基于 uC/FS 的大容量微存储 FAT32 格式的实现与应用

郝伟,李敬兆
安徽理工大学,安徽 淮南 232001)

摘要:进入后 PC 时代以后,嵌入式操作系统成为目前已经成为了最热门的技术之一,在嵌入式系统的应用中,微型大容量存储器的文件管理系统的重要性越来越突出,其中 uC/FS 以源码开放,价格低廉,兼容性好,效率高等特点受到越来越高的重视。

关键词:嵌入式;uC/FS;应用;开发
中图分类号:TP333　　文献标识码:A　　文章编号:1009-3044(2006)32-0113-01

Application and Development of File Management System Based on uC/FS-Ⅱ

HAO Wei,LI Jing-zhao
(Anhui University of Science and Technology, Huainan 232001, China)

Abstract:After the PC post, embedded system is now one of the hottest technologies. In any application of the embedded system, file management system for micro massive storage is of most importance. uC/OS, The Real-Time Kernel, is a highly portable, ROMable, scalable, preemptive real-time, multitasking kernel (RTOS) for microprocessors and microcontrollers, and of most conveniences.

Key words:embedded system; uC/FS; application; development

1 引言

在现在日益信息化的社会中,计算机和网络已经全面渗透到日常生活的每一个角落。对于我们每个人,需要的已经不再仅仅是那种放在桌上处理文档,进行工作管理和生产控制的计算机"机器";各种各样的新型嵌入式系统设备在应用数量上已经远远超过通用计算机,任何一个普通人可能拥有从大到小的各种使用嵌入式技术的电子产品,小到 MP3,PDA 等微型数字化产品,大到网络家电,智能家电,车载电子设备。而在工业和服务领域中,使用嵌入式技术的数字机床,智能工具,工业机器人,服务机器人也将逐渐改变传统的工业和服务方式。当我们满怀憧憬与希望跨入二十一世纪大门的时候,计算机技术也已经进入一个新的时代——后 PC 技术的时代。

另一方面,计算机的存储技术的发展,尤其是微型存储器的技术发展上出现了欣欣向荣,百家争鸣的局面,MMC,RS-MMC,SD,CF,Mini-SD, FLASH 闪存等等种量不断增加,拒不完全统计,市面上常见的微型存储器就有十几个种类上百种产品之多,在容量上也不断向更大发展,近日,日立又创新高发布了高达 6G 容量的 1 英寸微型硬盘,并且还宣布研发一款体积比现有产品减小 20%,容量高达 8-10GB 的 1 英寸 Microdrive 硬盘。同时在微存储器的速度上也越来越快,甚至可以和 PC 机上的硬盘相提并论。因此对文件管理系统提出了越来越高的兼容性好,管理效率高和速度快等要求。

当前的微操作系统中,常见的有 VxWorks,Palm OS, uCLinux, WinCE,uC/OS 等等。其中 uC/OS 以源码开放,结构简单,安全性高 通过美国国家航天局的认证

2 uC/FS 简介

uC/FS 是使用标准 ANSI C 编写的一种可应用于任何存储介质的文件管理系统,主要为其提供基本的硬件访问功能,它具有以下特点:

支持 DOS/WINDOWS 环境下的 FAT12,FAT16 和 FAT32;
多个存储器共同工作支持,可以同时访问多个存储器
多操作系统支持,可以很方便得移植到几乎任何操作系统
另外还有速度快,可扩展性强,可裁剪,支持广泛等特点

基于以上的特点,uC/FS 非常合适应作用于中小型的微存储器。

3 UC/FS 系统结构

uC/FS 采取分层工作方式,每一层负责不同的功能,由高层的数据抽象到底层的硬件实体分工明确,实现简单系统模型共分五层,见图 1。

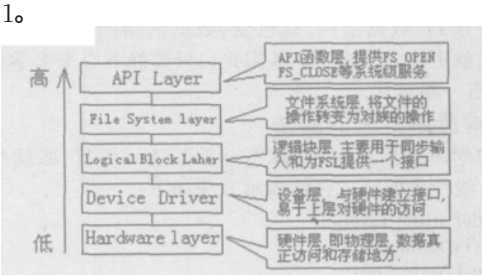

图 1

3.1 API (Application Programme Interface) 层

此层是应用程序和 uC/FS 的接口,它包涵了如文件打开(FS_FOpen),文件读写(FS_FWrite, FS_FRead)及相关函数的标准的 ANSI C 的函数库。API 层就是将这些对文件的调用转换到文件系统层。目前,uC/FS 已经提供了对 FAT,FAT32,NTFS 等众多分区格式的支持,并且可以同时对多分区格式同时进行操作,比如:使用 uC/FS 同时管理 FAT 和 NTFS 格式。

