

考研 408—操作系统讲义【文件管理】

一. 文件和文件系统

文件系统的管理功能是将管理的程序和数据通过组织为一系列文件的方式实现

1. 数据项、记录、文件

I. 数据项

数据项是文件系统中最低级的数据组织形式，可分为以下两种类型：

i) 基本数据项：用于描述一个对象的某种属性的一个值，如姓名、日期或证件号等，是数据中可命名的最小逻辑数据单位，即原子数据

ii) 组合数据项：由多个基本数据项组成。

II. 记录

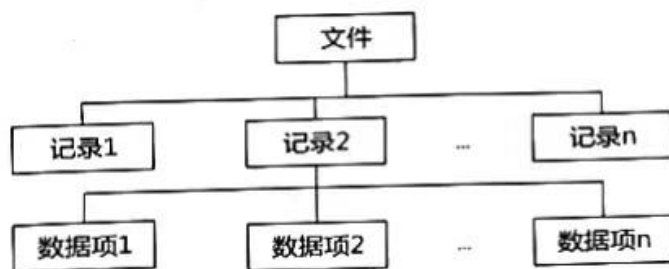
记录是一组相关的数据项的集合，用于描述一个对象在某方面的属性，如一个考生报名记录包括考生姓名、出生日期、报考学校代号、身份证号等一系列域

III. 文件

文件是指由创建者所定义的一组相关信息的集合，逻辑上可分为有结构文件和无结构文件两种。在有结构文件中，文件由一组相似记录组成，如报考某学校的所有考生的报考信息记录，又称记录式文件；而无结构文件则被看成是一个字符流，比如一个二进制文件或字符文件，又称流式文件。

文件的属性可以包括：

- ① 文件类型 ② 文件长度 ③ 文件的物理位置 ④ 文件的建立时间



2. 文件名和类型

I. 文件名和扩展名

在不同的系统中，对文件名的规定是不同的，在一些老的系统中，名字的长度受到限制。

扩展名是添加在文件后面的若干附加字符，有称为后缀名，用于指示文件的类型。

II. 文件类型

按文件的性质和用途分类

- i) 系统文件：系统软件构成的文件，如内核，系统应用程序，数据；只允许用户执行，不能读写和修改
- ii) 用户文件：由用户建立的文件，如源程序、目标程序和数据文件等；用户将这些文件托给系统保管；只允许文件所有者和所有者授权用户使用
- iii) 库文件：标准子例程及常用的例程构成的文件。这类文件允许用户调用，但不允许用户修改

按文件中数据的形式分类

- i) 源文件：由源程序和数据构成的文件。通常由终端或输入设备输入的源程序和数据所形成的文件都属于源文件
- ii) 目标文件：把源程序经过相应语言的编译程序编译过，但尚未经过链接程序链接的目标代码所构成的文件。它属于二进制文件。通常，目标文件所使用的后缀名是“.obj”
- iii) 可执行文件：把编译后所产生的目标代码再经过链接程序链接后所形成的文件

按存取控制属性分类

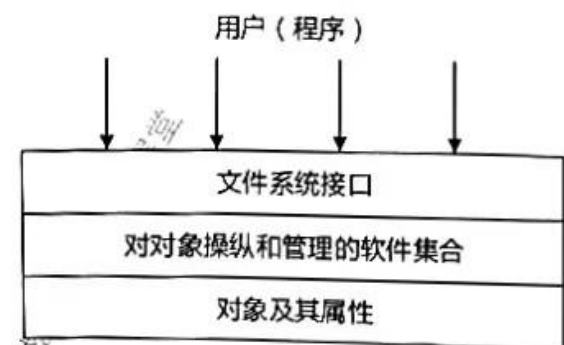
- i) 只执行文件：只允许被核准的用户调用执行，既不允许读，更不允许写
- ii) 只读文件：只允许文件主及被核准的用户去读，但不允许写
- iii) 读写文件：允许文件主和被核准的用户去读或写的文件
- iv) 无保护文件

按文件的内容形式和系统处理方式分类

- i) 普通文件：由 ASCII 码或二进制码组成的文件。一般用户建立的源程序文件、数据文件、目标代码文件及操作系统自身代码文件、库文件等都是普通文件，它们通常存储在外存储设备上
- ii) 目录文件：由文件目录组成的，用来管理和实现文件系统功能的系统文件，通过目录文件可对其它文件的信息进行检索。目录文件也是由字符序列构成，可进行与普通文件一样的各种操作
- iii) 特殊文件（设备文件）：特指系统中的各类 I/O 设备。为了便于统一管理，系统将所有的输入/输出设备都视为文件，按文件方式提供给用户使用。

3. 文件系统的层次结构

文件系统的模型可以分为三个层次：最底层是对象及其属性，中间层是对对象进行操纵和管理的软件集合，最高层是文件系统提供给用户的接口



图：文件系统模型

I. 对象及其属性

文件管理系统管理的对象如下：

① 文件 ② 目录 ③ 磁盘存储空间

II. 对对象操作和管理的软件集合

文件系统的功能大多是在这一层实现的，其中包括有：

- i) 对文件存储空间的管理
- ii) 对文件目录的管理
- iii) 用于将文件的逻辑地址转换为物理地址的机制
- iv) 对文件读和写的管理
- v) 对文件的共享与保护等功能

把与文件系统有关的软件分为四个层次

- i) I/O 控制层：文件系统的最低层，主要由磁盘驱动程序等组成，称为设备驱动程序层
- ii) 基本文件系统层：主要用于处理内存与磁盘之间数据块的交换
- iii) 基本 I/O 管理程序：该层用于完成与磁盘 I/O 有关的事务
- iv) 逻辑文件系统：用于处理与记录 and 文件相关的操作

文件系统的接口

- i) 命令接口：是指作为用户与文件系统交互的接口，用户可通过键盘终端键入命令取得文件系统的服务
- ii) 程序接口：是指作为用户程序与文件系统的接口，用户程序可通过系统调用取得文件系统的服务

4. 文件操作

最基本的文件操作包括创建、删除、读、写和设置文件的读/写位置

一般 OS 都提供了更多对文件的操作，如打开和关闭一个文件及改变文件名等等

I. 最基本的操作

- i) 创建文件
- ii) 删除文件
- iii) 读文件
- iv) 写文件
- v) 设置文件的读/写的位置 (指针)

II. 文件的打开和关闭操作

当用户要求对一个文件实施多次读/写操作时，每次都要从检索目录开始，为了避免多次重复检索目录，在大多数 OS 中引入了“打开”这一文件系统调用，当用户第一次请求对某文件进行操作时，须先利用 open 系统调用将该文件打开。

