

文章编号:1001-9081(2005)07-1679-03

Linux 操作系统实时性测试及分析

李 江,戴胜华

(北京交通大学 电子信息工程学院,北京 100044)

(leestart2000@163.com)

摘 要:在特定平台上对 Linux 的实时性重要指标:中断响应和上下文切换开销加以测试并分析,为 Linux 嵌入式中的实时应用提供了依据。Linux 中断响应与 Linux 内核机制和内存管理机制紧密相关,而 Linux 上下文切换开销与 Linux 进程空间和硬件平台情况紧密相关。

关键词:实时性;中断响应;上下文开销;Lmbench;嵌入式

中图分类号: TP316 **文献标识码:** A

Real-time performance test and analysis of Linux

LI Jiang, DAI Sheng-hua

(School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: The interrupt latency and context switch latency of Linux were measured and analysed to manifest the real-time performance of Linux in special platform, providing the datum when Linux was used as embedded operation system. Experiment results show that the interrupt latency of Linux is influenced by kernel mechanism and memory management mechanism, and the context switch latency of Linux is influenced by process space of Linux and hardware.

Key words: real-time; interrupt latency; context switch latency; lmbench; embeded system

Linux 作为一种通用的操作系统被广泛应用,由于它体积小、可裁减、易开发,因此也广泛应用于嵌入式领域的开发。对于操作系统来说,它的实时性,指的是可以及时响应外部的事件的请求,并在一定的时间内完成对该事件的处理的能力。一般而言,嵌入式操作系统对实时性要求较通用的操作系统来说要高得多。随着 Linux 越来越多地应用于嵌入式领域,它的实时特性就越越来越引起人们的注意。分析 Linux 的实时性,必须了解 Linux 的内核工作原理。

1 Linux 实时机制概述

Linux 是一种基于优先级调度策略的操作系统。Linux 将进程分成两种类型:实时进程和非实时进程。对于实时进程采用优先级的轮转法调度(SCHED_RR)和基于优先级的先入先出(SCHED_FIFO)服务调度。对于非实时进程采用了基于优先级的轮转法(SCHED_OTHER)调度实时进程的优先级比非实时进程的优先级的基数要高 1000。Linux 通过优先级基数设定实现了分级,实时进程和非实时进程同在一个调度队列,但实际已被分成了两级,即实时进程优先级绝对高于非实时进程而会优先得到调度。Linux 中,实时进程无论何时进入进程等待队列,都将被排在非实时进程之前。

Linux 又是一种非抢占式的操作系统,当系统正在执行一个低优先级的任务时,高优先级的任务不能抢占低优先级的任务,而是处于等待状态。高优先级的任务的等待时间由低优先级的任务的执行情况决定。所以说,Linux 的实时性不仅由实时进程执行时间决定,也由非实时进程的执行时间决定。

Linux 中断处理机制有其自身的特点,在 Linux 里,中断处理程序可以分为低级处理部分(top half)和高级处理部分(bottom half)。低级处理部分作用是用来及时响应中断信

息,而具体的中断处理工作则交给中断的高级处理部分完成;高级处理工作并不是立即完成,而是将处理程序放在一个特殊的队列中,以后再执行。

通过这种机制,Linux 缩短了对外界请求的响应时间。

2 Linux 的中断响应测量

2.1 测量方法

中断响应延迟(interrupt latency)是 Linux 操作系统实时性的重要指标。系统的实时性越强,中断响应的的时间越少。中断响应时间 = 中断延迟时间 + 保存 CPU 内部寄存器的时间。其中中断延迟时间是指从外部中断信号产生到当前任务停止这一段时间。保存 CPU 内部寄存器的时间为将 CPU 内部寄存器内容压入堆栈的时间。测量采用的硬件平台和 Linux 操作系统版本为:

处理器型号,AMD Athlon(TM)XP;存储器容量,261.664 KB RAM;主机频率,1 800Mhz;Linux 类型, Linux Redhat 9.0;内核版本,2.4.20-8。

基于特定的测试环境和测试对象可以采用两种测量方法:

测量方法 1 采用示波器监测的方法测量中断响应。

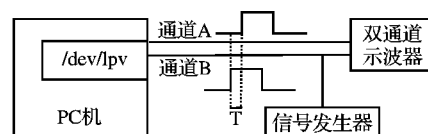


图 1 测量方法 1

示波器的两通道 A,B 分别与 PC 机的并口相连,信号发生器连接在示波器的 B 通道上。由信号发生器产生一脉冲信号,脉冲信号经通道 B 分别传向双通道示波器和 PC 机并口,示波

