

章节导学

**文件的基
本概念**

文件的逻辑结构

**文件目录
文件共享**

本章要求

1. 文件的基本概念和属性

1. 理解文件的属性和内部数据组织方式
2. 学习文件之间的组织方式
3. 理解操作系统应提供的相关文件管理功能

2. 文件的逻辑结构:

1. 区分无结构文件和有结构文件
2. 掌握顺序文件、索引文件、索引顺序文件、多级索引顺序文件等逻辑结构的特点和应用
3. 理解文件在外存中的存放方式

3. 文件系统的实现和文件共享:

1. 学习各种文件目录结构，如单级目录结构、两级目录结构、多级目录结构、无环图目录结构和索引结点
2. 理解文件共享的概念，区分硬链接和软连接，以及它们的实现方式
3. 了解文件系统的实现和文件的层次结构

提示

本章内容就帮助你理解文件系统的实现机制，
这些内容是操作系统课程的**重点内容，也是考研中的重点。**

文件

一切皆文件 宿船长

一切皆文件：Linux中的七种文件

在Linux系统中，有七种基本的文件类型。它们是：

1. 普通文件（Regular file）：通常用来存储文本或二进制数据，如文本文件、图像文件等。
2. 目录文件（Directory file）：用来存储其他文件和目录的位置信息。
3. 符号链接文件（Symbolic link file）：类似于Windows系统中的快捷方式，它指向另一个文件或目录。
4. 块设备文件（Block device file）：用于存储数据的设备，如硬盘、U盘等。
5. 字符设备文件（Character device file）：用于传输字符流的设备，如键盘、鼠标等。
6. 套接字文件（Socket file）：用于进程间通信的文件。
7. 管道文件（FIFO file）：类似于套接字文件，也用于进程间通信，但是它只能用于相关进程之间的通信。

drwxr-xr-x

Type	User			Group			Other		
d	r	w	x	r	-	x	r	-	x
	1	1	1	1	0	1	1	0	1
	2^2	2^1	2^0	2^2	0	2^0	2^2	0	2^0
	4	2	1	4	0	1	4	0	1
	7			5			5		

普通文件

普通文件在Linux中的文件类型占位符为 `-`，普通文件主要有以下三种：

1. 文本文件：源代码，txt文档等
2. 二进制可执行文件：源码编译后的可执行文件
3. 其他特殊编码格式的数据文件：`/var/log/wtmp`，图片，音视频，数据库文件等等

目录文件

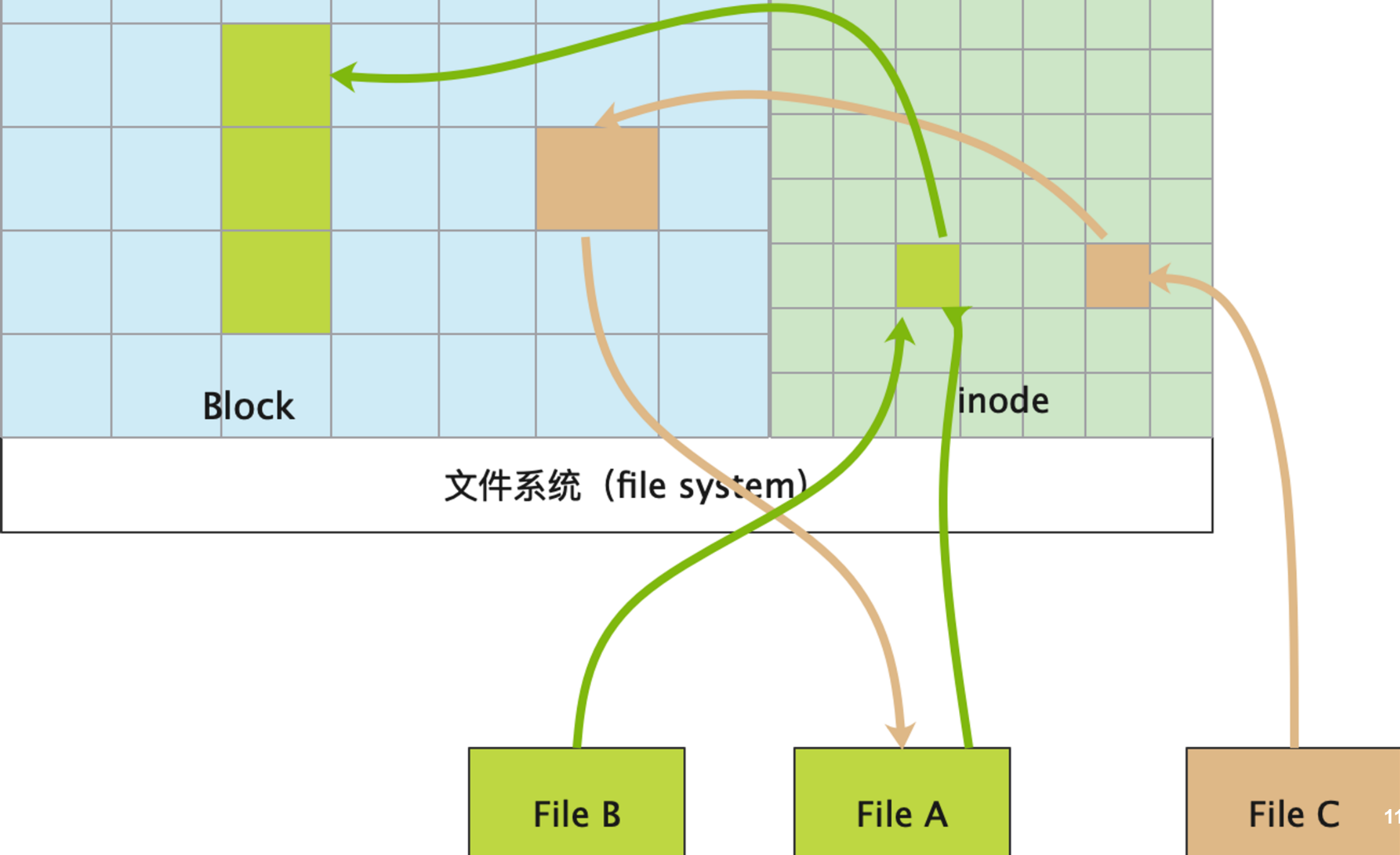
目录文件在Linux中的文件类型占位符为 **d**，在Linux中，有一些跟目录相关的概念，需要着重强调

1. 家目录不是/home目录：每个用户都有自己的家目录，除root的家目录在/root外，其他普通用户的家目录位于/home目录下面的以自己用户名为名的目录下
2. PATH路径：准确的说，PATH是shell中的一个环境变量，在各种操作系统中都有，用以存储可执行文件所在的目录，当我们在shell里写入指令时，实际是在所有PATH路径里查找是否存在该可执行文件

链接文件

链接文件（符号链接，软链接）在Linux中的文件类型占位符为 `l`，软链接相对于硬链接，是操作系统中，必须要掌握的知识

1. 软链接：一个新的文件，文件内容为原始文件的具体存储位置，跟Windows中的快捷方式一样；
2. 硬链接：本质上是同一个文件，只是这个文件有了不止一个名字，相当于与一个具有双国籍的人，或者有两个户口的人，虽然名字，国籍，户口所在位置不同，但本质上还是同一个人。



块设备文件

块设备文件在Linux中的文件类型占位符为 **b** ，块设备是系统中用来存储数据的设备

块（block）的概念在操作系统中广泛存在，虽然所处的层，模块各不相同，但其本质思想是相通的。它指的是一段固定大小的连续内存或磁盘空间，用于数据存储和管理。块被用于以结构化和高效的方式组织和管理数据。它们将数据分解成更小、可管理的部分，从而使数据的访问、检索和修改更容易。这个概念被应用于操作系统的许多不同部分，从文件系统到输入/输出操作。