III. 其他文件操作

常用的一类是有关对文件属性的操作，即允许用户直接设置和获取文件的属性

二. 文件的逻辑结构

用户看到的文件称为逻辑文件，它是由一系列的逻辑记录组成的。文件的逻辑记录时能够

被存取的基本单位

系统中的所有文件都存在两种形式的文件结构：

- I. 文件的逻辑结构：从用户观点出发所观察到的文件组织形式
- II. 文件的物理结构：文件的存储结构

1. 文件逻辑结构的类型

I. 按文件是否有结构分类

i) 有结构文件

- ✓ ● 定长记录：文件中所有记录的长度都是相同的，个数据项都处在记录中相同的位置，具有相同的顺序和长度。

优点：有效提高检索记录的速度和效率，方便对文件进行处理和修改

- ✓ ● 变长记录：文件中各记录的长度不相同，产生变长记录的原因可能是由于一个记录中所包含数据项数目并不相同。

缺点：记录的检索速度慢，不方便对文件进行处理和修改

ii) 无结构文件

- 系统中运行的大量的源程序、可执行文件、库函数等采用的是无结构文件形式（流式文件）

II. 按文件的组织方式分类

- ##### i) 顺序文件
- 指由一系列记录按某种顺序排列所形成的文件，其记录可以是定长记录也变长记录

- ##### ii) 索引文件
- 指为可变长记录文件建立一张索引表，为每个记录设置一个表项，以加快对记录的检索速度

- ##### iii) 索引顺序文件
- 为每个文件建立一张索引表，并不是为每一个记录建立一个索引表项。

2. 顺序文件

I. 顺序文件的排列方式

i) 串结构

按照存入时间的先后进行排序，各个记录之间的顺序与关键字无关。对文件进行检索的时候，每次都要从头开始，逐个记录的查找。缺点：耗时长

ii) 顺序结构

用户指定哪一个字段作为关键字。优点：提高检索效率

II. 顺序文件的优缺点

优点：① 存取效率高

缺点：① 查找性能差 ② 增加删除困难

3. 记录寻址

I. 隐式寻址方式

对于定长的顺序文件，如果已知当前记录的逻辑地址，很容易确定下一个记录的逻辑地址。

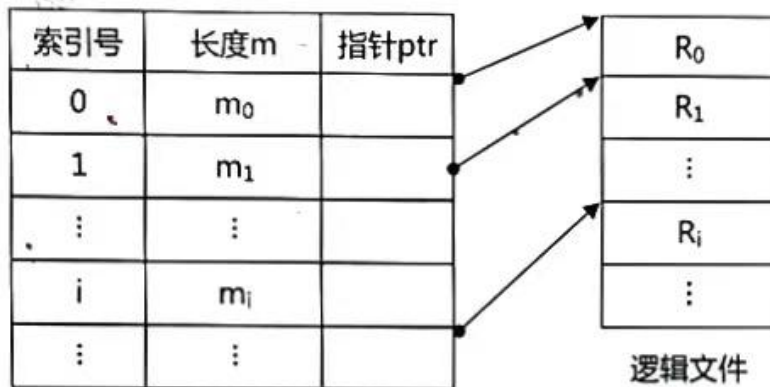
II. 显式寻址方式

用于对定长记录的文件实现直接或随机访问

- i) 通过文件中记录的位置
- ii) 关键字

4. 索引文件

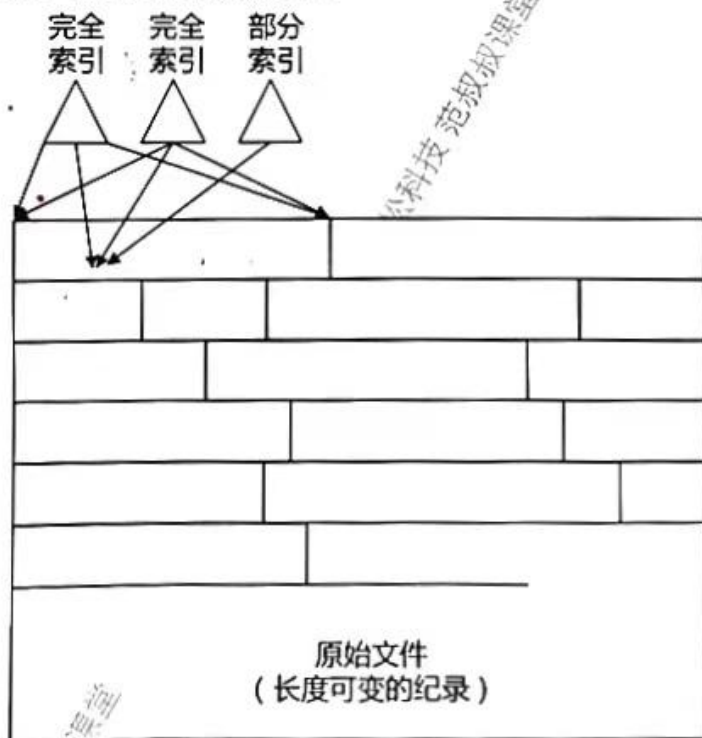
I. 按关键字建立索引



索引表

图：具有单个索引表的索引文件

II. 具有多个索引表的索引文件



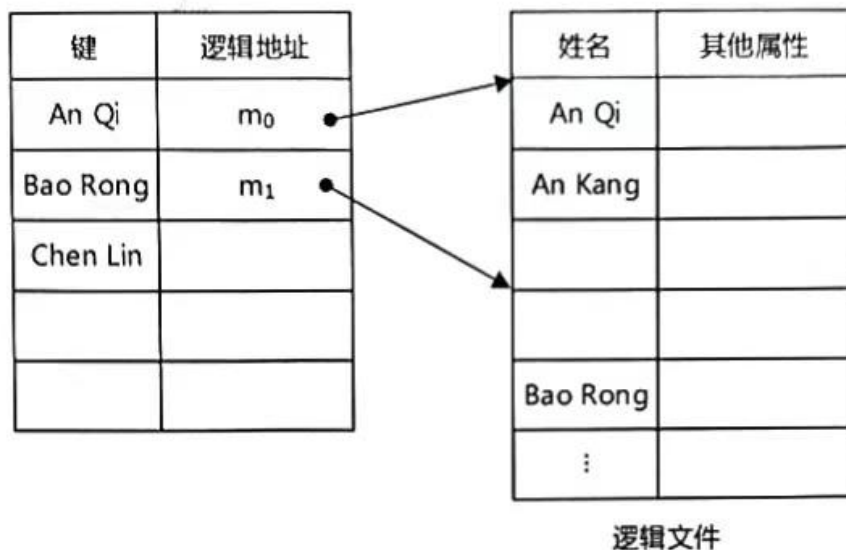
图：具有多个索引表的索引文件

5. 索引顺序文件

I. 索引顺序文件的特征

优点：克服变长记录的顺序文件不能随机访问，以及不便记录的删除和插入的缺点
两个新特征：① 引入文件索引表 ② 增加溢出文件

II. 一级索引顺序文件



图：索引顺序文件

III. 两级索引文件

6. 直接文件和哈希文件

I. 直接文件

对于直接文件，则根据给定的记录键值，直接获得指定记录的物理地址，换言之，**记录键值本身就决定了记录的物理地址**，这种由记录键值到记录物理地址的转换被称为键值转换。

