



第7章 文件系统的设计和实现



实验目的

- 深入了解文件管理涉及的概念和功能
- 理解文件系统如何组织和管理信息
- 通过编程模拟实现一个文件系统



主要内容

- 背景知识
 - 文件系统基本概念
 - 文件管理的数据结构
 - Ext2文件系统
- 实验内容
 - 模拟实现一个Linux文件系统





文件系统基本概念

- 文件和文件系统
- 文件分类和属性
- 文件控制块和文件目录



UNIX类文件系统和非UNIX类文件系统

- UNIX类文件使用四种和文件系统相关的抽象概念：文件、目录项、索引节点和安装点。
- 文件(file)—文件是由文件名标识的有序字节串，典型的配套文件操作有读、写、创建和删除等。
- 目录项(dentry)—是文件路径名中的一部分。
- 索引节点(inode)—是存放文件控制信息的数据结构，又分磁盘块中的inode和主存中活动的inode。
- 安装点(mount point)—文件系统被安装在一个特定的安装点上，所有的已安装文件系统都作为根文件系统树中的叶子出现在系统中。



主要内容

- 背景知识
 - 文件系统基本概念
 - 文件管理的数据结构
 - Ext2文件系统
- 实验内容
 - 模拟实现一个Linux文件系统



磁盘上文件空间的组织(1)

- 文件是对设备的一种抽象，而且是对磁盘设备进行多层次抽象的结果：
- 第一层抽象，从磁盘到分区。一个物理磁盘可划分成分区，每个分区可从逻辑上看作是一个独立的磁盘，可安装和驻留一个文件系统。
- 第二层抽象，从分区到扇区。磁盘由柱面号、磁道号和扇区号来定位，扇区是磁盘上的基本存储单元，例如每个扇区存储1K字节，可从外向里一个柱面接一个柱面，一个磁道接一个磁道给每个扇区编号，一个将磁盘扇区编号的系统使得磁盘变成为一系列扇区的集合。



磁盘上文件空间的组织(2)

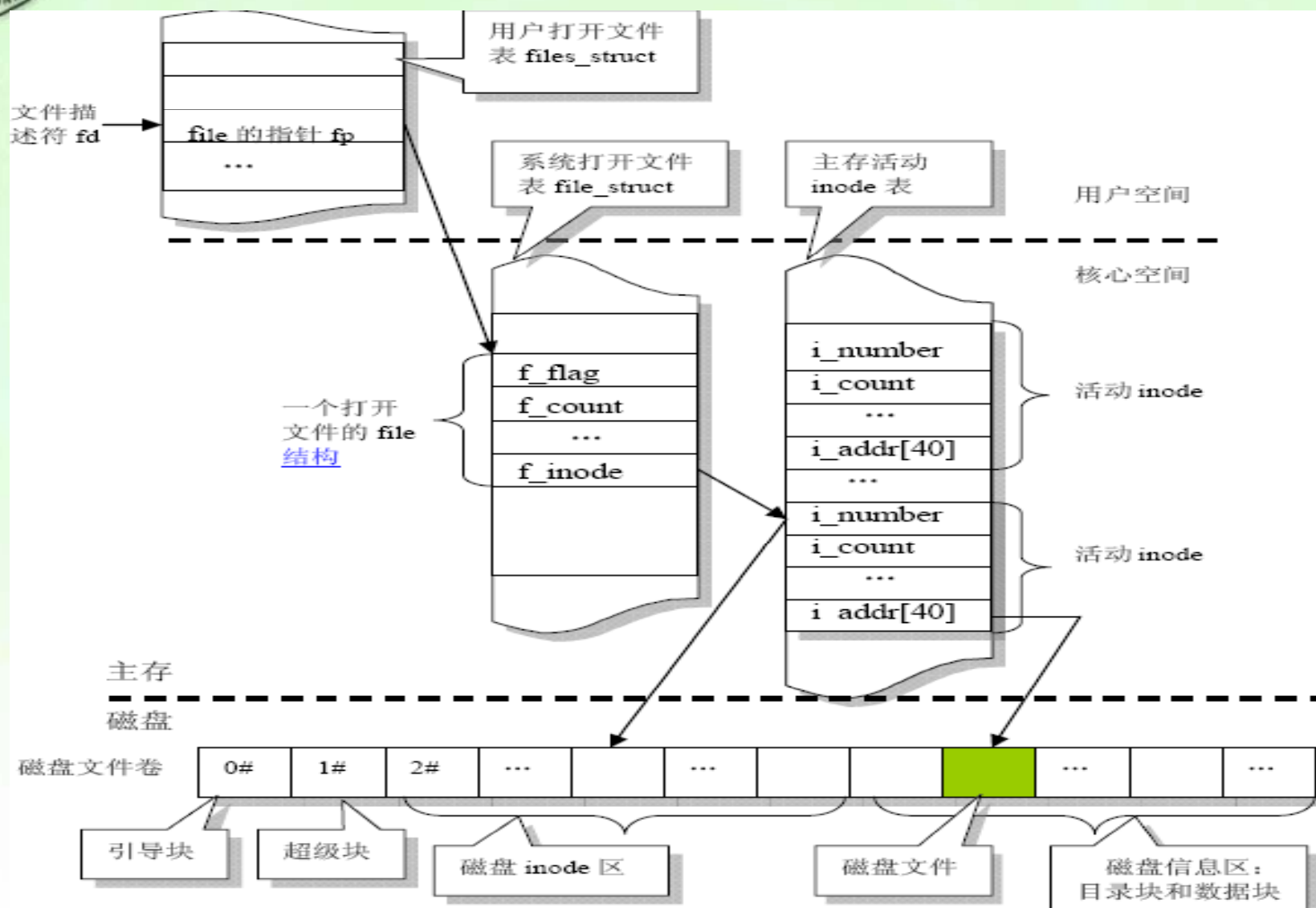
- 第三层抽象，从扇区到簇。不同磁盘的扇区大小可能不同，通过系统软件屏蔽这一事实并向高层软件提供统一的数据块尺寸，将若干扇区合并为一个逻辑块，称簇，再按簇进行编号，这样高层软件就只和大小都相同的簇交互，而不管物理扇区的尺寸。
- 第四层抽象，从簇到文件系统分区。内核再将簇序列分成超级块、**inode**区和数据块区等，再加上各种组织、控制和管理信息的软件便形成文件和文件系统。