字符设备

字符设备文件在Linux中的文件类型占位符为 **c**，意为字符文件，它的特点是该文件中传递的数据为字符数据，构成字符流，同时数据是实时的，不能存储

常见的字符设备有鼠标，键盘，串口等等

套接字文件

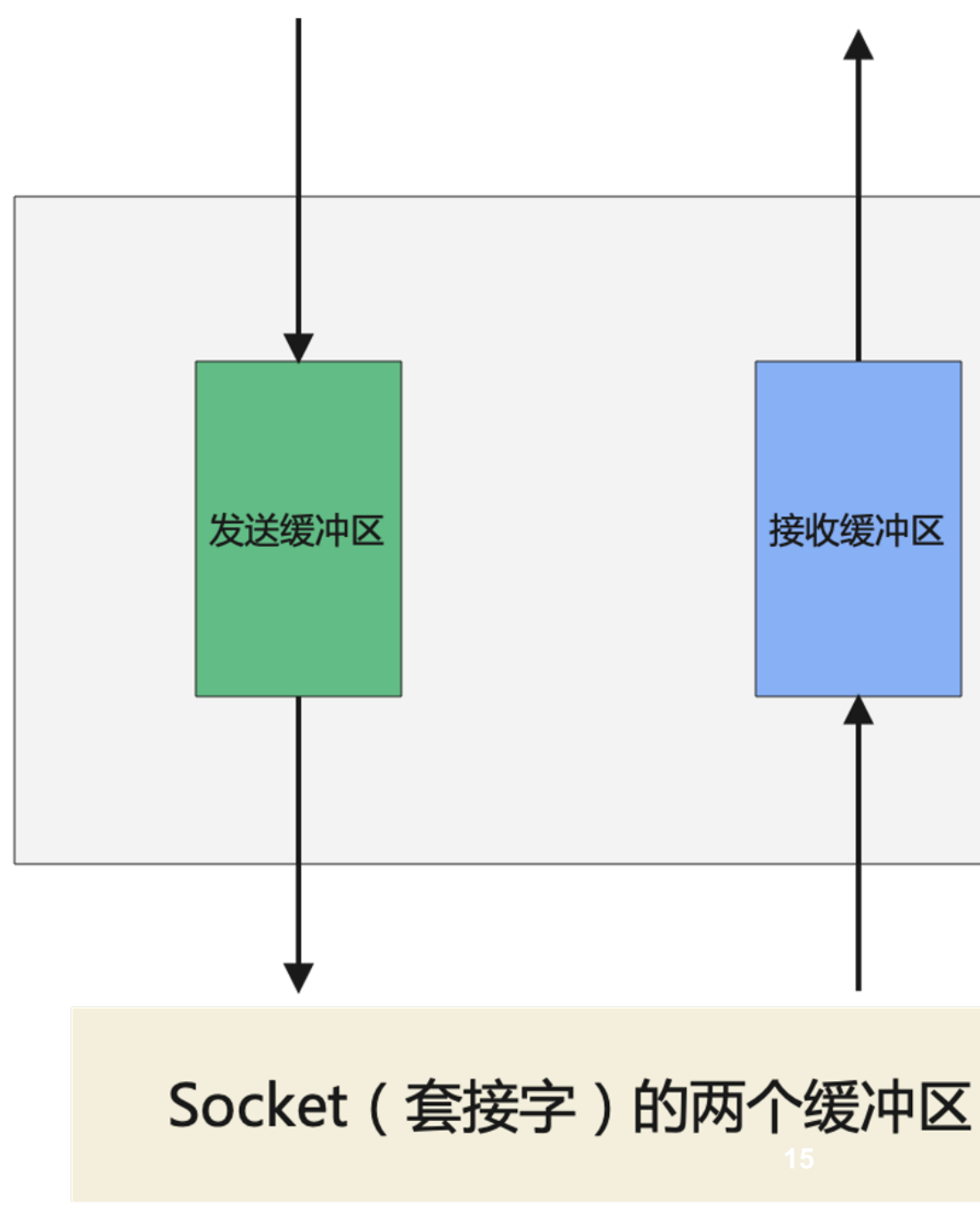
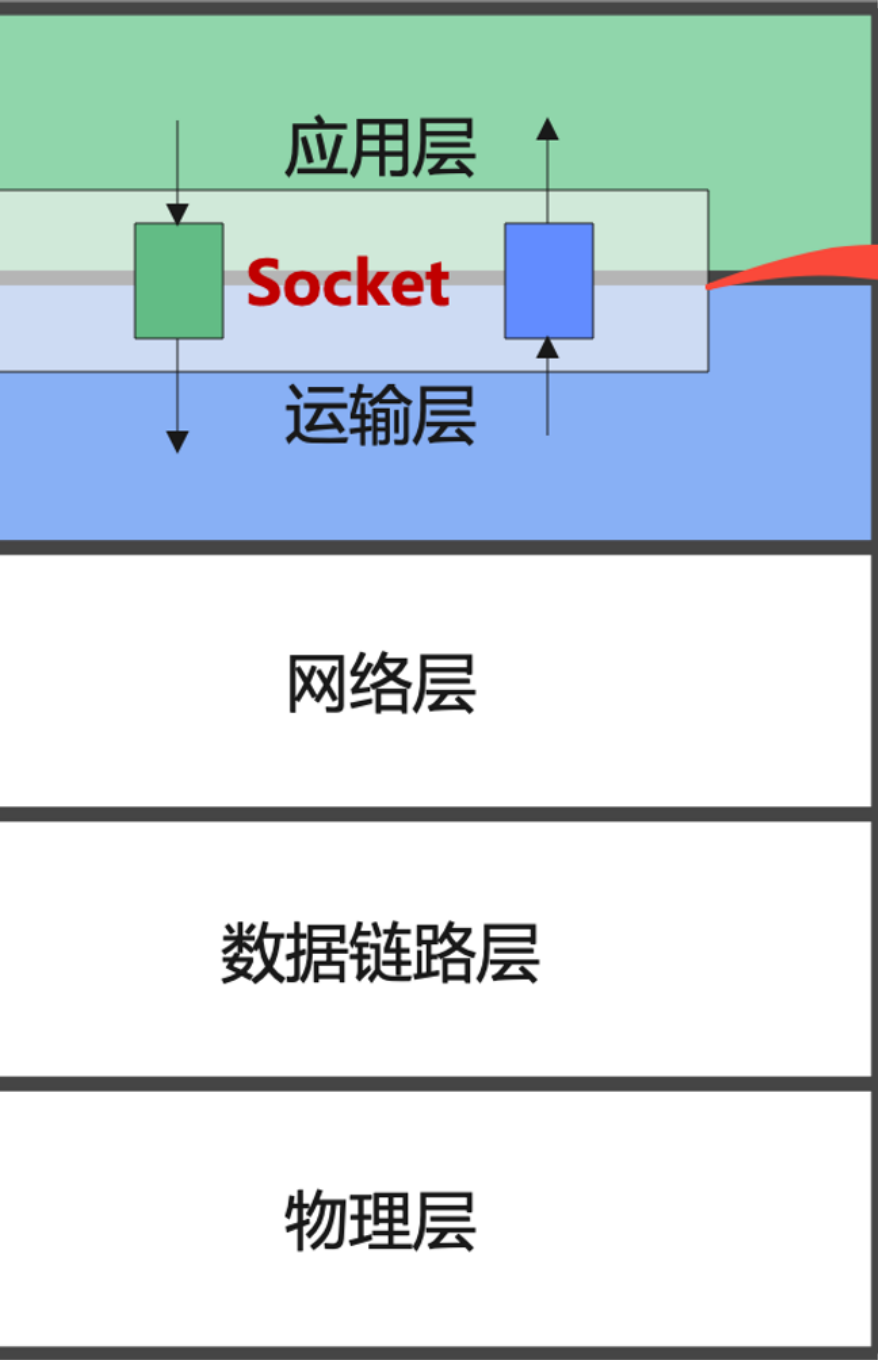
套接字文件在Linux中的文件类型占位符为 **s** ，主要用在网络通信

套接字（socket）是应用层和运输层的接口，负责将应用层的数据，通过运输层的TCP或者UDP传输出去。Socket是网络编程中，发送和接收数据的唯一接口，发数据则往socket文件写，收数据则是从socket中读数据。

套接字有两个缓冲区：

1. **接收缓冲区**：也称为输入缓冲区，它存储来自网络的待读取应用程序的数据。
2. **发送缓冲区**：也称为输出缓冲区，它存储应用程序等待发送到网络的数据。

这些缓冲区用于临时存储数据，直到应用程序准备处理或发送到网络为止。这些缓冲区的大小可以使用各种套接字选项进行调整。



管道文件

管道文件在Linux中的文件类型占位符为p，管道文件，也称为命名管道，是Linux中用于进程间通信（IPC）的一种文件类型。

1. 它允许两个或多个进程通过读写共享数据缓冲区来相互通信。管道文件类似于普通文件，但不同的是，它只存在于内存中，而不是存储在磁盘上。
2. 文管道文件有两个端口，一个用于写入，一个用于读取。
3. 当一个进程向管道文件的写入端口写入数据时，数据将被存储在共享缓冲区中。另一个进程可以从管道文件的读取端口读取数据。数据按照写入的顺序读取，所以第一个写入的数据也是第一个读取的数据。
4. 管道文件可用于在同时运行的进程之间传递数据。例如，一个进程可能生成需要由另一个进程处理的数据。第一个进程可以将数据写入管道文件，第二个进程可以从管道文件中读取数据并对其进行处理。

`ps -ef`

`grep ping`

PIPE

写端

读端

`ps -ef | grep ping`

一切皆文件的好处

这种“一切皆文件”的设计哲学带来了许多好处，包括：

1. **统一的接口**: 统一的文件I/O接口简化了系统的管理和编程，使得用户和开发者无需了解不同资源的特定接口和细节。
2. **易于扩展**: 新的设备和资源可以通过文件系统的形式轻松地添加到系统中，无需修改现有的程序或工具。
3. **灵活性**: 由于一切都被视为文件，因此可以使用标准的文件操作工具（如cat、cp、mv等）来操作各种资源，提高了系统的灵活性和可编程性。
4. **安全性**: Unix和Linux系统通过权限控制来限制对文件的访问，从而提高了系统的安全性。

总之，“一切皆文件”的设计哲学是Unix和Linux操作系统的重要特征之一，它在操作系统的设计和使用中起着至关重要的作用。

一切皆文件的例子

如何给云主机上的其他用户留言？

文件的基本概念

文件的相关概念——文件的属性

一个文件有哪些属性？

文件名：由创建文件的用户决定文件名，主要是为了方便用户找到文件，**同一目录下不允许有重名文件**。

标识符：一个系统内的各文件标识符唯一，对用户来说毫无可读性，因此标识符只是操作系统用于区分各个文件的一种内部名称。

类型：指明文件的类型

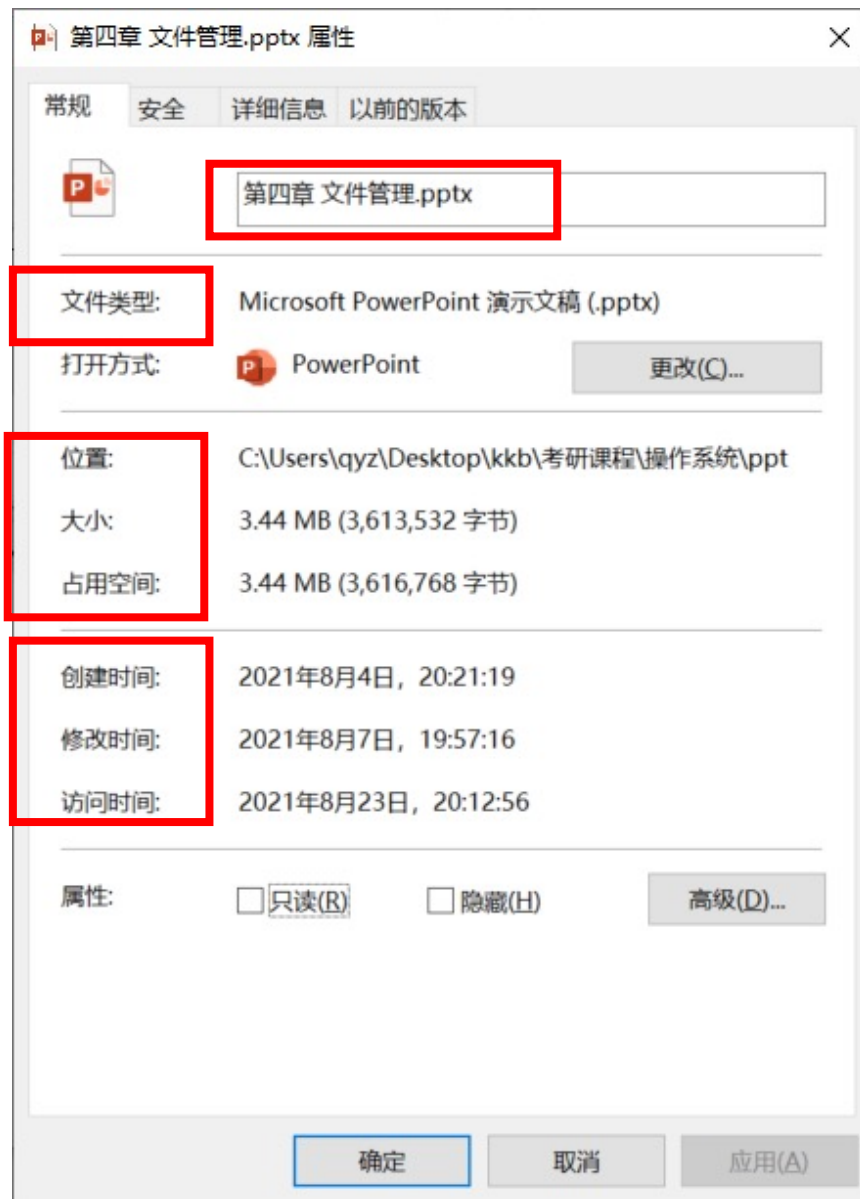
位置：文件存放的路径（让用户使用）、在外存中的地址（操作系统使用，对用户不可见）

大小：指明文件大小

创建时间、上次修改时间

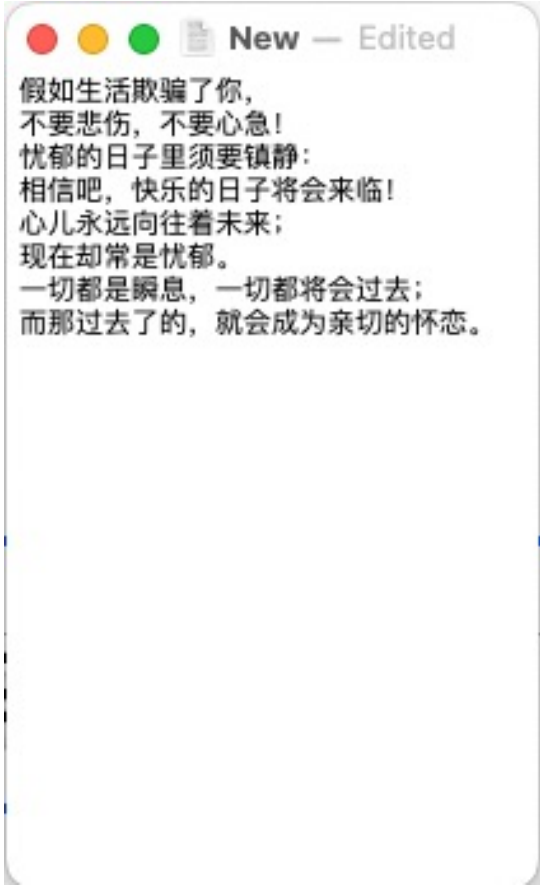
文件所有者信息

保护信息：对文件进行保护的访问控制信息



文件的相关概念——文件内部的数据应该怎样组织起来？

无结构文件（如文本文件）——由一些二进制或字符流组成，又称“流式文件”



