磁盘上的布局、内存中的数据结构

文件系统的实现1

下面我们介绍一下文件系统的实现

0:01 / 13:54

概述

- 实现文件系统需要考虑
  磁盘上 与 内存中 的 内容布局
- 磁盘上
  - ✓ 如何启动操作系统？
  - ✓ 磁盘是怎样管理的？怎样获取磁盘的有关信息？
  - ✓ 目录文件在磁盘上怎么存放？普通文件在磁盘上怎么存放？
- 内存中
  当进程使用文件时，操作系统是如何支持的？
  文件系统的内存数据结构

也就是在内存当中提供什么样的数据结构 下面我们介绍一下

相关术语

- 磁盘分区(partition)：把一个物理磁盘的存储空间划分为几个相互独立的部分，称为分区
- 文件卷(volume)：磁盘上的逻辑分区，由一个或多个物理块(簇)组成
  - 一个文件卷可以是整个磁盘 或 部分磁盘 或 跨盘（RAID）
  - 同一个文件卷中使用同一份管理数据进行文件分配和磁盘空闲空间管理，不同的文件卷中的管理数据是相互独立的
  - 一个文件卷上：包括文件系统信息、一组文件（用户文件目录文件）、未分配空间
  - 块（Block）或簇（Cluster）：一个或多个（2的幂）连续的扇区，可寻址数据块
- 格式化(format)：在一个文件卷上建立文件系统，即建立并初始化用于文件分配和磁盘空闲空间管理的管理数据

