

• 嵌入式系统工程 •

改进的嵌入式软件架构及其应用层开发模式

万相奎¹, 陈建明²

(1. 广东工业大学 信息工程学院, 广东 广州 510006; 2. 托肯恒山科技(广州)有限公司, 广东 广州 510663)

摘要:为了解决嵌入式软件并行开发和调试困难问题,在软件体系结构上采用改进的层模式架构,提高嵌入式代码复用度。在应用层开发上研发任务间通讯服务(ICS)消息总线和邮箱服务,承担模块化设计的各功能进程的通讯控制和数据交互。把相对独立的应用模块进程化,通过增加或删减进程来定制嵌入式设备软件功能,保持其扩展性和低耦合高内聚特征。通过实际的项目开发验证,方便地实现了嵌入式应用层功能模块的并行设计和调试,并使得嵌入式系统平台上的进程间通讯可靠、灵活、易维护。

关键词:嵌入式系统;层模式;任务间通讯服务;邮箱;定制

中图分类号:TP311 文献标识码:A 文章编号:1000-7024(2009)23-5358-03

Improved software architecture for embedded system and its development pattern for application layer

WAN Xiang-kui¹, CHEN Jian-ming²

(1. Information Engineering College, Guangdong University of Technology, Guangzhou 510006, China;

2. Tokheim Hengshan Technologies (Guangzhou) Limited Company, Guangzhou 510663, China)

Abstract: Considering the parallel developing difficulty for embedded software, a improved layer pattern is introduced and used in software architecture, which boosts the code reuse. The ICS message bus and mailbox are used in application layer of the software architecture, which make it easy to transfer data and communicate each other between processes. During developing related independent modules are implemented as processes, and the customization task is done by adding or removing the related process. Above development pattern is proved by the developed project, and the parallel design and debug is easy to be done, and the credibility, flexibility, maintenance of the processes are guaranteed.

Key words: embedded system; layer pattern; ICS; mailbox; customization

0 引言

随着后PC时代的来临,嵌入式设备已经越来越多地应用于各个领域,尤其在32位嵌入式微处理器出现后,嵌入式操作系统与其结合而产生的嵌入式系统,具备更好的稳定性和开放性,在各行业技术革新中发挥着重要作用。然而嵌入式系统作为一种紧凑型专用计算机,受系统资源限制,其软件的开发不同于普通台式计算机,随着需求的发展和应用的复杂,嵌入式系统的软件开发面临着一些共性的问题^[1]:

(1)从软件开发过程看,为了节约系统资源消耗,嵌入式应用软件基本采用面向过程的开发方式,代码复用度低,在面对复杂的多任务应用功能实现时,很难展开并行的开发和调试,这是限制嵌入式应用软件开发效率的一个重要因素。

(2)嵌入式应用软件常需要对硬件驱动接口进行操作,而嵌入式设备应用对象千差万别,多种多样的嵌入式操作系统

之间兼容性差,往往需要针对不同的硬件编写操作系统内核移植代码。

(3)在工程应用中,许多嵌入式软件进行代码升级、功能拓展时,需要向嵌入式操作系统平台做代码移植。

领域工程和应用工程中,基于复用的软件生产技术是其关键技术^[2]。然而上述问题造成了嵌入式软件开发效率低,开发周期长,且影响了产品的可靠性、移植能力和市场响应能力。目前嵌入式软件开发需要很好解决以下问题:

(1)软件复用问题,包括功能代码复用和过程代码复用。

(2)抽象层开发问题,包括目标硬件平台的抽象层开发和操作系统抽象层开发。

通过研究和项目实践,本文提出一种面向嵌入式应用的新的软件系统结构解决方案,以改进的层模式软件架构和基于ICS消息总线的功能模块进程化开发模式,来解决以上问题。

收稿日期:2009-01-05;修订日期:2009-02-26。

基金项目:广州市开发区科技计划基金项目(2007Q-P075)。

作者简介:万相奎(1976-),男,湖北广水人,博士,副教授,研究方向为嵌入式系统、网络测控与仪器;陈建明(1962-),男,高级工程师,研究方向为嵌入式系统、微机计量与测控。E-mail: xkwan@gdut.edu.cn