记录

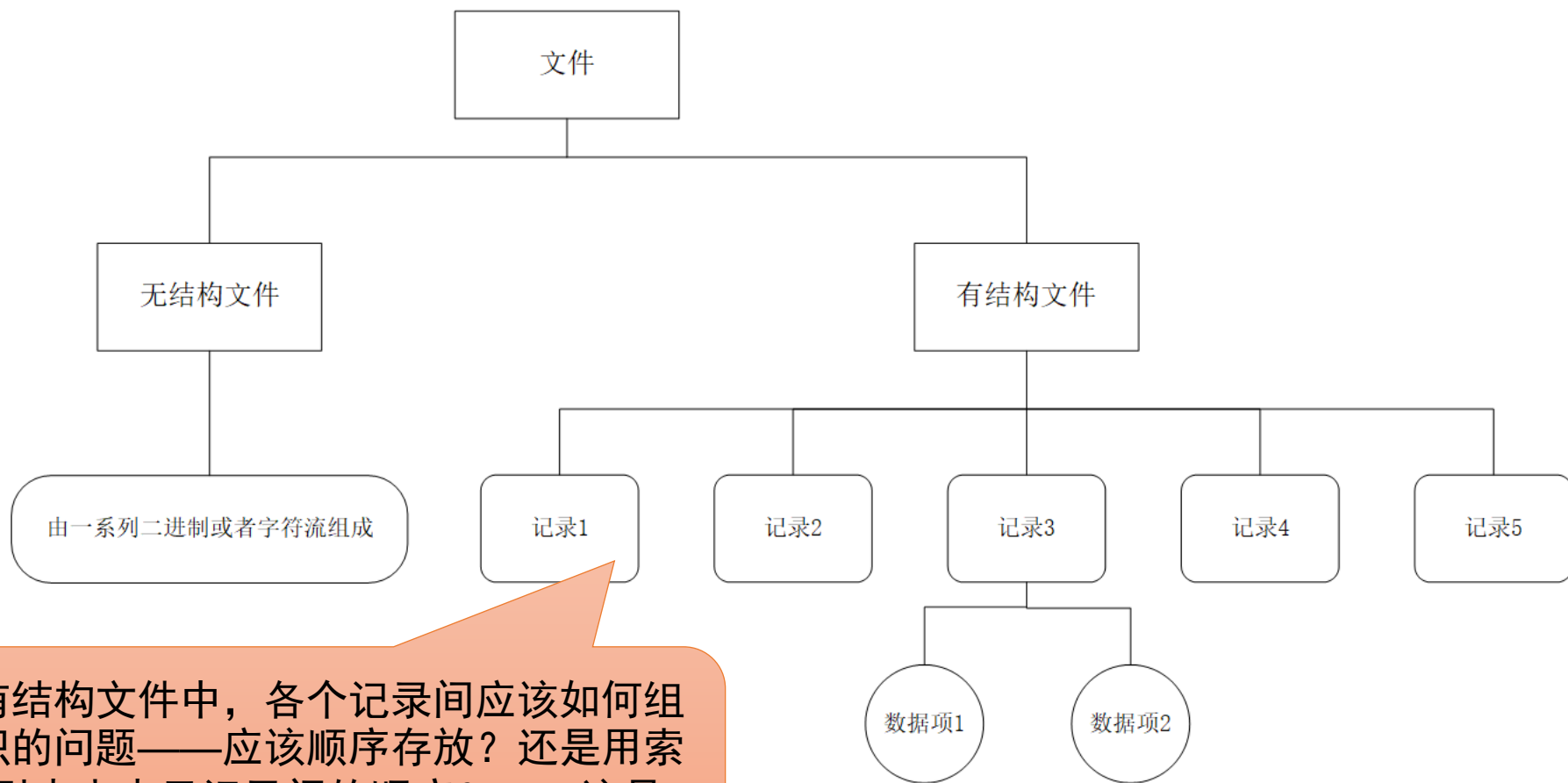
2	日期	题目名称	题目链接	难度	知识点
3		a+b	0d6a11417?tpId=40&&tqId=	☆	没啥知识点
4	1月24日	abc	d7b8750b6?tpId=40&&tqId=	☆☆	穷举
5		成绩排序	e0610418b1?tpId=40&&tqId=	☆☆	排序
6		球的体积和半径	5c8c2ec96b?tpId=40&&tqId=	☆	数学?
7	1月25日	字母统计	573b831f3c?tpId=40&&tqId=	☆	字符串
8		谁是你的潜在朋友	2b1c6f9cc?tpId=40&&tqId=	☆☆☆	排序
9		反序输出	b3b40171bb?tpId=40&&tqId=	☆	字符串
10	1月26日	手机键盘	05cd476cd?tpId=40&&tqId=	☆☆	字符串、模拟
11		搬水果	d2e1eef8a?tpId=40&&tqId=	☆☆☆	贪心
12		今年的第几天?	d70eaf04d?tpId=40&&tqId=	☆	模拟
13	1月27日	小白鼠排队	ba60cf372b?tpId=40&&tqId=	☆☆	排序、字符串
14		哈夫曼树	b2fcd155?tpId=40&&tqId=	☆☆☆	贪心
15		n的阶乘	998ca628c8?tpId=40&&tqId=	☆☆	模拟 (也可递归)
		约数个数	9f053fa4d6?tpId=40&&tqId=	☆☆	数学、穷举
		查找	3842aca8aa?tpId=40&&tqId=	☆☆	二分、排序
19	1月28日	求最大最小数	7895cc7b4?tpId=40&&tqId=	☆	查找
20		查找学生信息	ee0d86bb4?tpId=40&&tqId=	☆☆	哈希、查找
21		数字之和	d9657f4b0?tpId=40&&tqId=	☆	数学、模拟
22	1月30日	最大公约数	382536fd3?tpId=40&&tqId=	☆☆	数学
23		表达式计算	acdeb890b?tpId=40&&tqId=	☆☆☆	栈
24		特殊乘法	4c3a34240f?tpId=40&&tqId=	☆	模拟

有结构文件（如数据库表）——由一组相似的记录组成，又称“记录式文件”

