

操作系统 Operating System

北京理工大学计算机学院
马 锐
Email: mary@bit.edu.cn

版权声明

- 本内容版权归北京理工大学计算机学院操作系统课程组马锐所有
- 使用者可以将全部或部分本内容免费用于非商业用途
- 使用者在使用全部或部分本内容时请注明来源
 - 内容来自：北京理工大学计算机学院+马锐+材料名字
- 对于不准守此声明或其他违法使用本内容者，将依法保留追究权

2

第5章 文件系统

- 5.1 文件和文件系统
- 5.2 文件目录项和文件目录结构
- 5.3 文件操作命令
- 5.4 文件逻辑结构和文件访问方法
- 5.5 文件物理结构和文件存储介质
- 5.6 文件存储器存储空间管理
- 5.7 文件共享与保护
- 5.8 文件系统组织结构
- 5.9 存储区映射文件

3

5.1 文件和文件系统(1)

➤ 文件的基本概念

- 文件是具有符号名的相关信息的集合
 - ✦ 源程序
 - ✦ 一批数据
 - ✦ 各种语言的编译程序、各种编辑程序
 - ✦ 银行帐目，公司记录等

➤ 文件的基本特征

- 文件的内容为一组信息
 - ✦ 有结构
 - ✦ 无结构

4

5.1 文件和文件系统(2)

- 文件具有保存性
 - ✦ 长期保存，多次使用
- 文件可以实现按名存取
 - ✦ 每个文件都有唯一的标识符，可以通过该标识符存取文件中的信息，无须了解文件在存储介质上的具体位置

➤ 文件的分类

- 按用途
 - ✦ 系统文件
 - ✦ 库文件

5

5.1 文件和文件系统(3)

- ✦ 应用程序文件
- ✦ 用户文件
- 按保护方式
 - ✦ 只读文件
 - ✦ 读写文件
 - ✦ 无保护的文件
- 按信息流向
 - ✦ 输入文件
 - ✦ 输出文件
 - ✦ 输入/输出文件

6

5.1 文件和文件系统(4)

- 按信息保存期限
 - ◆临时文件
 - ◆永久文件
 - ◆档案文件
- 按文件所在设备
 - ◆硬盘文件
 - ◆磁带文件
 - ◆光盘文件

7

5.1 文件和文件系统(5)

- UNIX系统中，按组织和处理方式划分
 - ◆普通文件
 - ◆系统文件、应用程序文件、库文件，用户的各种文件
 - ◆目录文件
 - ◆特殊文件
 - ◆由系统中所有输入、输出和输入/输出型设备组成
 - ◆输入、输出设备：字符型特殊文件
 - ◆输入/输出型设备：字符块特殊文件

8

5.1 文件和文件系统(6)

- 文件属性
 - 文件名(扩展名)
 - 文件标识符
 - 文件类型
 - 文件大小
 - 文件的建立时间
 - 文件的物理位置
 - 文件的保护信息

9

5.1 文件和文件系统 (7)

file type	usual extension	function
executable	exe, com, bin or none	read to run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

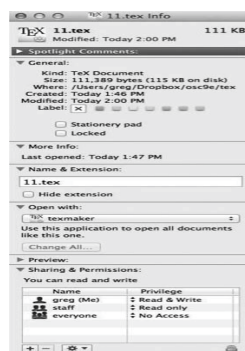
10

5.1 文件和文件系统 (8)

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

11

5.1 文件和文件系统 (9)



A file info window on macOS

12

5.1 文件和文件系统(10)

➤ 文件系统

- 为了方便用户使用软件资源，由OS提供的管理文件的软件机构
- 文件系统既包括OS中用于文件管理的程序，也包括运行这些程序所需的各种数据结构
- 一个理想的文件系统应具备的功能
 - 管理磁盘、磁带等组成的文件存储器
 - 实现用户的按名存取，负责名字空间到存储空间的映射

13

5.1 文件和文件系统(11)

- 具有灵活多样的文件结构和存取方法
- 向用户提供一套使用方便、简单的操作命令
- 保证文件信息的安全性
- 便于文件的共享

14

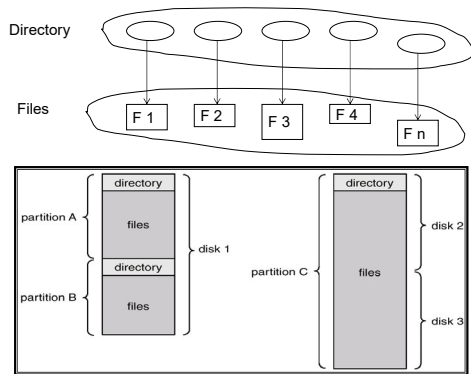
5.2 文件目录项和文件目录结构(1)

➤ 文件目录

- 文件的说明信息与控制信息
- 文件目录表
 - 一张记录所有文件的名字及其存放物理地址的映射表
 - 每个文件占据文件目录表中的一个表项
- 文件目录项
 - 文件控制块(FCB)：文件存在的唯一标志
 - 存放文件的全部控制信息
 - 存放文件名和文件存储位置

15

5.2 文件目录项和文件目录结构(2)



16

5.2 文件目录项和文件目录结构(3)

➤ 文件控制块(File Control Block, FCB)

- 基本信息
 - ◆ 文件名：唯一性
 - ◆ 文件类型
 - ◆ 文件物理位置
 - 设备名
 - 文件在外存上的起始盘块号
 - 文件长度：文件占用的盘块数和字节数
- 文件逻辑结构

17

5.2 文件目录项和文件目录结构(4)

- 文件物理结构
- 存取控制信息
 - ◆ 文件主的存取权限
 - ◆ 其他用户的存取权限
- 使用信息
 - ◆ 文件的建立时间
 - ◆ 上一次修改时间
 - ◆ 当前使用信息

file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks

18

5.2 文件目录项和文件目录结构(5)

➤ 文件目录的实现

● 线性表

- 存储文件名和数据块指针
- 编程较简单，但运行较费时
- 缺点
 - 查找文件时需要进行线性搜索
- 改进
 - B+树/红黑树

19

5.2 文件目录项和文件目录结构(6)

