

# 第6章 Linux文件系统



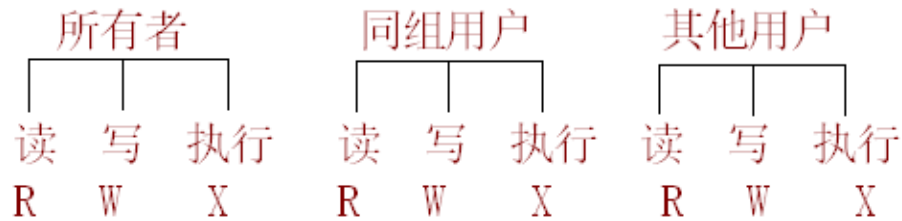
文件系统是操作系统用于确认磁盘或分区上的文件的方法和数据结构。

- 文件系统是负责存取和管理文件信息的机构，用于对数据、文件以及设备的存取控制，它提供对文件和目录的分层组织形式、数据缓冲以及对文件存取权限的控制功能。
- Linux使用标准的目录结构—树型结构，无论操作系统管理几个磁盘分区，这样的目录树只有一个。

# 访问权限和文件模式

Linux给文件设定了一定的访问权限

Linux对文件的访问设定了三级权限：文件所有者，与文件所有者同组的用户，其他用户。对文件的访问主要是三种处理操作：读取、写入和执行



文件访问权和访问模式

# Linux文件系统

文件系统：文件存在的物理空间，Linux系统中每个分区都是一个文件系统，都有自己的目录层次结构

Linux文件系统使用索引节点来记录文件信息，系统给每个索引节点分配了一个号码，称为**索引节点号**。文件系统正是靠这个索引节点号来识别一个文件

## 软链接和硬链接

可以用链接命令ln（Link）对一个已存在的文件再建立一个新的链接，而不复制文件的内容

**硬链接(hard link):** 让一个文件对应一个或多个文件名，或者说把我们使用的文件名和文件系统使用的节点号链接起来，这些文件名可以在同一目录或不同目录

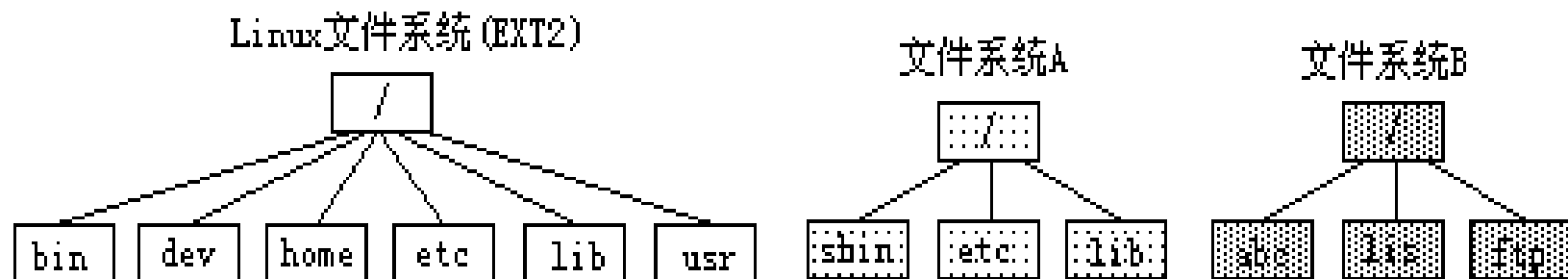
**软链接（也叫符号链接）：** 是一种特殊的文件，这种文件包含了另一个文件的任意一个路径名。这个路径名指向位于任意一个文件系统的任意文件，甚至可以指向一个不存在的文件

# 安装文件系统

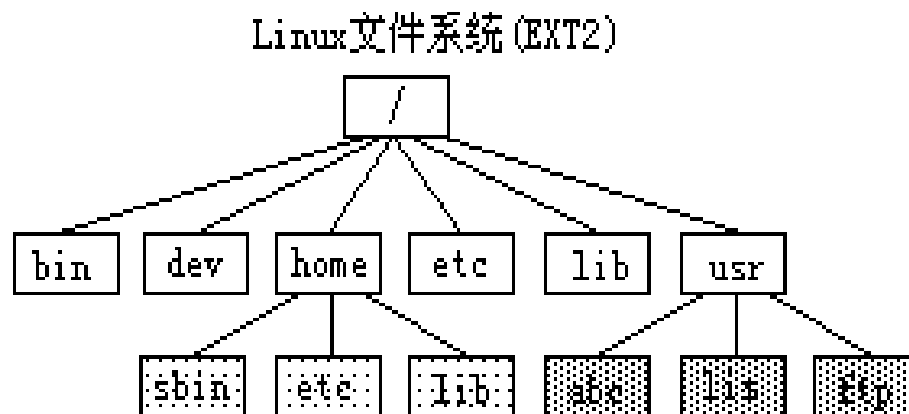
将一个文件系统的顶层目录挂到另一个文件系统的子目录上，使它们成为一个整体，称为“安装（mount）”。把该子目录称为“安装点(mount point)”

EXT2/3/4是Linux的标准文件系统，系统把它的磁盘分区做为系统的根文件系统，EXT2/3/4以外的文件系统则安装在根文件系统下的某个目录下，成为系统树型结构中的一个分枝

安装一个文件系统用mount命令



(a) 安装前的三个独立的文件系统



(b) 安装后的文件系统

文件系统的安装

## 文件系统具有以下主要功能：

- 1.对文件存储设备进行管理，分别记录空闲区和被占用区，以便于用户创建、修改以及删除文件时对空间的操作。
- 2.对文件和目录的按名访问、分层组织功能。
- 3.创建、删除及修改文件功能。
- 4.数据保护功能。
- 5.文件共享功能



