



第8章 Linux文件系统

8.1 Linux文件系统的特点

8.2 ext2文件系统

8.3 Linux虚拟文件系统VFS

8.4 Linux的设备管理



8.1 Linux文件系统的 特点

1. 字节流文件

Linux不关心文件的结构

ASCII文件与二进制文件无本质区别

2. 目录当作文件



8.1 Linux文件系统的 特点

3. 文件描述符 (file descriptor)

一个整数，用于标识文件

open/creat时由系统分配，close时释放

3个固定的标准文件描述符：

0：标准输入

1：标准输出

2：标准错误输出

第1个打开的文件描述符是3



8.1 Linux文件系统的 特点

4. 文件的种类

(1) 正规文件 (regular file) : 普通文件

(2) 目录文件

(3) 特殊文件

设备文件 : 字符特殊文件 , 块特殊文件

(4) 套接字 (socket)

只是抽象成文件 , 不是真正的文件

(5) 管道

(6) 符号链接和硬链接 : 用ln命令可创建



8.1 Linux文件系统的 特点

5. 支持多种文件系统

- ✓ 支持ext , FAT , ext2 , ext3 , MINIX , MS DOS等

Linux的第1个文件系统是MINIX

1992 , 第1个专门为Linux设计的文件系统ext

ext: Extended File System

ext → ext2 → ext3

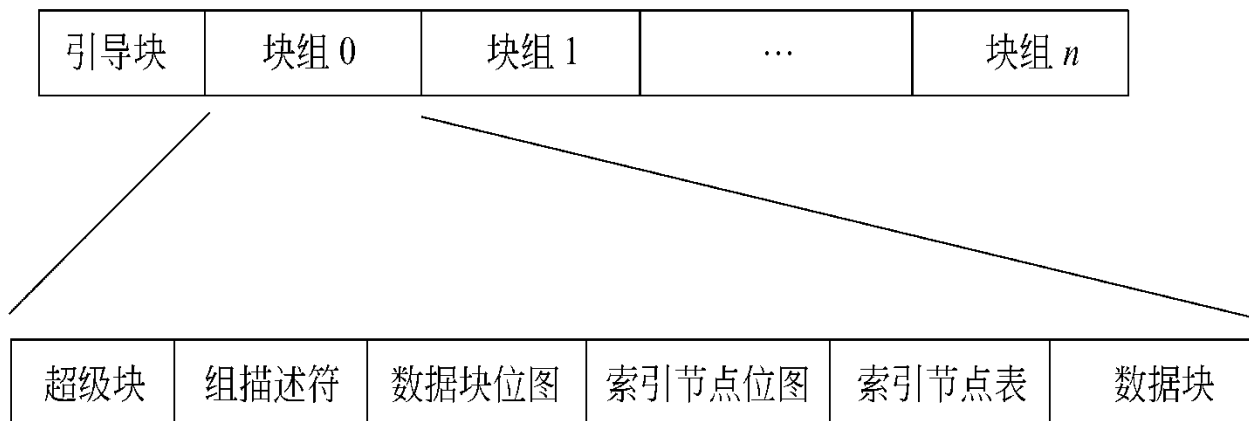
ext2是Linux的主要文件系统

- ✓ 引入虚拟文件系统VFS (Virtual File System)

VFS为用户程序提供了一个统一、抽象、虚拟的文件系统接口 , 该接口主要由一组标准的、抽象的有关文件操作的系统调用构成

8.2 ext2 文件系统

一、ext2文件系统的物理结构



- ✓ **引导块 (boot block)** : 装有启动OS的引导代码。当有多个文件系统时 , 只有1个有引导代码
- ✓ **块组 (block group)** : 由1个或多个连续的柱面组成



8.2 ext2 文件系统

✓ **超级块 (super block)** : 每个块组一个副本

包含文件系统本身的基本信息 :

数据块的大小

每组数据块的个数

首个I节点的号, 即根目录的入口

通常只有块组0的超级块才被读入内存, 其他块组的超级块用于备份。在系统运行期间, 超级块被复制到内存缓冲区中, 形成了一个 `ext2_super_block` 结构。



