

基于一个实时内核的嵌入式 Internet 技术的研究与实现^{*}

关沫 赵海 韩光洁 张文波

(东北大学信息科学与工程学院 沈阳 110004)

摘要 基于微内核体系结构的思想,从嵌入式系统开发 Internet 应用的角度,提出一个实时内核—WebitX 的体系结构及设计方案。为支持嵌入式 Internet 环境下的复杂应用,WebitX 被设计成一个基于优先级的抢占式多任务内核,解决了任务调度、任务间同步与通信、定时管理、网络协议处理等主要问题,为嵌入式 Internet 应用提供了完备的系统服务,确保了嵌入式 Internet 应用的实时性和可靠性。为实现嵌入式设备与 Internet 的互联,WebitX 提供了一个嵌入式 TCP/IP 协议栈,该协议栈在嵌入式系统资源受限的条件下,采用模块化的方法实现了 TCP/IP 协议簇的一个功能子集,为传统设备提供了网络接口,从而实现了高效的嵌入式 Internet 通信。

关键词 实时内核,RTOS,嵌入式 Internet,微内核体系结构,WebitX,TCP/IP

Research and Design of Embedded Internet Technology Based on a Real-Time Kernel

GUAN Mo ZHAO Hai HAN Guang-Jie ZHANG Wen-Bo

(School of Information Science & Engineering, Northeastern University, Shenyang 110004)

Abstract A real-time kernel—WebitX's architecture and design scheme with microkernel architecture is proposed from the viewpoint of development Internet applications in embedded system. In order to support complicated applications under embedded Internet environment, WebitX is designed a multitask preemptive kernel based on priority. To guarantee the timeliness and reliability of embedded Internet applications, such system services as task scheduling, synchronization and communication between tasks, timer management and network protocol are provided by WebitX. An embedded TCP/IP stack is provided by WebitX to realize devices linked to Internet and partial functions of TCP/IP suite are implemented using a modular solution with the limit of hardware resource. A network interface for traditional devices is provided by this stack and accordingly efficient embedded Internet communication is achieved.

Keywords Real-time kernel, RTOS, Embedded internet, Microkernel architecture, WebitX, TCP/IP

1. 引言

传统的嵌入式系统功能比较简单,程序代码直接控制硬件,没有具体区分系统软件和应用软件,系统软件完全依赖硬件,可移植性差。但随着嵌入式技术的不断发展,传统的单任务工作方式已不适应复杂应用的需要。为提高嵌入式系统的可靠性,并满足多任务应用的需要,在嵌入式应用中使用实时操作系统(Real-Time Operating System, RTOS)正逐渐成为一种趋势。并且随着网络技术在嵌入式系统中日益广泛的应用,基于微控制器的嵌入式 Internet 技术的进一步发展也需要 RTOS 的支持^[1]。在这种情况下,基于 RTOS 的嵌入式 Internet 技术逐渐成为嵌入式系统和 Internet 应用研究的一个新热点。

嵌入式 Internet 技术是一种设备接入技术或异种网络互连技术,主要解决如何通过 Web 和嵌入式技术实现从不同子网、不同的物理区域对接入到 Internet 的设备和异类子网进行监控、诊断、管理及维护等操作,使用户对接入到 Internet 上的各种设备或其它类型的子网具有远程监控、诊断和管理的能力^[2]。与传统的嵌入式应用不同,嵌入式 Internet 应用要求 RTOS 不仅要具有良好的实时内核,而且要提供相应的网络协议栈,并具有一定的文件管理能力。

本文从嵌入式系统开发 Internet 应用的角度,提出一个针对嵌入式 Internet 环境下的 8 位 RTOS 的实时内核 WebitX 设计方案,解决了任务调度、任务间同步与通信、嵌入式 TCP/IP 协议栈处理等主要问题,旨在为嵌入式 Internet 应用提供强大的系统服务,从而实现高效的嵌入式 Internet 通信,减小软件对低成本处理器的开销。