# 6.1

Part One

## Linux文件系统概述

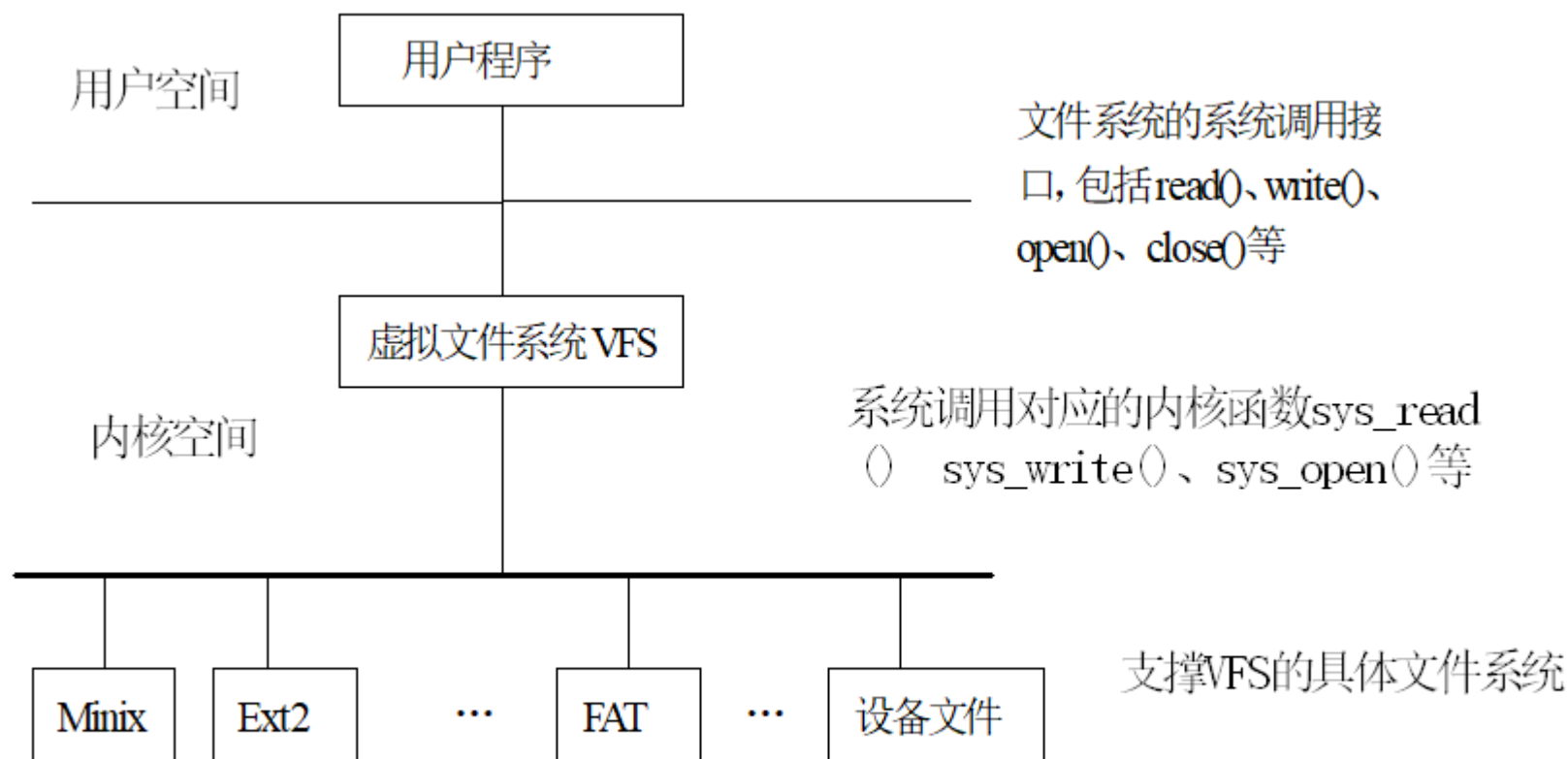
---

# 虚拟文件系统VFS

Linux最初采用Minix的文件系统，其大小限于64MB，文件名长度也限于14个字节

为了支持多种不同的文件系统，采用了虚拟文件系统VFS（Virtual File system）技术。虚拟文件系统是对*多种实际文件系统的共有功能的抽象*，它*屏蔽了各种不同文件系统在实现细节上的差异*，为用户程序提供了统一的、抽象的、标准的接口以便对文件系统进行访问，如打开，读，写等操作。

*虚拟文件系统VFS确保了对所有文件的访问方式都是完全相同的。*



VFS与具体文件系统之间的关系

**VFS** 是底层文件系统的主要组件(接口)。这个组件导出一组接口，然后将它们抽象到行为可能差异很大的各个文件系统。**VFS**具有两个针对文件系统对象的缓存：**inode 索引节点对象**和**dentry 目录项对象**，它们缓存最近使用过的文件系统对象。

