

嵌入式实时操作系统移植技术的研究与应用

赵星星, 罗克露, 张 军, 邓 勇, 保 云

(电子科技大学计算机科学与技术系, 成都 610054)

摘要: 随着嵌入式系统在各个领域的不断发展, 嵌入式操作系统的系统软件移植技术的研究已成为嵌入式开发中的一个重要问题。该文研究了嵌入式操作系统在不同平台移植的理论与技术, 提出一种用层次化和模块化方法对嵌入式操作系统进行移植的一般方案。该方案成功运用于 CRTOSII 移植到集成 S3C2410 芯片主板的项目中, 实践证明了移植方案的正确性和可行性。

关键词: 实时操作系统; CRTOS II; 硬件相关层; 层次化; 模块化

Research and Application of Embedded Realtime Operating System's Porting Technology

ZHAO Xing-xing, LUO Ke-lu, ZHANG Jun, DENG Yong, BAO Yun

(Department of Computer Science and Technology, University of Electronical Science and Technology of China, Chengdu 610054)

【Abstract】 Along with the quick development in every fields of embedded system, the porting technology of embedded operating system has become an important problem of embedded development. This paper investigates the theory and technology of porting embedded operating system to different hardware flats, and advances a scheme which can be used to port embedded operating system with the means of blocking and layered. This scheme is applied to the project of porting CRTOSII to the board which integrates the chip of S3C2410. It proves the validity and feasibility of the scheme.

【Key words】 realtime operating system(RTOS); CRTOS II; hardware interrelated layer; layered; blocking

微电子技术的飞速发展, 使计算机硬件的集成度越来越高、体积越来越小, 其性能不断提高。而且硬件的复杂性日益增加, 对软件设计也提出了新的要求。软件开发的周期的增大, 导致软件开发成本急剧上升。嵌入式系统变得越来越复杂, 在充分发挥硬件性能的同时如何降低开发难度和成本成为了一个难题。

解决这个难题的通常做法就是运用嵌入式操作系统。嵌入式操作系统结合了嵌入式系统的特点, 其体系结构从最初的无序模块结构到层次结构, 最终发展成为现在最流行的微内核结构。微内核结构的基本思想是将操作系统中共同需要的核心功能提炼出来, 形成内核的基本功能(如进程通信、线程管理、任务通信等)。在移植性方面, 微内核结构将与机器特征相关的代码全部隔离在微内核的底层。这样, 只须修改内核中的少量代码, 嵌入式操作系统就可移植到不同的硬件平台^[1]。

通过对系统提供板级支持包和CPU依赖程序, 嵌入式操作系统实现了对不同硬件平台的支持。因此, 如何制定硬件平台与操作系统的接口以提高嵌入式操作系统的可移植性, 尽可能使操作系统的内核可以跨越多种硬件平台, 是嵌入式操作系统移植技术研究的重点^[2]。

1 嵌入式操作系统移植原理

1.1 硬件相关层原理

根据嵌入式操作系统与硬件平台的相关性和与用户的交互性, 可以将嵌入式操作系统分为4层: 用户 API 接口层, 组件层, 内核层和硬件相关层。如图1所示。

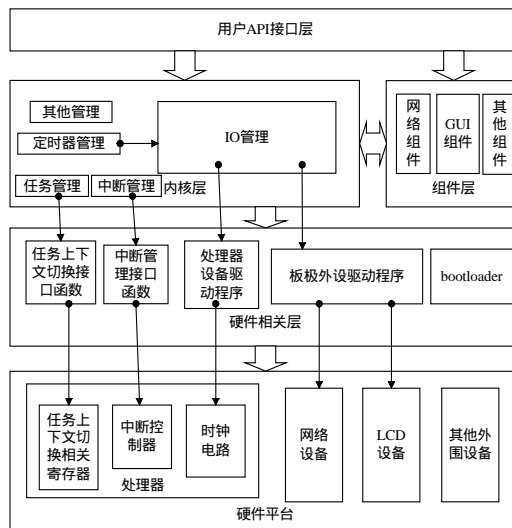


图1 RTOS的分层与分模块结构

(1)硬件相关层。此层屏蔽了底层硬件系统, 向上层提供统一的接口, 避免了操作系统移植后上层软件也需要修改的缺陷。

(2)内核层。它是操作系统的核心, 主要包括任务管理、

作者简介: 赵星星(1981 -), 男, 硕士研究生, 主研方向: 嵌入式实时系统及应用; 罗克露, 教授; 张 军, 工程师; 邓 勇, 硕士研究生; 保 云, 工程师

