

窦轶////yi.dou@njupt.edu.cn

O p e r a t i n g S y s t e m s

操作系统



文件的物理 结构

Linux
Android
Linux
OpenStack
Mac OS
Windows



课程回顾 | 上节课的重点内容

1 文件系统类型

FAT	MS-DOS操作系统使用的文件系统。文件地址以FAT表结构存放，文件名采用8.3格式，即主文件名为8个字符，扩展名为3个字符。
Vfat / FAT32	Windows 95以上版本支持的文件系统，突破了FAT系统中，每一个分区不能超过2GB的限制，增加了对长文件名（最多255个字符）支持。
NTFS	Windows NT/XP/8/10操作系统使用的文件系统。它具有很强的安全性和文件系统恢复功能。
Ext32	Linux操作系统使用的文件系统，也称二级扩展文件系统。支持256字符的文件名，最大可支持到4TB的文件系统。
HPFS	OS/2操作系统使用的文件系统。突破了FAT文件系统的一些限制，极大地改善了文件在大目录下的访问时间，支持2TB硬盘容量。
UDF	是符合ISO9660标准的支持CD-ROM的文件系统。

课程回顾 | 上节课的重点内容

1 逻辑结构

定义

是从用户观点出发，所观察到的文件组织形式，是用户可以处理的数据及其结构，它独立于物理特性，又称为文件组织 (File Organization)。

课程回顾 | 上节课的重点内容

2 逻辑结构

分类

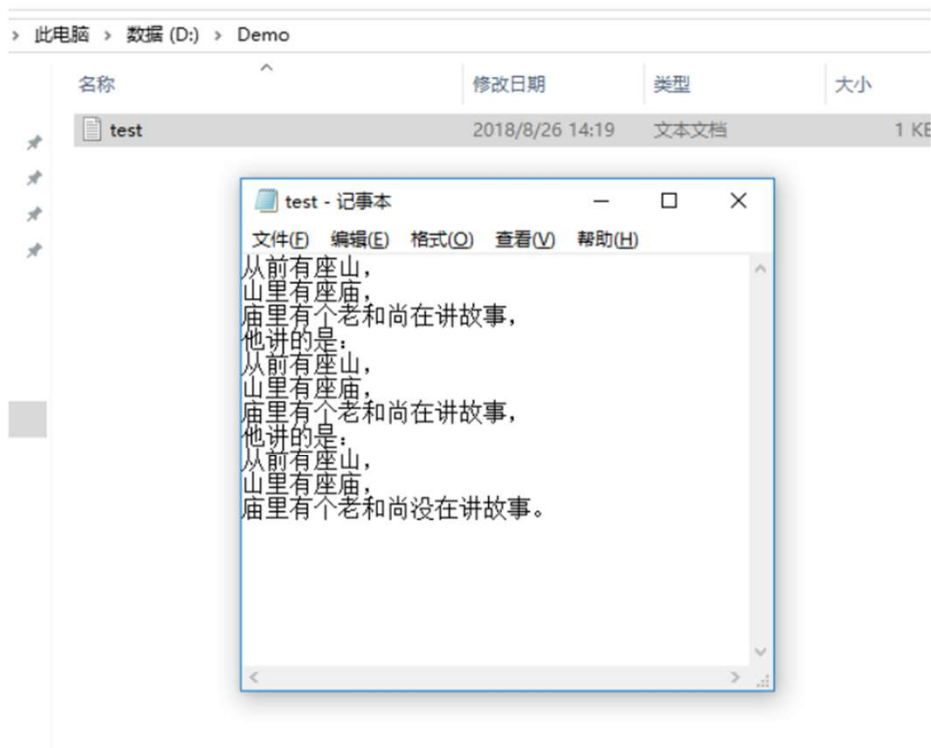
(2) 无结构的流式文件



无结构文件

按文件是否有结构分类，可以分为无结构文件、有结构文件两种。

无结构文件：文件内部的数据就是一系列二进制流或字符流组成。又称“**流式文件**”。如：Windows 操作系统中的 .txt 文件。

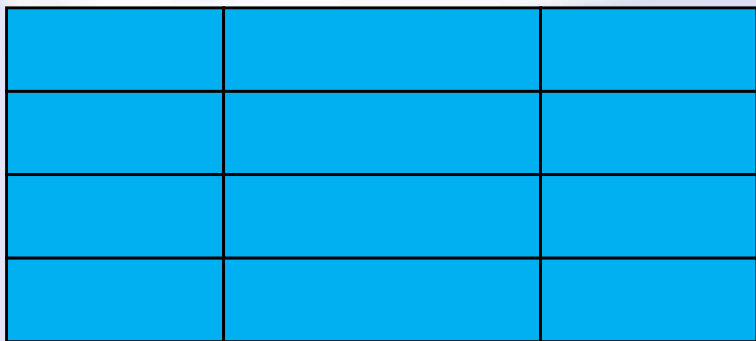


课程回顾 | 上节课的重点内容

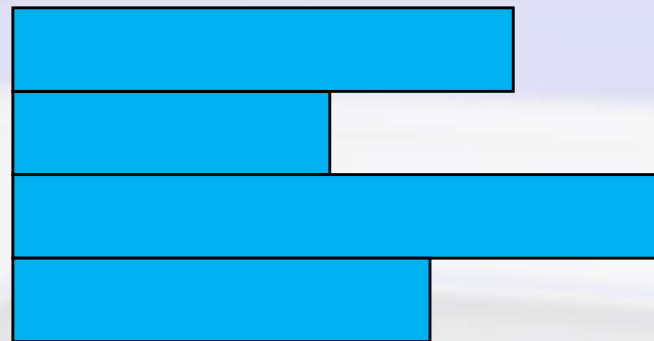
2 逻辑结构

分类

(1) 有结构的记录式文件



定长记录



变长记录

知识总览

文件的逻辑结构

无结构文件

有结构文件

所谓的“逻辑结构”，就是指在用户看来，文件内部的数据应该是如何组织起来的。而“物理结构”指的是在操作系统看来，文件的数据是如何存放在外存中的。

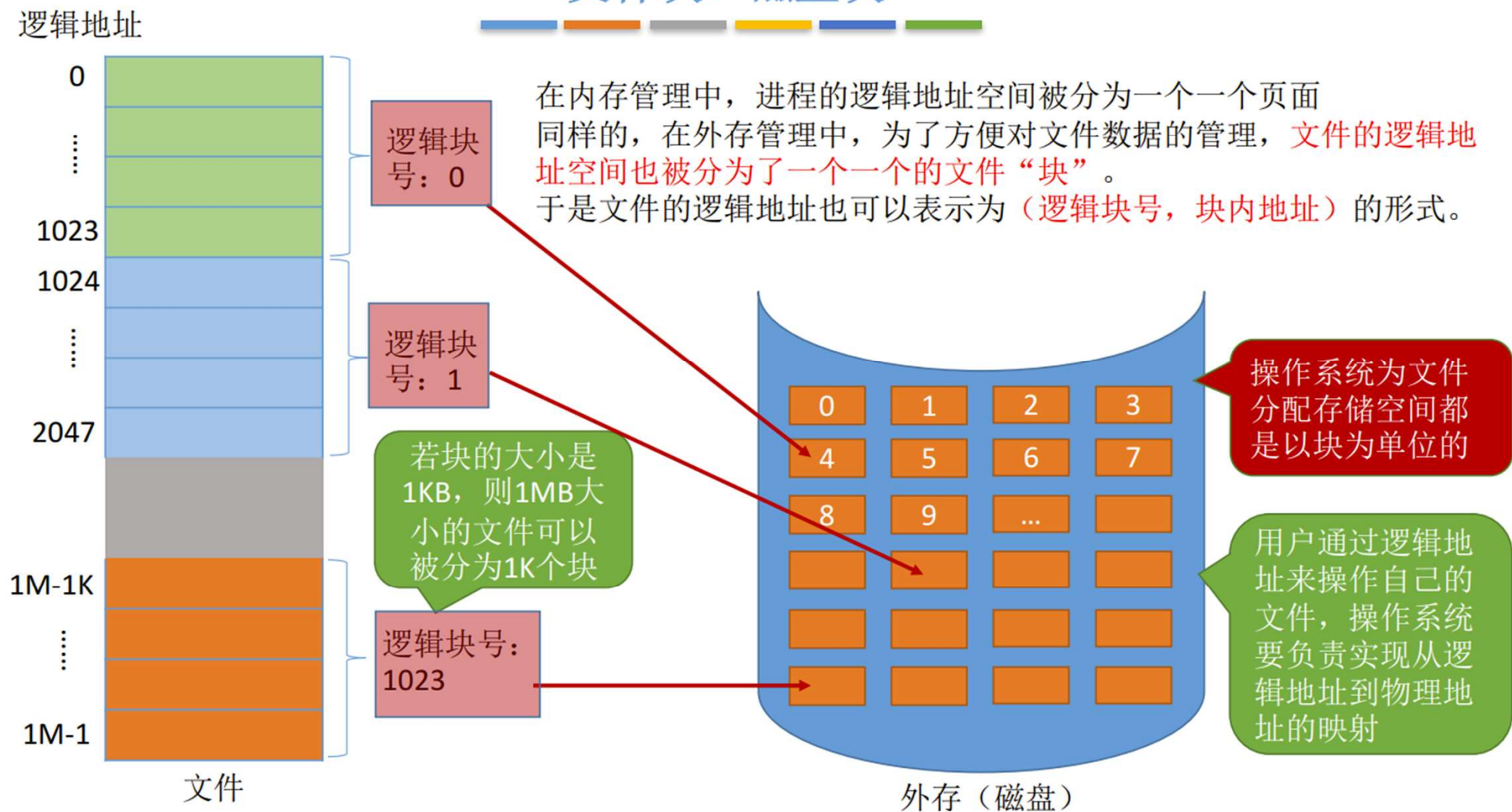
类似于数据结构的“逻辑结构”和“物理结构”。

如“线性表”就是一种逻辑结构，在用户角度来看，线性表就是一组有先后关系的元素序列，如：a, b, c, d, e

“线性表”这种逻辑结构可以用不同的物理结构实现，如：顺序表/链表。顺序表的各个元素在逻辑上相邻，在物理上也相邻；而链表的各个元素在物理上可以是不相邻的。因此，顺序表可以实现“随机访问”，而“链表”无法实现随机访问。

可见，算法的具体实现与逻辑结构、物理结构都有关（文件也一样，文件操作的具体实现与文件的逻辑结构、物理结构都有关）

文件块、磁盘块



逻辑结构（从用户视角看）

每个字符1B。在用户看来，整个文件占用一片连续的逻辑地址空间

H e l l o w o r l d ! H e l l o w o r l d ! H e l l o w o r l d !

Eg: 你要找到第16个字符（编号从0开始）

```
FILE *fp = fopen("test.txt","r"); //以"读"方式打开文件
if( fp == NULL ){
    puts("Fail to open file!");
    exit(0);
}
fseek(fp, 16, SEEK_SET); //读写指针指向16
char c = fgetc(fp); //从读写指针所指位置读出1个字符
printf("字符: %c", c); //打印从文件读出的字符
fclose(fp); //关闭文件
```

逻辑结构（从用户视角看）

每个字符1B。在用户看来，整个文件占用一片连续的逻辑地址空间

H e l l o w o r l d ! H e l l o w o r l d ! H e l l o w o r l d !

Eg: 你要找到第16个字符（编号从0开始）

```
FILE *fp = fopen("test.txt","r"); //以"读"方式打开文件
if( fp == NULL ){
    puts("Fail to open file!");
    exit(0);
}
fseek(fp, 16, SEEK_SET); //读写指针指向16
char c = fgetc(fp); //从读写指针所指位置读出1个字符
printf("字符: %c", c); //打印从文件读出的字符
fclose(fp); //关闭文件
```

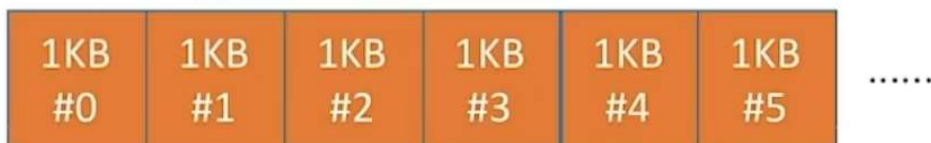
用户用逻辑地址访问文件

```
Run: CLionProject x
"/Users/honglin/L
字符: 0
Process finished
```

物理结构（从操作系统视角看）

Hel lo wor ld ! Hel lo wor ld ! Hel lo wor ld !

操作系统视角：反正就是一堆二进制数据，每个磁盘块可存储1KB，拆就完了！



被操作系统拆分为若干个块，逻辑块号相邻

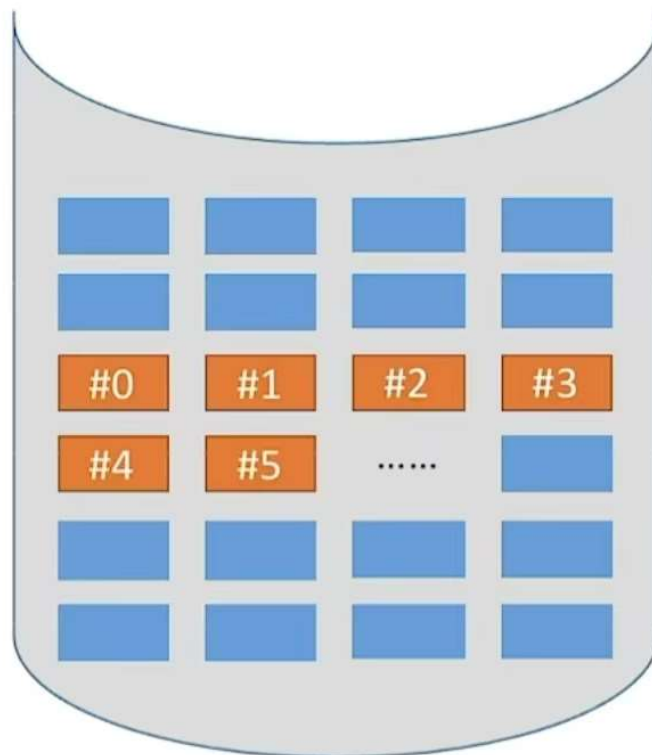
指明逻辑地址

用户：

使用 C语言库函数 `fseek`，将文件读写指针指向位置 `n`

使用 C语言库函数 `fgetc`，从读写指针所指位置读出 1B 内容

`fgetc` 底层使用了 `Read` 系统调用，操作系统将（逻辑块号，块内偏移量）转换为（物理块号，块内偏移量）

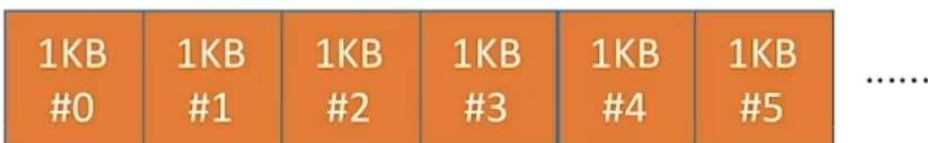


连续分配：逻辑上相邻的块物理上也相邻

物理结构（从操作系统视角看）

H e l l o w o r l d ! H e l l o w o r l d ! H e l l o w o r l d !

