接口(SATA,PCIE等),其主体就是一块PCB。从软件角度看SSD:SSD内部运行固件(Firmware)负责调度数据从接口端到介质端的读写,还包括嵌入核心的内存介质寿命和可靠性管理的调度算法以及SSD内部算法。SSD三大技术核心:SSD控制器,闪存和固件

#### 考点七: SSD

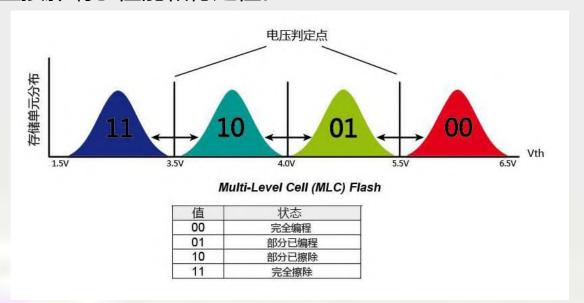
SSD 的基本存储单元分为三类: SLC(Single Level Cell 单层单元) 和MLC(Multi-Level Cell 多层单元) 以及较新的 TLC(Triple Level Cell 三层单元)。

低端产品一般采用 MLC(multi-level cell) 或者 TLC(Triple Level Cell) 闪存,其特点是功耗高、容量大、速度慢(2MB/S)、可靠性低、存取次数低(3000次 [25nm]-1万次[50nm],制程越先进次数反而越小)、价格也低。高端产品一般采用 SLC(single-level cell) 闪存,其特点是技术成熟、功耗低、容量小、速度快 (8MB/S)、可靠性高、存取次数高(10万次)、价格也高。造成这种差异的原因在于,每个 MLC/TLC 存储单元中存放的资料较多,结构相对复杂,出错的几率会增加,必须进行错误修正,这个动作导致其性能和可靠性大幅落后于结构简单的 SLC 闪存。

SLC 就是在每个存储单元里存储 1bit 的数据,存储的数据是0还是1是基于电压 阀值的判定,对于 NAND Flash 的写入(编程),就是控制 Control Gate 去充电 (对 Control Gate 加压),使得浮置栅极存储的电荷够多,超过4V,存储单元就 表示 0(已编程),如果没有充电或者电压阀值低于4V,就表示 1(已擦除)。



MLC 就是每个存储单元里存储 2bit 的数据,存储的数据是"00","01","10","11"也是基于电压阀值的判定,当充入的电荷不足3.5V时,就代表"11",当充入的电荷在3.5V和4.0V之间,则代表"10",当充入的电荷在4V和5.5V之间,则表示"01",当充入的电荷在5.5V以上,则表示"00"。同时由前面的图可以看到,MLC 相比 SLC 虽然使用相同的电压值,但是电压之间的阀值被分成了4份,可以想象这样就直接影响了性能和稳定性。



而 TLC 就更加复杂,因为每个存储单元里存储 3bit 的数据,所以它的电压阈值的分界点就更细致,导致的结果也就每个存储单元的可靠性也更低

> 静态磨损和动态磨损

就磨损均衡命令自身而言,根据算法的差异,可分为动态磨损和静态磨损。目前,大多数固态硬盘产品都是使用动态磨损,不采用静态磨损很重要的原因在于过高的写入放大。

> 静态磨损和动态磨损

动态磨损就是主控芯片,会优先选择"较新"的闪存颗粒进行擦除写入,"较老"的颗粒则不使用。这种方式的优点在于,算法简单粗暴,主控处理压力小,占据的资源小,而缺点就很明显,算法太过一刀切,优化不够,精细化程度不够导致无法全面覆盖和实现所有颗粒的磨损均衡。

#### > 静态磨损和动态磨损

静态磨损就是主控芯片在执行擦除写入的命令时,从文件的角度出发,优先把长久不用的冷数据从较新的闪存颗粒中提出来,放入较老的闪存颗粒中,并将写入的新数据放在较新的闪存颗粒中,以实现均衡化。这样的方式优点也很明显,算法更加优化,从数据角度出发更全面的实现了磨损均衡,并提高了颗粒的寿命,而缺点在于算法的复杂,给主控带了相当的压力,占有了更多的读写资源,并影响部分写入速度。

	优点	缺点
动态磨损	算法简单、主控压力小、读写占据资源小	算法优化不足、效果不够全面
静态磨损	科学全面实现颗粒均衡、提高颗粒耐久度	主控压力大、影响写人速度、写人放大明显