● 目录查询技术——线性搜索

根目录	结点 6 是 /usr 的目录	132 号盘块是 /usr 的目录	结点 26 是 /usr/ast 的目录	496 号盘块是 /usr/ast 的目录																																																												
<table><tr><td>1</td><td>.</td></tr><tr><td>1</td><td>..</td></tr><tr><td>4</td><td>bin</td></tr><tr><td>7</td><td>dev</td></tr><tr><td>14</td><td>lib</td></tr><tr><td>9</td><td>etc</td></tr><tr><td>6</td><td>usr</td></tr><tr><td>8</td><td>tmp</td></tr></table>	1	.	1	..	4	bin	7	dev	14	lib	9	etc	6	usr	8	tmp	<table><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td>132</td></tr><tr><td></td><td></td></tr></table>						132			<table><tr><td>6</td><td>.</td></tr><tr><td>1</td><td>..</td></tr><tr><td>19</td><td>dick</td></tr><tr><td>30</td><td>erik</td></tr><tr><td>51</td><td>jim</td></tr><tr><td>26</td><td>ast</td></tr><tr><td>45</td><td>bal</td></tr></table>	6	.	1	..	19	dick	30	erik	51	jim	26	ast	45	bal	<table><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td>496</td></tr><tr><td></td><td></td></tr></table>						496			<table><tr><td>26</td><td>.</td></tr><tr><td>6</td><td>..</td></tr><tr><td>64</td><td>grants</td></tr><tr><td>92</td><td>books</td></tr><tr><td>60</td><td>mbox</td></tr><tr><td>81</td><td>minik</td></tr><tr><td>17</td><td>src</td></tr></table>	26	.	6	..	64	grants	92	books	60	mbox	81	minik	17	src
1	.																																																															
1	..																																																															
4	bin																																																															
7	dev																																																															
14	lib																																																															
9	etc																																																															
6	usr																																																															
8	tmp																																																															
	132																																																															
6	.																																																															
1	..																																																															
19	dick																																																															
30	erik																																																															
51	jim																																																															
26	ast																																																															
45	bal																																																															
	496																																																															
26	.																																																															
6	..																																																															
64	grants																																																															
92	books																																																															
60	mbox																																																															
81	minik																																																															
17	src																																																															
在结点 6 中查找 usr 字段																																																																

在结点 6 中查找
usr 字段

查找/usr/ast/mbox的步骤

20

5.2 文件目录项和文件目录结构(7)

● 哈希表

- 使用线性表存储目录，并使用哈希数据结构，并根据文件名进行哈希
- 优点：减少了目录搜索时间
- 缺点
 - 冲突问题
 - 哈希表大小固定及哈希函数对大小的依赖性
- 改进
 - Chained-overflow 哈希表

21

5.2 文件目录项和文件目录结构(8)

➤ Linux文件目录

- 把通常的文件目录项分成简单目录项和索引节点两部分
 - 简单目录项包含了文件名和索引节点号等，可以提高文件目录的检索速度
 - 提高了文件目录的检索速度和文件系统的使用效率
 - 文件的管理控制信息存放在索引节点中
 - 系统只保留一个索引节点，就可实现多条路径共享文件，减少信息冗余

22

5.2 文件目录项和文件目录结构(9)

● 目录项结构

```
struct ext2_dir_entry_2{
    _le32 inode;    //索引节点号
    _le16 rec_len;  //目录项长度
    _u8  name_len;  //实际文件名长度
    _u8  file_type; //文件类型
    char name[255]; //文件名是4B整数倍
                  //变长数组
}
```

23

5.2 文件目录项和文件目录结构(10)

● 文件类型

- 0, 不可知
- 1, 普通文件
- 2, 目录文件
- 3, 字符设备文件
- 4, 块设备文件
- 5, 有名管道
- 6, 套接字文件
- 7, 符号链接

24

5.2 文件目录项和文件目录结构(11)

- 文件名
 - name, 存放文件名, 最多255个字符
 - name_len, 文件名实际长度
 - 文件名长度不同, 但必须是4字节的整数倍
 - 不足时, 以字符'\0'填充
- 目录项长度
 - 至少占12个字节

25

5.2 文件目录项和文件目录结构(12)

- 目录文件示例
- 删除目录项

$$28=12+16$$

	inode	rec_len	name_len	file_type	name
0	21	12	1	2	.
12	22	12	2	2	.
24	53	16	5	2	h o m e
40	67	28	3	2	u s r
52	0	16	7	1	o l d f i l e
68	34	12	4	2	s b i n

26

5.2 文件目录项和文件目录结构(13)

- 目录管理的要求
 - 实现“按名存取”
 - 提高对目录的检索速度
 - 文件共享
 - 允许文件重名
- 文件目录结构
 - 一级目录结构
 - 二级目录结构
 - 多级目录结构

27

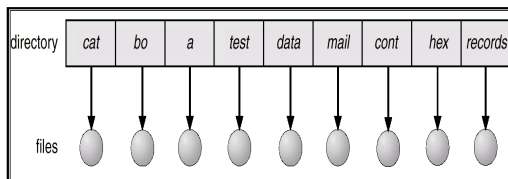
5.2 文件目录项和文件目录结构(14)

➤ 一级目录结构

- 为文件存储器上保存的全部文件建立一张目录表，每个文件在表中占有一项
- 创建文件/删除文件
- 优点
 - 目录结构简单
- 缺点
 - 不允许文件重名，否则将出现二义性
 - 文件较多时，查找目录耗时太多
 - 限制了对文件的共享

28

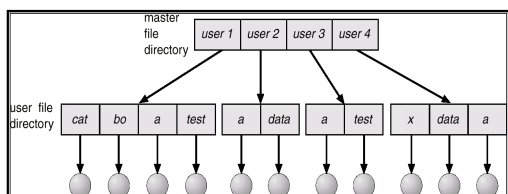
5.2 文件目录项和文件目录结构(15)



一级目录结构

29

5.2 文件目录项和文件目录结构(16)



二级目录结构

30

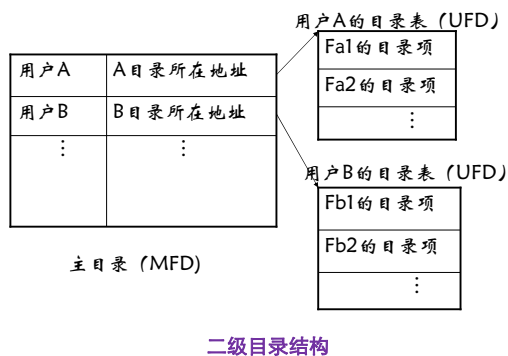
5.2 文件目录项和文件目录结构(17)

