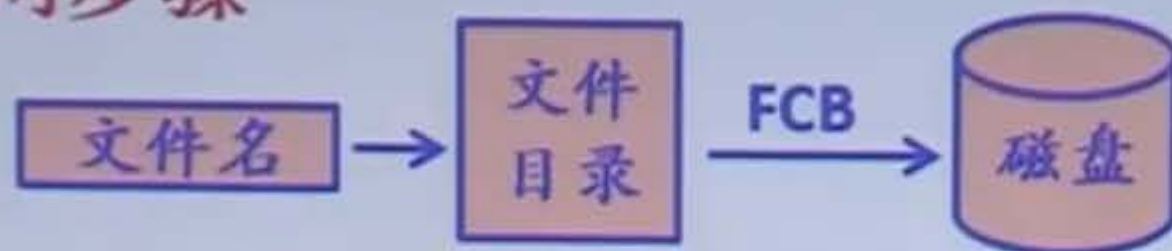


文件目录检索

访问一个文件 → 两步骤



➤ 目录检索

用户给出文件名 → 按文件名查找到目录项/FCB

根据路径名检索:

全路径名: 从根开始 **\A\B\C\File1**

相对路径: 从当前目录开始 **C\File1**

➤ 文件寻址

根据**目录项/FCB**中文件物理地址等信息, 计算出文件中任意记录或字符在存储介质上的地址



目录文件实现时的改进

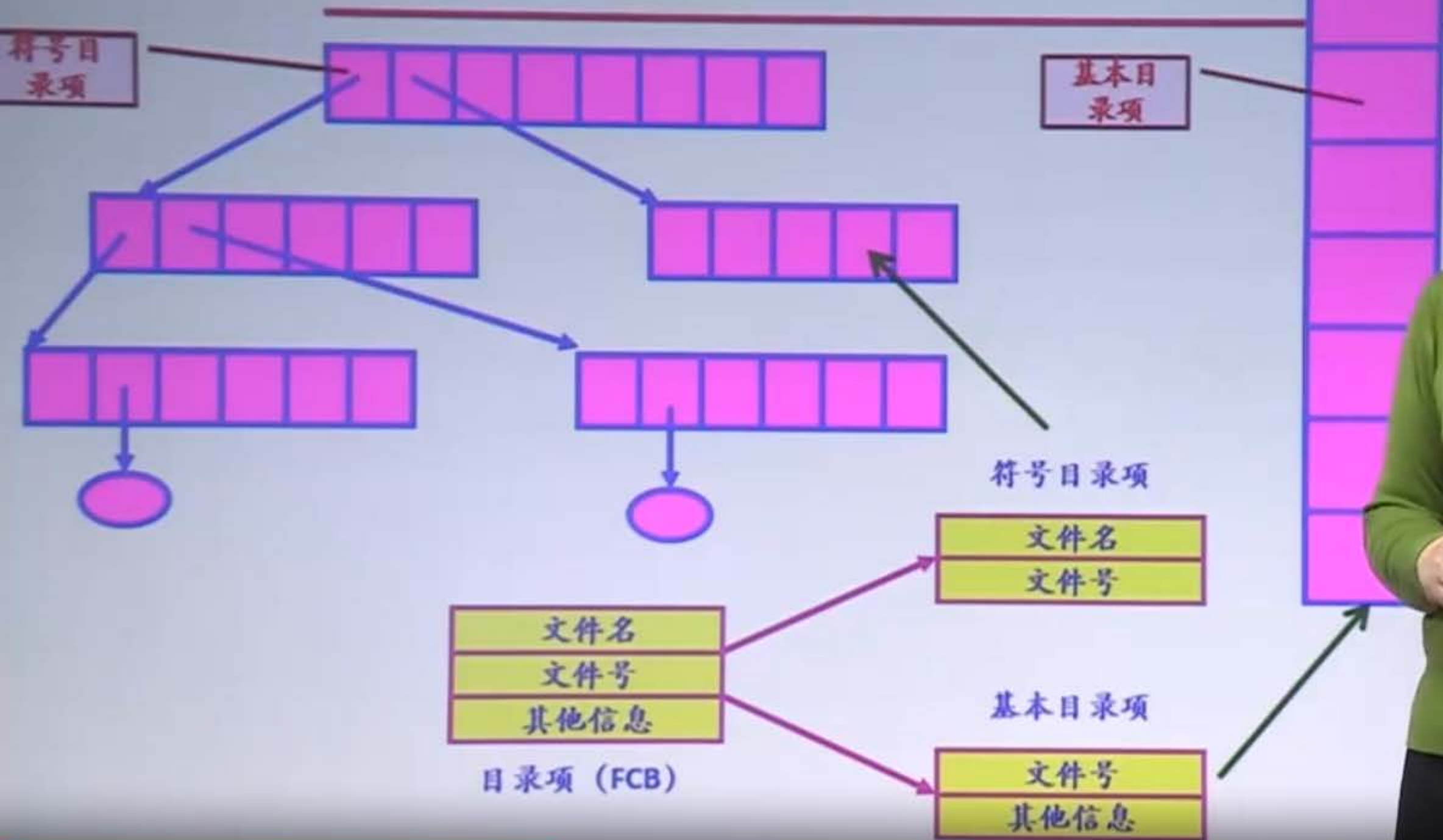
- 提问：如何加快目录检索？
- 一种解决方案：
目录项分解法：即把**FCB**分成两部分
 - 符号目录项
文件名，文件号
 - 基本目录项
除**文件名**外的所有字段

