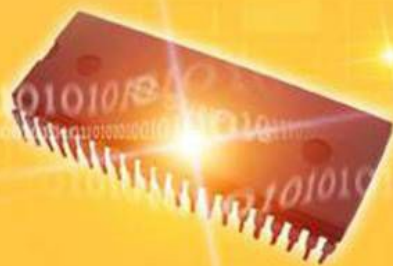


嵌入式系统工程师



嵌入式文件系统



- 文件系统简介
- Linux文件系统分类
- Linux根文件系统目录结构
- Linux启动流程
- 构建自己的根文件系统

- 文件系统简介
- Linux文件系统分类
- Linux根文件系统目录结构
- Linux启动流程
- 构建自己的根文件系统

- 在计算机系统中，需要用到大量的程序和数据，它们大部分以文件的形式存放在外存当中，根据需要可随时调入内存使用
- 如果用户直接管理外存文件所面临的问题：
 - 必须熟悉外存的物理特性
 - 了解各种存储文件的属性
 - 记录文件在外存上的存储位置
 - 在多用户环境下，必须能保证数据的安全性和一致性
- 显然，这是用户不能胜任也不愿意承担的工作

- 为了解决文件的管理问题，在操作系统中出现了
—文件系统
- 文件系统的功能：
 - 负责存储器中文件的组织和分配
 - 提高对存储器资源的利用效率
 - 对文件的存取、共享和保护等手段提供给操作系统和用户
 - 简化用户对文件的各项操作
 - 保证在多用户环境下文件的安全性和一致性
- 文件系统是操作系统的重要组成部分，是实现文件多元化管理的必要条件
- 文件系统的一般定义是：**负责管理和存储文件信息的软件组件。**

➤ 操作系统对磁盘的管理

- 无论是Windows还是Linux都将磁盘划分成指定大小的分区进行管理
- 在特定分区安装操作系统、系统启动所必需的文件，其它磁盘空间作为普通分区提供给用户使用
- 所有分区都可以根据用户需要，安装不同的文件系统对文件进行管理
- 一般操作系统都会支持多种常用文件系统格式，比如：FAT、NTFS、EXT4、YAFFS2等

- 文件系统简介
- Linux文件系统分类
- Linux根文件系统目录结构
- Linux启动流程
- 构建自己的根文件系统

- Linux支持多种文件系统，包括minix、ext、vfat、ntfs、iso9660、jffs、yaffs和nfs等
- 系统启动之后，便可以自动或手动挂载需要用到的文件系统
- 一般不同的存储设备会采用不同的文件系统类型，以便更好的组织和管理存储设备上的文件与目录
- 下面简单介绍几种常用的嵌入式文件系统：

➤ jffs文件系统

- 最早是由瑞典Axis Communications公司为嵌入式系统开发的日志型文件系统
 - 目前jffs已经大量被jffs2取代
 - jffs2是RedHat公司基于jffs开发的闪存文件系统
 - jffs2最初是针对RedHat公司eCos相关产品开发的嵌入式文件系统，所以jffs2也可以用在Linux、 μ CLinux等嵌入式OS中
- ## ➤ jffs主要用于nor型闪存，基于MTD驱动层，特点是：
- 可读写的、支持数据压缩的、日志型文件系统基于哈希表，并提供了崩溃/掉电安全保护，提供“写平衡”支持等。
 - 缺点：当文件系统已满或接近满时，因为垃圾收集的关系而使jffs2的运行速度大大放慢。

➤ yaffs文件系统

- yaffs/yaffs2是专为嵌入式系统使用NAND型闪存而设计的一种日志型文件系统
- 与jffs2相比，它减少了一些功能(例如不支持数据压缩)，所以速度更快，挂载时间更短，对内存的占用较小
- yaffs是跨平台的文件系统，除了Linux和eCos，还支持WinCE, pSOS和ThreadX等

➤ 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。
- 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文件系统

- 由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
- cramfs文件系统以压缩方式存储，在运行时解压缩，不支持应用程序以XIP方式运行，所有的应用程序要求被拷到ram里去运行
- cramfs经常跟yaffs文件系统配合使用，以达到修改必须内容的目的

➤ nfs文件系统

- nfs是由Sun公司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。
- 我们最开始即采用此种方式来开发和调试根文件系统。

- 使用nfs作为根文件系统的步骤:
- 查看u-boot中启动命令行参数:

```
sunplusEDU-210 #printenv
```

```
bootargs=root=/dev/mtdblock4 rootfstype=yaffs2  
init=/linuxrc console=ttySAC0,115200
```