每个文件系统实现（如 ext2、JFFS2等等）可以导出一组通用接口供 VFS 使用。缓冲区缓存会缓存文件系统和相关块设备之间的请求。

## VFS的索引节点

文件系统处理文件所需要的所有信息都放在称为索引节点的数据结构[inode](#)中。

在同一个文件系统中，每个索引节点号都是唯一的

inode 中有两个设备号，i\_dev（常规文件的设备号）和i\_rdev（某一设备的设备号）

# 目录项对象

每个文件除了有一个索引节点inode数据结构外，还有一个目录项dentry数据结构。

dentry结构代表的是逻辑意义上的文件，描述的是文件逻辑上的属性，目录项对象在磁盘上并没有对应的映像

inode结构代表的是物理意义上的文件，记录的是物理上的属性，对于一个具体的文件系统，其inode结构在磁盘上就有对应的映像

一个索引节点对象可能对应多个目录项对象

# 文件系统的安装

安装一个文件系统实际上是安装一个物理设备

自己（一般是超级用户）安装文件系统时，需要指定三种信息：文件系统的名称、包含文件系统的物理块设备、文件系统在已有文件系统安装点。

`$ mount -t iso9660 /dev/hdc /mnt/cdrom` 其中，  
iso9660是光驱文件系统的名称，/dev/hdc是包含文件系统的物理块设备，/mnt/cdrom就是将要安装到的目录，即安装点。

在用户程序中要安装一个文件系统则可以调用mount（）系统调用。安装过程主要工作是创建安装点对象，将其挂接到根文件系统的指定安装点下，然后初始化超级块对象，从而获得文件系统基本信息和相关的操作。

综合看来，Linux虚拟文件系统采用了面向对象设计思想，文件系统中定义的VFS相当于面向对象系统中的抽象基类，从它出发可以派生出不同的子类，以支持多种具体文件系统，但从效率考虑内核纯粹使用C语言编程，故没有直接利用面向对象的语义。



# 6.2 Part Two

## Ext2文件系统格式

---

## 6.2.1 Ext2文件系统

The Second Extended File System(ext2)文件系统是Linux系统中的标准文件系统，主要包括普通文件、目录文件、特殊文件和符号链接文件。Ext2文件系统是通过对Minix的文件系统进行扩展而得到的，其存取文件的性能良好，可以管理特大磁盘分区，文件系统最大可达4TB。早期Linux都使用ext2文件系统。

在ext2文件系统中，文件由包含有文件所有信息的**节点inode**进行唯一标识。

Ext2文件系统采用**三级间接块**来存储数据块指针，并以块（默认为1KB）为单位分配空间。

在ext2系统中，所有元数据结构的大小均基于“**块**”，而不是“扇区”。块的大小随文件系统的大小而有所不同。而一定数量的块又组成一个块组，每个块组的起始部分有多种描述该块组各种属性的元数据结构。

每个块组依次包括**超级块**、**块组描述符**、**块位图**和**节点inode位图**、**inode表**及数据块区。

## 1. 超级块

每个**ext2**文件系统都必须包含一个超级块，其中存储了该文件系统的大量基本信息，如块的大小、每块组中包含的块数等。同时系统会对超级块进行备份，备份被存放在块组的第一个块中。超级块的起始位置为其所在分区的第**1024**个字节，占用**1KB**的空间，

## 2. 块组描述符

一个块组描述符用以描述一个块组的属性。块组描述符组由若干块组描述符组成，描述了文件系统中所有块组的属性，存放于超级块所在块的下一个块中。

## 3. 块位图和节点inode位图

块位图和inode位图的每一位分别指出块组中对应的哪个块或inode是否被使用。

## 4. 节点inode表

节点inode表用于跟踪定位每个文件，包括位置、大小等，不包括文件名。一个块组只有一个节点inode表。

## 5. 数据块

数据块中存放文件的内容，包括目录表、扩展属性、符号链接等

# 6.3

Part Three

## EXT3和EXT4文件系统

---

### 6.3.1 EXT3文件系统

EXT3是第三代扩展文件系统（Third extended file system），EXT3是在EXT2基础上增加日志形成的一个日志文件系统，常用于Linux操作系统。

日志文件系统最大的特点是将整个磁盘的写入动作完整记录在磁盘的某个区域上，以便有需要时可以回溯追踪。

除开日志文件系统所具有的优点，Ext3的特点还主要有：

- （1）ext3文件系统在非正常关机状况下，系统无需检查文件系统，而且ext3的恢复时间也极短。
- （2）ext3文件系统能够极大地提高文件系统的完整性，避免了意外宕机对文件系统的破坏。
- （3）ext3文件系统可以不经任何更改，而直接加载成为ext2文件系统。由ext2文件系统转换成ext3文件系统也非常容易。
- （4）3种日志模式可选：日记、顺序、回写。可适应不同场合对日志模式的要求。
- （5）便于移植，无论是硬件体系或是内核修改，其移植工作均较容易。



## 6.3.2 EXT4文件系统

第四代扩展文件系统EXT4（Fourth extended filesystem）是Linux系统下的日志文件系统，是ext3文件系统的后继版本。2008年12月25日，Linux Kernel 2.6.28的正式版本发布。随着这一新内核的发布，Ext4文件系统也结束实验期，成为稳定版。

## Ext4文件系统的特点主要包括：

- 1.Ext4的文件系统容量达到1EB，而文件容量则达到16TB；
2. Ext4理论上支持无限数量的子目录；
- 3.Ext4文件系统具有64位空间记录块数量；
- 4.Ext4在文件系统层面实现了持久预分配并提供相应的API，比应用软件自己实现更有效率；
- 5.Ext4支持更大的i-节点和支持快速扩展属性和i-节点保留；
- 6.Ext4给日志数据添加了校验功能，日志校验功能可以很方便地判断日志数据是否损坏；
- 7.Ext4支持在线碎片整理，并将提供e4defrag工具进行个别文件或整个文件系统的碎片整理。

## ext文件系统 — 缺点

- **ext**文件系统是针对**IDE**设备设计，逻辑块是**512B**的倍数，不太适合扇区大小的闪存设备
- **ext**文件系统没有提供对基于扇区的擦除/写操作的良好管理，比如擦除一个字节需复制整扇区到**RAM**中
- **ext**文件系统在出现电源故障时不防崩溃
- **ext**文件系统不支持损耗平衡，缩短闪存寿命