➤ 二级目录结构

- 为每个用户建立独立的目录结构
- 由一个主文件目录和若干用户文件目录组成
- 主目录中的目录项记录各用户目录的名字及目录文件所在的磁盘地址
- 创建新用户
 - 系统为新用户在主文件目录表中找一个空表目，并为其分配一个存放用户文件目录的区域，然后把用户名和该区域的起始地址填写在该对应表目中

31

5.2 文件目录项和文件目录结构(18)



32

5.2 文件目录项和文件目录结构(19)

- 创建新文件
 - 用户建立一个新文件时，文件系统先按用户名在主目录中找到相应的文件目录的地址，然后在用户目录表中为其建立一个目录项，填上文件名以及文件的管理和控制信息
- 访问文件
 - 先按用户名在主目录中找到相应文件目录项 (地址信息)

33

5.2 文件目录项和文件目录结构(20)

- 按文件名在用户文件目录中找到文件的相关信息
- 完成指定的文件操作
- 删除文件
 - 按用户名在主目录中找到相应文件目录项(地址信息)
 - 按文件名在用户文件目录中找到相应文件目录项, 释放文件所占用的块并撤消该目录项

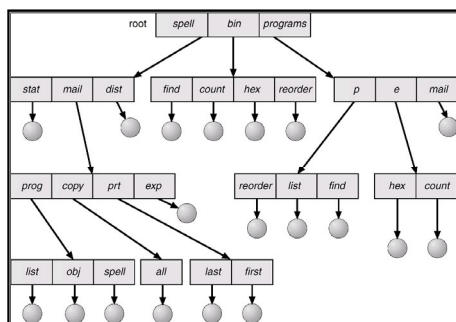
34

5.2 文件目录项和文件目录结构(21)

- 当该用户的文件目录中的文件都被撤消时, 将其所占目录表区释放, 并将该用户在主目录表中的目录项撤消
- 优点
 - 解决了文件的重名(用户名|文件名)问题和文件共享问题
 - 降低了文件查找时间
- 缺点
 - 增加了系统开销
 - 对用户进行隔离

35

5.2 文件目录项和文件目录结构(22)



多级目录结构

36

5.2 文件目录项和文件目录结构(23)

➤ 多级目录结构 (层次型/树型)

- 有一个主目录(又称根目录), 在根目录下又有许多子目录和普通文件的记录, 每个子目录下依次也有许多子目录或文件作为其目录项
- 优点
 - 层次结构清晰, 解决重名问题
 - 便于管理和保护
 - 有利于文件分类
 - 提高文件检索速度
- 缺点
 - 禁止共享文件和目录

37

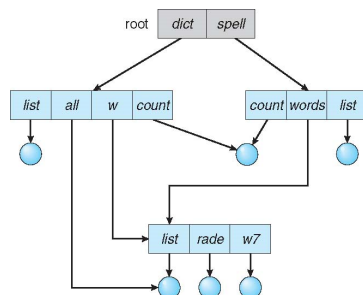
5.2 文件目录项和文件目录结构(24)

- 按路径名逐层查找文件, 需多次访问磁盘, 影响速度
- 绝对路径
 - 文件的固有名称由根(或主目录)到文件通路上所有目录与该文件的符号名拼接而成
- 相对路径
 - 当前目录 (工作目录)
 - 文件名由“当前目录”到欲访问文件之间的所有符号名拼接而成
- 增加目录与删除目录

38

5.2 文件目录项和文件目录结构(25)

➤ 无环图目录结构



39

5.3 文件操作命令(1)

➤ CREATE: 创建一个文件

- 在指定设备上为文件产生一个目录项, 分配必要的外存空间, 并设置文件相关属性

➤ DELETE: 删除一个文件

- 当一个文件不再需要时, 可用此命令将它删除。文件删除后, 其对应的目录项不再存在, 并回收该文件占用的存储空间

➤ READ: 读文件

- 调用者必须指出要读取的文件名, 数据大小以及存放数据的主存缓冲区地址

40

5.3 文件操作命令(2)

➤ WRITE: 写文件

- 将主存缓冲区中的数据写到指定的文件中, 通常是写在当前位置
- 如果当前位置是文件尾, 则文件内容增加, 否则, 将覆盖已存在的数据

➤ APPEND: 追加命令

- 将数据追加到文件尾

➤ SEEK

- 将文件的读写指针由当前位置重新定位到指定位置, 以便随机存取文件中任意位置的数据

41

5.3 文件操作命令(3)

➤ OPEN: 打开文件

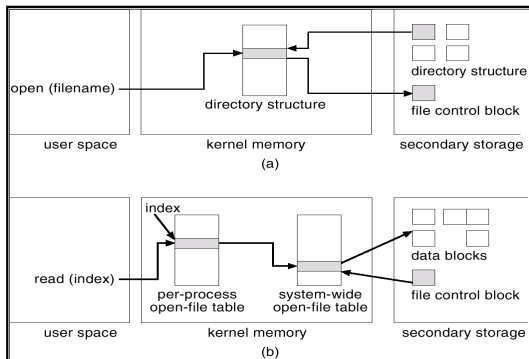
- 进程在使用文件前, 必须先打开该文件
- 打开: 将文件的目录项复制到主存一个专门区域, 从而建立进程与文件的联系, 加快进程对文件的存取速度

➤ CLOSE: 关闭文件

- 当进程对文件的所有存取完成后, 该文件在主存专用区对应的目录项就不再需要, 应该将它关闭, 以便释放这部分主存空间, 让以后需要打开的文件占用。若该目录项调入主存后修改过, 还要复制到磁盘
- 文件关闭后, 不能再使用

42

5.3 文件操作命令(4)



43

5.3 文件操作命令(5)

- GET ATTRIBUTES
 - 获取文件属性
- SET ARRTIBUTES
 - 设置文件属性
- RENAME
 - 更改文件名字

44

5.4 文件逻辑结构和文件访问方法(1)

- 文件结构
 - 文件的组织形式
 - 用户使用角度
 - 文件的逻辑结构：用户如何组织和使用文件
 - 系统实现角度
 - 文件的物理结构：文件的物理存储
- 文件系统的重要作用
 - 在用户逻辑文件和相应存储设备上的物理文件之间建立映射，实现二者之间的相互转换。

45

5.4 文件逻辑结构和文件访问方法(2)