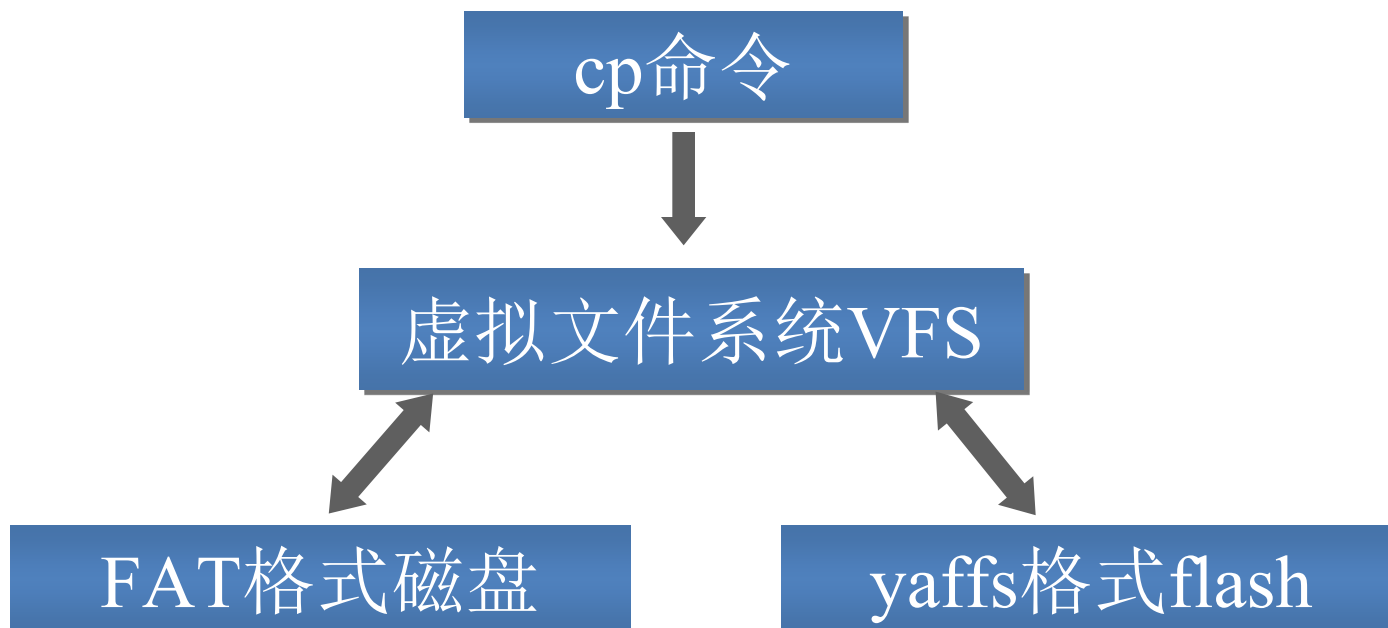
- 设置u-boot从nfs启动根文件系统:
- sunplusEDU-210 # setenv bootargs noinitrd root=/dev/nfs
nfsroot=/home/edu/rootfs/,rsize=1024,wsiz=1024
ip=172.20.223.123:172.20.223.24:172.20.223.254:255.255.
255.0::eth0:off init=/linuxrc console=ttySAC0,115200 (代
表从172.20.223.24机子上/home/edu/rootfs/启动根文件系统)

➤ 虚拟文件系统

- 为了对各类文件系统进行统一管理，Linux引入了虚拟文件系统——VFS (Virtual File System)，为各类文件系统提供统一的操作界面和应用编程接口
- VFS并不是一种实际的文件系统，它是物理文件系统与服务之间的一个接口层
- VFS只存在于内存中，不存在于任何外存空间
- VFS在系统启动时建立，在系统关闭时消亡

