



操作系统结构分析

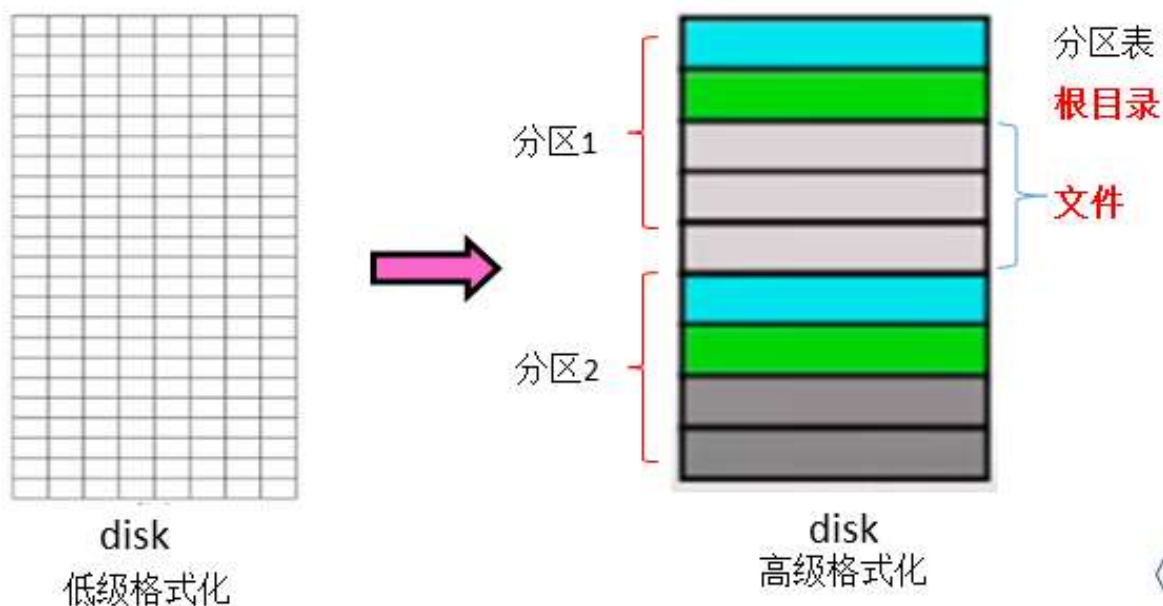
第6章文件管理

南京邮电大学
计算机学院 信息安全系
曹晓梅 陈丹伟

手机: 189-0518-4599
QQ: 757375652
email: caoxm@njupt.edu.cn

引言

对于用户而言，文件系统是操作系统中最明显的部分。它提供机制，以便对系统和用户的数据与程序进行在线存储和访问。文件系统指文件、管理文件的软件及数据结构的总体



2

内容纲要

Contents Page



1.文件

2.目录

3. 文件系统的实现



3

纲要

1. 文件 »

1.1 基础知识



- I. 怎么看文件?
- II. 文件系统功能
- III. 文件的属性
- IV. 文件的命名
- V. 文件的类型

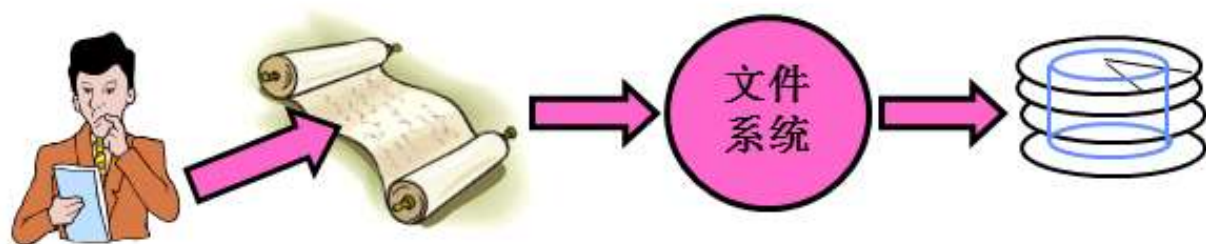
1.2 文件的逻辑结构

1.3 文件的访问方法

1.4 文件的访问控制

4

□ 怎么看文件？



• 用户观点 文件逻辑结构

文件是一些具有结构的信息集合，如一篇文章分成很多段落、一个程序分成很多函数。

• 系统观点 文件物理结构

文件就是一组存放在扇区里的数据，不关心它的内容和格式。

5

□ 文件系统功能

- OS的组成部分，管理文件的存储、检索、更新，提供安全可靠的共享和保护手段。
- 文件系统的主要目的是实现对文件的“按名存取”，具体功能如下：
 - ✓ 文件的命名
 - ✓ 实现文件从逻辑结构到物理结构的映射
 - ✓ 统一管理文件存储空间，实施存储空间的分配与回收
 - ✓ 实现文件信息的共享，并提供文件保护和保密机制
 - ✓ 向用户提供一个方便使用的接口

1.1 基础知识

□ 文件的属性

- 文件名 **按名存取**
- 类型
- 大小
- 位置
- 创建时间、修改时间
- 文件所有者ID
- 存取权限



1.1 基础知识

1. 文件 »

□ 文件的命名

- 各个操作系统的命名规则略有不同，其一般形式为“**文件名.扩展名**”
- 命名规则一般包含：
 - ✓ 文件长度：Windows的文件名允许长达255个字符
 - ✓ 特殊字符：Windows文件名中不允许出现\、/、<、>等字符
 - ✓ 大小写：Windows不区分文件名的大小写，但Unix/Linux区分
- 扩展名用于定义文件的类型

□ 文件的类型

文件类型	常用扩展名	功能
可执行文件	exe, com	可运行的机器语言程序
目标文件	obj, o	已编译的、尚未链接的机器语言
源代码文件	c, cpp, java, perl, asm	各种语言的源代码
批处理文件	bat, sh	命令解释程序的命令
标记文件	xml, html, tex	文本数据、文档
文字处理文件	xml, rtf, docx	各种文字处理程序的格式
库文件	lib, a, so, dll	为程序员提供的程序库
打印或可视文件	gif, pdf, jpg	打印或图像格式的ASCII或二进制文件
档案文件	rar, zip, tar	相关文件组成的一个文件，用于归档或存储
多媒体文件	mpeg, mov, mp3, mp4, avi	包含音频或A/V信息的二进制文件

纲要

1. 文件 »

1.1 基础知识

1.2 文件的逻辑结构



1.3 文件的访问方法

1.4 文件的访问控制

I. 总述

II. 流式文件

III. 记录式文件

IV. 文件记录的成组与分解

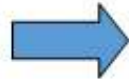
1.2 文件的逻辑结构

1. 文件 »

□ 总述

- 文件结构（File Structure）

逻辑结构
Logical Structure



从用户观点出发所观察到的文件组织形式，独立于文件的物理特性。

1.2

物理结构
Physical Structure



从系统实现观点出发，文件在外存上的存储组织形式，与存储介质的存储特性、所采用的外存分配方式有关。

3.3

- 文件逻辑结构

流式文件

记录式文件

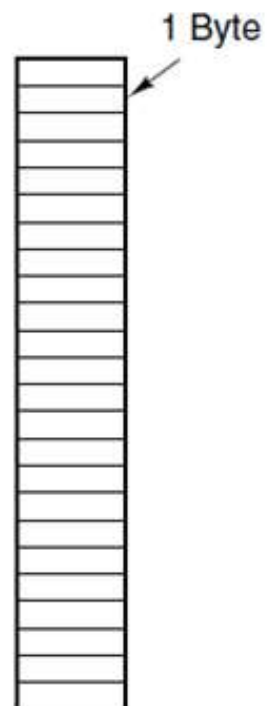
11

1.2 文件的逻辑结构

1. 文件 »

□ 流式文件

- 文件中的信息仅仅是一串顺序的字符集合，没有明显的结构。操作系统所见到的只是字节流，其文件内容的任何含义在用户程序中解释。
- 简单易行，多数现代操作对用户只提供此种文件类型，例如所有UNIX版本和Windows等。



字节序列

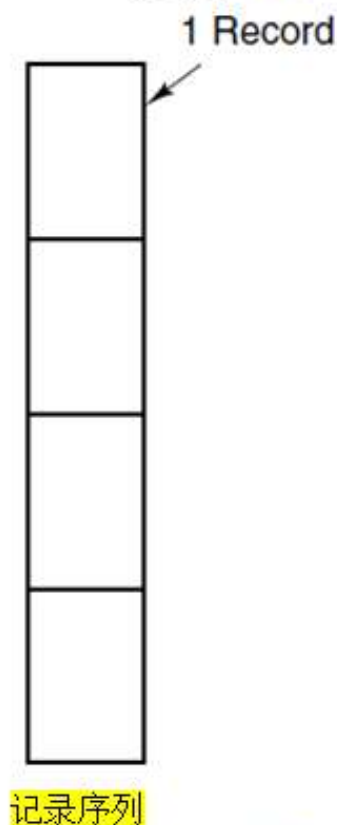
12

1.2 文件的逻辑结构

1. 文件 »

□ 记录式文件

- 文件的信息单位是逻辑记录，一个逻辑记录包含了若干数据项，记录分为：
 - ✓ 定长记录
 - ✓ 变长记录
- 每条记录中至少含有一个记录键/关键字，用于区别同一文件的不同记录。
- 常用于处理商业数据的大型计算机，如IBM OS/370等，一般需要高级语言程序和数据库管理系统支持。



13

1.2 文件的逻辑结构

1. 文件 »

□ 文件记录的成组与分解

磁盘空间分配、内存和磁盘之间的I/O传输以块（簇）为单位的，逻辑记录的大小和物理块不相等。

- 成组：将若干逻辑记录打包成组后再装入物理块的过程。
- 分解：把一个物理块里的逻辑记录分离出来的过程。



14

1.1 基础知识

1.2 文件的逻辑结构

1.3 文件的访问方法



- I. 顺序访问
- II. 直接访问
- III. 索引访问

1.4 文件的访问控制

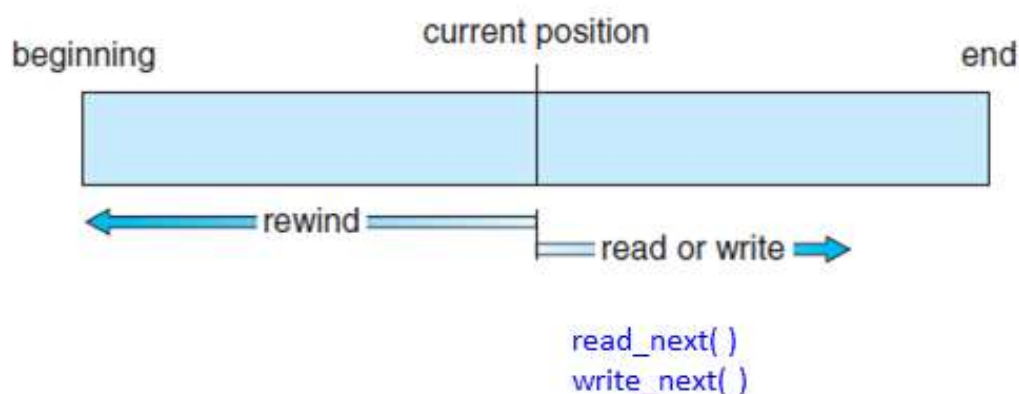
15

1.3 文件的访问方法

1. 文件 »

