

文件的物理结构

- 文件在存储介质上的存放方式

主要解决两个问题：

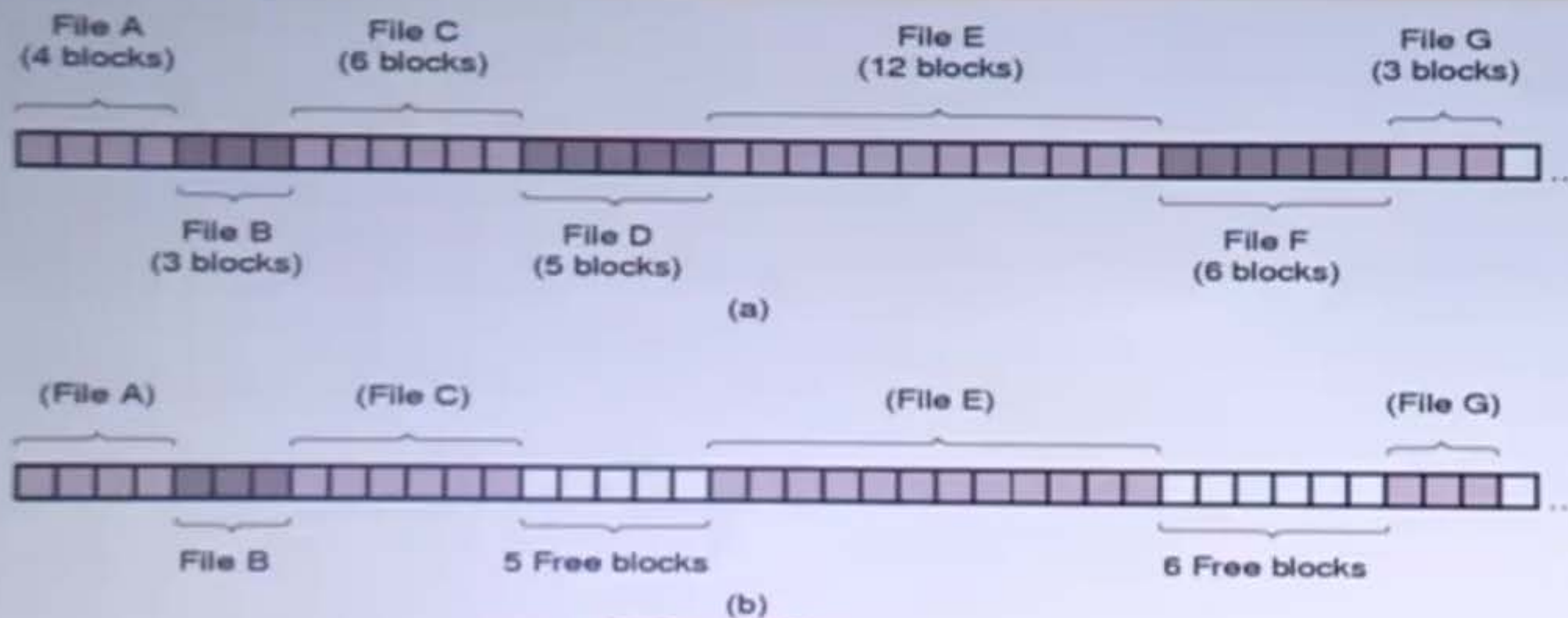
- 假设一个文件被划分成N块，这N块在磁盘上是怎么存放的？
- 其地址(块号或簇号)在FCB中是怎样记录的

当中是怎么记录的？

所以我们要解决这两个问题，针对文件的物理结构，我们要把这两个问题解决

连续(顺序)结构

- 文件的信息存放在若干连续的物理块中



在FCB中如何记录文件地址?