其中，记录是一组相关数据项的集合；
数据项是文件系统中最基本的单位

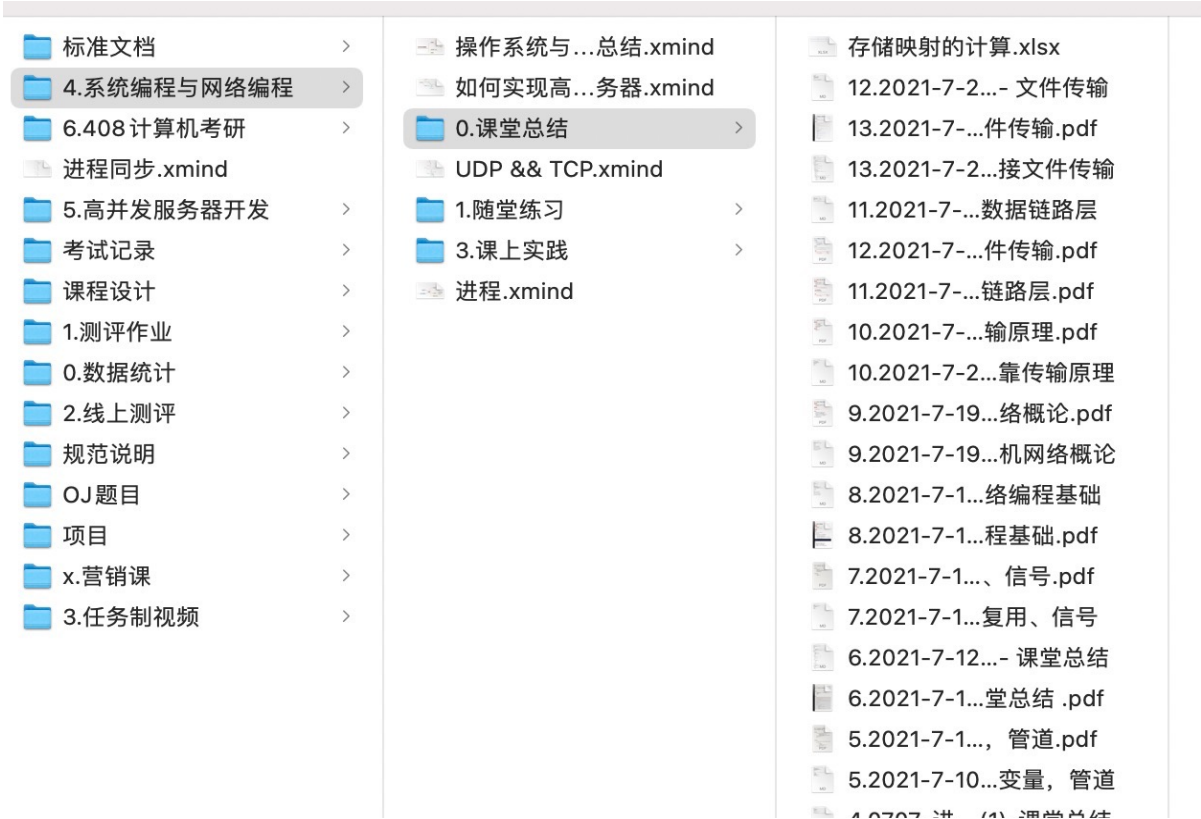
数据项

文件的相关概念——文件内部的数据应该怎样组织起来？

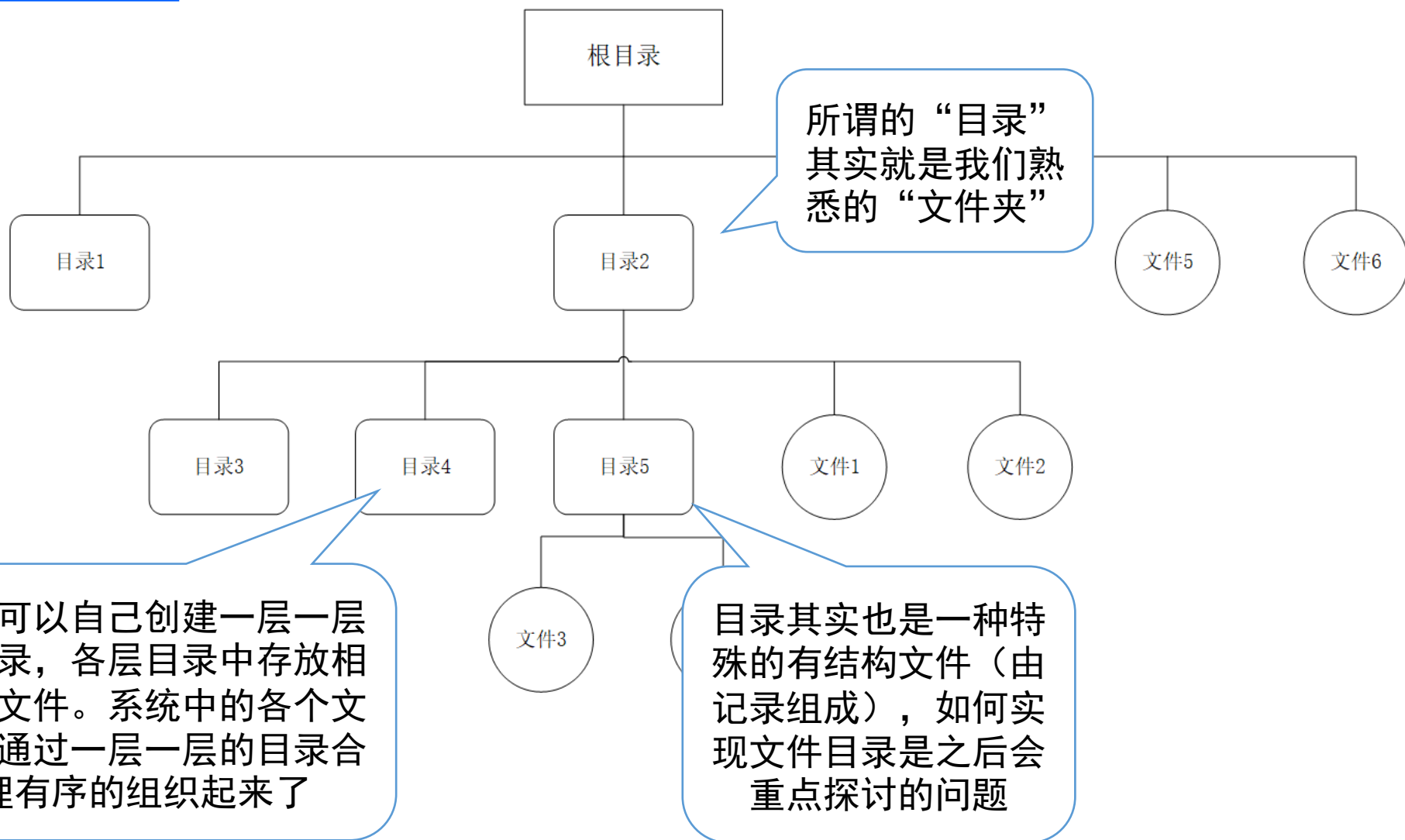


有结构文件中，各个记录间应该如何组织的问题——应该顺序存放？还是用索引来表示记录间的顺序？——这是“文件的逻辑结构”重点要探讨的问题

文件的相关概念——文件之间应该怎样组织起来？



文件的相关概念——文件之间应该怎样组织起来？



文件的相关概念——操作系统应该向上提供哪些功能？

操作系统应向上提供哪些功能

可用几个基本操作完成更复杂的操作，比如：“复制文件”：先创建一个新的空文件，再把源文件读入内存，再将内存中的数据写到新文件中

操作系统在背后做的处理会在以后进行探讨

创建文件 (create系统调用)

删除文件 (delete 系统调用)

读文件 (read 系统调用)

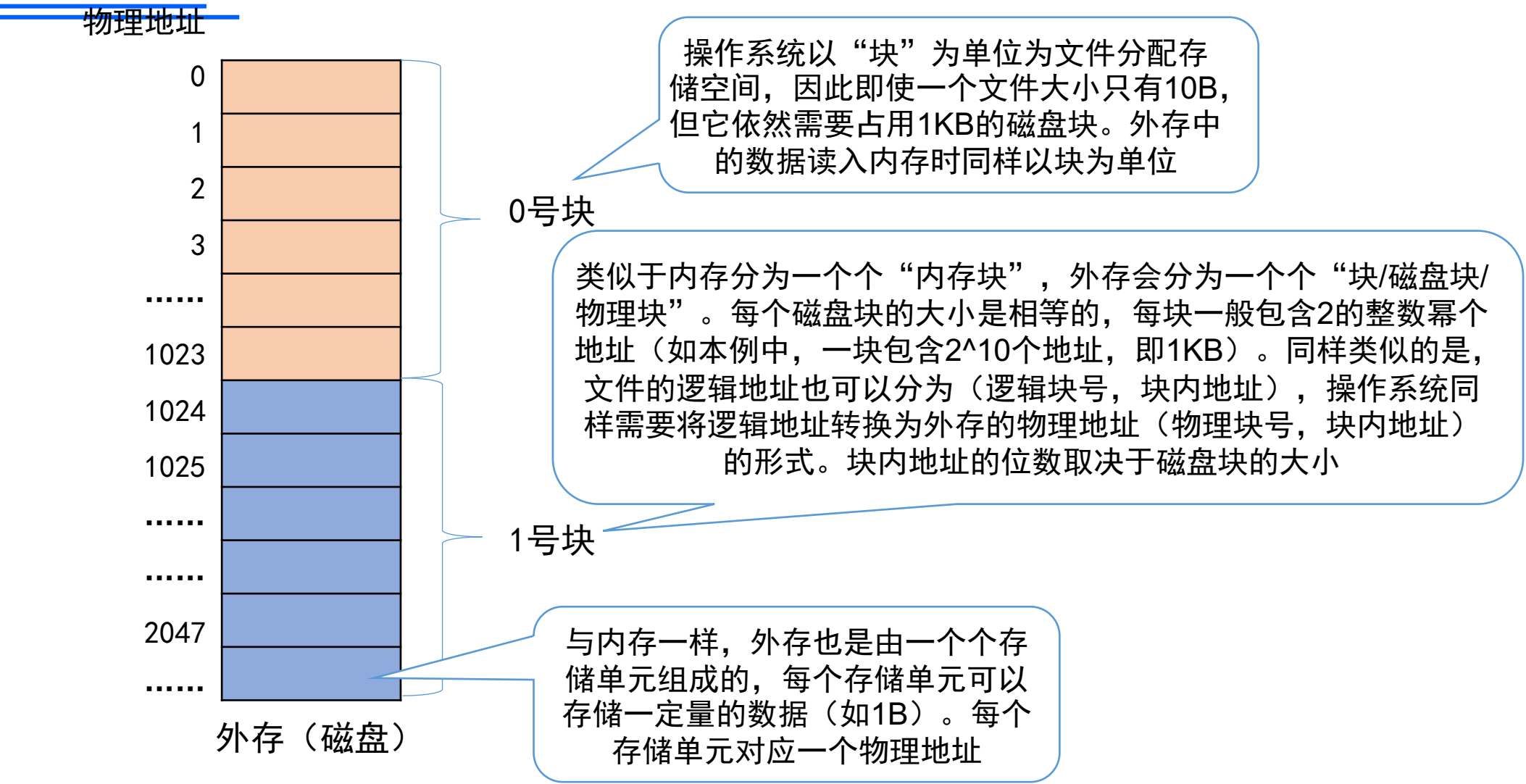
写文件 (write系统调用)

打开文件 (open系统调用)

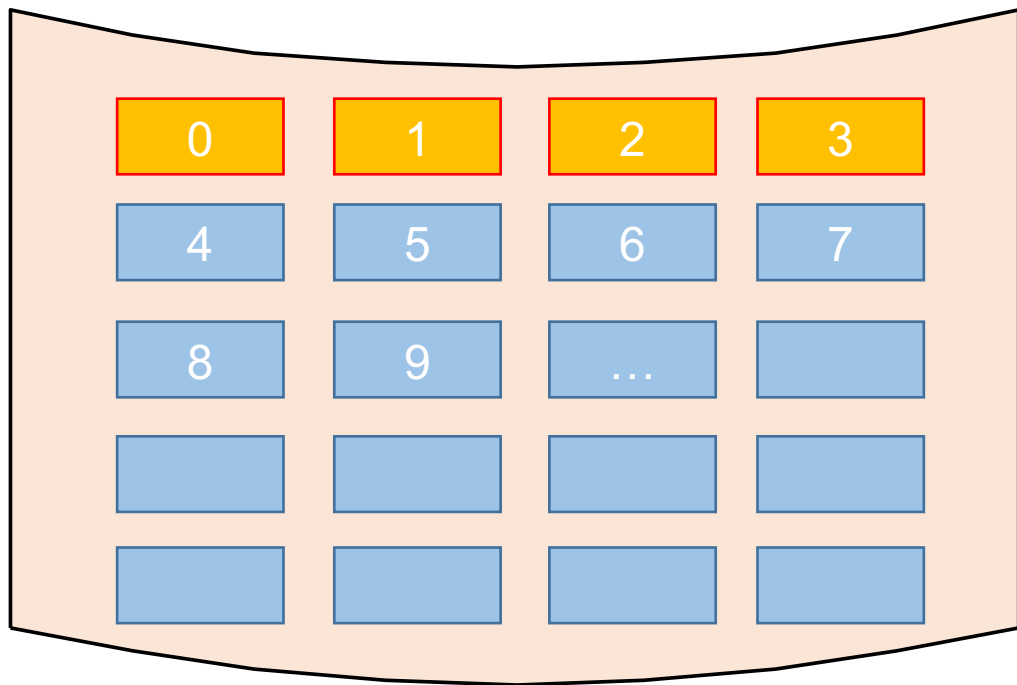
关闭文件 (close 系统调用)

读/写文件之前，需要“打开文件”
读/写文件结束之后，需要“关闭文件”