➤ 文件逻辑结构的基本要求

- 能够提高检索速度
- 便于修改
- 降低文件的存储费用

➤ 文件逻辑结构分类

- 无结构的字符流
- 有结构的记录式文件
 - ◆ 记录：一组相关联的数据项
 - 定长记录
 - 变长记录

46

5.4 文件逻辑结构和文件访问方法(3)

一个字节



字节流文件

定长记录



变长记录



记录式文件

47

5.4 文件逻辑结构和文件访问方法(4)

➤ 文件访问方法

- 顺序访问
 - ◆ 严格按照字符流或记录排列的先后次序依次访问
 - ◆ 对文件的每一次访问都在前一次访问的基础上进行
 - ◆ 系统设置两个位置指针指向其中要读写的字节位置或记录位置。根据要读写的字节个数或记录长度，系统自动修改指针位置

48

5.4 文件逻辑结构和文件访问方法(5)

- 通常对于变长记录文件采用顺序访问取法
- 直接访问（随机访问）
 - 通常针对定长记录文件
 - 在请求对某个文件进行访问时，要指出访问的记录号
- 按键访问
 - 按给定的字段值进行访问
 - 广泛应用于数据库系统

49

5.4 文件逻辑结构和文件访问方法(6)

- 索引化顺序访问方法

键	逻辑地址	姓 名	其它属性
An Qi		An Qi	
Bao Rong		An Kang	
Chen Lin			
		Bao Rong	
		⋮	

逻辑文件

50

5.5 文件物理结构和文件存储介质

5.5.1 文件的物理结构

5.5.2 文件的存储介质

51

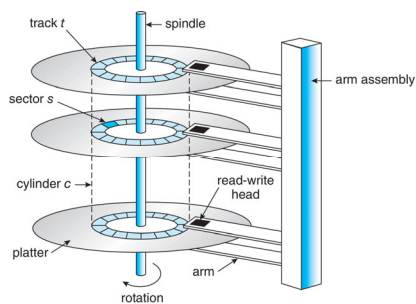
5.5.1 文件的物理结构(1)

文件的物理结构

- 指一个文件在文件存储器上存储方式及其与文件逻辑结构的关系
- 说明
 - 物理块：将文件存储器的存储空间划分成的若干个大小相等的块
 - 物理文件：文件存储器上的文件
 - 物理记录：存放文件记录的物理块
 - 无结构的字符流
 - 记录文件：假定逻辑记录长度与物理块大小相等

52

5.5.1 文件的物理结构(2)



53

5.5.1 文件的物理结构(3)

文件的物理结构

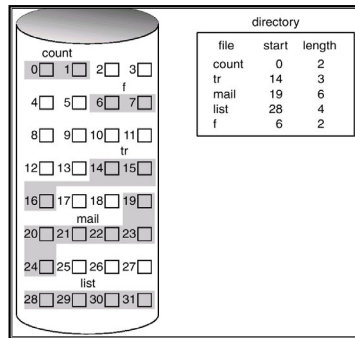
- 连续
- 链接
- 索引

连续文件

- 把逻辑上连续的文件信息存储在连续的物理块中的一种组织方式
- 优点
 - 实现简单
 - 只要记住文件的第一块所在位置及文件包括的块数即可

54

5.5.1 文件的物理结构(4)



文件的连续分配

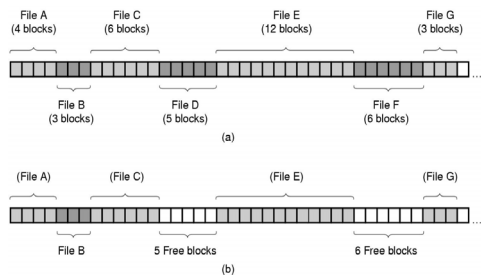
55

5.5.1 文件的物理结构(5)

- 存取速度快
 - 只要访问一次文件的管理信息，就可以方便地存取到任一记录
- 缺点
 - 不灵活
 - 要求在文件创建时，就给出文件的最大长度，文件不能动态扩展
 - 产生碎片
 - 当文件被删除时，文件存储空间可能出现许多小的无法利用的碎块

56

5.5.1 文件的物理结构(6)



(a) 7个文件的连续分配

(b) 文件D和F释放后的磁盘空间状态

57

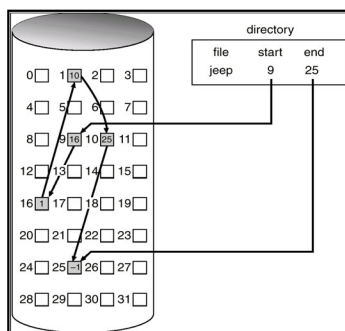
5.5.1 文件的物理结构(7)

➤ 链接文件

- 把逻辑上连续的信息文件存储在不连续的物理块中，存放信息的物理块中另设一个指针指向下一个物理块。文件最后一个物理块的指针通常为0/-1，以指示该块是链尾
- 优点
 - 提高磁盘空间利用率,不存在外部碎片问题
 - 有利于文件插入和删除
 - 有利于文件动态扩充
- 缺点
 - 链接指针需要额外空间
 - 不利于随机存取

58

5.5.1 文件的物理结构(8)



文件的链接分配

59

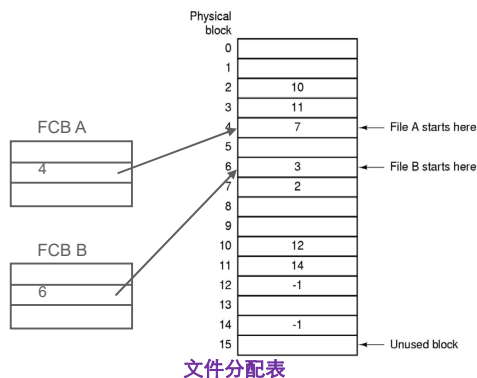
5.5.1 文件的物理结构(9)

• 改进

- 把指针字从文件的各物理块中取出，放在一个表中，此表叫做盘文件映射表
- MS-DOS就使用这种方式分配和管理磁盘空间，并将该表叫做文件分配表FAT
- 优点
 - 既能方便地实现顺序存取，而且也容易实现随机存取
- 缺点
 - 不缓存FAT，将导致大量磁头寻道时间
 - 缓存FAT，系统工作期间，整个表必须在主存，占用内存太多

60

5.5.1 文件的物理结构(10)



61

5.5.1 文件的物理结构(11)

