嵌入式火控软件时间性能测试

王 蕾
(中国船舶工业软件测试中心,江苏 连云港 222061)

摘 要: 嵌入式实时操作系统是能在确定的时间内执行计算或处理功能并能对外部事件做出响应的系统,由此可见时间逻辑会影响软件系统的正确性,时间指标一般指任务切换、完成时间和中断响应时间。根据这些特点,文中简要介绍了静态分析方法,着重介绍了一种利用8253可编程间隔定时器进行插装的实时动态检测时间指标的方法。此方法在嵌入式火控系统软件时间性能指标的测试中得到了充分利用,取得很好的应用效果,同时也介绍了此方法的局限性,便于在其他测试项目中更充分的运用。

关键词: 可编程间隔定时器8253; 时间性能测试; 嵌入式软件
中图分类号: TP31 **文献标识码:** A **文章编号:** 1673-629X(2014)03-0250-04
doi: 10.3969/j.issn.1673-629X.2014.03.062

Test of Embedded Fire Software Time Performance

WANG Lei
(Software Test Center of CSBI, Lianyungang 222061, China)

Abstract: Embedded real-time operating system is able to determine the amount of time to perform calculations or processing functions and can respond to external events system, which shows temporal logic will affect the correctness of software systems, the time indicators generally refers to task switching, completion time and interrupting response time. According to these characteristics, briefly describe the static analysis methods, focusing on a real-time dynamic detection time indicator method of using a programmable interval timer 8253. This method has been fully utilized in embedded fire control system software performance tests, obtaining a good application result, at the same time introduce the limitations of this approach, in order to facilitate the better use in other test projects.

Key words: programmable interval timer 8253; time performance test; embedded software

1 概 述

近年来,随着科技水平的提高,嵌入式系统因体积小、性能高、可伸缩性和性能稳定等优点,被广泛应用,因此嵌入式软件的作用和地位也正变得越来越重要。嵌入式系统的处理器运行速度越来越快,存储器的容量越来越大,存储器容量和处理器性能的提高,使得嵌入式系统成为更复杂更强大的系统软件。

嵌入式系统主要由操作系统、硬件和应用软件部分组成,同时也分为分时系统和实时系统,实时系统又分为两类:软实时系统和硬实时系统[1]。由于航空航天、通信等应用对嵌入式操作系统的性能要求较高,如果时间性能出现差错将带来重大损失或不可预料的后果,因此对嵌入式实时操作系统的性能进行评测和分析变得非常重要。

2 嵌入式系统实时性

嵌入式实时操作系统软件通常要求是强实时的。强实时系统的软件通常需要满足一定的时间性能指标约束,如果没有满足系统软件的时间约束请求处理要求,即被认为是系统的bug,所以说验证系统软件是否正确不仅仅由软件的功能要求决定,还取决于系统的各种时间性能指标,这就是实时操作系统与非实时操作系统的最大区别[2]。

与非实时系统软件相比,实时软件有如下特点:

- 响应输入时间:在规定的时间内(包括特定时刻或周期性)或在外部事件发生时刻(事件性)响应;
- 处理时间:在规定的时间内必须完成相应的处理数据;
- 输出时间:输出结果的时刻要满足对系统输出时间的要求。

收稿日期: 2013-04-22 **修回日期:** 2013-07-25 **网络出版时间:** 2014-01-07
基金项目: 国防预研基金(51319080202)
作者简介: 王 蕾(1978-),女,江苏连云港人,工程师,研究方向为软件测试和软件工程化。
网络出版地址: http://www.cnki.net/kcms/detail/61.1450.TP.20140107.1517.018.html

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

衡量实时操作系统性能的最重要指标有:

1) 任务切换时间。

多任务系统特性决定运行一个任务时,须先将当前任务的运行状态保存到自己的堆栈中,再把即将运行的任务状态从任务的堆栈区中加载到寄存器中,同时运行此任务,此过程称之为任务切换。而任务切换所需的时间快慢取决于 CPU 有多少个寄存器,并非 CPU 的寄存器越多越好,寄存器越多系统负载就越重。