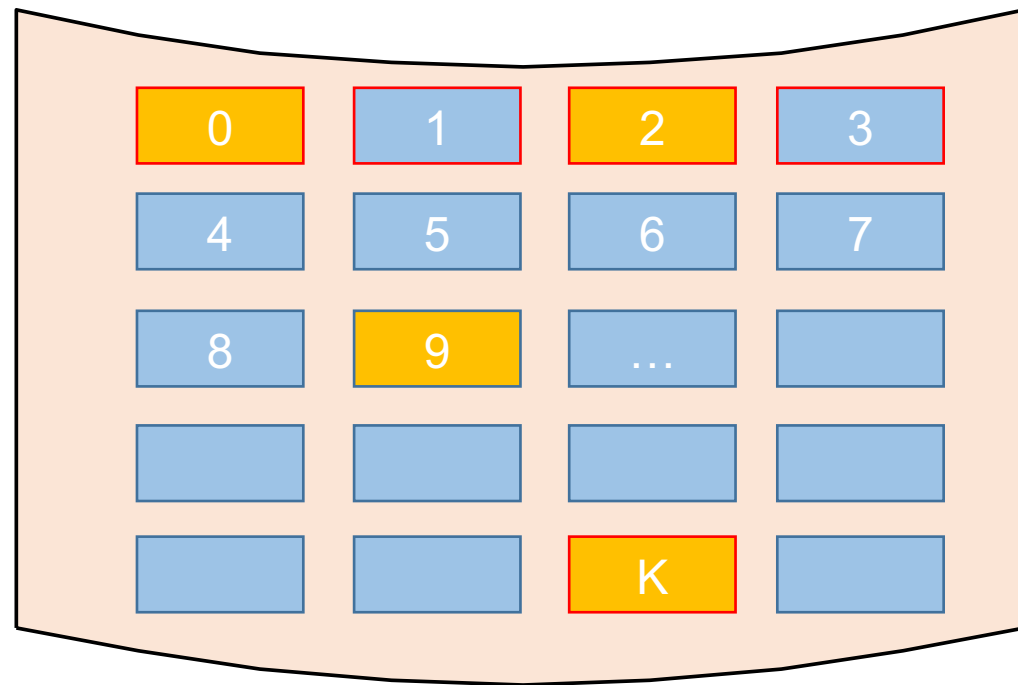
文件的相关概念——从上往下看，文件应如何存放在外存？



文件的相关概念——从上往下看，文件应如何存放在外存？



文件数据放在连续的
几个磁盘块中



文件数据放在离散的几个磁盘块中。此时，应该如何记录各个磁盘块之间的先后顺序呢？

文件的相关概念——其他需要由操作系统实现的文件管理功能

文件共享：使多个用户可以共享使用一个文件

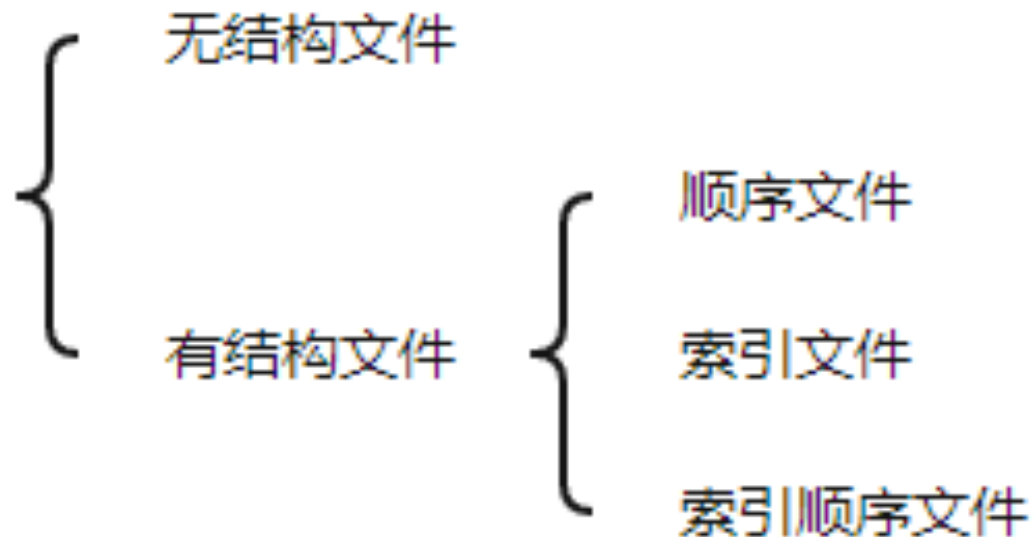
文件保护：如何保证不同的用户对文件有不同的操作权限

文件的逻辑结构

文件的逻辑结构

文件的逻辑结构

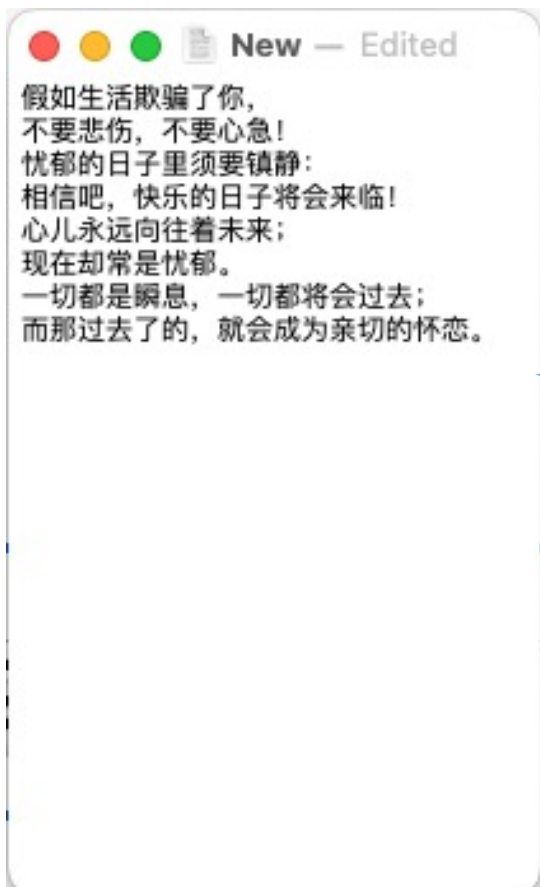
所谓的“逻辑结构”，就是指在用户看来，文件内部的数据应该是如何组织起来的。“物理结构”指的是在操作系统看来，文件的数据是如何存放在外存中的。



文件的逻辑结构——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。



文件内部的数据其实就是一系列字符流，没有明显的结构特性。因此也不用探讨无结构文件的“逻辑结构”问题。

文件的逻辑结构——有结构文件

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。

有结构文件：由一组相似的记录组成，又称“**记录式文件**”。每条记录又若干个数据项组成。如：数据库表文件。一般来说，每条记录有一个数据项可作为**关键字**（作为识别不同记录的ID）

学号	姓名	性别	专业
2021061201	王五	男	计算机
2021061202	李二	女	网络工程
2021061203	王五	男	计算机
2021061204	刘二	女	网络工程
2021061205	赵一	男	计算机
2021061206	刘二2	女	网络工程
2021061207	赵一2	男	计算机
2021061208	刘二3	女	网络工程
2021061209	赵一3	男	计算机
2021061210	刘二4	女	网络工程

在本例中，“学号”即可作为各个记录的关键字

这是一张数据库表，记录了各个学生的信息

每个学生对应一条记录，每条记录由若干个数据项组成

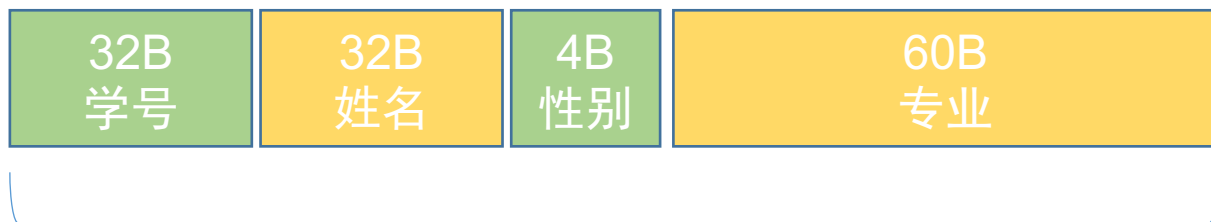
文件的逻辑结构——有结构文件

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。

有结构文件：由一组相似的记录组成，又称“**记录式文件**”。每条记录又若干个数据项组成。如：数据库表文件。一般来说，每条记录有一个数据项可作为**关键字**（作为识别不同记录的ID）
根据各条记录的长度（占用的存储空间）是否相等，又可分为**定长记录**和**可变长记录**两种。



这个有结构文件由**定长记录**组成，每条记录的长度都相同（共128 B）。各数据项都处在记录中相同的位置，具有相同的顺序和长度（前32B一定是学号，之后32B一定是姓名……）

文件的逻辑结构——有结构文件

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。

有结构文件：由一组相似的记录组成，又称“**记录式文件**”。每条记录又若干个数据项组成。如：数据库表文件。一般来说，每条记录有一个数据项可作为**关键字**（作为识别不同记录的ID）
根据各条记录的长度（占用的存储空间）是否相等，又可分为**定长记录**和**可变长记录**两种。



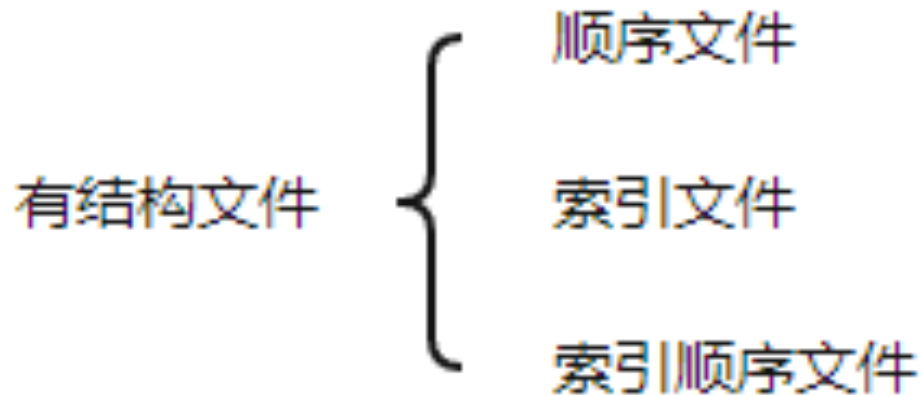
这个有结构文件由**可变长记录**组成，由于各个学生的特长存在很大区别，因此“特长”这个数据项的长度不确定，这就导致了各条记录的长度也不确定。当然，没有特长的学生甚至可以去掉“特长”数据项。

文件的逻辑结构——有结构文件的逻辑结构

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。

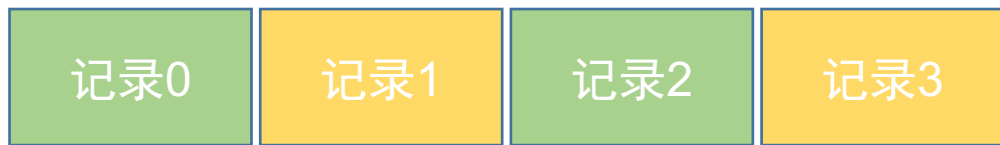
无结构文件：文件内部的数据就是一系列二进制流或字符流组成。又称“**流式文件**”。如：Windows操作系统中的.txt文件。

有结构文件：由一组相似的记录组成，又称“**记录式文件**”。每条记录又若干个数据项组成。如：数据库表文件。一般来说，每条记录有一个数据项可作为**关键字**（作为识别不同记录的ID）根据各条记录的长度（占用的存储空间）是否相等，又可分为**定长记录**和**可变长记录**两种。

