

一种VxWorks系统CPU利用率图形化显示方法的设计与实现

中国船舶重工集团公司第七一六研究所 崔娟

【摘要】本文介绍了嵌入式VxWorks操作系统下一种CPU利用率图形化的显示方法。它利用系统辅助时钟硬件提供的周期性中断服务,对系统中当前现有任务的执行时间进行周期性抽样,并基于WindML多媒体库窗口系统的重绘机制,在预定义的网格化画布中,将当前时刻及其之前的所有采样点用直线相连接,实时描绘出相应曲线,为用户在目标机下提供CPU利用率的图形化观察手段,从而辅助系统运维人员方便地去诊断和解决VxWorks系统中存在的任务调度、性能瓶颈等问题。

【关键词】VxWorks; 辅助时钟; CPU利用率; WindML窗口系统

Design and implementation of a graphical display method for CPU utilization of VxWorks system

Cui Juan

(Jiangsu Automation Research Institute, Lianyungang 222061, China)

Abstract: This paper introduces a graphical display method of CPU utilization in embedded VxWorks operating system. Using periodic interrupt service which provided by system auxiliary clock hardware, periodic sampling of execution time in all running tasks. Based on the window system of WindML multimedia library, on the grid in the canvas, portrayed all the sampling points corresponding to CPU utilization value by directly connecting every point. The paper provides users with a graphical means of observation of CPU utilization in the target machine, easily helping operations engineer diagnose and solve the existing VxWorks system task scheduling problems, performance bottleneck issues etc..

Key words: VxWorks; Auxiliary Clock; CPU Utilization; WindML Windows System

0 引言

VxWorks是美国风河公司推出的在国防、工业、通信、医疗、航空、航天、物联网等领域广泛应用的操作系统,以高性能、高可靠、强实时等为其显著特点^[1]。

在嵌入式系统中,通常采用交叉方式进行开发与调试,由主机端开发工具实现对目标机的性能监控^[2]。在目标机没有特别方便的图形化系统监察手段,尤其运行态过程没有开发机的情况下,很难通过图形化手段观察系统运行情况,因此本文针对此需求研究一种在WindML窗口系统下实时观察VxWorks系统下CPU利用率的手段,为软件开发人员诊断VxWorks系统中存在的任务调度问题,性能问题提供技术参考。

1 VxWorks任务

1.1 任务

在VxWorks系统下,一个任务可以理解成一段程序的执行,系统内核通过TCB(Task Control Block)任务控制块来对每个任务当前用到的CPU各相关寄存器的值、对应的任务栈空间、任务优先级等各类任务运维所需的信息进行管理与控制。每个任务具有睡眠、就绪、运行、挂起、阻塞等状态。

1.2 任务运行调度

任务调度就是要由系统核心服务决定CPU当前需要执行哪个任务的代码,而调度的核心就是需要设计一种调度算法来为就绪等待运行的任务分配CPU资源。VxWorks操作系统通常采用基于优先级抢占式调度算法和Round-Robin轮转调度算法^[4]这两类算法相结合的方式运行多任务调度。

在一个多任务运行的生产环境下,应用程序可以创建并生成多个独立任务完成相关业务,这可以使CPU利用率得到最优发挥。VxWorks多任务Wind内核^[5]采用中断机制来驱使调度器正常运转,实现较快的上下文切换和较低的中断延迟。

2 WindML媒体库

WindML(Wind Media Library)风河媒体库是VxWorks系统下为用户提供的一个用于开发GUI图形用户界面程序的框架系统。

WindML主要由驱动开发包和软件开发包组成:DDK提供硬件驱动的开发标准接口库,用户可以开发显卡、声卡、操控等设备驱动程序,它是SDK和硬件的中间层^[6],SDK用来开发应用程序,在图元、操控、音频、字体、窗口管理及可视化控件等方面提供API。

3 任务的CPU利用率监测方法

3.1 spy的使用方法

CPU利用率其实就是运行的程序占用CPU资源的时间比例,表示机器在某个时间点的运行程序的情况。使用率越高,表明机器在这个时间上运行了很多程序,或者说程序比较耗CPU资源,反之较少。

目前有多种在嵌入式计算机操作系统中检测CPU利用率的方法,嵌入式操作系统VxWorks中主要有两种CPU占用率检测的方法:

第一种:通过创建两个特殊的任务,一个优先级最高和一个优先级最低的任务,在高优先级任务中运行一个循环次数很大的循环代码,用于占用所有的CPU时间,并记录一段时间内计数值;然后再设置一个优先级最低的任务,当所有任务都停止时CPU才会调度此任务,此任务记下在无其他任务占用CPU的情况下的计数值,则CPU占用率=(total-idle)% total。

第二种:通过操作系统自带的spy--Task activity monitor^[7],其利用辅助定时器产生中断,并且内核的任务控制块TCB会为每个被调度的任务维护一个计数器。通过记下被中断的任务,并且将该任务的计数器不断累加。经过一段时间后,每个任务的计数器就真实反映该任务占用CPU的情况。spy利用抽样技术,得到的CPU利用率的准确程度取决于任务的持续性和周期性。

3.2 spy的使用方法

以VxWorks6.9系统对应的集成开发环境Workbench 3.3为例,说明spy工具的配置及使用方法。选择镜像工程的Kernel Configuration,进入内核组件编辑配置界面。按如下路径依次点击:

Hardware(default)\peripherals\locks\Aux clock, 右键包含该组件,如图1所示为辅助时钟组件已经配置进系统。

Description	Name	Type	Value
peripherals	FOLDER_PERIPHERALS		
Detailgate FlashFX Pro Flash	FOLDER_FLASHFX		
TrueFS	FOLDER_TFFS		
USB Devices	FOLDER_USB_DEVICES		
clocks	FOLDER_CLOCK		
AUX clock	INCLUDE_AUX_CLK		
AUX clock rate min	AUX_CLK_RATE_MIN	(STR)	
AUX clock rate max	AUX_CLK_RATE_MAX	(C)	

图1 辅助时钟组件配置

Description	Name	Type	Value
spy	INCLUDE_SPY		
Number of tasks to spy	SPY_TASKS_MAX	Int	200

图2 spy监测工具配置

Name	Source	TID	PPA	Local S (chain)	chain S (chain)
task0	task0	0	0	0	0
task1	task1	1	1	1	1
task2	task2	2	2	2	2
task3	task3	3	3	3	3
task4	task4	4	4	4	4
task5	task5	5	5	5	5
task6	task6	6	6	6	6
task7	task7	7	7	7	7
task8	task8	8	8	8	8
task9	task9	9	9	9	9
task10	task10	10	10	10	10
task11	task11	11	11	11	11
task12	task12	12	12	12	12
task13	task13	13	13	13	13
task14	task14	14	14	14	14
task15	task15	15	15	15	15
task16	task16	16	16	16	16
task17	task17	17	17	17	17
task18	task18	18	18	18	18
task19	task19	19	19	19	19
task20	task20	20	20	20	20
task21	task21	21	21	21	21
task22	task22	22	22	22	22
task23	task23	23	23	23	23
task24	task24	24	24	24	24
task25	task25	25	25	25	25
task26	task26	26	26	26	26
task27	task27	27	27	27	27
task28	task28	28	28	28	28
task29	task29	29	29	29	29
task30	task30	30	30	30	30
task31	task31	31	31	31	31
task32	task32	32	32	32	32
task33	task33	33	33	33	33
task34	task34	34	34	34	34
task35	task35	35	35	35	35
task36	task36	36	36	36	36
task37	task37	37	37	37	37
task38	task38	38	38	38	38
task39	task39	39	39	39	39
task40	task40	40	40	40	40
task41	task41	41	41	41	41
task42	task42	42	42	42	42
task43	task43	43	43	43	43
task44	task44	44	44	44	44
task45	task45	45	45	45	45
task46	task46	46	46	46	46
task47	task47	47	47	47	47
task48	task48	48	48	48	48
task49	task49	49	49	49	49
task50	task50	50	50	50	50
task51	task51	51	51	51	51
task52	task52	52	52	52	52
task53	task53	53	53	53	53
task54	task54	54	54	54	54
task55	task55	55	55	55	55
task56	task56	56	56	56	56
task57	task57	57	57	57	57
task58	task58	58	58	58	58
task59	task59	59	59	59	59
task60	task60	60	60	60	60
task61	task61	61	61	61	61
task62	task62	62	62	62	62
task63	task63	63	63	63	63
task64	task64	64	64	64	64
task65	task65	65	65	65	65
task66	task66	66	66	66	66
task67	task67	67	67	67	67
task68	task68	68	68	68	68
task69	task69	69	69	69	69
task70	task70	70	70	70	70
task71	task71	71	71	71	71
task72	task72	72	72	72	72
task73	task73	73	73	73	73
task74	task74	74	74	74	74
task75	task75	75	75	75	75
task76	task76	76	76	76	76
task77	task77	77	77	77	77
task78	task78	78	78	78	78
task79	task79	79	79	79	79
task80	task80	80	80	80	80
task81	task81	81	81	81	81
task82	task82	82	82	82	82
task83	task83	83	83	83	83
task84	task84	84	84	84	84
task85	task85	85	85	85	85
task86	task86	86	86	86	86
task87	task87	87	87	87	87
task88	task88	88	88	88	88
task89	task89	89	89	89	89
task90	task90	90	90	90	90
task91	task91	91	91	91	91
task92	task92	92	92	92	92
task93	task93	93	93	93	93
task94	task94	94	94	94	94
task95	task95	95	95	95	95
task96	task96	96	96	96	96
task97	task97	97	97	97	97
task98	task98	98	98	98	98
task99	task99	99	99	99	99