8.2 ext2 文件系统

✓ 块组描述符

每个块组都有一个相应的组描述符来描述它，所有的组描述符形成一个组描述符表，并在使用时被调入块高速缓存。

一个文件系统的所有数据块组描述符组成一个表，每一个块组在超级块后都包含一个数据块组描述符表的副本，以防遭到破坏。

✓ 位图

ext2中每个块组有两个位图块，一个用于表示数据块的使用情况，叫数据块位图；另一个用于表示索引节点的使用情况，叫索引节点位图。

位图中的每一位表示该组中一个数据块或一个索引块的使用情况，0表示空闲，1表示已分配。



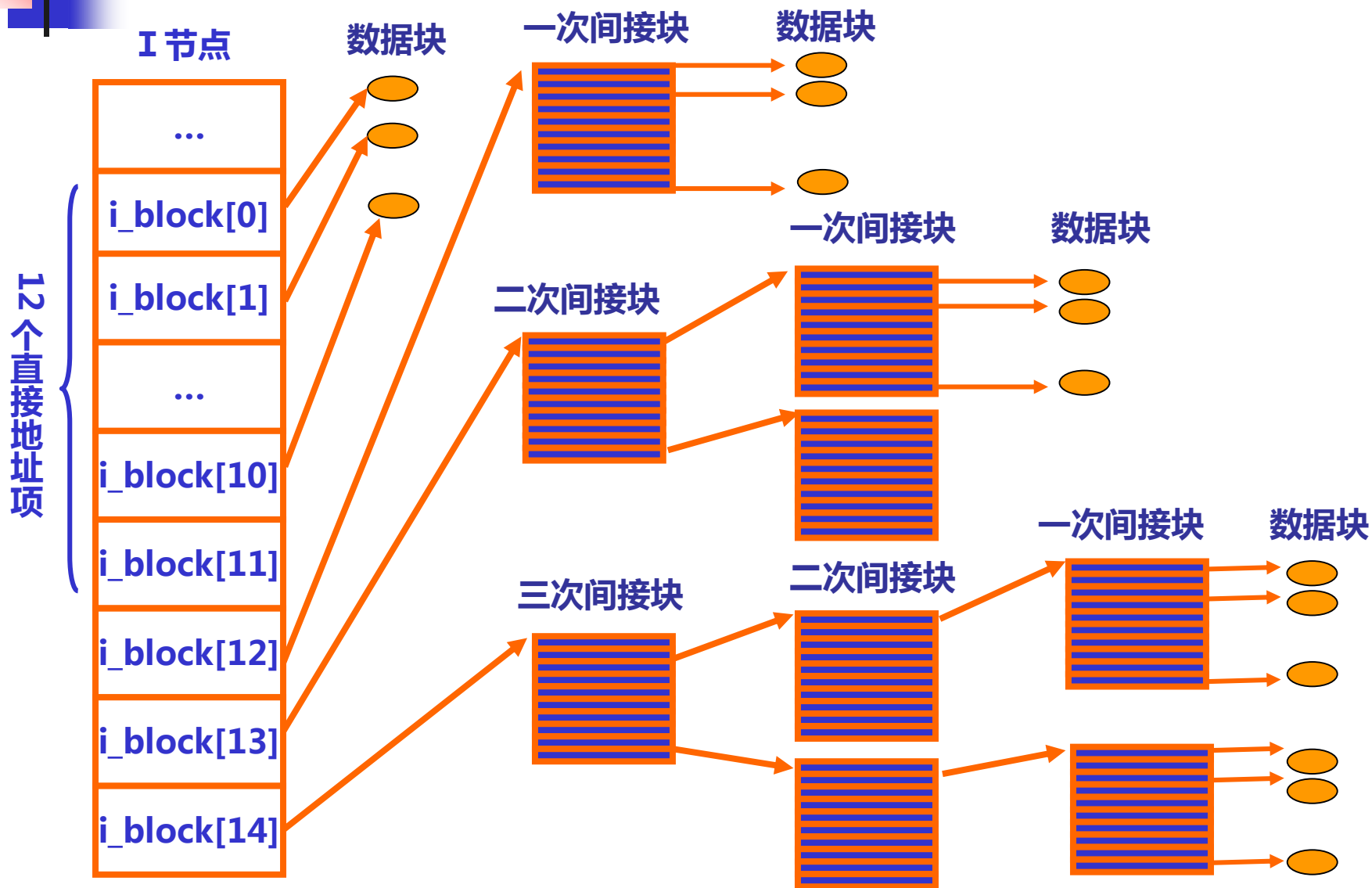
8.2 ext2 文件系统

二、几个重要的数据结构

1. 磁盘索引节点ext2_inode

- ✓ 文件类型：普通文件、目录文件、字符特殊文件、块特殊文件
- ✓ 访问权限
- ✓ 文件长度（字节数）
- ✓ 用户id、组id
- ✓ 与该I节点链接的文件数
- ✓ 文件的建立时间、最近访问/修改时间
- ✓ 文件的物理地址（数据块指针）i_block[0]~i_block[14]

磁盘I节点ext2_inode





磁盘I节点ext2_inode

说明：

设每个盘块1KB，块号用4B表示

用i_block[0]~i_block[11]：文件最大长度 = 12KB

