

嵌入式 Linux 解决方案关键技术研究^{*}

Key Technologies of Embedded Linux

卢 凯 吴庆波 卢锡城

(国防科技大学计算机学院 长沙410073)

Abstract For the low developing cost, full of applications, configurable and API compatible with the General Linux, the embedded Linux has become an important branch of Embedded Operating Systems. In this paper, we introduced some key technologies of the development of Embedded Linux, such as run environment, GUI, developing tools and other technologies, etc.

Keywords Embedded Operating System, Linux

一、引言

“嵌入式操作系统解决方案”是指以嵌入式操作系统为核心的包括操作系统、运行环境、开发工具、高层应用等运行于嵌入式计算机设备上的所有软件系统。嵌入式系统工作环境特殊,故该解决方案必须具有体积小、占用资源少、实时性强、可靠性高等特点。目前典型的嵌入式操作系统解决方案有 pSOS^[1]、Vx-Works^[1]、QNX^[2]等。随着嵌入式系统硬件技术的发展和应用需求的不断提高,嵌入式 Linux 技术由于其通用化、网络化正成为当前嵌入式操作系统领域一个新的发展点^[3]。现在,已有许多厂商推出了基于嵌入式 Linux 的解决方案,还有许多老牌嵌入式操作系统厂商也纷纷实现其产品与嵌入式 Linux 的兼容性^[2]。可以说,嵌入式 Linux 解决方案在一定程度上将成为未来嵌入式解决方案的一个重要发展方向。

本文将介绍嵌入式 Linux 解决方案的若干关键技术,具体包括内核技术、运行环境技术、GUI 技术和开发工具技术等。

二、嵌入式 Linux 的特点

Linux 是遵循 GPL^[4]规则的多任务操作系统,源码公开,价格低廉,功能强大,应用丰富。嵌入式 Linux 是指在标准 Linux 基础上经过裁剪、修改,适用于嵌入式系统的专用 Linux 系统。

Linux 经裁减能应用到嵌入式设备的原因主要有以下几个:首先,Linux 本身具有很好的模块化和可裁

剪性,可根据要求生成不同配置,非常适用于嵌入式环境;其次,嵌入式硬件设备技术大大提高,尤其是内存资源和固化存储器件资源(Flash、ROM 等)性价比迅速增长,为体积较大的 Linux 应用于嵌入式系统提供了可能;再次,用户对嵌入式应用需求多样化,尤其是网络化,为 Linux 技术应用于嵌入式设备提供了源动力;最后,也是最重要的,嵌入式 Linux 与标准 Linux 有很好的兼容性,基于标准 Linux 开发的设备驱动程序、网络服务和应用等都适用于嵌入式 Linux,故基于嵌入式 Linux 解决方案的产品从研发到上市时间短,研发人员少和研发经费低。而采用 VxWorks、pSOS 等,任何应用不是自己开发就是购买,存在开发周期长、开发成本高等缺陷。

目前,嵌入式 Linux 解决方案已成为许多手机、机顶盒、PDA 等嵌入式设备厂商的首选。

三、嵌入式 Linux 关键技术

完整的嵌入式 Linux 解决方案应包括嵌入式 Linux 操作系统内核、运行环境、图形化界面和应用软件等,应用和系统开发者还需要嵌入式 Linux 开发环境。由于嵌入式设备的特殊要求,嵌入式 Linux 解决方案中的内核、环境、GUI 等都与标准 Linux 有很大不同,其主要挑战是如何在狭小的 FLASH、ROM 和内存中实现高质量的任务实时调度、图形化显示、网络通信等功能。

3.1 内核技术

嵌入式设备的 CPU 平台多种多样,目前常用的有

^{*} 本课题受863支持。卢 凯 博士,主要研究方向是嵌入式操作系统。吴庆波 副教授,主要研究方向是嵌入式操作系统。卢锡城 工程院院士,主要研究方面是计算机体系结构和分布处理。