图3 spy启动后的周期性CPU利用率报告

按如下路径依次点击:

development tool components\spy,右键包含该组件,如图2所示为spy监听工具组件已经配置进系统,并且可支持最多200个任务对CPU占用的情况监控。

配置并重新编译VxWorks镜像,便可以使用spy工具。在shell下输入spy命令即可启动实时监控,图3所示为某次监控报告的摘要信息。

输出的报告有两列关键信息,分别描述如下:

total列信息反映了从启动辅助时钟和监测任务开始后,每个任务的CPU活跃度

delta列信息反映了从上一次CPU利用率报告以来的CPU活跃度

spyLib库提供spyCommon函数,带一函数指针参数。函数指针类型为(int) (*p) (char * fmtStr,...)。可以通过编写自定义函数来间接获取spy需要打印的数据,spy调用该函数按行打印结果,可以不断分析每一行字符串来得到需要的结果。

```
int SpyPtnFunc(const char * fmtPtn,...)
{
    char buf[128];
    CPUUSAGEDATMSG data;
    va_list vl;
    va_start(vl,fmtPtn);
    //得到一行字符串。
    vsprintf(buf,fmtPtn,vl);
    //分析buf,得到一个任务的Cpu的使用情况。
    va_end(vl);
    return 0;
}
```

4 CPU利用率图形化方法

4.1 总体结构与计

嵌入式VxWorks系统运行时CPU利用率图形化显示方法主要由三部分组成:(1)辅助时钟中断服务;(2)周期采样任务,(3)图形化显示任务。

如图4所示为它的总体结构图。

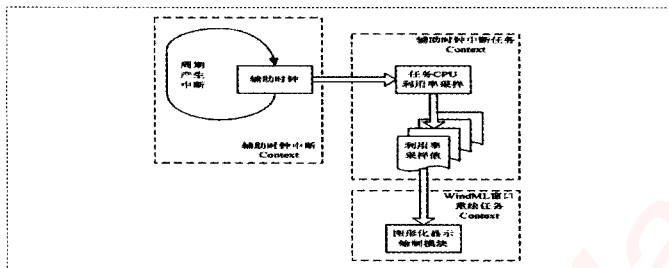


图4 总体结构图

本工具提供了实时监测任务占用CPU时间的控件类实现类的接口spyCpuStatus会发起一个任务周期调用spyCpuReport来显示每个任务、中断、内核、空闲分别消耗的CPU时间同时会显示自辅助时钟启动开始以来的CPU时间总使用情况两次spyCpuReport之间相关使用情况的变化相关CPU占用时间数据由连接到辅助时钟中断的中断级程序来收集。当前该类控件还不能用在无辅助时钟的CPU上。那些在辅助时钟中断级之上的中断不能被监测到

4.2 实施方案及关键技术

嵌入式VxWorks系统运行时CPU利用率图形化显示方法的设计主要包括以下几个方面的关键技术:

(1) 初始化各类时间变量

在任务运行上下文、中断程序运行上下文、空闲idle任务运行上下文、内核态运行上下文,都需要获取CPU资源来执行,因此需要分别为这些情况定义相关的时间变量来统计各类情况下对CPU的时间占有计数。且需要分别为它们定义两种类型的变量:统计自开始监控以来的总时间;每两次抽样之间的时间累积变化。

同时需要获取系统当前所有的任务列表,并根据每个任务的TCB任务控制块,将每个任务的运行时间统计变量tick值归零。

(2) 初始化辅助时钟

挂接辅助时钟中断程序ISR该函数在辅助时钟每产生一个滴答tick数的时候被调用,调用频率为设置的辅助时钟中断频率。ISR中断服务程序主要用于统计中断、内核空闲、内核时间、运行任务的tick计数的累加,亦即按如下情况进行处理:

如果打断一个中断,则中断tick数增加

如果打断内核运行,则内核tick数增加

如果打断空闲任务,则空闲tick数增加

如果打断就绪任务,则任务tick数增加

以上各个tick都是辅助时钟的tick,当设置辅助时钟频率200时,则1个tick为1/200秒。

(3) 周期性更新CPU利用率数据

一个优先级为5的抽样任务,该任务调用频率设置为每秒调

用一次。该任务负责累加在两次抽样之间整个系统中“辅助时钟中断”+“所有的任务”+“内核”+“中断”+“空闲任务”占用CPU的总时间,每类时间的更新方法相同,描述如下:

首先将两次抽样之间某类型的时间增量spyCpu_XXX_Inc_Ticks临时保存在临时变量tmp_XXX_Inc_Ticks中;

随后将spyCpu_XXX_Inc_Ticks清零,供辅助时钟中断继续从头累加;

然后将临时保存的增量tmp_XXX_Inc_Ticks累加至总时间量spyCpu_XXX_Total_Ticks

亦即可以得出每次采样点下的各类信息占用CPU的百分比之和,便可以给图形化程序提供数据支撑。

(4) 图形控件的结构定义

图形化的关键,是利用WindML图形系统下的窗口重绘函数进行CPU利用率的实时绘制,需要在窗口内部区域定义栅格,整个控件所需的数据结构定义如下:

```
typedef struct cpu_win_data
{
    UGL_TASK_ID taskId;
    WIN_ID winId;
    // 动态曲线框12条垂直直线x坐标值数组
    int m_VerticalX[12];
    // 动态曲线框每条垂直直线左移的步值索引
    int m_VerticalLeftMove;
    // 采样值步进的像素值
    int m_SamplingStep;
    // 动态曲线框底框所同时容纳的采样点个数
    int m_SamplingNum;
    // 已采样的次数
    int m_SamplingTimes;
    // 所有采样点的(x,y)坐标
    int m_Samplings[61][2];
    // 采样CPU的周期(单位:秒)
    int m_frequency;
} CPU_WIN_DATA;
```

(5) CPU利用率的实时绘制

每一个WindML图形窗口,均有与其对应的消息回调函数机制^[8],通过调用winCreate标准API创建窗口时,注册MSG_DRAW窗口重绘消息对应的处理回调函数,来实现对CPU利用率的绘制。以下为所需注册的窗口回调函数表,其中cbWinDraw为窗口刷新重绘函数。

```
WIN_CB_ITEM cpu_win_callbacks [] =
{
    {MSG_DRAW,0,(WIN_CB*)cbWinDraw,UGL_NULL},
    ...
    {0, 0, UGL_NULL, UGL_NULL}
};
```

在窗口重绘函数中,根据采样点的刷新频率、每次刷新的步进像素值等配置信息,利用WindML提供的以下uglXXX绘图函数:

uglLineWidthSet	设置线宽函数
uglBackgroundColorSet	设置背景色
uglForegroundColorSet	设置前景色
uglRectangle	绘制矩形
uglLine	绘制线条

