

Linux 运用于嵌入式系统的技术分析

王亚军^{1,2}, 刘金刚¹

(1. 首都师范大学 & 中国科学院计算所 计算机科学联合研究院, 北京 100037; 2. 中国人民武装警察部队学院, 河北 廊坊 065000)

摘要: Linux 和嵌入式系统都是计算机学术领域中的技术热点, 两者的结合, 为它们开辟了新的发展空间, 也带来了许多新的技术问题。针对 Linux 运用于嵌入式系统所产生的技术问题及解决方案, 作了详实的阐述。

关键词: Linux; 嵌入式系统; 嵌入式 Linux 系统

中图分类号: TP316 文献标识码: A 文章编号: 1001-3695(2005)05-0102-03

Technology Research on Linux Applied in Embedded System

WANG Ya-jun^{1,2}, LIU Jin-gang¹

(1. *Join Research Academy for Computer Science, Capital Normal University & Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100037, China;* 2. *The Chinese Peoples Armed Police Forces Academy, Langfang Hebei 065000, China*)

Abstract: Linux and Embedded System are both hot points in the academic field of computer technology. The combination of them two further expands their respective development area. On the other hand, some new technology problems are also produced. Focusing on the application of Linux on the embedded system, this paper goes into a little more detail about these problems and their solutions.

Key words: Linux; Embedded System; Embedded Linux System

自 1991 年至今, 开放源代码的 Linux 操作系统在很多方面已经赶上甚至超过了商用的 UNIX 操作系统, 其在全球计算机产业界中掀起的巨澜, 是从前任何一个操作系统所无法比拟的。从台式个人机到大型服务器, Linux 都表现出稳定、高效、健壮以及功能齐全等优越性能, 成为当前最流行的操作系统之一。同时, 随着后 PC 时代的来临, 信息家电的兴起, 作为信息家电的核心技术, 嵌入式系统也获得了广阔的应用空间和发展前景。将 Linux 应用于嵌入式系统, 开发嵌入式 Linux 系统, 已经成为嵌入式系统发展的重要分支。而两者的融合, 却带来了许多技术上新的问题。

1 嵌入式系统转向 Linux

按照英国电气工程师协会 (IEE) 所下的定义: 嵌入式系统是用来控制或监视机器、装置或工厂等的大规模系统的设备。嵌入式系统是电脑软件和硬件的综合体, 它是以应用为中心, 以计算机和信息技术的发展为基础, 将用户所需的特定功能嵌入到产品、装置或大型系统中, 软硬件可裁剪, 从而能够适应实际应用中对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。嵌入式系统的特点主要表现在以下几个方面:

(1) 嵌入式系统是将先进的微处理器技术、超大规模集成电路技术、网络通信技术以及嵌入式软件技术与各个行业的具体应用相结合的产物。

(2) 嵌入式系统是面向用户、面向产品、面向特定应用的。从系统开发到升级换代, 嵌入式系统始终与具体应用有机地结

合在一起。

(3) 嵌入式系统的嵌入特性意味着这些系统本身与其所控制和管理的对象是融为一体的, 是其中的一个有机组成部分。

(4) 嵌入式系统中的 CPU 具有低功耗、体积小、集成度高等特点, 能够把通用 CPU 中许多由板卡完成的任务集成到芯片内部, 从而有利于整个系统设计趋于小型化。

(5) 嵌入式操作系统所使用的技术, 基本上是从台式计算机操作系统演变过来的。由于应用的需要和硬件条件的限制, 嵌入式操作系统必须重视效率, 针对具体需求, 去除冗余, 才能达到理想的性能。

嵌入式应用对操作系统的要求是: 功能专一而高效, 高度节约资源, 启动速度快。在嵌入式系统应用方面, Linux 的内核完全可以裁减到 100KB 以下。Linux 的内核精简而高效, 可修改性强, 支持多种体系结构, 而且具有非常好的网络性能, 这些优点使得它正适合于嵌入式开发和应用。另外, 各种商业嵌入式操作系统价格昂贵, 使得生产嵌入式产品的成本急剧上升。相比之下, 免费授权的 Linux 支持 GNU 公共许可证 (Public License), 并公开其源代码供所有人任意获取和免费使用。

2 Linux 应用于嵌入式系统

Linux 是一种类 UNIX 操作系统, 它不仅具有 UNIX 操作系统的优点, 而且还有它自身的优势。概括起来, Linux 的特点主要表现在以下几个方面:

(1) 多用户支持。Linux 支持多用户模式, 允许多个用户同时登录系统。

(2) 多任务支持。Linux 支持多任务模式, 允许多个进程并发执行, 从而极大地利用了 CPU 资源。

(3) 可移植性强。Linux 支持多种硬件平台, 并且兼容 POSIX 1003.1 标准。与国际标准接轨使得 Linux 在界面上具有很强的通用性。

(4) 真正的 32 位操作系统, 支持虚拟内存和共享库, 并且使用处理器提供的内存保护模式来保证内核程序的安全可靠。

(5) 强大的网络功能。Linux 系统在内核中集成了大量的网络应用程序, 支持 TCP/IP, PPP 等众多的网络协议, 并且在强大的网络需求下表现出优良的健壮性。

(6) 源代码开放, 可修改性强。由于 Linux 是一个代码开放的操作系统, 并且模块性比较强, 任何人都可以根据实际需要对其进行修改和裁减, 从而为 Linux 在嵌入式系统中的广泛应用创造了条件。

尽管 Linux 性能优越, 但是它必定不是专为嵌入式应用开发的操作系统, 当把这种操作系统应用于嵌入式领域时, 还必须解决许多技术问题。首先, Linux 的内核为非抢占式的, 不能通过改变优先级来影响内核当前的执行流程。这样, Linux 内核可以对某些重要的数据结构进行修改而不加任何保护措施, 因为不担心被其他程序抢占。但如此一来, 实时任务也就不能得到保证, 从而不能满足许多嵌入式系统所要求的实时性。其次, Linux 提供了强大的虚拟内存管理功能, 这也是现代操作系统的重要特征, 但是由于嵌入式系统的硬件资源有限, 有的系统甚至没有内存管理单元, 因此还需要重新编写这一部分的内核代码。另外, Linux 为了追求代码的高效, 而采用了单内核结构, 这是以损失系统的结构清晰为代价的, 从而为代码的修改带来了不便。但是, 业界已经开发出许多嵌入式 Linux 系统, 针对 Linux 运用于嵌入式领域所涉及的技术问题提出了宝贵的解决方案。

3 主流嵌入式 Linux 系统的解决方案

(1) μ Clinux 的嵌入式方案。Lineo 公司的拳头产品 μ Clinux 是开放源代码的嵌入式 Linux 的典范之作。 μ Clinux 设计的目标平台是那些不具有内存管理单元 (Memory Management Unit, MMU) 的微处理芯片。为了达到嵌入式系统的需求, μ Clinux 不得不重写部分内核代码, 而且对原标准 Linux 中不再用到的代码, 采用预处理宏的方法加以屏蔽。另外, μ Clinux 采用 Romfs 文件系统, 这种文件系统相对于一般的 EXT2 文件系统需要更少的空间。空间的节约来自两个方面: 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, 在建立超级块时需要更少的存储空间。 μ Clinux 小型化的另一个做法是重写了 C 函数库, 相对于越来越大且功能越来越全的 Glibc 库, μ Clibc 对 Libc 作了精简, 这也是嵌入式系统的通常做法。

(2) Embedix 的嵌入式方案。Lineo Embedix 是一种嵌入式 Linux 软件解决方案。Lineo 将实时技术集成到 Embedix 软件开发工具包 (SDK) 中, 以此为嵌入式设备提供软件和硬件的实时能力。Embedix SDK 的一个关键特性是包含了 Target Wizard 库编辑工具。这是一种自动操作方式, 可以快速、简便地选择目标映像的所有需要, 为嵌入式设备快速建立尽可能小的软件

映像。另外, Windows 下的 Embedix SDK 为开发者提供了虚拟的开发环境, 允许 Windows 开发者在他们熟悉的 Windows 环境下开发嵌入式 Linux 应用软件。

(3) RTLinux 的嵌入式方案。RTLinux 是世界上最早的实时 Linux 系统, 是针对 Linux 系统缺少实时性的一个非常成功的解决方案。RTLinux 并没有对 Linux 内核做大的改动, 而是利用 Linux 内核模块机制, 采用插入模块的方式, 通过一个独立的内核来管理实时任务, 如图 1 所示。在加载了 RTLinux 实时内核之后, 原来的 Linux 内核就作为实时操作系统的空闲任务, 仅当没有实时任务要运行时才执行。RTLinux 使用的关键技术是中断控制硬件的软件模仿。无论 Linux 正在做什么, 无论 Linux 是否在核心态运行, 无论 Linux 是否禁止中断, 也无论 Linux 是否被锁住了, 实时系统总是能响应中断, 从而保证了系统的实时性。