# 6.4 Part Four

## 嵌入式文件系统JFFS2

---

## 6.4.1 嵌入式文件系统

嵌入式文件系统是指嵌入式系统中实现文件存取、管理等功能的模块，这些模块提供一系列文件输入输出等文件管理功能，为嵌入式系统和设备提供文件系统支持。在嵌入式系统中，文件系统是嵌入式系统的一个组成模块。它是作为系统的一个可加载选项提供给用户，由用户决定是否需要加载它。嵌入式文件系统具有结构紧凑、使用简单便捷、安全可靠及支持多种存储设备、可伸缩、可剪裁、可移植等特点。

# 主流的嵌入式操作系统的文件系统

## QNX

QNX提供多种资源管理器，包括各种文件系统和设备管理，支持多个文件系统同时运行；包括提供完全的POSIX以及UNIX语法的文件系统，支持多种闪存设备的嵌入式文件系统，支持对多种文件服务器WINDOWS、LANManager等的透明访问的SMB文件系统、FAT文件系统、CD-ROM文件系统等，并支持多种外部设备。

# VxWorks

**VxWorks**的文件系统提供的组件—**FFS** “快速文件系统”非常适合于实时系统的应用。它包括几种支持使用块设备（如磁盘）的本地文件系统，这些设备都使用一个标准的接口从而使得文件系统能够被灵活的在设备驱动程序上移植。另外，也支持**SCSI**磁带设备的本地文件系统。 **VXWORKS**的 I/O的体系结构甚至还支持一个单独的从系统上同时并存几个不同的文件系统，支持四种文件系统**FAT**、**RT11FS**、**RAWFS**、**TAPES**。

## Yaffs/yaffs2

Yaffs(Yet Another Flash File System)/yaffs2是专为嵌入式系统使用 NAND型闪存而设计的日志型文件系统。速度快，挂载时间很短，对内存的占用较小。另外它还是跨平台的文件系统，除了Linux和eCos，还支持WinCE， pSOS和ThreadX等。

Yaffs/yaffs2自带NAND芯片的驱动，并且为嵌入式系统提供了直接访问文件系统的API，用户可以不使用Linux中的MTD与VFS，直接对文件系统操作。当然，yaffs也可与MTD驱动程序配合使用。yaffs与 yaffs2的主要区别在于，前者仅支持小页(512 Bytes) NAND闪存，后者则可支持大页(2KB) NAND闪存。同时，yaffs2在内存空间占用、垃圾回收速度、读/写速度等方面均有大幅提升。



# Cramfs

Cramfs(Compressed ROM File System)是Linux的创始人 Linus Torvalds参与开发的一种只读的压缩文件系统。在cramfs文件系统中，每一页(4KB)被单独压缩，可以随机页访问，其压缩比高达2:1，为嵌入式系统节省大量的Flash存储空间，使系统可通过更低容量的FLASH存储相同的文件，从而降低系统成本。

# NFS

网络文件系统**NFS (Network File System)**是由**Sun**开发并发展起来的一项在不同机器、不同操作系统之间通过网络共享文件的技术。在嵌入式**Linux**系统的开发调试阶段，可以利用该技术在主机上建立基于**NFS** 的根文件系统，挂载到嵌入式设备，可以很方便地修改根文件系统的内容。

# 闪存存储器Flash Memory

- 闪存存储器是一种半导体集成电路存储器
  - 在EEPROM的基础上进化而来
  - 英文原文是Flash Memory，简称闪存
  - 主要物理特点是非易失
  - Non-Volatile，也叫做不挥发

# 两种类型的Flash

- Nor Flash，称为或非型闪存，或者NOR闪存
- Nand Flash，称为与非型闪存，或者NAND闪存
- Nor Flash是在EEPROM基础上发明的。
  - Intel公司于1983年首次提出，在1988年商品化。
- Nand Flash是1989年东芝公司和三星公司发明的。
- 十几年以来，世界主要闪存生产商分成Nor和Nand两大技术阵营，积极开展研发和生产。
  - Nor阵营主要有Intel和AMD公司
  - Nand阵营主要有Toshiba和Samsung公司

# Nor Flash和Nand Flash的对比

Nor Flash通常容量较小，其主要特点是程序代码可以直接在Flash内运行。Nor Flash 具有RAM 接口，易于访问，缺点是擦除电路复杂，写速度和擦除速度都比较慢，最大擦写次数约10万次，典型的块大小是128kB。

Nand Flash通常容量较大，具有很高的存储密度，从而降低了单位价格。Nand Flash 的块尺寸较小，典型大小为8kB，擦除速度快，使用寿命也更长，最大擦写次数可以达到100 万次，但是其访问接口是复杂的I/O 口，并且坏块和位反转现象较多，对驱动程序的要求较高。

**一般结论：**由于**Nor Flash**和**Nand Flash**各具特色，因此它们的用途也各不相同，**Nor Flash** 一般用来存储体积较小的代码，而**Nand Flash** 则用来存放大体积的数据。

# Nor Flash和Nand Flash的共同特点

- 向芯片中写数据必须先将芯片中对应的内容清空，然后再写入。
- 闪存擦写的次数都是有限的，当闪存的使用寿命快到时，经常会出现写操作失败。超过了最大可擦写次数，这个擦写块就成为坏块(bad block)。
- 为了延长使用寿命，不要对某个特定区域反复地进行写操作。我们应该在尽量小的影响性能的前提下，使擦写操作均匀的分布在每个擦写块上。这个过程叫做磨损均衡（wear leveling）。
- 闪存的读写操作不仅是一个物理操作，还需要算法支持。一般在驱动程序的**内存技术设备（MTD, Memory Technology Drivers）**模块中或者在**闪存转换层（FTL, Flash Translation Layer）**内实现，具体算法同芯片生产商以及芯片信号有关。