通过图形编程实时将采集到的数据绘制为曲线图,窗口每次重绘的可视化逻辑流程图见图5。

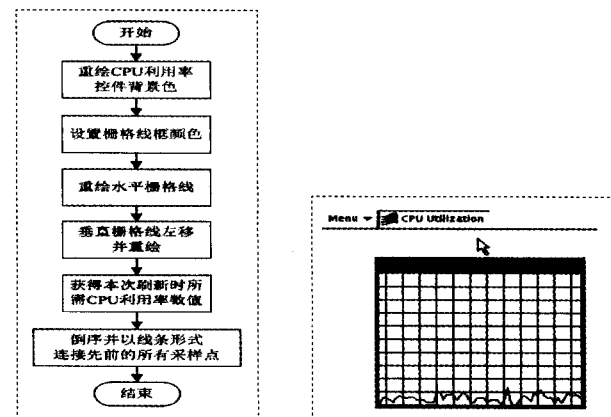


图5 CPU利用率曲线每次重绘的流程图 图6 WindML窗口下的CPU利用率绘制

(下转第133页)

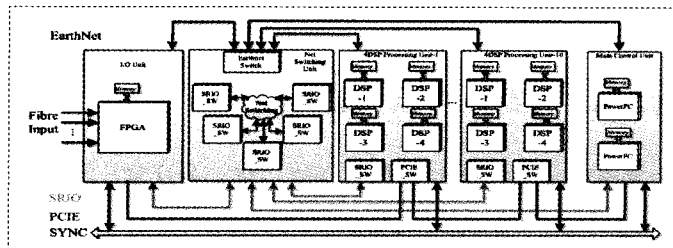


图5 高性能嵌入式实时信号处理系统总线互联图

5 总结

本文结合业界嵌入式并行处理系统的发展,对加速比扩展分析的基础上,提出一种分布式存储的嵌入式高性能并行处理架构,并基于该架构在OpenVPX 6U平台上开发了基于TMS320C6678的多DSP的并行处理板,使用该板构建了某高性能嵌入式实时信号处理系统。满足大带宽、大容量存储、高处理性能的系统需求,能较好的适应各种实时信号处理算法。充分验证了本文提出的基于分布式存储的嵌入式高性能并行处理架构具有标准化、模块化、可扩展、可重构的特点。

(上接第128页)

3.4 全球定位模块软件设计

ATGM332D-5N的多模功能定位模块的定位模式有三种:BD北斗模式,GP GPS模式,GN双模模式。为了定位准确精度更高,我们选用双模模式。定位的语句格式GGA用来输出具体的定位信息,包含与接收机定位、测时相关的数据(表1)。具体GGA语句格式:

\$--GGA,hhmmss.ss,llll.llll,a,yyyy.yyyy,a,x,xx,x,x,x,U,x,x,xxxx,x,x,x,xx*hh<CR><LF>

这部分的软件设计就是循环获取的GNGGA语句,并进行解析和处理,得到地理位置信息存储并发送到用户手机,并和原存储信息进行对比,判断地理位置信息是否发生变化。

3.5 微信端软件设计

微信端软件设计的目的是使车主用户操作方便简洁,最优的方法就是在RaspberryPI3B上安装微信客户端并自动登录,编写程序来控制微信客户端接收车主手机微信发来的命令“启动”“停止”等来启动、停止汽车安全智能防护系统。同时如果监测到不安全情况发生,能够自动给车主手机微信发送报警信息、车辆周围的图片、视频、车辆震动信息、车辆地理位置信息等。要实现以上功能,这部分软件要根据总控部分发来的状态和图片视频等自动控制微信端来给车主微信发送。为了开发效率,我们使用一个开源的微信个人号接口itchat为基础来进行Python开发,具体实现根据状态来自动发送文本、图片、视频等信息,并接收文本信息来控制系统的开启和关闭。

(上接第130页)

通过定制修改spyLib库,并集合WindML窗口系统重绘机制以及ugl绘图函数,最终实现了在VxWorks平台下图形化实时显示CPU利用率的功能,效果如图6所示。

5 结论和进一步的工作

本文介绍了嵌入式VxWorks操作系统下的一种CPU利用率图形化显示方法。它采用系统辅助时钟硬件提供的周期性中断服务,对系统中当前所有运行的任务执行时间进行周期性抽样,基于VxWorks系统的WindML多媒体图形库提供的窗口系统,在从右往左移动的网格化画布中,实时将当前以及先前所有采样点对应的CPU利用率通过直连接描绘,为用户在目标机下提供CPU利用率的图形化观察手段。该方法帮助嵌入式VxWorks系统开发人员更直观、形象的了解目标VxWorks的实时性能,以方便开发者监测和分析诊断目标操作系统运行以及执行的应用程序可能出现的各种问题。