2) 中断响应时间。

中断响应时间通常是指从接收到中断信号到切换到中断服务程序的时间,进行中断切换时需要调用一个特定的函数,该函数通知内核即将开始中断服务,这时内核就进行跟踪中断的嵌套,计算中断响应时间。中断响应时间是系统在最坏情况下响应中断的时间,某系统 50 次有 49 次在 40μs 之内对中断做出响应,只有一次中断响应的时间为 250μs,那么认为该中断响应时间为 250μs。

另外,还有系统响应时间(系统发出处理要求到系统做出响应给出的应答信号的时间)、最长关中断时间、非屏蔽中断响应时间等辅助的衡量指示。

时间指标特性是实时性操作系统的重要指标,也是决定软件是否正确的关键因素之一,实时性操作系统软件的时间指标通常为执行某个程序或进程所花费的时间约束,因此验证实时操作系统软件的时间性能指标的工作是尤其重要的。

对嵌入式实时操作系统软件的时间性能测试一般有 2 种测试方法,即静态时间分析和动态实时检测[3]。

3　静态时间分析

静态时间分析的定义是不运行被测软件,而直接分析源程序的架构,根据相应硬件参数预估某个进程或子程序最大的可能执行时间。对实时系统软件的时间性能测试通常要求是在最恶劣条件下,在规定时间之内完成相关任务调度。静态时间分析关键是根据不同要求来分析任务的最大执行时间,不仅要考虑单个任务的执行时间,还要考虑多个任务调度时间问题[4]。

过去几年中,单任务的最大静态时间计算与分析已经成为实时操作系统研究的重要内容之一,许多研究小组进行了大量的研究[5-6]。目前用于静态时间分析的专业化工具还不多,北航软件所研制的 C 分析和测试工具 Safepro(Ver. 2) 已经具备静态时间的最优和最劣的可视化分析,但还缺少语义层的约束。在多任务调度的静态时间分析方面也进行了相关研究,其中 Liu 和 Layland 对于强实时系统调度问题有着很深的研究,假设每个任务的执行时间是个常量[7]。

从理论上讲,只有通过静态时间分析(包括手工计算)得到程序最大的执行时间才是合理的,其他方法都是测量值,而且缺少必要的安全性,有些简单的静态时间分析工具所产生的结果也比较可靠,所需花费的时间也较少(由于不再需要构造测试用例)。更重要的一点是静态时间分析具有可预测性,对于实时性较强的操作系统软件的开发来说是非常重要的,在编码阶段就可以确定时间约束,但是大多数软件人员喜欢使用动态实时检测数据的方法。

文中就不对静态时间分析法作详细研究,以下着重介绍动态实时检测方法。

4　动态实时检测

动态实时检测方法是通过动态执行被测试的程序,以此得出相应时间数据[8]。

动态实时检测程序的最常用三种方法为在线仿真器(ICE)、程序插装和模拟器。在线仿真器可以同时度量和监视程序的运行,此方法优点是不需要修改被测试程序,就可以得到测量值,并且测量值也相对精确,但最大的不足是结果只是测量数据。模拟器也可以计算被测程序的运行时间,就相当于一个“软件在线仿真器”,与在线仿真器有同样的优点和缺点,针对这两种方法,航天总公司 204 所和北航软件所进行了测试工具的开发研究[9]。

下面着重介绍采用程序插装技术实现的嵌入式火控系统软件的时间性能动态实时检测测试方法。

火控系统软件通常在需求规格说明书中规定了一些时间性能指标,例如目标航行时间、指令处理时间、目标绪元解算时间等等,在详细设计中对某些进程或程序的动态执行时间也做了限制,关心周期进程的运行是否超标。需要对这些指标进行测试,验证是否满足要求。此时可以利用可编程间隔定时器 8253 和程序插装完成火控软件时间性能指标的测试。