扇区序列分成以下3个部分：

- 超级块：占用1#号块，存放文件系统结构和管理信息。
- 索引节点区：2# ~ k+1#块，存放**inode**表，**inode**记录文件属性，每个**inode**都有相同大小和唯一编号；文件系统中的每个文件在该表中都有一个**inode**。
- 数据区：k+2# ~ n#为数据块，文件的内容保存在这个区域的块中。



文件管理内部数据结构





主存中文件管理的数据结构

- 系统打开文件表：为解决多用户进程共享文件、父子进程共享文件而设置的系统数据结构，当打开一个文件时，通过系统打开文件表的表项把用户打开文件表的表项与文件活动inode联接起来，以实现数据的访问和信息的共享。
- 用户打开文件表：进程的PCB结构建立一张用户打开文件表或称文件描述符表，表项的序号为文件描述符，该登记项内登记系统打开文件表的一个入口指针，通过此系统打开文件表的表项连接到打开文件的活动inode。
- 主存索引节点表：为解决频繁访问磁盘索引节点inode表的效率问题，系统开辟的主存区，正在使用的文件的inode被调入主存活动索引节点inode中，以加快文件访问速度。



Ext2文件系统(1)

- Ext2(second extended file system)文件系统支持标准UNIX文件类型，包括普通文件、目录文件、特别文件和符号链接文件；Ext2文件系统可管理特大磁盘分区，文件系统最大可达4TB；此外，还支持长文件名(最长1012个字符)、可选的逻辑数据块大小、提供数据更新时同步写入磁盘和快速符号链接功能。
- Ext2文件系统的信息都保存在数据块中，数据块的长度相同，其物理结构如图7-5所示。Ext2所占用的磁盘除引导块外，逻辑分区划分为块组，每个块组依次包括超级块、块组描述符表、块位示图、inode位示图、inode表以及数据块区，重复保存有关文件系统的关键信息及存储的文件和目录信息。文件系统保存逻辑块号，由块设备驱动程序将逻辑块号转换成块设备的物理存储位置。引导块是磁盘上第1个数据块，只有根文件系统才有引导程序放在这里，Linux以Ext2作为它的根文件系统，新的发行版本已默认使用Ext3文件系统。



Ext2文件系统(2)