连续结构的优缺点

◎ 优点

- 简单
- 支持顺序存取和随机存取
- 所需的磁盘寻道次数和寻道时间最少
- 可以同时读入多个块，检索一个块也很容易

◎ 缺点

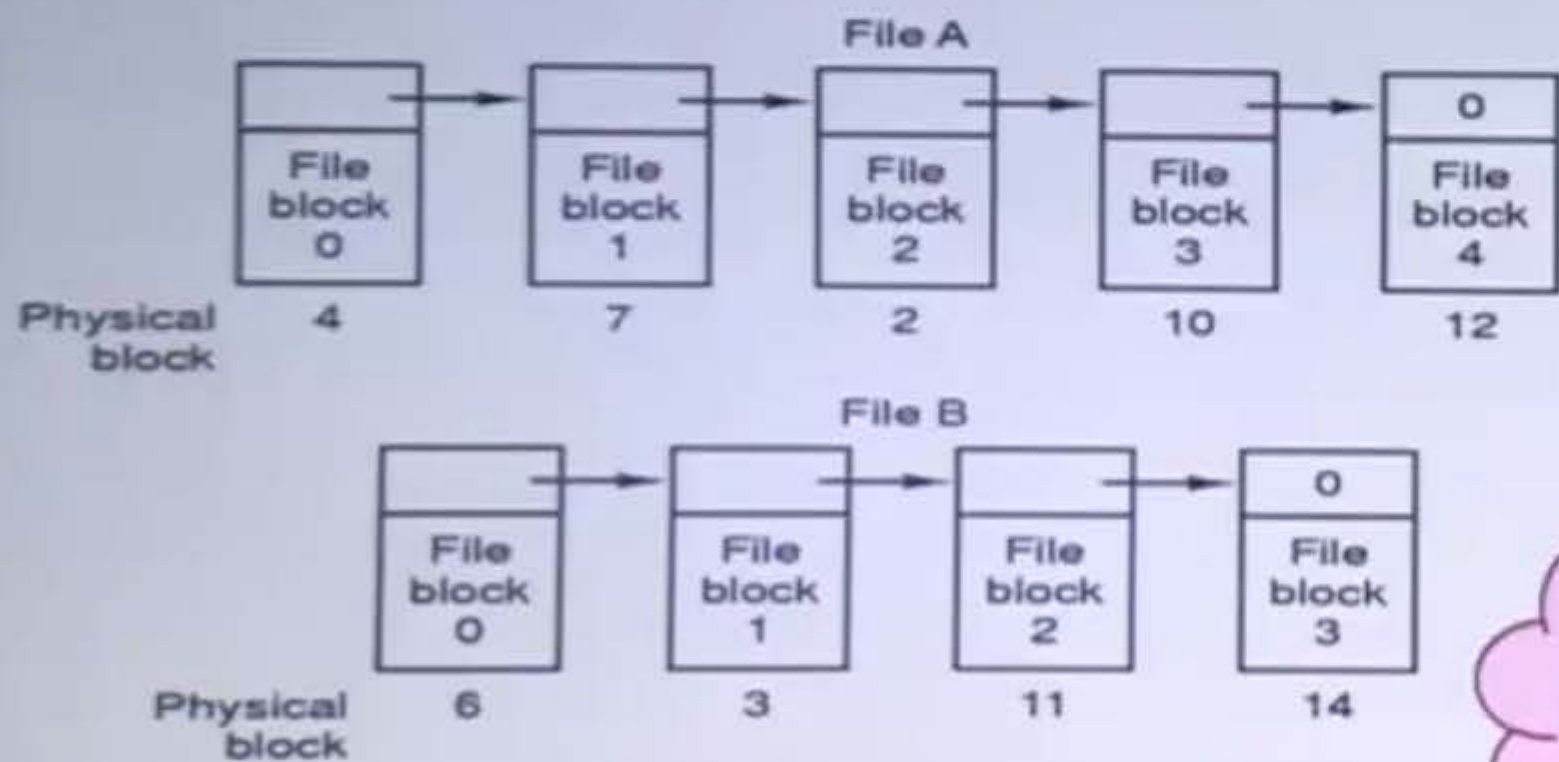
- 文件不能动态增长
预留空间：浪费 或 重新分配和移动
- 不利于文件插入和删除
- 外部碎片，紧缩技术