用i_block[12]：1K/4 = 256，文件最大长度 = 256KB

用i_block[13]：文件最大长度 = $256 * 256 = 64\text{MB}$

用i_block[14]：文件最大长度 = $256 * 256 * 256 = 16\text{GB}$



磁盘I节点ext2_inode

如何根据文件的逻辑地址偏移量得到物理地址？

字节偏移量/每块字节数

商：逻辑块号

余数：块内偏移量

if 逻辑块号 $i < 12$: 物理块号为 $i_block[i]$

$12 \leq i < 268$: 需要查一次间接块

等等。



8.2 ext2 文件系统

2. 目录项

- ✓ 文件名
- ✓ 文件名长度
- ✓ I节点号

3. 内存索引节点inode

文件被打开后，在内存中建立相应的I节点，利于访问。

除了磁盘I节点的信息外，还包括：

引用计数：记录访问该I节点的进程数



8.2 ext2 文件系统

4. 用户打开文件表和系统打开文件表

(1) 用户打开文件表

又称为进程打开文件表，或**文件描述符表**

由task_struct的字段files所指向

- ✓ 记录进程当前打开的文件
- ✓ 其中的fd字段是一个指向file对象（结构）的指针数组，数组的索引就是文件描述符。file对象构成的链表就是系统打开文件表