ARM 系列、MIPS 系列、Power PC 系列、M68K 系列和 X86 系列等。多平台的异构性要求嵌入式操作系统有较好的平台移植性。专用嵌入式操作系统一般通过硬件抽象层(HAL)来满足该需求。Linux 的结构不适合采用 HAL 层,虽然无 HAL 层,但经过广大 Linux 爱好者的多年努力, Linux 已可支持包括 X86、Alpha、MIPS、ARM、M68K 等常见的多种 CPU 系列,可充分满足用户多平台移植的需求。

嵌入式 Linux 内核一般由标准 Linux 内核裁剪而来。用户可根据需求配置系统,剔除不需的服务功能、文件系统和设备驱动,经过裁剪、压缩后的系统内核一般只有 300k 左右,十分适合嵌入式设备,图1显示了针对 PDA 设备的支持 Micro Window 的嵌入式 Linux 内核配置参数。

CONFIG_EXPERIMENTAL=y
CONFIG_1GB=y
CONFIG_BLK_DEV_LOOP=y
CONFIG_BLK_DEV_INITRD=y
CONFIG_UNIX=y
CONFIG_VT_CONSOLE=y
CONFIG_PSMOUSE=y
CONFIG_MINIX_FS=y
* CONFIG_VIDEO_SELECT=y
* CONFIG_FB_VESA=y
* CONFIG_FBCON_CFB32=y
CONFIG_M586=Y
CONFIG_NET=y
CONFIG_BLK_DEV_RAM=y
CONFIG_PACKET=y
CONFIG_VT=y
CONFIG_MOUSE=y
CONFIG_AUTofs_FS=y
* CONFIG_VGA_CONSOLE=y
* CONFIG_FB=y
* CONFIG_VIDEO_SELECT=y
* CONFIG_FONT_8x8=y

(注: * 为支持 Micro Window 必备的选项)

图1 针对 PDA 设备的嵌入式 Linux 内核配置参数

同标准 Linux 不同的是嵌入式 Linux 必须要实现从 FLASH 或 ROM 的启动。标准 Linux 启动代码 LILO 实现了系统初始化和从软盘、硬盘 0 扇区引导内核。嵌入式 Linux 一般保存在 FLASH 或 ROM 中,标准 LILO 无法引导。在支持直接从 FLASH 设备引导的系统中,如 CL-PS7500 开发版,引导程序主要完成对硬件系统的初始化工作和操作系统的解压、移位工作。在不支持直接从 FLASH 引导的系统中,如 AMD 的 SC520 开发版,FLASH 设备只能作为非引导磁盘使

用。此时,可采用先从硬盘或软盘加载一个小操作系统,如嵌入式 DOS,然后再执行“Loadlin”加载程序从 FLASH 引导嵌入式 Linux。

对标准 Linux 内核的修改主要集中在虚存管理和实时化任务调度模块。由于许多嵌入式系统无硬盘设备,不必采用虚存管理。为减少系统尺寸,有的嵌入式 Linux 取消了虚存管理模块,例如主要用于网络路由器的 uCLinux^[6] 系统等。但采用虚存管理的好处是由于有了地址保护机制,不会产生因地址访问越界而导致全系统崩溃的问题。

某些嵌入式应用有强实时特性,如手机等。嵌入式 Linux 同标准 UNIX 内核一样,内核运行不可剥夺,其中断响应时间不定,无法满足强实时需求。针对此缺陷,有的嵌入式 Linux 修改了任务调度模块,如 Hard Hat Linux^[5] 等。此类系统的改进主要是在内核和设备驱动程序中加入了许多切换点,在该点处,系统检测是否存在未处理的紧急中断,有则剥夺内核的运行,及时处理中断。实现实时性服务的另一个方法是采用双内核技术,如 RTLinux^[7] 等。RTLinux 中有两个内核:标准 Linux 内核和实时性内核。标准 Linux 内核作为一个任务运行于实时内核之上,强实时性应用则直接运行在实时内核上。图2显示了 RTLinux 的结构。平时, Linux 内核运行普通应用,有实时性任务时,实时接管全系统运行实时性应用。RTLinux 实际上是两个操作系统的结合,虽然实现了实时性服务,但作为嵌入式操作系统存在体积大、复杂等缺陷,不宜采用。