□ 顺序访问(sequential access) 磁带模型

- 最简单的访问方法。
- 文件信息按顺序（即一个记录接着一个记录地）加以处理，读取/写入当前文件信息时，将文件指针移向下一个临接记录。



16

□ 直接访问(direct access) 磁盘模型

- 若文件的逻辑记录长度固定，那么允许在访问文件信息时可按任意顺序进行快速读取和写入。

read (N)

write (N)

读写文件的第N个逻辑记录（编号从0开始）

假设逻辑记录长度为L，访问则可转换成：

“访问从文件起始位置 $L*N$ 开始的L个字节”

17

□ 索引访问(index access)

- 一个逻辑文件中的记录过多时，读取一个想要的块可能要花费大量的时间去查找。
- 对记录创建索引表，文件的记录按键值排序，用户通过键值查找文件记录(通常使用折半查找)。
- 此方法需要占用存储空间存放索引表，索引表的大小与文件记录的数量成正比。

18

1.1 基础知识

1.2 文件的逻辑结构

1.3 文件的访问方法

1.4 文件的访问控制



19

1.4 文件的访问控制

Linux的访问控制

- 对文件的访问类型一般有：读r、与w、执行x

- 为了实现基于身份的访问控制，引入了访问控制列表(ACL)和访问类型。

- 文件所有者，同组

- Linux的文件保护

- 第1位表示普通文件(-)，目录(d)，后三个域分别表示三类用户的访问权限

- 使用“chmod xxx filename”可以更改文件的保护属性

0~7

```
caoxm@ubuntu:~$ chmod 777 pi.c
```

```
caoxm@ubuntu:~$ ls -l
total 64
drwxr-xr-x 2 caoxm caoxm 4096 Oct  7 21:38 Desktop
drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Documents
drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Downloads
drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Music
drwxrwxr-x 5 caoxm caoxm 4096 Nov 22 21:23 OS
-rwxrwxr-x 1 caoxm caoxm 16688 Oct 11 19:39 pi
-rw-r--r-- 1 caoxm caoxm 218 Oct 11 19:39 pi.c
```

```
caoxm@ubuntu:~$ ls -l
total 64
drwxr-xr-x 2 caoxm caoxm 4096 Oct  7 21:38 Desktop
drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Documents
drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Downloads
drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Music
drwxrwxr-x 5 caoxm caoxm 4096 Nov 22 21:23 OS
-rwxrwxr-x 1 caoxm caoxm 16688 Oct 11 19:39 pi
-rwxrwxrwx 1 caoxm caoxm 218 Oct 11 19:39 pi.c
```

单选题 1分

CQ1.1 文件系统是指（ ）。

- ☐ A 文件的集合
- ☐ B 文件的目录
- ☐ C 实现文件管理的一组软件
- ☒ D 文件、管理文件的软件及数据结构的总体

单选题 1分

CQ1.2 文件系统的主要目的是（ ）。

- ☒ A 实现对文件的“按名存取”
- ☐ B 实现虚拟存储
- ☐ C 提高外存的读写速度
- ☐ D 存储系统文件

单选题 1分

CQ1.3 下列 () 不是文件系统的功能。

- ☐ A 实现对文件的“按名存取”
- ☐ B 负责实现逻辑文件到物理文件的转换
- ☒ C 提高磁盘的读写速度
- ☐ D 提供对文件的存取方法和对文件的操作

单选题 1分

CQ1.4 由字符序列组成，文件内的信息不再划分结构，这是指 ()。

- ☒ A 流式文件
- ☐ B 记录式文件
- ☐ C 顺序文件
- ☐ D 有序文件

单选题 1分

CQ1.5 文件的顺序存取指的是（ ）。

- ☐ A 按终端号依次存取
- ☒ B 按文件记录的逻辑序号逐一存取
- ☐ C 按文件的物理块号依次存取
- ☐ D 按文件逻辑记录的大小逐一存取

单选题 1分

CQ1.6 以下关于文件存取的叙述中，正确的是（ ）。

- ☐ A 适合于顺序存取的文件也一定适合随机存取
- ☒ B 适合于随机存取的文件也一定适合顺序存取
- ☐ C 适合于随机存取的文件不一定适合顺序存取
- ☐ D 以上都不对

内容纲要

Contents Page



1. 文件

2. 目录

3. 文件系统的实现



27

纲要

2.1 概述



- I. 如何组织用户文件?
- II. 目录管理的功能
- III. 目录文件
- IV. 目录项

2.2 文件控制块和索引节点

2.3 目录结构

2.4 目录的检索和实现

2.5 文件的共享

2. 目录 »

28

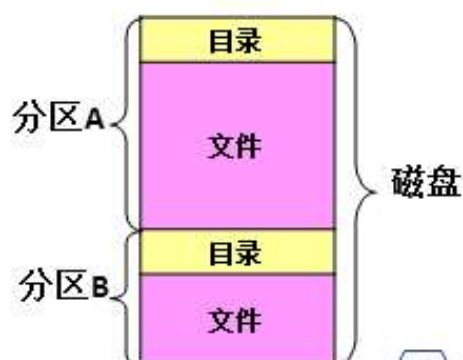
□如何组织用户的文件?



- 用户使用文件是“**按名存取**”的，这个“名”放在哪里？
- 对于以兆为单位的文件数量，用户怎么管理？如果不允许文件名重名，用户不得不想出数百万计的文件名。
- 文件系统给用户提供了组织管理文件的功能：

分区：将一个磁盘分若干分区，安装了文件系统的分区称为**卷**，每个磁盘至少一个卷

目录：存储一个分区中的文件信息。



29

□目录管理的功能

- 允许文件重名**——以方便用户按照自己的习惯来命名和使用文件。
- 按名存取**——将文件名转换成此文件信息在磁盘上的物理位置。
- 提高检索速度**——合理地组织目录结构，可以加快对目录的检索速度，从而提高对文件的存取速度。
- 文件共享**——允许多个用户共享同一文件，以节省存储空间，同时也方便用户。

□文件目录实现的关键

- 目录项与文件物理地址的关联方法

目录项的构成 2.2、3.3

文件的物理结构 3.3

- 在目录中“按名”搜索的效率

目录的结构 2.3
目录检索和实现 2.4

30

□ 目录文件

- 文件目录也用文件形式保存在磁盘上，被称为目录文件，包含该目录中的所有文件和子目录的目录项；
- 目录文件具有固定格式，由系统生成并管理，用户不能直接访问目录。
- 目录文件不能为空，至少包括两个目录项，即当前目录项“.”和父目录项“..”。

□ 目录项 2.2

- 文件控制块(File Control Block, FCB)
- 文件名&索引节点(index node, inode)编号

31

2.1 概述

2.2 FCB & inode

2.3 目录结构

2.4 目录的检索和实现

2.5 文件的共享

32

□ FCB (文件控制块)

文件 = FCB + 文件体

- 操作系统为每个文件建立的包含文件全部属性信息的唯一数据结构，其目的是为方便操作系统对文件的管理、控制和存取。

- 文件被创建时，系统为其建立一个FCB，用来记录文件的属性信息。



目录项



File Control Block

- file name
- permission
- date & time
- owner/group/ACL
- file size
- pointer to file data

详见3.3节！

- 每当存取文件时，先找到其FCB，将其读入内存，再从中找到文件首块物理位置或索引表，随后存取文件信息。

33

□ 目录文件大小及效率

- 文件目录存放在磁盘上，当文件很多时，目录文件要占用大量的磁盘块。

例6-1(a): 设盘块大小为512Bytes，一个FCB为64Bytes，其中文件名为14Bytes，若一个目录中有640个文件，计算顺序查找一个文件平均访盘次数？

一个盘块中可存放的FCB项数为： $512/64=8$

640个FCB需要存放在 $640/8=80$ 个盘块中

故平均访盘次数为 $(1+80)/2 = 40.5$ 次

在查找过程中，仅用到了文件名信息！



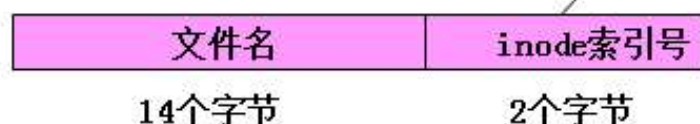
索引节点

34

□ Linux的目录项

- 把文件属性信息用一个称为索引节点的数据结构来描述，而在文件目录的每个目录项中，仅存有文件名和该文件的索引节点编号。

● Linux目录项



inode的编号，用于查找inode的存放位置

文件 = 目录项 + inode + 文件体



例6-1(b): 设盘块大小为512Bytes，文件名为14Bytes，索引节点号为2Bytes，若一个目录中有640个文件，计算顺序查找一个文件的起始地址，平均访盘次数。

一个盘块中可存放的目录项的项数为： $512/(14+2)=32$

640个文件的目录需要存放在 $640/32=20$ 个盘块中。

故平均访盘次数为 $(1+20)/2+1=11.5$

引入索引节点后，可大大减少启动磁盘的次数，有效提高检索文件的速度！

2.2 FCB & inode

```
caoxm@ubuntu:~$ sudo dumpe2fs -h /dev/sda5 |grep "Inode size"
dumpe2fs 1.45.5 (07-Jan-2020)
Inode size: 256
```

Linux的inode

文件类型

文件的ACL

文件的硬链接数量

文件主的用户ID

文件的所属组ID

文件字节数

文件最后访问的日期和时间 `atime`

文件最后修改的日期和时间 `mtime`

文件状态最后改变的日期和时间 `ctime`

含15个盘块地址的数组 详见3.3节!

inode
<ul style="list-style-type: none">• permission• data & time• Owner ID/group ID• file size•• an array of 15 disk-block address



Linux中inode占多少字节?

37

2.2 FCB & inode



一个block存放几个inode?

