

文章编号:1005-8451(2005)02-0011-03

## uClinux下多文件系统的研究与实现

肖建<sup>1</sup>, 张宇<sup>2</sup>

1. 南京邮电学院 电子工程系, 南京 210003; 2. 东南大学 电子工程系, 南京 210096

**摘要:**通过研究和比较在嵌入式系统中常用的存储设备与对应的文件系统,提出了以多文件系统作为uClinux的文件系统构架,同时给出自主的一款SOC为目标系统来实现多文件系统的uClinux构建。

**关键词:**uClinux; 嵌入式系统; 多文件系统; 存储设备

**中图分类号:** TP39

**文献标识码:** A

### Study and implementation on Multifile System based on uClinux

XIAO Jian<sup>1</sup>, ZHANG Yu<sup>2</sup>

(1. SEU EE Department, Nanjing University of Post & Telecommunication, Nanjing 210003, China;

2. Department of Electronic Engineering of Southeast University, Nanjing 210096, China)

**Abstract:** It was produced Multifile System as architecture of uClinux File system, by researched and compared used memory devices and corresponding file system. Meanwhile, a case of SOC was given to demonstrate the Implementation of Multifile Systems based on uClinux.

**Key words:** uClinux; Embedded Target System; Multifile Systems; memory device

Linux 是一种很受欢迎的操作系统,它与 UNIX 系统兼容,并且是开放性源代码。近年来已逐渐地应用于嵌入式设备。因为没有 MMU 单元的 CPU 相对而言更为便宜,所以不带 MMU 功能的 uClinux 应运而生。目前 uClinux 可以稳定地运行在 ARM, MIPS 和 Powerepc 等系列 CPU 上。作为此类的嵌入式系统一般采用 Flash 作为存储设备, RAM 作为操作系统运行环境。本文主要研究 uClinux 系统在运行时使用 Flash 和 RAM 相对应的多文件系统。

### 1 存储设备与对应的文件系统

在过去 20 年里,嵌入式系统一直使用 ROM (EPROM) 作为它们的存储设备。近年来,Flash 全面代替了 ROM (EPROM) 在嵌入式系统中的地位。因为相比较 ROM 而言,Flash 有成本低、可靠和容易改写等优点。在 uClinux 中提供了 MTD (memory technology device 内存技术设备) 作为解决方案。MTD 是用于访问 memory 设备 (ROM、flash) 的 Linux 的子系统。MTD 的主要目的是为了使用新的存储设备的驱动更加简单,为此它在硬件和上层之间提供了一个抽象的接口。MTD 的所有源代码在 /drivers/mtd

子目录下。我们将 CFI 接口的 MTD 设备分为 4 层。从设备节点直到底层硬件驱动,这 4 层从上到下依次是:设备节点、MTD 设备层、MTD 原始设备层和硬件驱动层。有了 MTD,文件系统对存储设备的读和写就变成了对 MTD 的简单操作。

在 uClinux 中,建立在 Flash 上的文件系统通常有 Ext2、Ext3、romfs 和 jffs2 等。Ext2 文件系统具备稳定性、可靠性和健壮性等优点,然而,用于嵌入式设备中,因为 Ext2 文件系统没有提供对基于扇区的擦除/写操作的良好管理;在出现电源故障时,Ext2 文件系统不是防崩溃的,也不支持损耗平衡。由于这些原因,Ext2 通常并不用于嵌入式系统中。Ext3 是在 Ext2 的基础上发展起来的,它向下兼容 Ext2,同时提供异步的日志。romfs 是 uClinux 的默认根文件系统,这种文件系统相对于一般的 Ext2 文件系统要求更少的空间。空间的节约来自于两个方面,首先内核支持 romfs 文件系统比支持 Ext2 文件系统需要更少的代码,其次 romfs 文件系统相对简单,在建立文件系统超级块 (superblock) 需要更少的存储空间。但是,romfs 文件系统并不支持动态擦写保存。jffs2 文件系统是瑞典的 Axis 通信公司开发的一种基于 Flash 的日志文件系统。所谓日志文件系统,其工作原理是跟踪记录文件系统的变化,并将变化内容记录入日志,并保存在磁盘分区。书写