2. 实时内核 WebitX 体系结构研究

2.1 WebitX 体系结构

嵌入式系统由于其自身硬件资源有限,不同于通用计算机系统,因此在通用计算机操作系统内核实现的存储器管理、设备管理、文件系统、网络通信等功能模块不可能在嵌入式操作系统的内核中全部实现。本文提出的嵌入式实时多任务内核 WebitX 就是根据嵌入式系统的特点,尽量减小系统的尺寸和额外开销,采用微内核的思想进行设计^[3],即在操作系统内核中仅保留最基本、最重要的系统服务,而将通用计算机操作系统内核中的网络通信、文件管理、设备管理等模块,从内核中移出,并在用户空间实现,其体系结构如图 1 所示。

WebitX 在内核空间实现任务调度、定时管理,并提供信号量和消息传递机制实现任务间同步与通信。在内核以外的用户空间可以利用内核提供的基本服务实现网络通讯、文件

^{*} 基金项目:国家自然科学基金资助项目(69873007)。关沫 博士研究生,研究方向为嵌入式互联网和计算机网络;赵海 教授,博士生导师,研究方向为嵌入式互联网、计算机网络和数据融合;韩光洁 博士研究生,研究方向为嵌入式互联网和计算机网络;张文波 博士研究生,研究方向为嵌入式互联网和计算机网络。

管理、I/O 管理等功能。

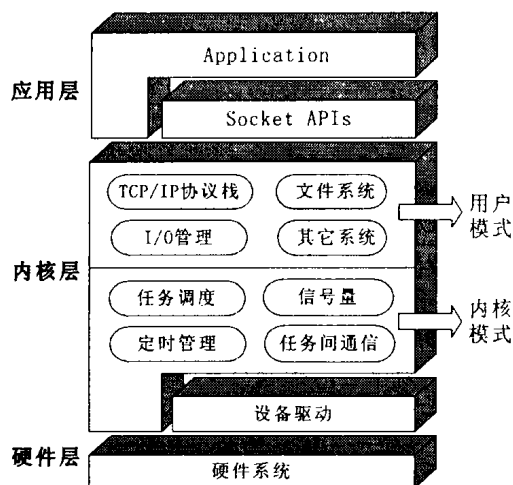


图1 实时内核 WebitX 体系结构

2.2 硬件平台

实时内核 WebitX 的开发及运行环境主要是基于 8 位微控制器(MCU),系统拥有 8kB 的 Flash 程序存储器、512 字节的片内 EEPROM、512 字节片内 SRAM(外部 SRAM 可扩展到 64kB)、32 位 I/O 口和一个内置的 UART。MCU 通过 UART 口经电平转换或 I/O 口与现场设备的控制线相连,可实现对设备的维护与监控。另外,MCU 通过连接到 ISA 总线的网卡芯片上,能够对 Internet 上的用户提供一种通过以太网接口访问嵌入式系统的通道^[2,4]。系统还在 MCU 外部扩充了一个 32kB 的串行 EEPROM 芯片来存储系统工作时要用到的文件数据,其中的文件数据由 WebitX 的文件系统来管理,包括 HTML 网页、GIF 和 JPEG 图像等。

3. 实时内核 WebitX 的实现

3.1 任务调度

任务调度是系统内核最主要的功能,为确保嵌入式 Internet 应用的实时性和可靠性,WebitX 采用了基于优先级的可抢占式调度,并配以同优先级时间片轮转的任务调度策略。

实时系统中,基于优先级的任务调度可分为静态优先级调度和动态优先级调度^[3,5]。静态优先级调度给任务分配固定优先级,短周期任务具有高优先级,例如单调率 RM 算法^[5,6]。动态优先级调度为一个任务的不同调用分配不同的优先级,例如最早时限优先(EDF)调度算法根据不同任务的时限进行任务调度,时限最早的任务具有最高优先级。静态调度运行时开销小,而调度利用率低;动态调度比静态调度具有更好的调度利用率,但由于要根据任务时限的变化不断进行任务排序,导致运行时开销增大。