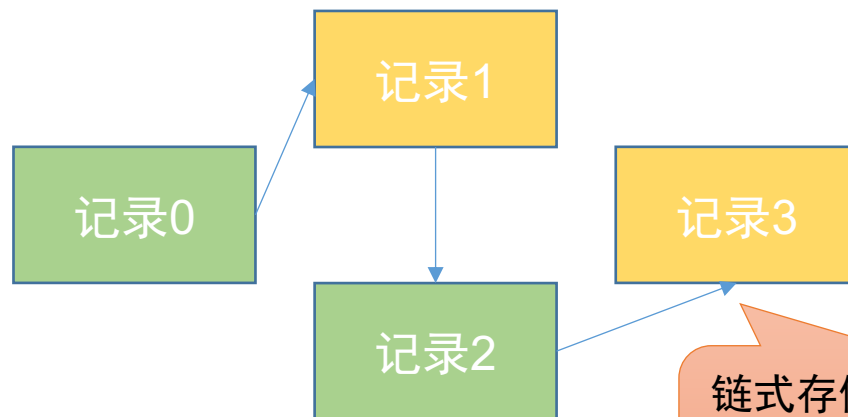


文件的逻辑结构——顺序文件

顺序文件：文件中的记录一个接一个地顺序排列（逻辑上），记录可以是**定长的**或**可变长的**。各个记录在物理上可以**顺序存储**或**链式存储**。

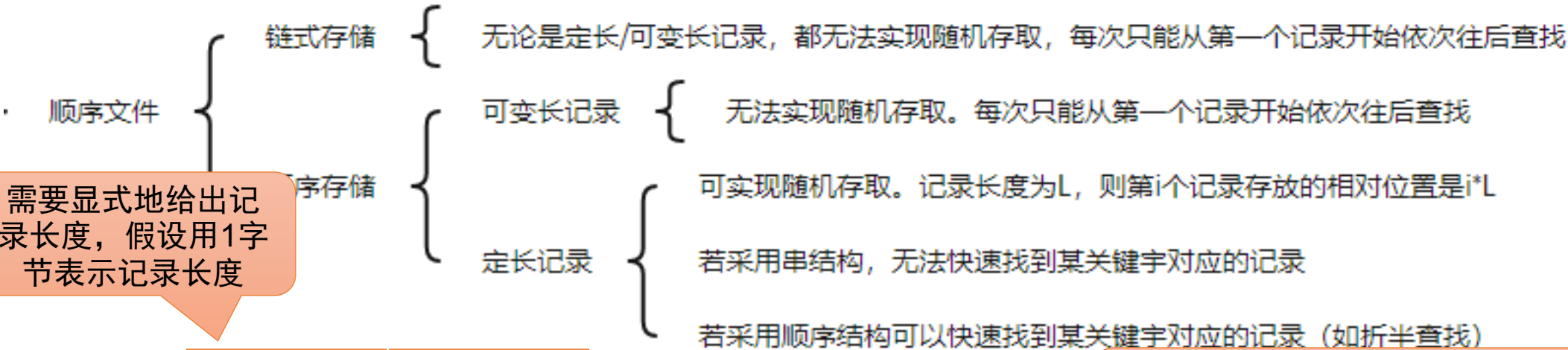


顺序存储——逻辑上相邻的记录物理上也相邻（类似于顺序表）



链式存储——逻辑上相邻的记录物理上不一定相邻（类似于链表）

文件的逻辑结构——顺序文件



需要显式地给出记录长度，假设用1字节表示记录长度

	记录长度	记录内容
0		
L_0+1	L_0	R_0
	L_1	R_1

$L_0+L_1+\dots+i$	L_i	R_i

可变长记录

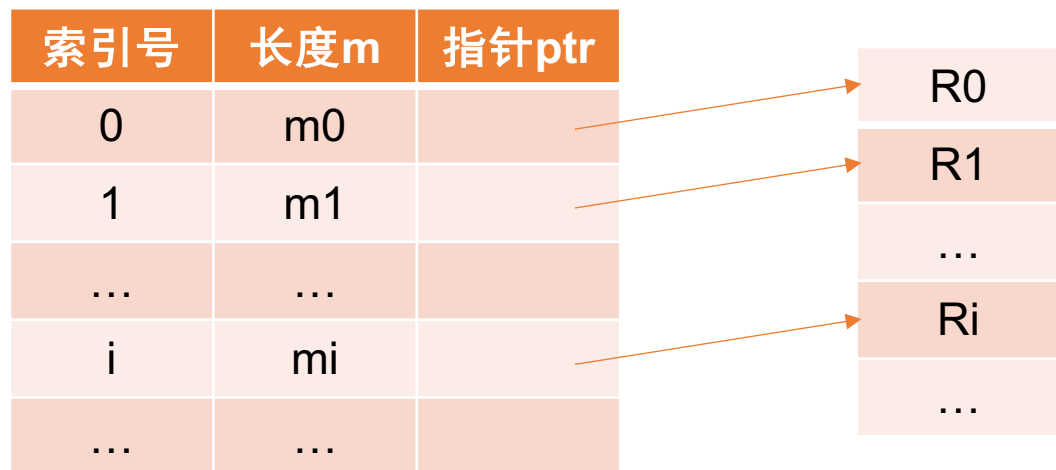
	记录内容
0	
	R_0
L	R_1
	...
$i*L$	R_i
	...

定长记录

结论：定长记录的顺序文件，若物理上采用顺序存储，则可实现随机存取；若能再保证记录的顺序结构，则可实现快速检索（即根据关键字快速找到对应记录）

注：一般来说，考试题目中所说的“顺序文件”指的是物理上顺序存储的顺序文件。之后的讲解中提到的顺序文件也默认如此。可见，顺序文件的缺点是增加/删除一个记录比较困难（如果是串结构则相对简单）

文件的逻辑结构——索引文件



建立一张索引表以加快文件检索速度。每条记录对应一个索引项。

文件中的这些记录在物理上可以离散地存放。

索引表本身是定长记录的顺序文件。因此可以快速找到第*i*个记录对应的索引项。可将关键字作为索引号内容，若按关键字顺序排列，则还可以支持按照关键字折半查找。每当要增加/删除一个记录时，需要对索引表进行修改。由于索引文件有很快的检索速度，因此主要用于对信息处理的及时性要求比较高的场合。

另外，可以用不同的数据项建立多个索引表。如：学生信息表中，可用关键字“学号”建立一张索引表。也可用“姓名”建立一张索引表。这样就可以根据“姓名”快速地检索文件了。（Eg: SQL就支持根据某个数据项建立索引的功能）

文件的逻辑结构——索引顺序文件

键	地址
A Ke	
Bai Qi	
Dian Wei	...
Cao Cao	...
...	...

姓名	其他属性
A ke	
An Qi La	
...	...

姓名	其他属性
Bai Qi	
Bu Zhi Huo Wu	
...	...

索引顺序文件是索引文件和顺序文件思想的结合。索引顺序文件中，同样会为文件建立一张索引表，但不同的是：并不是每个记录对应一个索引表项，而是一组记录对应一个索引表项。

索引顺序文件的索引项也不需要按关键字顺序排列，这样可以极大地方便新表项的插入

逻辑文件

在本例中，学生记录按照学生姓名的开头字母进行分组。每个分组就是一个顺序文件，分组内的记录不需要按关键字排序

文件的逻辑结构——索引顺序文件（检索效率分析）

若一个顺序文件有10000个记录，则根据关键字检索文件，只能从头开始顺序查找（这里指的并不是定长记录、顺序结构的顺序文件），平均须查找5000个记录。

键	地址	姓名	其他属性
A Ke		A Ke	
Bai qi		An Qi La	
Dian Wei
Cao Cao	...	姓名	其他属性
...	...	Bai Qi	
		Bu Zhi Huo Wu	
	

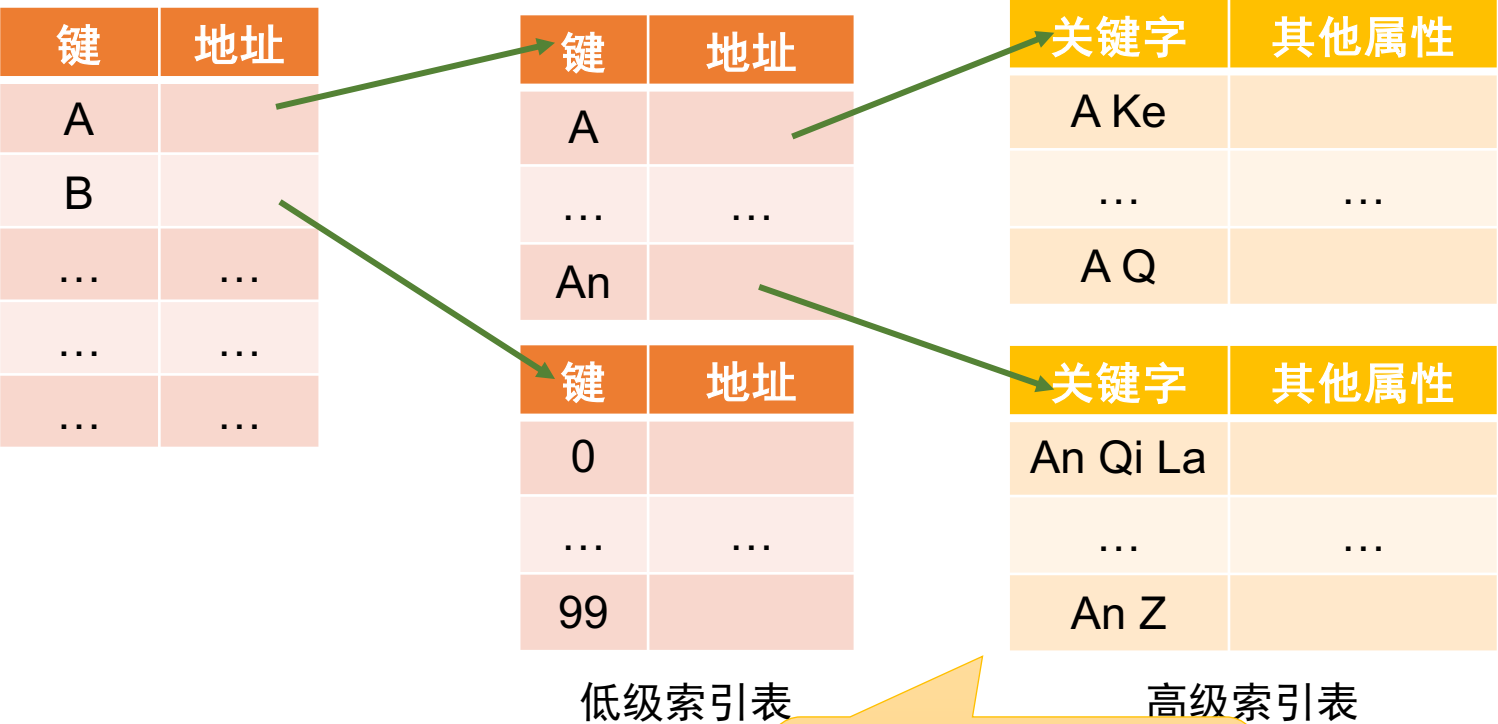
若采用索引顺序文件结构，可把10000个记录分为 $10000^{1/2} = 100$ 组，每组100个记录。则需要先顺序查找索引表找到分组（共100个分组，因此索引表长度为100，平均需要查50次），找到分组后，再在分组中顺序查找记录（每个分组100个记录，因此平均需要查50次）。可见，采用索引顺序文件结构后，平均查找次数减少为 $50+50 = 100$ 次。

逻辑文件

同理，若文件共有 10^6 个记录，则可分为1000个分组，每个分组1000个记录。根据关键字检索一个记录平均需要查找 $500+500 = 1000$ 次。这个查找次数依然很多，如何解决呢？

文件的逻辑结构——多级索引顺序文件

为了进一步提高检索效率，可以为顺序文件**建立多级索引表**。例如，对于一个含10000个记录的文件，可先为该文件建立一张低级索引表，每100个记录为一组，故低级索引表中共有10000个表项（即10000个定长记录），再把这10000个定长记录分组，每组100个，为其建立顶级索引表，故顶级索引表中共有100个表项。



Tips: 要为N个记录的文件建立K级索引，则最优的分组是每组 $N^{1/(k+1)}$ 个记录。

检索一个记录的平均查找次数是 $((N^{1/(k+1)})/2) * (K+1)$

如：本例中，建立2级索引，则最优分组为每组 $100000^{1/3} = 100$ 个记录，平均查找次数是 $(100/2) * 3 = 150$ 次

此时，检索一个记录
平均需要查找
 $50+50+50 = 150$ 次

文件目录——文件控制块



目录文件

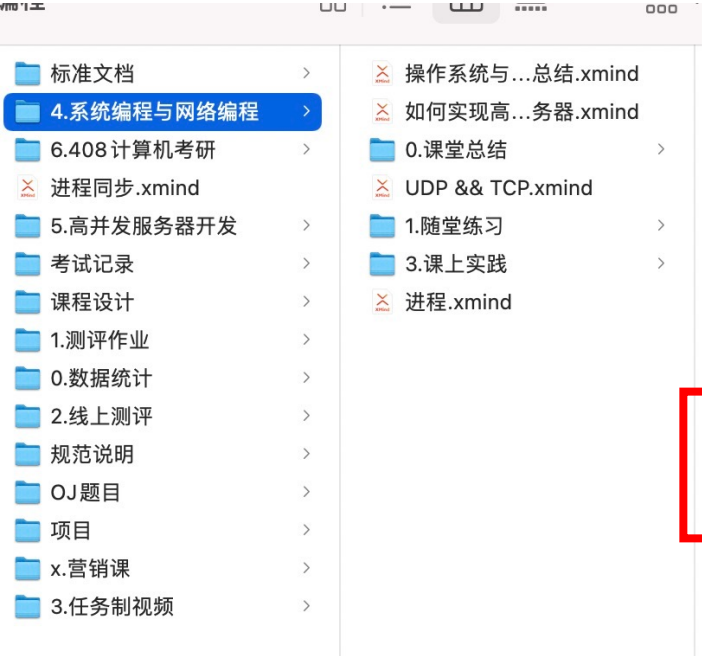
目录本身就是一种有结构文件，由一条条记录组成。每条记录对应一个在该放在该目录下的文件