II. 哈希文件

利用 Hash 函数可将记录键值转换为相应记录的地址，为了能够实现文件存储空间动态分配，通常由 Hash 函数所求得并非相应记录的地址，而是指向一目录表相应表目的指针，该表目的内容指向相应记录所在的物理块。



三. 文件目录

—— 存放指向文件 FCB，FCB 的有序集合

为了能够对文件实施有效的管理，必须对它们加以妥善组织，这主要是通过文件目录实现的，**文件目录也是一种数据结构，用于标识系统中的文件及其物理地址，供检索时使用**，对目录的管理要求如下：

- ✓ ● **实现按名存取**：即用户只须向系统提供所需访问的文件的名字，便能够快速准确地找到指定文件在外存上的存储位置，这是目录管理中最基本的功能
- ✓ ● **提高对目录检索速度**：通过合理地组织目录结构的方法，可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**：在多用户系统中，应该允许用户共享一个文件
- ✓ ● **允许文件重名**：系统应允许不同用户对不同文件采用相同的名字，以使用户按照自己的习惯给文件命名和使用文件

1. 文件控制块

为了能对文件进行正确的存取，必须为文件设置用于描述和控制文件的数据结构，称之为文件控制块 FCB，文件管理程序可借助于文件控制块中的信息，对文件施加各种操作，



文件与文件控制块一一对应，而人们把文件控制块的有序集合称为文件目录，一个文件控制块就是一个文件目录项。通常，一个文件目录也可被看成是一个文件，称为目录文件。文件控制块包含基本信息、存取控制信息、使用信息

I. 基本信息类

- i) 文件名 (标识一个文件的符号名，在每个系统中，每个文件都有唯一的名字，用户利用该名字进行存取)
- ii) 文件物理位置 (指文件在外存上的存储位置，包括存放文件的设备名、文件在外存上的起始盘块号、指示文件所占用的盘块数或字节数的文件长度)
- iii) 文件逻辑结构 (指示文件是流式文件还是记录式文件、记录数，文件是定长还是变长记录)
- iv) 文件物理结构 (指示文件是顺序文件、链式文件还是索引文件)

II. 存取控制信息

包括文件主的存取权限、核准用户的存取权限及一般用户的存取权限

III. 使用信息

包括文件的建立日期和时间、文件上一次修改的日期和时间及当前使用信息 (这项信息包括当前已打开该文件的进程数、是否被其他进程锁住、文件在内存中是否已被修改但尚未拷贝到盘上)

2. 索引结点

I. 索引结点的引入

文件目录通常是存放在磁盘上的，当文件很多时，文件目录可能要占用大量的盘块，在查找的过程中，先将存放目录文件的第一个盘块中的目录调入内存，然后把用户所给定的文件名和目录项中的文件名逐一对比。若未找到指定文件，则再将下一个盘块中的目录项调入内存。

在检索目录文件时，只用到了文件名，仅当找到一个目录项 (即其中的文件名与指定要查找的文件名相匹配) 时，才需要从该目录项中读出该文件的物理地址，而其他一些对该文件进行描述的信息，在检索目录时一概不用，显然，这些信息在检索目录时不需要调入内存。为此，在有的系统中，如 UNIX 系统，便采用了把文件名和文件描述信息分开的方法，亦即，使文件描述信息单独形成一个称为索引结点的数据结构，简称为 i 结点，在文件目录中的每个目录项由文件名和指向该文件所对应的 i 结点的指针所构成。

II. 磁盘索引结点

每个文件都有唯一的磁盘索引结点，磁盘索引结点信息与文件名等信息一起构成了

FCB)

还包括：

- i) 文件主标识符
- ii) 文件类型
- iii) 文件存取权限
- iv) 文件物理地址
- v) 文件长度
- vi) 文件连接计数

vii) 文件存取时间

(3) 内存索引结点

当文件被打开，要将磁盘索引结点拷贝到内存的索引结点中，增加以下内容

- i) 索引结点编号
- ii) 状态
- iii) 访问计数
- iv) 文件所属文件系统的逻辑设备号
- v) 链接指针

3. 目录结构

目录结构的组织，关系到文件系统的存取速度，也关系到文件的共享性和安全性，目前常用的目录结构形式有单级目录、两级目录、多级目录。

I. 单级目录结构

在整个系统中只建立一张目录表，每个文件占一个目录项，目录项中含文件名、文件扩展名、文件长度、文件类型、文件物理地址、状态位（表示目录项是否空闲）等

每当要建立一个新文件时，必须先检查所有的目录项，以保证新文件名在目录中是唯一的，然后再从目录表中找到一个空白目录项，填入新文件的文件名及其他说明信息，并置状态为 1，删除文件时，先从目录中找到该文件的目录项，回收该文件所占用的存储空间，然后再清除该目录项