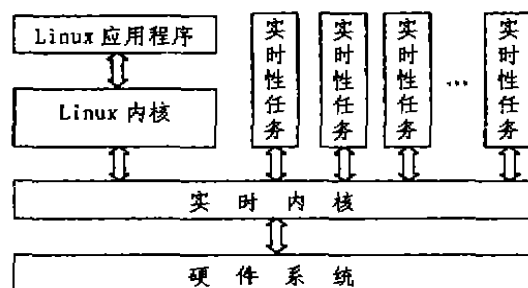


图2 RTLinux 的双内核结构

标准 Linux 内核虽无法提供强实时服务,但由于目前 CPU 处理速度很高,并且大多数应用对实时性要求较弱,如 STB、PDA 等,因此嵌入式 Linux 依然可满足大多数应用需求,适用于大多数嵌入式设备。

文件系统是嵌入式 Linux 操作系统必不可少的。目前,由于不存在硬盘,嵌入式设备文件系统主要使用了 RamDisk 技术和网络文件系统技术。RamDisk 可驻留于 Flash,运行时加载到内存中,为减少体积, RamDisk 可压缩存储,非常适用于嵌入式环境。

RamDisk 的缺省尺寸是4M,可满足大多数应用需求。Linux 系统支持对压缩 RamDisk 的自动展开,但此时 RamDisk 只能使用 EXT2 或 MINIX 文件系统。网络文件系统可使用在有网络的嵌入式设备中。嵌入式设备启动后从网络下载整个文件系统。

3.2 运行环境技术

运行环境指用户运行任何应用的基础设施,主要包括函数库和基本命令集等。

Linux 系统同时向用户提供了静态和动态函数库,静态函数库在生成应用时直接链接到用户应用中,动态库在应用运行时才链接。由于嵌入式系统应用一般都是在开发平台上预先生成的,因此嵌入式系统只需向应用提供动态函数库。Linux 应用运行所需的函数库主要有 C 库、数学库、线程库、加密库、网络通信库等,其中 C 库是最基础的函数库,标准 GNU 动态 C 库有 4MB^[1];采用静态库生成的应用体积十分庞大,无法运行于嵌入式系统中,所以针对应用定制的嵌入式 C 库是嵌入式 Linux 解决方案必须提供的。

定制 C 库一般都是基于 GNU C 库的标准 API 设计的。有的嵌入式 C 库在 GNU C 库的基础上直接删减而成,例如 CC-Linux 的 C 库。采用此方案的优点是实现简单,但体积依然过大。有的嵌入式 C 库遵照 GNU 标准,采用全新结构的重新构造,同时满足了体积小、功能全等要求,例如 Linero 的嵌入式共享 C 库^[2]和国防科大研制的 Nelix 系统嵌入式 C 库等。Nelix 嵌入式 C 库主要在系统结构、函数数量、函数功能、函数实现方式等方面较 GNU 标准 C 库进行了较大改动。(1)在系统结构方面,为满足动态可配置要求,Nelix C 库采用了简单的两级目录树结构,分为系统调用、函数库两大部分,一个函数一个文件,生成时根据配置文件自动确定所需函数;(2)在函数数量方面,标准 C 库提供了相似语义的多种函数,而 Nelix 只提供其中最基础的,例如执行函数,标准 C 库有 execl,execle,execlp,execvp;而 Nelix 只有 execl 和 execvp。这虽然要求用户基于该基础函数适当修改应用,但却对减少函数库体积十分有利。(3)在函数功能方面,Nelix C 库省略了一些不常用的或可替换的功能,例如打印函数中的浮点显示功能等。(4)在保证函数功能不变的前提下,Nelix 修改了函数实现方式,例如 printf 函数:标准 printf 涉及复杂的缓冲区管理、文件操作等,体积异常庞大;Nelx 则通过直接屏幕输出简单地实现了打印功能。采用 Nelix 嵌入式静态 C 库生成的“Hello, world!”应用仅 3.6kB,而采用标准动态 C 库生成的为 11k,标准静态 C 库的为 200k。