(4) RTAI 的嵌入式方案。RTAI (Real-Time Application Interface) 是一套在 Linux 上用于编写实时应用程序的接口。如图 2 所示, 它的实现机制与 RTLinux 非常相似, 唯一不同的是 RTAI 在 Linux 上定义了一个实时硬件抽象层 RTHAL (Real-Time Hardware Abstraction Layer), 并针对 RTAI 开发了 LXRT (Linux-RT), 让 RTAI 可以使用 Linux 本身的系统调用功能。RTHAL 的作用是使 RTAI 能够在实时任务需要运行的任何时刻中断 Linux, 从而使 Linux 任务总是作为优先级最低的任务来运行, 并且只有在没有实时任务时才可以运行。作为 Linux 的实时功能的扩展, RTAI 已经相当稳定和成熟。

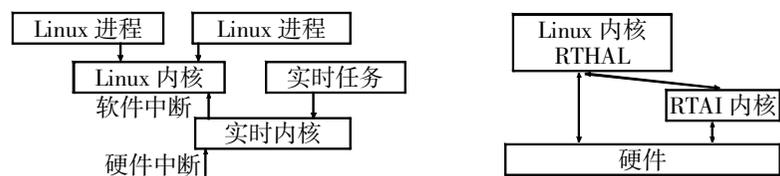


图 1 实时内核与 Linux 内核的关系

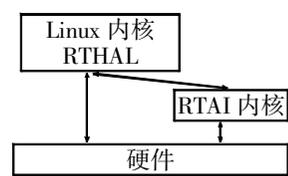


图 2 RTAI 的实现原理

(5) MontaVista Linux 的嵌入式方案。MontaVista Linux 是针对嵌入式设备量身定做的、实时的、专业的嵌入式操作系统。MontaVista Linux 在嵌入式系统市场处于领先地位的原因之一, 就是它能够基于 Linux 提供很好的实时解决方案。在 Linux 中存在三个不利于实现实时性的问题: 中断封锁时间过长; 非抢占式的 Linux 内核; 耗尽式的、机会均等的进程调度策略。

针对这三个问题, MontaVista Linux 提出了下列解决方案: 完全的抢占性内核, 而且对 Linux 的 API 不作任何改变。透明的实时调度器。如图 3 所示, MontaVista Linux 实时调度器在 Linux 调度器之前运行, 然后由它来管理有实时要求的进程。如果不存在有实时要求的进程, 那么 MontaVista Linux 实时调度器会十分有效地将控制权转交给标准 Linux 调度器。

(6) ETLinux 的嵌入式方案。ETLinux 对 Linux 实现嵌入式改造的思路是: 通过修改内核以及应用程序来减少对内存空间的需求。通过对普通 Linux 内核的配置和修改源代码, 它一方面减少内核数据结构的大小, 另一方面去除不需要的功能。最大的空间节省来自于将控制台代码从内核中删除, 节省了大约 90KB 空间。此外, 一个 ELF 文件格式的可执行程序总是具有一个复杂的文件头, 用来在将控制权转给 main 函数之前准备运行环境。由于每个可执行文件都包括一份类似信息的复

制, 这样的头文件被复制了多次, 占用了大量宝贵的磁盘空间。ETLinux 的解决方案是采用一种解释性语言 Tcl 来作为驱动器, 把许多基本功能放在解释器的可执行程序中, 使它们成为语言的原语, 然后将脚本作为应用程序来运行。由于 Tcl 是一种轻量级解释性语言, 具有非常简单的语法, 因此非常适合于作为嵌入式系统的命令行界面。

4 建立嵌入式 Linux 系统的技术实现

在分析业界主流嵌入式 Linux 系统解决方案的基础之上, 下面着重阐述针对具体应用建立嵌入式 Linux 系统的方法。一般来说, 可以把建立嵌入式 Linux 系统分成三个方面: 根据需要修改 Linux 的源代码并编译一个 Linux 内核; 编写自己系统的脚本; 建立一个尽量小的文件系统。

(1) 修改并编译 Linux 内核源代码。要裁减 Linux 内核源代码, 必须首先精通 Linux 的内核原理和体系结构。Linux 内核有自己的结构体系, 其中进程管理、内存管理和文件系统是其最基本的三个子系统, 如图 4 所示。