1 改进的层模式体系结构

一个通常的嵌入式系统软件体系可按层次划分为如图 1 所示^[3]。

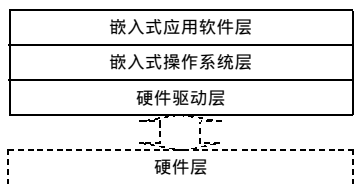


图 1 通常嵌入式系统软件层结构

专用的嵌入式设备,其嵌入式应用软件层通常为一个进程,实现某个特定的功能。嵌入式操作系统层集成了完整的文件管理、进程管理、内存管理、网络管理、电源管理等功能。硬件驱动层实现对特定外设的直接访问,提供系统访问接口。开发一个具有特定功能的嵌入式设备,必须具备相关硬件、软件、嵌入式操作系统和特定领域知识,这对开发人员提出了相当高的要求,而且当开发新的嵌入式设备时,代码复用度很低。

本文将以上层结构进行了细化和改进,新的嵌入式系统软件层结构如图 2 所示。

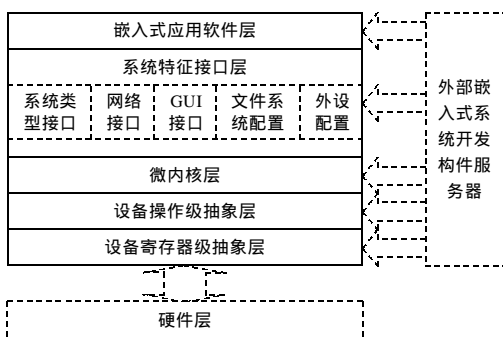


图 2 改进的嵌入式系统软件层结构

这里的嵌入式应用软件层由多个进程组成,它们协调工作,共同完成一个嵌入式设备的特定功能。其详细工作机理将在第 2 节详述。

系统特征接口层提供嵌入式设备个性化配置接口,它包含了用户开发该嵌入式设备所希望的一些特征:系统芯片信息、功能模块、文件系统类型、网络类型、GUI 图形库、外设信息库等。根据这些特征,结合下层微内核相关模块,通过配置定制出特定的系统软件平台。

微内核层只提供基本的管理功能,包括任务调度、内存管理、时钟管理等。其它应用构件,如网络协议、文件系统和 GUI 库等均工作于用户态,以系统进程或函数调用的方式工作,这样它们被剥离开来,可存储于嵌入式系统开发构件服务器上,以实现它们的复用,用户可以根据需要选择配置相应的应用构件。

设备操作级抽象层抽象并封装了硬件的功能行为,提供了对各种板子的硬件功能操作的统一的软件接口,它是保证

操作系统可移植性的关键。可将不同板子对应的功能操作包开发出来后,存储于嵌入式系统开发构件服务器上,实现硬件功能操作代码的复用。

设备寄存器级抽象层直接与硬件相关,封装了外设寄存器接口及其行为,定义了完整的寄存器序列名称。针对不同的芯片编写不同对应的驱动,将其存放于嵌入式系统开发构件服务器上,以实现对同一功能元器件操作的代码复用。

嵌入式系统开发构件服务器是一个提供可复用的嵌入式构件的外部容器。所谓构件指可被多个软件系统复用的,具有相对独立功能的系统构成成分^[4]。

在改进后的嵌入式软件体系结构下进行应用开发,通过复用技术,应用软件层只需根据嵌入式设备的特定应用需求,通过特征接口层提供的配置服务,下载嵌入式系统开发构件服务器上对应层的软件构件,来实现对操作系统的定制和硬件相关代码的开发,避免了重复开发工作,且保证了系统的扩展性、可移植性和稳定性。

2 基于 ICS 的多进程嵌入式应用层软件开发

基于改进的嵌入式系统软件层模式体系结构,可提高其操作系统层和抽象层的软件复用度。而通常嵌入式设备面向特定专业应用领域,嵌入式应用软件层实现其特定的控制或处理任务^[5]。目前基于单进程的嵌入式应用软件开发难以实现复杂多任务功能需求,本文提出基于 ICS 通讯机制的多任务进程化嵌入式应用软件开发新方案。

2.1 ICS 消息总线通讯机制

ICS(intertask communication and services)即任务间通讯服务,它是作者所在的项目组基于套接字和 UDP 协议研发,用于嵌入式设备应用层的复杂功能实现而开发的。它独立于操作系统,由 3 部分组成:

ICS 客户端:需要使用任何 ICS 功能的进程;