➤ 虚拟文件系统的优势

- 通过虚拟文件系统，我们可以利用标准的Linux系统调用，对不同介质上的不同文件系统进行读写操作，例如：



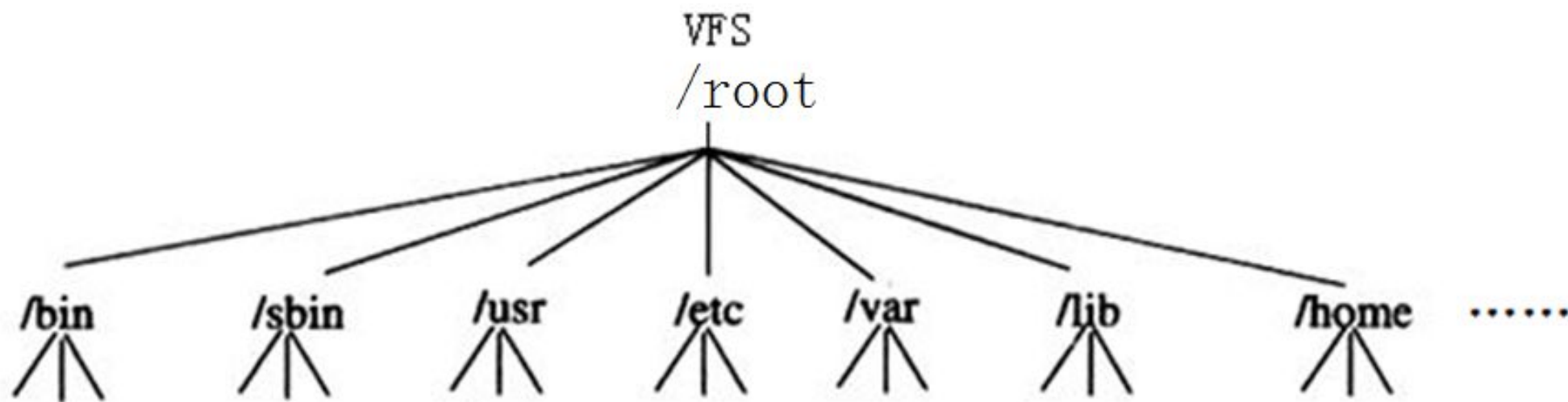
- 文件系统简介
- Linux文件系统分类
- Linux根文件系统目录结构
- Linux启动流程
- 构建自己的根文件系统

➤ 根文件系统特点:

- Linux系统启动后, 首先挂载的文件系统称为根文件系统, 其中包括系统启动所必需的文件
- 在根文件系统的最顶层目录中, 每个目录都有其具体的目的和用途
- 根文件系统的类型没有限制, 可以是Linux系统所支持的任意稳定可靠的文件系统类型
- 在Linux下所有磁盘被挂载后才能使用, 这个挂载的目录被称为**挂载点**或**安装点**, 然后通过此目录来访问这个分区上的文件或其它目录

➤ Linux目录结构

- Linux和unix都是一个以“/”作为根的树状文件结构
- “/”因此被称为根目录，其它所有的文件和目录都置于根目录之下



- 在早期的unix系统中，各个厂家都定义了自己的unix文件系统结构，且互不兼容，比较混乱
- 1994年推出FHS (Filesystem Hierarchy Standard)，对根文件系统目录做了统一规范，形成了FHS标准
- Linux根文件系统标准都包括那些目录呢！

目录名	内容
bin	提供基本的用户命令
boot	用于Bootloader的表态文件
dev	设备或其它的特殊文件
etc	系统配置文件, 包括启动文件
home	多个用户的主目录
lib	基本的系统库, 如C库、内核模块等
mnt	用于临时挂载的文件系统
opt	可选择的软件包
proc	内核虚拟文件系统和进程信息
root	根用户的主目录
sbin	基本的系统管理二进制库
tmp	临时文件
usr	其中的二级目录里包含许多应用程序和文档
var	一些变化的实例和工具等

FHS定义的根文件系统顶层目录

- 一个最基本的根文件系统由以上FHS指定的目录构成
- /bin: 提供基本用户命令, 如ls、cp等
- /sbin: 存放重要的系统命令, 如ifconfig、rmmod等
- /dev: 存放设备文件, 如mtdblock0、fb0等
- /etc : 存放系统启动所需要的各种配置文件:
 - 开机自动挂载的文件系统和自动运行的脚本
 - 终端配置文件profile
 - 用户登录信息配置文件
 - 各种开源工程配置文件, 如boa、TFTP等
- /mnt: 临时文件系统挂载目录, 比如挂载U盘、SD卡等

- /lib: 存放应用程序所需的基本共享库，如C、C++等标准库，提供给所有运行时需要共享库的应用程序，比如网络、GTK、QT等应用程序
- /usr: 存放共享的程序或库，一般由用户自由添加，比如各种不常用的命令和后期加入的开源库等
- /tmp: 用于存放临时文件，通常为空，比如一些需要生成临时文件的程序需要此目录，且必须保证此目录可写，一般挂载ramfs文件系统
- /var: 存放可变的数据，如一些系统日志文件等
- /home: 普通用户的家目录
- /proc: 一般挂载proc文件系统，用来表示系统的运行状态，如：模块加载数、中断申请与使用状态、进程运行状况等

➤ 总结:

- 制作根文件系统的过程，其实就是构造以上目录及系统启动所必需文件的过程
- 构建嵌入式根文件系统的工作，也就从构建这几个文件夹开始

- 文件系统简介
- Linux文件系统分类
- Linux根文件系统目录结构
- Linux系统启动流程
- 构建自己的根文件系统

➤ 回顾Bootloader引导内核的过程？

- 其中一步是Bootloader将系统控制权交给内核之前，会给内核传递参数，具体内容如下：

`root=/dev/mtdblock4 init=/linuxrc console=ttySAC0,115200`

➤ 各参数含义：

- root: 指定根文件系统在FLASH分区中的位置
- init: 指定Linux内核启动完毕后调用的第一个用户态程序
- console: 指定内核启动后首选的控制台