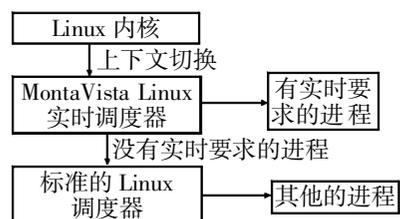


图 3 MontaVista Linux 实时调度器

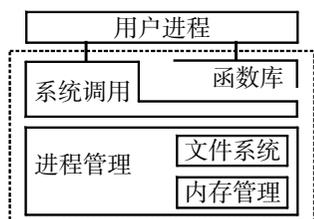


图 4 Linux 内核的结构

在图 4 中, 可将虚线部分看成是 Linux 内核的单内核结构。用户进程可直接通过系统调用或函数库来访问内核资源。单内核结构注重的是代码的效率和功能的强大, 但这是以操作系统结构清晰的损失为代价的。正因为 Linux 采用了单内核结构, 因此修改内核时必须注意各个子系统之间的协调。修改 Linux 源代码主要有两方面原因: 首先, 嵌入式系统硬件资源有限, 这就决定了嵌入式操作系统必须针对具体需求, 量体裁衣, 才能达到理想的性能。其次, Linux 操作系统本身的某些特征不能满足嵌入式系统的要求, 必须对这些特征进行修改。

综合分析可知, 需要在文件系统、内存管理和进程调度上来改动源代码。实际情况表明, EXT2 文件系统虽然在组织文件上有其自身的优势, 但嵌入式系统往往不需要额外的功能, 如果能对 EXT2 文件系统部分自定义, 将大大节省外存空间。另外, 许多嵌入式系统要求具备实时性, 而 Linux 系统不具有实时性。导致 Linux 内核实时性不强的原因之一在于它使用了虚拟内存的管理机制。虚存管理使得同时运行的进程数目更多一些, 却降低了 CPU 的突发处理能力。因此屏蔽内核的虚存机制最能改善其实时性。另外, Linux 内核为非抢占式的, 采用机会均等的调度策略, 所以实时任务得不到保证。采用抢占式内核和面向实时的调度策略, 也是提高内核实时性的重要途径。

(2) 嵌入式 Linux 系统中的脚本控制。Linux 系统在启动和退出时使用了大量的脚本, 用于启动或者停止服务进程。普通 Linux 系统从启动到字符登录界面的出现会花费很长的时间, 这对于嵌入式产品的用户来讲是不必要的。为了实现系统的快速启动和快速退出, 最好的方法就是重新编写系统的启动和退出脚本。要缩短 Linux 系统的启动时间, 必须先清楚

Linux 系统的启动过程。系统启动后, 首先执行 BIOS 程序进行系统自检, 包括检查内存、硬盘和光驱等, 这段时间是由硬件来限制的。接下来, BIOS 读取主引导记录 (MBR), 并将其装入内存。MBR 上一般装载了 LILO (Linux Loader)。LILO 先装入 Linux 内核映像, 然后内核的启动代码将获得 CPU 控制权, 并依次运行 bootsect. s, head. s 和 setup. s 等代码。这一部分的时间相对于整个系统的启动时间来说可以忽略不计, 也不需要在这部分代码作任何修改。

32 位启动代码 setup. s 运行完后, 内核执行 start_kernel() 程序, 首先检查设备并初始化驱动程序, 然后启动第一个用户进程 init。进程 init 根据 /etc/inittab 决定进入哪个运行级别, 并根据运行级别执行相应的脚本来启动特定服务, 这些脚本包括 /etc/rc.d/rc. sysinit 和 /etc/rc.d/rcN. d。脚本运行完成后由进程 init 启动 getty 进程给出登录提示。要真正加快这一部分的启动时间, 就要重新编写 rc. Sysinit, rc 等启动脚本, 让它只作一些必要的初始化工作及服务的启动。修改以后, 从内核开始启动到字符登录界面的出现可以缩减到 7s ~8s 的时间。

Linux 的退出过程相对要简单一些。一般来说, 退出时 Linux 运行 /etc/rc6. d 中的脚本终止服务进程, 其中最重要的脚本是 S01reboot, 实际上执行这个脚本就可以终止系统的所有服务进程, 但是为了可靠地终止系统服务, 需要逐项检查。自定义退出脚本时, 可以屏蔽那些为可靠终止服务而编写的脚本, 直接让 S01reboot 负责终止所有进程。对 S01reboot 脚本的修改主要是除掉与应用无关的语句。