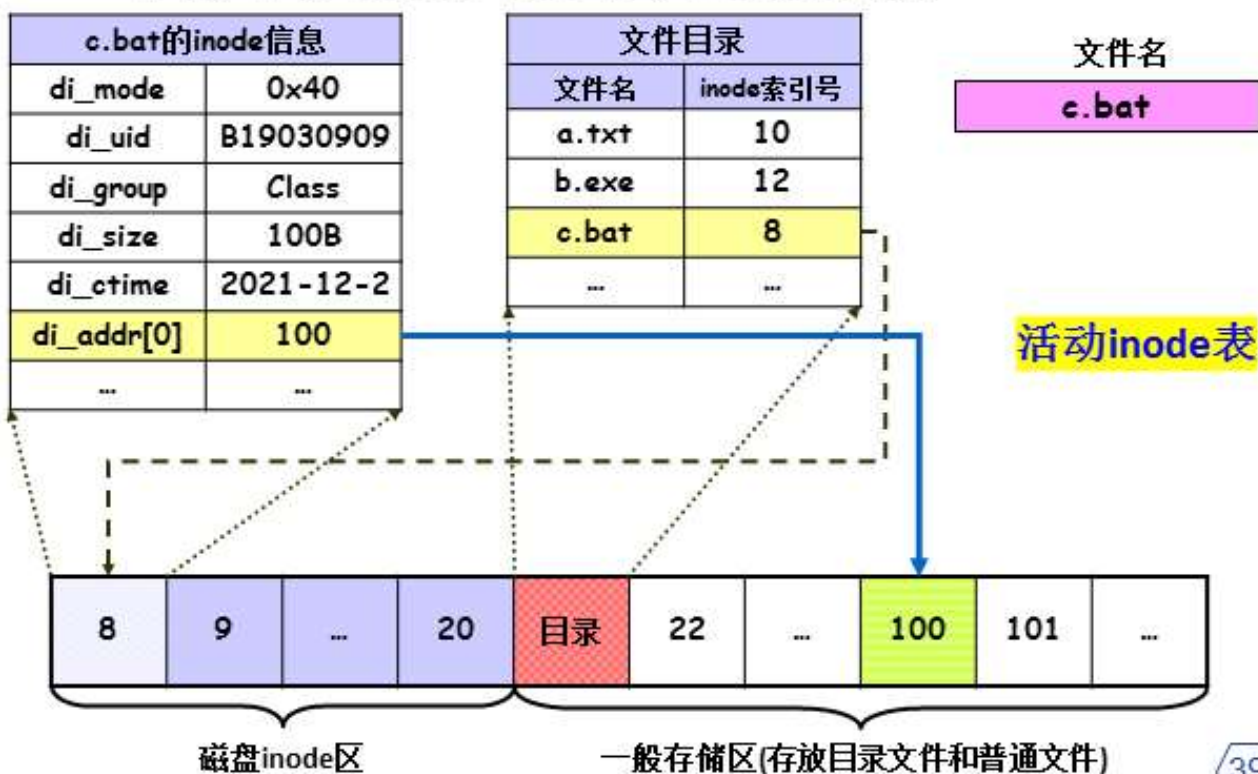
```
caoxm@ubuntu:~$ ls -li
total 64
1183759 drwxr-xr-x 2 caoxm caoxm 4096 Oct  7 21:38 Desktop
1183765 drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Documents
1183762 drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Downloads
1183766 drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Music
1578979 drwxrwxr-x 5 caoxm caoxm 4096 Nov 22 21:23 OS
1573424 -rwxrwxr-x 1 caoxm caoxm 16688 Oct 11 19:39 pl
1573425 -rw-rw-r-- 1 caoxm caoxm 218 Oct 11 19:39 pl.c
1183767 drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Pictures
1183764 drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Public
1590204 drwxr-xr-x 3 caoxm caoxm 4096 Sep  6 15:36 snap
1183763 drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Templates
1183768 drwxr-xr-x 2 caoxm caoxm 4096 Sep  6 15:03 Videos
```

```
caoxm@ubuntu:~$ stat pl.c
File: pl.c
Size: 218          Blocks: 8          IO Block: 4096   regular file
Device: 805h/2053d Inode: 1573425    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/   caoxm)   Gid: ( 1000/   caoxm)
Access: 2020-11-22 21:12:28.476044593 -0800
Modify: 2020-10-11 19:39:24.598120825 -0700
Change: 2020-11-22 21:29:27.764091514 -0800
Birth: -
```

Linux读取硬盘的基本单位是块(block)，连续八个sector组成一个block，一个block大小为4KB。

38

□ Linux下按名存取文件的过程（单级目录）



39

纲要

2. 目录 »

2.1 概述

2.2 FCB & inode

2.3 目录结构



- I. 单级目录结构
- II. 两级目录结构
- III. 树形目录结构

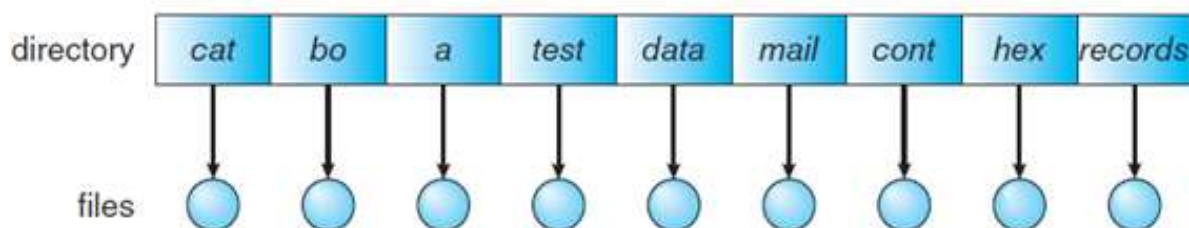
2.4 目录的检索和实现

2.5 文件的共享

40

□ 单级目录结构

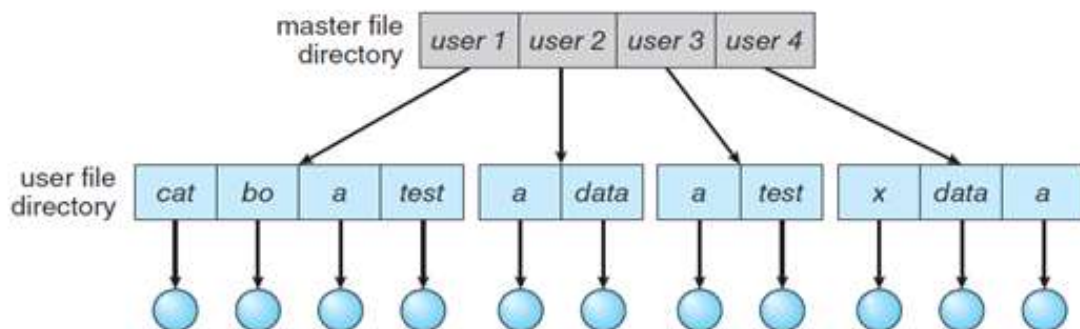
- 在整个文件系统中只建立一张目录表，其中每个目录项对应一个文件。
- 主要优点：实现简单。
- 缺点：不允许文件重名；文件检索速度慢。



41

□ 两级目录结构

- 解决不同用户之间引起的名称混乱问题，为每个用户创建一个独立的目录，即**用户文件目录(UFD)**，列出该用户的所有文件。
- 系统还有一级**主文件目录(MFD)**，用来管理所有用户文件目录。



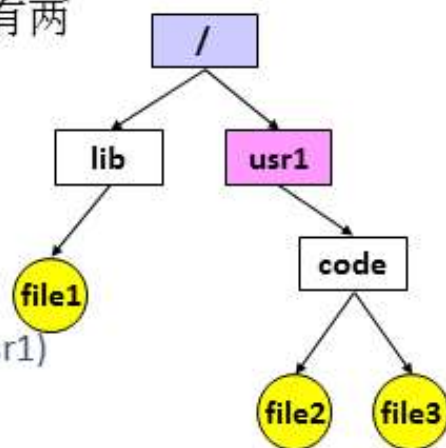
- 用户名和文件名构成了**路径名**，如” user1/ test”。user2要访问user1的文件必须指出全部路径名。

42

□ 树形目录结构

- 有一个根目录，而且除根目录外，其余每个目录或者文件都有唯一的一个上级目录。
- 每一级目录可以包含文件，也可以包含下一级目录，同一目录下的文件不可重名，不同目录下的文件可以重名。
- 系统内的每个文件都有**唯一的路径名**，有两种**路径搜索**方法：

- **绝对路径**：从根经过所有子目录到达指定文件的路径，如：
“/usr1/code/file2”
- **相对路径**：从当前目录开始定义的路径，如：“code/file2”(当前目录是usr1)
 - “.”：表示当前目录
 - “..”：表示父目录，如：“../lib/file1”



43

纲要

2. 目录 »

2.1 概述

2.2 FCB & inode

2.3 目录结构

2.4 目录的检索和实现

2.5 文件的共享

44

□ 基本步骤

- I. 系统根据用户提供的文件路径名，对目录文件进行查询，找出该文件的文件控制块或索引节点；
- II. 按照对应文件控制块或索引节点中所记录的文件物理盘块号，计算出文件在磁盘上的物理地址；
- III. 启动磁盘驱动程序，将所存取的文件块读入内存进行具体读写操作。

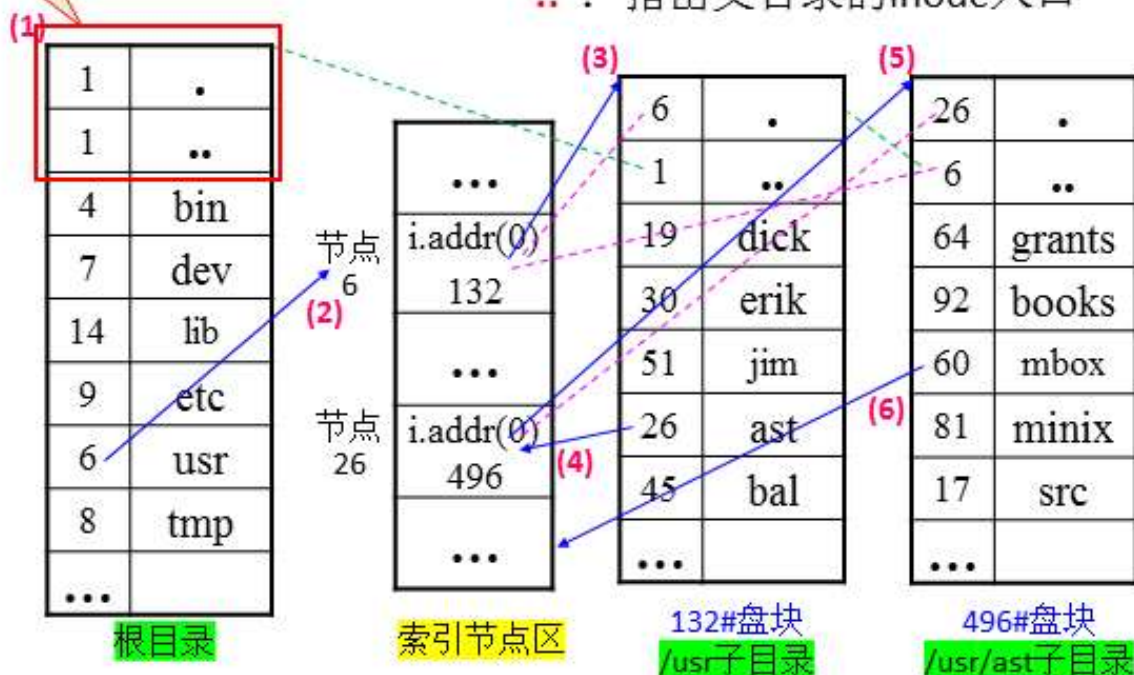
2.4 文件目录检索和实现

2. 目录 »