操作系统视角：反正就是一堆二进制数据，每个磁盘块可存储1KB，拆就完了！



被操作系统拆分为若干个块，逻辑块号相邻

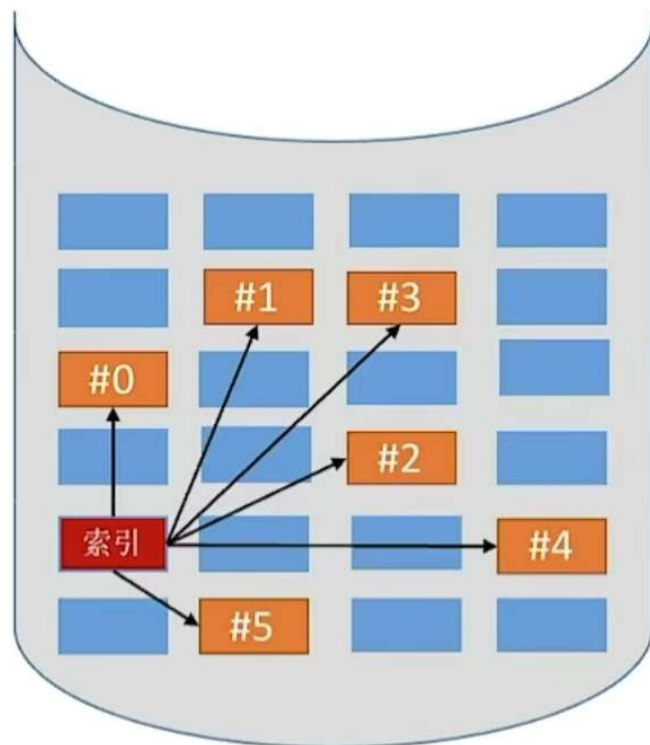
指明逻辑地址

用户：

使用 C语言库函数 `fseek`，将文件读写指针指向位置 `n`

使用 C语言库函数 `fgetc`，从读写指针所指位置读出 1B 内容

`fgetc` 底层使用了 `Read` 系统调用，操作系统将（逻辑块号，块内偏移量）转换为（物理块号，块内偏移量）



索引分配：操作系统为每个文件维护一张索引表，其中记录了逻辑块号→物理块号的映射关系

例：C语言创建顺序文件

```
typedef struct {
    int number;           //学号
    char name[30];       //姓名
    char major[30];     //专业
} Student_info;

//以"写"方式打开文件
FILE *fp = fopen("students.info", "w");
if(fp == NULL) {
    printf("打开文件失败!");
    exit(0);
}
Student_info student[N]; //用数组保存N个学生信息
for(int i = 0; i<N; i++) { //生成 N 个学生信息
    student[i].number=i;
    student[i].name[0]='?';
    student[i].major[0]='?';
}

//将 N 个学生的信息写入文件
fwrite(student, sizeof(Student_info), N, fp);
fclose(fp);
```

有结构文件

例：C语言创建顺序文件

```
typedef struct {
    int number;           //学号
    char name[30];       //姓名
    char major[30];      //专业
} Student_info;

//以"写"方式打开文件
FILE *fp = fopen("students.info", "w");
if(fp == NULL) {
    printf("打开文件失败!");
    exit(0);
}
Student_info student[N]; //用数组保存N个学生信息
for(int i = 0; i<N; i++) { //生成 N 个学生信息
    student[i].number=i;
    student[i].name[0]='?';
    student[i].major[0]='?';
}

//将 N 个学生的信息写入文件
fwrite(student, sizeof(Student_info), N, fp);
fclose(fp);
```

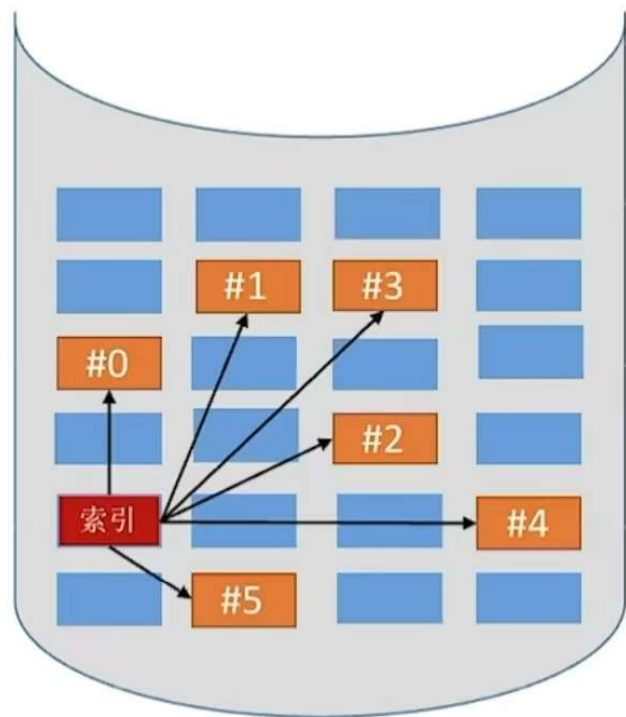
用户视角：
每个学生记录占 64B
sizeof(Student_info)



```
//以"读"方式打开文件
FILE *fp = fopen("students.info", "r");
if(fp == NULL) {
    printf("打开文件失败!");
    exit(0);
}
//文件读写指针指向编号为5的学生记录
fseek(fp, 5*sizeof(Student_info), SEEK_SET);
Student_info stu;
//从文件读出1条记录, 记录大小为 sizeof(Student_info)
fread(&stu, sizeof(Student_info), 1, fp);
printf("学生编号: %d\n", stu.number);
fclose(fp);
```