元数据

磁盘上的内容

- 引导区

  包括了从该卷引导操作系统所需要的信息

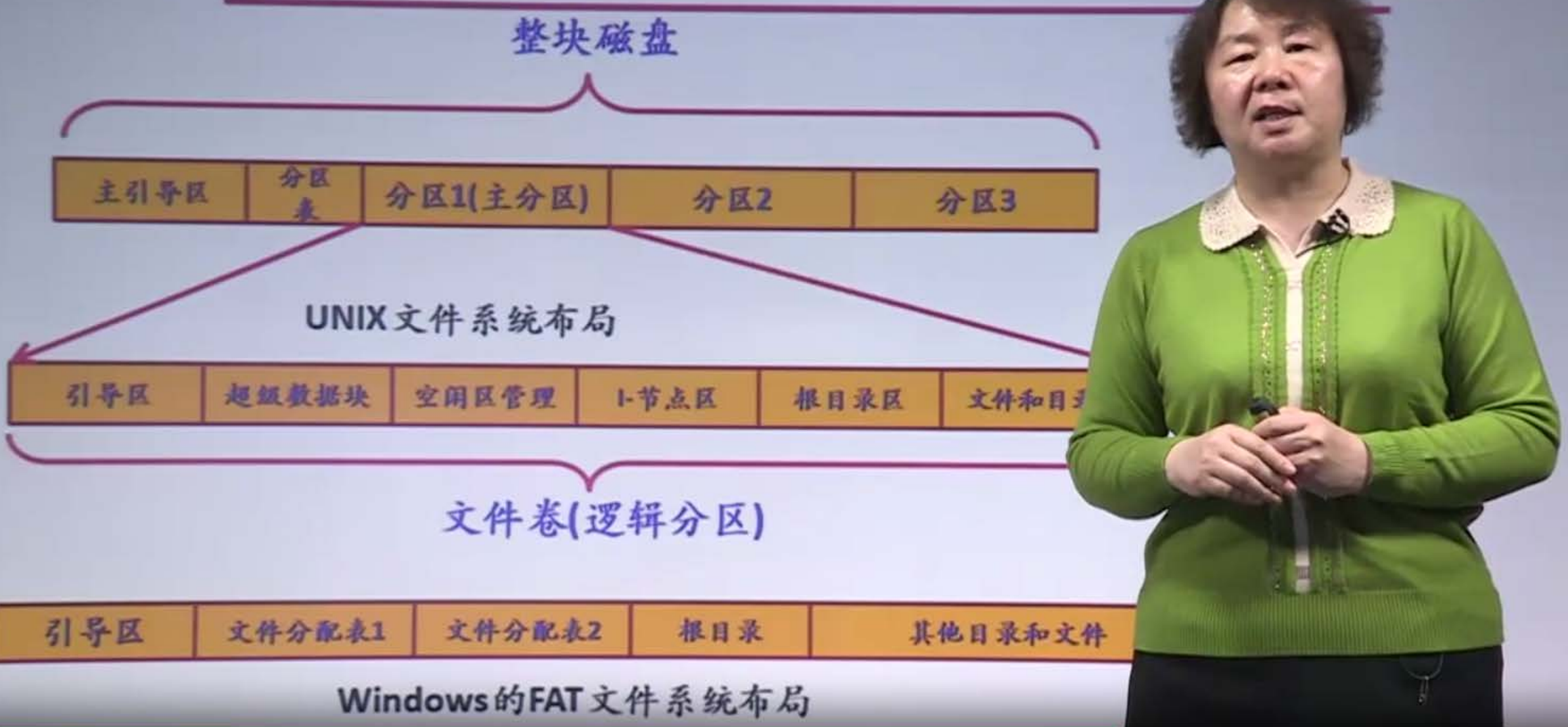
  每个卷（分区）一个，通常为第一个扇区

- 卷（分区）信息

  包括该卷（分区）的块（簇）数、块（簇）大小空闲块（簇）数量和指针、空闲FCB数量和指针.....

- 目录文件(根目录文件及其他目录文件)
- 用户文件

磁盘上文件系统的布局


整块磁盘
主引导区
分区表
分区1(主分区)
分区2
分区3
UNIX文件系统布局
引导区
超级数据块
空闲区管理
i-节点区
根目录区
文件卷(逻辑分区)
引导区
文件分配表1
文件分配表2
根目录
其他目录和文件
Windows的FAT文件系统布局

内存中所需的数据结构
——以UNIX为例

- 系统打开文件表
  - 整个系统一张
  - 放在内存：用于保存已打开文件的FCB

FCB(i节点)信息	引用计数	修改标记

- 用户打开文件表
  - 每个进程一个
  - 进程的PCB中记录了用户打开文件表的位置

文件描述符	打开方式	读写指针	系统打开文件表索引

找到对应这个文件的什么呢，FCB 的相关信息

0:00 下面我们介绍一下文件系统的实现 在实现文件系统时，我们需要考虑两方面内容 一个是在磁盘上如何实现文件系统的布局 包括了如何启动操作系统 磁盘是怎样管理的，如何获取磁盘上的信息 磁盘上有很多的文件，包括了目录 文件和普通文件，那么目录文件在磁盘上是怎么存放的？ 普通文件在磁盘上又是怎么存放的呢？ 第二个要考虑的呢，是在内存当中 在进程使用文件的时候操作系统如何提供支持 也就是在内存当中提供什么样的数据结构 下面我们介绍一下 在实现文件系统当中涉及到的一些术语 首先我们介绍一下磁盘分区 我们买来一块盘之后，通常我们会把这块物理硬盘 划分成若干个相互独立的部分，那么我们就称之为一个磁盘分区 便于分类保存信息。 第二个概念呢是文件卷 所谓文件卷呢实际上是磁盘的一个逻辑分区 它是由一个或多个物理块或者簇来组成的 那么一个文件卷呢 可以是整个一个盘一个文件卷 也可以是部分磁盘，也就是刚才我们说的分区，一个分区一个文件卷 还可以是跨盘，也就是几块硬盘构成一块大的逻辑盘 那么文件卷建立在这个逻辑盘之上 为什么要划分文件卷呢？因为 同一个文件卷使用相同的一份管理数据 那么对文件的分配、 空闲空间的管理都用这套管理数据来完成 那么不同的文件卷，这套管理数据呢是互相独立的 那么在一个文件卷上呢，我们主要 包括了以下一些内容。第一个呢是文件系统的相关信息 第二部分呢是你要保存的各种文件，普通文件 目录文件。 第三部分呢就是那些没有 使用的空间。 所以整个一个文件卷 这上面无非就是这三种内容 那么在 一个文件卷上它的存取单位 分配单位、 传输单位，我们都用的是什么呢，用的是块或者是簇 那么它通常和扇区的关系呢是 2 的整数次倍这样的一个关系 那么用这个块和簇可以直接进行数据块的寻址 当我们格式化一块盘的时候 或者我们格式化某一个分区的时候，实际上就是在这个 文件卷上建立一个文件系统 那么包括了建立并初始化用于管理这个文件卷的 各种各样的管理信息、 管理数据，这就是我们 前面提到过的叫元数据，所以格式化实际上就是把这些元数据建立起来 那么磁盘上 保存的文件它是怎样一个布局呢？ 我们来以文件卷作为一个单位 这个文件卷我们就可以看成是若干个物理块或者簇来组成的 每一个格子代表的是一个簇或者是一个物理块，相同大小的 那么在这样的一个空间当中可以存放的信息呢 第一个是引导区。 引导区主要是存放了 从这个卷引导操作系统所需要的信息。 通常呢每个分区其实都有一个引导区 大小呢可以是一个扇区，这是通常的尺寸，也可以是一块 那么 第二部分信息呢，就是这个整个这个卷的一些管理信息，我们也称之为卷信息 这里头包括了什么呢？可能包括了这一个卷当中的 有多少个簇或者多少个块来组成 每一簇或块到底有