那么我们也可以通过紧缩技术来进行一个碎片整理 第二种文件的物理结构呢，是
链接结构

链接结构

- 一个文件的信息存放在若干不连续的物理块中，各块之间通过指针连接，前一个物理块指向下一个物理块



在FCB中如何记录文件地址？



链接结构的优缺点

◎ 优点

- 提高了磁盘空间利用率，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

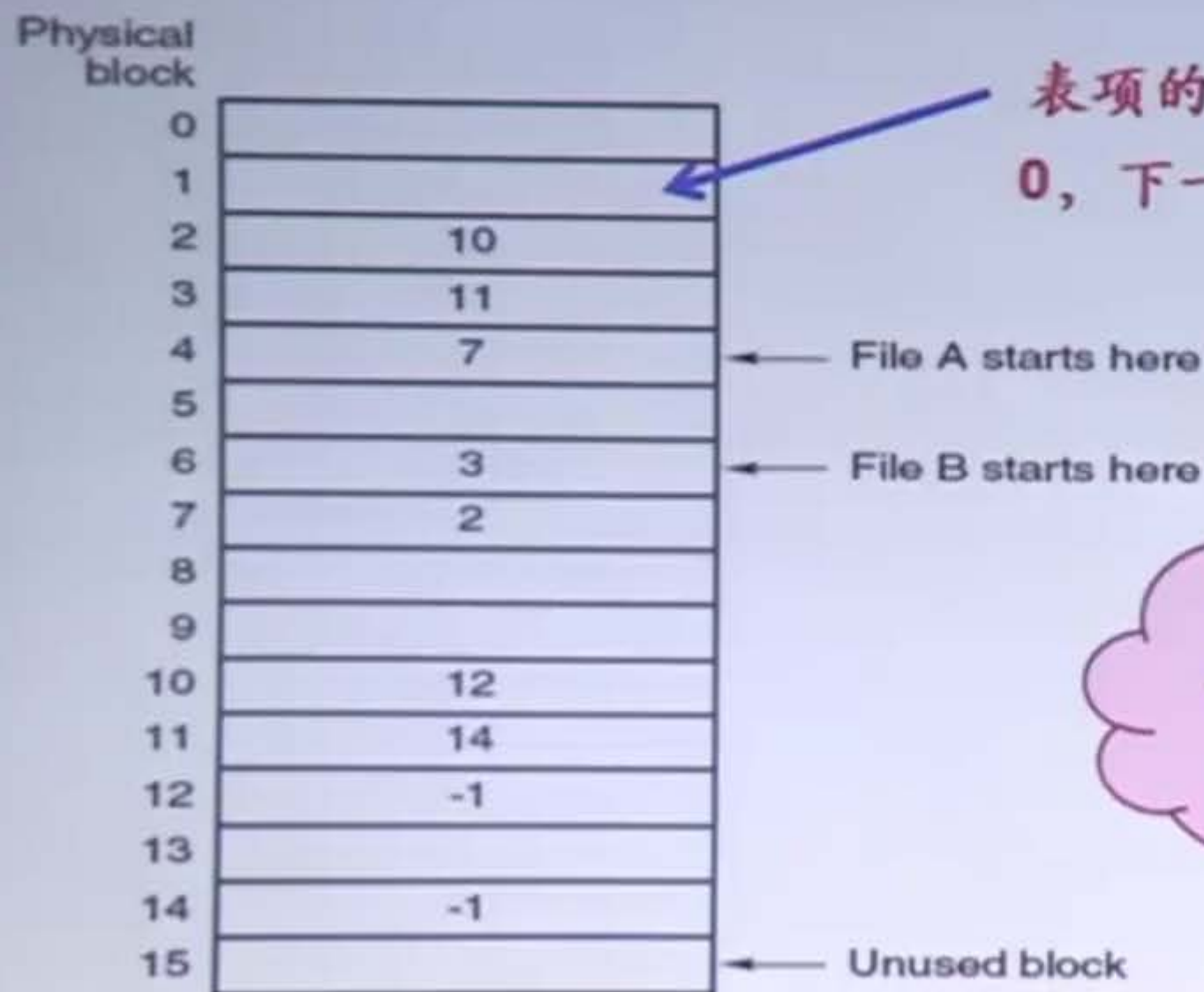
◎ 缺点

- 存取速度慢，不适于随机存取
- 可靠性问题，如指针出错
- 更多的寻道次数和寻道时间
- 链接指针占用一定的空间

链接结构的一个变形：
文件分配表
FAT

FAT 表，文件分配表，因此 对于磁盘上的任何一个物理块，我这表里头就有一个表项

文件分配表示意图



表项的值有三种：
0，下一块块号，-1

某文件的
始块号从何处得到？



索引结构

- 一个文件的信息存放在若干不连续物理块中
- 系统为每个文件建立一个专用数据结构——索引表，并将这些物理块的块号存放在该索引表中
- 索引表就是磁盘块地址数组，其中第 i 个条目指向文件的第 i 块

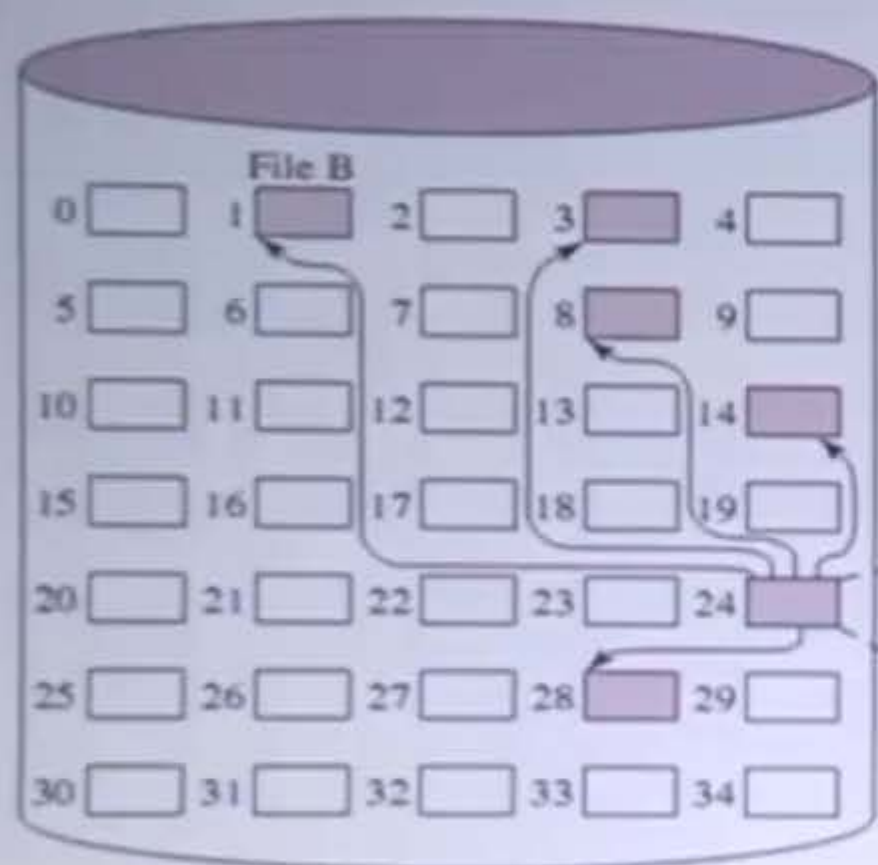
在FCB中如何记录文件地址？

索引表存放在何处？

这就是一般的一种处理方法，我们看一个例子 那么这个例子呢，那么这是一个磁盘的布局图



索引结构示意图



文件名	索引块块号
...
File B	24
...