➤ 示例：假定磁盘块大小为1KB，磁盘空间的管理采用文件分配表FAT，对一个512 MB的硬盘，需要占用多少磁盘空间？

➤ 解：

磁盘块大小为1KB时，该盘可划分的盘块数为： $512\text{MB} \div 1\text{KB} = 512\text{K}$ 块；

512K块要用19位二进制位表示，也即一个盘块号用19位才能表示出来；

$19 \div 8 \approx 2.5\text{B}$ ，即一个表项占用2.5B。

故FAT表的长度为 $2.5\text{B} \times 512\text{K} = 1280\text{KB}$ ；

所以占用的磁盘块为： $1280\text{KB} / 1\text{KB} = 1280$

62

5.5.1 文件的物理结构(12)

➤ 索引文件

● 在文件目录表中为每个文件保留一个索引表块号，该索引块指出文件的逻辑块与物理块的映射关系

● 优点

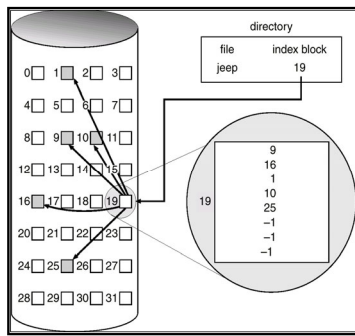
✦ 允许文件动态修改，增加了使用的灵活性

✦ 允许用户按照要求，直接对文件进行随机存取

● 缺点

63

5.5.1 文件的物理结构(13)



文件的索引分配

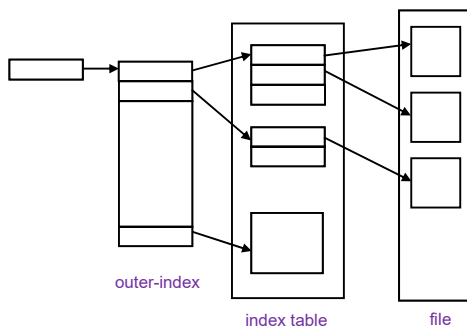
64

5.5.1 文件的物理结构(14)

- 索引表要占用额外空间
- 降低了文件存取速度
 - 至少需要访问存储器二次
 - 一次访问索引表
 - 一次访问文件信息
- 多级索引
- 混合索引

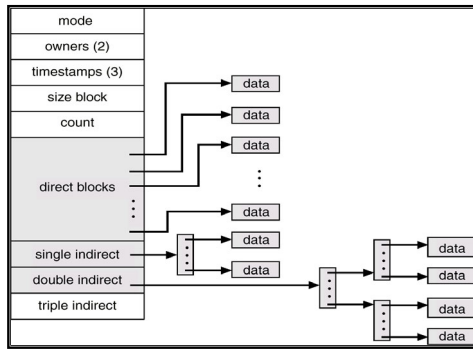
65

5.5.1 文件的物理结构(15)



66

5.5.1 文件的物理结构(16)



UNIX系统的混合索引结构

67

5.5.1 文件的物理结构(17)

➤ 示例：设文件索引节点中有7个地址项，其中4个地址项为直接地址索引，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的最大长度是

A.33KB

B.519KB

C.1057KB

D.16513KB

68

5.5.1 文件的物理结构(18)

```
struct ext2_inode {
    _le16 i_mode;           //文件类型和访问权限
    _le16 i_uid;            //拥有者的标识符
    _le32 i_size;           //以字节为单位的文件长度
    _le16 i_gid;           //组标识符
    _le16 i_links_count;    //硬链接计数
    _le32 i_block[15];     //索引表, 15×4B
    _le32 i_blocks;        //文件的数据块数
    _le32 i_file_acl;      //文件访问控制表ACL
    .....
};
```

69

5.5.1 文件的物理结构(19)

- 索引表：是一个有15个元素的数组，每个元素占4B
- 数组的15个元素有4种类型
 - 最初的12个元素是直接索引项，给出文件最初的12个逻辑块号对应的物理块号
 - 索引12是一次间接索引块，对应的文件逻辑块号从12到 $(b/4) + 11$ ，b是盘块大小
 - 索引13是二次间接索引块，对应的文件逻辑块号从 $b/4 + 12$ 到 $(b/4)^2 + (b/4) + 11$
 - 索引14是三次间接索引块，对应的文件逻辑块号从 $(b/4)^2 + (b/4) + 12$ 到 $(b/4)^3 + (b/4)^2 + (b/4) + 11$

70

5.5.1 文件的物理结构(20)

- 符号链接文件
 - 如果文件路径名小于60个字符，直接存放在 `i_block[]` 中
 - 如果大于60个字符，需要一个单独的数据块
- 设备文件、管道文件、套接字文件
 - 所有必要信息都存放在索引节点中，不需要额外的数据块

71

5.5.1 文件的物理结构(21)

- 示例：考虑当前有一个由100个磁盘块组成的文件。假定采用索引结构，且文件控制块已经在主存。对于采用连续、链接（单向链）和一级索引结构（且索引块已经在主存）时，如果下面的条件成立，访问一个文件块，需要执行多少次磁盘I/O操作？在连续分配时，假定在开头已经没有空间扩展文件，但在结尾处还有扩展的空间。并假定要扩展的信息块已经在主存。

72

5.5.1 文件的物理结构(22)

- 这个块被加到文件开头;
- 这个块被加到文件中间(50之后);
- 这个块被加到文件结尾;
- 从文件开头删除一块;
- 从文件中间删除一块(删除51);
- 从文件结尾删除一块。

➤ 解:

连续: 201, 101, 1, 0, 98, 0

链接: 1, 52, 102, 1, 52, 100

索引: 1, 1, 1, 0, 0, 0

73

5.5.2 文件的存储介质(1)

➤ 文件存储设备

- 磁带
 - 顺序存取设备
- 磁盘、光盘
 - 直接存取设备
- 特点
 - 存储容量大
 - 存取速度快
 - 以块为单位进行存取访问

74

5.5.2 文件的存储介质(2)

存取设备、存取方法、物理结构
之间的关系

存取设备	磁盘			磁带
物理结构	顺序结构	链接结构	索引结构	顺序结构
存取方法	直接/ 顺序	顺序	直接/ 顺序	顺序
文件长度	固定	可变/ 固定	可变/ 固定	固定

75

5.5.2 文件的存储介质(3)