嵌入式系统中的 C 库以何种方式提供取决于系统应用的多少。应用少时,可采用静态方式将函数直接

链接到应用中,大大节省了其他无用函数的空间和共享函数库必须提供的临界区互斥功能,有利于减少系统整体尺寸。应用多时,采用共享库可减少函数多份拷贝的开销,节省系统空间。

基础命令集同样是运行用户应用的基础,主要包括初始化进程 init,终端获取 getty、Shell 和基本命令等。嵌入式系统的启动过程可能与标准 Linux 不同,例如跳过登录过程直接启动 GUI 等,这就要求修改 init、getty 等。标准 Linux 命令集同样由于体积问题无法直接应用于嵌入式环境。目前,小命令集的解决方法主要有集成方法和汇编方法两种。集成方法采用集成公共部分减少命令集整体体积,例如 BusyBox 中集成了许多典型命令,通过符号链接实现各命令。汇编方法则采用汇编编程减少每个命令的体积,如 Asmutility 等。集成方法因为是 C 实现,有较好的平台移植性;汇编方案的平台移植性较差,但体积很小。

3.3 GUI 技术

是否能提供友好、高效的 GUI 已成为评价嵌入式系统的一个重要指标。目前,pSOS、QNX 等嵌入式操作系统都已提供了 GUI 环境,嵌入式系统 GUI 同台式计算机的有显著不同:体积小和运行环境苛刻。目前,嵌入式 Linux 上的 GUI 主要有 Micro Window、Mini Window、Meta Window^[10]等。标准 Linux 的 Xfree86^[11]由于体积庞大,运行环境要求高,无法运行于嵌入式环境。

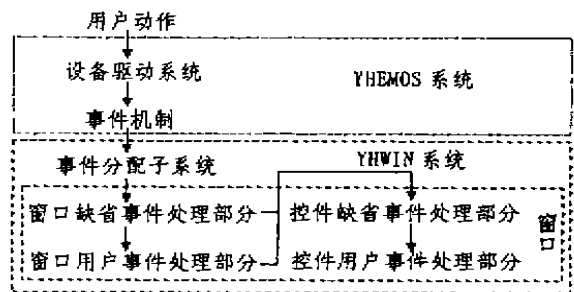


图3 EWIN 系统的结构

嵌入式 GUI 主要通过削减功能,降低性能实现体积小和占用资源少。目前嵌入式 Linux 上的 GUI 环境主要有两类:X 类和 Win32 类,X 类 GUI 分为服务方和客户方两方。服务器方提供鼠标、键盘处理和显示功能,客户方是用户应用,服务方和客户方之间通过 socket 接口和 X 协议通信。采用该方式十分有利于远程网络图形化服务,客户方和服务方可通过网络实现 X 协议和图形显示,典型的 X 类 GUI 有 Micro Window、TinyX 等,Win32 类的 GUI 不存在客户方和服务方。每个任务都自成一体,任务间的切换、事件分发由专门的

管理任务负责。图3显示了国防科大开发的嵌入式窗口系统 EWIN 的结构。

嵌入式 GUI 系统一般功能较标准 GUI 的弱, API 少, 编程困难, 这就要求提供开发模拟环境预先开发应用。TinyX^[11]同标准 Xfree86 有相似接口, 虽然体积较大, 但应用兼容性很好, 在一些存储资源不十分苛刻并且可以随时下载应用的系统中, 例如 STB, 有较大应用前景。但在 PDA 等设备中, 由于应用可预先定制, 专用的 EWIN、MetaWindow、MicroWindow 将占主要地位。