- 1
- 8
- 3
- 14
- 28

目录项示意



那我就可以去查索引表，我把索引表读进内存之后，我就在表里头找第五行就可以了

索引结构的优缺点

◎ 优点

保持了链接结构的优点，又解决了其缺点

- 既能顺序存取，又能随机存取
- 满足了文件动态增长、插入删除的要求
- 能充分利用磁盘空间

◎ 缺点

- 较多的寻道次数和寻道时间
- 索引表本身带来了系统开销

如：内存、磁盘空间，存取时间

所以这就是索引结构的优缺点 下面我们来看一下索引表的组织方式



索引表的组织方式

- 问题：索引表很大，需要多个物理块存放时怎么办？

- 链接方式

一个盘块存一个索引表，多个索引表链接起来

- 多级索引方式

将文件的索引表地址放在另一个索引表中

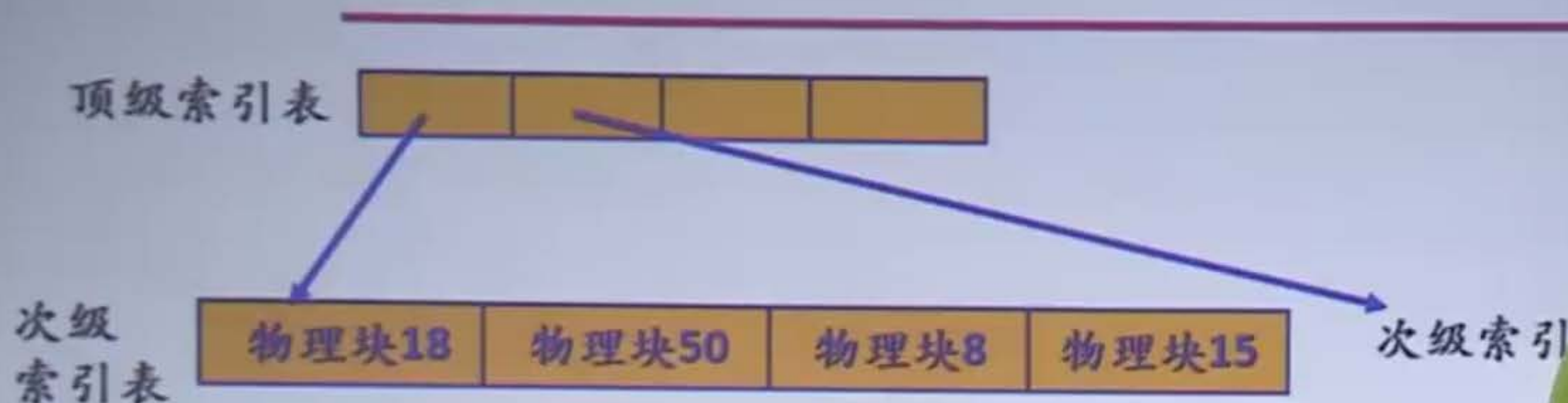
- 综合模式

直接索引方式 与 间接索引方式 结合

用一个直接的索引方式再加上一个间接的索引方式相结合，来构造我们的索引表



多级索引与综合模式



还有一部分呢是间接的寻址，当然也可以把这个二级再扩展成三级或者四级 下面我们介绍

UNIX的三级索引结构

UNIX文件系统采用的是多级索引结构(综合模式)

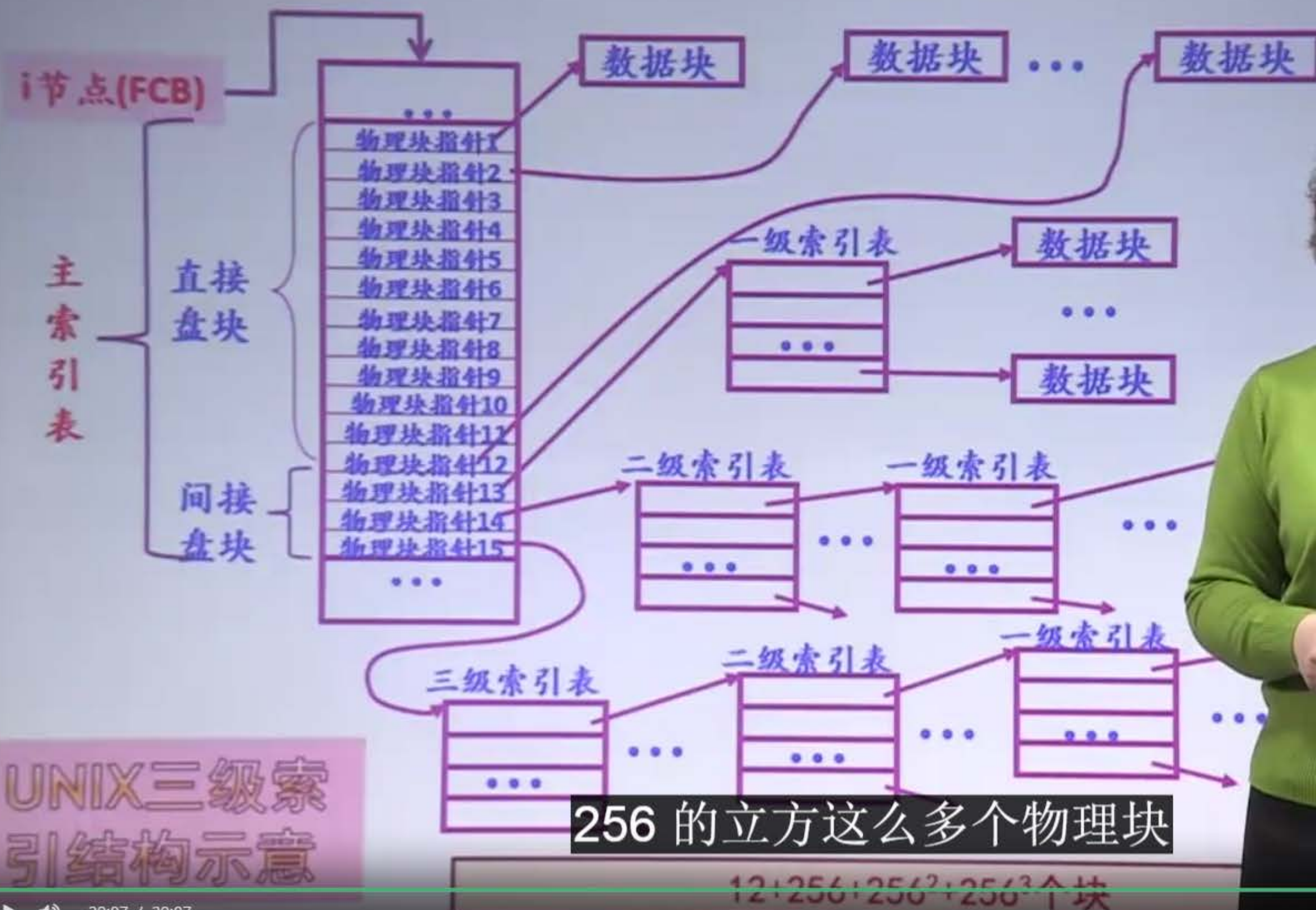
- 每个文件的主索引表有**15**个索引项(FCB中), 每项**2**个字节
- 前**12**项直接存放文件的物理块号 (直接寻址)
- 如果文件大于**12**块, 则利用第**13**项指向一个物理块, 在该块中存放的是一级索引表)