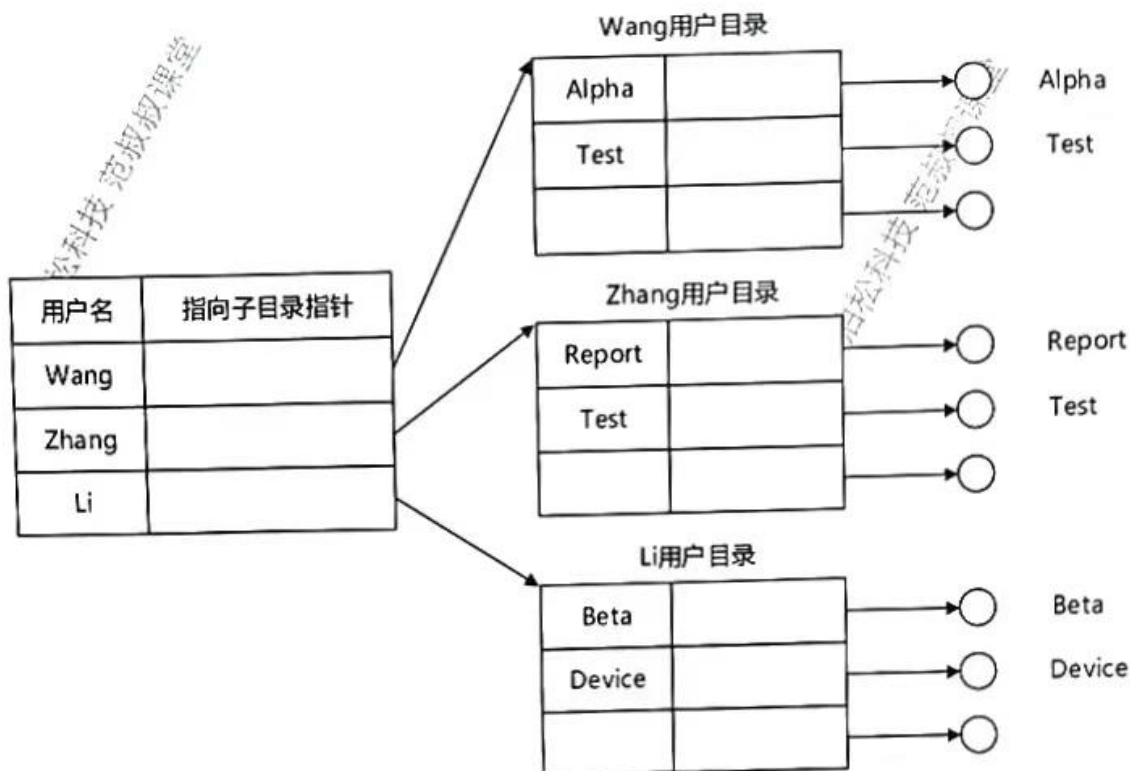
优点：是简单并且能够实现目录管理的基本功能-按名存取

缺点：

- 查找速度慢：查找一个目录项要花费较多的时间
- 不允许重名：在一个目录表中的所有文件，都不能与另一个文件有相同的名字，这是难以避免的
- 不便于实现文件共享：每一个用户都有自己的名字空间或命名习惯，因此，应该允许不同用户使用不同的文件名来访问同一个文件

II. 两级目录结构

每个用户建立一个单独的用户文件目录 UFD (User File Directory)，这些文件目录具有相似的结构，由用户所有文件的文件控制块组成。此外，系统中还有一个主文件目录 MFD (Master File Directory)，在主文件目录中，每个用户目录文件都占有一个目录项，其目录项包括用户名和指向用户目录文件的指针



两级目录结构克服了单级目录的缺点，具有如下优点：

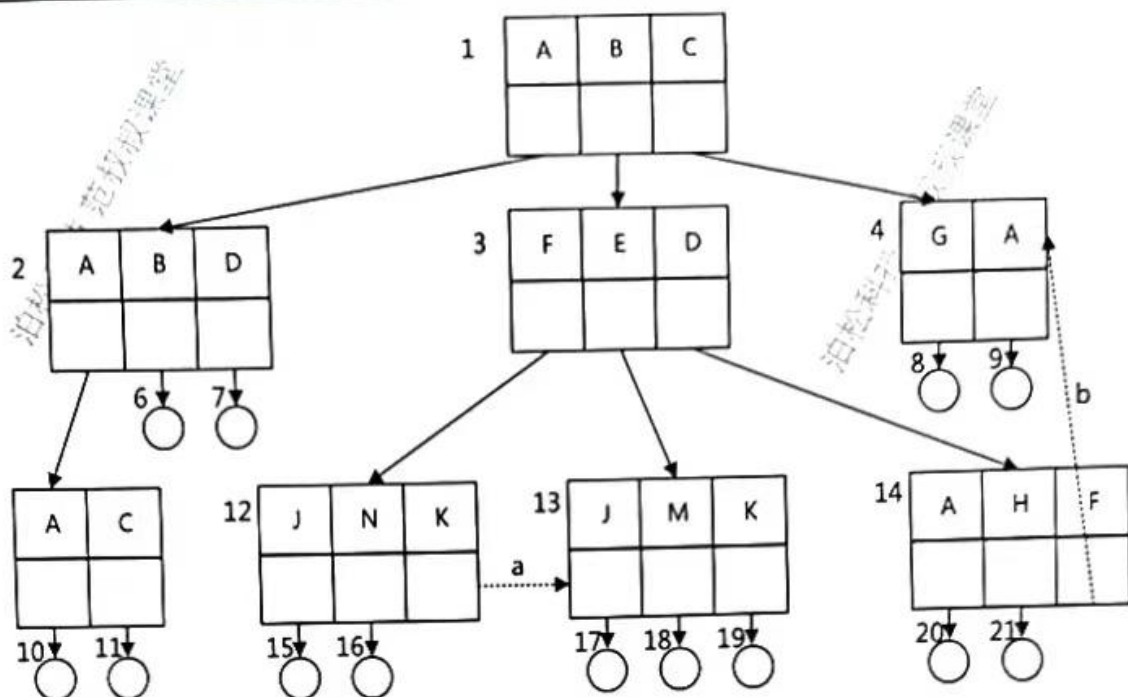
- 提高了检索目录的速度：如果在主目录中有 n 个子目录，每个用户目录最多为 m 个目录项，则为查找一指定的目录项，最多只需要检索 $n+m$ 个目录
- 在不同的用户目录中，可以使用相同的文件名：只要在用户自己的 UFD 中，每个文件名都是唯一的，不同用户可以有文件名相同的文件
- 不同用户还可使用不同的文件名来访问系统中同一个共享文件，但在多个用户需要合作完成一个大任务时，不便于用户之间共享文件

4. 树形目录结构

1. 树形目录

对于大型文件系统，通常采用三级或三级以上的目录结构，以提高对目录的检索速度和文件系统的性能。多级目录结构又称为树形目录结构，主目录被称为根目录，把数据文件称为树叶，其他的目录均作为树的结点

说明：方框代表目录文件，圆圈代表数据文件



II. 路径名和当前目录

系统中的每个文件都有唯一的路径名

III. 目录操作

- i) 创建目录
- ii) 删除目录
 - 不删除非空目录
 - 可删除非空目录
- iii) 改变目录
- iv) 移动目录
- v) 链接操作
- vi) 查找

5. 目录查询技术

当用户要访问一个已存在的文件时，系统首先利用用户提供的文件名对目录进行查询，找出该文件的文件控制块或对应索引结点，然后，根据 FCB 或索引结点中所记录的文件物理地址（盘块号），换算出文件在磁盘上的物理位置，最后，再通过磁盘驱动程序，将所需文件读入内存。目前常用的方式有线性检索法和 Hash 方法