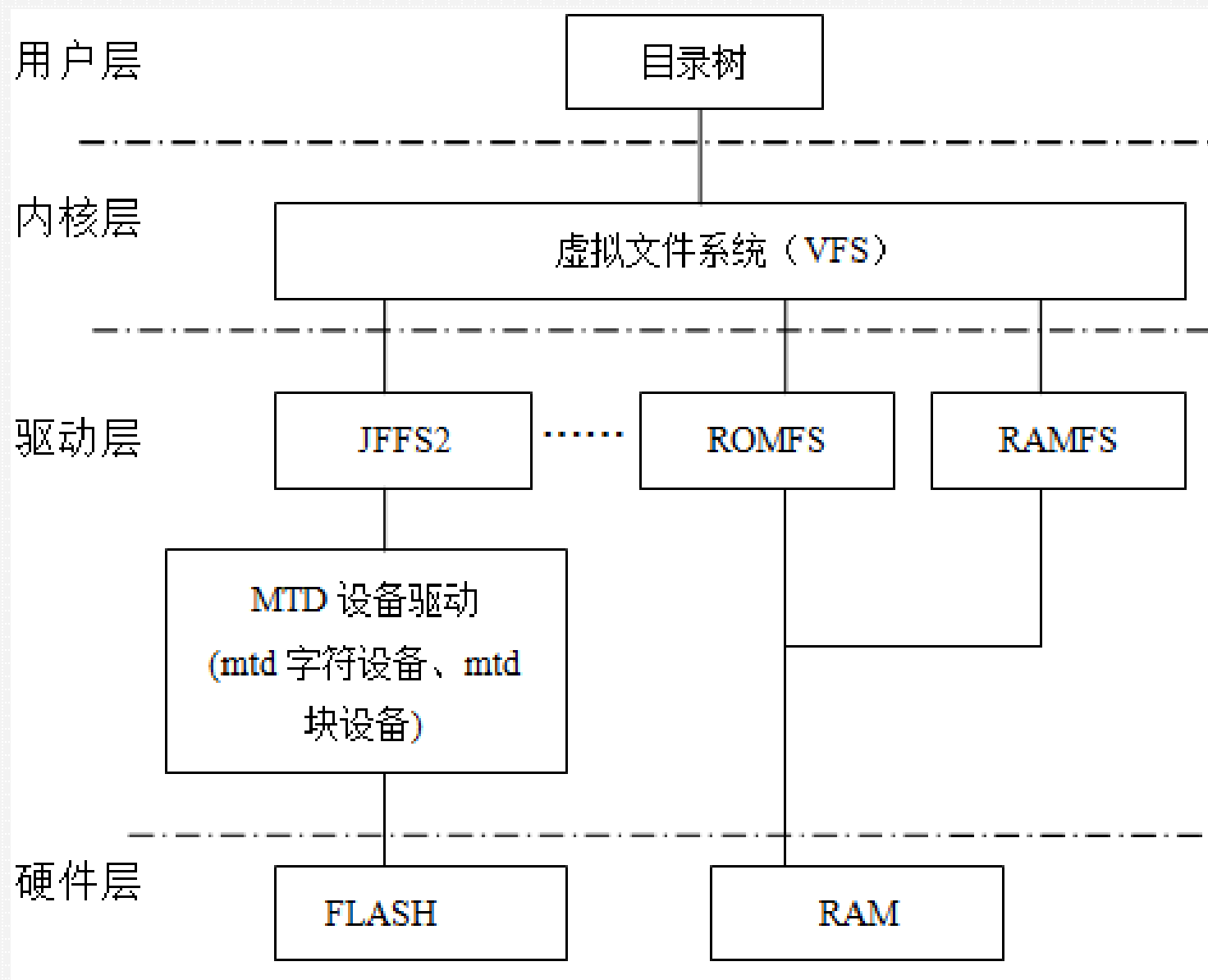
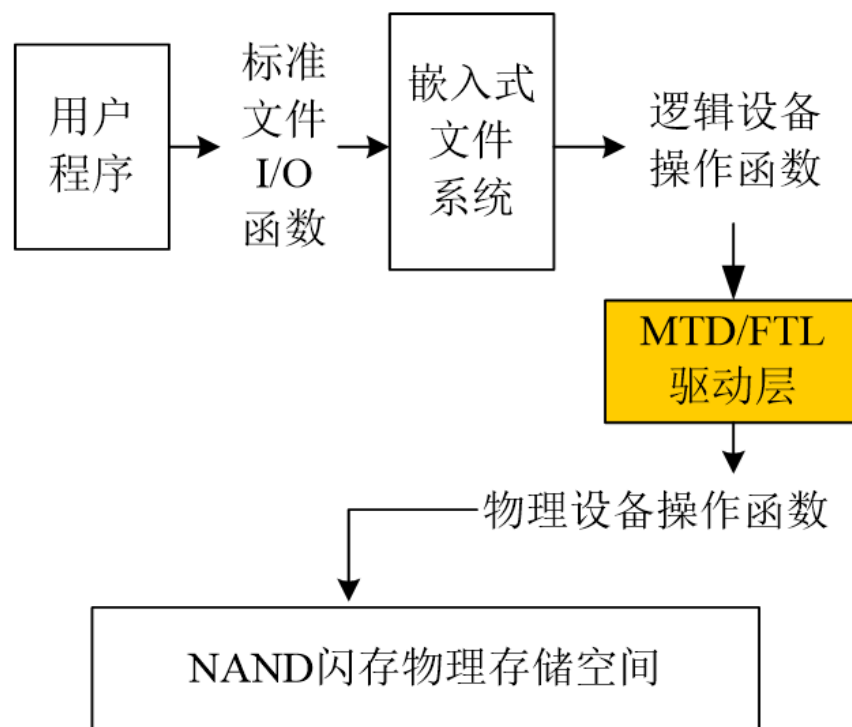