收稿日期:2004-06-06

作者简介:肖建,在读硕士研究生;张宇,在读硕士研究生。

操作首先是对记录文件进行操作,若整个写操作由于某种原因(如系统掉电)而中断,系统重启时,会根据日志记录来恢复中断前的写操作。jffs2这种日志文件系统的特性在嵌入式设备中越来越受欢迎。特别是jffs2是专门为象闪存芯片那样的嵌入式设备创建的,所以它的整个设计提供了更好的闪存管理。然而它也有缺点:当文件系统已满或接近满时,jffs2会放慢运行速度。这就是垃圾收集的问题。

在嵌入式系统中,RAM则常常成为操作系统的运行环境。常用的RAM有SDRAM,SRAM和memory-on-chip。有时当嵌入式系统没有存储设备但又需要文件系统时,内存RAM便成为文件的存储介质,用它来代替磁盘,这就是“Ramdisk”。如果需要的话,Ramdisk与存储介质也可以同时存在。在它上面所建立的文件系统通常是Ext2文件系统,以实现动态数据保存。

## 2 存储设备与多文件系统的选取

要建立uClinux下的根文件系统,通常有两种方式:1)用uClinux默认的romfs作为根文件系统,然后再做一个可读写的文件系统,作为普通文件在系统启动以后被挂载来作为数据库的存储;2)将根文件系统直接做成可读写的文件系统。

在建立可读写的文件系统时,通常选用jffs2文件系统。这是因为充分考虑了Flash的各种技术特性和操作限制,jffs2文件系统能够高效地直接对Flash芯片进行操作,同时充分考虑了非正常断电对文件系统的破坏,使文件崩溃后能够迅速地恢复。

因此,对于操作系统而言,其文件系统只有1个,即根文件系统;文件系统只有2个,即根文件系统加一个普通文件系统;文件系统只有3个,即根文件系统加一个普通文件系统和一个建立在Ramdisk的文件系统。之所以要加Ramdisk是因为有些系统可能在数据库处理时需要用到,好处就是可以得到高速度,简单而明快。对于那些只读的数据库来说,这可是一个非常实用的方案。

## 3 多文件系统的实现实例

在开发的嵌入式系统中,Garfield目标开发板拥有32 M SDRAM和16 M NORFLASH。其操作系统是

uClinux操作系统,Kernel版本是2.4.20。在顶层采用MINIGUI作为GUI。选择了3个文件系统:1个根文件系统romfs;2个普通文件系统:jffs2和ext2。

选择romfs作为根文件系统,因为romfs本身是uClinux的默认根文件系统,且相对简单,易于开发和维护;所需的存储空间也比其它文件系统要少;同时因为嵌入式系统中的大部分代码通常是固定不变,而romfs又是只读文件系统,还可防止根文件系统被意外修改。所以,选择romfs作为根文件系统。同时,在该系统运用中,需要存储一定的用户数据,这就需要有一个可存储的文件系统。选择jffs2文件系统不仅是因为它是日志文件系统,可以实现储存和断电保护功能,更重要的是Garfield目标开发拥有Flash,而jffs2文件系统恰好是为Flash专门设计的。在文件系统的实现中,将jffs2文件系统作为普通文件系统在系统启动以后挂载到romfs根文件系统上。之所以选择Ext2文件系统是因为在系统运行时需要用到Ramdisk来处理临时数据库。对于Ramdisk来说最好的文件系统实现就是Ext2。因为Ramdisk不需要最终存储数据,它是建立在内存中起缓存作用,也不需要支持损耗平衡,只需要满足高速度和简单的要求,而Ext2是标准Linux的文件系统,很多操作非常方便;所以Ext2文件系统就成为Ramdisk的首选。

图1所示是系统NORFLASH的布局。

Boot loader (只读)
Kernel (只读)
Romfs.img (只读)
Jffs2.image (只读)

图1 系统NORFLASH的布局

描述Flash硬件设备特征的代码位于uClinux-dist/linux-2.4.x/drivers/mtd/maps目录下。可以根据具体的Flash和挂载方式选择合适的文件加以修改,以适应开发的要求。对于Garfield目标开发的16M INTEL Flash而言,建立了自己的分区函数gfd-flash.c。