假设扇区大小为**512**字节, 物理块等于扇区块大小, 一级索引表可以存放**256**个物理块号

- 对于更大的文件还可利用第**14**和第**15**项作为二级和三级索引表

试问: 采用这种结构, 一个文件最大可达到 **?** 个物理块





UNIX三级索引结构示意

256 的立方这么多个物理块

下面我们介绍文件的物理结构 文件的物理结构呢，对于文件的逻辑结构 它是指文件在存储介质上的一个存放方式 谈到文件的物理结构，其实我们重要的要解决两个问题 第一个问题就是说，如果一个文件啊，被划分成 N 块，啊，我们这个块啊，就是 基本的单位，被划分成 N 块，那么这 N 块在磁盘上是怎么存放的？第二个要解决的问题就是说 我把一个文件存放在磁盘上的不同的物理块里头了 那么地址，或者是块号、簇号 究竟在 FCB 当中是怎么记录的？所以我们要解决这两个问题，针对文件的物理结构，我们要把这两个问题解决 那么文件的物理结构呢，典型的有三种 第一种呢，叫做顺序结构，或者叫做连续结构 它指的是文件的信息 存放在若干连续的物理块当中 那么我们说如果这个文件被划分成 N 块 那么，这 N 块是在磁盘上是连续存放的 我们这里有一个例子，我们来看一下 在这个图当中，有这样几个文件，文件 A, B, C, D, E, F, G 每个文件呢实际上都占据了一片连续的物理块 像文件 E 占据了 12 个连续的物理块 那么如果，我们可以看到，下面，啊 文件 D 啊，就是被删除了，那么它的 物理块呢，实际上就变成了空闲的物理块，那么这就是 5 个，啊，空闲的物理块，啊，就还回来了 那么如果，文件 F 啊，也被删除了，也得到了 6 个啊 空闲的物理块，啊，这就是相当于一些，啊，空闲物理块，如果太小了那么就 也可以称之为碎片，这就是连续结构 那么我们要回答的第二个问题就是说 在 FCB 当中如何记录文件的地址呢？那么如果我们拿，啊，文件 E 啊做一个代表 文件 E 呢，我们要给出在 FCB 当中只要给出文件 E 的第一块的，啊 块号，那么再给出来它的长度，比如说它有 12 个，啊 物理块，或者是给它一个，啊，字节的长度，再换算成 12 也可以 那么这就是通过这两个信息，我们就可以找到这个文件 E 的任何一块了，啊 这就是在 FCB 当中如何来记录文件的地址 那么顺序结构，连续结构有什么优缺点呢？它的优点呢是很简单，啊 它支持顺序存取，也支持随机存取，因为 如果我想把这个文件的第 6 块读出来 我只需要把它的起始块号，再加上 偏移，我就能够直接把第 6 块读上，读进内存了 当然了，如果我顺序读，第 1 块读第 2 块，这也是可以的 另外，它所需的磁道的寻道时间和 寻道的次数都是比较少的，因为它是连续存放在磁盘的一个区域里头，那么如果按照我们刚才的扇区磁道的话 这些扇区可能在同一个磁道上，就不需要再移动磁头了 那么还有一个优点是，它可以 同时读入多个块，检索一块也很容易，因为它是连续放的 缺点呢，也很明显 首先呢，文件不能够动态增长，因为我们刚才看到那个布局里头 如果一个文件，比如说文件 E 原来 12 块，它要变成 13 块、14 块了 可是它的后面已经有其他文件占据了，啊，物