用户用逻辑地址访问文件

物理结构（从操作系统视角看）

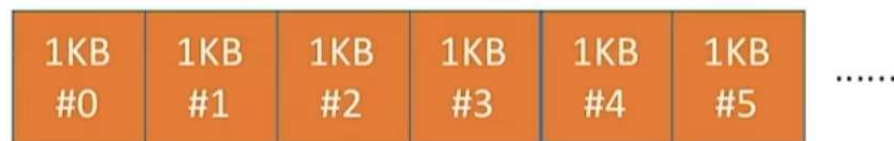


索引分配：操作系统为每个文件维护一张索引表，其中记录了逻辑块号→物理块号的映射关系

用户视角：
每个学生记录占 64B
`sizeof(Student_info)`



操作系统视角：反正就是一堆二进制数据，每个磁盘块可存储1KB，拆就完了！



慢下来消化一下8

逻辑结构 V.S. 物理结构

逻辑结构

用户（文件创建者）的视角看到的亚子

在用户看来，整个文件占用连续的逻辑地址空间

文件内部的信息组织完全由用户自己决定，操作系统并不关心

由操作系统决定文件采用什么物理结构存储

物理结构

操作系统负责将逻辑地址转变为（逻辑块号，块内偏移量）的形式，并负责实现逻辑块号到物理块号的映射



文件的物理结构

本讲内容

1、物理结构含义

2、顺序文件结构

3、链接文件结构

4、索引文件结构

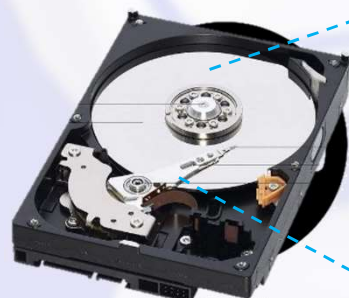
1 物理结构含义



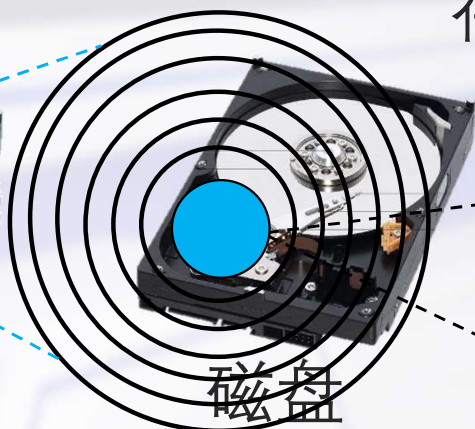
分类

指文件的内部组织形式，即文件在物理存储设备上的存放方法。

逻辑块：512B
存放在一个物理块中



磁带



磁盘

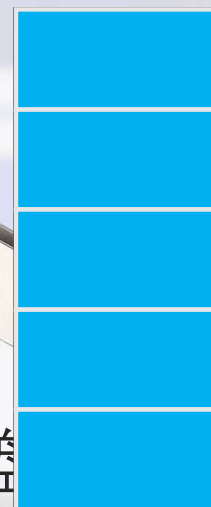


光盘

扇区：一般为600字节，其中512字节存放数据，其余存放控制信息。优盘也称之为物理块。



优盘

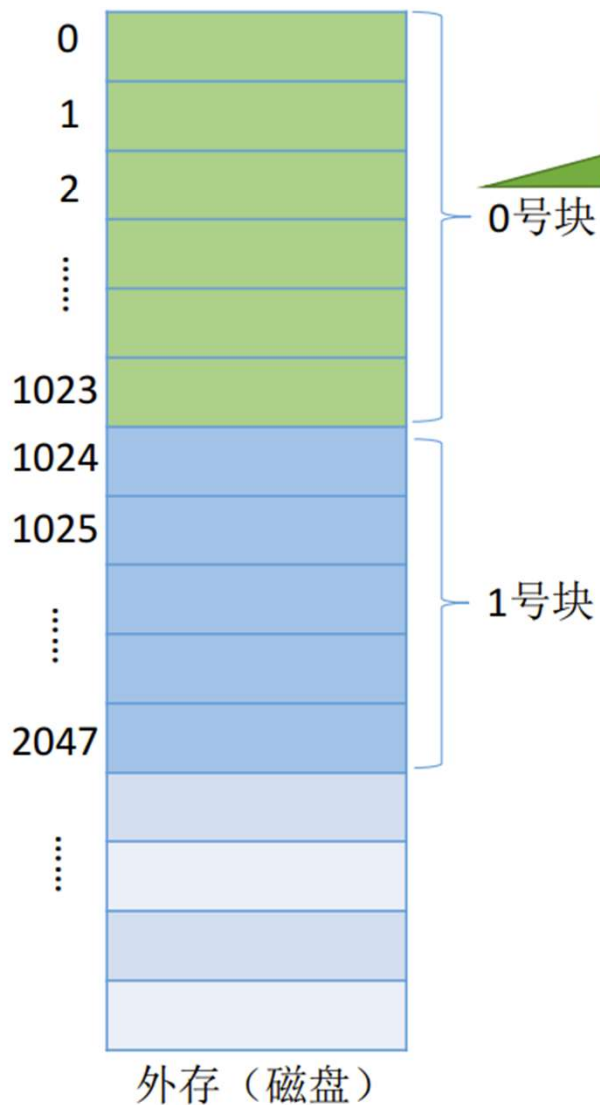


文件

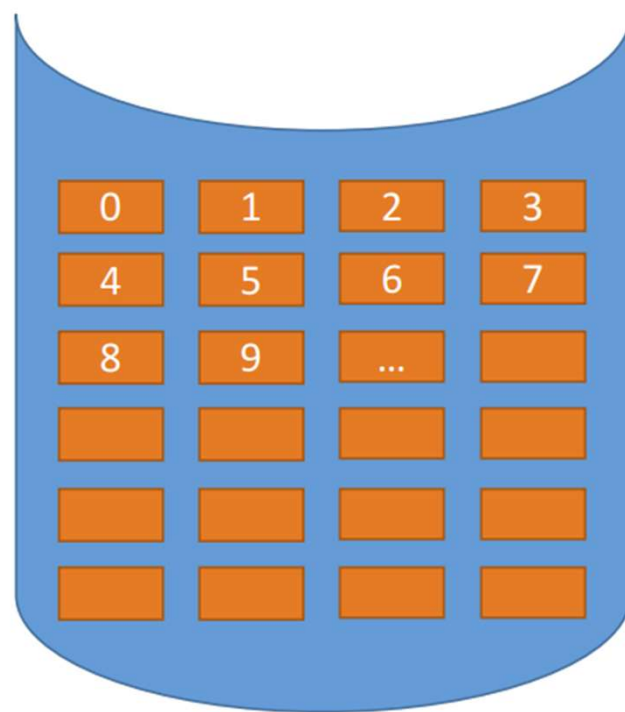
物理块：某个磁道的某个扇区

文件块、磁盘块

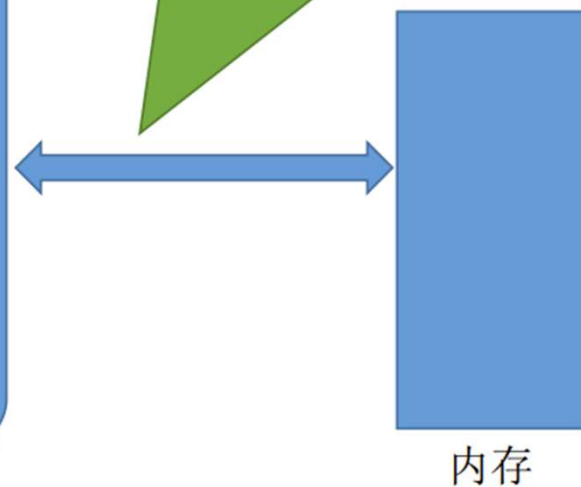
物理地址



类似于内存分页，磁盘中的存储单元也会被分为一个个“块/磁盘块/物理块”。很多操作系统中，磁盘块的大小与内存块、页面的大小相同



内存与磁盘之间的数据交换（即读/写操作、磁盘I/O）都是以“块”为单位进行的。即每次读入一块，或每次写出一块



文件的物理结构

本讲内容

1、物理结构含义

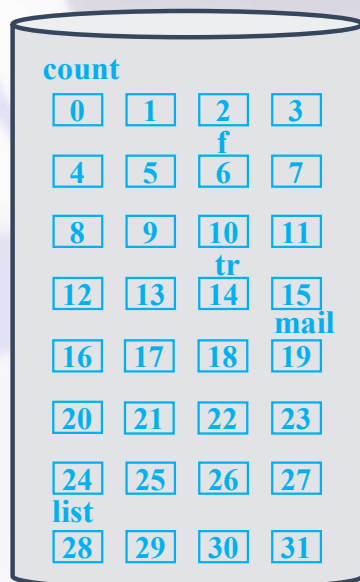
2、顺序文件结构

3、链接文件结构

4、索引文件结构

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

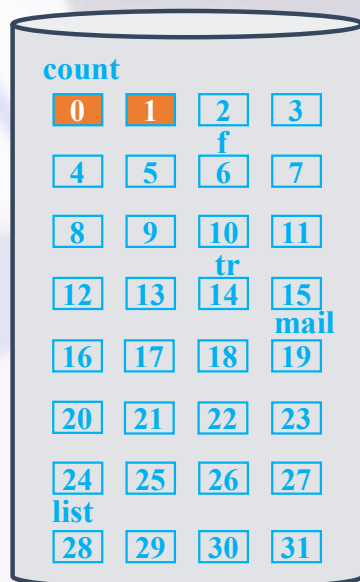


文件目录

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

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。



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

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

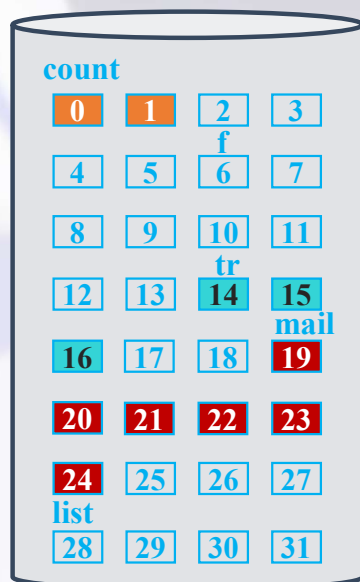


文件目录

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

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

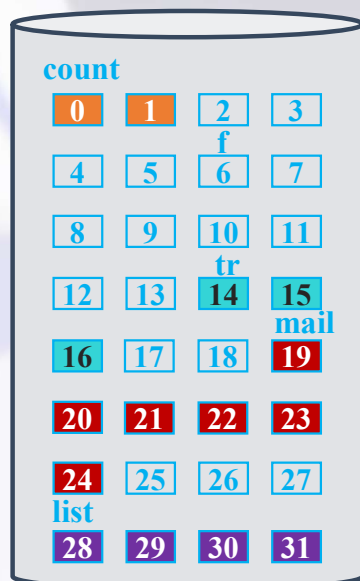


文件目录

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

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

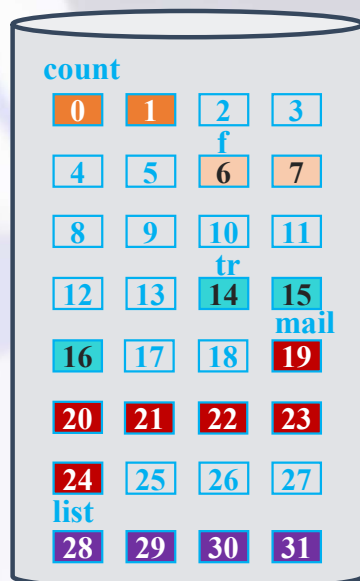


文件目录

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

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。



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

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

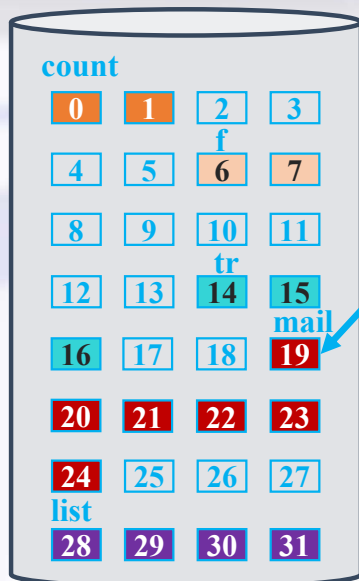


文件名	始址	块
mail	19	6
逻辑块号	物理块号	
0	19	
1	20	
2	21	
3	22	
4	23	
5	24	

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块读取



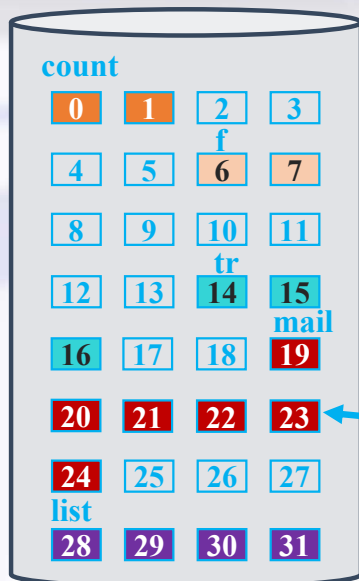
文件名	始址	块
mail	19	6

逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块读取



文件名	始址	块
mail	19	6

逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24

当要访问逻辑块号为4的数据时， $19+4$ 迅速定位到物理块23号就可以去读成功

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块添加



文件名	始址	块
mail	19	6
逻辑块号	物理块号	
0	19	
1	20	
2	21	
3	22	
4	23	
5	24	

X

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块添加



文件名	始址	块
mail	19	6
逻辑块号	物理块号	
0	19	
1	20	
2	21	
3	22	
4	23	
5	24	
6	25	

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块插入



文件名	始址	块
mail	19	6

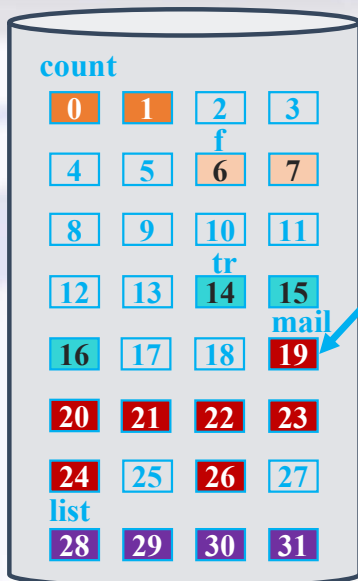
逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24
6	25

X

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块插入



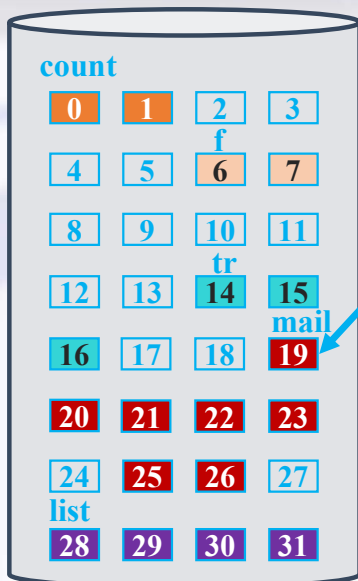
文件名	始址	块
mail	19	6

逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24
6	25

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块插入

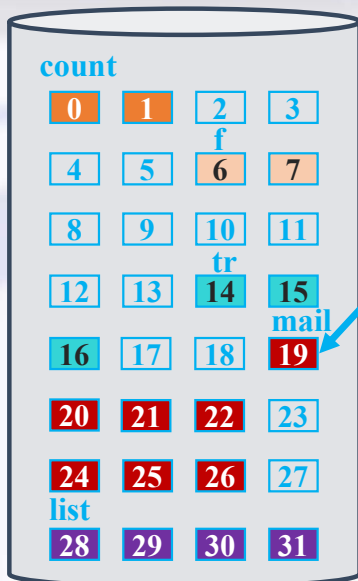


文件名	始址	块
mail	19	6
逻辑块号	物理块号	
0	19	
1	20	
2	21	
3	22	
4	23	
5	24	
6	25	

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块插入



文件名	始址	块
mail	19	6
逻辑块号	物理块号	
0	19	
1	20	
2	21	
3	22	
4	23	
5	24	
6	25	

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块插入



文件名	始址	块
mail	19	6

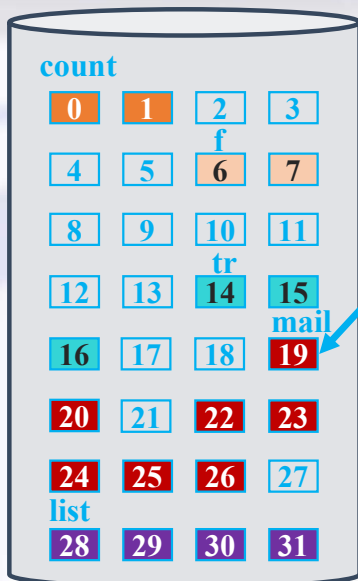
逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24
6	25

X

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块插入



文件名	始址	块
mail	19	6

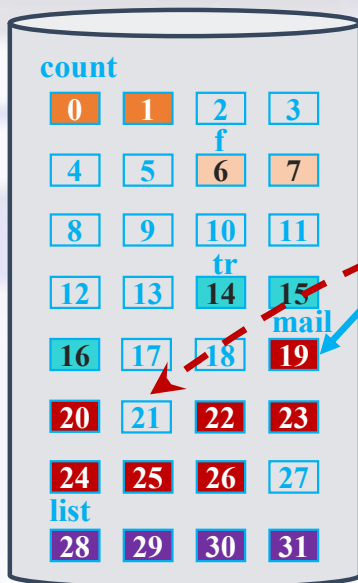
逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24
6	25

X

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块插入



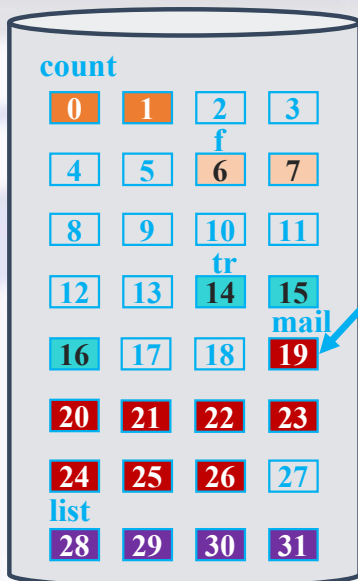
文件名	始址	块
mail	19	6

逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24
6	25

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

数据块插入



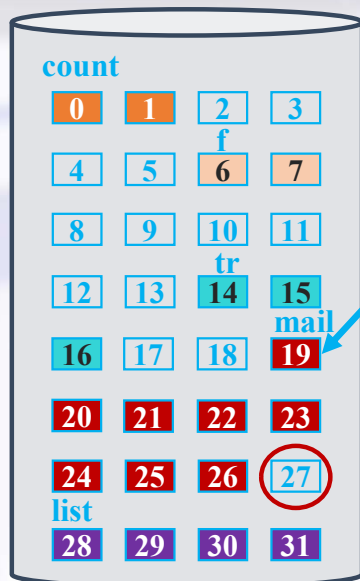
文件名	始址	块
mail	19	6

逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24
6	25
7	26

2 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

扩展性问题



文件名	始址	块
mail	19	6

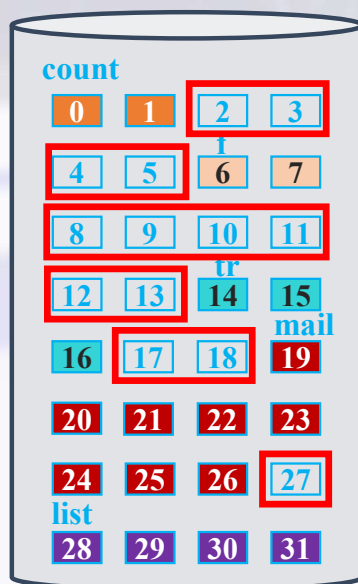
逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24
6	25
7	26

- 由于后面没有空闲的物理块时，导致扩展失败

② 顺序文件结构

将逻辑上连续的文件信息依次存放在编号连续的物理块上。

磁盘碎片



文件名	始址	块
mail	19	6

逻辑块号	物理块号
0	19
1	20
2	21
3	22
4	23
5	24
6	25
7	26

- 碎片虽然总容量很大，但是由于不连续，无法用于顺序存取文件

1、简单叙述顺序结构文件的优缺点。

答：

顺序结构文件的主要优点包括：

- 顺序存取**容易**：访问一个顺序结构文件，只要在目录中找到该文件的第一个物理块号，然后顺序读取下去即可。
- 顺序存取**速度快**：顺序结构文件所占用的物理块是同一磁道，或者是相邻磁道；当要获得一批相邻的记录时，其存取速度在所有的文件结构中是最快的。

顺序结构文件的主要缺点包括：

- 要求**定量的连续存储空间**：顺序结构文件的增加和删除时，为了保证连续性，必须移动大量信息；顺序结构文件结构不宜用来存放经常被增删修改的文件。存放顺序结构文件必须要有连续的空间，容易产生不能使用的小碎片区域。
- 必须**事先估计**文件的长度：要将一个文件装入一个连续的存储区域中，必须事先估计文件的长度，然后根据其大小找到一个足够大的存储区中。因为有的文件是动态增长的，所以这种文件事先估计是很难的。