收稿日期: 2006-10-14 **E-mail:** purplestar_zxx@sina.com

用户扩展管理、内存管理、进程间通信、同步与互斥、中断管理、时钟管理和定时器管理以及 I/O 管理、多处理器管理、出错处理等。

(3)组件层。该层在内核层上采用组件方式进行功能扩展,包括高可信服务的 Security Kernel 组件和高可靠组件,以及复杂应用中所需要的文件系统模块、网络系统模块、GUI 模块等。这些组件应能方便地配置、剪裁,以适应实际应用要求。

(4)用户 API 接口层。结合 OS 内核和组件功能,提供给用户编程需要的接口。

因为嵌入式硬件系统具有多样性,所以设计怎样的硬件相关层与操作系统上层之间的接口,使操作系统只需要修改必要的硬件相关代码就可以实现移植到不同的硬件平台,是嵌入式操作系统移植的关键问题。硬件相关层的结构应清晰和简洁,以便减少操作系统移植过程中所做的工作。

1.2 硬件相关层的结构与功能

嵌入式操作系统硬件相关层从整体上可以分为 CPU 依赖程序和板极支持包(BSP)2 部分,也可以按照它实现的主要功能细划分为 4 部分:Bootloader,芯片内部设备驱动,外围设备相关驱动,CPU 依赖程序。各部分完成的主要功能如下:

(1)Bootloader:在嵌入式操作系统运行之前,必须有一段程序来初始化部分硬件设备并把操作系统代码从 FLASH 拷贝到 RAM 中,在这之后程序为操作系统的运行建立软件级的初始化状态,最后跳转到操作系统内核入口。完成这样工作的一段代码,通常称之为系统引导程序,即 Bootloader。

(2)芯片内部设备驱动:包括内部时钟驱动和中断控制器驱动,内部时钟驱动提供时钟控制器与内核 IO 管理模块的软件接口,而中断控制器驱动提供中断控制器与内核中断管理模块的接口。其中,中断控制器驱动的实现是所有其他外设管理的基础,而且它的实现方式与其他驱动的实现也有很大不同^[3]。

(3)设备驱动部分:提供板极外围设备与内核 IO 管理模块的接口。

(4)CPU 依赖程序:提供与 CPU 类型密切相关的部分底层函数的实现,并根据 CPU 的类型提供操作系统的部分数据类型的定义。

2 嵌入式操作系统移植方案的设计与实现

目前 CRTOS II(Chinese real-time operating system, second edition)只支持 x86 和 POWER PC 的处理器,现结合 CRTOS II 在 S3C2410 芯片上的硬件相关层的实现,提出一种用层次化和模块化方法对嵌入式操作系统进行移植的一般方案。模块化是指使用高内聚低耦合的方法将硬件相关层划分为 4 个模块,各模块的开发工作可以并行进行,大大缩短了移植的时间。而层次化是指在同一模块中将与硬件密切相关、不便移植的部分放在第 1 层,主要以汇编语言编写;将与硬件无关、比较容易移植的部分放到第 2 层,以 C 语言编写,以提高程序的通用性和可读性。

2.1 移植环境的建立

在嵌入式软件开发过程中,开发环境的建立通常根据不同的应用有着较大的差别,其中,最主要的就是交叉开发工具的选择。一般的商业实时操作系统都会提供配套的开发工具,然而,要开发自己的操作系统就需要通过建立一套自己的开发工具。以下是 CRTOS II 操作系统在 S3C2410 上实现时的移植环境建立方案,在进行其他移植工作的时候可以以

此为参考建立对应的开发环境。

(1)开发模式:采用宿主机/目标机模式。

(2)宿主机:在 Windows 上用 Cygwin 虚拟类 Unix 操作系统环境,也可以直接建立 Linux 开发环境^[4]。

(3)目标机:采用华恒 HHARM2410 开发板,此开发板集成了 S3C2410 芯片,还提供 FLASH、SDRAM 和各种通用外围扩展接口。

(4)开发工具:使用 GNU 的开发工具链,由于没有现成的 CRTOS II 在 ARM 上的开发工具,因此需要下载 GNU 开发工具的源代码来编译一套 CRTOS for arm 的工具链,并且要注意 2 点:

1)下载的源代码包版本要匹配,否则在编译时会出现版本不匹配造成的编译出错。

2)由于 GNU 只支持现有的比较普及的操作系统,而要让它能编译支持 CRTOS 操作系统的交叉工具链,需要对源代码中的几个配置文件进行小的修改。

2.2 各模块的移植工作

2.2.1 Bootloader 模块的移植方案与实现

Bootloader 对硬件依赖很强,即使是基于同一种 CPU 构造的不同的硬件系统,通常也需要修改 Bootloader。为了提高 Bootloader 的可重用性和可移植性,可将 Bootloader 程序按层次化的方法分为 2 层。