Linux下按名存取文件的过程（树形目录） /usr/ast/mbox

根目录

“.”：指出目录自身的inode入口
 “..”：指出父目录的inode入口



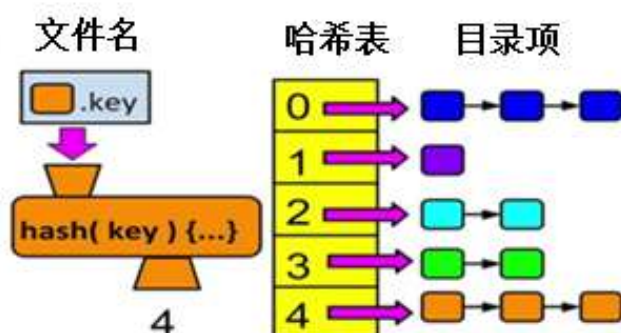
□ 目录实现

● 线性表

- ✓ 原理：使用存储文件名和数据块指针的线性表组织目录项
- ✓ 优点：编程简单
- ✓ 缺点：需要采用顺序方法查找特定项，执行费时

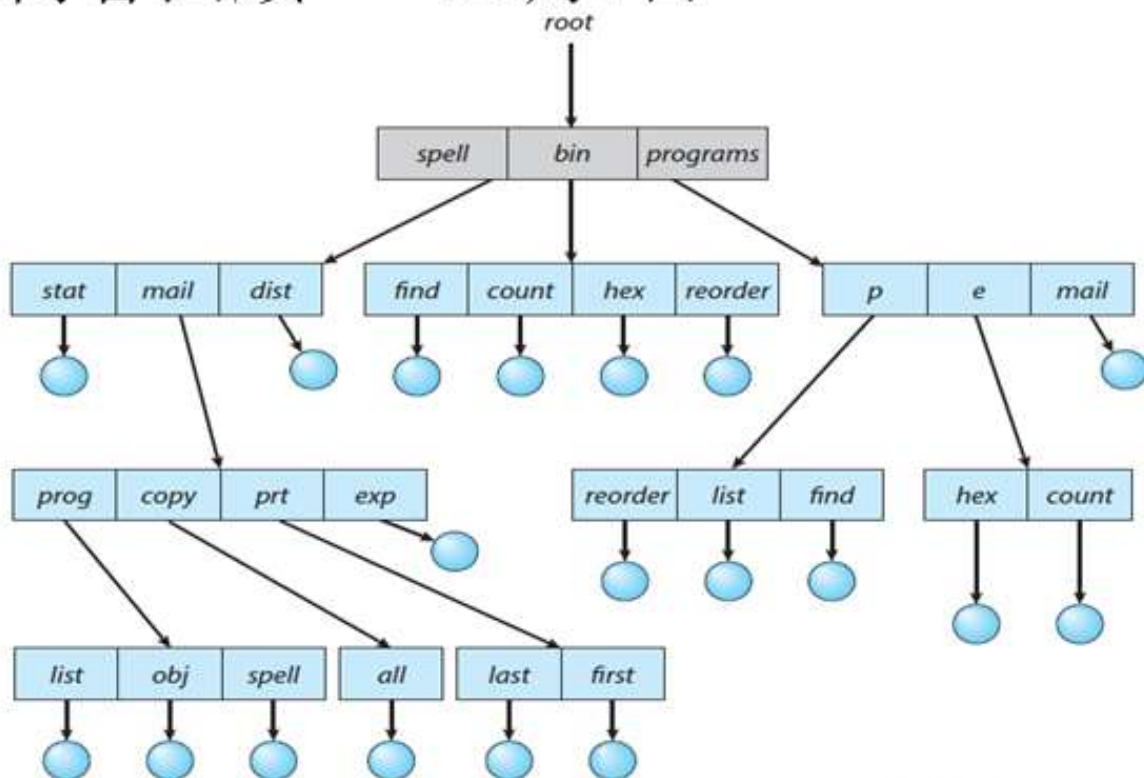
● 哈希表

- ✓ 原理：采用哈希数据结构实现检索
- ✓ 优点：查找非常迅速
- ✓ 缺点：需要复杂的管理



高速缓存

内容回顾——12月7日



缓存

/spell/mail/prog/list

2.1 概述

2.2 FCB & inode

2.3 目录结构

2.4 目录的检索和实现

2.5 文件的共享

49

2.5 文件的共享

- 概述 文件共享是指多个用户共同使用同一个文件。
在Linux中，可通过链接实现文件共享。

硬链接
Hard Link

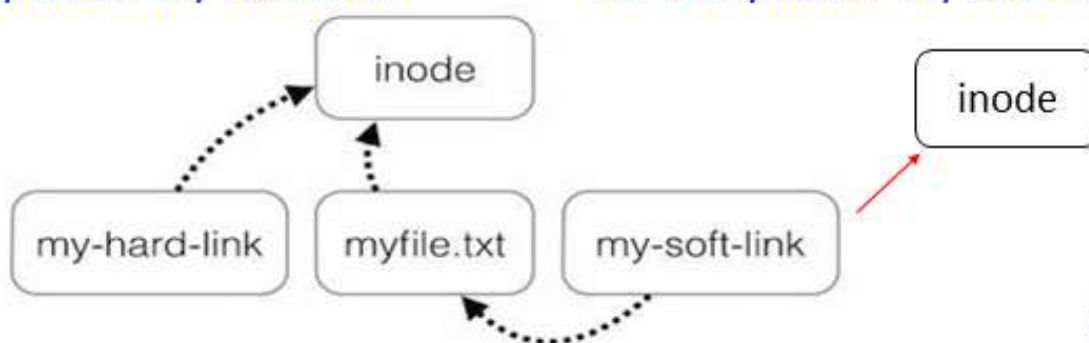
- 基于索引节点的链接

软链接
Soft Link

- 基于符号的链接
symbolic link

`ln myfile.txt my-hard-link`

`ln -s myfile.txt my-soft-link`



50

2.5 文件的共享

□ 硬链接(1/2)

- 实现原理

-- 多个文件名链接到同

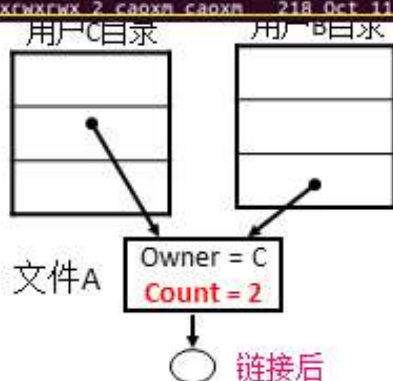
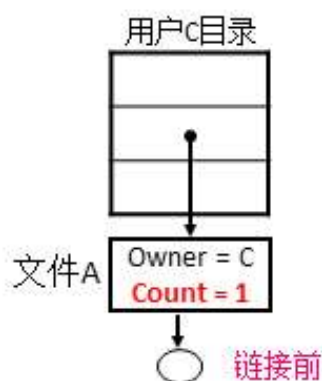
-- 索引节点的引用计数

若其减至0，则文件被删

```
caoxn@ubuntu:~$ stat pl.c
File: pl.c
Size: 218      Blocks: 8      IO Block: 4096   regular file
Device: 805h/2053d    Inode: 1573425    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/   caoxn)   Gid: ( 1000/   caoxn)
Access: 2020-11-22 21:12:28.476044593 -0800
Modify: 2020-10-11 19:39:24.598120825 -0700
Change: 2020-11-22 21:29:27.764091514 -0800
Birth: -
```

```
caoxn@ubuntu:~$ ln pl.c nyhardlink
caoxn@ubuntu:~$ stat pl.c
File: pl.c
Size: 218      Blocks: 8      IO Block: 4096   regular file
Device: 805h/2053d    Inode: 1573425    Links: 2
Access: (0777/-rwxrwxrwx)  Uid: ( 1000/   caoxn)   Gid: ( 1000/   caoxn)
Access: 2020-11-22 21:12:28.476044593 -0800
Modify: 2020-10-11 19:39:24.598120825 -0700
Change: 2020-11-30 17:20:32.663019300 -0800
```

```
caoxn@ubuntu:~$ ls -ld /home/caoxn/nyhardlink
lrwxrwxrwx 1 caoxn caoxn 4096 Sep  6 13:03 nyhardlink
lrwxrwxrwx 2 caoxn caoxn 4096 Nov 22 21:23 os
lrwxrwxrwx 1 caoxn caoxn 16688 Oct 11 19:39 pl
lrwxrwxrwx 2 caoxn caoxn 218 Oct 11 19:39 pl.c
```



51

2.5 文件的共享

2. 目录 »

□ 硬链接(2/2)

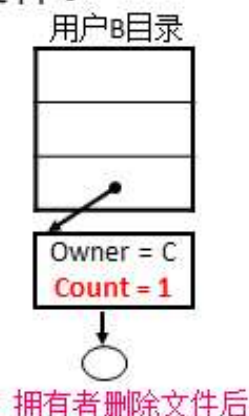
- 优点

实现简单，访问速度快，能够实现文件的异名共享。

- 缺点

-- 只允许在同一个文件系统范围内进行，不允许跨文件系统。

-- 文件拥有者不能删除被他人共享的文件。



52

□ 软链接(1/2)

- 实现原理 **LINK类型文件**
- 特殊类型的文件，其内容是另一个目录或文件路径的链接。
- 建立符号链接文件，并不影响原文件——它们是独立的文件。

```
caoxm@ubuntu:~$ ln -s pi.c mysoftlink
```

```
1575376 lrwxrwxrwx 1 caoxm caoxm 4 Nov 30 17:37 mysoftlink -> pi.c
1578979 drwxrwxr-x 5 caoxm caoxm 4096 Nov 22 21:23 OS
1573424 -rwxrwxr-x 1 caoxm caoxm 16688 Oct 11 19:39 pi
1573425 -rwxrwxrwx 1 caoxm caoxm 218 Oct 11 19:39 pi.c
```

```
caoxm@ubuntu:~$ readlink -f mysoftlink
/home/caoxm/pi.c
```

53

□ 软链接(2/2)

- 优点

文件所有者能删除被他人共享的文件；

给出合适的文件路径名，即可链接网络中任何地方计算机中的文件。

```
caoxm@ubuntu:~$ rm pi.c
```

```
1575376 lrwxrwxrwx 1 caoxm caoxm 4 Nov 30 17:37 mysoftlink -> pi.c
```

```
caoxm@ubuntu:~$ cat mysoftlink
cat: mysoftlink: No such file or directory
```