文件的物理结构

本讲内容

1、物理结构含义

2、顺序文件结构

3、链接文件结构

4、索引文件结构

3 链接文件结构

将离散存放的数据链接成一个整体

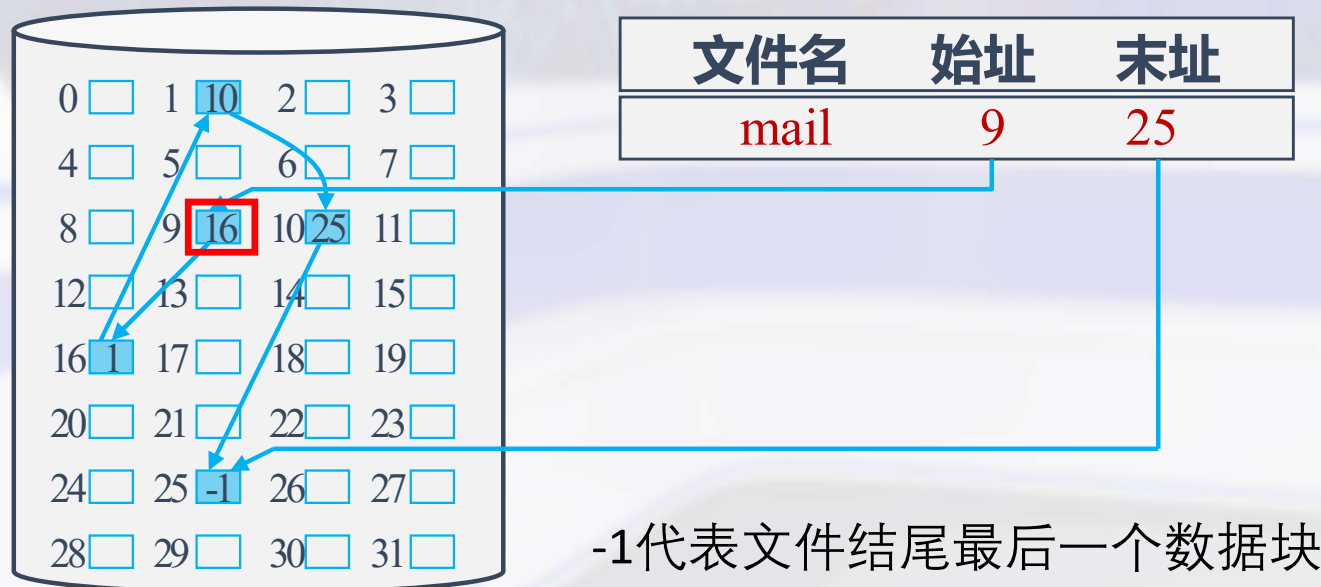
将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个**指针指向**下一个物理块。



3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块读取



物理块上不断存储了数据本身还存储了，下一块数据所在的物理块地址，**接力依次访问**后续的所有数据块，9->16->1->10->25

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块读取

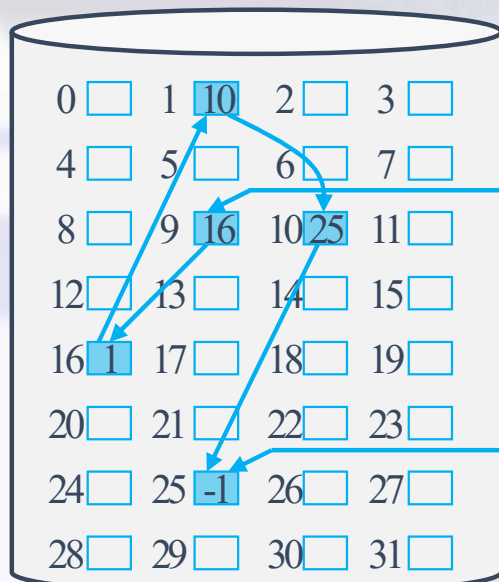


读取逻辑块号为4的数据，不能通过计算，而是从9号物理块开始，经过四次读取定位找到目标数据块，链接文件结构在文件数据块读取方面不如顺序文件结构。

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块添加

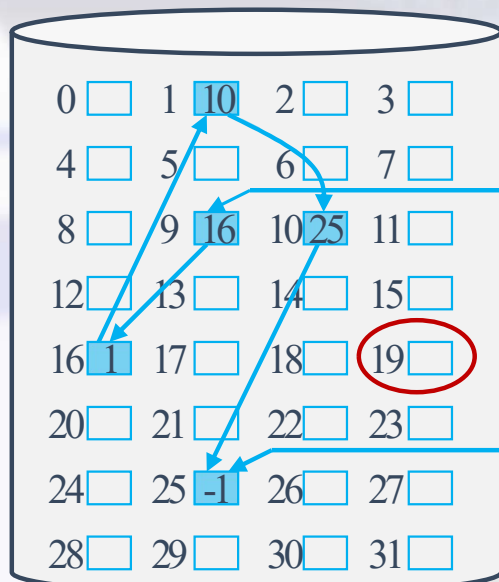


文件名	始址	末址
mail	9	25

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块添加

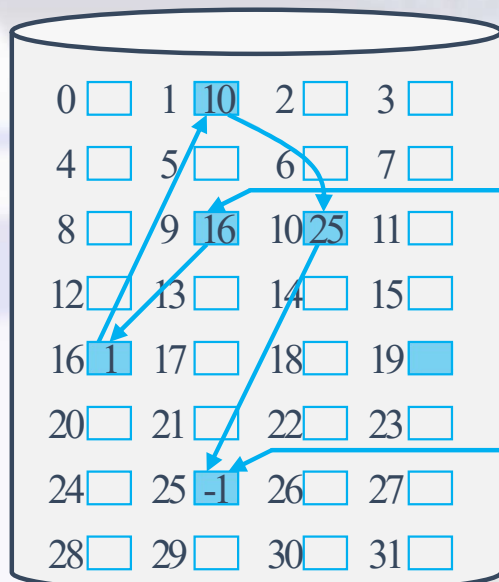


文件名	始址	末址
mail	9	25

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块添加



文件名	始址	末址
mail	9	25

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块插入

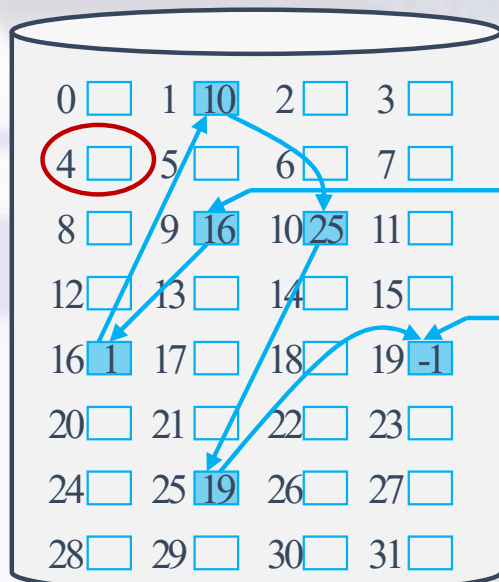


将新插入的19号逻辑块的物理地址，写入到逻辑块25号里面。

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块插入



文件名	始址	末址
mail	9	19

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块插入



在逻辑块号1和2之间插入一个新的数据时，首先申请磁盘上一个空闲的物理块，如4号物理块并将数据写入

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块插入



将逻辑块2的数据物理地址1写入4号块内容中

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块插入



然后将新插入的逻辑块2的地址4写入，前驱逻辑块1的数据中

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

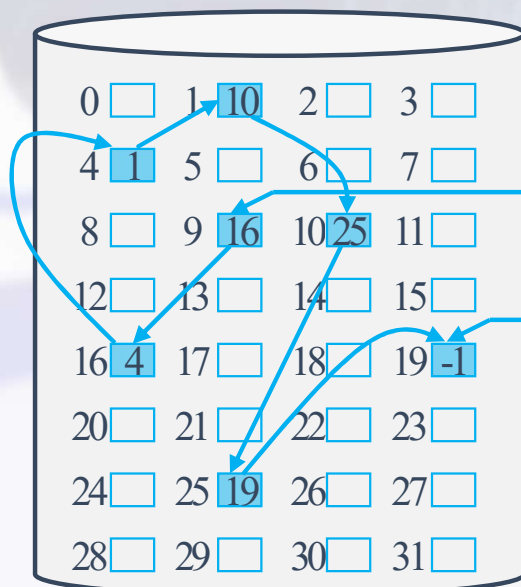
数据块插入



3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

数据块插入



文件名	始址	末址
mail	9	19

链接文件结构不需要文件在磁盘上连续存放，只要磁盘空闲碎片总和容量足够大即可实现文件的存取。

新插入的数据块就融入了文件mail中

3 链接文件结构

将逻辑上连续的文件信息存放在不连续的物理块上，每个物理块设有一个指针指向下一个物理块。

可靠性问题



随机读取性能差，可靠性差

第二个逻辑块链接错误，导致后续所有链接错误

1、文件物理结构是文件（ A ）的文件组织形式。

- A. 在外部设备上
- B. 从用户观点看
- C. 虚拟存储
- D. 目录

2、（ A ）适合存放在磁带上。

- A. 顺序文件
- B. Hash文件
- C. 索引文件
- D. 串联文件

3、顺序存取速度最快的文件物理结构是（ 顺序结构 ）文件，不适宜直接存取的文件物理结构是（ 链接结构 ）文件。

文件的物理结构

本讲内容

1、物理结构含义

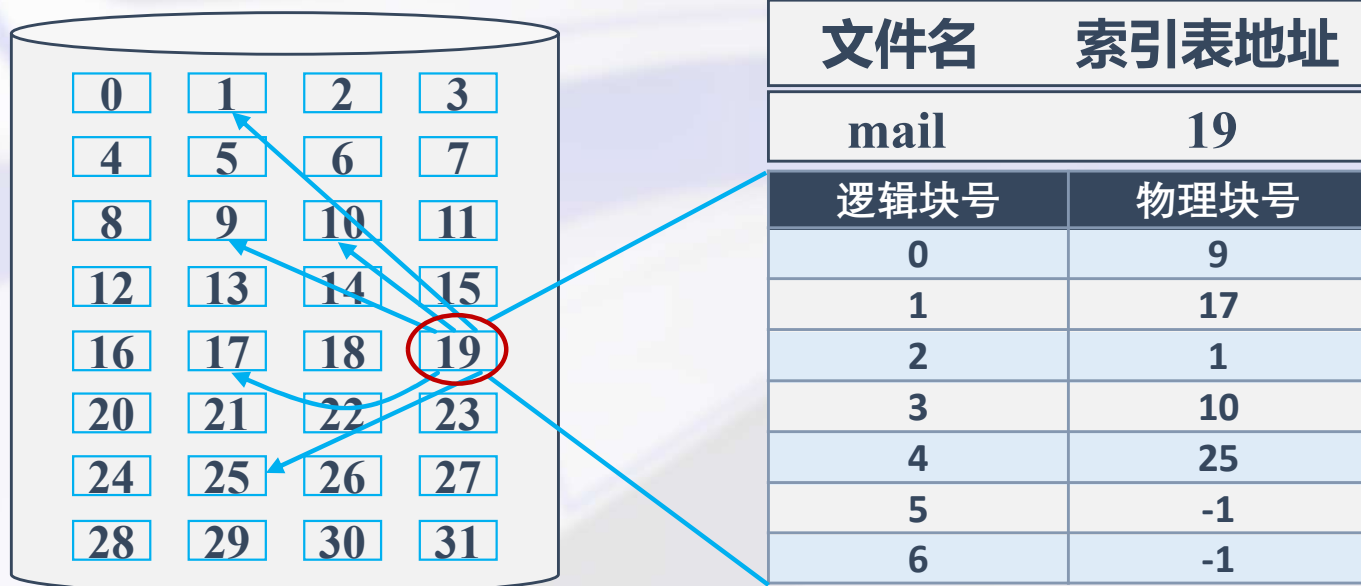
2、顺序文件结构

3、链接文件结构

4、索引文件结构

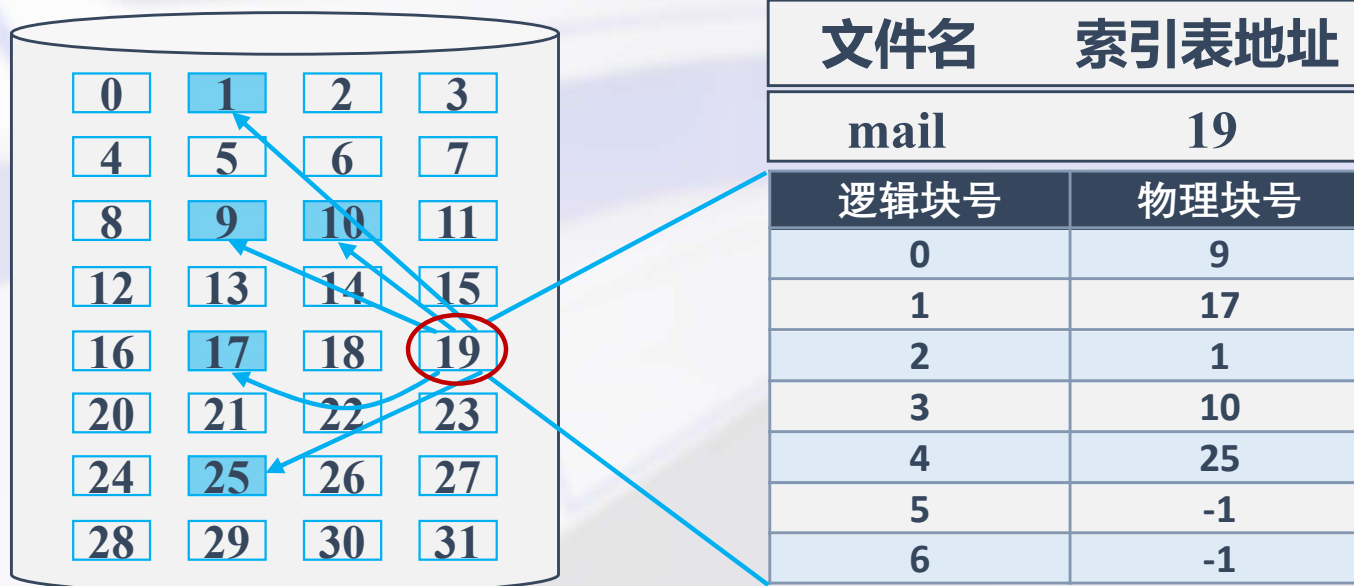
4 索引文件结构

将逻辑上连续的文件信息(记录)存放在不连续的物理块中，系统**为每个文件建立索引表**，存放文件的逻辑块号和物理块号的对应关系。



4 索引文件结构

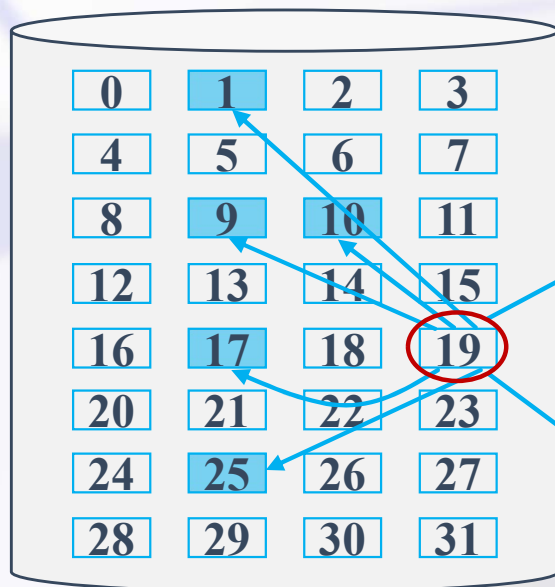
通过直接查询**索引表**，可以快速定位所有数据的物理块地址，而不必像链接存储中通过接力来定位所需的物理块。