文件名	类型	存取权限	物理位置
标准文档	目录	只读	...	外存7号块
考试记录	目录	读/写	...	外存18号块
.....			...	
4.系统编程与网络编程	目录	读/写	...	外存643块
.....				
进程同步.xmind	Xmind	只读	...	外存324块

当我们双击“4. 系统编程与网络编程”后，操作系统会在这个目录表中找到关键字“4. 系统编程与网络编程”对应的目录项（也就是记录），然后从外存中将“4. 系统编程与网络编程”目录的信息读入内存，于是，“4. 系统编程与网络编程”目录中的内容就可以显示出来了。

文件目录——文件控制块

“4.系统编程与网络编程”目录对应的目录文件



文件名	类型	存取权限	物理位置
0.课堂总结	目录	只读	...	外存25号块
1.随堂练习	目录	读/写	...	外存26号块
.....			...	
3.课上实践	目录	读/写	...	外存27号块
.....				
进程.xmind	xmind	只读	...	外存995号块

目录文件中的一条记录就是一个“文件控制块（FCB）”

FCB实现了文件名和文件之间的映射。使用户（用户程序）可以实现“按名存取”

FCB的有序集合称为“文件目录”，一个FCB就是一个文件目录项。FCB中包含了文件的基本信息（文件名、物理地址、逻辑结构、物理结构等），存取控制信息（是否可读/可写、禁止访问的用户名单等），使用信息（如文件的建立时间、修改时间等）。
最重要，最基本的还是文件名、文件存放的物理地址。

文件目录——文件控制块

文件名	类型	存取权限	物理位置
0.课堂总结	目录	只读	...	外存25号块
1.随堂练习	目录	读/写	...	外存278号块
.....			...	
3.课上实践	目录	读/写	...	外存152号块
.....				
进程.xmind	xmind	只读	...	外存995号块