➤ linuxrc程序都做了哪些事情？它是怎么来的？

- linuxrc是由内核启动的第一个，也是惟一的一个用户进程，PID为1
- linuxrc通常和/bin/init一样，除非用户重新实现它
- linuxrc执行流程：
 - 设置SIGSEGV、SIGILL、SIGFPE、SIGBUS信号处理函数
 - 初始化控制台，设置环境变量（HOME、SHELL、USER等）
 - 解析/etc/inittab配置文件，否则将运行默认配置
 - 监听特定子进程状态
- 总之，linuxrc是后续进程的发起者
- /etc/inittab配置文件的作用？

- /etc/inittab决定了接下来将要启动的脚本、shell和应用程序
- /etc/inittab配置文件结构
 - 每一行都指定一个子进程，并确定了进程运行方式
 - 用冒号来分隔各字段的属性，比如：

```
$ vim etc/inittab  
1  ::sysinit:/etc/init.d/rcS  
2  ::respawn:-/bin/sh  
3  #tty0::askfirst:-/bin/sh  
4  ::restart:/home/a.sh  
5  ::ctrlaltdel:/bin/umount -a -r
```

➤ 具体格式:

<id> : <runlevels> : <action> : <process>

- **id:** 表示这个子进程使用的控制台，如果省略，则使用与linuxrc进程一样的控制台
- **runlevels:** 该字段主要用于PC机，对于嵌入式系统暂时没有处理，可以省略
- **action:** 表示linuxrc进程将如何控制这个子进程，具体取值见后面的表格
- **process:** 表示要启动的可执行程序或脚本，如果process字段前面有“-”字符，说明这个应用支持“交互”

Linux系统启动流程

名称	执行条件	说明
sysinit	系统启动后最先执行	指定初始化脚本路径，只执行一次，init进程等待它结束才继续执行其它动作
wait	执行完sysinit进程后	只执行一次，init进程等待它结束才继续执行其它动作
once	执行完wait进程后	只执行一次，init进程不等待它结束
respawn	启动完once进程后	init进程监测发现子进程退出时，重新启动它
askfirst	启动完respawn进程后	与respawn类似，不过init进程先输出“Please press Enter to activate this console”，等用户输入回车后才启动子进程。
shutdown	当系统关机时	即重启、关闭系统时执行的程序
restart	init进程接收到SIGHUP信号时	init进程重启时执行的程序，通常是init程序本身。先重新读取、解析/etc/inittab文件，再执行restart程序
ctrlaltdel	按下Ctrl+Alt+Del	按Ctrl+Alt+Del组合键时执行的程序

- linuxrc解析inittab配置文件过程中，被指定的用户程序解析执行顺序如下：

**sysinit→wait→once→respawn→askfirst→
ctrlaltdel→shutdown→restart**

- 程序执行顺序与配置文件中的书写顺序无关
- 执行完所有开机需要加载的进程或脚本后，开始监听子进程的运行状态
- 如果respawn、askfirst子进程退出，则重启该进程

- 文件系统简介
- Linux文件系统分类
- Linux根文件系统目录结构
- Linux系统启动流程
- 构建自己的根文件系统

- 我们从构建/bin、/sbin、/usr、linuxrc开始
 - 从网上下载所有常用命令的源码
 - 采用交叉编译器，重新编译好命令下载到开发板/bin、/sbin等目录下
- 问题：
 - 命令体积和数量都会很大，不适合嵌入式系统对资源的有效利用
- 嵌入式领域将采取专用工具来制作常用命令和相关应用

➤ 嵌入式Linux系统的瑞士军刀

— Busybox

- Busybox项目由Bruce Perens在1996年发起，目的是在一个软盘上创建一个可引导GNU/Linux的系统
- 从1999年开始，Busybox项目由uClibc的维护者Erik Andersen接手维护
- Busybox项目发展迅速，它是许多嵌入式Linux系统的基石之一，并且拥有非常活跃的用户群
<http://www.busybox.net>

➤ Busybox特点:

- 提供完善的Linux命令工具集
- 提供图形化的配置环境和默认配置选项
- 所有功能均整合到busybox程序中，实现不同命令的代码共享，占用磁盘空间极小
- 所有命令均通过软链接到/bin/busybox实现
- 帮助用户实现了1号用户进程(linuxrc)

➤ 任务：

➤ 参考如下文档完成根文件系统制作

《1、根文件系统的制作(busybox)》

《2、根文件系统验证及镜像制作》



凌阳教育官方微信：Sunplusedu

Tel: 400-705-9680, BBS: www.51develop.net, QQ群: 241275518