ICS 服务器:实现远程计算机上的邮箱访问任务;

ICS 节点检查器:控制是否允许当前应用程序指定的 ICS 邮箱能够通知。当某一进程需要使用 ICS 邮箱时,必须登记注册该邮箱,ICS 服务器将通过发送“事件”消息到该邮箱通知其是否能用。

ICS 工作机制通过“ICS 邮箱”来为操作系统上层应用程序提供统一标准的数据交互方式,使得应用进程间的通讯独立于操作系统,这样当操作系统版本升级和使用新的操作系统时,不必涉及上层应用的更改。

任意需要使用 ICS 消息的进程首先需要创建消息邮箱,并通过周期性检查邮箱中的消息来获得数据。ICS 库函数可以方便满足进程间的数据通讯。每个进程事先都需要创建一个“ICS 邮箱”,并通过周期性检查邮箱中的消息来获得数据。需要发送数据给其它进程的进程只需要将格式化后的数据以消息的形式发送给目的进程邮箱,对方即可接受到数据。

图 3 描述了基于 ICS 的多进程间数据交互过程。计算机启动后,位于同一台计算机内部的“进程 1”和“进程 2”开始运行,首先调用 ICS 函数库的 API 函数分别各自创建自己的邮箱“QX”及“RX”,当“进程 2”要发送数据给“进程 1”时,它将把数据通过 ICS 消息总线直接发送到“QX”邮箱;“进程 1”通

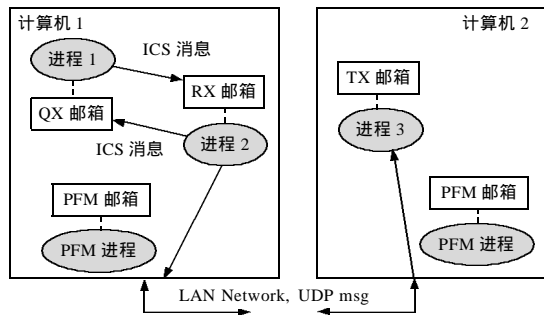


图3 基于ICS邮箱的进程间通迅过程

过查询自己的邮箱并提取数据来获得“进程2”发来的数据。同样的道理，“进程1”也可发送数据到“进程2”邮箱，进行数据通讯。

这种机制也可应用于远程网络上的进程间通讯，此时它需要一个ICS服务器(PFM进程及其邮箱)来支持。如图3，当计算机1中的“进程2”要和计算机2中的“进程3”通讯时，它调用ICS函数库的API函数，像发送本机其它进程一样发送ICS消息至对方邮箱，由于邮箱创建时，各个邮箱的ID号被登记，在PFM服务器作用下，根据目的邮箱的ID号，该消息被发送到远程计算机2中的“进程3”的邮箱中“TX”中，“进程3”通过定期检查自己邮箱，来获得“进程2”发来的数据，实现远程数据交互。

ICS消息的消息头结构定义如下：

```
struct{
    unsigned int msg_class;
    short msg_id;
    WORD msg_length;
    WORD msg_qualifier;
    int msg_token;
    BYTE fip_id;
}
```

其中msg_class代表消息类别，不同应用项目可定义不同的消息类。msg_id代表消息ID，ICS消息根据数据含义的不同化分为命令消息、事件消息和数据消息。msg_length表示该消息所附带的数据长度。msg_qualifier为消息附加参数，以区分同一命令(和事件)消息下不同的动作。msg_token代表消息应答区别号。多个命令和事件发出时，意味着有多个对应的应答，为了表示发送和应答的对应关系，定义了该项。所有使用ICS消息通讯的进程必须定义自己的ICS消息结构，所有的ICS消息结构中必须定义以上的消息头，以便于接受方解析收到的ICS消息。

2.2 基于ICS的功能模块进程化嵌入式应用层软件开发

ICS实现了嵌入式操作系统下的进程间通讯，使得嵌入式系统平台下进程间的通讯可靠、灵活。这为复杂多任务需求的嵌入式设备并行开发、设计和调试提供了条件。以前用一个进程来全部实现复杂任务的情况，现在基于ICS进程通讯机制，可以将其模块化，由多个研发人员同步开发，独立运行和调试各个功能模块，最后进行功能集成测试。