3.4 开发工具技术

开发工具在嵌入式系统开发中占有举足轻重的地位。嵌入式系统开发过程中的嵌入式 Linux 移植、设备驱动程序开发、C 库生成、应用生成等的进度都极大地依赖于开发工具的优劣。好的开发工具可极大地降低开发人员的工作强度, 减少错误, 加快开发进度, 甚至可以把不可能变成可能。嵌入式 Linux 的开发工具主要包括内核调试工具、系统生成配置工具、应用开发模拟环境等。

内核调试工具主要用于 Linux 内核的移植和设备驱动程序的开发, 目前常用的客户方调试工具有 GNU 的 gdb。gdb 通过串口直接支持远程的系统级调试。它具有断点设置、步进调试等功能。开发版一般自配有服务方开发调试工具, 例如 MIPS 系列的 PMON^[12]、ARM 开发版的 Angles^[13]和 JTAG^[14]等。PMON 和 Angles 都是软件调试工具, 它们运行于 CPU 上, 负责中断的截获和转发, 将 Linux 内核作为一个任务调试, 它们的断点设置、步进等都是通过软件方式实现的, 具有简单、易实现等优点, 但由于是软件方式, 因此存在系统崩溃时无法提供现场的缺陷, 在最需要调试工具的时候无法提供调试环境。JTAG 是硬件调试工具, 它直接访问内存, 具有高效、可靠等优点, 但需要开发版提供特殊的硬件设备。

开发内核和 C 库时, 针对应用的配置工具也十分重要。通过对不同应用的扫描、配置工具应能自动识别出应用所需的内核模块和 C 库的函数, 自动配置、生成适用的内核和 C 库或提供优化指导。目前, 可视化的配置工具, 尤其是内核配置工具已有很多成熟产品,

但扫描、配置工具还未见报道, 尤其是 C 库扫描、配置工具。

嵌入式 Linux 迅速发展的重要原因就是它同标准 Linux 有相似的接口, 所以嵌入式应用的开发可将标准 Linux 作为模拟环境, 充分利用标准 Linux 开发环境及其丰富的开发工具, 最后再移植到嵌入式系统中。嵌入式应用同嵌入式 Linux 内核和 C 库紧密相关, 嵌入式 Linux 内核和 C 库所能提供的服务决定了嵌入式应用同标准 Linux 应用的不同。移植就必须解决三者间的平衡问题: 究竟是修改应用, 还是增加 C 库和内核功能, 这取决于功能的必要性和体积的限度。

结束语 嵌入式系统解决方案是一个包括嵌入式操作系统和其他运行环境等各要素的系统。设计和实现完整的解决方案是一个复杂的系统工程, 要在多种因素间统筹考虑。嵌入式系统本身具有的异构性, 就决定了嵌入式 Linux 解决方案必然是多种多样的。本文介绍的是针对常见应用背景的嵌入式 Linux 解决方案中的各种关键技术。由于嵌入式 Linux 同标准 Linux 一脉相承, 其出身又在一定程度上决定了各方案间有很大的相似性, 这就注定嵌入式 Linux 技术将有良好的应用前景和后台支撑。可以预见, 嵌入式 Linux 技术将一统部分嵌入式系统的软件领域, 嵌入式 Linux 技术方兴未艾。

参考文献

- 1 www.rtos-link.com
- 2 www.qnc.com/products/os/qnxrtos.html
- 3 www.lem.org
- 4 www.linuxfrom.org
- 5 www.mvixta.com/products/cross_devkit.html
- 6 uclinux.lineo.org
- 7 www.reaktinelinux.org
- 8 www.gnu.org/software/libc
- 9 www.lineo.com
- 10 www.metagraphics.com/metawindow
- 11 www.xfree86.org
- 12 www.cirrus.com
- 13 www.amd.com
- 14 www.vcube.com