- 采用块组划分的目的：一是提高文件系统可靠性，每个块组中都有管理信息的副本，当文件系统崩溃时可很容易地恢复；二是提高文件系统性能，由于块组内数据块靠近其inode，文件inode靠近其目录inode，从而，将磁头定位时间减到最少，加快磁盘访问速度。

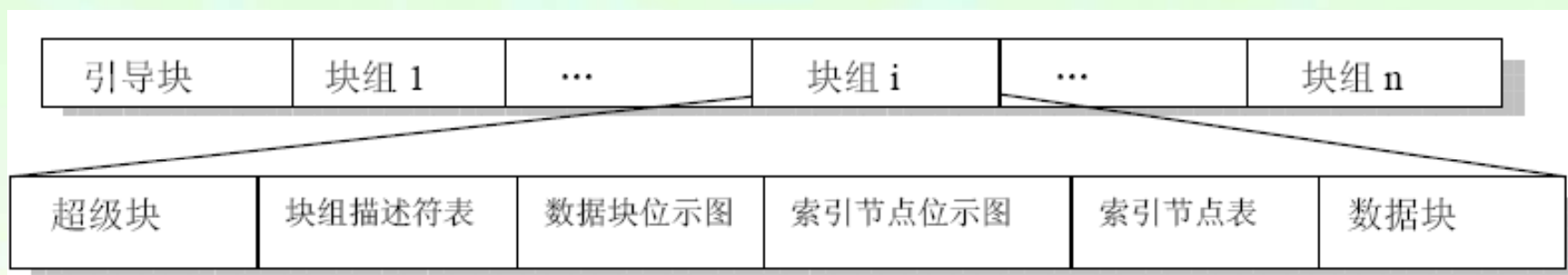


图 7-5 Ext2 文件系统结构



Ext2文件系统(3)

超级块

- Ext2超级块用**ext2_super_block**结构表示，用来描述目录和文件在磁盘上的静态分布，包括尺寸和结构，每个块组都有一个超级块，只有块1的超级块才被读入主存工作，直至卸载，其他块组的超级块仅作为恢复备份。超级块主要包括：块组编号、块数量、块长度(1KB至4KB)、空闲块数量、**inode**数量、空闲**inode**数量、第1个**inode**号、第1个数据块位置、每个块组中的块数、每个块组的**inode**数，以及安装时间、最后1次写时间、安装信息、文件系统状态信息等内容。



Ext2文件系统(4)

块组描述符

- 每个块组都有一个块组描述符`ext2_group_desc`，记录该块组的以下信息：
- 数据块位示图。表示数据块位示图占用的块号，此位示图反映块组中数据块的分配情况，在分配或释放数据块时需使用数据块位示图。
- `inode`位示图。表示`inode`位示图占用的块号，此位示图反映块组中`inode`的分配情况，在创建或删除文件时需使用`inode`位示图。
- `inode`表。块组中`inode`占用的数据块数，系统中的每个文件对应一个`inode`，每个`inode`都由一个数据结构来描述。
- 空闲块数、空闲`inode`数和已用数目。
- 一个文件系统中的所有块组描述符结构组成一个块组描述结构表，每个块组在其超级块之后都包含一个块组描述结构表的副本，实际上，Ext2文件系统仅使用块组1中的块组描述结构表。



Ext2文件系统(5)

文件目录

- 目录的数据块包含属于该目录的文件信息，用 `ext2_dir_entry_2` 结构描述，该结构内容有：`inode`号、本项长度、文件名长度、文件类型、文件名。目录的各项在数据块中依次存放，依靠本项长度计算每项的位置。



主要内容

- 背景知识
 - 文件系统基本概念
 - 文件管理的数据结构
 - Ext2文件系统
- 实验内容
 - 模拟实现一个Linux文件系统



模拟实现一个Linux文件系统

实验说明

- 在磁盘空间模拟实现一个Linux文件系统。
- 并提供基本的文件操作命令（如mk, cp, mkdir, rmdir, cd, ls, cat, chmod, chown, chgrp, chnam等）。
- 文件系统的实现要采用混合索引式文件结构，包括使用空闲结点号栈管理空闲结点和成组链接管理空闲盘块。



解决方案

- 物理磁盘块设计
- 空闲磁盘块管理
- 空闲inode结点
- 超级块、inode及目录结构设计
- 功能模块设计