- 1、下列文件物理结构中,适合随机访问且易于文件扩展的是\_\_\_\_。
- A. 连续结构

- B. 索引结构
- C. 链式结构且磁盘块定长
- D. 链式结构且磁盘块变长
- B【解析】文件物理结构就是文件的内容存储在计算机中所采用的方式,文件在计算机中有三种物理结构: 顺序结构使文件内容按顺序存在连续的存储单元中,由于地址连续,这就使顺序结构能够随机访问但元素移动能力差;链式结构指文件内容不一定放在连续的内存单元中,这些单元会开辟一小块单元存放下一单元的地址,依次类推形成一种地址链,由此可知链式结构能很好地移动、增加元素但只能顺着链子访问目标元素;索引结构的文件内容同样不一定放在连续单元,而是通过为每一个文件建立一个索引表,标明逻辑块号和物理地址的对应关系,索引结构可以通过索引表随机访问内容,也可以修改索引表对文件扩展。
- C、D项不论磁盘块是否定长,链式结构都无法使文件随机访问,由以上分析可知 B项索引结构满足题目要求,选B。

2、假设磁头当前位于第 105 道,正在向磁道序号增加的方向移动。现有一个磁道 访问请求序列为 35, 45, 12, 68, 110, 180, 170, 195, 采用 SCAN 调度 (电梯调度)算法得到的磁道访问序列是 。

A. 110,170,180,195,68,45,35,12

B. 110,68,45,35,12,170,180,195

C. 110,170,180,195,12,35,45,68 D. 12,35,45,68,110,170,180,195

A【解析】磁道访问的电梯调度算法是指,确定了当前移动方向后,磁头会由近 到远访问到最远处的磁道, 然后再调转方向对剩下的磁道进行访问。对于这种题 目,分为三步。首先将请求序列按从小到大的顺序排列好,然后确定磁头在序列 的位置,最后确定移动方向得到图中移动过程。

本题中磁头位于105道且往增加方向移动,那么就会从小到大经过110、170、 180、195,到达195磁道后,再反向移动回到105磁道,再从大到小经过68、 45、35、12, 最终可得访问序列为110,170,180,195,68,45,35,12, 选择A项。 3、文件系统中,文件访问控制信息存储的合理位置是\_\_\_\_。

A. 文件控制块

B. 文件分配表

C. 用户口令表

D. 系统注册表

A【解析】A项文件控制块(FCB)是文件十分重要的部分,其涉及文件的基本信息、存取控制信息和使用信息。

B项文件分配表 (FAT) 是文件管理系统记录每个文件所在的磁盘位置,与文件控制信息无关。

C项用户口令表保存着用户的口令和密码,并不涉及文件的访问控制。

D项系统注册表用于存放系统启动、程序运行装载的相关配置和设置信息,也与访问控制无关。由以上分析可知本题选A。

4、设文件 F1 的当前引用计数值为 1, 先建立 F1 的符号链接(软链接)文件 F2, 再建立 F1 的硬链接文件 F3, 然后删除 F1。此时, F2 和 F3 的引用计数值分别 是 \_\_\_\_。

A. 0, 1

B. 1, 1 C. 1, 2 D. 2, 1

B【解析】每个文件都有自己的索引节点号iNode,用于系统对文件进行操作。 Linux系统允许多个文件名指向同一个索引节点号,使用不同的文件名可以访问、 修改同一iNode指向的文件内容,以防止"误删"操作,设置的文件名就是硬链 接。每为一个iNode设置一个新的文件名,该iNode的链接数都会+1。每删除一 个文件名链接数都会-1, 但删除文件名并不会删除文件内容也不会影响其他文件 名的使用,只有当链接数减为0,也即没有文件名指向该索引节点,此时系统会 回收该索引节点及其对应的文件内容,也就删除了文件。

4、设文件 F1 的当前引用计数值为 1, 先建立 F1 的符号链接(软链接)文件 F2, 再建立 F1 的硬链接文件 F3, 然后删除 F1。此时, F2 和 F3 的引用计数值分别 是 \_\_\_\_。

