

# 嵌入式 Linux 的实时性改进技术

陈文星,张辉宜,陶 陶,周秀丽  
(安徽工业大学 计算机学院,安徽 马鞍山 243002)

**摘 要:**Linux 本身不是一个实时操作系统,但它具有源代码开放的特点。将其改造为一个实时的操作系统,已成为目前嵌入式系统应用领域的研究热点。文中详细介绍了广泛采用的几种将 Linux 改造为嵌入式 Linux 的实时性实现方法及其特点。同时阐述了操作系统实时性测试的几种方法,并对嵌入式 Linux 的发展趋势进行了展望。指出对一般基于 Linux 的嵌入式系统开发者而言,采用双内核法进行操作系统实时性改进是一种较好的方法。

**关键词:**嵌入式;操作系统;Linux;实时性

**中图分类号:**TP316.2

**文献标识码:**A

**文章编号:**1673-629X(2006)10-0114-04

## Amelioration Technology of Embedded Linux Real-Time Character

CHEN Wen-xing, ZHANG Hui-yi, TAO Tao, ZHOU Xiu-li  
(School of Computer, Anhui University of Technology, Maanshan 243002, China)

**Abstract:** Linux is not a real-time operation system, but its source code is open. So it has been the research focus in the field of the embedded system application that rebuilding Linux into a real-time operation system. This article has particularly introduced some broadly adopted method and their characteristic at the present-time that rebuild Linux into the embedded run-time Linux. And it has expounded some methods about testing operation system real-time character. Also it has expected the development of embedded Linux. At the end it pointed that two-kernel method is better than other methods in rebuilding operation system run-time character for the most common developer basing on the embedded Linux system.

**Key words:** embedded; operation system; Linux; real-time character

### 0 引 言

随着微处理器技术的发展,嵌入式系统已经成为计算机应用领域的一个重要的组成部分。嵌入式系统一般定义为:以应用为中心、以计算机技术为基础、软硬件可剪裁、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。简言之,一个嵌入式系统就是一个计算机硬件和软件的集合体<sup>[1]</sup>。而在嵌入式系统应用中,实时性是嵌入式操作系统中最为关键的技术。

文中介绍了将 Linux 改造为嵌入式 Linux(或称为  $\mu$ Linux, Micro-Control Linux)实时操作系统的研究现状与发展趋势。

### 1 操作系统的实时性

实时系统一般定义为:系统能够在限定的响应时间内提供所需水平的服务。而一个由 Donald Gillies 提出的更为大家所接受的定义是:一个实时系统是指计算的正确性

不仅取决于程序的逻辑正确性,也取决于结果产生的时间,如果系统的时间约束条件得不到满足,将会发生系统出错<sup>[2]</sup>。

根据对外界的事件做出反应的时间要求,操作系统的实时性分为硬实时和软实时。硬实时系统是必须对事件做出及时反应,绝对不能出现错过事件处理的情况。在硬实时系统中如果出现了这样的情况就意味着巨大的损失和灾难。软实时系统是指在系统负荷较重的时候,允许发生错过的情况而且不会造成太大的危害。

实现硬实时系统和软实时系统的主要区别是在选择调度算法上。选择基于优先级调度的算法足以满足软实时系统的需求,而且可以提供高速的响应和大的系统吞吐率;而对硬实时系统来说,需要使用的算法就应该是调度方式简单、反应速度快的实时调度算法了<sup>[3]</sup>。

### 2 嵌入式 Linux 实时性技术

众所周知, Linux 不是一个实时操作系统。因此,为使 Linux 满足实时应用的要求,就需要添加实时软件模块,而这些模块运行的内核空间正是操作系统实现调度策略、硬件中断异常和执行程序的部分,因此如果出现代码

收稿日期:2005-12-11

基金项目:安徽省高等学校自然科学研究基金(2005KJ070)

作者简介:陈文星(1980-),男,江苏南京人,硕士研究生,研究方向为嵌入式系统应用;张辉宜,教授,研究方向为嵌入式系统应用。

错误,就有可能破坏操作系统从而影响整个系统的可靠性,这对于实时应用是一个严重的弱点。

常用的方法是通过对 Linux 的内核进行裁减和修改,使其能够满足实时性的要求。目前,根据实际需要对于 Linux 的改造有很多方法,可以从可抢占式内核体系结构、实时任务调度策略、中断处理过程的优化以及提供微定时器等方面进行<sup>[4,5]</sup>。