多大。那么如果我要查找一些空闲的块，那么我的空闲块有多少数量，怎么能找到相应的空闲块 如果我想存放 FCB 的信息，我到磁盘上怎么能找到空闲的 FCB 那这些信息都属于系统的这个元数据管理信息之后呢其实就是我们的目录文件 目录文件其实也是操作系统它的主要的信息，因为操作系统为了管理文件 建立这个目录文件，所以这些内容呢也是操作系统必须的要管理的信息 那么目录文件呢特殊的是根目录文件，其他的是一些其他目录文件，那都是存放在这个文- 件卷上的 剩下的空间就存放用户的文件了这就是磁盘上内容呢分成的几个不同的内容 那么这些内容在磁盘上是怎么 布局的呢？我们来看两个操作系统的例子 首先我们看这是一块整个盘 这整个盘呢我们可以假设说分成了若干个分区 那么有一个主引导区呢通常操作系统是从这引导的 但是每个分区其实也有引导区，也可以引导操作系统 那么我们拿其中一个分区来看一下，UNIX 文件系统的一个布局。 所以这是一个分区，也就是一个文件卷 那么在它上面的 UNIX 文件系统的布局是怎么样安排的 在这样一个文件卷上，首先我们要有一个引导区，引导区 那么然后我们是有一些卷的信息，而卷的信息在这里头呢分成了 超级数据块还有空闲区的管理，这些都是跟卷相关的信息 那我们以这个空闲区管理为例 我们前面讲过了空闲区管理的数据结构无非 是位图、 空闲块表或者是空闲块链表 那么如果你是用位图的方法来管理空闲区，那么位图就保存在这个区域 如果你是用的是成组链接法，那么我们只需要一个专用块 而这个专用块呢就在这个空闲区这个位置，就是专用块的位置 在这之后有一个特殊的区域 叫 i 节点区，这是 UNIX 所特有的 那么这个区域呢其实就是放的是若干个 FCB 然后跟着就是根目录，因为前面的空间 分配出去之后固定大小了以后，根目录从什么地方开始就清楚了 然后就是其他的目录和文件，因为根目录特殊一点，我们单独放这儿 提出来。 这是 UNIX 的一个布局。 下面我们来看一下 Windows 文件系统的一个布局。 因为 Windows 有不同的文件系统，我们用 FAT 文件系统作为一个例子来看一下。 那么在 这样一个文件卷上，FAT 文件系统是怎么样布局的呢？也有一个引导区 然后呢有两个叫做文件分配表 一模一样的。 所以文件分配表2 是文件分配表1 的一个什么呢 镜像。 那么文件分配表1 呢 它就是管理了后面这些空闲块的 一些信息。 另外呢它还承接了 就是文件的这个我们叫链接结构的这样一个作用 因为我们前面在讲文件的物理结构的时候，我们说文件的物理结构当中的链接结构 其中把链指针全收集起来放在一张表里头，这就是文件分配表 所以通过这样一个联系