A. 0, 1

B. 1, 1 C. 1, 2 D. 2, 1

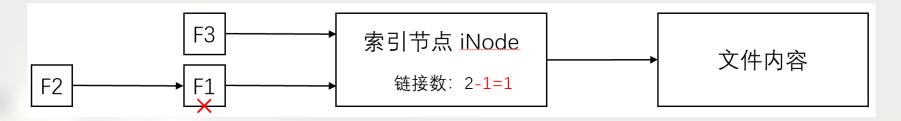
B【解析】不同于硬链接,软链接文件是一个独立的文件,拥有独立的iNode。 软链接也称符号链接,软链接文件存放的是另一个文件的路径而非另一文件指向 的iNode, 当建立软链接时, 软链接文件会直接复制源文件的链接数, 而不会增 加源文件的链接数。类似于windows的快捷方式,访问软链接文件就会去读取源 文件,这就意味着软链接文件是依赖于源文件的,当源文件被删除,访问软链接 文件就会报错。

4、设文件 F1 的当前引用计数值为 1, 先建立 F1 的符号链接(软链接)文件 F2, 再建立 F1 的硬链接文件 F3, 然后删除 F1。此时, F2 和 F3 的引用计数值分别 是\_\_\_。

A. 0, 1

B. 1, 1 C. 1, 2 D. 2, 1

B【解析】本题中的引用计数就是链接数。在为F1建立软链接文件F2时,F2复制 F1的链接数,F2的链接数为1。F1又建立硬链接文件F3,FI和F3指向的索引节点 中的链接数加1为2。再删除了F1后,只有一个文件名F3指向该索引节点,链接 数减1得到1, F2的链接数不变, 但不能访问了。所以最后F2和F3的链接数都是1, 选B。



5、设文件索引节点中有7个地址项,其中4个地址项是直接地址索引,2个地址项 是一级间接地址索引, 1个地址项是二级间接地址索引, 每个地址项大小为4字节。 若磁盘索引块和磁盘数据块大小均为256 字节, 则可表示的单个文件最大长度是

A. 33 KB

B. 519 KB C. 1057 KB D. 16513 KB

C【解析】混合索引结构当个文件的大小是各个地址所指示的空间之和。每个磁盘 索引块和磁盘数据块大小均为256B,每个磁盘索引块有256/4=64个地址项。因此, 每个直接地址索引指向的数据块大小为4×256B; 2个一级间接索引包含的直接地 址索引数为2×(256/4),即其指向的数据块大小为2×(256/4)×256B。1个二级间接 索引所包含的直接地址索引数为(256/4)×(256/4),即其所指向的数据块大小为 (256/4)(256/4) × 256B。 即 7 个 地 址 项 所 指 向 的 数 据 块 总 大 小 为  $4\times256+2\times(256/4)\times256+(256/4)\times(256/4)\times256=1082368B=1057KB$ 

- 6、设置当前工作目录的主要目的是。
- A. 节省外存空间 B. 节省内存空间
- C. 加快文件的检索速度 D. 加快文件的读/写速度
- C【解析】设置当前工作目录的主要目的是加快文件的检索速度。当前工作目录是指操作系统在进行相对路径文件操作时,查找文件时首先从当前工作目录开始查找的目录,通过设置当前工作目录,可以缩小查找文件的范围,提高查找效率。因此,选项 C 正确。其他选项不是当前工作目录的主要目的。

- 7、下列选项中,不能改善磁盘设备 I/O 性能的是\_\_\_\_\_
- A. 重排 I/O 请求次序 B. 在一个磁盘上设置多个分区
- C. 预读和滞后写 D. 优化文件物理的分布
- B【解析】磁盘设备 I/O包括寻道时间、旋转延迟和读写时间,
- □ 那么重排 I/O 请求次序可以改善寻道时间, 因此A 可以优化I/O性能,
- □ 优化文件物理的分布可以改善寻道时间和旋转延迟, 因此D可以优化I/O性能。
- □ 而预读是指文件读取通常采用预读来预测用户的行为,把用户可能需要的数据 预先读取到内存中,滞后写是将数据保存在缓冲区中,等缓冲区满时再写入磁 盘,这种方式可以极大的提高磁盘写入数据的效率,因此C可以优化I/O性能。
- □ B选项不能优化I/O性能,虽然设置多个分区,但是每个分区的I/O时间没有得到改善。

- 8、用户在删除某文件的过程中,操作系统不可能执行的操作是\_\_\_\_。
- A. 删除此文件所在的目录

B. 删除与此文件关联的目录项

C. 删除与此文件对应的文件控制块

D. 释放与此文件关联的内存级冲区

A【解析】删除文件不需要删除文件所在的目录,而文件的关联目录项和文件控制块需要随着文件一同删除,同时释放文件的关联缓冲区。此文件所在目录下可能还存在其他文件,因此删除文件时不能(也不需要)删除文件所在 的目录,而与此文件关联的目录项和文件控制块需要随着文件一同删除,同时释放文件关联的 内存缓冲区。