4 索引文件结构

将逻辑上连续的文件信息(记录)存放在不连续的物理块中，系统为每个文件建立索引表，存放文件的逻辑块号和物理块号的对应关系。

数据块插入



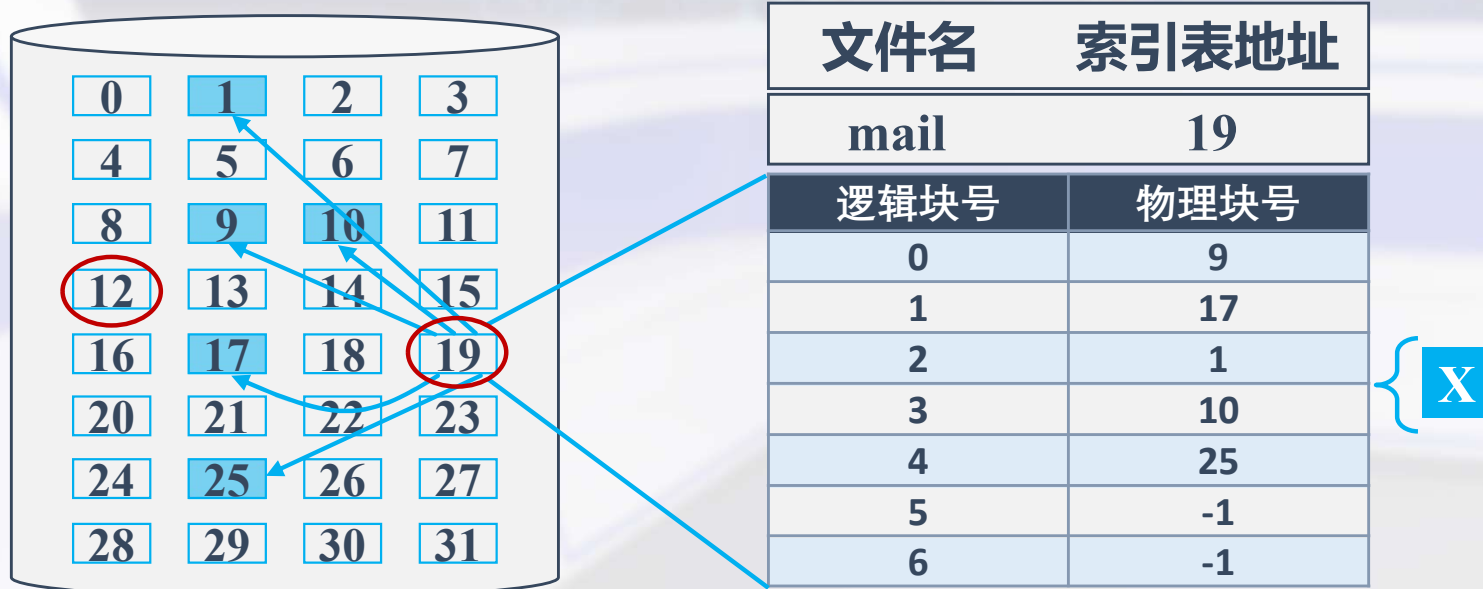
文件名	索引表地址
mail	19
逻辑块号	物理块号
0	9
1	17
2	1
3	10
4	25
5	-1
6	-1

X

4 索引文件结构

将逻辑上连续的文件信息(记录)存放在不连续的物理块中，系统为每个文件建立索引表，存放文件的逻辑块号和物理块号的对应关系。

数据块插入

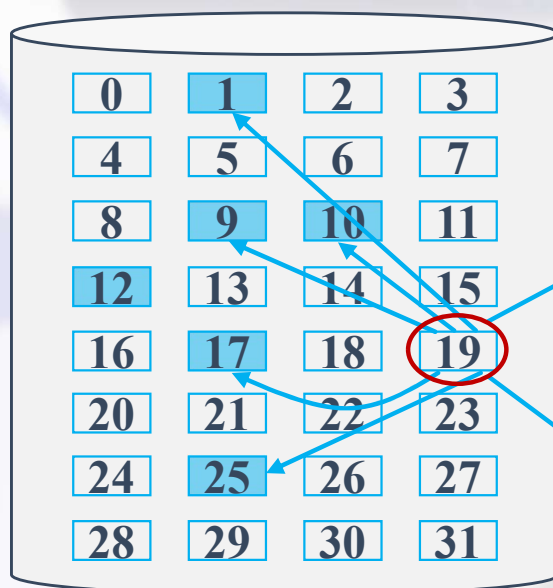


例: 在逻辑块2和3之间插入一个数据 X 时，首先申请磁盘上的一个空闲物理块12。

4 索引文件结构

将逻辑上连续的文件信息(记录)存放在不连续的物理块中，系统为每个文件建立索引表，存放文件的逻辑块号和物理块号的对应关系。

数据块插入



文件名	索引表地址
mail	19
逻辑块号	物理块号
0	9
1	17
2	1
3	10
4	25
5	-1
6	-1

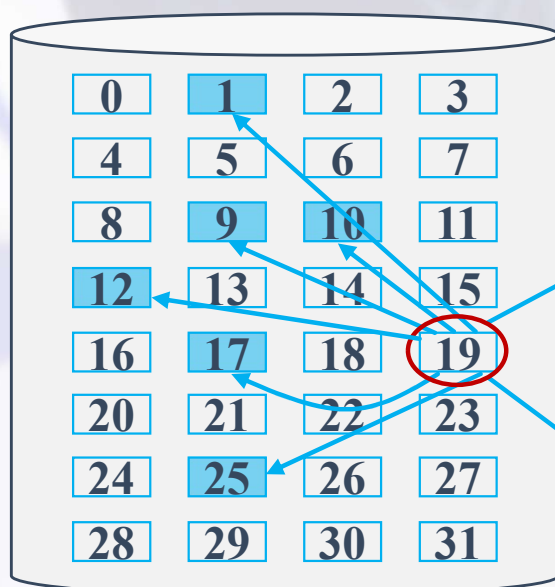
X

将数据写入12号物理块中，然后修改索引表。

4 索引文件结构

将逻辑上连续的文件信息(记录)存放在不连续的物理块中，系统为每个文件建立索引表，存放文件的逻辑块号和物理块号的对应关系。

数据块插入



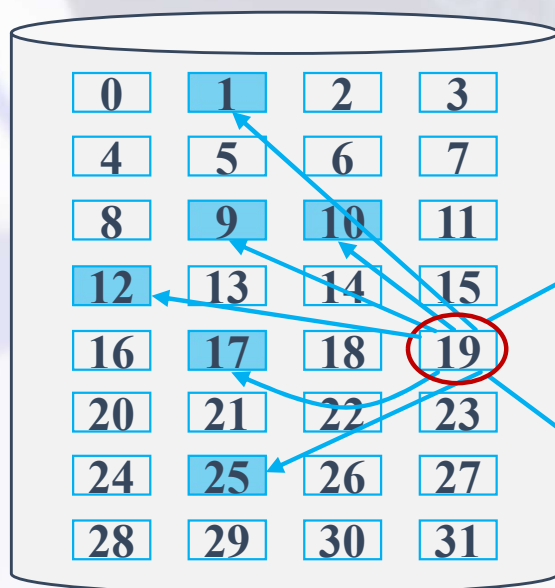
文件名	索引表地址
mail	19
逻辑块号	物理块号
0	9
1	17
2	1
3	12
4	10
5	25
6	-1

将12号物理块填入索引表，就是在索引表中增加一条记录，这样插入的数据块就融入文件mail中。

4 索引文件结构

将逻辑上连续的文件信息(记录)存放在不连续的物理块中，系统为每个文件建立索引表，存放文件的逻辑块号和物理块号的对应关系。

数据块删除



文件名	索引表地址
mail	19
逻辑块号	物理块号
0	9
1	17
2	1
3	12
4	10
5	25
6	-1

删除文件中的某个记录块，只需要在索引表中删除对应的一条记录就可以了。

总结、思考与拓展学习

(1) 分析总结



性能	顺序结构	链接结构	索引结构
读取性能	顺序读取、随机读取速度高	顺序读取速度一般，随机读取速度差	顺序读取速度一般，随机读取速度高
插入、删除性能	插入、删除开销大	插入、删除开销低	插入、删除开销低
扩展性	扩展性受限	扩展性好	扩展性好
资源利用率	资源利用率差，存在碎片问题	资源利用好，解决了碎片问题	资源利用好，解决了碎片问题
可靠性	可靠性强	链接问题，可靠性受到影响	可靠性强

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（ **B** ）。

- A. 顺序结构
- B. 索引结构
- C. 链接结构
- D. 串联结构

2、文件的链式结构的缺点是（ **C** ）。

- A. 不便于动态增、删
- B. 必须连续分配物理块
- C. 不便于直接存取
- D. 必须事先提出文件的最大长度

3、下面不适合于直接存取的外存分配方式是（ **B** ）。

- A. 连续分配
- B. 链接分配
- C. 索引分配
- D. A~C都适合

4、按文件的逻辑存储结构划分，文件分为有结构文件和无结构文件。其中，有结构文件又称为记录式文件；无结构文件又称为流式文件。

文件分配方式——索引分配



索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块（索引表的功能类似于内存管理中的页表——建立逻辑页面到物理页之间的映射关系）。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

文件分配方式——索引分配

索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块（索引表的功能类似于内存管理中的页表——建立逻辑页面到物理页之间的映射关系）。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

假设某个新创建的文件“aaa”的数据依次存放在磁盘块 2 → 5 → 13 → 9。

逻辑块号	物理块号
0	2
1	5
2	13
3	9

文件“aaa”的索引表

文件分配方式——索引分配

索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块（索引表的功能类似于内存管理中的页表——建立逻辑页面到物理页之间的映射关系）。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

假设某个新创建的文件“aaa”的数据依次存放在磁盘块 2 → 5 → 13 → 9。7号磁盘块作为“aaa”的索引块，索引块中保存了索引表的内容。

逻辑块号	物理块号
0	2
1	5
2	13
3	9

文件“aaa”的索引表

文件分配方式——索引分配

索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块（索引表的功能类似于内存管理中的页表——建立逻辑页面到物理页之间的映射关系）。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

文件名	...	索引块
aaa	...	7
bbb	...	23

目录中需要记录文件的索引块是几号磁盘块

假设某个新创建的文件“aaa”的数据依次存放在磁盘块 2 → 5 → 13 → 9。7号磁盘块作为“aaa”的索引块，索引块中保存了索引表的内容。

文件“aaa”所在目录的FCB

逻辑块号	物理块号
0	2
1	5
2	13
3	9

文件“aaa”的索引表

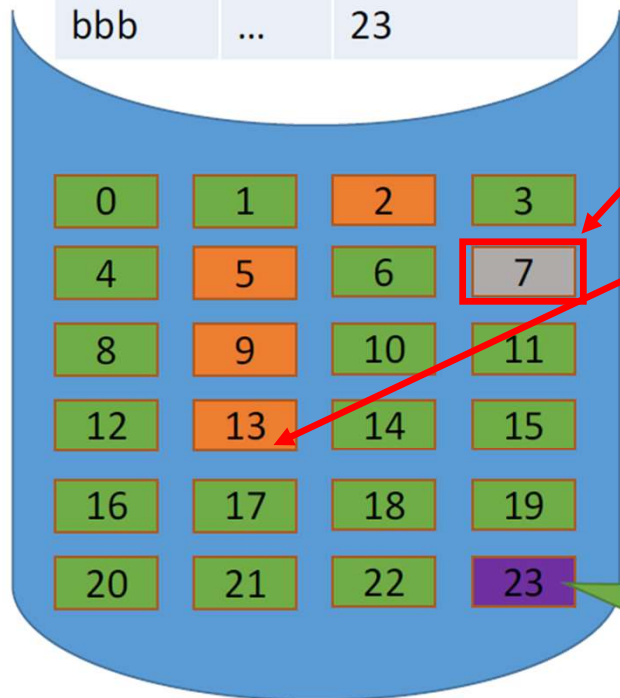
根据索引块号7找到索引表

文件分配方式——索引分配

索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块（索引表的功能类似于内存管理中的页表——建立逻辑页面到物理页之间的映射关系）。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

文件名	...	索引块
aaa	...	7
bbb	...	23

目录中需要记录文件的索引块是几号磁盘块



逻辑块号	物理块号
0	2
1	5
2	13
3	9

文件“aaa”的索引表

假设某个新创建的文件“aaa”的数据依次存放在磁盘块 2 → 5 → 13 → 9。7号磁盘块作为“aaa”的索引块，索引块中保存了索引表的内容。

文件分配方式——索引分配

索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块（索引表的功能类似于内存管理中的页表——建立逻辑页面到物理页之间的映射关系）。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

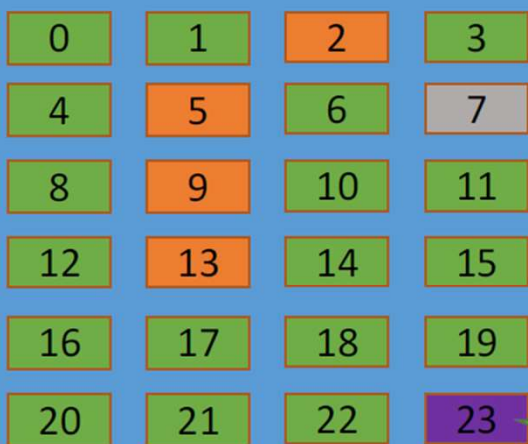
文件名	...	索引块
aaa	...	7
bbb	...	23

目录中需要记录文件的索引块是几号磁盘块

假设某个新创建的文件“aaa”的数据依次存放在磁盘块 2 → 5 → 13 → 9。7号磁盘块作为“aaa”的索引块，索引块中保存了索引表的内容。

逻辑块号	物理块号
0	2
1	5
2	13
3	9

文件“aaa”的索引表



类似的，文件“bbb”的索引块是23号磁盘块，其中存放了文件“bbb”的索引表

文件分配方式——索引分配

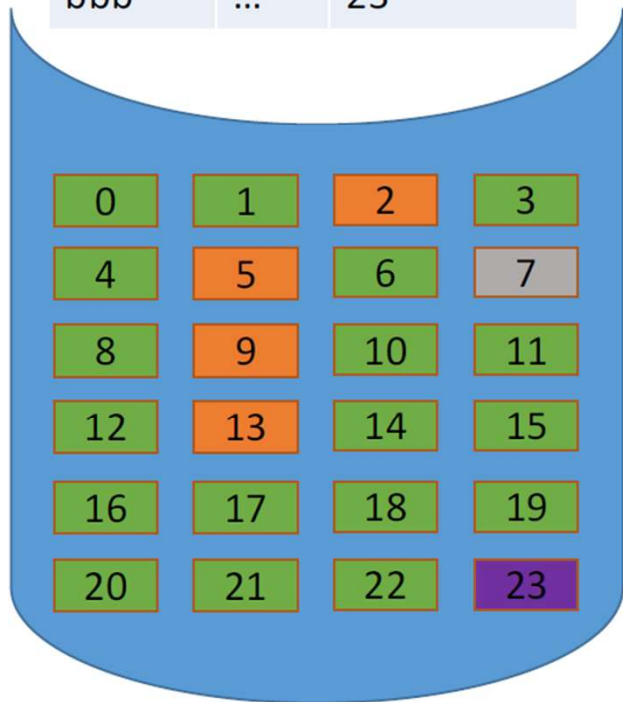
索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