8.2 ext2 文件系统

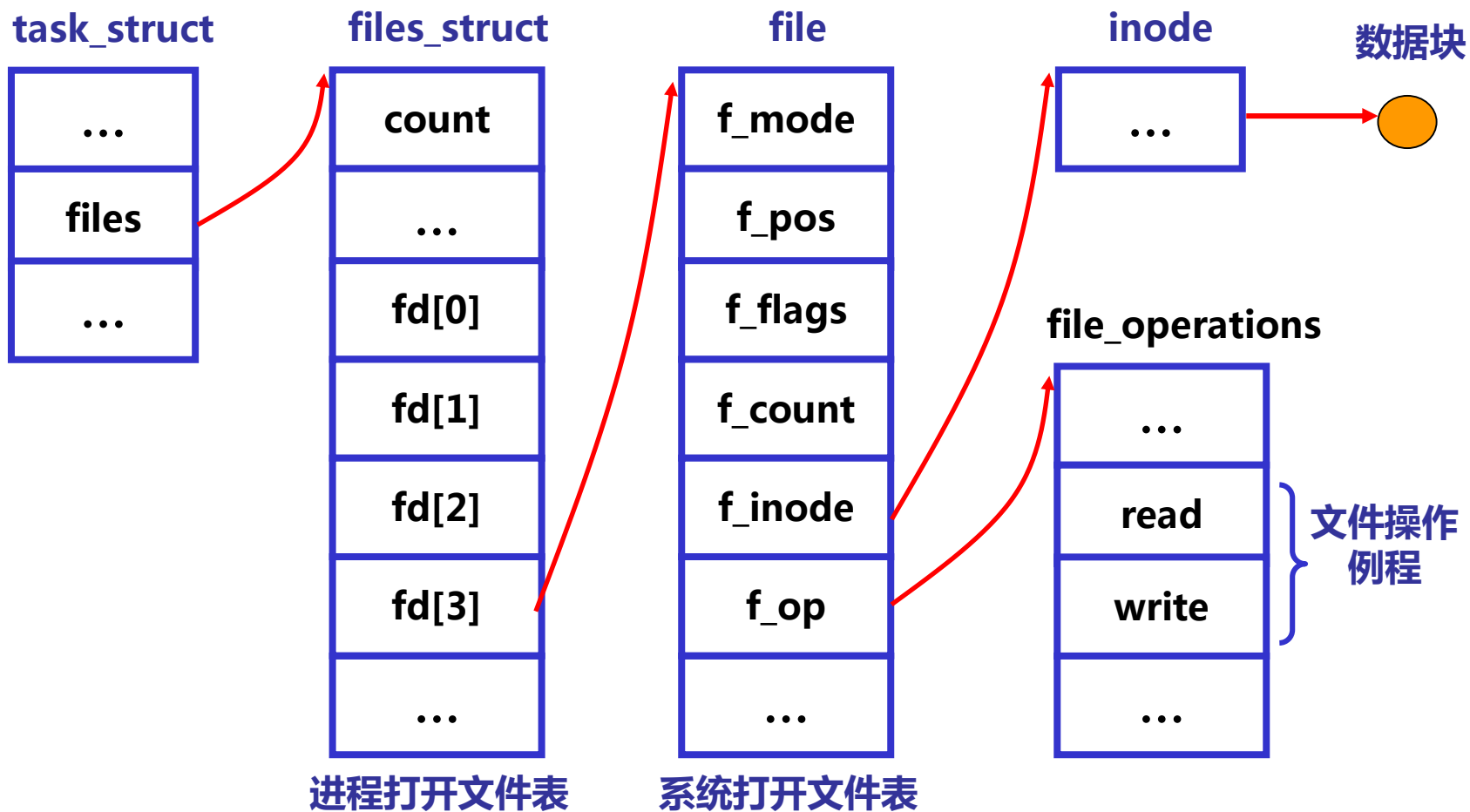
(2) 系统打开文件表

由若干file结构组成

每个file结构包含下列字段：

- ✓ f_mode：文件打开的方式
- ✓ f_pos：文件当前的读写位置
- ✓ f_flags：文件属性（有许多标志位）
- ✓ f_count：引用计数
- ✓ f_inode：指向VFS中该文件的内存I节点
- ✓ f_op：指向file_operations结构，该结构包含文件操作的各种例程

8.2 ext2 文件系统





8.2 ext2 文件 系 统

说明：

- ✓ 当父子进程共享同一个文件时，它们使用同一个file结构；
- ✓ 以上介绍的内存I节点以及系统打开文件表、用户打开文件表，实际上是VFS提供的。



8.3 Linux 虚拟文件系统VFS

一、VFS的原理

VFS建立在具体文件系统之上，为用户程序提供一个统一、抽象、虚拟的文件系统接口。

虚拟文件系统也叫虚拟文件系统转换（Virtual File system Switch，VFS）。

- ✓ VFS的各种数据结构都是随时建立或删除的，在磁盘上并不永久保存，只能存放在内存之中。
- ✓ VFS是Linux核心的一部分，其他内核子系统与VFS打交道，VFS又管理其他具体的文件系统。所以VFS是文件系统和Linux内核的接口，VFS以统一数据结构管理各种具体的文件系统，接受用户层对文件系统的各种操作。



8.3 Linux 虚 拟 文 件 系 统 VFS

二、VFS的通用文件模型

VFS的文件模型由下列对象组成：

- 1) 超级块super_block：存放已挂装文件系统的有关信息
- 2) 索引节点inode：一个具体文件的信息
- 3) 文件file：打开文件与进程之间进行交互的有关信息
- 4) 目录项dentry