收稿日期:2004-12-21;修订日期:2005-04-23

作者简介:李江(1979-),男,山西太原人,硕士研究生,主要研究方向:智能交通控制、嵌入式应用开发;戴胜华(1962-),男,安徽芜湖人,副教授,主要研究方向:智能交通控制、嵌入式应用开发。

器接收信号显示。并口接到脉冲后,触发外部中断 IRQ7,IRQ7 唤醒 Linux 中的 read 命令,read 命令作用于驱动程序并口驱动/dev/lpv,通过并口返回相同的脉冲信号,经通道 A,显示在示波器上。示波器前后两次脉冲间的时间间隔 T,可近似地认为 Linux 中断响应时间为 T。Linux 中断程序代码为:

```
void main()
{
    int fd;
    int i, j;
    unsigned char buff;
    fd = open("/dev/lpv", O_RDWR);
    if (fd < 0)
    {
        perror("/dev/lpv: could not open");
        exit(-1);
    }
    //使能中断
    ioctl(fd, LPV_IRQ, LPV_ENABLE);
    while(1)
    {
        //等待中断
        read(fd, &buff, 1);
        //产生脉冲
        write(fd, "1", 1);
        for(j = 0; j < 10000; j++)
            write(fd, "3", 1);
    }
    //禁止中断
    ioctl(fd, LPV_IRQ, LPV_DISABLE);
    close(fd);
}
```

测量方法2 读计数器状态寄存器法测量中断响应。

在 Linux 运行一个测试程序,测试程序包括主程序和一个中断处理程序。主程序实现 8254 的初始化和中断触发条件。中断处理程序用以获取中断响应后定时器的数值并存在指定文件中。设定 8254 的计数值达到 100 时中断,100 作为中断响应前的起始时间,存于指定文件中的计数值是中断响应后 8254 的计数值,也是中断响应的终止时间。由测试数据,可以计算 Linux 中断响应时间:Linux 中断响应时间 = (指定文件中的计数值 - 100) * 定时器计数间隔。

2.2 测量中断响应

这两种测量方法各有优缺点:采用方法读计数器状态寄存器的方法测量中断响应,可以得到更精确的测量值,但这种方法需要实现的程序代码比较复杂;相比而言,采用示波显示测量中断响应的方法则更为直观,实现代码也比较简单,缺点是受到测试工具精度的影响。

为得到更为直观的测量结果,采用示波显示测量中断响应的方法测量中断响应。首先在系统开机默认负载的情况下,测量 10 次取其平均值,测量得到在当前硬件平台下 Linux 中断响应的平均时间为 425.4 μ s。进一步测试 Linux 中断响应,可以打开 10 个 Mozilla 应用程序,来增加系统负载,在这种条件下测量 10 次取其平均值,结果发现 Linux 中断响应的平均时间变为 557.2 μ s。在打开 20 个 Mozilla 应用程序时,测试其平均结果为 647.5 μ s,打开 30 个 Mozilla 应用程序时为 721.8 μ s。究其原因,是由于 Linux 中采用了虚拟内存的机制,虚拟内存机制使 Linux 程序依照调度算法在内存和外存间交互。当系统有过多的应用程序在运行时,交互更为频繁,这样就消耗了大量的系统时间,从而增加了 Linux 中断响应时间。Linux 中虚拟内存机制与内存管理单元 MMU 紧密相

关,而 Linux 作为嵌入式操作系统时,为了减小中断响应时间,常常采用裁减 Linux 中与 MMU 相关的部分,同时修改相关内核源码的方法来改进 Linux。

3 测量上下文切换开销

3.1 测量原理

Lmbench 是一种功能强大的操作系统的测试工具,它提供了硬件性能的监测(例如:存储器延迟),同时也包括了操作系统的性能检测(例如:上下文切换延迟)。它将操作系统性能检测分解到它所依赖的硬件组件的性能检测,Lmbench 作为一种建立操作系统性能及其结构特性之间联系的工具,提供了一种从高层操作系统到底层硬件的方法。Linux 系统性能和其结构特征之间由上到下依赖关系如下:

- 1) Linux 系统性能;
- 2) Linux 系统高层操作性能:上下文切换延迟(context switch latency);
- 3) Linux 系统底层操作性能:进程操作延迟(fork + /bin/sh-c latency);
- 4) 系统硬件性能:存储器延迟(memory latency)。