第 1 层应当映射到 CPU 启动代码处,系统启动后从这里开始执行,这个阶段的程序主要做的工作是:(1)设置异常向量表,为启动过程中用到的中断和异常做准备。(2)初始化 CPU、IO 端口、看门狗等硬件设备,将其置于一个合适的状态^[5]。(3)配置存储控制器,为 Bootloader 的第 2 层的代码准备内存空间。(4)设置堆栈,为运行接下来的第 2 层的 C 语言程序做准备。(5)将 Bootloader 的第 2 层的代码和操作系统内核映像复制到内存空间中。(6)跳转到第 2 层的 C 语言入口。

第 2 层程序主要做的工作是:(1)完成 CPU 相关参数配置,并进行相关的初始化。(2)完成 BSP 相关参数的配置,并进行相关的初始化。(3)设置操作系统内核启动参数。(4)建立操作系统的运行工作空间。(5)跳转到操作系统内核入口。

按照以上的原理和步骤,Bootloader 具体实现中的函数调用情况如图 2 所示。

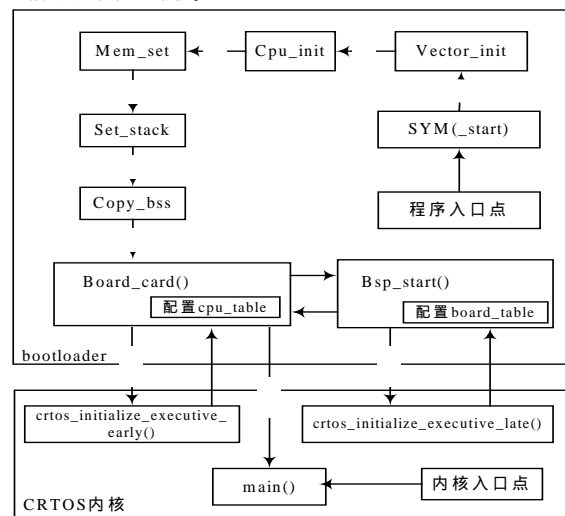


图 2 Bootloader 程序的运行流程

Bootloader 从 _start 处进入,按照上述的原理一步步执行,在第 1 层的任务完成后,进入 Boot_card() 函数,开始第 2 层

的任务,所有任务完成后从 OS 内核入口点函数 main()进入 OS 内核,然后就可以在 main()函数中调用用户应用程序。

2.2.2 内部设备驱动和外围设备驱动程序的移植方案与实现

芯片内部设备驱动部分主要包括内部时钟驱动和中断控制器驱动,因为内核中的很多模块的功能需要这 2 部分的支持,所以在移植过程中应当首先实现它们。而其中中断控制器驱动的实现方式与标准外设驱动不一样,它的实现是根据操作提供安装中断服务程序函数,以及对中断发生时的中断接管函数的功能来实现的。

内部时钟主要向外界提供一个时间参考对象,使一些需要时间控制的事件或者进程等得到可靠的控制。诸如进程控制的时间片、实时的精度、WATCHDOG 模块,都需要相应的定时器模块的支持。

外围设备驱动程序提供对板级外设的软件支持,它的实现首先需要开发板提供外设所需要的硬件接口,一般包括 USB 设备、LCD 设备等,这些驱动只在要提供扩展功能时才需要,没有它们操作系统也能提供基本的功能。

由于内部时钟驱动和外设驱动是根据操作系统的 IO 管理模块的要求来实现的,而 IO 管理模块是操作系统内核的一部分,因此这部分不需要移植,只要开发的硬件驱动按照 IO 管理模块的要求实现标准接口函数即可,开发的难点在于对外设本身结构和功能的了解。中断控制驱动也可以按照层次化方法分为 2 层,它们主要提供如下 4 个功能:

(1)根据 CPU 中断控制器结构的不同,为 OS 内核的中断管理模块提供不同的中断处理程序的安装函数和卸载函数。

(2)设置中断响应的软硬件环境。

(3)提供中断接管函数,当发生外设中断请求时,操作系统跳到中断接管函数处,它保存了最小的上下文并重新设置中断指针后,才执行真正的 ISR,ISR 执行完后返回控制给该函数,由它决定是否进行线程的重调度并在所有的中断执行完后调用线程调度操作,该函数以汇编语言实现。