I. 线性检索法

其又称为顺序检索法，在单级目录中，利用用户提供的文件名，用顺序查找法从文件目录中找到文件的目录项，在树形目录中，用户提供的文件名是由多个文件分量名组成的路径名，此时须对多级目录进行查找，假定用户给定的文件路径名为 /usr/ast/mbox，

根目录	节点6是/usr的目录	132号盘块是/usr的目录	节点26是/usr/ast的目录	496号盘块是/usr的目录
1 .		6 .		26 .
1 ..	132	1 ..	496	6 ..
4 bin		19 dick		64 grants
7 dev		30 erik		92 books
14 lib		51 jim		60 mbox
9 etc		26 ast		81 minik
6 usr		45 bal		17 src

步骤：

- i) 读入第一个文件分量名 `usr`，用它与根目录文件（或当前目录文件）中各目录项中的文件名顺序地进行比较
- ii) 得到匹配项的索引结点号是 6
- iii) 从 6 号索引结点中得到 `usr` 目录文件放在 132 号盘块中，将该盘块内容读入内存
- iv) 再将路径名中的第二个分量名 `ast` 读入，用它与放在 132 号盘块中的第二级目录文件中各目录项的文件名顺序进行比较
- v) 依次类推

II. Hash 方法

系统利用用户提供的文件名并将它转换为文件目录的索引值，再利用该索引值到目录中去查找，这将提高检索速度

数据库 (第44)

student

学号	姓名	性别	院系
501	张三	男	计算机
502	李四	女	英语
503	王二	女	会计
504	赵子	男	通信

记录: 数据库的记录
(501, '张三', '男', '通信')

数据库集合

FILE *fp;

c:\aa\bb\Fan.txt — 文件名字

①

= fopen("c:\aa\bb\Fan.txt", "r");

①

文件：有文件名的一组相关信息的集合。

文件组成：

文件名

文件物理地址。

文件属性：

逻辑结构、物理结构

文件长度

文件存取权限

建立日期、访问时间、日期

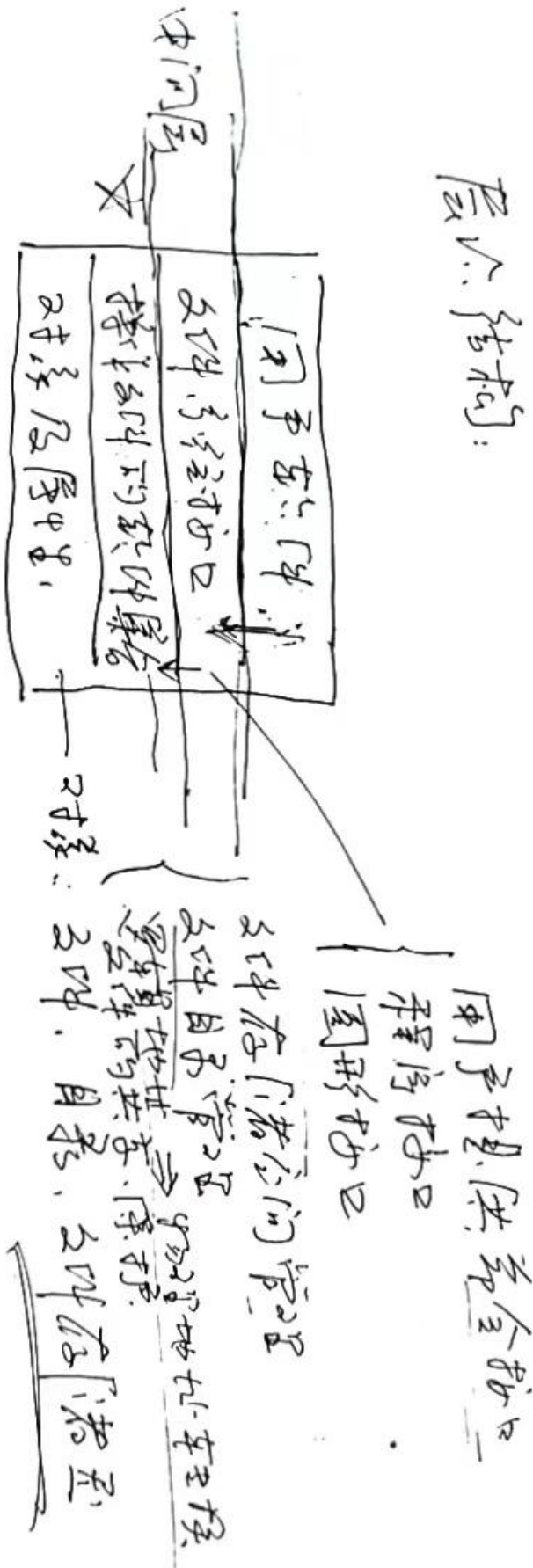
文件存取控制

文件标识符

文件. 系统: OS 的一子集,

—— 管理和控制文件. 系统.

层次结构:



10. 3. 3. 3. 西之石井田

ii) 在 214A 第三-5 号孔 - 在 1:1 米 FCB. 2. 2 号位置

1. 在文件系统中删除该文件
2. 在文件 FCB 及目录项中
根目录 FCB 中记录该文件的地址信息
3. 释放磁盘空间
4. 更新磁盘空闲块表
5. 更新文件目录表
6. 更新文件 FCB 表
7. 更新文件目录表
8. 更新文件 FCB 表
9. 更新文件目录表
10. 更新文件 FCB 表
11. 更新文件目录表
12. 更新文件 FCB 表
13. 更新文件目录表
14. 更新文件 FCB 表
15. 更新文件目录表
16. 更新文件 FCB 表
17. 更新文件目录表
18. 更新文件 FCB 表
19. 更新文件目录表
20. 更新文件 FCB 表
21. 更新文件目录表
22. 更新文件 FCB 表
23. 更新文件目录表
24. 更新文件 FCB 表
25. 更新文件目录表
26. 更新文件 FCB 表
27. 更新文件目录表
28. 更新文件 FCB 表
29. 更新文件目录表
30. 更新文件 FCB 表
31. 更新文件目录表
32. 更新文件 FCB 表
33. 更新文件目录表
34. 更新文件 FCB 表
35. 更新文件目录表
36. 更新文件 FCB 表
37. 更新文件目录表
38. 更新文件 FCB 表
39. 更新文件目录表
40. 更新文件 FCB 表
41. 更新文件目录表
42. 更新文件 FCB 表
43. 更新文件目录表
44. 更新文件 FCB 表
45. 更新文件目录表
46. 更新文件 FCB 表
47. 更新文件目录表
48. 更新文件 FCB 表
49. 更新文件目录表
50. 更新文件 FCB 表
51. 更新文件目录表
52. 更新文件 FCB 表
53. 更新文件目录表
54. 更新文件 FCB 表
55. 更新文件目录表
56. 更新文件 FCB 表
57. 更新文件目录表
58. 更新文件 FCB 表
59. 更新文件目录表
60. 更新文件 FCB 表
61. 更新文件目录表
62. 更新文件 FCB 表
63. 更新文件目录表
64. 更新文件 FCB 表
65. 更新文件目录表
66. 更新文件 FCB 表
67. 更新文件目录表
68. 更新文件 FCB 表
69. 更新文件目录表
70. 更新文件 FCB 表
71. 更新文件目录表
72. 更新文件 FCB 表
73. 更新文件目录表
74. 更新文件 FCB 表
75. 更新文件目录表
76. 更新文件 FCB 表
77. 更新文件目录表
78. 更新文件 FCB 表
79. 更新文件目录表
80. 更新文件 FCB 表
81. 更新文件目录表
82. 更新文件 FCB 表
83. 更新文件目录表
84. 更新文件 FCB 表
85. 更新文件目录表
86. 更新文件 FCB 表
87. 更新文件目录表
88. 更新文件 FCB 表
89. 更新文件目录表
90. 更新文件 FCB 表
91. 更新文件目录表
92. 更新文件 FCB 表
93. 更新文件目录表
94. 更新文件 FCB 表
95. 更新文件目录表
96. 更新文件 FCB 表
97. 更新文件目录表
98. 更新文件 FCB 表
99. 更新文件目录表
100. 更新文件 FCB 表

1254 1254

⇒ 关闭文件 close

打开文件：
读文件的 FCB (文件属性) 同 6 内容。

17/84

与叶厚良的接触

讀之

3. 3. 3 主内存中24bit FCB (2141344) . 3
地址. 启动函数. 将24bit数据读入内存

172nd: 18] ~~18~~ 1-.

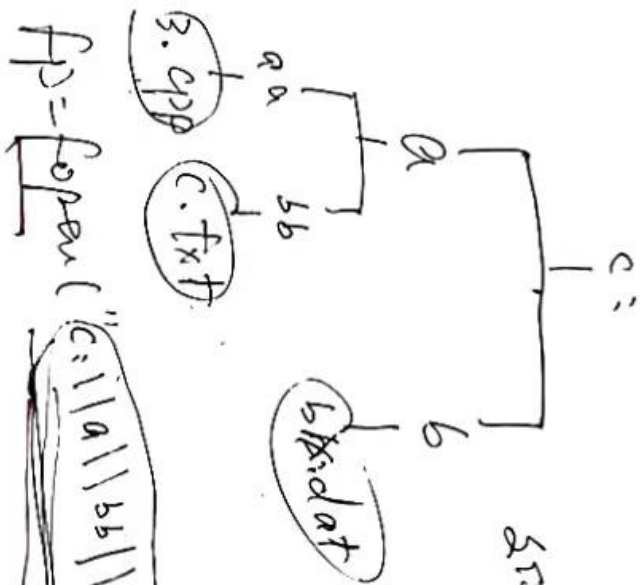
(4)

★ 设置文件的读写指针：

设置要读/写的数据相对于文件为字节偏移量。

关闭文件：

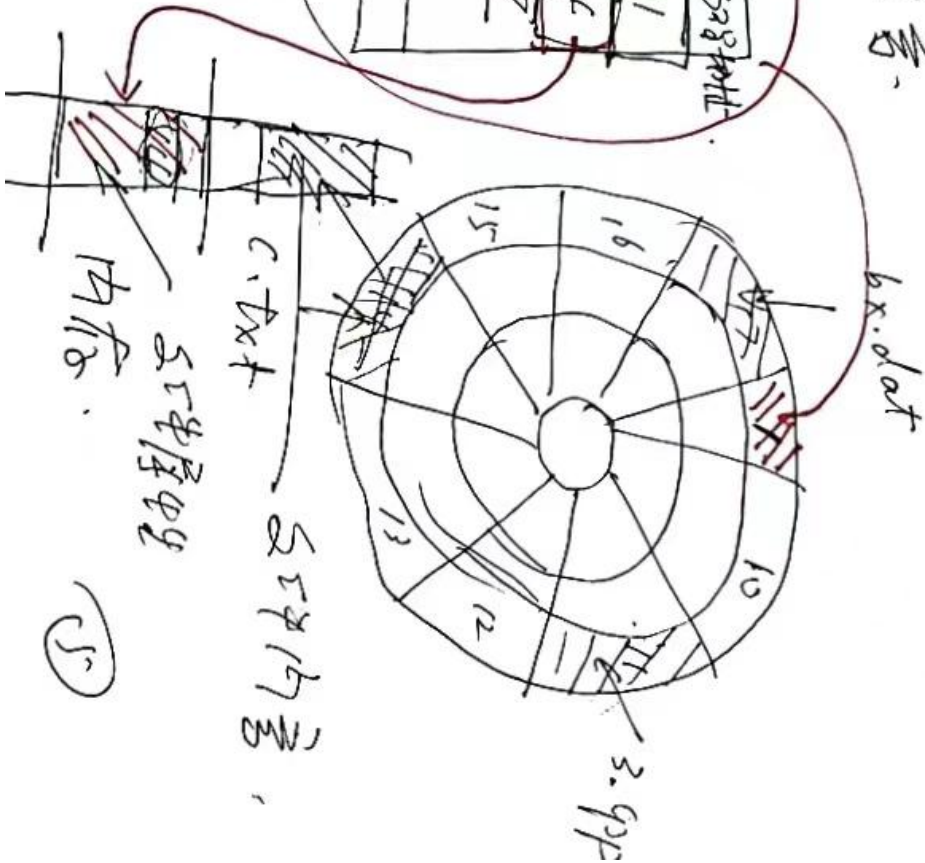
- i> 在内存中删除指定文件的 FCB.
- ii> 若已读及文件数据. 写回内存
- iii> 删除内存中文件内容.



fp = fopen("c:\\a\\b\\c.txt", "r");

文件目录

文件名	文件属性	物理地址
3.cpp	////	11
c.txt	////	14
b.dat	////	17
:		