3.2 文件系统层

文件系统层将对文件的操作转化对对逻辑块的操作。当完成了些层的转换以后,uC/FS 就会调用相在的逻辑块并为器件调用相应的驱动。

3.3 逻辑块层

逻辑块层的主要任务是同步对一个器件的存取和为文件系统层提供一个简单接口。此层直接调用器件驱动以实现块操作。

3.4 器件驱动

器件驱动层是较底层的服务用于直接访问你的硬件。该层的结构比较简单,易于实现和硬件的整合。

3.5 硬件层

真正的硬件。此层服务只是简单读和写定长的簇。而且为了硬件系统和上层集成的简易性,此层的结构也是尽可能得简单。

4 基于 uC/OS-Ⅱ和 uC/FS 的 FAT32 文件管理系统

首先,本文建立在对 uC/OS-Ⅱ系统的移植已经完成 具体可以参阅相关文章)的基础上。

4.1 主文件目录

---- FS\API FS 系统的 API 层源文件

(下转第 186 页)

收稿日期:2006-08-10

各个阶段工作的多样性以及各种层次人员之间工作的配合关系等因素,使得开发的每一个环节都可能产生错误。所以软件测试不仅仅是软件开发的一个独立阶段,而应当把它贯穿到软件开发的各种阶段中,坚持各个阶段的技术评审,才能尽早发现和预防错误。一定要尽早地、不断地进行软件测试,这类似于对硬件的自检,软件错误发现得越早,为改正它而需付出的代价也就越小。图2所示的软件测试的W模型形象地说明了软件测试与开发的同步性。

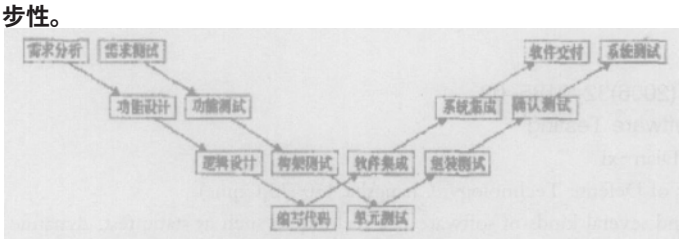

图2 软件测试的W模型

5.3 合理设计测试用例,适当选择测试方法

完整的测试用例不但需要测试的输入数据,而且需要对应这些输入数据的预期输出结果。在设计测试用例时,应当包括合理的输入条件和不合理的输入条件。合理的输入条件是指能验证软件的输入条件;不合理的输入条件则是指异常的、临界的、可能引起问题异变的条件。用不合理的输入条件测试软件能核实软件的容错能力和完全性,往往比合理的输入条件能发现更多的错误。测试方法中还要注意合理选择黑盒测试(功能测试)和白盒测试(结构测试)方法,一般采用黑盒为主、白盒为辅的综合测试方案。适当选用自动测试工具,如测试数据生成程序、动态分析程序、静态分析程序和文件比较程序等,同时还需重视人工测试方法,如代码审查。总之,合理选择测试方法将有利于增强测试效果、提高测试效益。

5.4 妥善保留全部测试用例

软件测试开发过程中,一定要做好测试用例的保存工作,这样在测试人员发生变动或者开展回归测试时会减少许多工作。我们在在程序改良或者Bug改正后需要重新测试时,就避免大量的枯燥乏味的重复工作,从而在提高测试效果的同时也相应的节省了软件开发成本。

5.4 程序员应避免检查自己的程序

测试工作需要严格的作风,客观的态度和冷静的情绪,由别人来测试程序员编写的程序,可能会更客观,更有效,并更容易发现问题。这是因为:一方面,程序中可能包含一些由于对问题的叙述和说明误解而导致的错误,编写程序本人一般很难发现;另一方面,就心理上来讲,程序员总不希望自己的程序出错,因此不能有效测试自己的软件。同时要注意的是,这点不能与程序的调试相混淆。调试由程序员自己来做可能更有效。