图6-2 嵌入式Linux文件系统框架

# 物理闪存操作层次结构

- MTD/FTL层向上将闪存设备抽象成逻辑设备（逻辑页面和块），为文件系统提供对物理设备操作的接口；向下实现对物理闪存设备的读写、**擦除(erase)**、**ECC校验**等工作。参看下图。





为了延长Flash 的整体寿命需要均匀地使用各个块，这就需要磨损均衡的功能；

为了提高Flash 存储器的利用率， 还应该对存储空间碎片收集功能；

在嵌入式系统中， 要考虑出现系统意外掉电的情况，所以文件系统还应该对掉电保护的功能， 保证系统在出现意外掉电时也不会丢失数据。

**因此在Flash 存储设备上， 目前主要采用了专门针对Flash 存储器的要求而设计的JFFS2**

## 6.4.2 JFFS2嵌入式文件系统

### 1. JFFS2 文件系统简介

JFFS(Journaling Flash File System)是瑞典的Axis Communications 公司专门针对嵌入式系统中的Flash 存储器的特性而设计的一种日志文件系统。

Redhat公司的David Woodhouse 在JFFS 的基础上进行了改进，从而发布了JFFS2(Journaling Flash File System Version 2)。

和JFFS相比，JFFS2 支持更多节点类型，提高了磨损均衡和碎片收集的能力，增加了对硬链接的支持。JFFS2 还增加了数据压缩功能，这更利于在容量较小的Flash 中使用。和传统的Linux 文件系统如ext2 相比，JFFS2 处理擦除和读写操作的效率更高， 并且具有完善的掉电保护功能， 使存储的数据更加安全。

### 3. JFFS2的不足之处

（1）挂载时间过长。JFFS2 的挂载过程需要对闪存从头到尾的扫描，这个过程比较花费时间。

（2）磨损平衡具有较大随意性。JFFS2 对磨损平衡是用概率的方法来解决的，这很难保证磨损平衡的确定性。在某些情况下，可能造成对擦写块不必要的擦写操作；在某些情况下，又会引起对磨损平衡调整的不及时。

（3）扩展性很差。首先，闪存越大，闪存上节点数目越多挂载时间就越长。其次，虽然 JFFS2 尽可能的减少内存的占用，但实际上对内存的占用量是同节点数和闪存上的节点数成正比的。

# 6.5

Part Five

## YAFFS与YAFSS2文件系统简介

---

## 6.5.1 YAFFS文件系统

**YAFFS**（Yet Another Flash File System）是第一个专门为**NAND Flash**存储器设计的嵌入式文件系统，适用于大容量的存储设备。**YAFFS**遵守**GPL**（General Public License）协议并且可以免费使用。

**YAFFS** 是基于日志的文件系统，提供磨损平衡和掉电恢复的健壮性。它还为大容量的**Flash** 芯片进行了优化设计，针对启动时间和**RAM** 的使用做了改进。**YAFFS**适用于大容量的存储设备，已经在**Linux** 和**WinCE** 商业产品中使用。

**YAFFS**中，文件是以固定大小的数据块进行存储的，块的大小可以是**512**字节、**1024**字节或者**2048**字节。这种实现取决于它能够将一个数据块头和每个数据块关联起来。每个文件都有一个数据块头与之相对应，数据块头中保存了**ECC**(纠错码)和文件系统的组织信息，用于错误检测和坏块处理。**YAFFS**充分考虑了**NAND Flash**的特点，**YAFFS**把这个数据块头存储在**Flash**的**16**字节备用空间中。当文件系统被挂载时，只须扫描存储器的备用空间就能将文件系统信息读入内存，并且驻留在内存中，不仅加快了文件系统的加载速度，也提高了文件的访问速度，但是增加了内存的消耗。

# 6.6

Part Six

## 根文件系统

---

## 6.6.1 根文件系统概述

根文件系统是一种特殊的文件系统，该文件系统不仅具有普通文件系统的存储数据文件的功能，它还是内核启动时所挂载（**mount**）的第一个文件系统，内核代码的映像文件保存在根文件系统中，系统引导启动程序会在根文件系统挂载之后从中把一些初始化脚本和服务加载到内存中去运行。



**Linux**启动时，第一个必须挂载的是根文件系统；若系统不能从指定设备上挂载根文件系统，则系统会出错而退出启动。成功之后可以自动或手动挂载其他的文件系统。因此，一个系统中可以同时存在不同的文件系统。

在 Linux 中将一个文件系统与一个存储设备关联起来的过程称为挂载（mount）。使用 mount 命令将一个文件系统附着到当前文件系统层次结构中。在执行挂载时，要提供文件系统类型、文件系统和一个挂载点。根文件系统被挂载到根目录下“/”上后，在根目录下就有根文件系统的各个目录和文件：/bin、/sbin、/mnt等，再将其他分区挂接到/mnt目录上，/mnt目录下就有这个分区的各个目录和文件。