4.1　定时器 8253 工作原理

用 8253 定时器来进行计数的关键部分是计数常数,8253 定时器具有 3 个独立的计数通道,每个通道的结构完全相同,如图 1 所示。8253 定时器的每一个通道都包含一个 16 位减法计数器和对应的 16 位初值寄存器、输出锁存器,减法计数器有效的计数范围为 0 ~65 535,计数器一般都是按二进制或十进制数值来进行计数,计数器速率通常可以达到 2 MHz,8253 定时器包含 6 种不同的工作方法[7]。

CLK(时钟脉冲输入端):主要用于数脉冲或时脉冲信号输入,CLK 可以是系统时钟脉冲,也可以是其他脉冲源。如果输入为周期精确的时钟,则 8253 一般

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

工作在定时方式; 如果输入是周期不定的脉冲,或只是脉冲的数量而不是脉冲的时间间隔,则此时 8253 一般作为计数器使用,因此 8253 定时器规定加在 CLK 引脚的输入时钟周期不得小于 380 ms 。

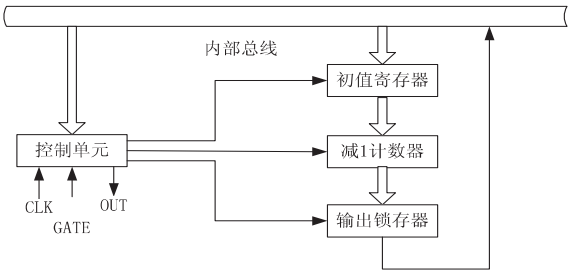

图 1 计数器内部结构

GATE (门控输入端) : 主要用于外部控制计数器的启动或停止计数,当 GATE 为高电平时,允许计数器工作; 当 GATE 为低电平时,禁止计数器工作; 当有两个或两个以上计数器连用时,可用此信号来同步,也可实现与外部信号的同步。

OUT (计数输出端) : 在不同工作方式中,当计数器计数值为 0 时, OUT 引脚上必相应地输出信号。

4.2 8253 计数器原理

8253 计数器采用加 1 和减 1 的计数方法,在控制信号有效情况下,每输入 1 个计数脉冲,通道做 1 次计数操作[7] 。在计数操作时,当计数脉冲为已知周期的时钟信号时,计数是定值,当某通道用作计数器时,将需要计数次数写入到该通道的计数器中,被计数的事件以脉冲形式从 CLK 端输入,每输入一个计数脉冲,通道的计数器值减 1,待计数器的值减为 0 时,OUT 端就做出相应的输出,表示计数器值为 0[8] 。如果某个通道用作定时器时,由 CLK 端输入一定频率的时钟脉冲,根据定时要求的时间长短确定所需的计数值,并在计数器中设置所需的数值,每输入一个时间脉冲,计数器内容减 1,待计数内容减到 0 时,OUT 端输出相应的内容,表示定时器时间到[9] 。CLK 端允许输入的时钟频率在 1 ~ 2 MHz 范围内,因此任一通道作计数器或作定时器,其内部操作完全相同,区别仅在于前者是由计数脉冲减 1 计数,而后者是内时钟脉冲进行减 1 计数[10] 。当用作计数器时,要求计数的次数直接作为计数器的初始值设置到减 1 计数器中,作定时器时,计数器的初值即定时系统根据要求定时的时间进行相应的运算才能得到,定时系统 = 需要定时的时间/时钟脉冲周期,其实计数常数 = 定时计数 = 计数初值[11] 。计算方法是一样的。

4.3 利用 8253 计数器测试时间性能

通过向实时操作系统软件程序的特定位置插入特殊探针从而记录程序的执行时间测试环境,采用宿主机 - 目标机交叉测试环境,需设计两个探针子程序:

(1) 计时开始子程序: 将定时器 8253 作为测试时钟,设置定时器 8253 选择计数器 1、工作方式 2(计数器减 1 记数) ,设置计数器初始值为 65 535[12] 。在 VxWorks 操作系统的 Tornado 开发环境下编写的 C 语言源程序为:

```
void ClockStart( )
{
    sysOutByte( 0xD6,0x74) ;
    sysOutByte( 0xD2,0xff) ;
    sysOutByte( 0xD2,0xff) ;
}
```

(2) 计时结束子程序: 读取计数器的当前值 val2。如果目标机加固 PC 机上 8253 定时器芯片的晶振频率是 1.193 18 MHz(因目标机不同可能不同) ,那么定时器记数周期为:

$T = 1/1.193\ 18\ \text{MHz} = 0.84\ \mu\text{s}$

计时结束子程序所测时间的计算公式为:

$t = (65\ 535 - \text{val2}) * 0.84/1\ 000\ \text{ms}$

相应 C 语言源程序为:

```
float  ClockEnd( )
{
    union
    {
        unsigned char ch [2];
        unsigned short Clk;
    } ClkNum;
    sysOutByte( 0xD6,0x40) ;
    ClkNum. ch [0] = sysInByte( 0xD2) ;
    ClkNum. ch [1] = sysInByte( 0xD2) ;
    return ( (65 535 - ClkNum. Clk) * 0.000 84 ) ; //单位: ms
}
```

首先定位被测程序,在该段程序前插装开始计时程序 ClockStart( ) ,在被测程序结束时插装结束计时子程序 Time = ClockEnd( ) ,在宿主机上编译通过后,下载到被测设备(目标机) 上运行程序,借助被测设备的某一显示页面或者串口终端显示最终测试结果。

4.4 改进和完善

为了减少插装程序带来的影响,使测试方法更加实用,对该方法进行了一些改进和完善:

(1) 减去插装的计时程序的执行时间。由于插装程序的执行本身要消耗一定的时间,所以测出的时间会略长,影响了测试精度,原则上应该减去这部分时间。可连续执行 ClockStart( ) 和 time = ClockEnd( ) 这两条语句,测得插装程序的执行时间。假定测试时间是 20 个周期,应该将 ClockStart( ) 最后一句修改为:

```
return ( ( (65 535 - 20) - ClkNum. Clk) * 0.000 84 ) ;
// 单位: ms
```

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

(2) 改写为汇编程序，使之具有通用性。由于用高级语言编写的插装程序运行时间长，不具有通用性，将其改写成汇编程序，在应用中只要和被测试的程序混合编译就能实现时间性能测试，汇编语言的插装程序执行时间短，对测试的影响小，具有通用性，此处汇编语言的插装源程序略。

(3) 改写为测试最大执行时间。人们往往最关心的是火控系统软件中模块或进程的最大执行时间，现将计时结束子程序作如下修改，通过执行连续循环的测试，比较计算产生最大运行时间，对其他时间可以不计算和输出，节省执行时间:

在 ClockEnd( ) 程序的开始部分增加一个内部静态变量:

```
static  unsigned  short  MinLeftClk = 65 534;
```

将最后一句改为:

```
if( ClkNum. Clk < MinLeftClk)
{ return ( (65 525 - ClkNum. Clk) * 0.000 84 ); //单位: ms
  MinLeftClk = ClkNum. Clk;
}
else  return( 0) ;
```

可见这种时间测试方法插装语句少，简单实用。

5 局限性

这种方法也受自身特点的限制:

(1) 该方法适用于单任务或多任务程序的时间测试[13]。即只能针对顺序执行的逻辑模块进行测试，也可以是一次性对多个顺序逻辑模块进行测试，测试时被测程序不能被中断，否则测试结果不准确。例如有两个优先级不同任务，任务 1 的优先级别低于任务 2 的优先级别，要求对任务 1 的执行时间进行测试，在执行任务 1 测试过程中，任务 2 被外部事件激活并运行，此时由于任务 1 优先级低被挂起，必须等任务 2 执行结束后，才可以继续执行任务 1，所以测试得到的时间包含了任务 2 的执行时间，是不合理的，这种情况下，就要分别对任务 1 和任务 2 都进行计时[14-15]。