- 9、为支持 CD-ROM 中视频文件的快速随机播放,播放性能最好的文件数据块组织方式是\_\_\_\_。
- A. 连续结构 B. 链式结构 C. 直接索引结构 D. 多级索引结钩
- A【解析】CD-ROM 中的文件采用顺序方式存储,因此要保证最短的查询时间,即不能选取链表和索引结构,因此连续结构最优。

- 10、若某文件系统索引结点 (inode) 中有直接地址项和间接地址项,则下列选项
- 中,与单个文件长度无关的因素是。

A. 索引结点的总数 B. 间接地址索引的级数

C. 地址项的个数

D. 文件块大小

A【解析】索引结点的总数等于文件的总数,索引结点越多对应于文件总数越多, 与单个文件的长度无关;根据混合索引文件大小的计算原理,间接地址级数越多、 地址项数越多、文件块越大,单个文件的长度就会越大。

- 11、在一个文件被用户进程首次打开的过程中,操作系统需做的是\_\_\_\_。
- A. 将文件内容读到内存中
- B. 将文件控制块读到内存中
- C. 修改文件控制块中的读写权限
- D. 将文件的数据缓冲区首指针返回给用户进程
- B【解析】一个文件被用户进程首次打开的过程中,操作系统需要使用open系统调用,将把文件的 FCB(文件的管理控制信息或者文件的属性信息) 调入内存,并把文件放入到系统打开文件表中,以供进程后续的使用。因此选B。而当关闭文件时,将把文件的 FCB写入磁盘,同时也将文件的更新写入磁盘,并将文件从系统打开文件表中删除。

12、在系统内存中设置磁盘缓冲区的主要目的是。

A. 减少磁盘 I/O 次数 B. 减少平均寻道时间

C. 提高磁盘数据可靠性 D. 实现设备无关性

A【解析】磁盘和内存之间存在巨大的速度差异,通常在内存和磁盘之间设置缓冲 区,该缓冲区位于内存中。可以将磁盘中的数据先放入到磁盘缓冲区中,之后再从 磁盘缓冲区读入到内存, 从而减少磁盘 I/O 次数。请注意, 磁盘缓冲区, 逻辑上 属于磁盘,物理上属于内存。

13、在文件的索引节点中存放直接索引指针10个,一级和二级索引指针各1个。磁 盘块大小为1KB, 每个索引指针占4个字节。若某文件的索引节点已在内存中,则 把该文件偏移量(按字节编址)为 1234和 307400处所在的磁盘块读入内存,需 访问的磁盘块个数分别是。

A. 1,2 B. 1,3 C. 2,3 D. 2,4

B【解析】每个索引指针占4B,则每个磁盘块可存放IKB/4B=256个索引指针。10个 直接索引指针指向的数据块大小为10\*1KB=10KB; 一级索引指针指向的数据块大 小 为 : 256\*1KB=256KB , 二 级 索 引 指 针 指 向 的 数 据 块 大 小 为 :  $256*256*1KB=2^{16}KB=64MB$ 

13、在文件的索引节点中存放直接索引指针10个,一级和二级索引指针各1个。磁 盘块大小为1KB, 每个索引指针占4个字节。若某文件的索引节点已在内存中,则 把该文件偏移量(按字节编址)为 1234和 307400处所在的磁盘块读入内存,需 访问的磁盘块个数分别是。

A. 1,2 B. 1,3 C. 2,3 D. 2,4

B【解析】系统按照字节编址,偏移量为1234时,因1234B<10KB,则由直接索引 指针可得到其所在的磁盘块地址故仅需1次访盘即可。偏移量为307400时,因 10KB+256KB<307400B<64MB,可知该偏移量的内容在二级索引指针所指向的 某个磁盘块中, 故共需3次访盘。

14、文件系统用位图法表示磁盘空间的分配情况,位图存于磁盘的 32~127 号块中,每个盘块占1024个字节,盘块和块内字节均从0开始编号。假设要释放的盘块号为 409612,则位图中要修改的位所在的盘块号和块内字节序号分别是\_\_\_\_。A. 81、1 B. 81、2 C. 82、1 D. 82、2

C【解析】因为每个磁盘是1024B,那么每个磁盘包含1024 \* 8比特。当采用位图 法表示磁盘空间的分配情况时,每个磁盘可以表示1024 \* 8个磁盘的分配情况(1 个磁盘需要1个比特)。盘块号为 409612多占用的盘块号=起始块号+[(盘块号/(1024\*8))]=32+[(409612/(1024\*8))]=32+50=82。而盘块号为 409612位于82 块的字节号是[(盘块号%(1024\*8)/8)]=1。