- 缺点

当其他用户读共享文件时，需要根据文件路径名的各个分量逐个查找，访问开销大。

54

单选题 1分

CQ2.1 以下叙述中错误的是（ ）。

- ☐ A 一个文件对应一个文件控制块
- ☐ B 目录的作用在于实现文件按名存取。
- ☒ C 目录文件可以为空
- ☐ D 文件目录用于将文件名转换成该文件信息在磁盘上的物理位置

单选题 1分

CQ2.2 文件控制块不包括（ ）。

- ☐ A 文件名
- ☐ B 文件访问权限信息
- ☐ C 文件物理位置信息
- ☒ D 磁盘坏块信息

单选题 1分

CQ2.3 Unix系统中关于文件目录和索引节点的概念叙述中正确的是（ ）。

- ☐ A 文件目录和索引节点相同
- ☐ B 文件目录和索引节点无关
- ☐ C 文件目录中有文件的控制信息
- ☒ D 索引节点中有文件的控制信息

单选题 1分

CQ2.4 在文件系统中，为了有效解决重名问题，通过（ ）来实现。

- ☐ A 重名翻译机构
- ☐ B 建立索引表
- ☒ C 树形目录结构
- ☐ D 建立指针

CQ2.5 设文件F1的当前引用计数值为1,先建立文件F1的符号链接(软链接)文件F2,再建立文件F1的硬链接文件F3,然后删除文件F1.此时,文件F2和文件F3的引用计数值分别是()。

- ☐ A 0,1
- ☒ B 1,1
- ☐ C 1,2
- ☐ D 2,1

内容纲要

Contents Page



1.文件

2.目录

3. 文件系统的实现



60

3.1 文件系统的组成

3.2 通用数据结构

★ 3.3 文件的物理结构

3.4 空闲空间管理

61

3.1 文件系统的组成

3. 文件系统的实现 »

□ 逻辑文件系统(logical file system)

管理文件系统的元数据信息，如文件控制块、目录结构、文件保护等。

□ 文件组织模块(file-organization module)

实现文件逻辑块号向物理块号的转换以及空闲空间管理器。

□ 基本文件系统(basic file system)

以磁盘地址标识要读写的物理块，向设备驱动程序发送读写命令，读取和写入磁盘物理块。

• 典型的文件系统：

- Windows:FAT16/FAT32/NTFS
- Unix:UFS UNIX File System
- Linux:Ext2/Ext3/Ext4 Extended File System



3.1 文件系统的组成

3.2 通用数据结构



- I. 磁盘管理结构
- II. 内存管理结构

3.3 文件的物理结构

3.4 空闲空间管理

63

3.2 通用数据结构

3. 文件系统的实现 »

磁盘管理结构(1/2)

□ 引导块 boot block

卷的第一块，包含引导操作系统所需的信息

□ 超级块 superblock

包括卷(或分区)的详细信息，分区的块的数量、块的大小、空闲块的数量和指针，空闲FCB数量和FCB指针等

□ 目录结构

用于组织文件

□ 文件控制块/inode

包含该文件的详细信息

64

磁盘管理结构(2/2)

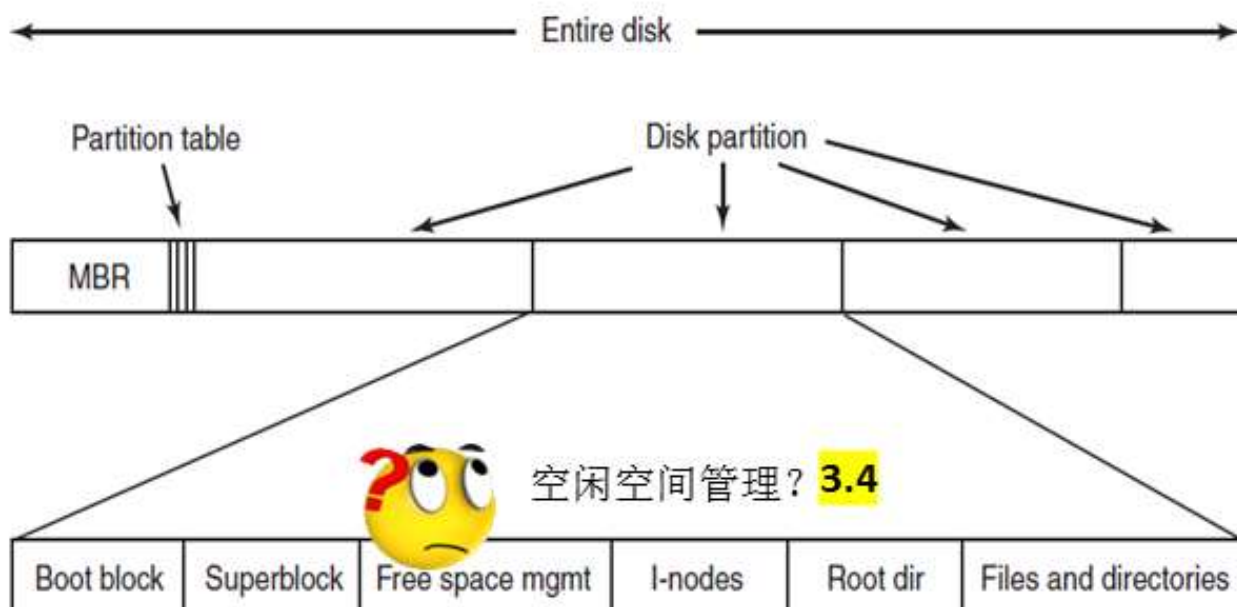


图 一个可能的文件系统布局

内存管理结构(1/4)

- 进程打开文件表 `per-process open-file table (descriptor table)`

进程PCB的一个数据域，记录进程所有打开文件的集合。该域每条记录指向系统打开文件表中的一个表项。

- 系统打开文件表 `system-wide open-file table (file table)`

记录系统所有打开文件的当前文件位置、引用计数(reference count)、指向活动inode表对应表项的指针。

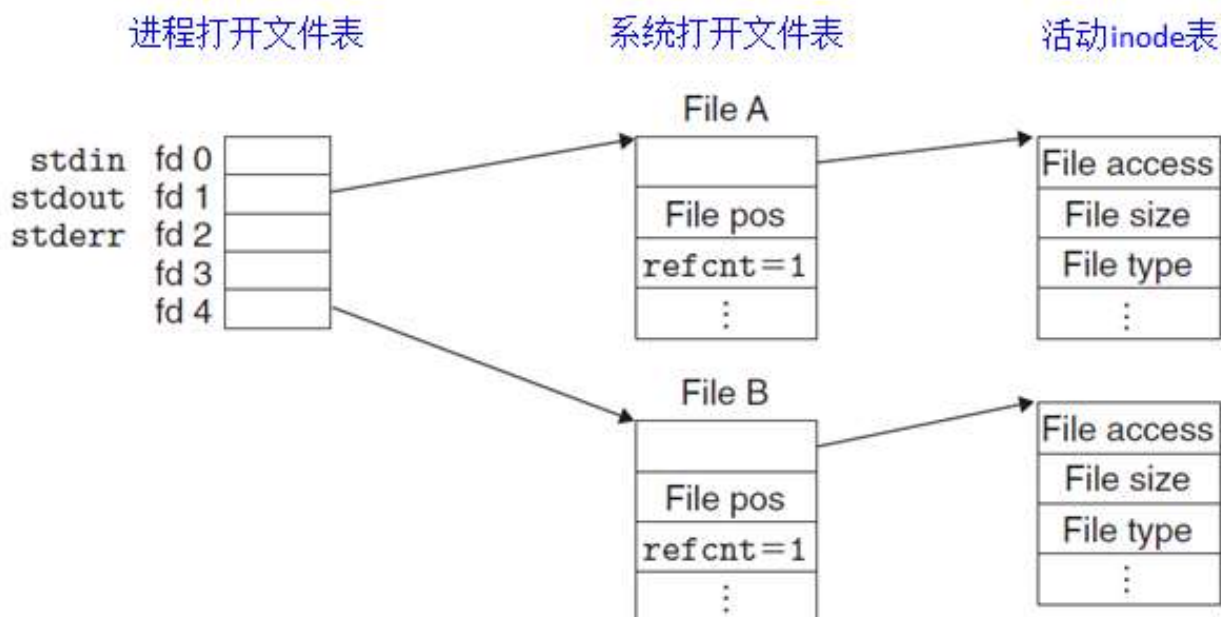
- 活动inode表 `v-node table`

存放最近打开文件的inode。

所有进程共享

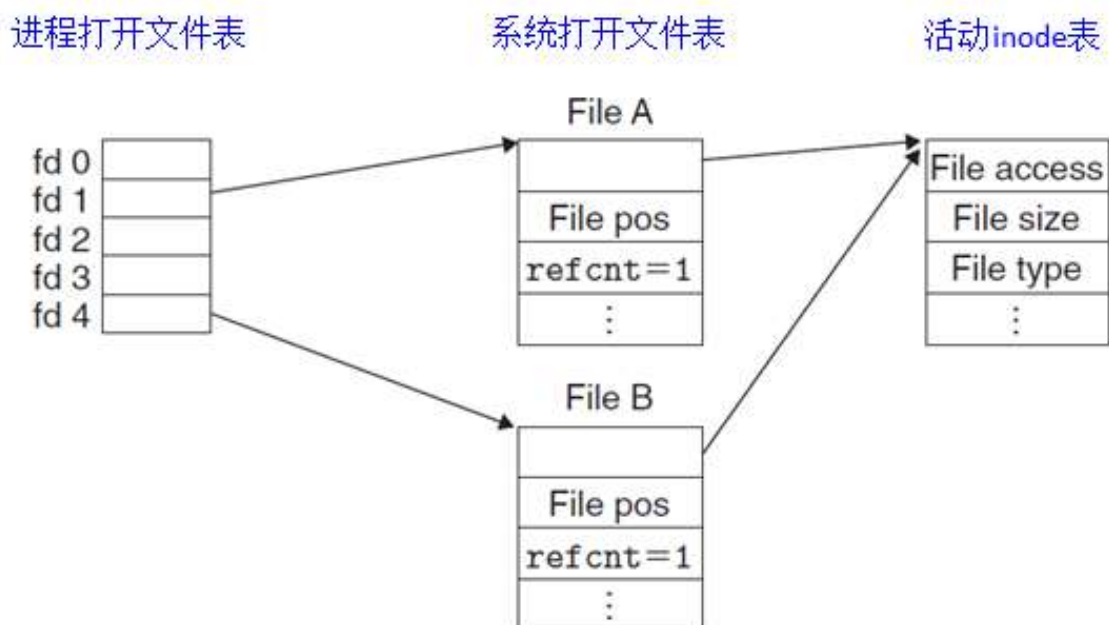
内存管理结构(2/4)

例：一个进程打开两个不同文件



内存管理结构(3/4)

例：一个进程两次打开同一个文件



3.2 通用数据结构

详见 课本6.4.1
文件类系统调用

3. 文件系统的实现»

