

Linux/RTLinux 下实时多线程实时性检测方法

曹廷发, 吉 华, 马 江, 付耀国

(四川大学, 四川 成都 610065)

摘 要:实时多线程是实时控制软件实现的重要手段,但是基于 Linux/RTLinux 的实时控制软件的多线程实时性检测不易,尤其连续输出大量测试信息困难。设计了三种多线程的实时性检测方法,并分析了其优缺点。重点讨论了使用 `rtl_printf` 函数和 `dmesg` 命令协同输出测试信息的这一随机采样的测试方法,并给出设计细节和应用实例。分析如应用表明,这种使用 `rtl_printf` 和 `dmesg` 输出信息的测试方法在需要连续采集大量数据、随机采样能够满足要求的场合下是有效的。

关键词:Linux/RTLinux;多线程;实时性检测

中图分类号:TP273

文献标识码:A

文章编号:1673-629X(2010)03-0113-03

Real-time Performance Test Methods for Real-time Multi-thread Based Linux/RTLinux

CAO Ting-fa, JI Hua, MA Jiang, FU Yao-guo

(Sichuan University, Chengdu 610065, China)

Abstract: Real-time multi-thread is an important implementation method of real-time control software. However, it is difficult to test the real-time performance of real-time multi-thread of control software based on Linux/RTLinux, especially when outputting large amount of test-messages continuously is desired. Three methods were designed to test the real-time performance of multi-thread, whose advantages and disadvantages were analyzed. Moreover, the random sampling method that uses the function `rtl_printf` and command `dmesg` together to output test-message was discussed in detail. And an application example was also given. The analysis and application showed this method is effective when a large amount of data is needed to be collected continuously and when random sampling meets the testing requirement.

Key words: Linux/RTLinux; multi-thread; real-time performance test

0 引言

Linux+RTLinux 实时扩展作为实时控制软件的操作系统,可利用 RTLinux 的出色硬实时性^[1]和 Linux 下丰富的软件资源,是实时控制软件操作系统的良好选择之一^[2]。大多数实时控制软件的核心功能是基于多线程实现的,多线程的实时性很大程度上决定了实时控制软件的性能,因此设计一种有效适用的多线程实时性检测方法对于实时控制软件的开发具有重要意义。

基于 RTLinux/Linux 的实时控制软件的多线程实时性检测是一件困难的事,主要表现在:

1)多线程可以认为是并发执行,多线程之间的关

系复杂,在软件多个线程运行的状态下连续测量多线程的状态更有实际的意义,但如何在多线程同时运行的状态下测量多线程的状态不易;

2)实时软件中线程运行精度高,运行速度快,测量需要高精度的时钟,如何测量多线程的运行时间有一定的难度;

3)RTLinux 线程运行 Linux 内核态, Linux 内核空间程序调试困难,信息输出的方式少、约束多,在内核中读取数据不容易。

目前虽然基于 RTLinux/Linux 的实时控制软件的开发和研究很多,但对于基于 RTLinux/Linux 的实时控制软件的多线程实时性讨论较少,对于多线程实时性检测方法的讨论更少。文献[3]在 Window 下实现了一个测试工具用于多线程运行时序的测试,但是没有涉及测试工具的具体实现方法。文献[4]使用与本文最接近的方法,它主要测试中断响应时间和线程运行时间,但是其设计的试验方法,由于 message 文件长度有限,只能对短时间内的少量数据进行采集,不能实

收稿日期:2009-07-03;修回日期:2009-10-28

基金项目:四川省科技厅科技支撑项目(2009GZ0009)

作者简介:曹廷发(1984-),男,贵州遵义人,硕士研究生,研究方向为过程装备控制技术;吉 华,博士,副教授,研究方向为机械制造及其自动化。

现数据的连续采集,而多线程实时性的测试应该是连续采集大量数据的统计才有效。文献[5~8]使用硬件的测试方法对操作系统或者应用系统进行中断响应等实时性指标检测,但都不是连续性测量。目前一些 Linux 下的性能测试工具^[9],如:Timepegs、Intlat、Schedlat、Lmbench 等,由于实时控制软件的实时模块运行在 RTLinux 内核空间,由实时内核控制,不受 Linux 内核控制,所以不能应用于 RTLinux 下实时控制软件实时多线程的测试。

笔者设计了三种基于 RTLinux/Linux 的实时控制软件多线程实时性检测方法,分析了它们的优缺点。第一种方法使用硬件输出测试信息;第二种方法用文件输出测试信息;第三种方法用 rtl_printf 函数和 dmesg 命令协同输出测试信息。文中重点对第三种测试方法进行了详细设计,并给出了应用实例。