下面以作者所在项目组研发的项目：嵌入式电脑加油控

制器为例来简单阐述基于ICS机制的功能模块进程化嵌入式应用软件开发。

嵌入式电脑加油控制器以ARM和嵌入式Linux操作系统构建嵌入式基础运行平台，它要求实现复杂的多任务，包括支持图形化人机交互、智能的远程诊断、集成油气回收控制、多种交易支付方式、控制和计量四枪同时加油、复杂的交易数据和配置参数管理等等。采用单一进程在嵌入式系统上实现以上任务是难以想象的。

基于ICS消息通讯机制，则可将整个控制系统的功能进行分割，把相对独立的任务以独立进程来实现。任务进程化使得各任务的内部实现变得独立和清晰，各任务间的通讯和数据交互通过不同类型的ICS消息来实现，如图4所示。

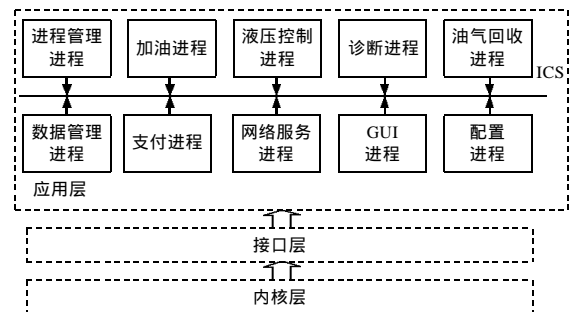


图4 基于ICS消息的嵌入式加油控制器应用层结构

应用层功能软件由多个任务进程组成，其中的进程管理进程是一个总的控制进程，它负责监控所有任务进程的状态、通过发送ICS消息命令控制其状态切换(每个进程都具有初始态、挂起态、运行态和终止态这4种基本状态)。所有任务进程也都基于ICS通讯机制与其它相关进程通讯。这使得只要定义好任务接口后，各个任务进程模块在开发过程中被完全当作独立软件模块来实现，最大化实现了软件模块内部高内聚、模块间的低耦合，增强了控制系统的扩展性和开放性。甚至对于一些通用任务进程，完全可以将其存储于嵌入式系统开发构件服务器中，供相关项目的研发人员直接拿来使用，这无疑在嵌入式系统应用层开发这一级提高了代码的复用度。

3 结束语

目前嵌入式系统已经广泛应用于各个领域，但其产品开发方式仍然是一个百花齐放的局面，不像通用计算机软件体系结构已形成事实标准^[6]。随着复杂需求的提高，这将影响嵌入式产品开发周期、成本、质量。软件复用概念提出已久，并在通用计算机上得到广泛应用，但在嵌入式软件开发中，却鲜见利用^[7-8]。本文是在研发一个嵌入式产品项目的基础上，对涉及的嵌入式软件体系结构和应用层软件开发的归纳和总结。从实际开发效果看，显著提高了开发效率，专业研发人员和领域人员反应积极，不断就开发过程和细节提出改进意见。本文介绍了改进的嵌入式软件体系结构及其应用层的软件开发一种新模式，使用的方法是形式化的，这还不太规范，这方面的研究和开发工作还有待进一步努力，才能加以规范和推广使用。

(下转第5364页)

4KB,所以读取一个页只需进行一次读命令,节省了在寻道上面的时间。对于下层的驱动程序来说,当出现有大于4KB的特殊情况时,下层的驱动程序就会对其进行拆分。也就是说,通过修改Block的大小,减少了读操作的次数,合并了过多的拆分,从而提高了系统的读操作的性能。

如图6所示,当块大小由512Byte改为pagesize大小,也就是4KByte时,速度有比较明显的提高。

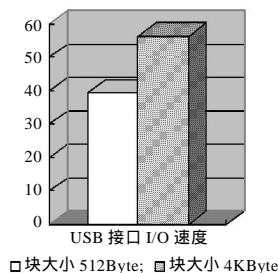


图6 文件系统优化前后的 I/O 接口速度对比/(Mbit/s)

(3) 页机制

因为在整个读取文件的过程中,都是按页为单位进行考虑和执行的,然后提交到下层,也就是内核中的 submit 动作。但是,1个页的大小,实际上也就4K,这对真实的磁盘系统、特别是高清文件来说实在是太小。而在每次提交一个页时,我们需要花费比较多的时间去将这4K的内容传到底层,执行完整个流程。如果能够在提交之前进行页的合并的判断,使其达到一定数量的页时进行传送,这样就会大大减小在传输过程中的时间,提高运行速度。