文件名	...	索引块
aaa	...	7
bbb	...	23

目录中需要记录文件的索引块是几号磁盘块



如何实现文件的逻辑块号到物理块号的转换？



逻辑块号	物理块号
0	2
1	5
2	13
3	9
4	19

文件“aaa”的索引表

文件分配方式——索引分配

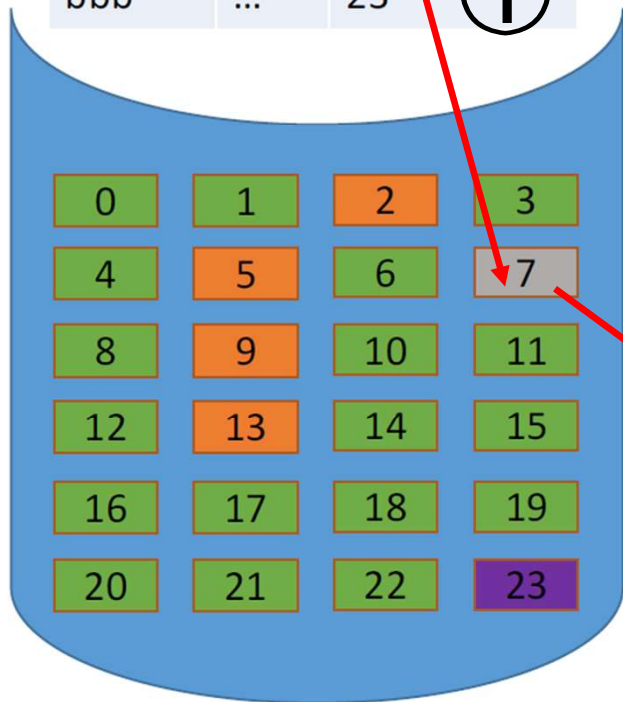
索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

文件名	...	索引块
aaa	...	7
bbb	...	23

目录中需要记录文件的索引块是几号磁盘块



如何实现文件的逻辑块号到物理块号的转换？



逻辑块号	物理块号
0	2
1	5
2	13
3	9
4	19

文件“aaa”的索引表

用户给出要访问的逻辑块号 i ，操作系统找到该文件对应的目录项（FCB）...

文件分配方式——索引分配

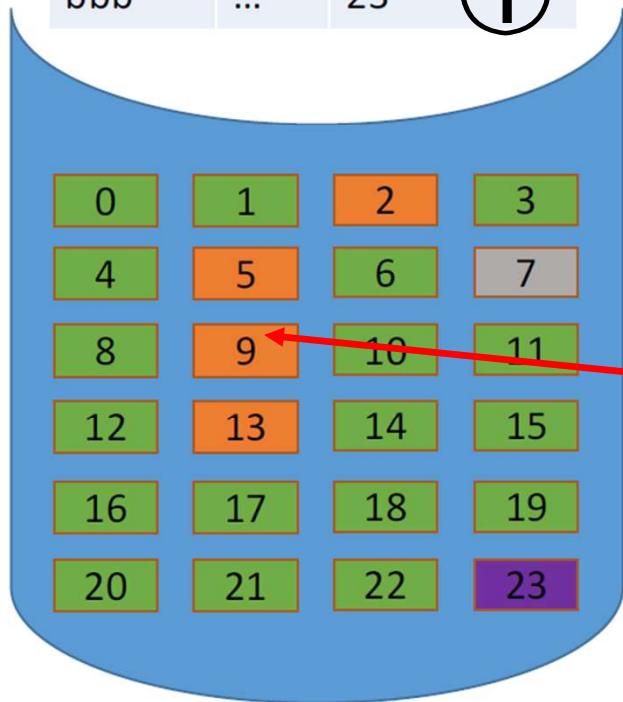
索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

文件名	...	索引块
aaa	...	7
bbb	...	23

目录中需要记录文件的索引块是几号磁盘块



如何实现文件的逻辑块号到物理块号的转换？



逻辑块号	物理块号
0	2
1	5
2	13
3	9
4	19

文件“aaa”的索引表

②

用户给出要访问的逻辑块号 i ，操作系统找到该文件对应的目录项（FCB）...

从目录项中可知索引表存放位置，将索引表从外存读入内存，并查找索引表即可知 i 号逻辑块在外存中的存放位置。

文件分配方式——索引分配

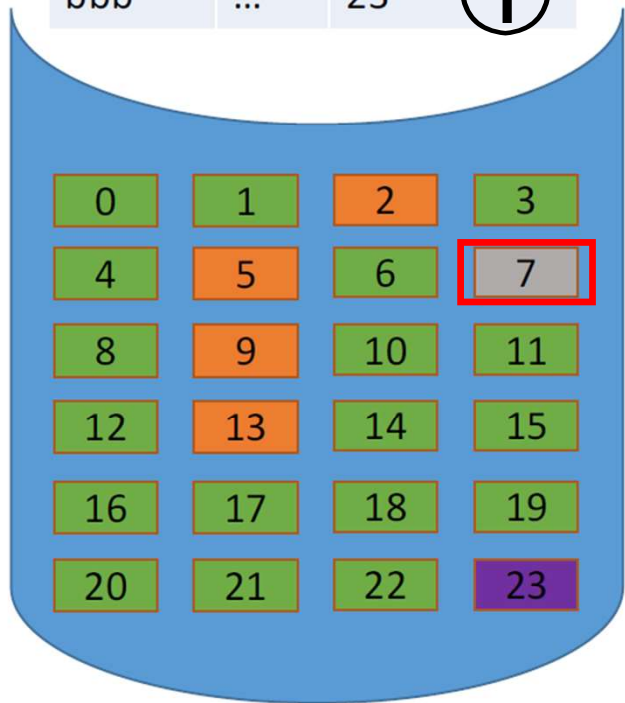
索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

文件名	...	索引块
aaa	...	7
bbb	...	23

目录中需要记录文件的索引块是几号磁盘块



如何实现文件的逻辑块号到物理块号的转换？



逻辑块号	物理块号
0	2
1	5
2	13
3	9
4	19

文件“aaa”的索引表

用户给出要访问的逻辑块号 i ，操作系统找到该文件对应的目录项（FCB）...

从目录项中可知索引表存放位置，将索引表从外存读入内存，并查找索引表即可只 i 号逻辑块在外存中的存放位置。

可见，索引分配方式可以支持随机访问。文件拓展也很容易实现（只需要给文件分配一个空闲块，并增加一个索引表项即可）

但是索引表需要占用一定的存储空间

②

文件分配方式——索引分配

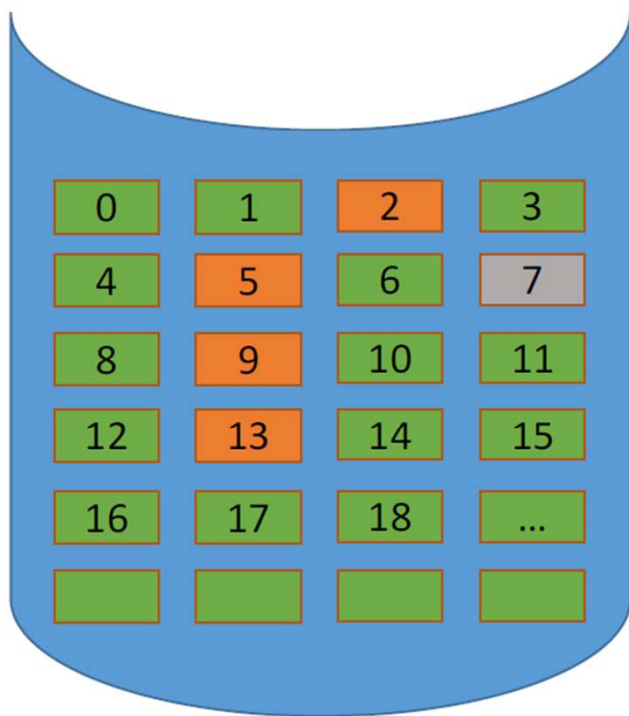
索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

文件名	...	索引块
aaa	...	7

目录中需要记录文件的索引块是几号磁盘块



若每个磁盘块1KB，一个索引表项4B，则一个磁盘块只能存放 256 个索引项。



逻辑块号	物理块号
0	2
1	5
2	13
3	9
4	
⋮	
255	

索引表项4B
= 磁盘块号占4个字节
= 每个物理块号占4个字节

1KB

$$1\text{KB} / 4\text{B} = 1024\text{B} / 4\text{B} = 256$$

文件分配方式——索引分配

索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。

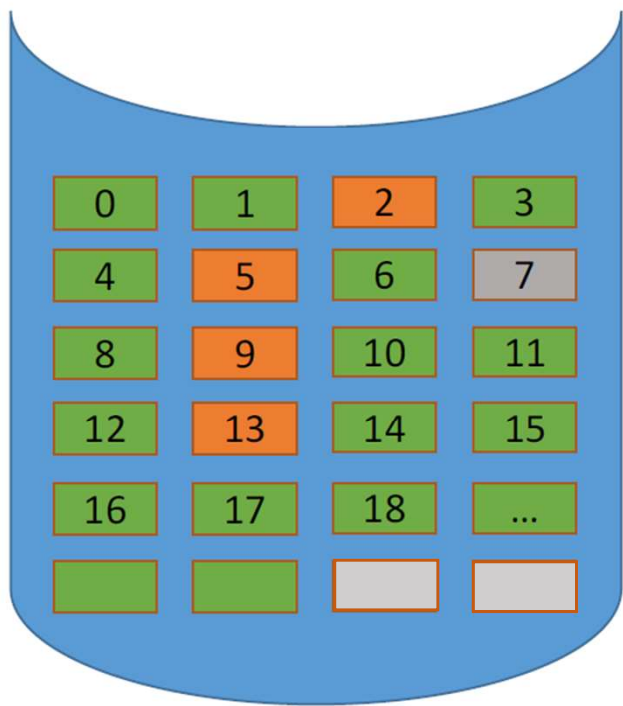
文件名	...	索引块
aaa	...	7

目录中需要记录文件的索引块是几号磁盘块



若每个磁盘块1KB，一个索引表项4B，则一个磁盘块只能存放 256 个索引项。

如果一个文件的大小超过了256块，那么一个磁盘块是装不下文件的整张索引表的，如何解决这个问题？



逻辑块号	物理块号
0	2
1	5
2	13
3	9

文件“aaa”的索引表

文件分配方式——索引分配



多层索引：建立多层索引（原理类似于多级页表）。使第一层索引块指向第二层的索引块。还可根据文件大小的要求再建立第三层、第四层索引块。

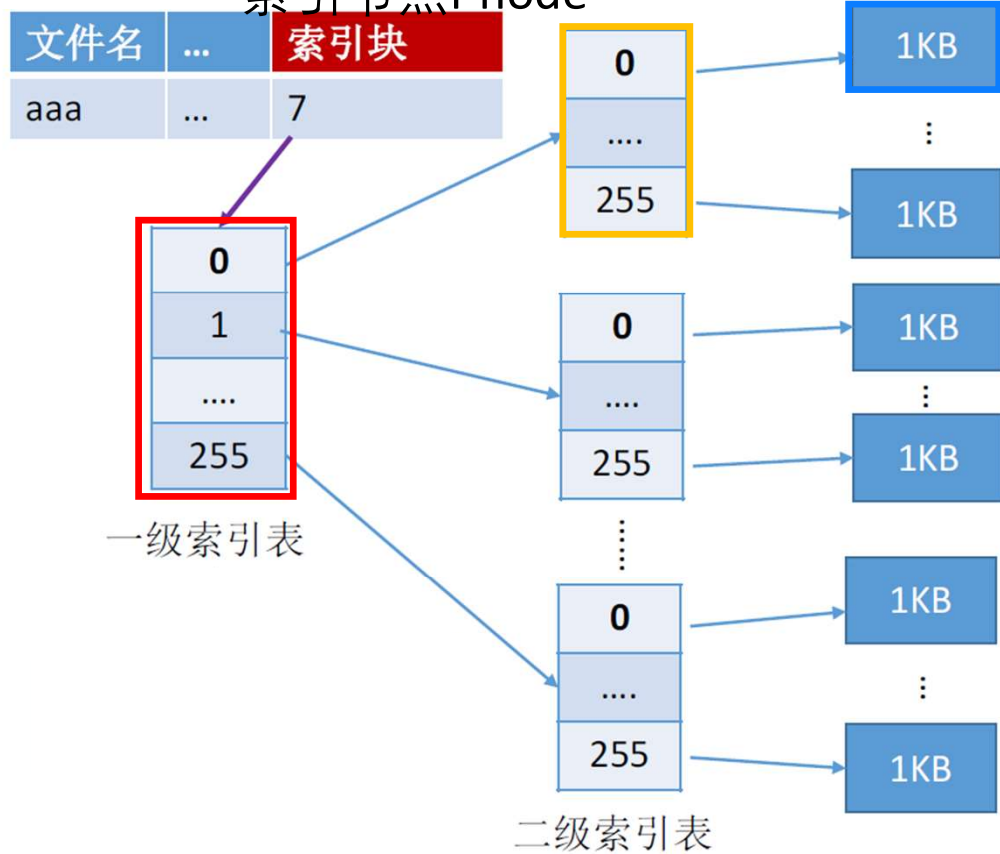
假设磁盘块大小为1KB，一个索引表项占4B，则一个磁盘块只能存放256个索引项。