**Linux**根文件系统中一般有如下几个目录：

### **(1) /bin目录**

该目录下的命令可以被**root**与一般账号所使用，由于这些命令在挂接其它文件系统之前就可以使用，所以**/bin**目录必须和根文件系统在同一个分区中。

**/bin**目录下常用的命令有：**cat**、**chgrp**、**chmod**、**cp**、**ls**、**sh**、**kill**、**mount**、**umount**、**mkdir**、**[**、**test**等。其中“**[**”命令就是**test**命令，在利用**Busybox**制作根文件系统时，在生成的**bin**目录下，可以看到一些可执行的文件，也就是可用的一些命令。

## (2) /sbin 目录

该目录下存放系统命令，即只有系统管理员能够使用的命令，系统命令还可以存放在/usr/sbin， /usr/local/sbin目录下， /sbin目录中存放的是基本的系统命令，它们用于启动系统和修复系统等，与/bin目录相似，在挂接其他文件系统之前就可以使用/sbin，所以/sbin目录必须和根文件系统在同一个分区中。

/sbin目录下常用的命令有：shutdown、reboot、fdisk、fsck、init等，本地用户自己安装的系统命令放在/usr/local/sbin目录下。

### (3) /dev目录

该目录下存放的是设备与设备接口的文件，设备文件是Linux中特有的文件类型，在Linux系统下，以文件的方式访问各种设备，即通过读写某个设备文件操作某个具体硬件。比如通过"dev/ttySAC0"文件可以操作串口0，通过"/dev/mtdblock1"可以访问MTD设备的第2个分区。比较重要的文件有/dev/null，/dev/zero， /dev/tty， /dev/lp\*等。

## （4）/etc目录

该目录下存放着系统主要的配置文件，例如人员的账号密码文件、各种服务的其实文件等。一般来说，此目录的各文件属性是可以让一般用户查阅的，但是只有root有权限修改。对于PC上的Linux系统，/etc目录下的文件和目录非常多，这些目录文件是可选的，它们依赖于系统中所拥有的应用程序，依赖于这些程序是否需要配置文件。在嵌入式系统中，这些内容可以大为精减。

### （5）/lib目录

该目录下存放共享库和可加载驱动程序，共享库用于启动系统。

### （6）/home目录

系统默认的用户文件夹，它是可选的，对于每个普通用户，在/home目录下都有一个以用户名命名的子目录，里面存放用户相关的配置文件。

### （7）/root目录

系统管理员（**root**）的主文件夹，即是根用户的目录，与此对应，普通用户的目录是/home下的某个子目录。

## (8) /usr目录

/usr目录的内容可以存在另一个分区中，在系统启动后再挂接到根文件系统中的/usr目录下。里面存放的是共享、只读的程序和数据，这表明/usr目录下的内容可以在多个主机间共享，这些主要也符合文件系统层次标准FHS标准的。文件系统层次标准FHS(Filesystem Hierarchy Standard, FHS)，它规范了在根目录“/”下面各个主要的目录应该放置什么样的文件。/usr目录在嵌入式中可以精减。

## （9）/var目录

与/usr目录相反，/var目录中存放可变的数据，比如spool目录（mail, news），log文件，临时文件。

## （10）/proc目录

这是一个空目录，常作为proc文件系统的挂接点，proc文件系统是个虚拟的文件系统，它没有实际的存储设备，里面的目录，文件都是由内核临时生成的，用来表示系统的运行状态，也可以操作其中的文件控制系统。



### （11）/mnt目录

用于临时挂载某个文件系统的挂载点，通常是空目录，也可以在里面创建一引起空的子目录，比如/mnt/cdram /mnt/hda1 。用来临时挂载光盘、移动存储设备等。

### （12）/tmp目录

用于存放临时文件，通常是空目录，一些需要生成临时文件的程序用到的/tmp目录下，所以/tmp目录必须存在并可以访问。

对于嵌入式Linux系统的根文件系统来说，一般可能没有上面所列出的那么复杂，比如嵌入式系统通常都不是针对多用户的，所以/home这个目录在一般嵌入式Linux中可能就很少用到。一般说来，只有/bin，/dev，/etc，/lib，/proc，/var，/usr这些需要的，而其他都是可选的。

## 6.6.2 根文件系统的制作工具-- **Busybox**

根文件系统的制作就是生成包含上述各种目录和文件的文件系统的过程， 可以通过直接拷贝宿主主机上交叉编译器处的文件来制作根文件系统，但是这种方法制作的根文件系统一般过于庞大。也可以通过一些工具如**Busybox** 来制作根文件系统， 用**Busybox**制作的根文件系统可以做到短小精悍并且运行效率较高。

Busybox 的源码可以从官方网站[www.busybox.net](http://www.busybox.net)下载，然后解压源码包进行配置安装，操作如下：

```
#tar -xjvf busybox -1.24.1 .tar .bz2
```

```
#cd busybox -1 .24 .1
```

```
#make menuconfig
```

```
#make
```

```
#make install
```

### 6.6.3 YAFFS2 文件系统的创建

创建YAFFS2根文件系统的步骤如下：

(1) 创建根目录myrootfs，把busybox生成的三个文件复制到myrootfs目录下，并在此目录下分别建立dev, lib, mnt, etc, sys, proc, usr, home, tmp, var等目录(只有dev, lib, sys, usr, etc是不可或缺的，其他的目录可根据需要选择)。在etc目录下建立init.d目录。

(2) 建立系统配置文件inittab, fstab, rcS，其中inittab, fstab放在etc目录下，rcS放在/etc/init.d目录中。

(3) 创建必须的设备节点，该类文件必须在/dev目录下创建。

(4) 如果Busybox 采用动态链接的方式编译， 还需要把busybox 所需要的动态库:libcrypt.so.1， libc.so .6 ， ldlinux .so .2 放到lib 目录中。为了节约嵌入式设备的FLASH 空间， 通常会采用动态链接方式， 而不采用静态链接方式。如目前国内较有名气的厂商“友善之臂”官方提供了动态链接库的下载， 直接将库文件拷贝至 rootfs/lib 目录下即可：

```
# cp -a /tmp/Friendly ARM-lib/*.so.* ${ROOTFS}/lib
```