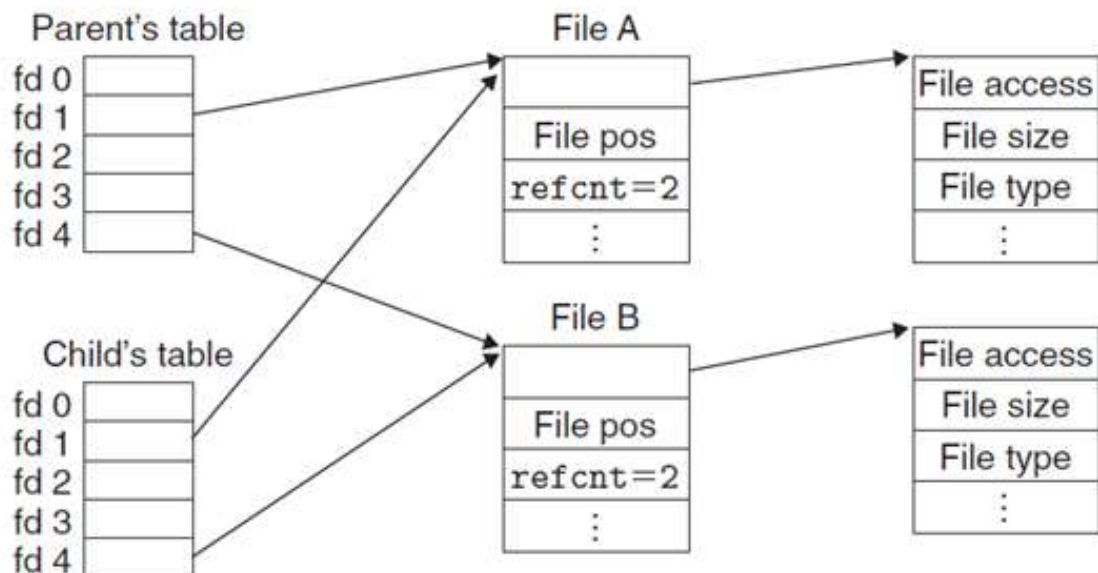
内存管理结构(4/4)

例：子进程继承父进程的打开文件 父进程调用fork后

进程打开文件表

系统打开文件表

活动inode表



纲要

3. 文件系统实现»

3.1 文件系统的组成

3.2 通用数据结构

3.3 文件的物理结构



- I. 概述
- II. 顺序文件
- III. 链接文件
- IV. 索引文件

3.4 空闲空间管理

概述

- 文件的物理结构是指逻辑文件在物理存储空间中的存放方法和组织关系，也称作文件的磁盘分配方案。
- 要找到文件的物理位置，文件目录起到了什么作用？
目录里直接或间接地记录了文件内容的磁盘分布
- 用户凭什么在目录中查找？
文件名
- 逻辑文件在磁盘空间如何组织？
 - 顺序文件 → 连续分配
 - 链接文件 → 链接分配
 - 索引文件 → 索引分配
 } 离散分配

71

顺序文件(1/2)

□ 分配原理

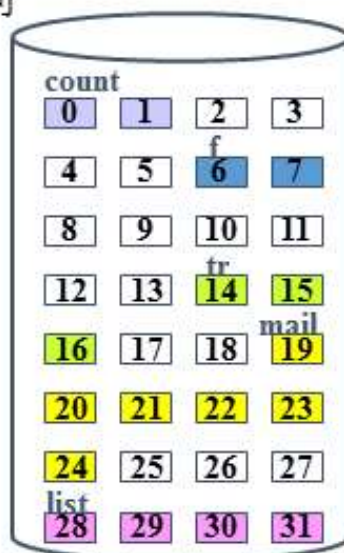
- 文件中逻辑上连续的信息存储在磁盘上也保持连续的结构
- 文件目录记录了文件第一块块号和块数
- 逻辑结构与物理结构相同

文件目录

文件名	始址	块数
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

连续分配

Contiguous Allocation



72

顺序文件(2/2)

□ 优点

- 简单，只需要记录文件的起始块号和长度
- 寻道时间最少
- 支持直接存取

□ 缺点

- 存在外部碎片问题：磁盘空间分割后形成的较小的无法存储文件的连续区，虽可使用紧凑技术，但开销太大。
- 必须事先知道文件的长度，且不适合文件的生长。

73

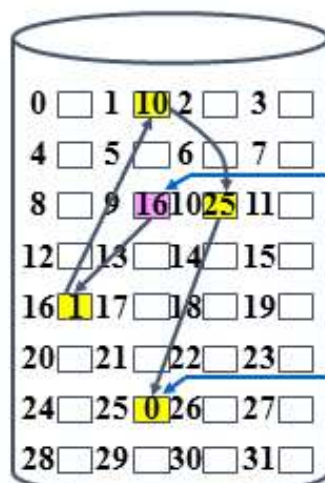
链接文件(1/4)

□ 分配原理

- 通过盘块上的指针实现同一文件多个离散物理盘块的链接；
- 文件目录记录了文件第一块和最后一块的块号；
- 各块之间通过指针连接，前一个盘块指向下一个盘块，当链接指针为0时表示本块是文件的最后一块。



Linked Allocation



文件目录

文件名	始址	结束
jeep	9	25

74

链接文件(2/4)

□ 优点

- 消除了外部碎片，外存空间利用率高
- 按需分配，且无需事先知道文件长度
- 支持文件动态增长，方便文件增删改

□ 缺点

- 指针占用盘块额外空间 $4B/512B=0.78\%$
- 只适合顺序访问，对随机存取极其低效
- 存在断链可能，可靠性较差。



为了提高检索速度和减少指针所占用的存储空间，可将几个盘块组成一个簇，以簇为单位进行盘块分配。

若8块/簇 则 $4B/4096B < 0.001\%$

会产生内部碎片，对小文件特别明显

75

链接文件(3/4)

□ 文件分配表 File Allocation Table: FAT

一个分区有一个FAT表，记录分区中所有簇的状态

- FAT是链接分配的变种，一种简单有效的磁盘空间分配方法，用于MS-DOS、Windows和OS/2等操作系统中。
- FAT表存储在分区的开始部分，分区中的每簇在FAT中有一个表项，该表可以通过簇号来索引。
- 表项的取值有三种可能：

表项的大小决定了簇的最大数量

 - 文件中下一个簇的簇号；
 - EOF (文件结束符)；
 - 0 (空闲簇)
- 文件目录项中存放含有文件首个簇的簇号。

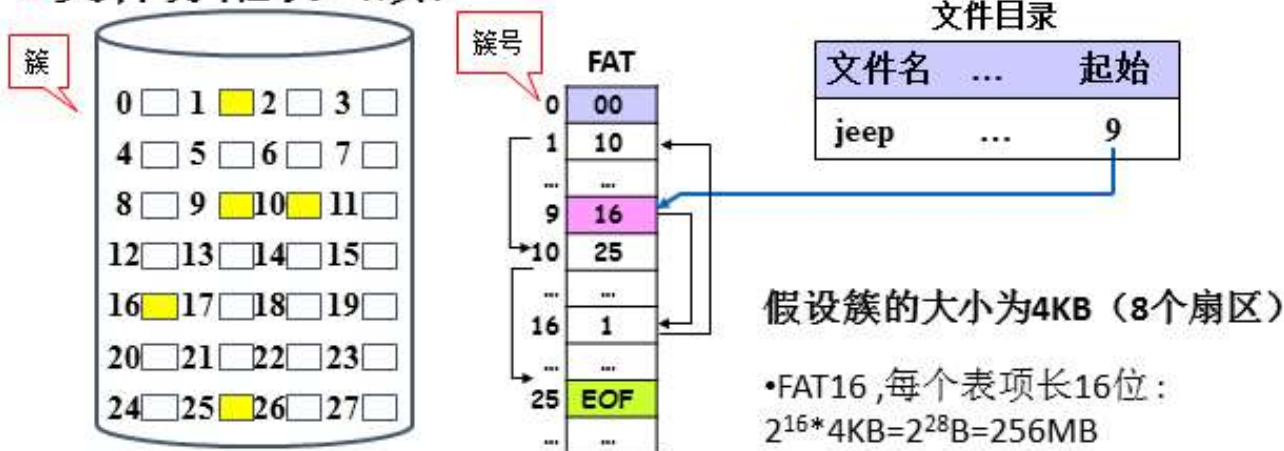
76

3.3 文件的物理结构

3. 文件系统的实现

链接文件(4/4)

文件分配表 (续)



文件分配表的局限性

- 不能支持高效地直接存取

对一个较大的文件进行直接存取，须首先在FAT表中顺序地查找许多簇号。

➡ 索引文件

77

3.3 文件的物理结构

3. 文件系统的实现

索引文件(1/7) Indexed Allocation

分配原理

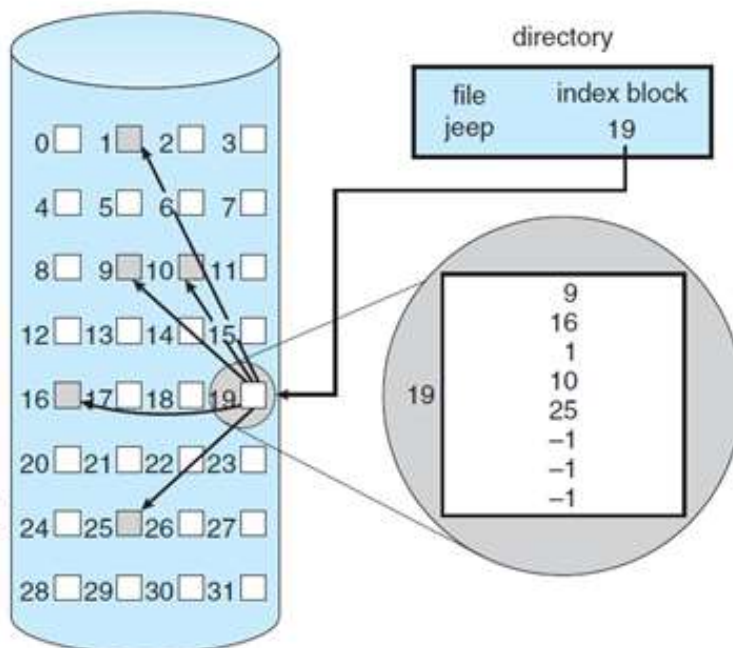
- 将文件占用的所有物理块号按逻辑顺序保存在一张索引表中，存有索引表的物理块称为索引块(index block)
- 目录项中指出索引块块号

优缺点

- 优点：随机访问
- 缺点：索引块占用磁盘空间



索引块如何组织、管理？



78

索引文件(4/7)