➤ 文件存储介质的主要参数

- 容量
- 物理记录尺寸
- 可拆卸性
- 潜在时间
- 寻找时间
- 传输速度

76

5.5.2 文件的存储介质(4)

➤ DOS文件卷的结构

- 硬盘低级格式化
 - 将磁盘划分成若干磁道、扇区,由生产厂家完成
 - 扇区的头标记录其柱面号、磁头号 and 扇区号
- 硬盘分区 (FDISK命令)
 - 硬盘主引导扇区
 - 硬盘的0面0道1扇区,属于整个硬盘而不属于某个独立的分区
- 分区格式化 (FORMAT命令)
 - 对分区进行具体数据组织,即制作文件系统

77

5.5.2 文件的存储介质(5)

- 硬盘主引导扇区
 - 硬盘主引导程序: 位于该扇区0-1BDH处
 - 硬盘分区表DPT: 位于该扇区1BEH-1FDH处
 - 每个分区表占用16B,共有4个分区表
 - 16B意义如下: 0为自举标志(80H为可引导分区,00H为不可引导分区); 1-3是分区的起始地址(磁头号扇区号柱面号); 4是分区类型(07H为NTFS分区); 5-7是分区的结束地址; 8-11是分区首扇区的绝对扇区号; 12-15是分区占用的总扇区数
 - 主引导扇区的有效标志: 位于该扇区1FEH-1FFH处,固定值为AA55H

78

5.5.2 文件的存储介质(6)

- 分区规范规定
 - 一个硬盘可以有多个主分区，但最多只能有一个扩展分区
 - 一个扩展分区可以划分为多个逻辑分区，所以一个硬盘最多只能划分为4个主分区或者3个主分区和一个扩展分区
 - 扩展分区只作为数据盘使用
- DOS卷的组成
 - 引导或保留扇区
 - 文件分配表1/文件分配表2
 - 根目录区
 - 文件数据区

79

5.6 文件存储器存储空间管理(1)

- 文件存储器存储空间的基本分配单位为磁盘块/簇
- 常用的管理方法
 - 空白文件目录
 - 空闲块链表
 - 位映像表(bit map)或位示图
- 空白文件目录
 - 空白文件
 - 一个连续未用的空闲盘块区

80

5.6 文件存储器存储空间管理(2)

- 空白文件目录
 - 为所有空白文件建立的一张表，用来记录整个文件存储器的空闲未用空间，每个空白文件占用其中的一个表目
- 适合于文件的静态分配(连续分配)
- 存储空间的分配与回收

第一个物理块号	空白块个数
15	4
23	10
...	...

81

5.6 文件存储器存储空间管理 (3)

空白块链表

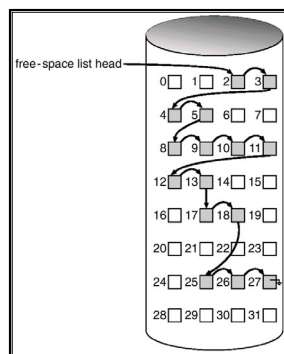
- 适合文件动态分配
- 将所有空闲块链接成一个链，在主存保留一个链头指针
- 存储分配与回收
 - 直接从链头取出相应数量的空闲块分配
 - 释放的空闲块依次放入链头
- 优点
 - 管理简单

82

5.6 文件存储器存储空间管理 (4)

缺点

- 工作效率较低，分配和释放空闲块时要多次访问磁盘，读写几个盘块才能完成对链的修改工作



83

5.6 文件存储器存储空间管理 (5)

成组空闲块链表

- 利用盘上的空闲块管理空闲块
- 每个磁盘块记录尽可能多的空闲块而组成一组，各组之间也用链指针链在一起，在主存保留该链表的链头指针
- 存储分配
 - 将链头指针所指的组块读入主存，从中找到一个空闲块进行分配；当未满足分配要求时，再从链中将记录下一组空闲块的盘块读入内存，再进行分配

84

5.6 文件存储器存储空间管理(6)

回收

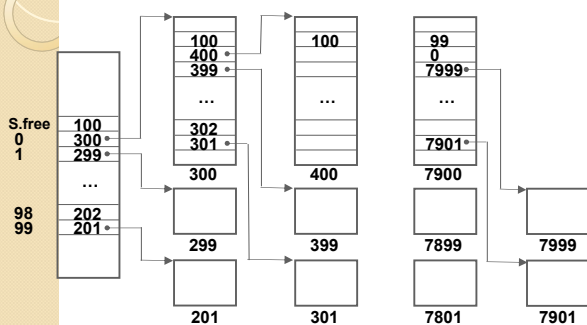
- 将释放的盘块记录在主存的组盘块中即可。若该块已满，除将该块内容写回磁盘外，可用刚释放块记录其它要释放块，并将其放到链头即可

适用结构

- 连续结构
- 链接结构
- 索引结构

85

5.6 文件存储器存储空间管理(7)



86

5.6 文件存储器存储空间管理(8)

► 示例：某个系统采用成组空闲块链法管理磁盘的空闲空间，目前磁盘的状态如下图所示：

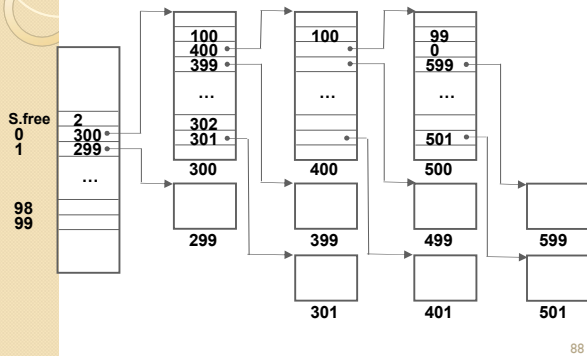
(1) 该磁盘块中目前还有多少个空闲块？

(2) 在为某个文件分配3个盘块后，系统要删除另一个文件，并回收它所占用的5个盘块，它们的盘块号依次为700、711、703、788、701，请画出回收后的盘块链接情况。

解：(1) 301

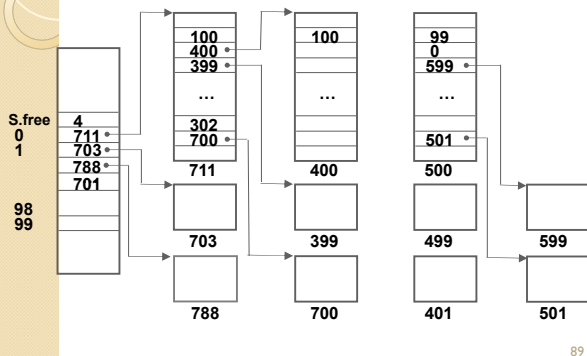
87

5.6 文件存储器存储空间管理(9)