1 基于 Linux/RTLinux 的多线程实时控制软件实时性检测方法设计

实时控制软件的核心功能大多由实时线程完成,为了准确检测实时控制软件的实时性,必须准确获取实时线程的运行状态,主要有线程运行时间、线程状态(如:进入、离开、挂起)。下面设计了三种不同的多线程实时性检测方法,这三种方法采用相似方法获取实时线程的运行状态,对于运行时间的检测,使用 RTLinux 时钟,RTLinux 时钟精度高,可利用其对实时控制软件的多线程进行运行时间检测^[10];对于线程状态的获取,直接在相应代码(如:进入、离开、挂起代码处)加入测试信息输出代码。这三种方法不同的是测试信息输出方法。

1.1 测试信息输出方法

1.1.1 使用硬件输出测试信息

如图 1,使用硬件(例如并口)输出运行状态,用不同的端子线标示不同线程的运行状态,并在连接线的另一端使用示波器等仪器测量输出信息。但是由于线程运行状态值多,需要很多的连接线进行标示。也可以不使用并口这类计算机自带的接口,而使用能够进行模拟量输出的板卡,用不同的电压值表示线程的运行状态,这种测量方法用于测量多线程的时序时表现直观。使用硬件输出测试信息的方法具有检测量反映

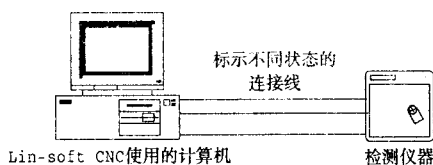


图 1 硬件输出测试信息

直观、测量精度高的优点,但是需要专门的高精度仪器或者自行设计的仪器进行检测,所以此方法成本高,限制比较多。

1.1.2 用文件输出测试信息

使用内核文件输出的方法将实时控制软件的多线程运行状态输出到文件中,然后对文件进行分析,可以得到实时线程运行状态、运行时间等信息。由于 RTLinux 的实时线程运行于 Linux 内核空间,传统的文件输出方法,如 fopen、fwrite、fclose 等不能使用,Linux 内核编程方法中,也提供了一套文件读写方法,如 filp_open、write、fput 等。但是内核读写文件的方法不能输出浮点型等复杂数据,只能输出整型数据;而且对于实时控制软件,检测实时线程运行状态所需要的频繁读写文件并不适用,因为频繁读写会增大系统开销,降低软件实时性。在我们的测试实验过程中,这种方法经常造成内核崩溃。这种方法对于间断性采集少量数据的场合是适用的,但对于连续大量采集数据的场合不适用。

1.1.3 使用 rtl_printf 函数和 dmesg 命令协同输出测试信息

rtl_printf 是 RTLinux 提供的打印函数,典型的用途是和用户空间的 printf 函数一样将需要显示的信息显示在屏幕上。通过测试和分析,发现 rtl_printf 和内核打印函数 printk 相似,将打印的内容写到内核消息缓冲区中,在用户空间使用 Linux 命令 dmesg 和配合管道操作可将缓冲区中的数据写入文件保存。使用 rtl_printf 和 dmesg 输出测试信息的方法如图 2。

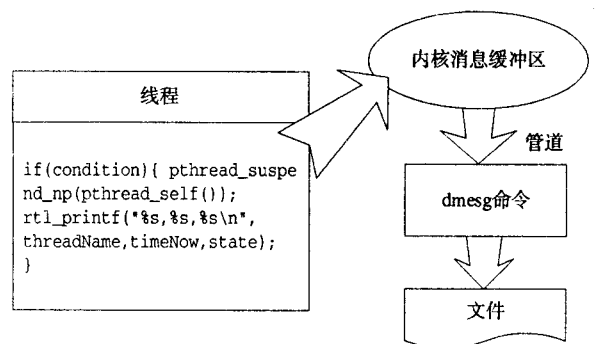


图 2 使用 rtl_printf 和 dmesg 输出测试信息

rtl_printf 循环使用缓冲区,如果缓冲区填满,则 rtl_printf 就绕到缓冲区的开始处填写新数据,覆盖旧数据,这样就会丢失旧的数据;而且 dmesg 和命令管道运行于用户空间,和实时线程相比优先级低,不能保证缓冲区的数据及时输出到文件中。因此使用这种方法会丢失数据,但是由于数据是丢失随机的,这种方法相当于随机取样,当大量数据产生时,该方法是有效的。

1.2 使用 rtl_printf 和 dmesg 输出测试信息的方法的具体实现

使用 rtl_printf 函数和 dmesg 命令协同输出测试信息的测试方法具有自由度高,取样方便,能够连续大量采集数据等优点,是一种随机采样的测试方法,在需要连续大量采集数据的场合能够有效地反映实时控制软件多线程的运行状态。因此重点对这种随机取样的方法进行了详细设计,具体实现如下:

第一,在需要反映运行状态处加入调试代码,主要标示两种状态。一种是线程运行时间,使用 RTLinux 下高精度时钟函数 gethrtime 可以获得线程运行时间,然后使用 rtl_printf 输出,例如:

```
hrtime_t t2,t1;
t1 = gethrtime(); //取得线程开始运行的时间
...//线程功能段
t2 = gethrtime(); //取得线程功能完成的时间
rtl_printf("%s,t2 - t1 = %d \n", threadName, t2 - t1); //
输出线程运行时间
```

另一种是线程状态,在线程出现相应状态时,直接使用 rtl_printf 输出该状态,例如:

```
if(condition)
rtl_printf("%s: suspend. \n", threadName); //输出线程状态
```

第二,由于需要在系统运行过程中连续不断地从系统缓冲区中读出数据,并保存到文件中,所以使用 Linux 的 Shell 编程,实现了一个脚本程序 msgFile,连续不断将 rtl_printf 输出信息保存到文件中, msgFile 实现如下:

```
number=0
while (test 1)
do
dmesg > "$number" //输出测试信息
number='expr $number + 1'
done
echo
```

第三,先运行 msgFile,再运行实时控制软件进行实时性测试。

2 应用实例

可编程控制器(PLC)是一种在工业控制领域应用很广泛的自动控制控制装置,近年来软件 PLC 的研发也是工业控制领域的一个热点,软件 PLC 也是一个典型的多线程实时控制软件。软件 PLC 的主要功能是从硬件读入输入数据,根据用户逻辑对输入数据进行计算得到输出数据,将输出数据写入硬件;同时还要完成和其他软件通讯等功能。其主要功能在一个线程中

循环执行,最大的工作能力与线程的运行周期有关。在不同运行周期下软件 PLC 的最大工作能力是软件 PLC 满足实时性的基础数据,如果在某一线程运行周期下用户逻辑代码超过最大值,则软件 PLC 的实时性不能得到满足。

下面讨论应用文中第 3 种测试方法,当软件 PLC 运行周期为 2ms 时,当用户逻辑程序为 600 个梯级时,软件 PLC 的最大工作能力的测试过程。测试代码如下:

```
start_time = gethrtime(CLOCK_REALTIME);
.....
if (rtl_msg.commu_state == START_COMMU)
{
do_communication(); //如果有通信要求执行通信函数
}
readIO(); //读取板卡 I/O 状态写到表中
if (rtl_msg.rung_state == START_SCAN) //如果状态为扫描则扫描梯形图
{ for (j=0 ;j<10; j++)
{
for (i=0;i<50;i++)
RefreshRung((struct StrRung *)&Rung[i]);
}
}
writeIO(); //如 I/O 状态为实时则写到板上
now = gethrtime(CLOCK_REALTIME);
elapsed_time = now - start_time;
rtl_printf("elapsed_time = %Ld \n", (long long)elapsed_time);
```

图 3 是大量数据中截取的 15 个数据,更多的数据保存在文件中。通过这些数据可以得到线程最大计算时间、计算出实时线程计算的平均时间。这些数据表明在线程运行周期在 2ms 时,软件 PLC 的最大计算时间没有超过 2ms,即软件 PLC 在 2ms 的运行周期可以运算 600 个梯级。

3 结束语

文中分析了多线程实时控制软件实时性检测的现状和难点,设计了三种多线程实时性检测方法,分析了这三种方法的优缺点。重点对使用 rtl_printf 函数和 dmesg 命令协同输出测试信息的方法进行了详细设计,给出应用实例。应用表明使用 rtl_printf 函数和 dmesg 命令协同输出测试信息的多线程实时性检测方法在需要连续采集大量数据,随机采样即可满足测试要求的场合下能够有效检测基于 RTLinux/Linux 的多

(下转第 120 页)