例子：UNIX的I节点（索引节点或inode）

节点 也叫索引节点。



改进方案示意



改进后的好处

分解前：占13块

分解后：符号文件占2块
基本文件占11块

例子：

假设 一个FCB 占48个字节，物理块大小512字节
符号目录项占8字节（文件名6字节，文件号2字节）
基本目录项占 $48 - 6 = 42$ 字节

一个目录文件有128个目录项

查找一个文件的平均访盘次数：

分解前：7次

分解后：2.5次

目录文件改进后减少了访盘次数，提高了文件检索速度



UNIX文件系统(1/3)

- FCB = 目录项 + i节点
- 目录项: 文件名 + i节点号
- 目录文件由目录项构成
- i节点: 描述文件的相关信息
- 每个文件由 一个目录项、一个i节点 和 若干磁盘块 构成

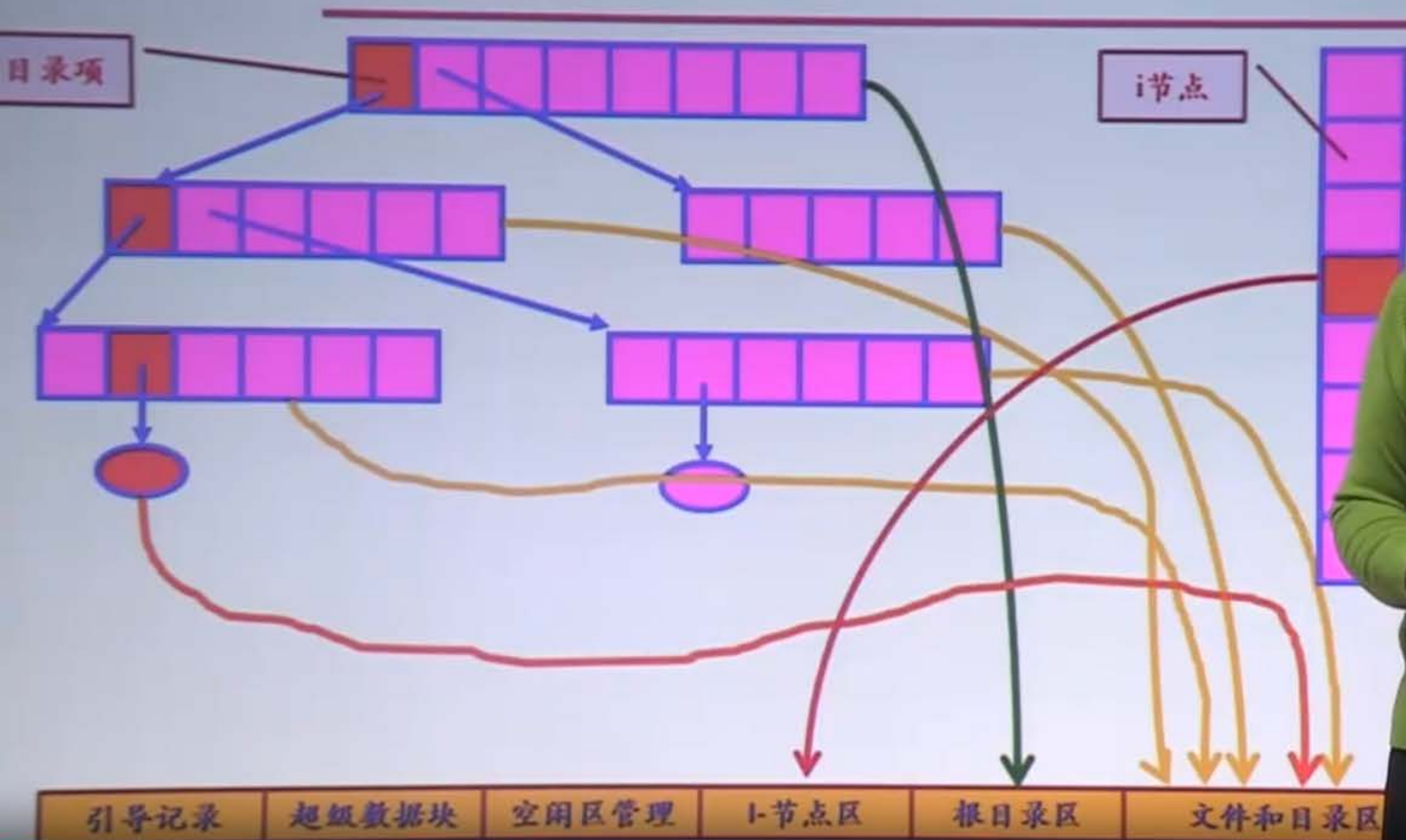


.....

引导记录	超级数据块	空闲区管理	i-节点区	根目录区	文件和目录区
------	-------	-------	-------	------	--------



UNIX文件系统(2/3)



UNIX文件系统(3/3)

UNIX的I节点
参考教材
4.5.3

根目录文件

1	.
1	..
6	bin
8	dev
12	etc
5	lib
7	usr
9	tmp

查找根目录
得到usr的
I节点7

/usr/ast/mbox

I节点7

128

记录了usr
在128中

块128:
usr目录文件

7	.
1	..
29	clan
41	ned
62	ast
34	hat

usr/ast/的
I节点62

I节点62

496

记录了usr/ast
在496中

块496:

usr/ast目录文件

62	.
7	..
65	cloud
80	mbox
18	rain
99	wind

usr/ast/mbox
的I节点80

引导记录

超级数据块

空闲区管理

I-节点区

根目录区

文件和目录区

本讲重点

- 掌握文件系统涉及的相关概念
文件、文件分类、文件的逻辑结构和物理结构
- 掌握文件目录的实现
文件控制块、目录项、目录文件
- 掌握文件系统的实现
磁盘布局、内存数据结构
- 了解磁盘空间的管理
存储介质、扇区、物理块、簇



物理块、簇这样一些概念 这里列出了本讲重点阅读的教材内容

本周要求

- 重点阅读教材

第4章相关内容: 4.1、4.2、4.3、4.5.3

- 重点概念

文件 文件系统 UNIX文件分类 文件逻辑
存储介质(磁盘) 物理块(块、簇) 文件物理结构
文件控制块FCB(文件属性) 文件目录 目录项 目录项
磁盘上文件系统的布局 内存中数据结构
磁盘空间管理(位图、空闲块表、成组链接法)