目录文件

基础信息

① 一级文件 \longleftrightarrow 二级控制块

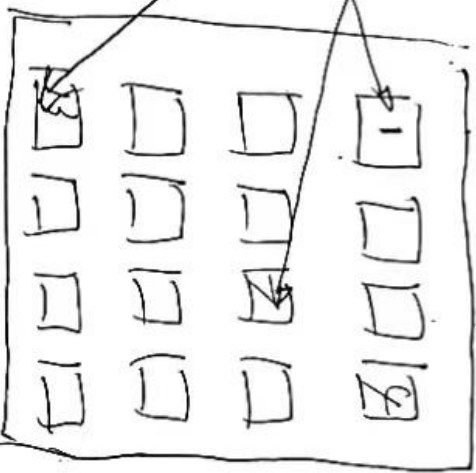
存取控制信息 (权限、共享、属性)

使用信息

② 目录：FCB 信息集合 (目录以文件形式存在于外存——目录文件)
——一级文件下的 FCB

编号	文件名	控制信息	存取控制	使用信息	物理地址
01	aaa.txt	///	///	///	7
02	bbb.txt	///	///	///	13
03	c.dat	///	///	///	4

备注：文件在盘时目录文件信息占据大量磁盘空间，
——读写效率低下。



⑥

改进：在 Unix 3.3 版 (atrous)

将文件名和文件信息分离。

—— 通过索引结查找到文件信息。

目录表 (dentry)

索引结表 (指向文件的 FCB)

第一级地址

天 ⇒

文件名	索引结号
a.txt	1
b.dat	4
c.cup	20
...	...

文件信息	文件的下一级地址
0	1
1	13
2	4
3	
4	
...	...

同时查找目录文件

→ 获得文件的索引结号

→ 通过结号查找到索引结表获得文件的 FCB

目录文件 { 文件名 — 60B
 ⇒ 64B
 索引结点数: — 4B

→ 可存放多少文件?

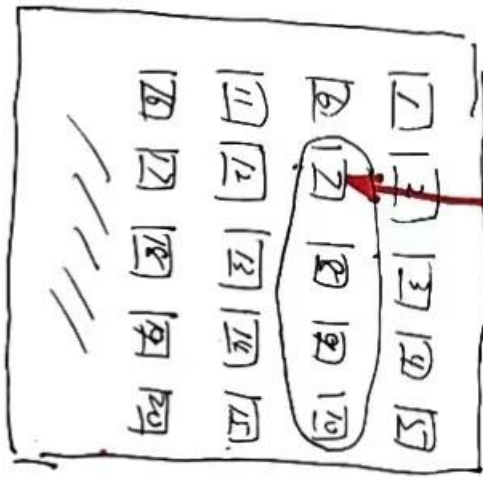
索引结点数 — 4B — 32 bit — 2³² 个

共可表示 2³² 个文件

文件的物理结构

1. 连续分配 —— 将文件数据在磁盘上连续使用连续的磁盘块

有缺陷



文件目录

文件名	索引值
文件1	1
文件2	2
文件3	3

索引值表

文件名称	索引值
文件1	1
文件2	2
文件3	3

起始地址
磁盘号

连续

随机访问

优点: 查找方便

缺点: ① 不易插入、删除、删除

③ 会产生外部碎片

②

不易扩充

⑨

⑩ 连续空间 —— 不连续的不连续块在[磁头]叫。

A> 隐式连续 —— 降低而下一不连续块地址的指针
在的与不连续块中。

目录文件

索引结构为 连续不连续

文件名	索引值
1	1
2	2
3	3
4	4

文件名	索引值
1	1
2	2
3	3
4	4

隐式连续

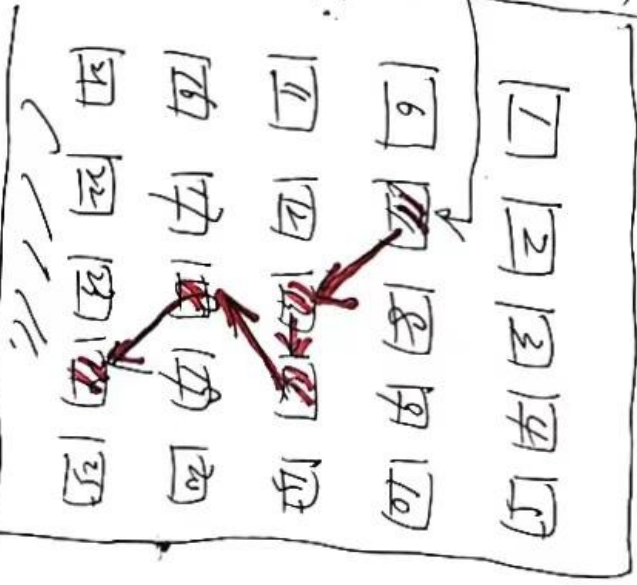
11/12/13/14/15

特点：效率低

优点：方便插入、删除、

和记录长度。

⑩



连续

降低于不连续空间
上操作与效率。

—— 记录长度与[磁头]文件

—— 优点：提高记录效率。

B> 链式结构

FAT (文件分配表) —— 存放每个文件的下一个数据块号

—— 在文件分配表中每一个文件数据所占的数据块构成链式结构

文件 100 个数据块 3, 5, 8, 12

data 在分配表中 7, 12, 14, 18, 24 号

起始块号

1 数据块

1	2	3	4	5
data	2			

1	2	3	4	5

下一个数据块

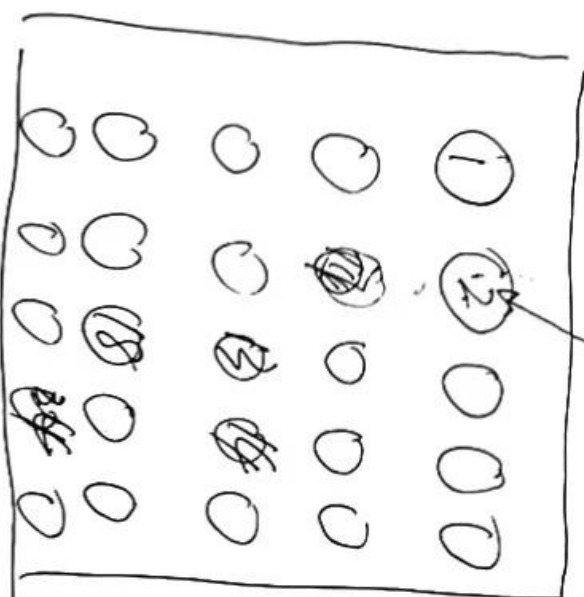
文件	FAT
数据块	下一个数据块号
0	-2
1	-2
2	-2
3	5
4	8
5	
6	
7	
8	12
9	
10	
11	
12	-1
...	
99	