## 2.1 可抢占式内核实现方法

抢占(preemption)是指当系统中出现一个优先级高于当前运行进程的进程时,系统立刻中断当前进程运行该进程,以保证在任意时刻当前所运行的进程都是系统中优先级最高的,这也是绝大多数实时系统的调度方式<sup>[6]</sup>。

### 1) 双内核方法。

双内核方法是在硬件平台上增加一个实时内核,建立双内核,例如现在流行的 RTLinux 采用的就是这种方法<sup>[7]</sup>。这两个内核相互配合、协同工作;实时核负责硬件管理并提供精确的实时多任务管理,而 Linux 核提供复杂的非实时通用功能;Linux 核心被看作实时核心中优先级最低的任务来调度,只有当没有可运行的实时任务时 Linux 核心才被调度,如图 1 所示。

这种方法的关键是运行在常规 Linux 核上的所有非实时任务必须是支持可抢占式调度的,这样才能做到对实时核提供精确实时不产生影响。由于实时核非常小,实现容易,并不会过多地增加整个系统的负载,而且保证对于现有的 Linux 内核改动较小<sup>[8]</sup>,因此,这种方法也是目前广泛采用的一种实时性改进方法。

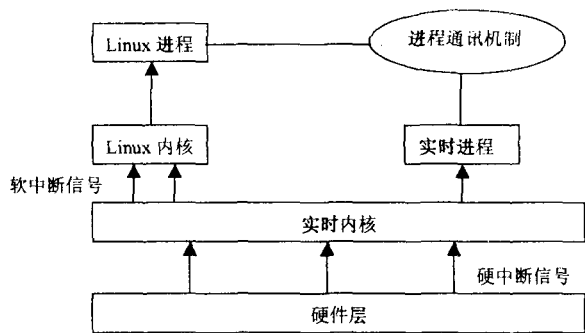


图 1 双内核结构图

### 2) 修改核方法。

这种方法是基于已有 Linux 系统对实时软件开发的支持,对 Linux 内核进行源代码级修改而使 Linux 任务调度的响应时间得到提高<sup>[9]</sup>。目前有两种比较成熟的方法,一种是 Monta Vista 公司提供的抢占补丁(preemption patches)对于内核的修改,这种方法的基本思想是尽最大可能为任务提供调度机会,以及减少事件发生到事件任务被调度之间的时间,从而提高系统的响应时间。

另一种方法是 Ingo Molna 为实现实时 Linux 内核而开发的低延迟补丁程序(Low-Latency Patches)。这种方法不像抢占补丁直接实现内核的可抢占性,而是通过在一些执行时间较长的任务中间加入抢占点,使这些运行时间

长的任务在执行期间,能被一些高优先级或实时性强的任务抢占,从而实现 Linux 内核上的任务抢占。

### 3) 兼容核方法。

这种方法的设计思想是重新设计一个完全独立的,同时具有实时性的内核,但其 API(Application Programming Interface)函数与 Linux 核心兼容,用这个内核去替换 Linux 内核。这种方法的缺点在于,由于设计了一个完全独立的实时核心而没有使用原有 Linux 核,导致 Linux 系统的一些优势难以继承。并且,开发一款兼容所有 Linux API 函数的实时核也决不是件容易的事。因此,这种方法目前只是设想而已,还未见具体实践的报道。

## 2.2 定时器改进方法

Linux 中每次时钟中断都触发 schedule() 进程调度函数,因此,定时器频率直接影响到系统响应速度和上下文切换的系统开销。如果简单地只把定时器时间减小(如设定为 1 微妙),则定时中断过多,CPU 频繁地进行时钟中断处理,反而会使系统性能得不到提高。

在嵌入式系统中,定时器设定的一般原则是:不要求系统在每一个微秒都产生中断,而是在任何一个微秒都能允许发生中断<sup>[10]</sup>。这就要求定时器一般都设定为一次模式,而不是周期模式,即定时器的时钟频率不必固定,仅需设置为下一个事件要发生的时间。或者采用一次模式和周期模式结合方式;或者采用两个定时器,一个设定为周期模式,另一个设定为一次工作模式。