本文给出了该方法的设计模型和一些核心技术的实现原理,完成了一个嵌入式VxWorks操作系统的CPU利用率分析工具的框架。但与Windows相比,无论从数据采集的灵活性和可配置性方面,还是对于可视化部分的自定义查看等灵活操作等方面都存在着一些差距,同时对于多核系统下的CPU利用率监控,还需要进一步的研究充实和完善。

参考文献

- [1]李方慧,王飞,何佩琨.TMS320C6000系列DSPs原理与应用[M].电子工业出版社,2005.
- [2]Sam Fuller等著,王勇,林粤伟,吴冰冰等译. RapidIO嵌入式系统互连[M].电子工业出版社,2006.
- [3]丁云霞,胡善清,龙腾.典型SAR算法在多核处理器上并行处理映射实现[J].计算机工程与应用,2012,48(S2).
- [4]杨俊,杜金榜,王跃科.基于FPGA和多DSP并行处理的可扩展数字处理终端设计[J].国防科技工业试验技术高层论坛,2007.
- [5]TMS320C6678 Multicore Fixed and Floating-point Digital Signal Processor Data Manual, November 2010.
- [6]DDR3 Design Requirements for KeyStone Devices, April 2011.
- [7]KeyStone Architecture Peripheral Component Interconnect Express User Guide, December 2010.
- [8]KeyStone Architecture Multicore Navigator User Guide, February 2011.
- [9]Virtex-6 FPGA SelectIOResources User Guide, August 16, 2010.

作者简介:

安长俊(1983—),男,江苏扬州人,硕士,江海职业技术学院讲师,从事电气自动化技术教学与研究。

4 结论

本文以RaspberryPI3B为基础,利用多种传感器模块和无线传输及地理位置定位等模块构建出一个从初始的针对汽车附件人或物体接近距离判断、语音报警提醒,中间到对接近人或物体的摄像,以及最后车辆震动和车辆位置信息变化的判断的车辆安全智能防护系统,使车主能通过自己的手机对车辆周围的状况、车辆的位置变化及有无故意破坏、是否被盗及盗贼的信息等方面完全掌握。此系统在实际运行中得到测试和检验,确定防护效果良好。此技术稍作修改也可以应用到其他的安全防护系统和监测系统中去,具有较好的推广应用前景。

参考文献

- [1]王亚军,张艳.4G通信中的关键技术之智能天线技术[J].
- [2]韩宇,张磊,吴泽民.基于嵌入式树莓派和OpenCV的运动检测与跟踪系统[J].
- [3]柯博文.树莓派(Raspberry Pi)实战指南[S].

作者简介:

姜宗元(1976—),男,陕西扶风人,软件工程硕士,研究方向:气象信息化和嵌入式开发。

参考文献

- [1]陈智育,温彦军,陈琪.VxWorks程序开发实践[M].北京:人民邮电出版社,2004.
- [2]孔祥营,张保山,俞烈彬.VxWorks驱动及分布式编程[M].北京:中国电力出版社,2007.
- [3]王金刚,官霄霖,杨锡勤.基于VxWorks的嵌入式实时系统设计[M].北京:清华大学出版社,2004.
- [4]周启平,张杨,吴琼.VxWorks开发指南与Tornado实用手册[M].北京:中国电力出版社,2004.
- [5]曹桂平,张勇,王丽娜.VxWorks设备驱动开发详解[M].北京:电子工业出版社,2011.
- [6]练学辉,朱佳丽,乔大雷.基于WindML的图形开发与应用[J].雷达与对抗,2015,35(1):65-68.
- [7]VxWorks Kernel Programmers Guide 6.9.USA: WindRiver Corporation,2015
- [8]赵刚,张群,江勇.计算机图形显示、加速及实现技术——基于VxWorks的嵌入式图形系统开发实例[M].北京:电子工业出版社,2009.

作者简介:

崔娟(1982—),女,陕西延安人,硕士,高级工程师,主要研究方向为嵌入式操作系统、平台软件。