88

5.6 文件存储器存储空间管理(10)



89

5.6 文件存储器存储空间管理(11)

位映像表 (位示图 Bit Vector)

- 适合文件静态和动态分配的最简单方法
- 使用一个位向量，磁盘中的每个块占用其中的一位
- 表中为0的位相应于空闲块，为1的位相应于已被占用的块
- 优点
 - 比较容易找到一个或几个连续的空白块
 - 尺寸固定，对于小磁盘，位映像表可以保存在主存中，从而可高速地实现文件的分配和回收工作

90

5.6 文件存储器存储空间管理(12)

➤ 示例:

- 一个容量为1.3G的磁盘，每个磁盘块大小为512B，则
 - 位映像表占据的空间大小为 $(1.3G/512B)/8=332.8KB$
 - 若按簇进行分配，每个簇包含4个磁盘块，则位映像表占据的空间大小为 $(1.3G/512B)/4/8=83.2KB$
- 一个容量为1T的磁盘，每个磁盘块大小为4KB，则位映像表占据的空间大小为 32MB ($2^{40}/2^{12} = 2^{28} \text{ bits} = 2^{25} \text{ bytes} = 2^5 \text{ MB}$)

91

5.6 文件存储器存储空间管理(13)

• 转换工作

- 将位映像表中的字位号转换成相应盘的物理地址(即磁头号、磁道号和扇区号)
 - 磁盘的相对块号=字节号 $\times n$ +位号
- 反转换
 - 字节号=相对块号/ n 商
 - 位号=相对块号 $\%n$ 余数

92

5.6 文件存储器存储空间管理(14)

➤ 示例: 有一计算机系统利用下图所示的位示图(行号、列号都从0开始编号)来管理磁盘块。如果盘块从1开始编号，每个盘块的大小为1KB，则: (1) 现要为文件分配两个盘块，试具体说明分配过程。2) 若要释放磁盘的第300块，应如何处理?

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5																
6																

93

5.6 文件存储器存储空间管理(15)

➤ 解:

(1) 两个空闲盘块位置(2,2), (3,6)

则空闲盘块号为:

$$b1 = 2 * 16 + 2 + 1 = 35$$

$$b2 = 3 * 16 + 6 + 1 = 55$$

(2) 第300块对应的行列号为

$$i = (300 - 1) / 16 = 18$$

$$j = (300 - 1) \% 16 = 11$$

94

5.7 文件共享与保护(1)

➤ 文件共享

- 允许多个用户共同使用一个文件
- 既节省外存空间, 又能减少I/O信息量和主存中文件的副本

➤ 文件保护

- 防止数据丢失和被无权使用的用户窃取

➤ 文件存取控制

- 一个用户对其文件所做的“谁能使用和如何使用”的规定, 主要对文件共享加以限制

95

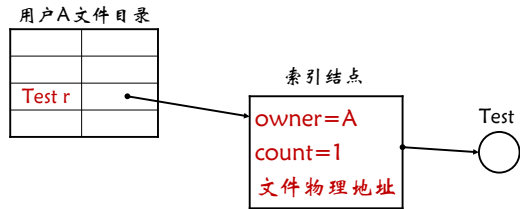
5.7 文件共享与保护(2)

➤ 文件共享

- 基于索引节点的共享 (链接计数)
 - 索引节点: 存放文件的物理地址和文件属性等信息
 - FCB: 存放文件名及指向索引节点的指针
 - 链接计数
 - 存放在索引节点中
 - 记录链接到本索引节点的用户数目
- 共享方式

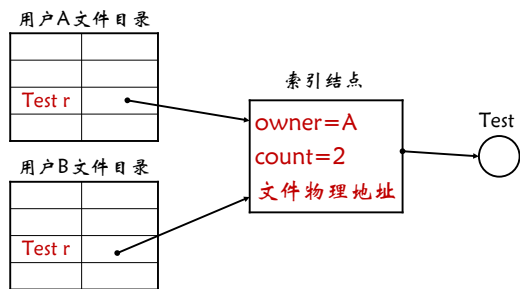
96

5.7 文件共享与保护(3)



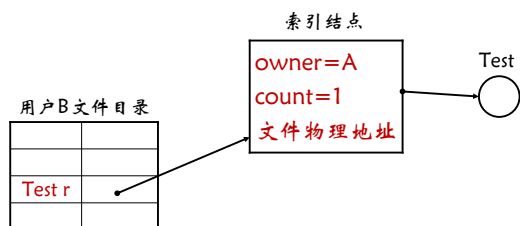
97

5.7 文件共享与保护(3)



98

5.7 文件共享与保护(3)



99

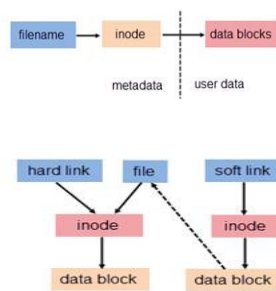
5.7 文件共享与保护(4)

• 基于符号链的共享

- 由系统创建一个LINK类型的新文件，新文件中只包含被链接文件的路径名
- 只有文件主拥有指向被共享文件索引节点的指针，其他用户只拥有路径
- 文件拥有者删除共享文件后，其他用户试图通过符号链访问该共享文件将失败
- 优点：可以链接任意文件
- 缺点：耗费磁盘空间，增加启动磁盘的频率

100

5.7 文件共享与保护(5)



101

5.7 文件共享与保护(6)

- 示例：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

答案：B

102

5.7 文件共享与保护(7)

➤ 文件保护

- 防止文件的丢失和破坏
 - 解决方法：复制/恢复
 - 周期性的全量转存
 - 增量转储
- 防止其他用户对文件的有意破坏或窃取
 - 增设防护措施
 - 防止由计算机病毒引起的文件破坏
 - 用户的鉴别——口令

103

5.7 文件共享与保护(8)