Date/Time	Level	Message
20090702-14:42:09	Start	JobID: 1, File: D:\STAF\services\stax\samples\sample1.xml, Machine: local://local, Function: MonitorTest, Args: { }, JobName: <N/A>
20090702-14:42:10	Info	Holding block: main
20090702-14:42:10	Info	Received RELEASE BLOCK main request
20090702-14:42:10	Info	Releasing block: main
20090702-14:42:57	Fail	Testcase: Timer.local, Pass: 4, Fail: 1, Last Status: fail, Message: value=99. Expected 0.
20090702-14:43:04	Fail	Testcase: Timer.local, Pass: 4, Fail: 2, Last Status: fail, Message: value=99. Expected 0.
20090702-14:43:22	Fail	Testcase: Timer.local, Pass: 4, Fail: 3, Last Status: fail, Message: value=100. Expected 0.
20090702-14:43:29	Fail	Testcase: Timer.local, Pass: 4, Fail: 4, Last Status: fail, Message: value=100. Expected 0.
20090702-14:44:11	Pass	Testcase: Timer, Pass: 1, Fail: 0, Last Status: pass, Message: Timer ran for 120 seconds
20090702-14:44:11	Status	Testcase: Timer, Pass: 1, Fail: 0, ElapsedTime: 00:02:00, NumStarts: 1
20090702-14:44:11	Status	Testcase: Timer.local, Pass: 0, Fail: 4, ElapsedTime: 00:03:53, NumStarts: 4
20090702-14:44:11	Status	Testcase: Totals: Tests: 2, Pass: 0, Fail: 4

图 4 日志查询界面

与实现[J]. 计算机工程, 2002, 28(9): 152 - 157.

- [4] Xin Feng, Marr S, O'Callaghan T. ESTP: An experimental software testing platform[C]//Testing: Academic and Industrial Conference Practice and Research Techniques, TAIC PART2008. [s.l.]: [s.n.], 2008: 59 - 63.
- [5] PEI Songwen, WU Baifeng, ZHU Kun, et al. Novel Software Automated Testing System Based on J2EE[J]. Tsinghua Science and Technology, 2007, 12(s1): 51 - 56.
- [6] 接 卉, 兰雨晴, 骆 沛. 一种关键字驱动的自动化测试框架[J]. 计算机应用研究, 2009, 26(3): 927 - 929.
- [7] 李夏安, 陈志泊. 基于 STAF 的软件自动化测试系统的研究和实现[J]. 计算机应用, 2009, 29(3): 699 - 704.
- [8] Tang Jingfan, Cao Xiaohua. Towards Adaptive Framework of Keyword Driven Automation Testing[C]//Proceedings of the IEEE International Conference on Automation and Logistics. Qingdao, China: [s.n.], 2008: 1631 - 1636.

(上接第 115 页)

线程实时控制软件的实时性能, 对于多线程实时软件的开发具有实用价值。

```

elapsed_time = 153000013984
elapsed_time = 153002014496
elapsed_time = 153004014208
elapsed_time = 153006014432
elapsed_time = 153008014656
elapsed_time = 153010014080
elapsed_time = 153012013952
elapsed_time = 153014014496
elapsed_time = 153016014208
elapsed_time = 153018013920
elapsed_time = 153020014464
elapsed_time = 153022014144
elapsed_time = 153024014720
elapsed_time = 153026014432
elapsed_time = 153028014080

```

图 3 扫描耗时

参考文献:

- [1] Hilton E F. Real-Time Applications with RTLinux[J/OL]. 2001 - 01 - 20. <http://www.linuxjournal.com/article/4444>.
- [2] Ji Hua, Li Yan, Wang Jian. A software oriented CNC system based on Linux/RTLinux[J]. International Journal of Advanced Manufacturing Technology, 2008, 39(10): 291 - 301.
- [3] Li B, Zhou Yin-fei, Tang Xiao-qi. Research on open CNC system based on architecture/component software reuse technology[J]. Computers in Industry, 2004, 55: 73 - 85.
- [4] 广 涛, 施 华, 尤晋元. 远程机器人控制系统的实时性能评估[J]. 计算机工程, 2004, 30(16): 99 - 101.
- [5] 陈晓明, 王治森, 董伯麟, 等. 基于 Windows CE 5.0 的嵌入式数控系统实时性研究[J]. 工业仪表与自动化装置, 2007(6): 6 - 10.
- [6] 李 江, 戴胜华. Linux 操作系统实时性测试及分析[J]. 计算机应用, 2005, 25(7): 1679 - 1680.
- [7] 褚文奎, 张凤鸣, 樊晓光. 嵌入式 Linux 系统实时性能测试研究[J]. 系统工程与电子技术, 2007, 29(8): 1385 - 1388.
- [8] Stillerman J A, Ferrara M, Fredian T W, et al. Digital real-time plasma control system for Alcator C - Mod[J]. Fusion Engineering and Design, 2006, 81: 1905 - 1910.
- [9] 朱响斌, 涂时亮. Linux 的实时性能测试[J]. 微电子学与计算机, 2004, 21(11): 85 - 88.
- [10] 张 帆. 基于 RTLinux 的硬实时性研究[J]. 武汉理工大学学报: 信息与管理工程版, 2006, 28(7): 165 - 167.