为了充分利用嵌入式系统的资源,减少系统开销,实时内核 WebitX 任务调度采用了 RM 和 EDF 调度相结合的策略,目的是针对不同的任务特性采用不同的调度方法^[5,7]。根据不同任务的时间要求,将一部分任务插入到静态优先级任务队列,采用 RM 调度方法;将另一部分任务插入到动态优先级任务队列,采用 EDF 调度方法。对于具有同一优先级的任务则根据先来先服务的原则采用轮转式调度机制,以保证同优先级任务具有平等的运行的权利。

3.2 任务间同步与通信

WebitX 提供信号量和消息邮箱两种机制来实现任务间

的同步和通信。

1. 信号量(Semaphore) 是被 RTOS 普遍使用的管理共享资源、实现任务间同步与互斥的机制。在 WebitX 中, Semaphore 被定义为 3 个字节的数据结构:第一个字节存放信号量的值;另外两个字节作为指针,指向等待该信号量的任务控制块队列(waiting list)。Semaphore 的值为 1 时,该信号量有效;值为 0 说明该信号量已被占用,但 waiting list 为空;负数则表明已经有被它阻塞的任务,所有被阻塞任务的任务控制块(Task Control Block, TCB)在信号量的 waiting list 里按优先级排队。考虑到信号量操作的互斥性,WebitX 中对以下信号量操作的系统函数都被设计成不可中断的。

(1)InitSemaphore():初始化一个信号量,把信号量的初始值设为 1,并将其 waiting list 初始化为空。

(2)WaitSemaphore():请求获得或等待一个信号量。一个任务要访问某种共享资源(如 EEPROM)前会请求代表该资源的信号量。如果该任务请求的信号量值为 1,此函数把信号量的值减 1,表明任务获得了信号量,可继续执行;如果请求的信号量值为 0 或负数,该任务因无法获得信号量而被挂起,被挂起任务的 TCB 从就绪队列中移出,并插入该信号量的等待队列。

(3)SignalSemaphore():释放指定信号量,把该信号量的值置为 1,并检查该信号量的等待队列是否为空。如果该队列不为空,就令 waiting list 中第一个任务进入就绪态。然后调用相关函数寻找下一个要执行的任务,并在需要时进行任务切换。

2. 消息邮箱 为使相互合作的任务之间能够相互通信,实时内核 WebitX 采用了消息邮箱机制,提供的邮箱操作有:创建和删除邮箱、初始化邮箱、发送和接收消息、等待消息进入邮箱等。

在 WebitX 中,消息邮箱被设计为一个指针型变量,该指针指向一个包含了特定“消息”的数据结构。通过内核服务,一个任务或一个中断服务程序可以把一则消息放到邮箱里去。同样,一个任务也可以通过调用内核服务接收这则消息。每个邮箱都有等待其中消息的任务列表,要得到消息的任务会因为邮箱为空而被挂起。

3.3 定时管理

定时管理也是实时内核的一个重要功能,它为用户提供任务定时等系统服务。要实现定时功能,系统内核需要有一个周期性的定时中断来作为记时的单位,这个定时中断称为时钟节拍。WebitX 的时钟节拍是通过设置系统硬件的定时/计数器获得的,具体的时钟节拍频率由具体任务决定。WebitX 通过软件的定时器数据结构 Timer 和一个定时器队列 TimerQue 实现定时服务。其中 Timer 有三个字段,分别是:指向定时器队列中下一个 Timer 的指针;指向等待该定时器任务的指针;一个 16 位整数 Count 记录定时器的当前值。全局变量 TimerQue 是定时器队列的头指针,如图 2 所示。用户定义并使用的定时器在时限到来之前都按定时长短的顺序排放在定时器队列中,等待期限短的定时器放在队列前面。处在定时器队列后面的定时器要等到其前面的定时器期满后开始计时。