## 2.3 中断处理改进方法

在 Linux 内核上,对外部中断都是开启的,都采用中断处理方法。在实时操作系统中,一般不采用这种方法,而采用一种比较折中的方法:对于一部分中断,如时钟中断和一些要求及时处理的中断都采用中断方式处理,而对于其它的中断则采用轮询方式<sup>[11]</sup>。这样既可以有效地及时处理实时中断,又可以避免系统陷入频繁的中断处理中,将一些非实时要求中断作为任务而被调度处理。

## 2.4 优先级反转问题

在 Linux 内核采用可抢占式调度策略后,当有多个实时进程互斥请求共享资源时,有时候会出现一种现象:由于多进程共享资源,具有最高优先权的进程被低优先级进程阻塞,反而使具有中优先级的进程先于高优先级的进程执行,最终高优先级的进程得不到处理而饿死,导致系统的崩溃。这就是所谓的优先级反转或优先级倒置(Priority Inversion)<sup>[12~14]</sup>现象。目前解决优先级反转问题普遍使用的有 2 种方法:

(1) 优先级继承方案(priority inheritance):当高优先级任务在等待低优先级任务占有的信号量(semaphore)时,让低优先级任务继承高优先级任务的优先级,即把低优先级任务的优先权提高到高优先级任务的优先级;当低优先级任务释放高优先级任务等待的信号量时,立即将其优先权降低到原来的优先权。著名的 WindFiver 公司的 Vx-Works 采用的就是这种方法。

(2) 优先级极限方案(priority ceilings):系统把每一个临界资源与一个极限优先级相联系,这个极限优先级等于系统此时最高优先级加 1。当一个任务进入临界区时,系统便把这个极限优先级传递给这个任务,使得这个任务的优先级最高;当这个任务退出临界区后,系统立即把它的优先级恢复正常,从而保证系统不会出现优先级反转的情况。

### 3 实时性测试方法

对 Linux 实时性改进结束后,必须对其进行测试。目前,常用的测试实时性的方法有三种:Rhealstone 法、进程分派延时法和三维表示法<sup>[15~17]</sup>。

#### 3.1 Rhealstone 法

Rhealstone 方法对 ERTOS(Embedded Run-time Operation System)中 6 个关键操作的时间量进行测量,并用它们的加权和——Rhealston 数衡量实时性的好坏。这 6 个时间量如下:任务切换时间(task switching time),也称上下文切换时间,定义为系统在两个独立的、处于就绪态并具有相同优先级的任务之间切换所需要的时间;抢占时间(preemption time),即系统将控制权从低优先级的任务转移到高优先级任务所花费的时间;中断延迟时间(interrupt latency time),指从中断发生到系统获知中断,并开始执行中断服务程序(ISR)第一条指令所持续的时间间隔;信号量混洗时间(semaphore shuffling time),指从一个任务释放信号量到另一个等待该信号量的任务被激活的时间延迟;死锁解除时间(deadlock breaking time),即系统解开处于死锁状态的多个任务所需花费的时间,死锁解除时间反映了 RTOS 解除死锁的算法的效率;数据包吞吐率(datagram throughput time),指一个任务通过调用 ERTOS 的原语,把数据传送到另一个任务去时,每秒可以传送的字节数。

#### 3.2 进程分派延时法

进程分派延迟时间 PDLT(Process Dispatch Latency Time)是另一个常用的测量 ERTOS 性能的方法。在实时系统中,实时任务总是等待外部事件引发的中断来激活它。当一个中断产生后,系统必须迅速停止当前运行的低优先级任务,将控制权交给被激活的实时任务。PDLT 定义为从中断的产生到由中断激活的实时任务开始执行之间的时间间隔。这段间隔由几个部分组成,如图 2 所示。