目录项分解法

物理块、簇这样一些概念 这里列出了本讲重点阅读的教材内容



下面我们以 UNIX 为例，介绍一下文件系统的实现。首先我们先来讨论一下文件目录检索的过程。当要访问一个文件的时候呢，用户通常给出了一个文件名，那么用文件名，最后把这个文件的相关内容读进内存，通常需要两个步骤。第一个步骤呢，我们叫做目录检索。目录检索主要是根据用户给出的文件名，去查找这个文件目录，找到这个文件所对应的 FCB 或目录项。那么这个查找的过程，要根据你提供的文件名，如果你提供的文件名是一个全路径名，就要从根目录开始查找。如果你提供的是一个相对路径名，就从当前目录开始查找。我们以全路径名为例，当要查找这样一个文件的时候，A\B\C\File1 的时候，我们首先要用，在根目录里头去找 A，找到 A 了之后呢，在 A 里头再找 B，在 B 里头呢再找 C，然后在 C 里头再找 File1，所以它有这样一个按数、按层来查找的这样一个过程。通过这样一个过程最后才找到了 File1 的什么呢？FCB，找到了 FCB，我们就进行下一个步骤，就叫文件寻址。因为只要有了 FCB，或者目录项，我们就得到了这个文件的物理地址。我们就可以算出来任何这个文件的一个记录，一个字符在存储介质上的物理地址了。所以目录检索是非常重要的。那么这个过程，它的速度，它的性能直接影响了你对文件的一个处理的一个整体性能。所以现在呢我们就来探讨怎么样去来加快目录检索的速度。所以我们的问题就是：如何加快目录检索？我们这里给出了一种叫做目录项分解法，也就是把 FCB 分解成两部分。原来 FCB 保存了这个文件的很多的信息。现在我们把这个 FCB 分解成两个部分，第一个部分我们叫做符号目录项，只包括了文件名和文件号。第二个部分呢叫基本目录项，包括了除了文件名之外的其他的所有的信息。那么 UNIX 其实就是采用了这样一种目录项分解法的方式来实现目录文件的。当然它的基本目录项呢，它的名字叫什么？叫做 I 节点，也叫索引节点。下面呢我们来看一下目录项分解法的一个图示。在这个树形结构当中，这些代表的是目录文件，它由目录项组成。这个圆的，圆圈呢代表的是普通文件。由于目录文件是由目录项组成，在没有分解之前，目录项就等于 FCB。它的内容呢是一个很多的信息，除了文件名、文件号之外，还有大量的其他的一些文件属性。如果我们采用目录项分解法把它分解成两部分，那么符号目录项里头呢只有简单的文件名和文件号两部分内容。那么经过了这个目录项分解法，原来的我们的每一个目录文件的目录项呢实际上就瘦身了，就变成了很小的内容，因此呢现在我们把它叫做什么呢？符号目录项，而基本目录项呢，那么就需要另找一块地方存放。通常呢我们是保存在了磁盘的一个专用的区域。那么每一个格子呢实际上就代表的的是一个基本的目录项，这就是目录项分解法的一个图示。那么采用