用户任务通过调用系统函数 TimerDelay()将指定的 Timer 根据其延时的时钟节拍个数插入到定时器队列的适当位置。在时钟节拍的中断服务程序中调用 TimerTick()函数,将定时器队列 TimerQue 上第一个 Timer 的 Count 值减 1 后判断该值是否等于 0。如果不等于 0,说明没有定时器到期,此函数结束;如果等于 0,说明此 Timer 已经到期,内核将它从

队列 TimerQue 中删除,并使等待这个 Timer 的任务进入就绪态,然后继续处理 TimerQue 上的下一个 Timer,直到把所有到期的 Timer 处理完。

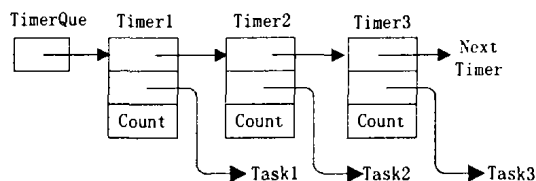


图2 定时器队列

4. 基于 WebitX 的嵌入式 Internet 技术

4.1 网络通信系统模型

由于嵌入式系统自身资源有限的特点,充分利用资源、提高系统性价比是开发嵌入式应用的根本特点。本文利用嵌入式实时内核 WebitX 提供的基本系统服务,提出在 Ethernet 网上实现嵌入式设备连接到 Internet 的设计方案,完成 TCP/IP 网络通信中的协议处理功能,在低成本的嵌入式系统中实现高性能的 Internet 通信。

嵌入式网络通信模块在实时内核 WebitX 的用户空间加以实现,其结构模型如图3所示。实时内核 WebitX 不参与具体的协议处理,只负责网络通信模块、设备驱动程序和应用程序进程之间的任务调度和资源分配,及时响应数据发送和接收中断请求。

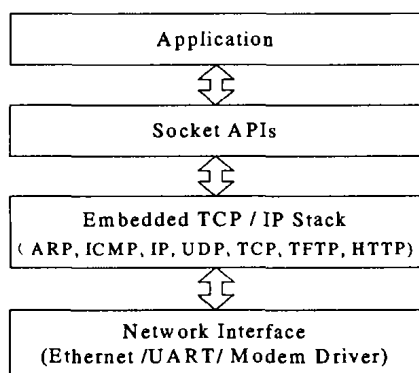


图3 WebitX 网络通信结构

系统初始化时,WebitX 在 SRAM 中创建共享的数据区,存放本地系统的网络配置参数。不同层次的网络进程对数据封装时,WebitX 根据进程标识符识别出不同的网络进程,将协议头部数据地址指针传递给相应的进程,同时使用信号量和消息传递机制,保证进程间的同步与通信。

4.2 嵌入式 TCP/IP 协议栈

嵌入式 Internet 技术很好地解决了设备和 Internet 的互联问题,人们可以利用现有的公共通信网络,不受地域的限制,通过 Web 对网络中各个节点的现场设备进行远程监控和维护。实现这一功能的核心技术是将 TCP/IP 协议栈嵌入到设备中,使其具有嵌入式 Web 服务器功能。

传统的 TCP/IP 协议把大量的精力用在保证数据传送的可靠性和数据流量的控制上,而没有过多地考虑系统实时性的要求。而在实时性要求比较高的嵌入式系统,传统的 TCP/IP 协议并不能满足其实时性的要求;另外传统 TCP/IP 的实现过于复杂,需占用大量系统资源,当用于自身资源有限的嵌入式系统时必须考虑其性价比,根据需要有所取舍,合理选择相关协议的实现和处理方法。本系统根据需要对 WebitX 的网络协议栈进行了模块化裁剪,设计并实现了 TCP/IP 协