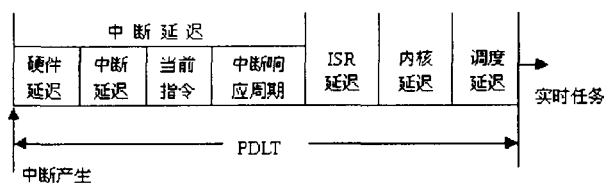


图 2 进程分派延迟时间

不同操作系统中,PDLT 差异的主要部分是内核延迟部分。目前绝大多数 ERTOS 为了减少内核延迟,采用可抢占式的内核,可有效地提高系统对外部事件的响应速

度。

#### 3.3 三维表示法

这种方法,将 ERTOS 的工作分为 3 个阶段:响应传感器或者其他输入设备的请求,并获取数据;对获得的数据进行处理(主要由应用程序进行处理);输出处理结果。相应地,ERTOS 的性能可以用对应的 3 个特性来描述:CPU 的计算能力,其度量单位为 MIPS1 (Millions of Instructions Per Second);中断处理能力,其度量单位为 MIPS2 (Millions of Interrupts Per Second);I/O 吞吐率,其度量单位为 MIPS3 (Millions of I/O Per Second)。

上述 3 个特性的最大值可分别单独测得,但这 3 个特性之间并不是相互独立的。为了直观地表现 ERTOS 的实时性能,可以用一个三维的图形来表达 3 个特性之间的依赖关系。如图 3 所示。

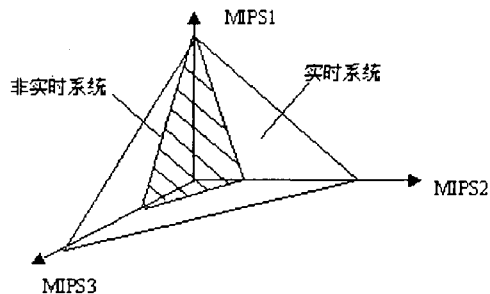


图 3 系统实时性能的三维表示法

图 3 中用曲面来表现 ERTOS 3 个特性之间的依赖关系。如果随着一个特性的增加,另外两个特性下降的速度比较缓慢,可以认为该曲面所表现的系统是 ERTOS;反之,如果随着一个特性的增加,另外两个特性下降的速度超过了一定的范围,就可以认为该系统非 ERTOS,如图 3 中的阴影部分所示。

### 4 总结与展望

Linux 是类 UNIX 分时系统,在实时性支持方面存在很大不足,为满足实时应用,必须对其进行改造。在众多实时性改造的方法中,双内核法以其实时内核小,容易实现,不会过多地增加整个系统的负载,并且保证对现有的 Linux 内核改动较小等优点,得到广泛应用。但是,在嵌入式系统开发的过程中不能一概而论,在对实时性改进时,应根据实际系统对实时性的敏感度要求,决定如何改进。随着实时性的逐步完善,嵌入式 Linux 应用中仍有很多工作要做,比如:个性化定制、网络/安全、集成开发环境等,只有这些工作进一步的完善才可以使嵌入式 Linux 得到更广泛的应用。

#### 参考文献:

- [1] 许海燕,付炎.嵌入式系统技术与应用[M].北京:机械工业出版社,2002.
- [2] 亢建卫.嵌入式 Linux 操作系统研究[D].西安:西安交通大学,2004.

- [3] 张荣斌. 基于嵌入式 Linux 的实时操作系统研究[D]. 阜新: 辽宁工程技术大学, 2003.
- [4] 毛德操, 胡希明. Linux 内核源代码情景分析(上, 下册)[M]. 杭州: 浙江工业大学出版社, 2001.
- [5] 李善平. Linux 内核 2.4 版源代码分析大全[M]. 北京: 机械工业出版社, 2002.
- [6] 董晓峰. 嵌入式 Linux 及其调度策略的研究[D]. 西安: 西安电子科技大学, 2004.
- [7] Dankwardt K. Real Time and Linux[EB/OL]. <http://www.linuxdevices.com/articles/AT5997007602.html>, 2005-01.
- [8] Vista M. Linux Real - Time Performance[EB/OL]. <http://www.mvista.com/dswp/realtime.pdf>, 2004-12.
- [9] Williams C. Which is better the preempt patch, or the low - latency patch? [EB/OL]. <http://www.linuxdevices.com/articles/AT8906594941.html>, 2005-03.
- [10] 陈敏, 周兴社. 基于嵌入式 Linux 的实施优化方案[J]. 计算机应用研究, 2005(3): 235-237.
- [11] 刘彦忠. 嵌入式 Linux 操作系统实时控制的改进[J]. 齐齐哈尔大学学报, 2005, 21: 64-65.
- [12] 中原电子. RTOS 的优先级反转问题[EB/OL]. <http://www.ccelec.com/Docs/RTOS/RTOSPriorityReverse2.htm>, 2004-01.
- [13] 钟汉如, 王创生. 嵌入式中调度算法的实现及优化[J]. 计算机工程与科学, 2002, 24(3): 5-7.
- [14] 黄廷辉, 衣毅. 嵌入式的实时性能[J]. 桂林电子工业学院学报, 2002, 22(1): 72-75.
- [15] 李庆诚, 顾健. 嵌入式实时操作系统性能测试方法研究[J]. 单片机与嵌入式系统应用, 2005(8): 19-21.
- [16] Heursch A C, Cramhow D, Horstkotte A, et al. Steps Towards A Fully Preemptable Linux Kernel[J]. Real - time Programming, 2003, 3(1): 17-20.
- [17] Williams C. Linux Scheduler Latency[Z]. Redhat Inc, 2002. 8-10.