15、某硬盘有200个磁道 (最外侧磁道号为 0), 磁道访问请求序列为: 130,42,180,15,199, 当前磁头位于第 58 号磁道并从外侧向内侧移动。按照 SCAN 调度方法处理完上述请求后,磁头移过的磁道数是。。

A. 208

B. 287 C. 325

D. 382

C【解析】SCAN算法就是电梯调度算法,既要考虑移动距离、又要考虑磁头的移 动方向。如果最外侧磁道号为 0, 表示外侧是小编号磁道。当前磁头位于第 58 号磁道并从外侧向内侧移动,意味着磁头的移动是从小编号磁道到大编号磁道移 动。于是当前磁头位于58号并从外侧向内侧移动,依次访问130和199,然后再 返回向外侧移动,依次访问42和15,故磁头移过的磁道数是: (199-58)+(199-15)=325.

16. 某文件系统的簇和磁盘扇区大小分别为1KB和512B。若一个文件的大小为1 026B,则系统分配给该文件的磁盘空间大小是。

A. 1026 B B. 1536 B C. 1538 B D. 2048 B

D【解析】在磁盘分配时,存在两个分配的单位,即块和簇。其使用原则是,如果 只有块描述,就按照块来进行分配;如果只有簇描述,就按照簇来进行分配;如 果二者都有,就按照簇来进行分配;

根据题目,我们知道按照簇进行分配,一个文件的大小为1 026 B,簇大小是1KB (1024B), 因此需要2个簇, 大小是2\*1024=2048B。

- 17. 下列选项中, 磁盘逻辑格式化程序所做的工作是 。
- I. 对磁盘进行分区
- Ⅱ. 建立文件系统的根目录
- Ⅲ. 确定磁盘扇区校验码所占位数
- IV. 对保存空闲磁盘块信息的数据结构进行初始化
- A. 仅口
- B. 仅II、IV

- C. 仅皿、IV D. 仅 I、 I、IV

B【解析】磁盘的格式化分为物理格式化和逻辑格式化。物理格式化又称低级格 式化,是对磁盘的物理表面进行处理,在磁盘上建立标准的磁盘记录格式,划分 磁道 (track) 和扇区 (sector), 对应于I。同时低级格式化为磁盘的每个扇区采 用特别的数据结构,包括校验码, 对应于111。逻辑格式化又称高级格式化,是在 磁盘上建立一个系统存储区域,包括引导记录区、文件目录区FCT、文件分配表 FAT,对应于II和IV。因此I和III是物理格式化,II和IV是逻辑格式化。因此选B。

18. 某文件系统中,针对每个文件,用户类别分为4类:安全管理员、文件主、文件主的伙伴、其他用户;访问权限分为5种:完全控制、执行、修改、读取、写入。若文件控制块中用二进制位串表示文件权限,为表示不同类别用户对一个文件的访问权限,则描述文件权限的位数至少应为\_\_\_\_。

A. 5

B. 9

C. 12

D. 20

D【解析】在系统中,4类用户是同时存在的,为了对不同用户的不同权限进行标记,可以把用户访问权限抽象为一个矩阵,行代表用户,列代表访问权限。这个矩阵有4行5列,1代表 true,0代表 false,所以需要20位,选D。

- 19. 若文件f1的硬链接为f2,两个进程分别打开f1和f2,获得对应的文件描述符为 fd1和fd2,则下列叙述中,正确的是。
- I. f1和f2的读写指针位置保持相同
- Ⅱ. f1和f2共享同一个内存索引结点
- Ⅲ.fd1和fd2分别指向各自的用户打开文件表中的一项
- A. 仅皿

- B. 仅 I、 I C. 仅 I、 I D. I、 I 和 II
- B【解析】文件的硬链接共享方式是,文件只有一个索引结点,每次新增一个共 享进程,就将文件索引结点中的共享计数器count加1,且多个文件指向同一个索 引结点,因此||正确。不同讲程打开的文件不同,因此每个文件都有自己的打开 文件表,当共享文件时,文件描述符为fd1和fd2分别指向各自的用户打开文件表 中的一项, III正确。进程对于文件的访问是不一样的, 读写的过程也是不一样的, 各自维持自己的读写指针,因此f1和f2的读写指针位置不一定保持相同,I错误。 因此选B。

谢谢大家