另外，在某些版本的linux系统中还需要为 bin/busybox 加上 SUID 和 SGID 特殊权限， 否则某些命令如passwd 等命令会出现权限问题：

```
# chmod 675 ${ROOTFS}/bin/busybox
```

(5) 改变rcS 属性为可执行。

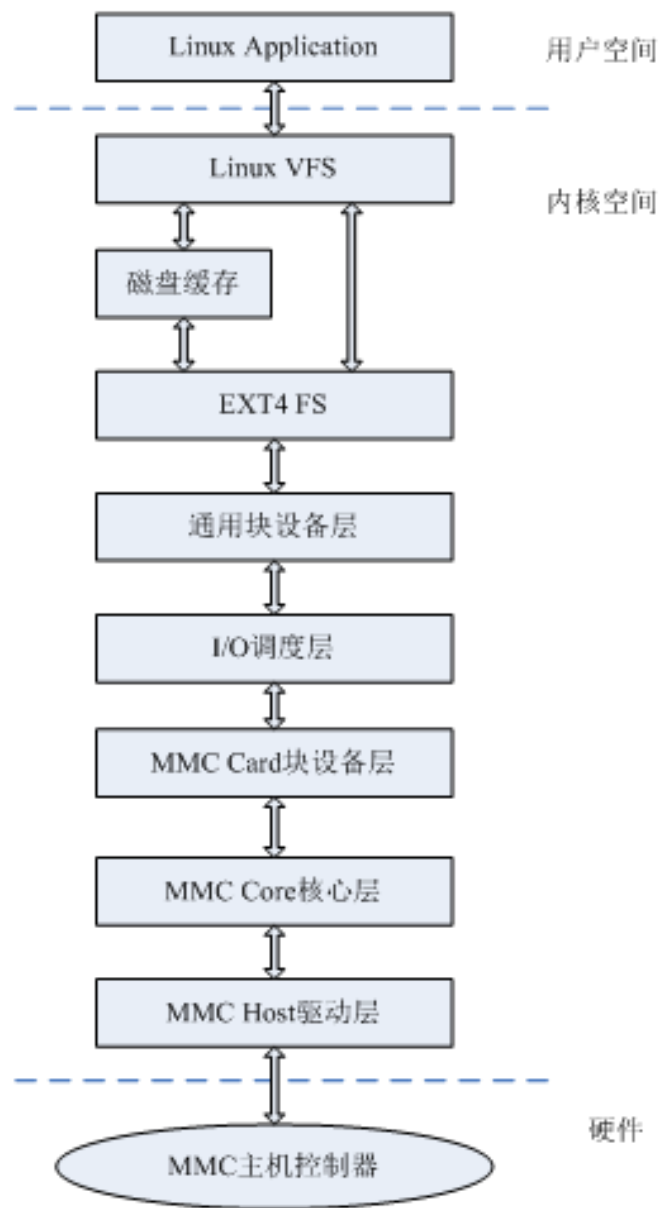
(6) 上面已经建立了根文件目录myrootfs， 然后使用mkyaffs2image-128M工具， 把目标文件系统目录制作成yaffs2格式的映像文件， 当它被烧写入Nand Flash中启动时， 整个根目录将会以yaffs2文件系统格式存在， 这里假定缺省的Linux内核已经支持该文件系统，

# EMMC

- **eMMC**，全称**embedded Multi Media Card**，即嵌入式多媒体存储卡。**MMC**协会所制定的内嵌式存储器标准规格，主要应用于智能手机和移动嵌入式产品。**eMMC**是一种嵌入式非易失性存储系统，由标准**MMC**接口、主控制器和**NAND Flash**三部分组成，采用**JEDEC**标准**BGA**封装。



- 在Linux系统中，eMMC仍然作为一个传统的块设备工作，上层可以搭载各种文件系统，Android智能手机早期采用yaffs2，目前使用ext4。曾经主流中端手机采用的都是eMMC5.1的闪存，其理论带宽为600M/s。顺序读取速度为250M/s，顺序写入速度为125M/s



# UFS

- UFS，全称Universal Flash Storage，即通用闪存存储，采用串行数据传输技术，只有两个数据通道(但速率高于eMMC)，工作模式为全双工模式，同一条通道允许同时进行读写传输，传输效率得到提高。

# UFS

- **UFS前期版本并未得到很大的关注，直到2013年9月发布的新一代闪存存储标准UFS 2.0才开始得到大众的追捧，UFS 2.0闪存读写速度可以达到5.8Gbps(约为750MB/s)，而UFS 2.1的数据读取速度更是高达11.6Gbps(约为1.5GB/s)，是UFS 2.0的两倍，**

# eMMC和UFS各项数据对比

	eMMC5.0/eMMC5.1	UFS2.0/UFS2.1
数据总线模式	并行	串行
理论接口宽度	eMMC5.0: 400MB/s eMMC5.1: 600MB/s	UFS2.0: 5.8GB/s (约 750MB/s) UFS2.1: 11.6GB/s (约 1.5GB/s)
工作模式	半双工	全双工
数据指令	eMMC 原生指令集	SCSI 指令集
数据通道数	8	2
多线程/数据队列/分区引导	不支持	支持
满载功耗	低	高
生产厂商	众多 (eMMC5.1)	三星、东芝、海力士、美光 (UFS2.1)
价格	约 145 RMB (64G eMMC5.1)	约 145 RMB (64G UFS5.1)