理块，所以它就不能够动态增长 如果要增长怎么办？那么几种办法，第一种办法可以预留一些空间 但这个事情呢，第一，不能预测预留多少，第二呢，预留的空间也是浪费 或者是，采用另外一种方法，重新分配给它另外一块 啊，一片连续的物理块，然后把它，这个文件，整个搬到新的区域 当然了，搬到新的区域，就带来了其他的开销 时间上要花费很多。第二个缺点呢，就是 这种连续结构不利于文件的插入和删除 另外呢，它会产生，啊，外部碎片，也就是我们说的 在这个磁盘上，啊，这个文件如果被删除了，它所 占，占据的这些空闲块呢，可能成为了一个碎片 那么我们也可以通过紧缩技术来进行一个碎片整理

第二种文件的物理结构呢，是链接结构 所谓链接结构呢，指的是一个文件的信息呢，存放在 若干不连续的物理块当中，也就是说如果一个文件把它 划分成 N 块的话，这 N 块是保存在不连续的物理块当中 那么各块之间呢，是需要通过指针把它链接在一起的 也就是说，利用前一个物理块当中的，啊，一部分，啊，空间来 做指针，指向，啊，下一个物理块，这样的话 就会把一个文件的所有的物理块，把它链起来了 我们来看一个例子，文件 A 一共有 5 块，那么它的第 0 块，啊 是存放在物理块为 4 的，啊，这样一个物理块上 第二块，或者是块 1 存放在物理块为，块号为 7 的这个块上 然后，第 3 块也就是块号为 2 的存放在物理块 2 上 第 4 块物理块 10 上，第 5 块物理块 12 上 因此我们可以看通过这样一个链，就可以把这个文件 A 的所有内容找到 同样，这个最后一个块呢，它的就是用一个控制针来表示它已经是最后一个块了 那么文件 B 呢现在有 4 块。还是刚才那个问题 对于链接结构在 FCB 当中如何记录文件的物理地址 实际上我们可以看到，只要把 文件的第一块的块号给出来就可以了，就起始块号 保存在 FCB 中就可以了，其实不需要保存它的长度 或者是它的需要多少块，因为我们可以通过 指针的最后一块的这个空指针表示它是最后一块 知道这是文件的最后一块了。但是呢，通常我们知道 FCB 里头文件的大小，或者是长度，也是一个 必须存放的信息，这样会快一点地去找到相关信息 下面我们来看一下，链接结构的优缺点 第一个优点呢，是它提高了磁盘空间的利用率 如果一个文件被分成了 N 块 可以在磁盘上找 N 个 空闲的物理块分配给这个文件，而不需要找一个连续的 N 块 这样的话呢，系统里头，磁盘里头，就不存在了 外碎片的问题，因为任何一个物理块，都可以分配给某个文件 第二个优点呢，它就有利于文件的插入、删除这样的操作 可以动态的来，啊，对文件插入、删除 当然了，第三个优点 有利于文件的动态

扩充。链接结构的缺点是什么呢 存取速度慢, 不适合于随机存取 那么我们知道, 如果我想 读取这个文件的第 5 块 不可能直接把这第 5 块读进来 因为你首先要把第 1 块读进来, 从第 1 块里 得到第 2 块的地址, 然后再把第 2 块读出来 所以你要把第 5 块读出来, 要把前面 4 块都读完了 才能够把第 5 块读出来, 因此它的存取速度就会很慢 也不能够随机存取。第二个缺点呢 就是由于指针可能会出错, 就会对这个文件的可靠性 产生影响, 因为我们是 用 这个物理块来保存一部分, 啊, 拿出来一部分的字段, 来保存指针的 如果这一块出了问题, 指针链一断掉, 那么后序的这些物理块我们就找不到了 所以对可靠性有影响。由于在 在分配这个物理块的时候, 任何一块都可以分配给这个文件 因此这些块呢, 可能分散在磁盘的不同的位置 可能在磁盘的最里道, 也可能在磁盘的最外道 因此, 从最里道到最外道在读取这个文件的过程中, 就会带来 更多的寻道的次数, 同时呢增加了寻道的时间 当然最后一个问题 也比较严重, 因为我们的一个物理块 假如我们说, 就正好等于一个扇区, 啊, 通常呢 一个物理块呢 等于 2 的整数次倍扇区个数 那么我们假设, 一个物理块就等于一个扇区 那么这样的话呢, 512 字节, 我们还要拿出来一点点存放什么呢? 链指针 因此我们的数据就不够完美, 因为我们的 这个数据最好都是 2 的整数次倍的一种传输 那么拿出来一部分的字段来放链接指针, 它占用一部分空间 使得后序的处理带来了很多的这个麻烦 所以这也是链接结构的一个缺点 因此, 我们可以对链接结构进行某种改造 形成一种新型的链接结构, 这就是我们下面要介绍 的一种变形的链接结构, 文件分配表 FAT 在 Windows 操作系统里头, 早期的, 或者说我们现在用的这种优盘呢, 用的这种文件系统 那么通常都是用的是 FAT 文件系统 这呢也是一种链接结构 那么它的思想是什么呢? 它并不是说用一个 物理块的一部分的这个字段来存放链指针 它是把所有的物理块的这个 指针都集中存放在一张表里头 这张表就叫做 FAT 表, 文件分配表, 因此 对于磁盘上的任何一个物理块, 我这表里头就有一个表项 这个表项呢就可以记录相关的信息, 我们来看一下 这是文件分配表的一个示意, 那么这张表呢, 实际上就是根据 在磁盘里头的所有的物理块来建立的一张表, 也就是每一个 物理块就在表里头对应一行 我们可以看到, 在这里头呢 存放了两个文件, 一个是文件 A, 一个是文件 B 那么文件 A 它的起始在这里, 也就是在 放在了这一块, 应该是块号是 4, 啊, 就是文件 A 的起始块号是 4 那么在这个表项里头呢, 给出了 文件 A 的第二块, 啊, 下一块的块号是 7 所以我们通过 7 我们