8.3 Linux 虚拟文件系统VFS

VFS与具体文件系统的接口：

1) super_operations

超级块操作，实现文件系统的挂装、卸载等

用来将VFS对超级块的操作转化为具体文件系统处理这些操作的函数 内核
为每一个挂装的文件系统分配一个超级块

VFS超级块super_block包括的内容有：

- ✓ s_fs_info：指向具体文件系统的超级块信息
- ✓ s_op：指向super_operations



8.3 Linux 虚拟文件系统 VFS

2) file_operations

用来将VFS对file结构的操作转化为具体文件系统处理这些操作的函数
file有一个字段f_op，指向file_operations

3) inode_operations

用来将VFS对索引节点的操作转化为具体文件系统处理相应操作的函数



8.3 Linux 虚 拟 文 件 系 统 VFS

三、对Linux文件系统的整体理解

从资源管理的角度来看，

每个文件要在磁盘上占用哪些资源？

当文件被打开后，在内存要占用哪些资源（数据结构）？



对Linux文件系统的整体理解

每个文件要在磁盘上占用3种资源：

- 1) 1个目录项：记录文件名和I节点号
- 2) 1个磁盘I节点：记录除了文件名之外的文件属性（包括地址）
- 3) 若干盘块：存放文件内容

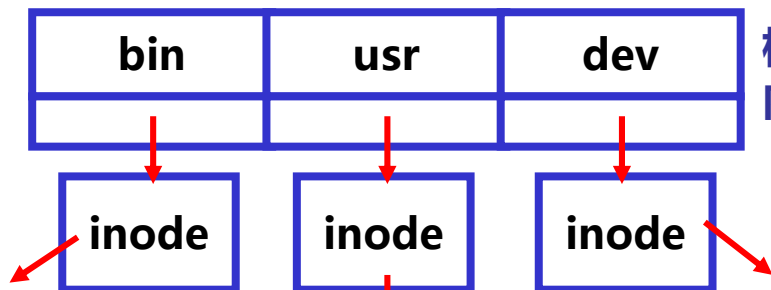
文件被打开后，在内存增加3种资源：

- 1) 1个内存I节点：文件说明信息
- 2) 系统打开文件表的1项：1个file结构
- 3) 用户打开文件表中的1项

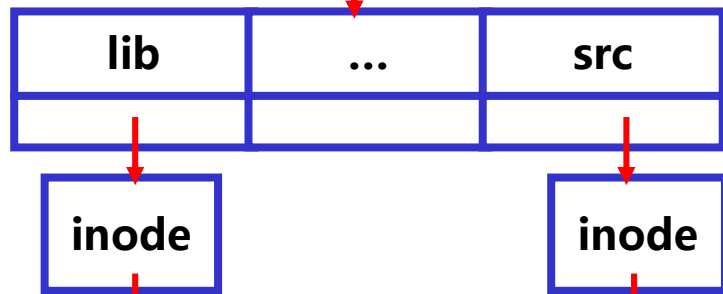
一个文件系统的结构例

根目录

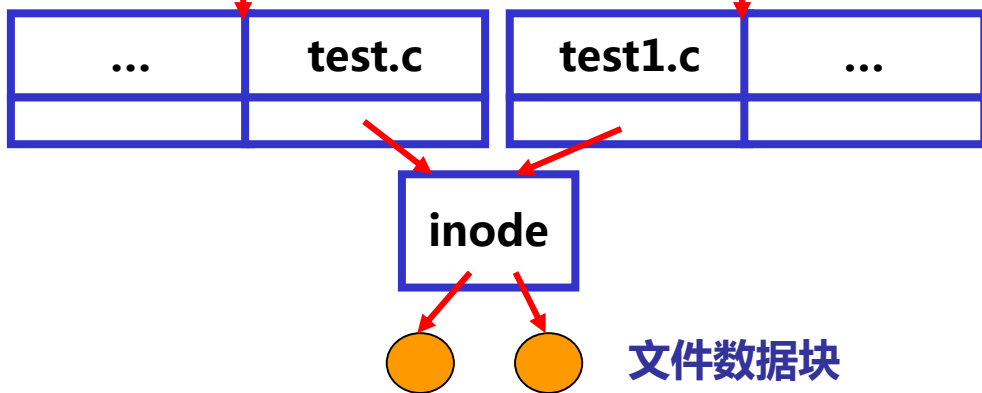
根目录文件的内容



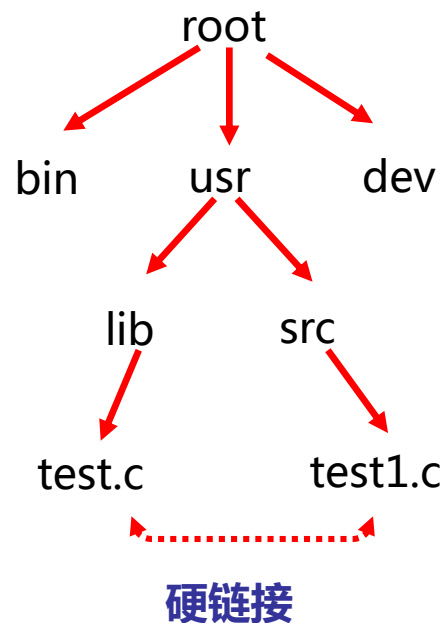
目录文件/usr的内容



目录文件/usr/src的内容



文件数据块





8.3 Linux 虚拟文件系统VFS

四、查找文件

根据路径名得到磁盘I节点

【例】查找文件/usr/src/test1.c。文件系统结构见前例。

1) 按照超级块中根目录的I节点号找到根目录的数据块；

2) 在根目录的数据块中找到/usr对应的目录项，然后

找到/usr的I节点→/usr文件的数据块地址→/usr/src对应的目录项
→/usr/src的I节点→/usr/src文件的数据块地址
→/usr/src/test1.c对应的目录项→/usr/src/test1.c的I节点；