同时,为了保证 Lmbench 的测量结果更为准确,Lmbench 采用了两种统计方法对测量值加以处理:

- 1) Lmbench 保留(100 - 2n)% 的统计值,它的含义是将 Lmbench 结果中的最好值和最差值的 n% 去掉。在通常的情况下 n = 10。
- 2) Lmbench 的测量结果中可能会出现测量值同系统设置值显著背离的情况。为保证测量的准确性,Lmbench 将这种显著背离的情况控制在一定范围内,所有测量值的显著背离的情况不超过 1%。

3.2 测量上下文切换延迟

上下文切换延迟(context switch latency)是操作系统性能的显著指标之一。Linux 上下文切换包括从 Linux 选择一个新进程到改变硬件状态去运行它的全过程。

使用 lmbench2.0 测量 Linux 的上下文开销,硬件平台与测量对象同测量中断延迟试验相同。结果如图 2 所示:4K, 8K, 16K, 32K, 64K 为组成进程队列的进程空间的大小。

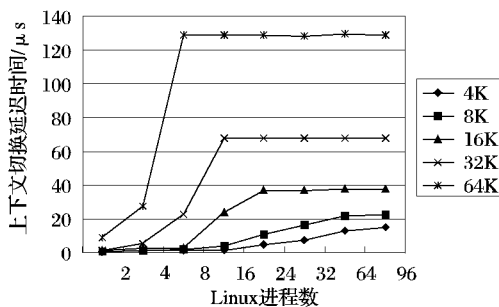


图2 测量结果

分析测量结果,可以得出以下几个结论:

- 1) Linux 上下文切换开销与进程空间的大小成正比。Linux 的上下文切换实质是 Linux 进程间的切换。当 Linux 内核运行在多进程状态时,每个进程都有自己的进程空间。当前进程的时间片即将用完,决定运行另外的进程时,它保存正在运行的进程的当前状态,即 CPU 寄存器的全部内容。这些内容保存在进程自身的堆栈中。入栈工作完成后,就把下一个将要运行的进程的当前状态重新装入 CPU 寄存器,并开始下一个任务的运行。

Linux 上下文切换时间 = 保存当前进程的当前状态的时间 + 将要运行进程的当前状态重新装入时间。对 Linux 的不同进程,保存进程的当前状态的时间是不同的,当前状态重新装入 CPU 寄存器时间也是不同的。在保存当前进程的当前状态时,占用空间越大的进程,由于自身的进程空间较大,压入自己进程堆栈所需的时间越长。同理,当占用空间越大的进程即将运行时,将自己进程堆栈里的内容重新装入 CPU 寄存器的时间也会越长。因此进程空间越大的进程在做上下文切换时,开销越大。

2) Linux 的上下文切换开销与 cache 的容量大小和放入 cache 的进程个数有关。Linux 的多个进程在做进程切换的时候,首先将进程队列将要运行的一个或多个进程放入或拿出 cache,这样多个进程在 cache 中竞争造成了延迟。如图 2 所示,大小为 32K 的进程组成的队列,当进程队列中进程数由 2 变为 16 时,这时由于需要将更多进程放入 cache,cache 中会相对拥挤,进程共享和竞争 cache 资源,从而造成延迟时间更大。而当进程队列中进程数 16 变为 96 时,延迟变化不大,这是因为 cache 空间是一定的,放入 cache 的进程达到饱和,此时延迟不再增加。

3) Linux 上下文切换开销还与系统内存的大小有关。Linux 上下文切换开销还包括 Linux 进程从内存读入 cache 的额外延迟。Linux 不支持内存页智能查找机制,因此在 Linux 将进程从内存页读进 cache 时需遍历内存页查找进程地址,从而造成额外延迟。为了验证这一点,可以让 Lmbench 分别在内存为 261M 和 523M(其余测试环境相同)的情况下,对 Linux 上下文开销进行测试。表 1 是两种情况下的对比,可见内存的容量越大,将进程放入 cache 所消耗时间越多,上下文开销越大。

表 1 内存不同时上下文开销对比(单位:μs)

内存大小	进程数				
	4	8	16	32	64
523M	1.65	1.51	1.72	7.34	13.64
261M	1.28	1.34	1.41	7.32	12.93

4) Linux 作为嵌入式操作系统,有其特殊性,由于功耗原因,嵌入式处理器一般都具有较低的主频,为了进一步分析 Linux 在较低主机频率下的实时性能,可以用降低 PC 机 CPU 主频频率方法,检测 Linux 在不同主频时的上下文开销。为得到普遍情况的测试值,测试选取组成队列的进程队列的进程空间大小为 16K,队列中进程数为 16 的典型情况,从 500MHz 到 1000MHz 之间不同的主频,每个频率测得 10 组数据,然后取平均值,测量结果如图 3。