就可以找到这个文件 A 的下一块 而这个文件的第三块呢 是放在了块号为 2 的位置 在这儿，然后是 10 在这儿，12 到这儿，好，那么文件 A 呢 结束 通过一个特定的标志，说这个这是文件的最后一块了 那么同样，文件 B 也有这样一个这个分配。那么每一个这里头，每一个表项实际上呢，它的作用起了三个作用，或者说它的值有三种 第一种是 0，就表示这个表项 是空的，那就是说它是一个空闲的物理块 所以我们可以查找 0，来找到空闲的物理块 或者这个值呢 保存的是一个块号 当然这个块号呢，指向的是说下一块它的块号 所以我们可以看到，文件 A 的起始块是 4，下一块块号 记录在 4 对应的这个表项里头，所以呢 它可以指出下一块的块号。第三个可能值呢 就是比如说 比如说 -1 它是一个特定的值，它就表示这是这个文件的最后一块了 所以我们说，文件分配表就是把 分散在这个磁盘上的这个文件的所有块的块号 都记录在这张表里头，当然记录的方式呢就是 刚才我们所介绍的 那么如果有了文件分配表这样一种链接结构 那么某个文件的起始块号，从哪里得到呢？也就是说我们知道文件 A 是从 块 4 啊开始的，那么块 4 这个 4 记录在哪里呢？实际上就记录在前面我们所说的 FCB 里头，起始块号记录在那里 我们查到了 FCB 以后，我们就得到了起始块号，我们就可以找到这个文件了 因为后面的就是通过这个链来找的 所以这是链接结构。第三种结构呢 是索引结构 索引结构呢 其实和链接结构一样 它的一个文件的信息，也是存放在若干不连续的物理块当中 但是呢，系统会为每个文件 建立了一个专门的数据结构，叫做索引表 索引表里头，就把这些物理块的块号放在了索引表里头 好，那么这个索引表，实际上就是 我们索引结构当中一个很重要的一个数据结构 那么索引表其实就是把磁盘块块号 都放在这个表里头，因此呢，它可以看成是一个磁盘块的一个地址的 数组，每个条目，比如第 i 个条目就指向了文件的第 i 块，所以通过对这个数组的索引，就能够找到这个文件 对应的那个物理块的块号，这就是索引结构 那么在索引结构当中，一样会 碰到说在 FCB 当中，怎么样去记录文件的地址呢？那么索引表究竟放在 什么地方呢？我们有这样两个需要大家解决的问题 那么我们可以这样来看，就是说如果索引表 就放在 FCB 里头，可不可以？那我们知道，一个文件可能很大，也可能很小 它的索引表是不等长的，有的很长，有的很短 所以呢，不宜把索引表放在，直接放在 FCB 里头 因为 FCB 是固定大小的，所以我们可以 在 FCB 里头 派生出一个字段来，专门存放索引表的位置 这就是一般的一种处理方法，我们看一个例子 那么这个例

子呢，那么这是一个磁盘的布局图 那我们可以看到，某个文件 B 那么在这里头呢，我们给出的是什么呢？给出的是一个索引块的块号 也就是索引表会放到某一个物理块里头 而这个物理块，我们就称之为索引块，啊，或者叫做索引表所占的块 那么这个索引块里头，就存放了这个文件的 所分配给它的若干个物理块的块号，比如说我们还是以文件 B 为例 文件 B 呢在磁盘上呢它是放在了 1 8 3 14 28 这五块里头，也就是说这是文件 B 的第一块 文件 B 的第二块呢在这儿，第三块呢在这里 第四块在这儿，第五块在这儿 然后我们把这个信息呢，集中放在了一个，一个块里头，这个块呢 是 24 块 24 块里头呢，当然现在没放满，就放了这么五块的块号，那么还可以继续放 啊，得看它这个一块，究竟能放多少个物理块块号 这就是索引表，把索引表的所在的 块号记录在了 FCB 里头，那么就可以通过查找到 FCB 就得到了索引表，那我们就得到了索引表之后，就知道这个文件的每一块放在什么地方了 那么在索引结构当中，其实我们可以做随机存取，大家可以看一下说，如果我要取第五块 那我就可以去查索引表，我把索引表读进内存之后，我就在表里头找第五行就可以了 下面我们来看一下索引结构的优缺点 索引结构呢既保持了链接机构的优点 又解决了它的缺点，它既能够顺序存取，也能够随机存取 同时能够满足文件的动态增长，插入 删除的要求。另外呢，它能够充分地利用磁盘空间 它的缺点呢当然是 由于它和链接结构一样，是随机分配的物理块 因此可能在查找一个文件的时候，寻道的次数和时间会比较多 但是呢另外一个缺点呢是索引表本身 会带来一定的开销，比如说索引表要耗费一定的内存、磁盘空间 另外呢，去存取索引表也需要花时间 所以这就是索引结构的优缺点 下面我们来看一下索引表的组织方式 我们提出这样一个问题，索引表很大 一个物理块不够，需要若干个物理块来放索引表 那么这些物理块在磁盘上是怎么样 存放的呢？通常呢有这样三种解决方案，第一种叫做链接方式 也就是说我们用一个磁盘块来放一个索引表 如果一个索引表占据多个磁盘块 这些磁盘块呢通过链的方式把它们链接起来 第二种方式呢我们可以采用多级索引方式 也就是把索引表的地址 放在另外一个索引表里头 这就是多级索引方式 那么通常我们还可以采用把两者结合起来 用一个直接的索引方式再加上一个间接的索引方式相结合，来构造我们的索引表 下面我们举两个例子。首先我们介绍一下多级索引 多级索引呢我们这里头介绍的是两级索引 顶级索引表每一个表项 都给出了一个次级索引表的地址 而次级索引表里头呢给的是这个文件所分配给它的物理块的地