3) 得到I节点后，就可以获取文件数据块地址，从而访问文件。



8.3 Linux 虚拟文件系统VFS

五、文件系统调用

1) open

```
fd = open(char *filename, int oflag [, int mode])
```

filename : 文件名

oflag : 打开方式

mode : 给出文件访问权限，文件创建时使用

功能：根据oflag的值打开文件，也可以创建文件。

成功则返回文件描述符，否则返回-1。

2) close

```
close(int fd) ; //关闭文件
```



文件系统调用

3) read

```
n = read(int fd, char* buf, int nbytes);
```

功能：读fd所指文件，从当前位置读nbytes个字节放入buf，返回实际读出的字节数。

4) write

```
n = write(int fd, char* buf, int nbytes);
```

功能：写fd所指文件，将buf中的nbytes个字节写入文件，返回实际写入的字节数。

5) lseek

```
pos = lseek(int fd, long offset, int origin)
```

功能：文件指针定位。



8.4 Linux的设备管理

一、基本思想

1) 设备作为一种特殊文件

字符特殊文件：用于字符设备，如键盘

块特殊文件：用于块设备，如磁盘

2) 用户通过文件系统访问设备，VFS为应用程序隐藏设备文件和普通文件的差异

3) 每类设备对应自己的驱动程序

4) 每个设备对应文件系统的一个I节点，都有一个文件名。

设备的文件名一般由2部分构成：

- ✓ 主设备号：设备类型，可唯一确定设备驱动程序和接口
- ✓ 次设备号：具体设备在同类设备中的序号



8.4 Linux的设备管理

二、设备驱动程序与文件系统的接口

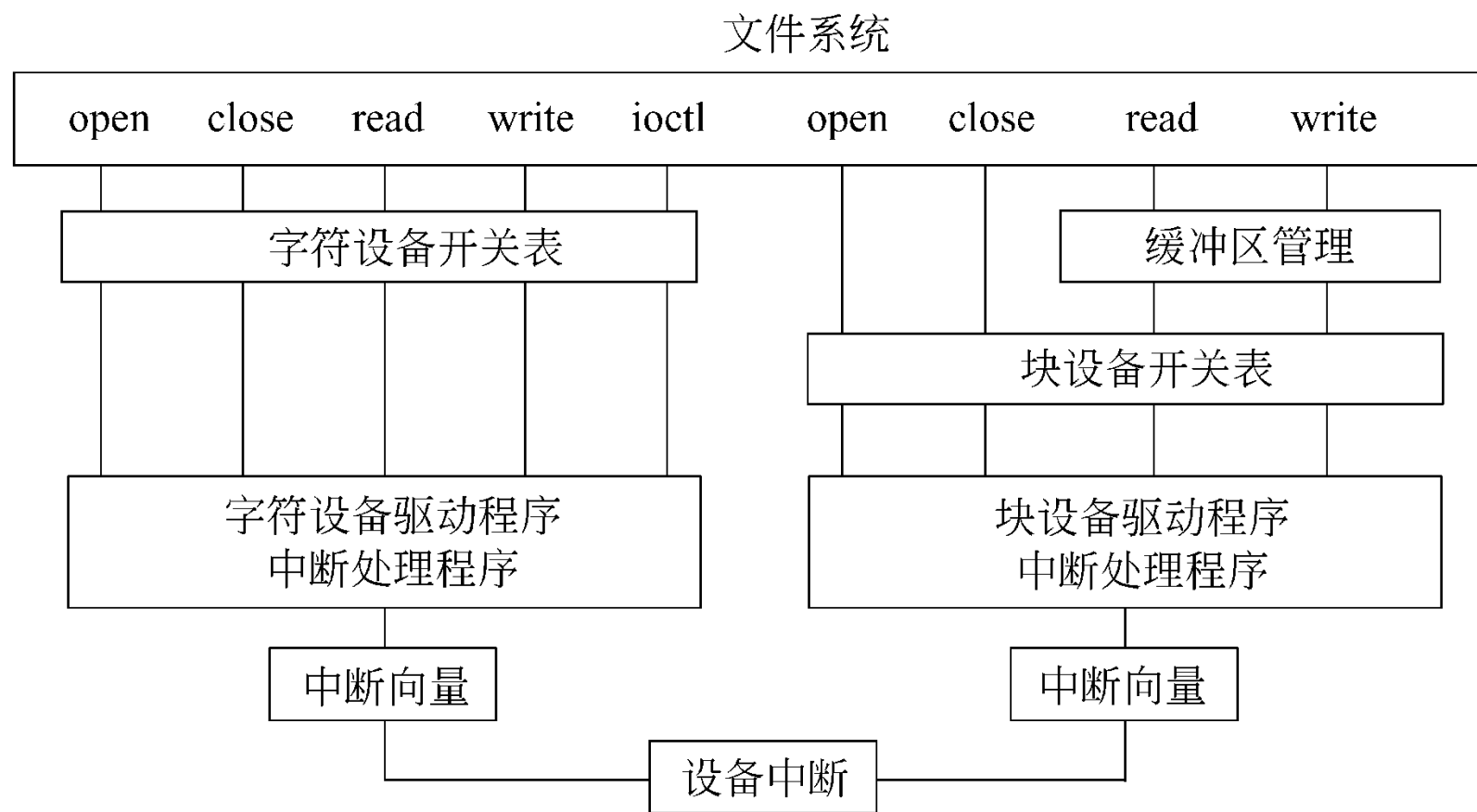
设备开关表：

引导系统调用转向相应驱动程序中的函数，包括块设备开关表与字符设备开关表

指出缺省的文件操作（open, close, read, write等）与相应设备操作函数的对应关系

例如：进程请求分配一个设备，即打开一个设备，可以通过系统调用open()来完成。

8.4 Linux的设备管理



文件系统与设备驱动程序的关联



设备开关表例

表 10-2 用于块设备文件的函数与缺省文件的操作方法对应表

方 法	用于块设备文件的函数	方 法	用于块设备文件的函数
open	blkdev_open()	write	generic_file_write()
release	blkdev_close()	mmap	generic_file_mmap()
llseek	block_llseek()	fsync	block_fsync()
read	generic_file_read()	ioctl	blkdev_ioctl()



8.4 Linux的设备管理

三、块设备缓冲

系统在文件系统和设备驱动程序之间设置了缓冲池
含有最近使用过的数据块