□ 常见题型-2 逻辑文件字节偏移量地址转换物理地址

- 步骤1. 将逻辑文件的字节偏移量转换为文件的逻辑块号和块内偏移

将逻辑文件的字节偏移量 n /盘块大小，商为文件的逻辑块号，余数是块内偏移。

- 步骤2. 将文件的逻辑块号转换为物理块号

使用多重索引结构，在索引节点中根据逻辑块号通过直接索引或间接索引找到对应物理块号。

81

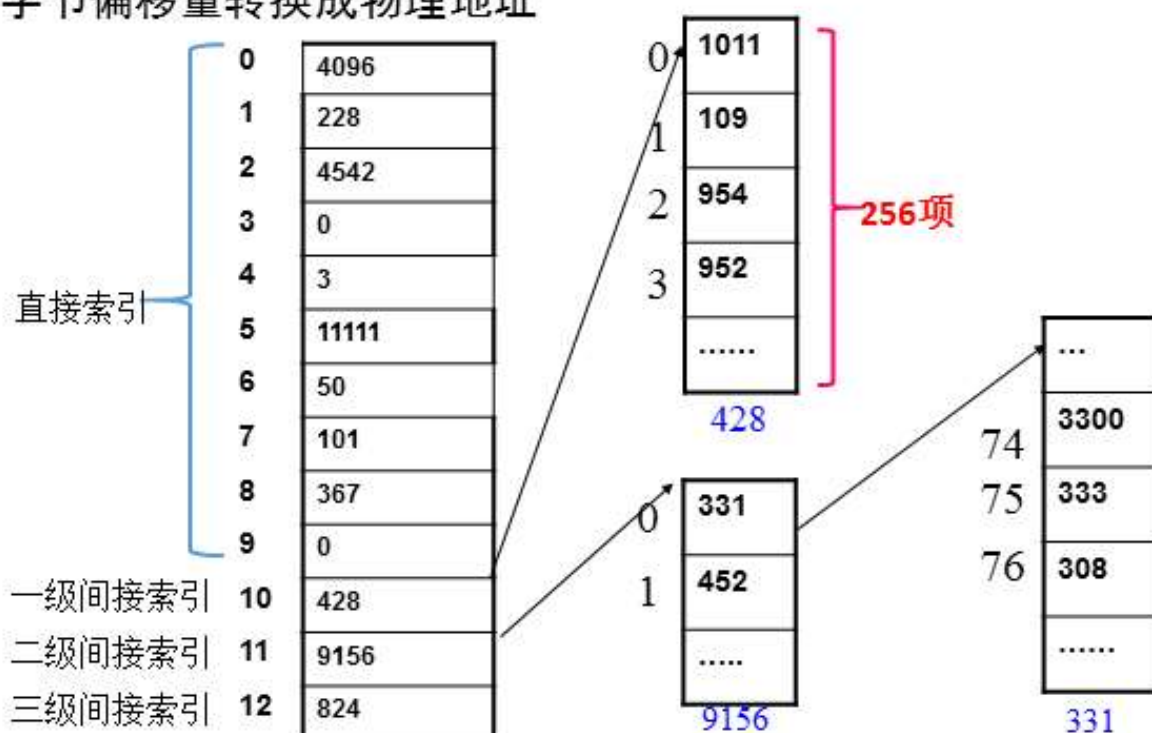
3.3 文件的物理结构

索引文件(5/7)

例6-3：Unix多级索引下文件的字节偏移量转换成物理地址

假定每个盘块1KB,每个盘块号占4B，如何将下列文件的字节偏移量转换为物理地址？

1. 9000 2. 14000



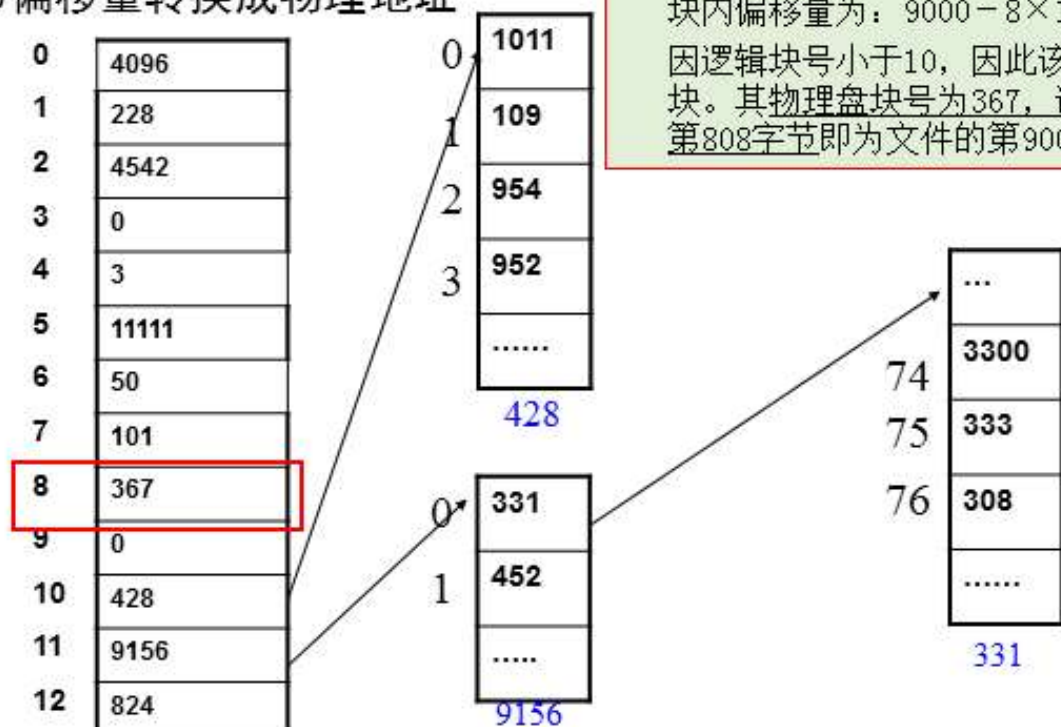
82

3.3 文件的物理结构

3. 文件系统的实现

索引文件(6/7)

例6-3：Unix多级索引下文件的字节偏移量转换成物理地址



1. 字节偏移量为9000

逻辑块号为： $9000/1024=8$

块内偏移量为： $9000-8\times1024=808$

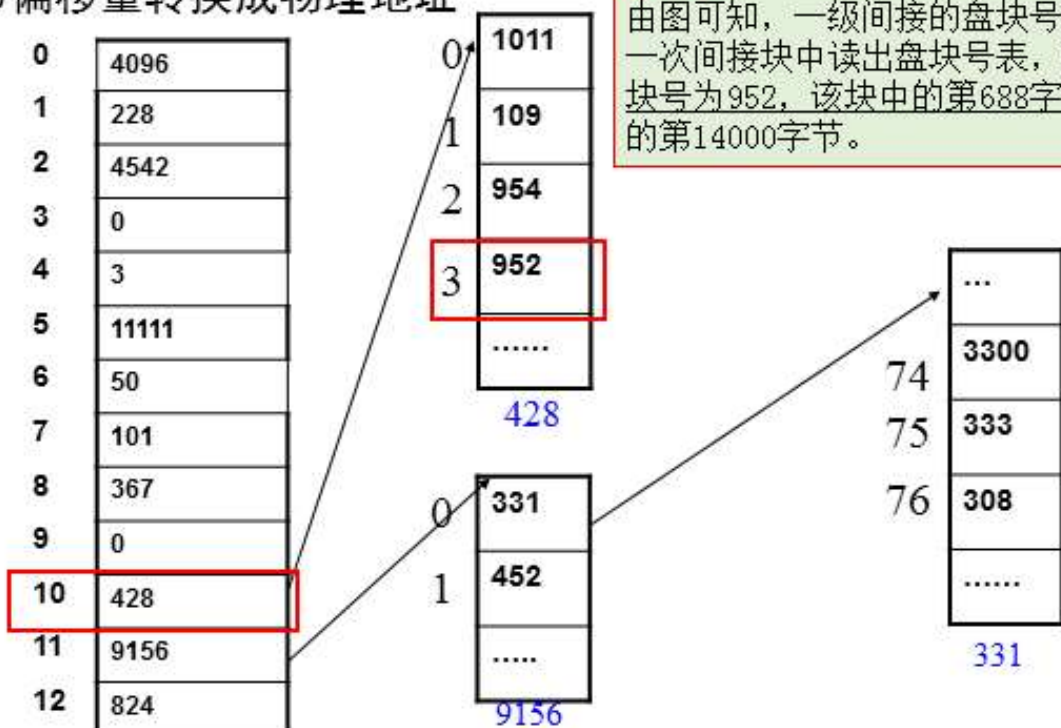
因逻辑块号小于10，因此该块为直接块。其物理盘块号为367，该块中的第808字节即为文件的第9000字节。

83

3.3 文件的物理结构

索引文件(7/7)

例6-3：Unix多级索引下文件的字节偏移量转换成物理地址



2 字节偏移量为14000

逻辑块号为： $14000/1024=13$

块内偏移量为： $14000-13\times1024=688$

因逻辑块号 $10<13<266$ ，因此该块为一级间接块。

由图可知，一级间接的盘块号为428，从一次间接块中读出盘块号表，查得其物理块号为952，该块中的第688字节即为文件的第14000字节。

84

例6-3：Un 用到2级间接

字节偏移量
为350000

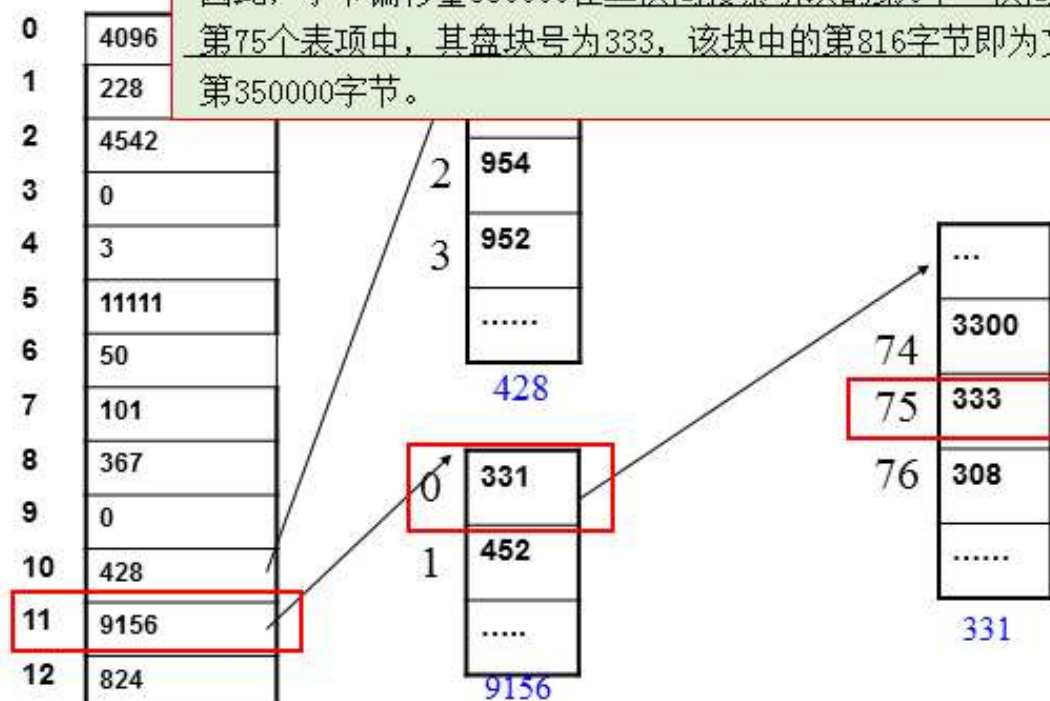
逻辑块号为： $350000/1024=341$

块内偏移量为： $350000-341\times1024=816$

因逻辑块号 $266<341<65802$ ，因此该块为二级间接索引块。

由图可知，二次间接索引块的盘块号为9156。由于一个一次间接索引块中可容纳256个块号， $341-10-256=75<256$

因此，字节偏移量350000在二次间接索引块的第0个一次间接块的第75个表项中，其盘块号为333，该块中的第816字节即为文件第350000字节。



85

3.3 文件的物理结构

3. 文件系统的实现 »