6 结束语

软件测试是保证软件可靠性的主要手段,是软件开发过程中比较艰巨和繁重的任务。软件开发人员要明确软件测试的目标,掌握软件测试方法、策略,选用最少量的高效测试数据,做到尽可能完善的测试,从而尽可能多地发现软件中的问题。这样才能最大程度的降低软件测试的成本,提高软件测试效率,开发出用户满意的高质量软件。

参考文献:

[1]张海藩. 软件工程导论[M]. 北京:清华大学出版社,1992.

[2]郑人杰. 软件测试技术[M]. 北京:清华出版社,1992.

[3]齐治昌, 谭庆平, 宁洪. 软件工程[M]. 高等教育出版社, 2004.

(上接第113页)

---- FS\FSL 系统的FSL层源文件
---- FS\LBL 系统的LBL层源文件
---- FS\DEVICE\NAND 系统的DEVICE层支持U盘的源文件
---- Applicaions 主要应用程度源文件
---- uCOS 存放uC/OS系统主要头文件和系统文件

4.2 cCFS\FS_Conf.h系统设置(更多参见FS_Conf.h源文件)如下:

FS_FAT_SUPPORT_FORMAT 设置成1表示支持FAT格式
FS_FAT_DISKINFO 设置1表示显示硬盘信息
FS_FAT_FWRITE_UPDATE_DIR 设置成1
FS_FAT_SUPPORT_FAT32 设置成1
FS_FAT_SUUUPOR_LFN 设置成1

4.3 操作系统设置

针对uC/OS-II只要设置一个宏:

FS_OS_UCOS_II 设置为1即可实现对uC/OS-II的支持

4.4 数据格式设置

数据类型设置是指根据需要的数据类型设置系统可以支持的设置,如下:

FS_SIZE_T 32位无符号数

以下是基于上述系统的应用实例,其实配置参见上文,以下是主函数的详细设置。

系统主文件 Application\main.c

```
/* 主应用程序 */
#include "includes.h"
#include "fs_api.h"
/* 系统设置 */
#define TASK_STK_SIZE 512
/* Size of each task' s stacks (# of WORDs) */
#define N_TASKS 10 /* Number of identical tasks */
/* 变量设置 */
OS_STK TaskStk[N_TASKS][TASK_STK_SIZE];
/* 系统任务栈 */
OS_STK TaskStartStk[TASK_STK_SIZE];
char TaskData[N_TASKS]; /* 每个任务栈参数 */
OS_EVENT *RandomSem;
/* 函数类型声明 */
void Task(void *data); /* 系统任务函数 */
void TaskStart(void *data); /* 系统任务函数启动 */
static void TaskStartCreateTasks(void);
static void TaskStartDispInit(void);
static void TaskStartDisp(void);
void MainTask(void);
/* 主任务,即应用程序需要处理的任务 */
{
FS_Init(); /* 文件系统初始化 */
DoSomething();
/* 所有要做的放在这里,如磁盘检索程序等 */
FS_Exit(); /* 文件系统结束退出 */
}
void main (void)
{
PC_DispClrScr(DISP_FGND_WHITE + DISP_BGND_BLACK);
/* 清屏 */
OSInit(); /* 初始化uC/OS-II */
PC_VectSet(uCOS, OSCtxSw);
/* 加载uC/OS-II任务指针向量 */
OSStart();
MainTask(); /* Start multitasking */
}
```

5 结束语

使用uC/OS-II+uC/FS的微型大存储器的文件系统具有生产成本低,使用速度快,开发简单方便等特点,而且由于是同一个公司出口的系统,所以具有很强的兼容性,而且移植方便,扩展性,非常合适中小型企业开发MP3,U盘使用。

参考文献:

[1]田泽.嵌入式系统开发与应用[M].北京航空航天大学出版社,2005.1.

[2]Jean J.Labrosse,邵贝贝,译.嵌入式实时操作系统uC/OS-II(第2版)[M],2000.6.