将其分为4个分区,分别为:Boot loader, kernel, romfs, img, jffs2。

同时还要定义Flash的起始地址和数据总线宽度:  
#define FLASH\_PHYS\_ADDR 0x20000000

```
#define INTEL_BUSWIDTH      4
```

在设备驱动注册时还需要将Flash设备注册上,同时将其对应的分区表加入mtd-table。这个工作可以在初始化的时候完成。如:

```
int __init init_gfd(void){
    mymtd=do_map_probe("cfi_probe", &gfd_intel_map);
    if (mymtd){
        mymtd->module = THIS_MODULE;
        add_mtd_partitions( mymtd, gfd_intel_partitions,4);
        return 0;
    }
}
```

该函数实现:

1) 调用do\_map\_probe()搜索MTD设备并将其赋给mymtd;

2) 调用add\_mtd\_partitions()将your\_partition的各个分区加入mtd\_table。

因为根文件系统在第三个分区,即mtd2。所以在linux-2.4.x/init/do\_mount.c文件中需要进行修改,也就是把将要加载到内核上的根文件系统去指向romfs文件系统:ROOT\_DEV = MKDEV(FLASH MAJOR, FLASH MINOR);

其中,FLASH MAJOR是NORFLASH的主设备号,FLASH MINOR是次设备号,在分区中是romfs位于次设备号2。

一旦分区建立,就是实现分区上的实际内容了:

#### 1. Bootloader

对于基于ARM的Garfield SOC开发系统而言,系统上电后PC将取0地址的代码执行。所以boot-loader将位于Flash的0地址。一个简单的boot-loader一般主要完成运行环境的建立,如对内存系统的初始化,CPU参数的配置。如果操作系统是运行在内存中,则还需要实现代码拷贝功能。

#### 2. Kernel

Kernel是在linux-2.4.x下通过移植编译后生成的。其中在配置block devices的时候要把ramdisk support选上,并指定大小以生成Ramdisk。

#### 3. romfs.img

romfs.img则是在uClinux-dist目录下编译生成的。它在uClinux默认状态下产生。

#### 4. jffs2

生成jffs2文件系统通常有两种方式:一是对所要生成的块设备直接做成jffs2文件系统,如mkfs.jffs2 /dev/mtdblk3,然后在挂接到所要挂接的目录夹上如挂接到mnt上:mount -t jffs2 /dev/mtdblk3 /mnt。另外一种方式是采用文件镜像的方法,即生成一个jffs2.image,然后再挂接到目录夹上如:mkfs.jffs2 -e 0x40000 -o jffs.image,其中-e选项确定闪存的擦除扇区大小通常是64 kbit, -o选项用于输出文件,通常是jffs2文件系统映像。在本例中是jffs.image。

最终生成的操作系统目录结构如图2。

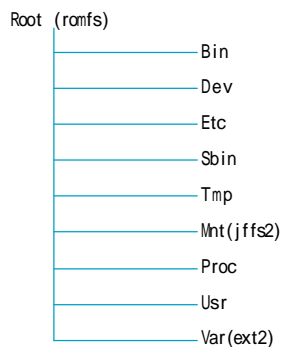


图2 操作系统目录结构

## 4 结束语

本文主要研究了uClinux下多文件系统的构成,比较和分析了它们之间建立在不同存储设备上和运用在不同场合下的优缺点,提出了在嵌入式uClinux系统中文件系统选取的常见方法以供参考。最后以Garfield目标开发板为例,给出了在NORFLASH上的分布和多文件系统的建立,并说明了这种多文件系统的优点所在。然而,这种以romfs为根文件系统的多文件系统构架也存在一个问题,就是需要内存的支持。但它运用于以NANDFLASH为主要存储设备的系统上来则可以说是一个比较好的选择。

#### 参考文献:

- [1] 来晓阳,都思丹,高敦堂. Linux下系统信息搜集软件的实现[J]. 计算机应用,2002,22(8):44-46.
- [2] 周余,都思丹,王自强,等. 应用处理器PXA255上的嵌入式Linux系统开发[J]. 计算机应用,2004,24(3):159-160.
- [3] 张翔,刘鹏,戴国骏. 嵌入式Linux闪存文件系统-JFFS2的研究[J]. 杭州电子工业学院学报,2003,23(3):62-65.