总结

□ 存储设备、文件物理结构与存取方法之间的关系

存储设备	磁 盘			磁 带
物理结构	顺序文件	链接文件	索引文件	顺序文件
存取方法	直接、顺序	顺序	直接、顺序	顺序

86

纲要

3. 文件系统的实现 »

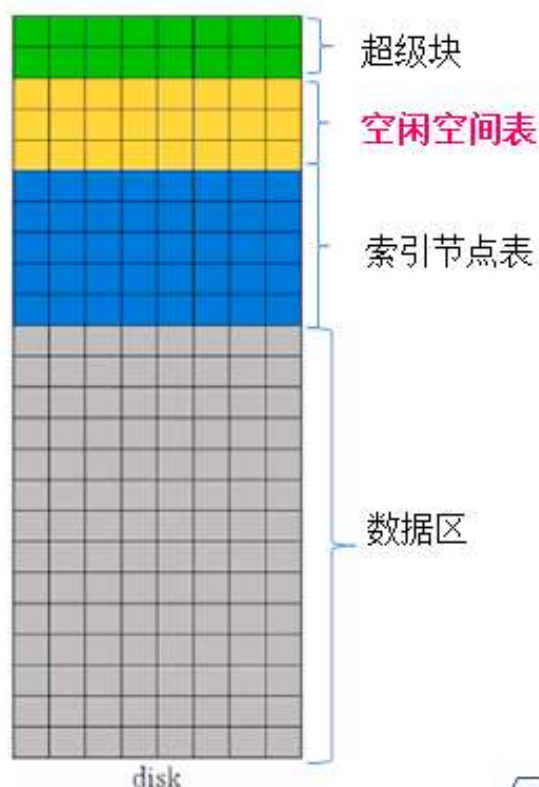
3.1 文件系统的组成

3.2 通用数据结构

3.3 文件的物理结构

3.4 空闲空间管理

- I. 位示图
- II. 空闲块链表



87

3.4 空闲空间管理

3. 文件系统的实现 »

位示图(1/2) bit map

□ 原理

- 每个磁盘块用一个位来表示，如果磁盘块空闲，位为0；如果已分配，位为1

0011110011111100011000000111000000...

□ 优点

- 非常高效

88

位示图(2/2)

设计算机字长为 n 位，位示图中字的编号从0开始

□ 盘块的分配

- 1) 逐字顺序扫描位示图，找出第一个不为1的字，从中找出第一个值为0的二进制位；
- 2) 求对应盘块号： $b = n \times (\text{值为1的字数}) + \text{该位的字内偏移}$ ；
- 3) 按盘块号分配盘块，同时修改位示图中相应位为1。

□ 盘块的回收

- 1) 将回收盘块的盘块号 b 转换为位示图中的字号 i 和字内偏移 j ：

$$i = b \text{ DIV } n;$$

$$j = b \text{ MOD } n;$$

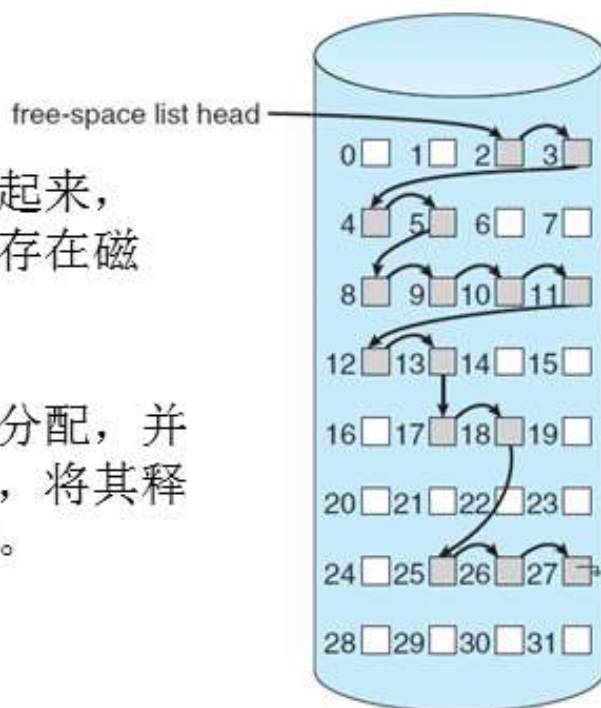
- 2) 按盘块号回收盘块，同时修改位示图中相应位为0。

89

空闲盘块链表

□ 原理

- 将所有空闲磁盘块用链表链接起来，将指向第一个空闲块的指针保存在磁盘的特殊位置。
- 新建文件时，从头部取空闲块分配，并修改链表头指针；删除文件时，将其释放的空闲块加入到链表的尾部。



90

总结

- 文件的概念及属性
- 文件的逻辑结构
 - 流式和记录式
- 文件的存取方式
 - 顺序、直接和索引
- 目录实现了“按名存取”，目录项可以是包含文件的FCB或者仅包含文件inode索引号。
- 文件的保护和共享
- 文件系统结构和管理信息
- 文件的物理结构
 - 顺序、链接和索引



91

单选题 1分

CQ3.1 文件的分配方式是由（ ）确定的。

- ☐ A 应用程序
- ☐ B 内存容量
- ☐ C 外存容量
- ☒ D 操作系统

单选题 1分

CQ3.2 文件信息的逻辑块号到磁盘物理块号的变换是由（ ）决定的。

- ☐ A 逻辑结构
- ☐ B 页表
- ☒ C 物理结构
- ☐ D 重定位寄存器

单选题 1分

CQ3.3 在有直接存取需求和文件长度动态增长的情况下，宜选择以下（ ）方式。

- ☒ A 索引分配
- ☐ B 连续分配
- ☐ C 链接分配
- ☐ D 都不对

单选题 1分

CQ3.4 以下关于索引文件的论述中，正确的是（ ）。

- ☐ A 在索引文件中，索引项的每个表项必须含有相应记录的关键字和存放该记录的物理地址
- ☒ B 对顺序文件进行检索时，首先从FCB中读出文件的第一个盘块号；而对索引文件进行检索时，应从FCB中读出文件索引表始址
- ☐ C 对于一个具有三级索引表的文件，存取一个记录在最坏情况下需要访问三次磁盘
- ☐ D 在文件较大时，无论是顺序存取还是直接存取，通常都是索引文件方式为最快

单选题 1分

CQ3.5 文件系统采用两级索引分配方式。如果每个磁盘块的大小是1KB，每个盘块号占4个字节，则该系统中，单个文件的最大长度是（ ）。

- ☒ A 64MB
- ☐ B 128MB
- ☐ C 32MB
- ☐ D 都不对

多选题 1分

CQ3.6 打开文件操作主要是（ ）

- ☐ A 把整个文件从磁盘拷贝到内存
- ☒ B 把文件目录项(FCB)从磁盘拷贝到内存
- ☐ C 把整个文件和文件目录项从磁盘拷贝到内存
- ☐ D 把磁盘文件系统的控制信息从辅存读到内存

单选题 1分

CQ3.7 文件系统中使用位示图实现（ ）。

- ☐ A 文件目录的查找
- ☒ B 磁盘空间的管理
- ☐ C 内存空间的共享
- ☐ D 文件的保护和保密

单选题 1分

CQ3.8 位示图可用于磁盘空间的管理。设某系统磁盘共有500块，块号是0~499，第0行的第0位表示第0块，第0行的第1位表示第1块，以此类推。若用位示图管理这500块的磁盘空间，当字长为32位时，第*i*个字第*j*位对应的块号是（ ）。

- ☒ A $32i+j$
- ☐ B $32i+j-1$
- ☐ C $32(i-1)+j-1$
- ☐ D $32(i-1)+j$

课堂练习

- 设某系统中文件寻址可用多次间接方式与直接寻址方式，每个磁盘块有512个字节，每个物理块地址占4个字节。

设直接寻址10块，间接寻址最多可有3次，试问：

1. 该系统的文件大小最大是多少？
2. 文件字节偏移量分别为6000B、6000KB的数据如何寻址？

参考答案

每个索引块最多索引
 $512/4=128$ 个地址

- 直接寻址空间 $i.addr(0) \sim i.addr(9)$: $10^4 \times 0.5K = 5KB$
共指向10个物理盘块 逻辑块号范围: $[0, 9]$
- 一级间接寻址空间 $i.addr(10) \sim i.addr(19)$: $10^4 \times 4K = 40KB$
共指向10个索引块 逻辑块号范围: $[10, 19]$
- 二级间接寻址空间 $i.addr(20) \sim i.addr(29)$: $10^4 \times 16K = 160KB$
共指向10个索引块 逻辑块号范围: $[20, 29]$
- 三级间接寻址空间 $i.addr(30) \sim i.addr(39)$: $10^4 \times 64K = 640KB$
共指向10个索引块 逻辑块号范围: $[30, 39]$
- 四级间接寻址空间 $i.addr(40) \sim i.addr(49)$: $10^4 \times 256K = 2.56MB$
共指向10个索引块 逻辑块号范围: $[40, 49]$
- 五级间接寻址空间 $i.addr(50) \sim i.addr(59)$: $10^4 \times 1GB = 10GB$
共指向10个索引块 逻辑块号范围: $[50, 59]$

文件大小最大为: $8M + 64M + 5K$

$L1 = \text{INT}(6000, 512) = 11$, $B1 = \text{MOD}(6000, 512) = 368$, 其逻辑块号为11, 通过一级间接索引 $addr[10]$ 可找到物理块号

$L2 = \text{INT}(6000K, 0.5K) = 12000$, $B2 = \text{MOD}(6000K, 0.5K) = 0$, 其逻辑块号为12000, 通过二级间接索引 $addr[11]$ 可找到物理块号

Thank You

Have A Nice Day

南京邮电大学计算机学院、
软件学院、网络空间安全学院