(2) 该方法受测量时间最大值的限制。由于受 8253 定时器初始值限制，能够测量时间最大值为:

$$T_{max} = 65\ 535 * 0.000\ 84 = 55.049\ 4\ \text{ms}$$

对时间指标超过 55 ms 将无法测试，但火控系统软件的强实时性要求(进程执行时间通常为十几毫秒) 就使得这种测试方法能够很好地满足测试要求。

6 结束语

用程序插装技术检测嵌入式火控系统软件的时间性能，动态实时检测的测试方法在许多实际测试项目中得到广泛的应用，并取得了很好的测试效果。如对 XXX 火控系统软件的人机接口计算机的主控程序进行了运行时间的性能测试，在进行系统测试时发现有不满足时间指标，有可能影响自动开火、停火、转火的控制。经过开发人员进行分析和问题定位，引起问题的主要原因是战术软件和其他模块软件有重复计算的现象，程序编码需要进行优化。对主控程序做些修改和优化: 将战术软件中脱靶量变换的计算部分内容删除，直接利用任务计算机传送来的变换好结果; 利用共享变量，避免重复判断赋值; 将射击效力在线评估中的多项式计算，提取公因式，减少乘法运算次数。经过修改解决了时间超时问题，提高了软件质量，从而达到了软件测试目的。

另外，还将该时间性能的插装测试方法应用在国产嵌入式实时操作系统 DeltaOS 和 XXXX 导弹终端控制软件的测试项目中，较好地完成了测试任务。

参考文献:

[1] 郭自勇. 基于 DSP 的嵌入式多功能载波信号测试系统的研究 [D]. 长沙: 湖南大学，2004.

[2] 周 涛. 航天型号软件测试 [M]. 北京: 宇航出版社，1999.

[3] 胡明华，汤铭端. 基于分布函数的程序执行时间的静态预估 [J]. 计算机工程与设计，2006，27(16) : 3045 - 3047.

[4] 孙昌爱，靳若明，刘 超，等. 实时嵌入式软件的测试技术 [J]. 小型微型计算机系统，2000，21(9) : 920 - 924.

[5] Puschner P, Schedl A. Computing maximum task execution time - a graph - based approach [J]. Real - time systems, 1997, 13(1) : 67 - 91.

[6] Park C Y. Predicting program execution times by analyzing static and dynamic program paths [J]. Real - time systems, 1993, 5(1) : 31 - 62.

[7] Liu C L, Layland J W. Schduling algorithms for multiprogramming in a hard - real - time environment [J]. Journal of the ACM, 1983, 20(1) : 46 - 61.

[8] Bernat G, Burns A, Wellings A. Portable worstcase execution time analysis using Java byte code [C] //Proceedings of the 12th Euromicro conference on real - time systems. [s. l.]: [s. n.], 2000: 81 - 88.

[9] 王学东. 基于软件仿真的汇编语言嵌入式软件测试平台研究与实现 [D]. 成都: 电子科技大学，2006.

[10] 晏寄夫. 微机原理及应用 [M]. 成都: 西南交通大学出版社，2006.

[11] 范东丽. 嵌入式软件测试技术的研究 [D]. 哈尔滨: 哈尔滨工程大学，2006.

[12] 陈 伟. 微秒级可调脉冲电源研制及微细电解加工工艺实验 [D]. 合肥: 合肥工业大学，2006.

[13] 蒋 轶. 嵌入式雷达软件系统级测试技术研究 [D]. 南京: 南京理工大学，2009.

[14] 崔吉岗. 火控软件测试方法研究 [D]. 成都: 电子科技大学，2002.

[15] 孔祥营，柏桂枝. 嵌入式实时操作系统 VxWork 及其开发环境 Tornado [M]. 北京: 中国电力出版社，2002.

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net