(4)它负责响应中断后从中断控制器的中断偏移量寄存器中获取中断向量号,跳转到对应的 ISR 中去执行。

CRTOS II 移植到 S3C2410 芯片上时,中断控制部分需要完成的主要工作如图 3 所示。

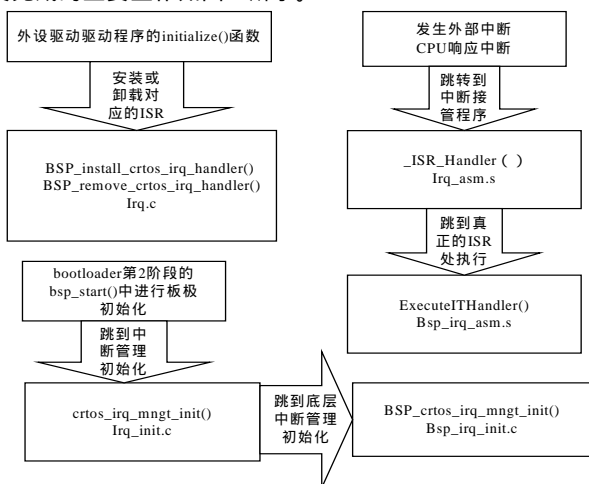


图 3 中断程序间的调用结构

其中,irq.c 提供上述功能 1;irq_init.c 提供功能 2;irq_asm.s 提供功能 3;bsp_irq_asm.s 提供功能 4。

2.2.3 CPU 依赖程序的移植方案与实现

CPU 依赖程序中包含既与底层硬件紧密相关又要在 OS 内核直接调用的函数。CPU 依赖程序的主要提供如下功能:

(1)针对不同体系结构 CPU 的数据类型和数据结构的定义。

(2)提供 CPU 相关数据结构的初始化,CPU 响应中断请求的屏蔽与开启。

(3)进程上下文切换函数,主要的功能是切换时对相关寄存器(R2-R14)内容的保存与恢复;进程上下文环境恢复函数,它在进程从就绪态进入运行态时调用,以恢复进程的上下文环境。

在实践中分别在文件 armtypes.h, Cpu.c, cpu_asm.s 中对 CPU 依赖程序的功能给予实现。

3 嵌入式操作系统移植后的测试

3.1 功能验证测试

为验证移植后的系统功能,笔者设计了一个包含 CRTOSII 所有模块 API 的程序,但不是覆盖性测试。经统计,该程序包含了超过 95% 的 CRTOS II API。这部分程序功能简单但是语句繁琐,不再一一列出。程序的执行全部运行正常。

3.2 系统实时性能测试

本次实时性能测试使用 SSC 测试。SSC 测试是美国的 SSC(superconducting super collider)实验室提出的一个测试标准。SSC 实验室作为一个实时操作系统的使用者,为了评价实时操作系统提出了自己的测试指标。本文主要就 EOS 实时性能的 3 个关键指标(中断服务响应时间,任务响应时间和任务上下文切换时间)对 CRTOS II 系统的实时性进行测试。SSC 测试结果及其分析见表 1。

表 1 CRTOS II 实时性能 SSC 测试结果

测试项目	测试次数	平均时间/ μ s
中断服务响应时间	1	22
交替悬置/恢复任务时间(Tir)	100	380
悬置/恢复任务时间(Thr)	100	206
上下文切换时间(Tc)	100	87
任务响应时间	100	249

由于硬件平台不同,测试结果无可比性,但是可以从表中的测试结果看出 CRTOS II 在 HHARM2410 平台下的 3 个关键实时性能指标都在 μ s(10^{-6} s) 级别,可以满足绝大部分的实时要求。

3.3 存储性能测试

该性能的测试分析主要针对内存空间开销。CRTOS II 操作系统内存空间开销主要包括 2 部分:操作系统内核内存空间开销和 BSP 级驱动代码内存空间开销,这里所说的 BSP 级驱动代码主要包括处理器驱动代码和外设驱动代码。CRTOS II 操作系统内存空间开销见表 2。

表 2 CRTOS II 操作系统内存空间开销

硬件平台	模块	
	操作系统内核 libkernel.a/KB	BSP 级驱动代码 libbsp.a/KB
x86 PC 平台	3 784	901
ARM 平台	3 826	496

由表 2 的 BSP 模块代码部分对比的空间开销可知,libbsp.a 在 x86 PC 硬件平台和 ARM S3C2410 硬件平台的内存空间分别为 901KB 和 409KB,移植后 BSP 部分的改动较大,这说明了 EOS 在不同硬件平台的移植主要体现在底层实现部分,如处理器驱动和外设级驱动部分。

4 结束语

本文研究了嵌入式操作系统的移植理论与技术,提出采用层次化和模块化方法对嵌入式操作系统进行移植的一般方案。在嵌入式软件系统移植中引入层次化、模块化的方法, (下转第 95 页)