因此,在 submit 动作前,我们可以在 ntfs_readpage 前增加一个函数来判断合并多个 page,这样就不用每个 page 不停地传输过程中浪费时间,达到一次传输多个 page 的目的。

3 实验结果

本硬件平台:图7是一个基于ARM嵌入式平台的高清播放系统的测试结果。该测试的实施的软硬件平台是:ARM7TDMI的芯片系统,CPU频率为198MHz,64MB内存,6MB cache,硬盘 maxtor 4 200转,40GB;系统采用 uCLinux 2.6.10。

从试验结果中我们可以看出,在通过上述几种方法后,文件读取速度由原来的39~40 Mbit/s 提高到80 Mbit/s,也就是说对NTFS文件系统USB接口I/O速度提高达到近100%。同时在实验过程中我们发现,对于目前市面上的绝大部分高清影片,都能够流畅的播放,达到了预想中的效果。

(上接第5360页)

参考文献:

- [1] 严和平,邓荣佐,王知衍.一种用于嵌入式应用开发的软件体系结构[J].计算机应用研究,2005,22(11):89-91.
- [2] 杨英清,王千祥,梅宏,等.基于复用的软件生产技术[J].中国科学(E辑),2001,31(4):363-371.
- [3] 万建成,卢雷.软件体系结构的原理,组成与应用[M].北京:科学出版社,2002.
- [4] 田韬,张悠慧,汪东升.基于C/S架构的可扩展嵌入式系统[J].计

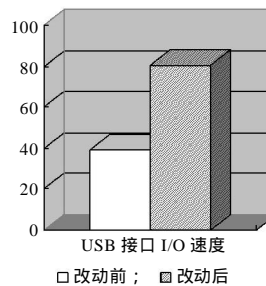


图7 文件系统优化前后的 I/O 接口速度对比/(Mbit/s)

4 结束语

Linux 内核是一个很庞大的工程,而文件系统^[9]是其中的基础。因此对于文件系统的研究涉及到很多方面复杂的知识。本文基于嵌入式Linux系统,提出了一种无卡顿播放高清影片的改良方法,通过修改Linux内核,特别是针对预读机制,块机制,页机制等,提高了USB接口I/O速度,提高比例将近100%,取得了很好的效果。另外,不仅针对NTFS文件系统,对于其它的文件系统,如FAT、SAMBA、ext文件系统等,对于USB接口I/O速度的提高都能起到不错的效果。

参考文献:

- [1] 毛德操,胡希明.Linux 内核情景分析[M].杭州:浙江大学出版社,2001.
- [2] 居锦武,王兰英.NTFS 文件系统剖析[J].计算机工程与设计,2007,28(22):5437-5439.
- [3] 梁金千,张跃.NTFS 文件系统的主要数据结构[J].计算机工程与应用,2004,39(8):116-118.
- [4] O'REILLY.深入理解 linux 内核[M].3版.北京:中国电力出版社,2007.
- [5] 郭玉东.Linux 操作系统结构分析[M].西安:西安电子科技大学出版社,2004.
- [6] 吴峰光.Linux 内核的文件预读[J].软件世界,2007(12):56-57.
- [7] 吴娴.嵌入式 Linux 文件系统的设计与实现[J].计算机工程与应用,2005,41(9):111-112.
- [8] 应骏,江峰,李莉.嵌入式设备上的文件系统优化设计[J].电子技术应用,2007(12):141-142.
- [9] 张明亮,张宗杰.浅析 FAT32 文件系统[J].计算机与数字工程,2005,33(1):56-59.

算机工程与设计,2008,29(7):1804-1808.

- [5] Kirk Zurell.嵌入式系统的C程序设计[M].艾克武,译.北京:机械工业出版社,2001.
- [6] 万相奎,徐杜,张军.基于嵌入式系统的科学仪器的构建[J].计算机工程,2007,33(14):213-215.
- [7] 许宪成,杨存详.组件技术及其在嵌入式系统设计中的应用[J].微计算机信息,2007,23(2):39-41.
- [8] 王琼,杜承烈.嵌入式实时软件通用框架模型研究[J].计算机工程与设计,2007,28(6):1372-1375.