议栈的一个子集,包括 ARP,ICMP,IP,UDP,TCP,TFTP,和 HTTP 等协议的全部或部分功能。该协议栈为传统设备提供了网络接口,为用户通过网络访问设备提供了嵌入式 Web 服务器。

在 WebitX 中,按照网络体系分层的设计思想^[8,9],TCP/IP 协议栈的每一层都被设计成一个独立的模块,各层只负责处理本层的数据,然后通过函数调用把控制权交给上层或下层的模块。在物理层,微控制器通过 I/O 口对以太网控制器芯片(RTL8019AS)进行控制和访问。在数据链路层,WebitX 提供了用汇编语言开发的网卡芯片驱动程序并实现了 ARP 协议的地址解析功能。在网络层,WebitX 的 IP 和 ICMP 模块分别实现了 IP 数据包的接收与发送和 ICMP 的回显应答(PING)功能。在传输层,TCP 模块和 UDP 模块分别为上层应用提供了有连接和无连接的数据传输服务。在应用层,WebitX 提供了基于 TFTP 协议的简单文件传输服务和基于 HTTP 协议的嵌入式 Web Server 等服务。

在设计上,WebitX 对 TCP/IP 数据包采用了全新的字节处理方式。为了尽量少地使用 SRAM 作为发送或接收数据的缓冲区,在不影响协议栈功能的前提下,WebitX 在发送或接收一个或多个字节数据的同时对数据进行打包或解包处理,而不是等到所有的数据打好包后再发送或等完整的数据包接收到再进行解包,从而降低了协议栈对 SRAM 作为数据缓冲区的要求。

结论 本文从嵌入式系统开发 Internet 应用的角度,提出一个基于 8 位微控制器的实时内核 WebitX 的软件体系结构及设计方案。为保证嵌入式系统的可靠性及实时性,WebitX 被设计为一个基于优先级的抢占式多任务内核,提供了任务调度、任务间同步与通信和定时管理等功能,为多任务的嵌入式 Internet 应用提供了基本的系统服务。为实现嵌入式设备与 Internet 的互联,WebitX 提供了一个嵌入式 TCP/IP 协议栈,实现了嵌入式 Web 服务器功能。该协议栈为传统设备提供了网络接口,人们可以利用现有的公共通信网络,通过 Web 对网络中各个节点的现场设备进行远程监控和维护。WebitX 的各个模块为独立开发,以静态库函数的形式向用户提供了 Socket API 编程接口,能够为基于 8 位微控制器的嵌入式 Internet 应用提供实时、可靠的支持。

参考文献

- 1 Ramamritham K, Stankovic J A. Scheduling algorithms and operating systems support for real-time system[J]. Proceeding of the IEEE, 1994, 82(1): 55~67
- 2 赵海. 嵌入式 Internet—21 世纪的一场信息技术革命[M]. 北京: 清华大学出版社, 2001. 158~167
- 3 Stallings W. Operating System Internals and Design Principles [M]. Prentice Hall, 1994. 56~78
- 4 赵海. 现场总线网络通信模型中的逻辑链路层[J]. 东北大学学报(自然科学版), 1996, 17(6): 230~234
- 5 Liu C L, Layland J W. Scheduling algorithms for multiprogramming in a hard real-time environment[J]. Journal of ACM, 1973, 20(1): 44~61
- 6 Audsley A C, Burns A, Wellings A J. Deadline monotonic scheduling theory and application [J]. Control Engineering Practice, 1993, 1(1): 71~78
- 7 Swaminathan V, Chakrabarty K. Real-time task scheduling for energy-aware embedded systems [J]. Journal of the Franklin Institute, 2001, 338: 729~750
- 8 Forouzan B A, Fegan S C. TCP/IP Protocol Suite [M]. Beijing: Tsinghua University Press, 2000. 147~299
- 9 Clark D D, Jacobson V, Romkey J, Salwen H. An analysis of TCP processing [J]. IEEE Communion Magazine, 1989, 27(1): 23~29