啊我们知道，在 FAT 文件系统当中，对于文件的存放的物理结构采用的是一种链接结构 它的链指针记录在了文件分配表当中 因此在 FAT 文件系统当中的卷的信息 包含在了引导区里头和文件分配表里头 一旦确定好了前面的这个位置，那么 根目录就定好了它的起点，然后就是其他的文件和目录 所以我们可以看一下这两个典型的操作系统当中的文件系统的布局 那么我们刚才讲的都是磁盘上的布局 那么当一个文件系统在运行过程中，某一个进程要使用文件系统的时候 那么就需要操作系统在内存里头提供相应的数据结构以支持这个进程对文件的使用 所以我们以 UNIX 为例来看一下内存当中所需要的数据结构 第一个数据结构呢我们叫做系统打开文件表 整个操作系统有这么一张表，它记录了 所有打开的文件的 FCB 的信息 那么这张表肯定是在内存里头。 我们来看一下这张表的主要内容 首先最主要的是保存的 FCB 的信息 因为在 UNIX 里头 FCB 的信息呢我们叫做 i 节点，所以我们也可以叫做 i 节点的信息 那么还有这样两个标志 其中第一个呢叫引用计数，这个引用计数呢是说如果我们 打开一个文件，在这张表里就把这个文件对应的 FCB 的内容放在这个 表项里头。 那么如果另外一个进程打开同一个文件，其实是不需要 在这个表里头保存两行的，那么只需要 将引用计数加 1 就可以了。 当把一个文件的 FCB 读进内存，当做了一些修改的时候，那么这个内容还要保存到 磁盘上。 因此呢我们需要设定一个修改的标记 根据这个标记呢就决定这些内容是不是有修改过 如果有修改过呢，那么当对文件进行关闭操作的时候 那么这个内容还要写回到磁盘的相应位置 第二个数据结构呢叫做用户打开文件表 那么这个表呢是每个进程有一张 它是在进程的 PCB 当中 记录了这张表的位置，也就是通过 PCB 可以找到这个进程的用户打开文件表 顾名思义，用户打开文件表里就保存了 这个用户打开的所有的文件的相关信息 而这些文件的 FCB 已经保存在了系统 打开文件表里头，所以在这张表里就不需要保存 FCB 了 那么保存什么信息呢？我们来看一下。 第一个信息呢是文件 描述符。 因为在 UNIX 里头 用户打开文件操作成功之后呢，系统会返回给它一个文件描述符 那么典型的文件描述符呢，比如说我们有标准输入文件，标准输出 文件和标准的错误输出文件，它的文件描述符呢是 0, 1, 2 三个整数 那么当一个进程运行过程中又打开文件的时候呢，那么这个 文件描述符呢就从 3 号以后 来接着来计数。 所以打开一个文件，系统返回一个文件描述符 那么这一行就记录了文件描述符的内容。 第二个 信息呢是打开方式，因为打开文件的时候可以以读的方式打开 也可能以写的

方式打开，所以要在这里记录这次打开的打开方式 然后是读写指针，因为打开一个文件要对文件进行 操作，从什么位置进行相应的操作呢？比如说读从哪个位置读数据呢？ 那么通过的是这个读写指针，那么我们写也是一样 从哪个地方接着写，那么要通过这个指针来定位 然后其他的信息就保存在了系统打开文件表 里头，于是在用户打开文件表里头会有一个索引 让我们能够通过这个索引去查找系统打开文件表 找到对应这个文件的什么呢，FCB 的相关信息