若某文件采用两层索引，则该文件的最大长度可以到
 $256 * 256 * 1KB = 65,536 \text{ KB} = 64\text{MB}$

文件分配方式——索引分配

多层索引：建立多层索引（原理类似于多级页表）。使第一层索引块指向第二层的索引块。还可根据文件大小的要求再建立第三层、第四层索引块。

索引节点i-node



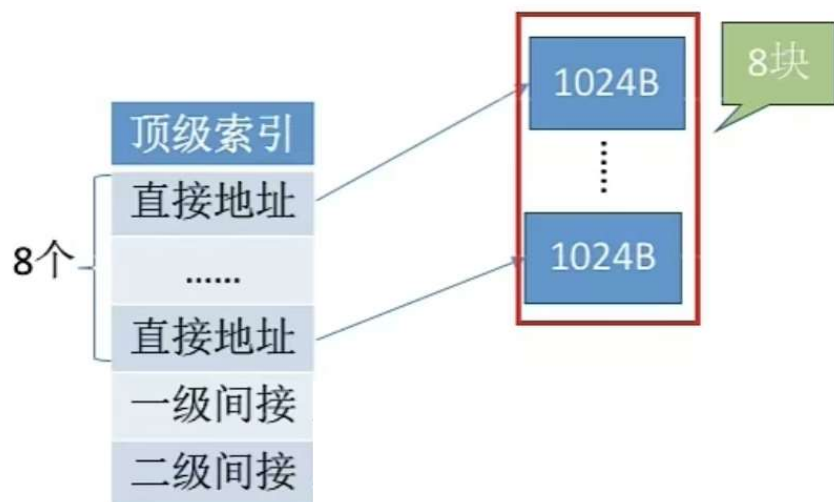
假设磁盘块大小为1KB，一个索引表项占4B，则一个磁盘块只能存放256个索引项。

若某文件采用两层索引，则该文件的最大长度可以到

$$256 \times 256 \times 1\text{KB} = 65,536 \text{ KB} = 64\text{MB}$$

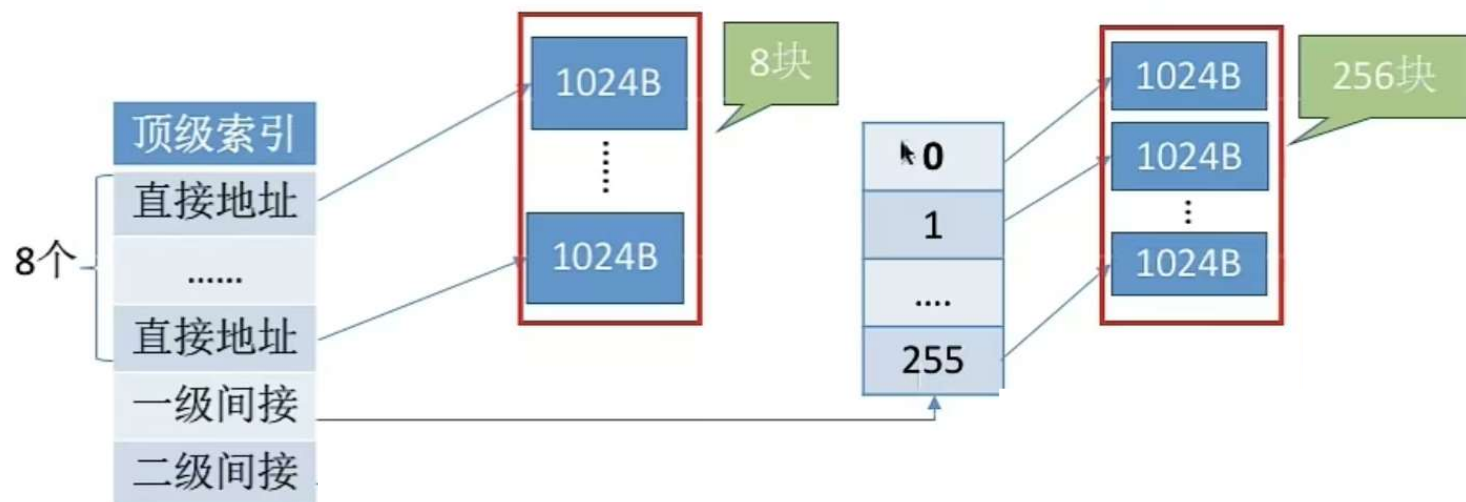
文件分配方式——索引分配

混合索引：多种索引分配方式的结合。例如，一个文件的顶级索引表中，既包含**直接地址索引**（直接指向**数据块**），又包含**一级间接索引**（指向单层索引表）、还包含**两级间接索引**（指向两层索引表）。



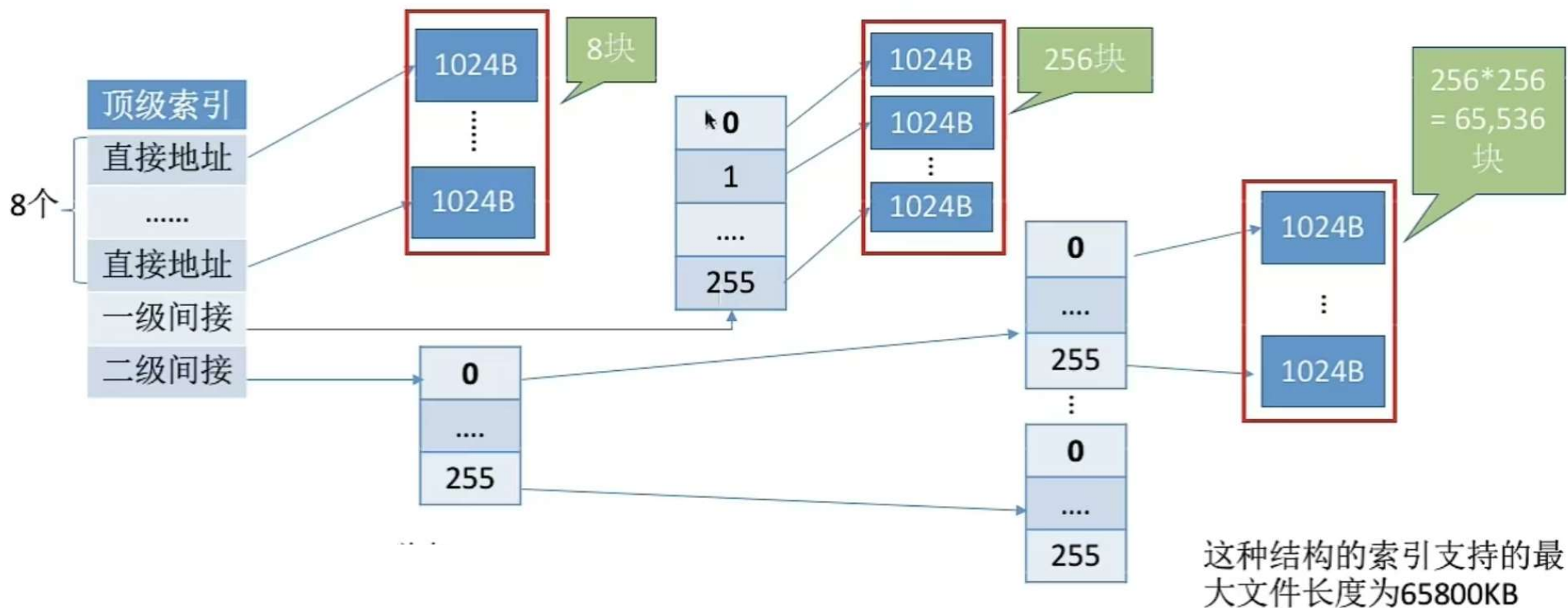
文件分配方式——索引分配

混合索引：多种索引分配方式的结合。例如，一个文件的顶级索引表中，既包含**直接地址索引**（直接指向**数据块**），又包含**一级间接索引**（指向单层索引表）、还包含**两级间接索引**（指向两层索引表）。



文件分配方式——索引分配

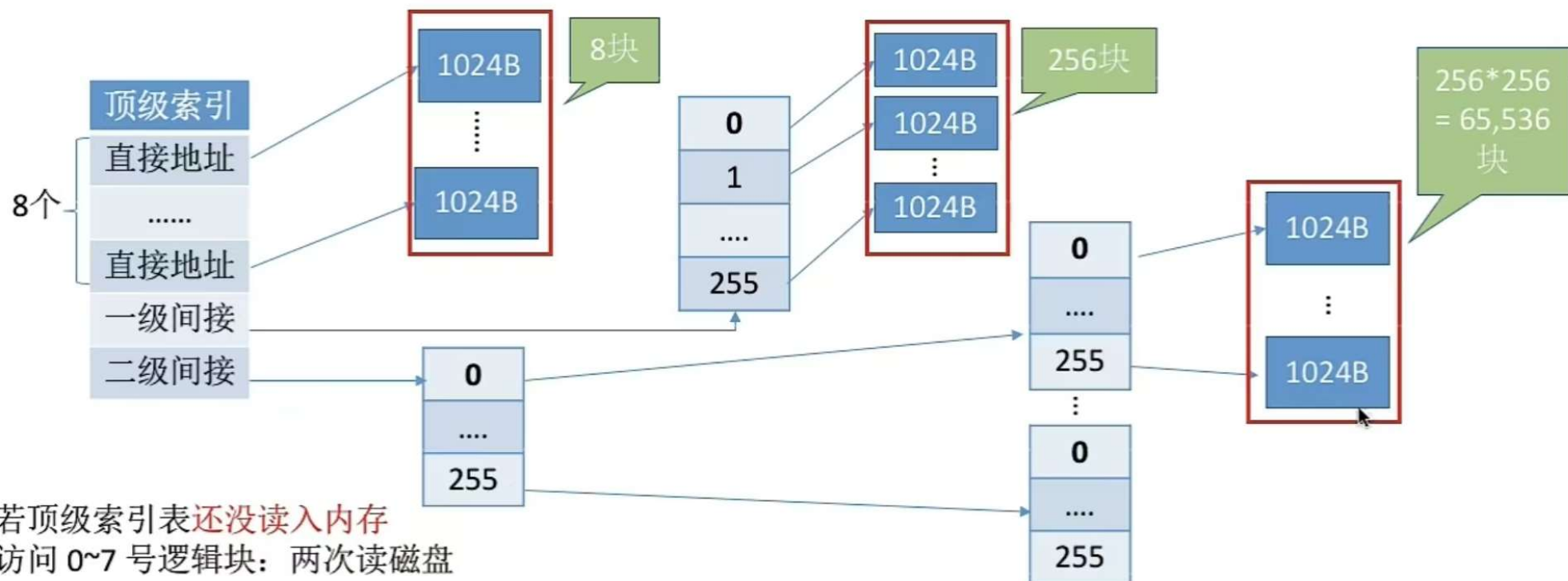
混合索引：多种索引分配方式的结合。例如，一个文件的顶级索引表中，既包含**直接地址索引**（直接指向**数据块**），又包含**一级间接索引**（指向单层索引表）、还包含**两级间接索引**（指向两层索引表）。



$$(8+256+256*256)*1KB=65800KB$$

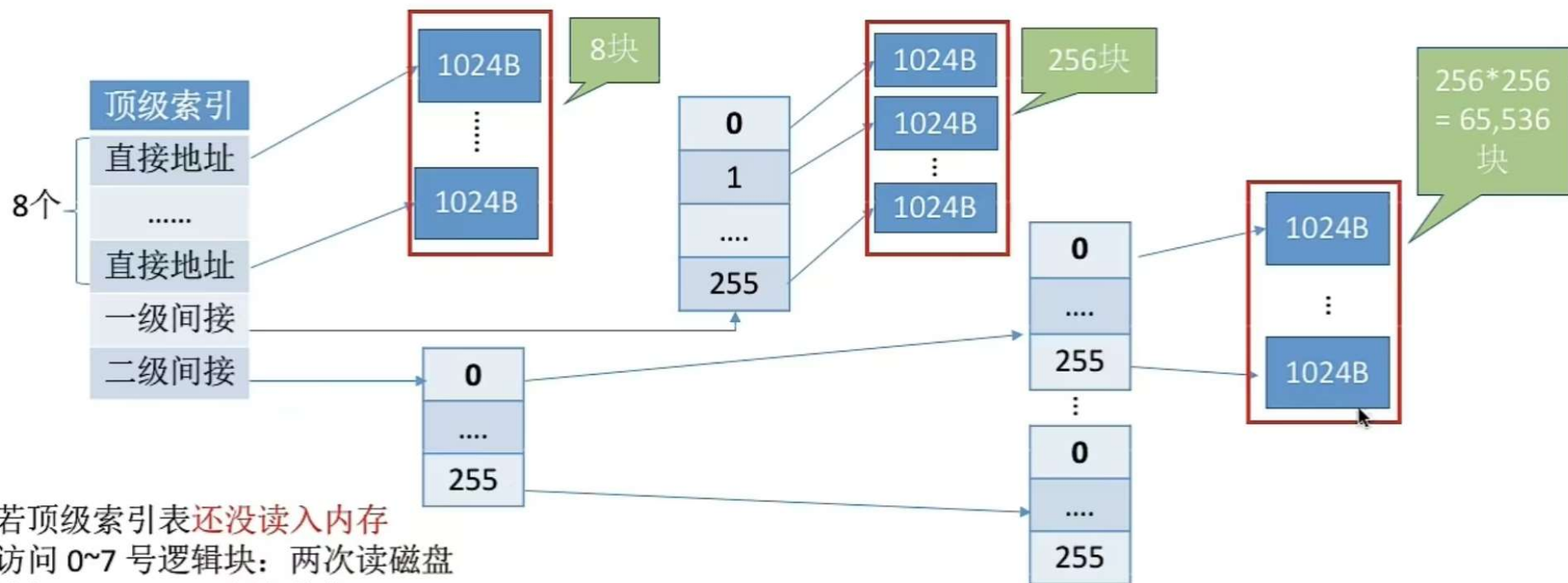
文件分配方式——索引分配

混合索引：多种索引分配方式的结合。例如，一个文件的顶级索引表中，既包含**直接地址索引**（直接指向**数据块**），又包含**一级间接索引**（指向单层索引表）、还包含**两级间接索引**（指向两层索引表）。