- 对文件加密
- 物理鉴定

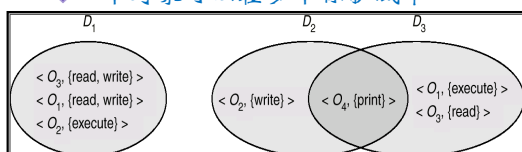
➤ 文件的存取控制

- 防止核准的用户误用文件和未核准的用户存取文件
- 文件目录中有关文件存取控制信息部分，规定了各类用户对各个文件的存取权限
- 存取控制方法
 - 保护域
 - 存取控制表

104

5.7 文件共享与保护(9)

- 保护域
 - 规定了进程对一组对象的存取权限
 - 一个域是一组(对象，存取权限)偶对，每一个偶对说明一个对象和允许对其施加的一组操作
 - 一个对象可以在多个保护域中



5.7 文件共享与保护(10)

• 存取访问矩阵

- 行: 域 / 列: 目标对象
- $\text{access}(i,j)$: 在域 D_i 中执行的进程能对对象 O_j 所施加的操作集合

object domain	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

5.7 文件共享与保护(11)

• 存取控制矩阵的实现

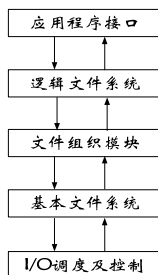
- 按列存取: 存取控制表ACL(Access Control List)
- 按行存取: 存取权限表

```
1 [root@study doc]# getfacl /root
2 getfacl: Removing leading '/' from absolute path names
3 # file: root
4 # owner: root
5 # group: root
6 user::r-x
7 user:chen:rw-
8 group::r-x
9 mask::rw-
10 other::---
```

5.8 文件系统组织结构(1)

- 文件系统是用户和外存设备之间的接口
- 文件系统的层次结构

- 应用程序接口
- 逻辑文件系统
- 文件组织模块
- 基本文件系统
- I/O调度及控制模块



5.8 文件系统组织结构(2)

➤ 应用程序接口

- 检查由用户提供的命令句法的正确性和参数的合法性

➤ 逻辑文件系统

- 负责目录的管理和维护，按照命令给定的文件名查找目录
- 创建文件：增加目录项
- 打开文件：检查各级目录，找到相应的目录项
- 读/写文件：检查文件是否已经以读/写方式打开

109

5.8 文件系统组织结构(3)

➤ 文件组织模块

- 负责将文件的读/写位置转换成文件中的相对块号，进而转换成在存储器中的物理块号
- 管理磁盘空闲空间
- 将上层传下来的命令转换成对基本文件系统的调用

➤ 基本文件系统

- 将上层传下来的命令和物理块转换成对设备驱动程序的调用

110

5.8 文件系统组织结构(4)

➤ I/O调度和控制模块

- 由设备驱动和中断处理模块组成
- 将上层传来的命令转换成硬件设备的专用I/O指令和设备地址，控制设备完成与主存之间的信息传输
- 负责多个读/写命令的排队和调度

111

5.8 文件系统组织结构(5)

示例：利用文件系统的一般模型，说明完成如下的用户进程中使用的读指令时，系统应该实现的各个步骤。

```
read file (sqrt) record(7)into  
location(15 000)
```

注：该命令的含义为将文件sqrt的7号记录读到内存的15 000单元中。

假定：

- (1)文件的存取方法为直接存取；
- (2)文件为连续文件、起始物理块号为5；

112

5.8 文件系统组织结构(6)

(3)保护措施采取了存取控制权限；

(4)物理块长为1 000B，逻辑记录长为500B；

(5)目录采用了符号文件目录(SFD)和基本文件目录(BFD)。其中，SFD包括文件名和文件的内部标识；BFD包括记录大小、记录个数、起始物理块地址、存取控制权限等。

113

5.8 文件系统组织结构(7)

- (1) 由文件名查SFD，若找不到，给出错误信息，返回用户程序；否则，得到其BFD中的内部标识。
- (2) 根据文件在BFD中的内部标识，找到该文件的管理控制信息。
- (3) 根据文件的管理控制信息，由存取控制模块检查该进程是否允许读文件，若不可以，错误返回；否则转下一步。

114

5.8 文件系统组织结构(8)

- (4) 根据文件的控制信息, 查得文件为连续文件、起始物理块号为5。物理块长为1 000B, 逻辑记录长为500B。要读的7号记录在文件中的字节号为 $7 \times 500 = 3\,500$ 。文件的相对块号 $3\,500 / 1\,000 = 3$, 块内地址为500。文件为连续文件, 故文件所在的物理块号为 $5 + 3 = 8$ 。
- (5) 检查文件高速缓存中是否有物理块8, 若有, 从块内500偏移处读500个字节送内存15 000单元中, 完成这次的读操作。否则, 转下一步。
- (6) 调用设备决策模块, 启动设备, 将第8个物理块读入内存。从块内500偏移处读500个字节送内存15 000单元中, 完成这次的读操作。

115

5.9 存储器映射文件(1)

- 允许进程分配虚存的一部分地址空间, 将磁盘上的一个文件映射到该空间
- 对文件块的存取通过访问虚存的一个页实现, 当访问的页不在主存时, 产生缺页中断, 将其读入主存
- 当需要对存储器映射文件存取时, 才实际传输文件数据
- OS提供映射文件的系统调用

116

5.9 存储器映射文件(2)

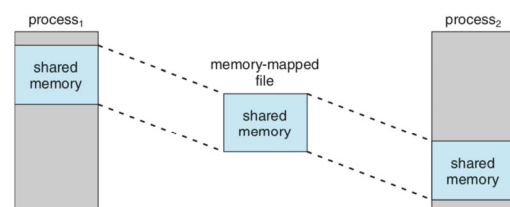


Figure 13.14 Shared memory using memory-mapped I/O.

117

5.9 存储器映射文件 (3)

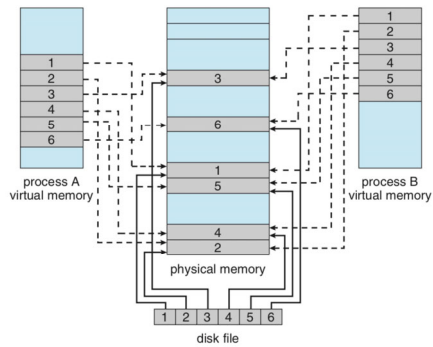


Figure 13.13 Memory-mapped files.

118

本章要点

- 文件与文件系统
- 文件目录与文件目录结构
- 文件目录项/文件控制块
- 文件逻辑结构和文件访问方法
- 文件物理结构
- 文件存储空间管理
- 文件共享
- 文件保护

119