数据块

注意：建立文件系统/0 文件时，成为链式结构

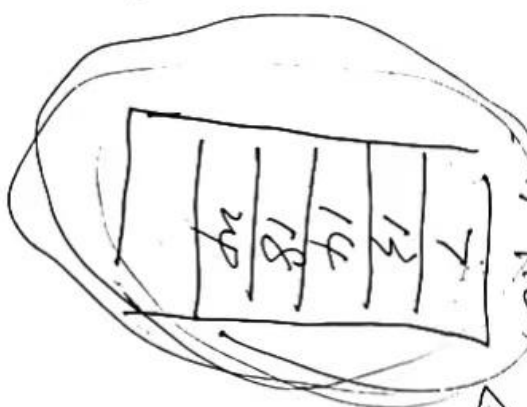
①

③ 4.3/3.42



28/11/15

索引表

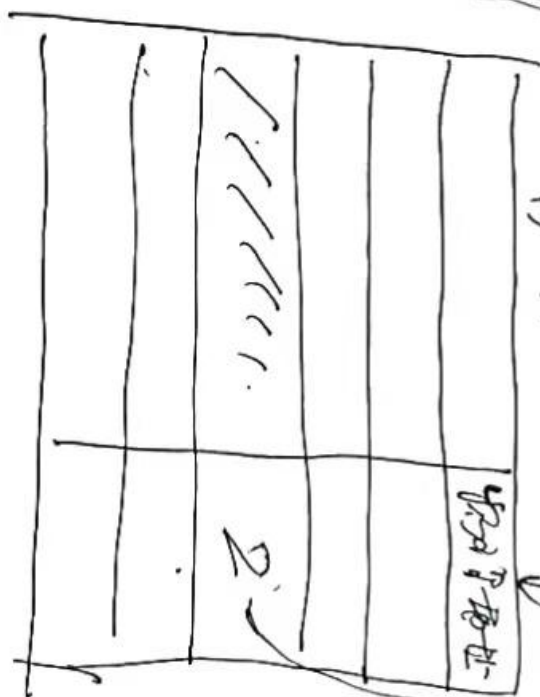


a. $\text{fx} \neq$

Ex. 13. 14 18. 24

索引与目录

李弘亮 敬上



trial

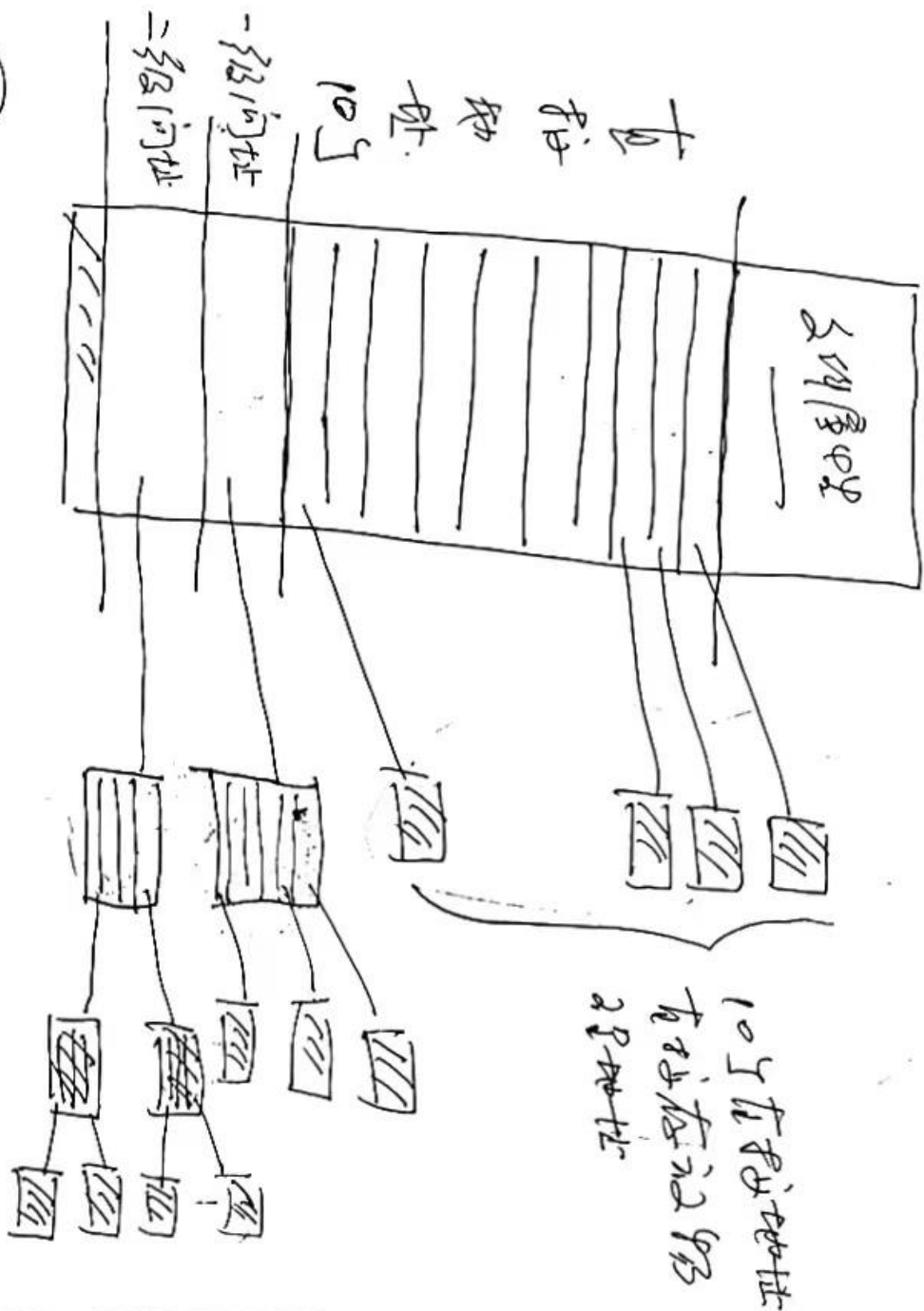
易打忘
方向速度快

未定：不易估计。

⑫

④ 混合索引/分配 (块号式索引)

文件索引结首 —— 13地址项



(13)

例: 一文件需块 4K
一地址项 4B
文件需大地址?

地址: $10 \times 4K = 40K$

地址: $\frac{4K}{4} = 1K = 1024$ 块
[1024 x 4K] 块

地址: $1024 \times 1024 \times 4K$

$40K + 4K \times 1024 + 4K \times 1024 \times 1024$

[40K + 4M + 4G]