(3) 组建嵌入式 Linux 的文件系统。它主要有两方面的原因: 首先现有的 Linux 文件系统的全集实在太太大; 其次, 需要改写现有 Linux 系统中的配置文件以满足应用程序的要求。由于系统启动所需的脚本以及系统的命令文件都在文件系统内, 所以具有文件系统是 Linux 系统启动的基础。EXT2 (或 EXT3) 不仅是 Linux 默认的文件系统, 而且也是嵌入式 Linux 系统中使用最广的文件系统。修改 EXT2 文件系统实际上是要剔除 EXT2 文件系统在 EXT 以及 MINIX 文件系统上的改进。MINIX 文件系统规定的文件名不许超过 14 个字符, 而在嵌入式应用场合, 文件的种类少且容量小, 14 个字符已经足够区分文件名之间的差别。由于 Linux 是单内核结构, 因此修改文件系统时, 至少要考虑到每一个进程都可能要打开的文件列表, 进程创建、运行和结束这一系列过程都与文件系统有牵连, 因此相应的代码也要修改。另外, 内存管理与交换文件有关, 执行交换功能的内存子系统函数也要修改。

至此, 我们已经获得了内核映像, 并且组建了文件系统, 嵌入式 Linux 系统的设计基本完成, 但是还需要花大量的时间进行测试以保证它的稳定性、可靠性和安全性。

5 结束语

Linux 在嵌入式环境中的应用, 是继 Linux 在桌面端的应用和在大型服务器上的应用之后, Linux 开辟的另一个重要的应用领域。嵌入式技术具有广阔的应用前景, 渗透于社会生活的诸多领域。把 Linux 应用于嵌入式系统, 把 Linux 自身固有的优越性融入嵌入式技术, 是嵌入式技术发展的重要方向。(下转第 128 页)

以及信息数据的安全,需要在各方面采取措施。如为了保证信息用户的相互信任,采用证书签名认证的策略;对于 UDDI 注册中心的安全,主要采取的是外包一层加以验证的方式来保证安全;对于信息数据的安全,主要采用对 SOAP 消息进行加密的技术;而为了进一步地保障个人隐私,则采用利用安全规则加以过滤的原则。图 2 给出了一个信息共享平台的安全结构。

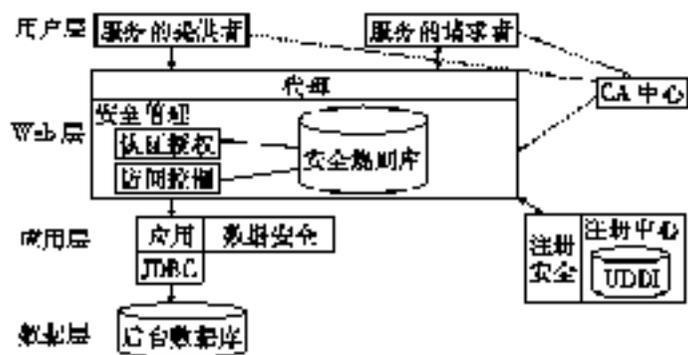


图 2 电子政务信息共享平台的安全

该系统的注册中心是由 Win2003 自带的私用 UDDI 完成的,SSL 保护这个 UDDI 的服务发布的安全。服务的发布者想要发布服务,首先要通过 SSL 的数字证书的认证。因此,系统同时用一台 Win2003 作为 CA 的验证中心(暂时),提供数字证书的认证授权。用户层运行于 Java 的环境下,使用 SUN 公司的 JWSDP 作为运行工具。

一个用户,如服务的发布者在取得 CA 的授权后,可以通过网页形式或者 Java 程序向注册中心发布服务。比如发布者开发了一个读取数据库中某段数据的 JDBC 连接数据库的 EJB 应用,并部署在 Tomcat 的服务器上,然后向 CA 获取证书,调出注册中心的网页登记了这个服务的描述以及 tModel 接口,接着填写了这个服务对应的安全规则。另外一个用户,即服务的请求者,通过 Java API 向代理发送服务的请求。比如他欲调用前述服务的 EJB 应用进行数据库的读取操作,这时调用安全管理模块,首先判定其有否进行查询服务的授权,然后再判定其访问控制的权限。访问控制要从访问控制列表取得其查询服务范围,如果通过,则向 UDDI 查询所需要的服务;服务如果找到,则对比请求者的角色与根据规则库中的运算得到的角色是否等价;如果等价,则意味着其拥有调用该服务的权利。根据 Web 服务的描述和 tModel 接口以及运用 JWSDP 的 Java API,可以绑定和调用相应的服务。

在应用层,为了进一步保证数据的安全,除了这个调用 JDBC 的 EJB 外,还给出了一个 [JP3] 数据安全模块。这个模块可以在 [JP]

(上接第 104 页)

参考文献:

[1]ayne Wolf. Computers as Components Principles of Embedded Computing System Design [M]. Beijing: Publishing House of Machinery Industry, 2002.