需要对目录进行哪些操作？

搜索：当用户要使用一个文件时，系统要根据文件名搜索目录，找到该文件对应的目录项

创建文件：创建一个新文件时，需要在其所属的目录中增加一个目录项

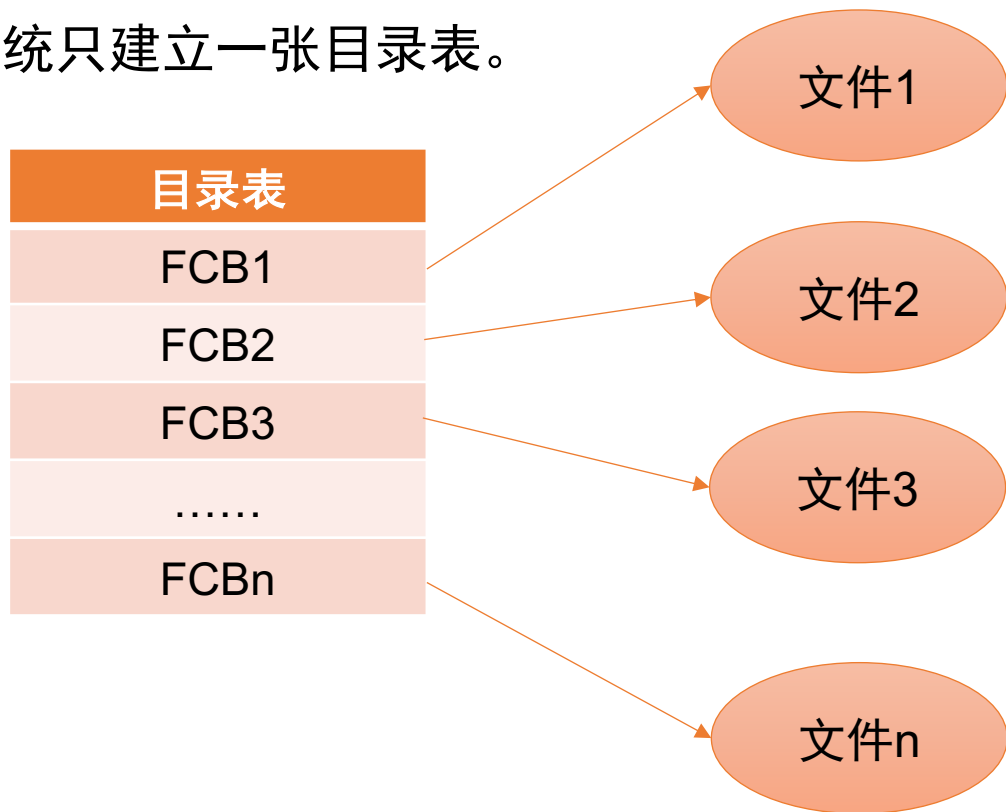
删除文件：当删除一个文件时，需要在目录中删除相应的目录项

显示目录：用户可以请求显示目录的内容，如显示该目录中的所有文件及相应属性

修改目录：某些文件属性保存在目录中，因此这些属性变化时需要修改相应的目录项（如：文件重命名）

文件目录——单级目录结构

单级目录结构：整个文件系统只建立一张目录表。



优点：

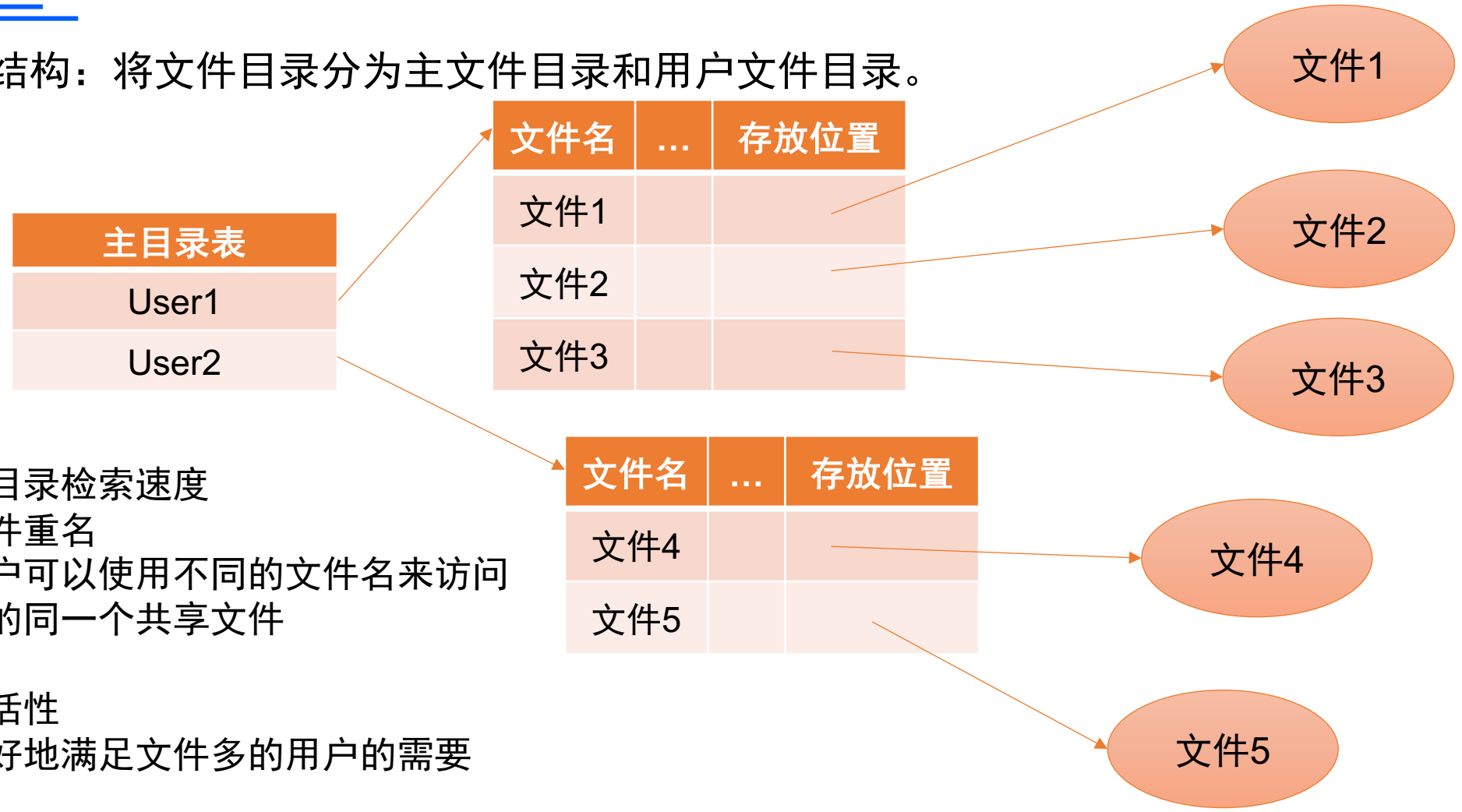
1. 实现简单
2. 能实现按名存取

缺点：

1. 查找速度慢；
2. 不允许重名；
3. 不便于实现文件共享

文件目录——两极目录结构

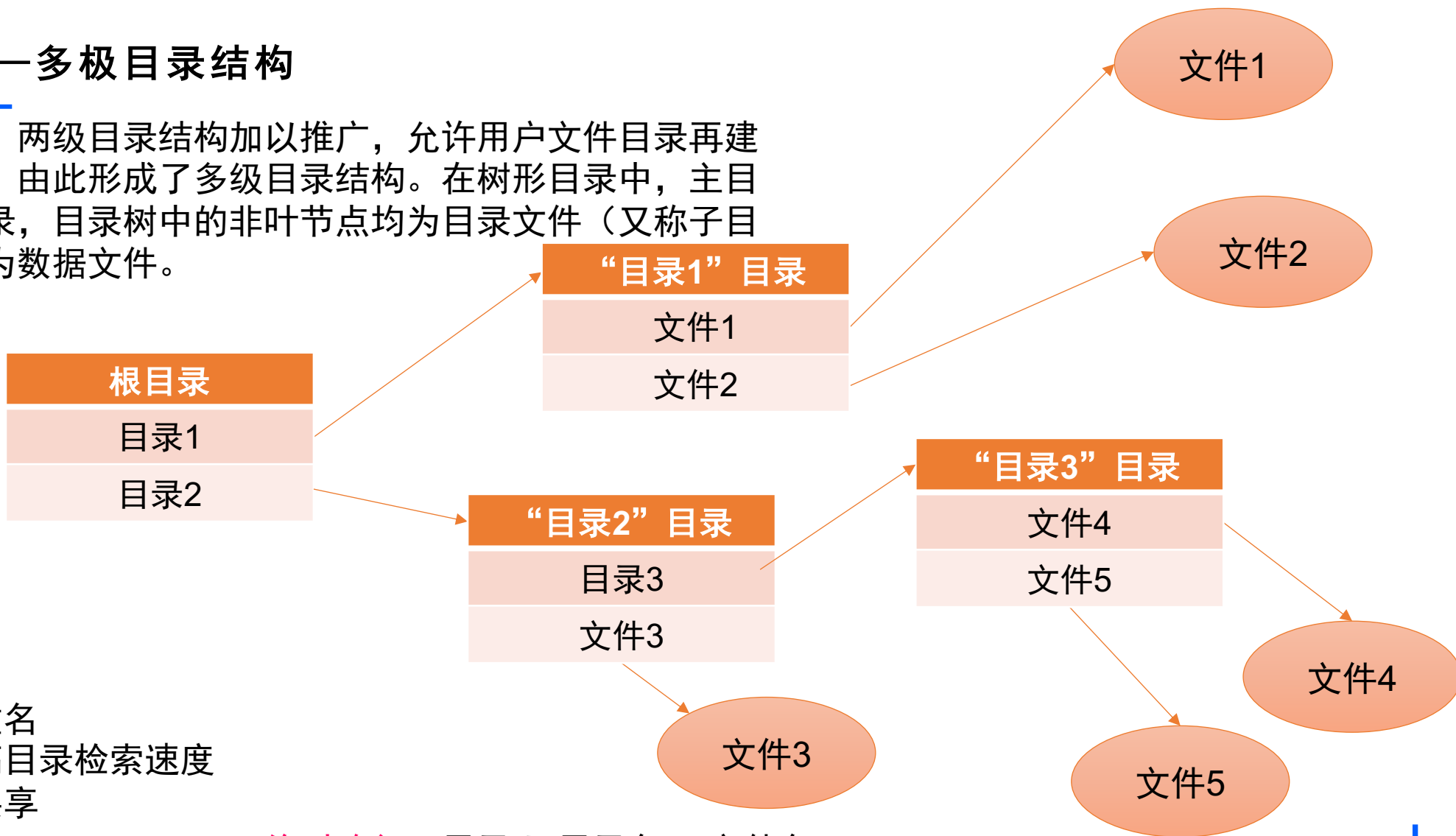
单级目录结构：将文件目录分为主文件目录和用户文件目录。



- 优点：
- 1. 提高了目录检索速度
 - 2. 允许文件重名
 - 3. 不同用户可以使用不同的文件名来访问系统中的同一个共享文件
- 缺点：
- 1. 缺乏灵活性
 - 2. 无法很好地满足文件多的用户的需要

文件目录——多极目录结构

多级目录结构：两级目录结构加以推广，允许用户文件目录再建立下级子目录，由此形成了多级目录结构。在树形目录中，主目录则称为根目录，目录树中的非叶节点均为目录文件（又称子目录），叶节点为数据文件。



优点:

1. 层次清楚
2. 允许文件重名
3. 进一步提高目录检索速度
4. 容易实现共享

- **绝对路径:** 目录/子目录名.../文件名
- **相对路径:** 当前目录/子目录名.../文件名

文件目录——无环图目录结构

可以用不同的文件名指向同一个文件，甚至可以指向同一个目录（共享同一目录下的所有内容）。

需要为每个共享结点设置一个共享计数器，用于记录此时有多少个地方在共享该结点。用户提出删除结点的请求时，只是删除该用户的FCB、并使共享计数器减1，并不会直接删除共享结点。

只有共享计数器减为0时，才删除结点。

注意：共享文件不同于复制文件。在共享文件中，由于各用户指向的是同一个文件，因此只要其中一个用户修改了文件数据，那么所有用户都可以看到文件数据的变化。

文件目录——索引结点（FCB的改进）

文件名	类型	存取权限	物理位置
标准文档	目录	只读	...	外存7号块
考试记录	目录	读/写	...	外存18号块
.....			...	
4.系统编程与网络编程	目录	读/写	...	外存643块
.....				
进	文件名	索引结点指针	...	外存324块

索引结点

除了文件名之外的文件描述信息都放到这里来

其实在查找各级目录的过程中只需要用到“文件名”这个信息，只有文件名匹配时，才需要读出文件的其他信息。因此可以考虑让目录表“瘦身”来提升效率。

思考有何好处？
假设一个FCB是64B，磁盘块的大小为1KB，则每个盘块中只能存放16个FCB。若一个文件目录中共有640个目录项，则共需要占用640/16 = 40个盘块。因此按照某文件名检索该目录，平均需要查询320个目录项，平均需要启动磁盘20次（每次磁盘I/O读入一块）。
若使用索引结点机制，文件名占14B，索引结点指针站2B，则每个盘块可存放64个目录项，那么按文件名检索目录平均只需要读入320/64 = 5个磁盘块。显然，这将大大提升文件检索速度。

文件目录——索引结点（FCB的改进）

文件名	索引结点指针
标准文档	
考试记录	
.....	
4.系统编程与网络编程	
.....	

索引结点
(包含除了文件名之外的文件描述信息)

思考有何好处?

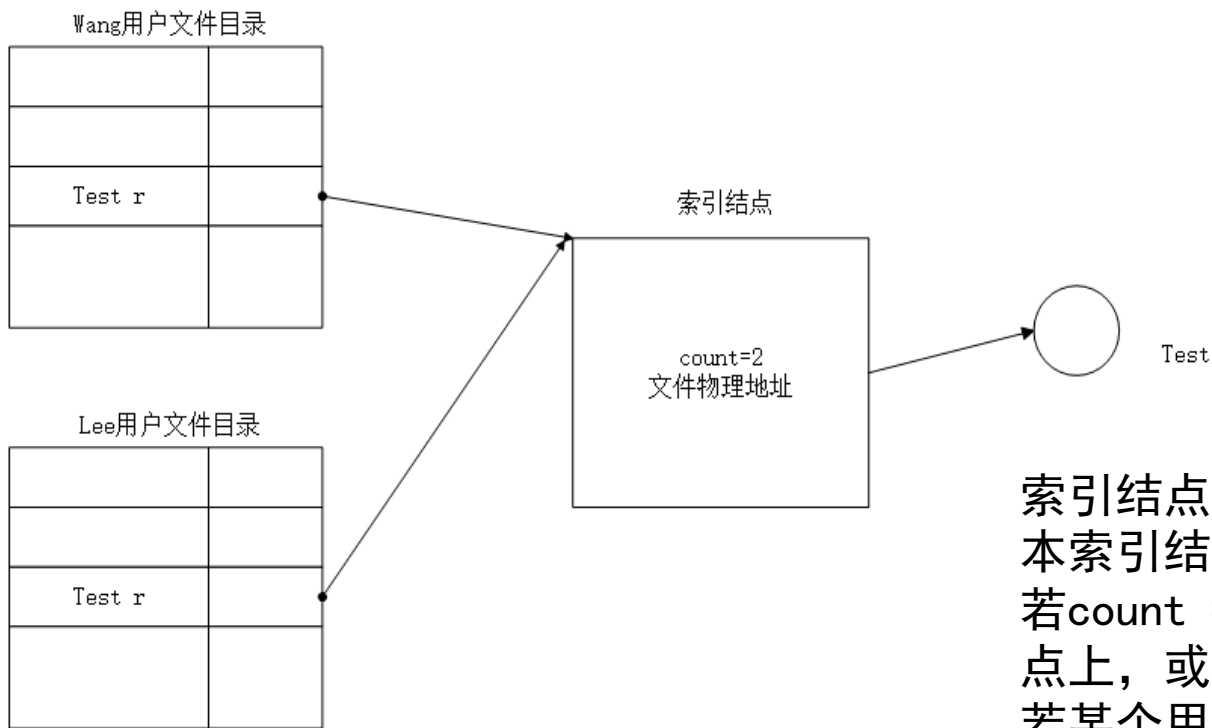
假设一个FCB是64B, 磁盘块的大小为1KB, 则每个盘块中只能存放16个FCB。若一个文件目录中共有640个目录项, 则共需要占用 $640/16 = 40$ 个盘块。因此按照某文件名检索该目录, 平均需要查询320个目录项, 平均需要启动磁盘20次(每次磁盘I/O读入一块)。

若使用索引结点机制, 文件名占14B, 索引结点指针站2B, 则每个盘块可存放64个目录项, 那么按文件名检索目录平均只需要读入 $320/64 = 5$ 个磁盘块。显然, 这将大大提升文件检索速度。

当找到文件名对应的目录项时, 才需要将索引结点调入内存, 索引结点中记录了文件的各种信息, 包括文件在外存中的存放位置, 根据“存放位置”即可找到文件。

存放在外存中的索引结点称为“磁盘索引结点”, 当索引结点放入内存后称为“内存索引结点”。相比之下内存索引结点中需要增加一些信息, 比如: 文件是否被修改、此时有几个进程正在访问该文件等。

文件共享——基于索引结点的共享方式（硬链接）



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。

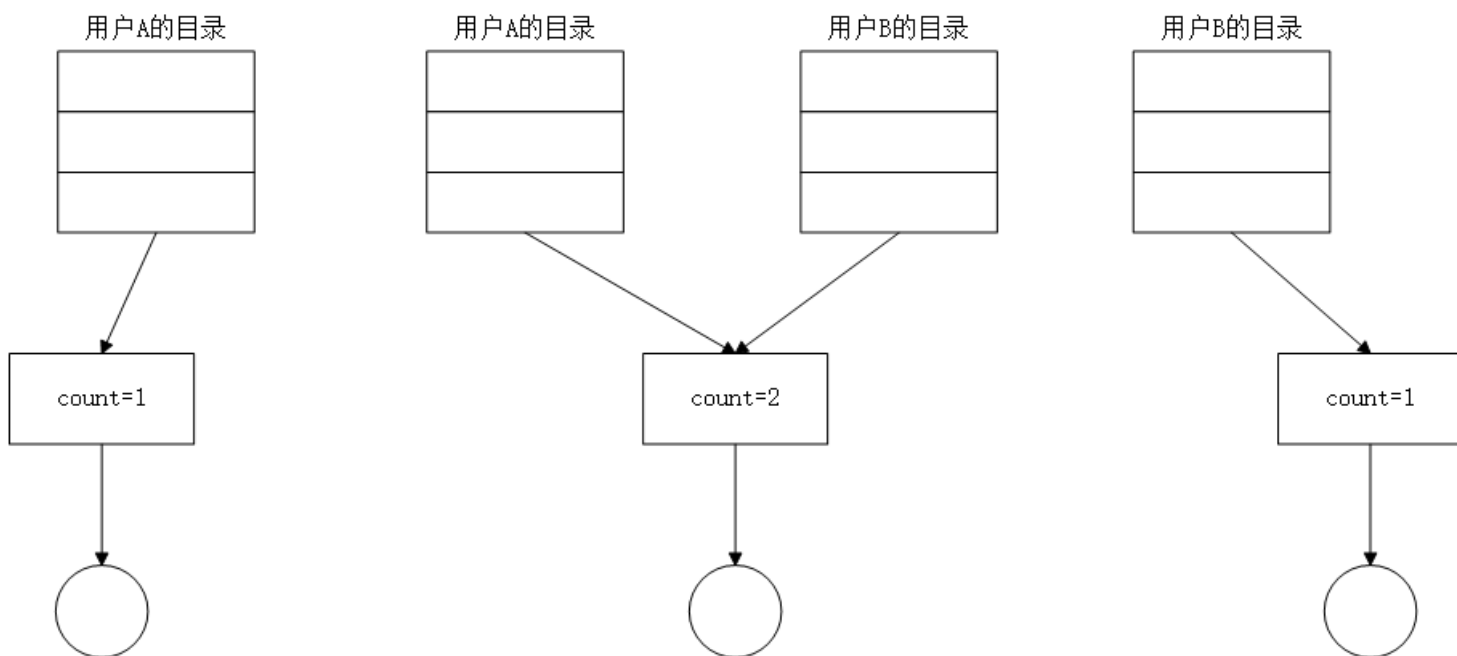
若count = 2，说明此时有两个用户目录项链接到该索引结点上，或者说是有两个用户在共享此文件。

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。

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。

文件共享——基于符号链的共享方式（软链接）



为共享文件创建一个link类型的新文件：
分配并填写一个空闲i节点；
建立目录项；
分配磁盘空间；
写入文件内容：共享文件的路径名

拜拜