了目录项分解法会带来什么好处呢？假设，一个 FCB 占了 48 个字节 物理块大小是 512 字节，如果我把它分解成两部分 那么符号目录项呢是占 8 个字节，文件名 6 个字节，文件号 2 个字节 基本目录项呢是占 42 个字节 我们现在有一个文件，目录文件 那么它呢包含了 128 个目录项 那么如果分解前，要把这个 目录文件保存在磁盘上，需要 13 个物理框 分解之后呢，那么这个目录文件其实就分成了两处保存 一部分呢叫做符号文件，需要两块存放 另外一部分叫做基本文件，需要 11 块 总块数其实并没有减少。但是我们来看一下 要查找这个文件，它的平均访盘次数 在分解前和分解后是不一样的。我们来看看分解前 分解前，我们有 13 块要查找 最好的情况在第一块里就找到了这个文件的相关信息 最差的情况是在最后一块才 找到这个文件的相关信息，或者甚至这个文件根本就不存在 那么平均的访盘次数就是 $1+13$ ，除以 2 呢，我们就来一个简单平均，就是 7 次 那么分解之后呢，我们要想查找这个文件是否存在 我们只需要在符号文件当中 找就可以了。符号文件占两块，所以 最好的情况在第一块，最差的情况在第二块 因此呢，平均的访盘的次数呢是 1.5 次 但是我们找到了这个符号 目录项之后，我们还要找到它的基本目录项，我们要根据文件的号 再到基本文件当中再次进行查找，找到它的其余的信息 因此呢这需要一次，因此呢在分解后需要 2.5 次 这样一个访盘的次数 通过这样一个计算，我们可以看到 采用了目录项分解法实际上是减少了访盘的次数 当然了，访盘是花费时间最大的，因此呢也就是提高了 检索文件目录的性能。

下面呢我们介绍 UNIX 文件系统的实现 在 UNIX 文件系统当中的 FCB 划分成两部分，一部分叫目录项 一部分叫 i 节点。UNIX 的 目录项呢是由文件名和 i 节点号组成的 在 UNIX 文件系统当中的目录文件就是由这样的目录项构成的 而 i 节点里头保存的是 管理这个文件所需要的其他的相关信息 在 UNIX 文件系统当中的每一个文件 目录文件和普通文件都是由这样几部分内容组成 第一个呢是要有一个目录项，任何一个文件 都对应了一个目录项，包括了这个文件的名字和它 i 节点号 第二个是它的 i 节点 通过目录项的 i 节点可以找到对应的 i 节点的内容 这个内容就保存了相关的管理信息 然后这个文件的实际内容呢是保存在这些磁盘块上 那么假设这是一个文件卷，由若干个 磁盘块组成，那么 UNIX 的文件系统的布局呢实际上就是从 这个磁盘块的一开始按照规定 哪一部分占多少，然后就构成了这样一个文件系统的布局 下面我们就介绍一下 UNIX 文件系统的具体布局 这是一个树形结构，那么 在 UNIX 文件系统当中，把 FCB 分成了