[2]Arnold Berger. Embedded Systems Design an Introduction to Processes, Tools and Techniques [M]. Beijing: Publishing House of Electronics Industry, 2002.

[3]Jean J Labrosse. Embedded Systems Building Blocks (2nd Edition) [M]. Beijing: Publishing House of Machinery Industry, 2002.

[4]李善平,刘文峰,王焕龙,等. Linux 与嵌入式系统 [M]. 北京:清华大学出版社, 2003.

EJB 读取数据库时,过滤掉重要的需要保密的数据。这么做加大了数据保护的力度,从而使信息更安全。

5 结束语

电子政务的安全体系包含有实体的安全、系统的安全、安全服务以及应用的安全。本文中所讨论的安全技术没有涉及实体的安全以及系统的安全,只是研究了应用安全以及安全服务中包含的安全访问、身份认证、数据的加密等问题。由于电子政府办公的特殊性,电子政务的实现是以安全为第一位的。电子政务中信息共享的安全一直都是实现政务集成的焦点问题之一,本文从描述信息共享平台开始,讨论了组成平台的三个方面的安全问题,最后给出了基于本文思想的一个信息共享平台的基本结构。

参考文献:

[1]蔡谊,沈昌祥. PKI 技术在电子政务中的应用 [J]. 计算机应用研究, 2002, 19(10): 11-16.

[2]周卫华,马如军,谭成翔. WebLogic Server 安全技术概述及其在电子政务应用中的实现 [J]. 计算机应用研究, 2003, 20(9): 79-81.

[3]任小林,桂侍伟,吴祈宗. 基于 XML 的 Web 信息发布系统及其 JEE 实现 [J]. 计算机应用, 2003, 23(10): 134-137.

[4]翁琦,沈志宏,陈秀敏,等. 电子政务系统中的访问控制研究 [J]. 计算机工程与应用, 2002, 38(3): 247-250.

[5]申德荣,于戈,王国仁,等. 一个基于 Web Service 的支持虚拟企业 B2B 的使能系统 [J]. 小型微型计算机系统, 2003, 24(9): 1621-1625.

[6]申德荣,于戈,宋宝燕,等. 虚拟企业内共享资源的 E-UDDI 管理机制 [J]. 计算机集成制造系统, 2003, 9(2): 90-95.

[7]Enterprise UDDI Services: An Introduction to Evaluating, Planning, Deploying, and Operating UDDI Services [R]. Microsoft Windows Server 2003 (White Book), 2003.

[8]张勇,冯玉才. XML 数字签名技术及其在 Java 中的具体实现 [J]. 计算机应用, 2003, 23(9): 93-95.

作者简介:

欧毓毅 (1974-), 女, 湖北人, 博士研究生, 研究方向为电子商务和电子政务的分布式网络信息系统的集成与安全、基于代理的集成电子服务等; 郭荷清 (1965-), 女, 江苏人, 教授, 博士生导师, 研究方向为网络信息系统的集成与安全; 尹建飞 (1974-), 男, 广东人, 博士研究生, 研究方向为电子商务和电子政务的分布式网络信息系统的集成与安全。

[5]探砂工作室. 嵌入式系统开发圣经(第2版) [M]. 北京: 中国铁道出版社, 2002.

[6]王学龙. 嵌入式 Linux 系统设计与应用 [M]. 北京: 清华大学出版社, 2002.

[7]毛德操, 胡希明. Linux 内核源代码情景分析 [M]. 杭州: 浙江大学出版社, 2002.

[8]彭晓明, 王强. Linux 核心源代码分析 [M]. 北京: 人民邮电出版社, 2002.

作者简介:

王亚军 (1977-), 男, 讲师, 硕士研究生, 研究方向为嵌入式 Linux 系统设计; 刘金刚, 男, 教授, 博士生导师, 研究方向为单片机与智能接口技术。