文件分配方式——索引分配

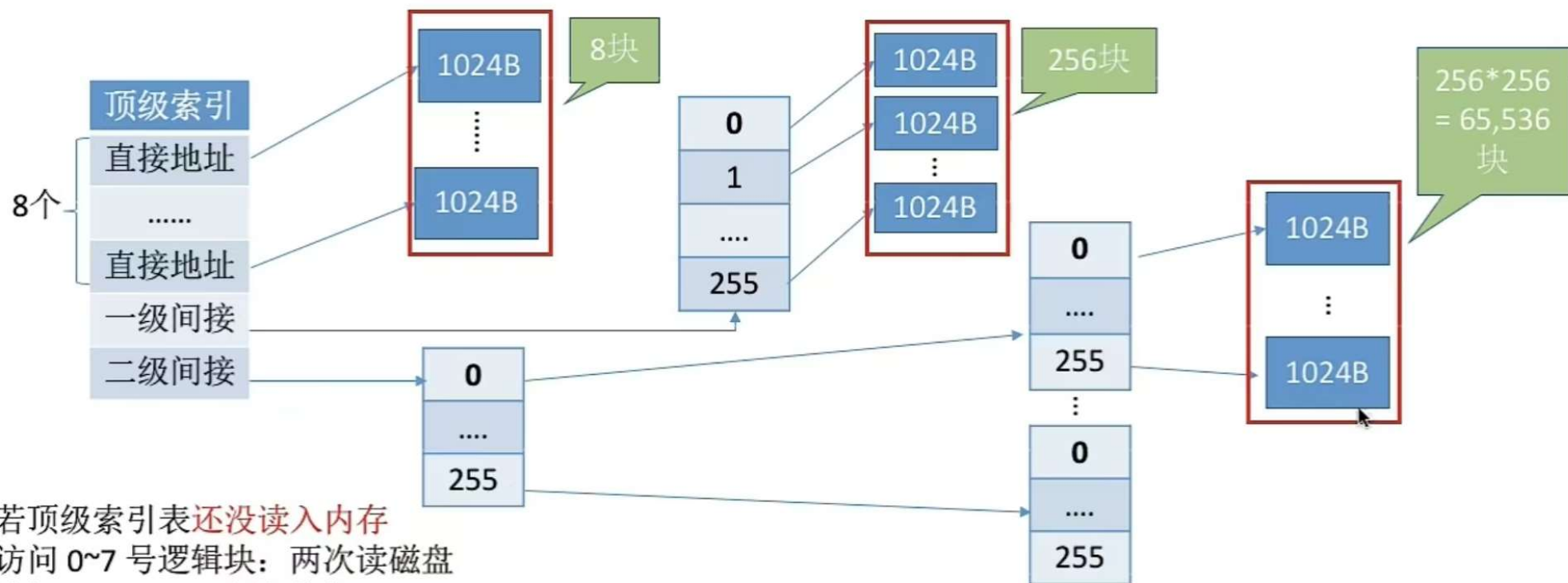
混合索引：多种索引分配方式的结合。例如，一个文件的顶级索引表中，既包含**直接地址索引**（直接指向**数据块**），又包含**一级间接索引**（指向单层索引表）、还包含**两级间接索引**（指向两层索引表）。



若顶级索引表还没读入内存
访问 0~7 号逻辑块：两次读磁盘
访问 8~263：三次读磁盘

文件分配方式——索引分配

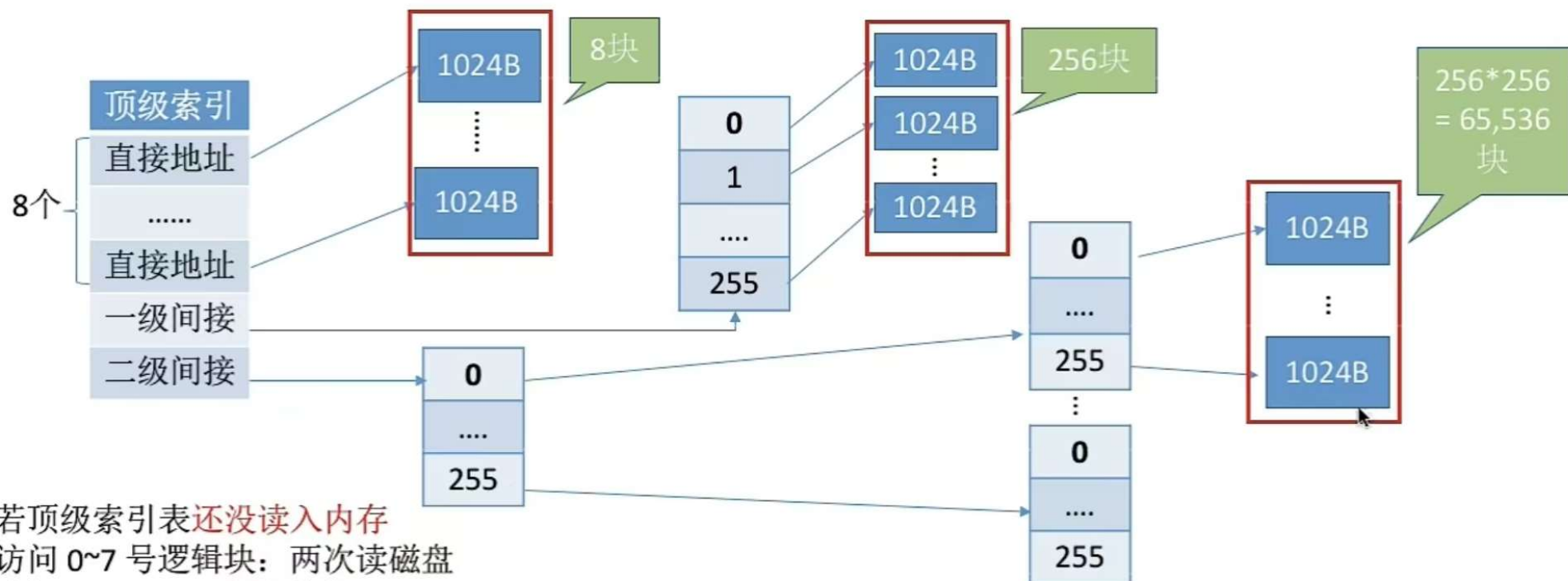
混合索引：多种索引分配方式的结合。例如，一个文件的顶级索引表中，既包含**直接地址索引**（直接指向**数据块**），又包含**一级间接索引**（指向单层索引表）、还包含**两级间接索引**（指向两层索引表）。



若顶级索引表还没读入内存
访问 0~7 号逻辑块：两次读磁盘
访问 8~263：三次读磁盘
访问 264~ 65799：四次读磁盘

文件分配方式——索引分配

混合索引：多种索引分配方式的结合。例如，一个文件的顶级索引表中，既包含**直接地址索引**（直接指向**数据块**），又包含**一级间接索引**（指向单层索引表）、还包含**两级间接索引**（指向两层索引表）。



若顶级索引表还没读入内存
访问 0~7 号逻辑块：两次读磁盘
访问 8~263：三次读磁盘
访问 264~ 65799：四次读磁盘

对于小文件，只需较少的读磁盘次数就可以访问目标数据块。
(一般计算机中小文件更多)

1、假设某文件系统中磁盘物理块大小为4KB，每个物理块号占4个字节，在两级索引结构中，允许的最大文件长度是多少？

答：

- 一个物理块号能够存储 $4\text{KB}/4\text{B}=1024$ 个物理块号；
- 两级索引共能存储指向实际存储文件内容的 $1024*1024$ 个物理块号
- 最大文件长度： $1024*1024*4\text{KB}=4\text{GB}$

1、某文件系统的文件目录项中有六个表目的数组来描述文件的物理结构，该数组的前四个表目用作直接索引，第五个表目为一级间接索引，第六个表目为二级间接索引，磁盘块的大小为512字节，块号占2个字节；

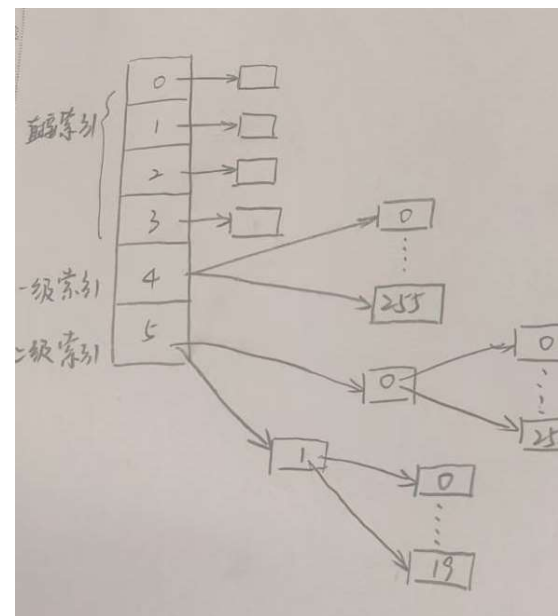
(1) 该文件系统能够构造的最大文件有多少个字节？

答: 每块 512 字节,块号 2 字节,则每个数据块可以指示 256 块,则最大文件的总块数:
 $4+256+256^2=65796$,字节数为 $65796*512$ 。

(2) 某个文件file有268KB，请画出该文件数据块的索引结构。

答: 268K 需占 $268K/512=536$ 块, $536=4+256+276$,要用到二级索引。 $276-256=20$

索引结构图:框中为文件的逻辑块号,从 0 开始编号。



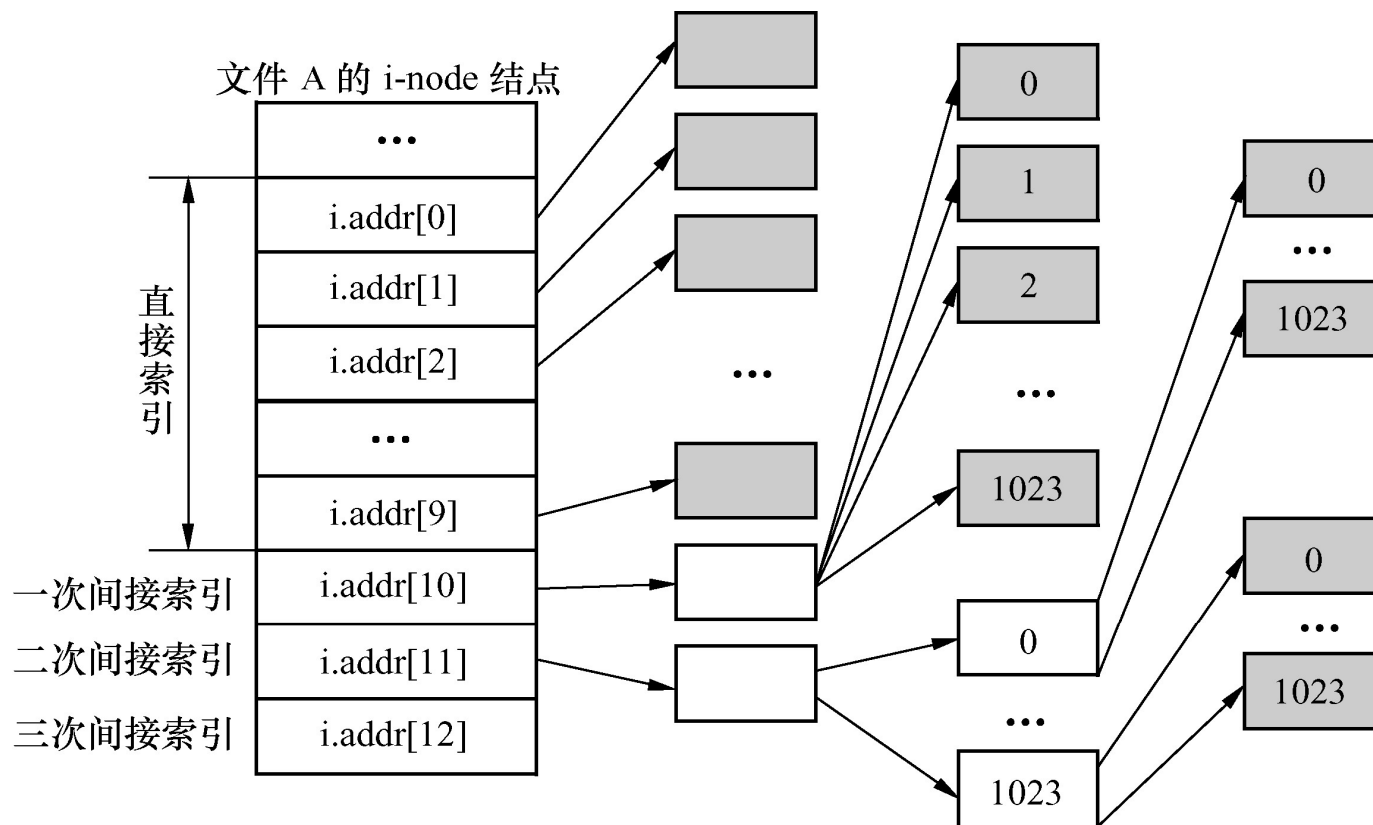
2、某版本的Linux操作系统采用的是混合索引方式。每个文件的索引表规定为13个索引项，每项4个字节，登记一个存放文件信息的物理块号。

这意味着索引表中有13个记录条目，每个记录条目占据4个字节来存储一个物理块号。前面的10项用于直接寻址，而后续的11、12、13项用于间接寻址。

前面10项存放文件信息的物理块号，叫直接寻址。假定物理块的大小是4KB，如果文件大于10块，则利用第11项指向一个物理块，该块中最多可放1K个存放文件信息的物理块的块号，这种方式是一次间接寻址。大型文件还可以利用第12和13项作二次和三次间接寻址。

请画出混合索引文件结构图，并回答直接寻址可以表示多大的文件？一次间接寻址，可以表示多大的文件？二次和三次间接寻址分别可以表示多大的文件？

答：



- 直接寻址可以表示40KB ($4\text{KB} \times 10$) 的文件；
- 这种方式是一次间接寻址，可以表示4MB ($4\text{KB} \times 1024$) 的文件；
- 二次间接寻址可以表示4GB ($4\text{KB} \times 1024 \times 1024$) 的文件；
- 三次间接寻址可以表示4TB ($4\text{KB} \times 1024 \times 1024 \times 1024$) 的文件。

文件的物理 结构

Linux
Android
Linux
OpenStack
Mac OS
Windows