通过测试结果,可以发现 Linux 用作嵌入式操作系统的两个工作特性:一是 Linux 的平均上下文切换延时随处理器频率的升高逐渐下降。二是 Linux 的上下文切换延时在处

理器频率较高时,其变化幅度较小。在处理器频率 600M 时, Linux 的上下文切换的最大延时和最小延时的差值为 8.51μs, 700M 时为 4.1μs,而到了 1000M 时, Linux 上下文切换的最大延时和最小延时的差值仅为 0.75μs。可见处理器频率越高, Linux 的上下文切换延时越稳定。而在较低频率时,变化幅度较大。

实验结果表明,在 Linux 上开发应用程序时,为了提高 Linux 的实时性,减小上下文切换开销,首要的一点就是在编写程序时尽量减小应用程序中进程的进程空间大小。其次对 Linux 的实时性,还要充分考虑 Linux 运行的硬件平台的具体因素,尤其是 cache 的大小和内存容量。Linux 的实时性与 cache 的大小成正比,与内存容量成反比。这一点在嵌入式开发中尤为明显。此外,由于嵌入式处理器较低的频率,限制了 Linux 的实时性能。这就需要我们在使用 Linux 作嵌入式开发时处理好性能与功耗之间的关系,使二者得到兼顾。

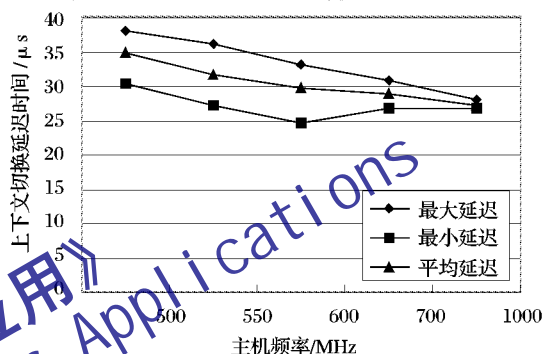


图 3 Linux 在不同主频时的上下文开销

4 结语

通过在特定的硬件平台上对 Linux 中断响应时间和上下文切换延迟这两项实时性指标进行测试,可以得出实际应用影响 Linux 实时性的主要因素。使开发人员可以在短时间内了解 Linux,并为 Linux 作为嵌入式操作系统的实时应用提供了依据。然而,对 Linux 实时性测试是一项复杂而细致的工作,有待进一步研究。

参考文献:

- [1] 陈莉君. 深入理解 Linux 内核[M]. 北京: 中国电力出版社, 2001.
- [2] 李善平, 郑扣根. Linux 操作系统及试验教程. 北京: 机械工业出版社, 1999.
- [3] 邹思铁. 嵌入式 Linux 设计与应用[M]. 北京: 清华大学出版社, 2002.
- [4] BARABANOV M, YODA IKEN V. Introducing real time Linux[J]. Linux Journal, 1999, 34(1).
- [5] WILLIAM C. Linux Scheduling latency[Z]. Red Hat Inc, 2002.
- [6] LOCKE D. A TimeSys response on Real-Time Linux[Z]. Monta Vista Software, 2001.

(上接第 1678 页)

- [11] YU Y, GOVINDAN R, ESTRIN D. UCLA/CSD-TR-01-0023, Geographical and Energy Aware Routing: A recursive Data Dissemination Protocol for Wireless Sensor Networks [R]. University of California, 2001.
- [12] 林亚平, 王雷. 传感器网络中的分布式数据汇聚层次路由算法[J]. 电子学报, 2004, 28(11): 1290-1297.
- [13] 彭伟, 卢锡诚. 一个新的分布式最小连通支配集近似算法[J]. 计算机学报, 2001, 24(3): 254-258.
- [14] 高协平, 张钊. 区间子波神经网络(I)——理论与实现[J]. 软件学报, 1998, 9(3): 217-221.
- [15] 高协平, 张钊. 区间子波神经网络(II)——理论与实现[J]. 软件学报, 1998, 9(4): 245-250.
- [16] ANDERSSON L, HALL N, JAWERTH B et al. Wavelets on closed subsets of the real line [A]. SCHUMAKER L. Recent Advance in Wavelet Analysis [C]. Boston: Academic Press, 1994. 1-61.