(上接第 113 页)

```
if(isNull(form.area_name, "Area Name is required!"))
    return false; }
</script>
</html:html>
```

#### 4 结束语

Struts 提供了一个非常好的 MVC 框架, 它的 taglib 标记库灵活性大, 在 Web 应用开发中把界面和业务逻辑进行分离, 有利于网页设计者和程序设计者的分工, 便于组件式的开发, 有助于提高代码的可重用性和灵活性, 大大提高了开发效率。同时它把业务逻辑放在一个配置文件中, 有利于大型 Web 应用的工程化开发。利用 Action. Servlet 配合 config. xml 实现对整个系统导航, 使系统脉络更加清晰, 增强了开发人员对系统的整体把握, 提高了系统的可维护性和可扩充性。运用 Struts 技

(上接第 110 页)

也随之变化, 提高了模拟的逼真度。

#### 4 结论与探讨

通过以上的基于 L 系统理论对树木和树林进行三维模拟的实例可以看出在对具有自相似结构的植物进行计算机生成的时候, L 系统是比较理想的选择。在以后的应用中, 还可以考虑利用随机插值算法生成三维地形图<sup>[8]</sup>, 在此基础上生成不规则地貌上的三维植物群落, 进一步提高模拟自然景物的逼真度。

#### 参考文献:

- [1] 曾文曲, 王向阳. 分形理论与分形的计算机模拟[M]. 沈阳: 东北大学出版社, 1993. 134-143.

术, 快速构建了一个结构清晰、可复用、可扩展的会员服务系统。

#### 参考文献:

- [1] 张娅, 杜友福, 张南平. 用 Struts 建立 MVC 模式的 Web 应用[J]. 微机发展, 2003, 13(12): 105-107.
- [2] 丁振国, 任新洁. 基于 Struts 的 Web 应用开发研究[J]. 微机发展, 2004, 14(1): 90-92.
- [3] Cavaness C. Programming Jakarta Struts[M]. 北京: 电子工业出版社, 2002.
- [4] 陈建峡. Struts 框架在 Web 程序设计中的应用[J]. 湖北工业大学学报, 2005(3): 219-221.
- [5] Price J. Oracle9i JDBC 程序设计[M]. 北京: 机械工业出版社, 2003.
- [6] Moss K. Java Servlet 开发人员指南[M]. 北京: 清华大学出版社, 2003.

- [2] Kenneth J F. 分形几何 - 数学基础及其应用[M]. 曾文曲, 刘世耀译. 沈阳: 东北大学出版社, 1991. 173-185.
- [3] 胡瑞安, 胡纪阳, 徐树公. 分形的计算机图像及其应用[M]. 北京: 中国铁道出版社, 1995. 113-137.
- [4] 杨光, 杨树红, 程晓青. 随机分形及其在自然景物模拟中的应用[J]. 试验技术与试验机, 1996(2): 118-120.
- [5] Edgar G A. The forest fractal puzzle[J]. Computers & Graphics, 2000, 24: 133-141.
- [6] 李峰, 陈天滋. 自然景物的模拟[J]. 计算机应用研究, 1999(2): 76-77.
- [7] 高旭, 姜楠. 分形 L 系统理论与植物图像的计算机模拟[J]. 扬州大学学报, 2000(3): 71-74.
- [8] 郭维红, 徐伟忠, 谈正. 基于分形几何学的自然景物虚拟实现的研究[J]. 西安交通大学学报, 1995(5): 77-84.