目录项和 i 节点两部分。那么构成目录文件的基本单位呢就是目录项 包括了文件名和 i 节点号。而 i 节点呢是专门存放在一个区域 那么在这里头我们可以看到，那么这是 i 节点区专门存放 i 节点的。好，那么在这样一个布局的情况下，我们假设这是根目录文件，根目录文件呢就存放在了这个区域 如果根目录文件底下又建了一个目录文件。那么我们就可以往后保存这个目录文件了 好，我们接着来看。若干个目录建立，那么就在这个文件 文件卷的后面的区域去把这些文件保存起来 那么这个普通文件呢，也是建立了普通文件，也保存在这个后面的区域啊 这就是一个具体的文件系统的布局的示意 下面呢我们来介绍在这样一个 UNIX 文件系统的布局当中如何查找一个文件 我们要查找的文件是这样一个全路径名 要查找这个文件涉及到了三个目录文件 第一个是根目录文件，第二个是 `usr` 目录文件，第三个呢是 `ast` 目录文件 我们以根目录文件为例介绍一下 UNIX 目录文件的一个构成 UNIX 的目录文件当然是由若干目录项构成 每个目录项呢包含了文件名和 i 节点号两个信息 在任何一个目录文件当中的第一个目录项呢是 本目录文件的目录项，就是 `.` 那么第二行、第二个目录项呢是它的 附目录的目录项，就是 `..`，用 `..` 来表示附目录项 所以我们可以看到后面都是这样做的，然后就是你建的各种各样的目录文件或者是其它的文件 每个文件都有一个文件名和一个 i 节点号 那么这就是一个目录文件的一个构成 现在我们要查找这样一个文件，我们首先要在根目录 文件中进行查找，找什么呢？要找 `usr` 这个目录文件 它对应的目录项，那么通过了 超级数据块等相关数据结构的信息，我们可以知道 根目录文件从什么地方开始，我们可以把根目录文件读入内存 然后呢去查找我们要找的目录项 因为我们前面说过，每一个文件，不管你是目录文件和普通文件，都有三部分 信息，第一个是目录项，第二个是 i 节点，第三个是它所占用的这些文件内容所占用的磁盘块 所以我们在根目录文件区去查找 `usr` 这个目录文件的目录项 那我们通过去比较文件名我们找到了 `usr` 这个目录项 那么 `usr` 目录项，我们可以看到，它的 i 节点号是 7，所以我们通过了 到 i 节点区当中用 7 来去索引找到 `usr` 这个目录文件所对应的 i 节点，那么这个呢就是 `usr` 这个目录文件的 i 节点，从这个 i 节点当中我们得到了 `usr` 这个目录文件它的第一块存放在 是块 128 这个位置，所以呢我们又通过了 128 到磁盘的第 128 块里头把它读进内存，也就相当于把 `usr` 这个目录文件读进内存，那么这是 `usr` 目录文件的内容，那么在这里头呢，我们要找什么呢？我们要找这个文件当中 有没有 `ast` 这个目录

文件的目录项 通过查找我们得到了，噢，这是 ast 所对应的这个目录项 我们得到了 ast 这个目录文件的 i 节点号是 62 号 好，那么我们又从 i 节点区找 62 号 i 节点 把 62 号 i 节点读入内存。从这个节，i 节点当中我们知道了 ast 这个目录文件是保存在 496 块这个位置 然后呢，我们通过到磁盘的 496 块 查找把它读进内存，把 ast 目录文件读入内存 那么读入内存之后，我们就要去查找 mbox 这个文件的目录项，我们找到了它的目录项，知道它的 i 节点号是 80 然后我们继续，在 i 节点区用 80 去把 mbox 这个文件的啊 i 节点读进内存 然后，这个时候这个 i 节点里头就存放了我们需要的，处理这个 mbox 的文件的所有的信息 我们要读哪些内容？我们要写哪些内容？就都可以通过 i 节点 80 得到了 那么这部分内容呢，大家可以参考啊，教材当中的 4.5.3 节 本讲的内容呢 主要希望大家能够掌握文件系统所涉及到的相关的概念 非常重点的是，文件目录的实现 大家要搞清楚，什么是文件控制块？什么是目录项？什么是目录文件？还有一个重点呢，就是文件系统的实现要 了解啊，要掌握这个文件系统在磁盘上的布局 在内存的数据结构。我们还要 了解呢，磁盘空间的一些管理。比如说，磁盘，存储介质 典型的是磁盘，然后是扇区、物理块、簇这样一些概念 这里列出了本讲重点阅读的教材内容 也给出了啊，在本讲当中 需要大家掌握的重要的概念 好，今天的内容呢，我们就介绍到这里。谢谢大家！