址当然，我们可以看到说如果两级不够，那我们还可以把它扩展成三级那么扩展成三级的话呢，就是在次级索引表里头的原来这是文件所占用的物理块的地址现在就把它再给出一个下一级的索引表的地址，就这样的去扩展那我们再看一下综合模式综合模式呢我们来看一下顶级索引表的每一个表项它的作用是不太一样的。比如说在我们这里头分成两类前面两个表项给出的是直接的物理块地址，也就是这个文件 18 和 50 块的块号，就是分配给这个文件的直接的物理块后面两个表项给出的是次级索引表的地址，也就是这个是一个次级索引表它的地址写在了这个表项里头。而次级索引表呢给出的是这个文件的其他的物理块的地址这就是一种综合的模式，有一部分是直接的寻址还有一部分呢是间接的寻址，当然也可以把这个二级再扩展成三级或者四级下面我们介绍 UNIX 的三级索引结构 UNIX 文件系统采用的是多级索引结构它就是我们上面所介绍的综合模式那么每个文件都有一个主索引表存放在 FCB 当中这个主索引表呢是固定长度有 15 个索引项，每个索引项占两个字节那么 15 个索引项当中的前 12 项保存了文件的前 12 个物理块的块号，这也就是我们所说的直接寻址那么如果这个文件很大，大过 12 块，超过了 12 块那么就利用第 13 个索引项指向一个物理块，而这个物理块是保存了一级索引表的地址一级索引表呢由于我们的每一个这个扇区大小是 512 字节物理块大小假设我们就等于扇区大小，这样一级索引表里头可以存放 256 个物理块号那也就是说文件大于 12 块的时候大于 12 块的部分可以在一级索引表里头记录 256 个物理块块号那么如果文件还要大，大过了 12 块加上 256 块，那就要利用第 14 个索引项和第 15 个索引项作为二级索引表和三级索引表下面我们提一个问题，在 UNIX 中采用三级索引结构，一个文件最大可以达到多少个物理块呢？我们通过这张图来回答这个问题在这张图当中，这里我们给出了一个文件的 FCB 在 UNIX 里头，FCB 呢这里头叫 i 节点我们就把它先看成是一个 FCB 就可以了在这个 FCB 当中呢有一个主索引表，一共有 15 个索引项，那么前 12 个索引项呢是直接盘块也就是说这 12 个索引项里头记录了这个文件的前 12 块的块号。如果这个文件大于 12 块那么它的块号呢就没有地方记录了，所以我们需要启动第 13 个索引项第 13 个索引项记录了一个一级索引表的块号通过它可以找到一个一级索引表。而一级索引表当中呢按照我们前面所说的，记录了 256 个磁盘块块号，因此呢可以指向 256 个物理块。如果这个文件大于 12 加上 256 块，那么我们就要启用第 14 个索引项，

第 14 个索引项里头呢给出的是二级索引表的地址，而二级索引表里头一共有 256 行这 256 项里头每一项都指向一个一级索引表 那这样的话呢，如果启用二级索引表，可以表达的块数呢 就达到了 256 的平方 那么如果再启用了第 15 个索引项，那么我们就可以看到 15 个索引项指向的是个三级索引表 而三级索引表呢每一个表项又指向了 二级索引表，这样就有 256 个二级索引表，而 每个索引表又指出了 256 个一级索引表 而每个一级索引表又给出了 256 个物理块的块号 因此如果用了三级索引表 那么物理块的块数就达到了 256 的立方。所以把所有的 汇总起来，在 UNIX 文件 系统当中，如果采用的是三级索引结构 用 15 个索引项记录主索引表 前 12 项作为直接的索引 第 13 项指向一级索引表 第 14 项指向二级索引表，第 15 项指向三级索引表 这样在 UNIX 文件系统当中 一个文件最大可以达到 12 加上 256 再加上 256 的平方，再加上 256 的立方这么多个物理块