

VxWorks 嵌入式系统远程监控系统的设计与实现

汪 毅 顾晓霞 奚宏明

(中国卫星海上测控部, 江阴 214431)

摘 要 VxWorks 广泛地应用于工业控制和航空航天等领域。分析了 VxWorks 嵌入式系统远程监控系统的功能和实现方法。该系统能够对 VxWorks 嵌入式系统的 CPU、内存等资源以及任务运行状况进行监控。综合运用 Spy 工具、异常处理机制等实现对监控信息的采集; 监控信息和控制命令的传输基于 TCP/IP 协议实现, 保证了监控的实时和可靠。

关键词 VxWorks 嵌入式系统 远程监控 异常处理

中图法分类号 TP393.09; **文献标志码** A

近年以来, 随着嵌入式计算机技术的蓬勃发展, 嵌入式操作系统在军事、航空航天、工业控制、交通运输以及金融等领域得到广泛地应用, 多数具有较高的实时性、可靠性和可用性要求。无论是在嵌入式应用的开发阶段还是运行阶段, 对嵌入式系统的监控显得越来越重要。在嵌入式应用的开发阶段, 对嵌入式系统进行监控可以帮助开发人员及时发现程序中存在的问题, 提高程序开发的效率; 而在运行阶段, 则可以使操作人员及时了解到嵌入式系统的运行信息, 从而针对不同的运行情况采取不同的措施。

目前, 市场上已经存在一些针对嵌入式系统监控的商业化工具, 如风河公司的 WindView 和 Lauterbach 的 Trace32^[1]等, 但是它们的价格比较昂贵, 并且作为应用程序开发的辅助工具, 不能实现对嵌入式系统的远程监控。基于项目研究的需求, 本文开发了针对运行 VxWorks 操作系统的嵌入式系统的远程监控工具。

1 嵌入式操作系统 VxWorks

VxWorks 操作系统是美国风河公司于 1983 年推出的一种嵌入式强实时操作系统^[2]。自问世以来, 以其高性能的内核、良好的可靠性和实时性以及友好的集成开发环境, 在嵌入式实时操作系统领域逐渐发展壮大, 被广泛应用于航空、航天等技术领域。VxWorks 的特点主要表现在以下方面^[3]:

- (1) 具有高性能可裁减的操作系统内核;
- (2) 支持多种类型结构的 CPU, 包括 x86、ARM、PowerPC、MIPS、SPARC、i960 等;
- (3) 具有友好的集成开发环境, 提供多种工具方便了开发者的开发调试工作;
- (4) 具有优秀的网络处理能力;
- (5) 兼容性良好, 兼容 POSIX1003.1b, 支持 ANSI C 标准。

2 远程监控系统的设计与实现

2.1 体系结构设计

嵌入式系统远程监控系统采用“客户/服务器”架构模式, 运行 VxWorks 5.5 的嵌入式系统作为服务器端, 运行 Windows NT 的监控端作为客户端, 服

2011年6月17日收到

第一作者简介: 汪 毅 (1985—), 男, 安徽东至人, 硕士, 助理工程师, 研究方向: 测控软件。E-mail: wangyidma@yahoo.com.cn。

务器端循环监听来自客户端的连接请求。当连接建立之后,嵌入式系统向监控端发送监视信息,周期性发送资源及任务信息、实时发送异常及告警信息,同时接收由监控端发出的控制命令并执行,系统体系结构如图 1 所示。

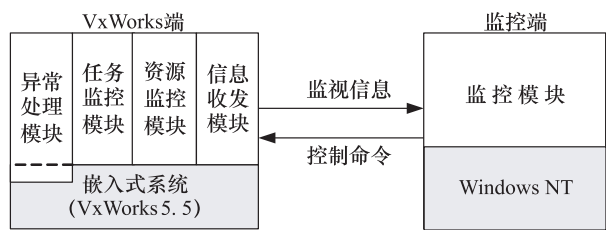


图 1 远程监控系统体系结构图

VxWorks 端作为被监控的对象,负责各类监视信息的采集和发送,以及控制命令的接收与执行,其所采集的监视信息包括:CPU 占用率、内存使用状况、任务属性、任务运行状况以及异常等信息。系统利用运行于 VxWorks 端的 6 个任务(task)协作实现监控功能,各任务介绍如下:

(1)“信息采集任务”负责采集各类监视信息,“信息发送任务”负责按照通信协议约定将采集到的信息发送到监控端,二者之间使用二进制信号量进行同步;

(2)“资源监视任务”负责监视系统当前运行的 CPU 及内存占用情况,当超过设定的告警阈值时,立即向监控端发送告警信息;

(3)“异常守护任务”负责在发生异常时,将异常信息发送到监控端。为了保证系统对异常状况的快速响应,设置该任务具有最高优先级;

(4)“命令处理任务”负责从监控端接收控制命令并执行。监控系统中的“控制命令”是指监控端发出的参数配置命令,以及发生异常时,由监控端发出的控制指令,由于命令类型较少,频率较低,所以将控制命令的接收、解释和执行放在单个任务中实现,降低了系统运行的开销;

(5)“网络监测任务”负责监测网络连接状况,发现网络连接中断时释放 socket、删除信号量,重新初始化网络,并监听新的连接。

2.2 系统功能模块设计

监视信息的采集和发送功能是监控系统的核

心功能,监控系统采用模块化设计,监视信息的采集和发送使用不同的模块设计实现。

2.2.1 信息采集模块

(1) 应用任务信息采集模块

应用任务运行时的状态信息主要包括:任务名、任务状态、CPU 占用率、任务优先级等。VxWorks 的 Spy 工具利用辅助时钟的中断服务程序,在指定的统计周期内,统计应用任务、Interrupt、Kernel、Idle 状态等所获得的 ticks 总数,以此分别统计它们的 CPU 占用率。VxWorks 的 spyLib 函数库提供了对任务名、ID、优先级等任务属性信息以及 CPU 空闲率信息进行监视的函数接口^[4]。其中,spy()函数默认将这些信息在标准输出 STD_OUT 打印,系统通过任务级的输出重定向将此类信息保存到指定的文件中,然后通过解析文件获得监视数据。此外,VxWorks5.5 内核的函数接口 spyReportCommon(FUNCPTR printRtn)在执行的过程中针对每一个任务分别调用 printRtn 函数,将任务的各项属性值赋给 printRtn 函数的参数。系统编写了 taskInfoSave()函数替代 printRtn 作为 spyReportCommon()的参数,在 taskInfoSave()中将任务的各项属性值保存到定义的结构体 taskInfoNode 中,从而实现了任务信息的采集。保存任务信息的结构体 taskInfoNode 列出如下:

```
struct taskInfoNode
{
    char bName[15]; /* 任务名 */
    int nStatus; /* 任务状态 */
    int nID; /* 任务 ID */
    int nPriority; /* 优先级 */
    int nCpu; /* cpu 占用率 */
    struct taskInfoNode *pNext;
}
```

(2) 资源信息采集模块

系统中的资源信息包括 CPU 运行情况信息和内存使用状况信息。VxWorks 提供了对 CPU 占用率和内存使用状况进行监视的函数接口。其中,CPU 占用率信息可以在 taskInfoSave()函数中处理得到;当前可用内存信息、最大分配内存信息、内存占用率信息可以通过调用 memShow()函数获取,该

函数默认将内存信息打印到标准输出,系统通过输出重定向将内存信息保存到指定的文件中,然后从中提取内存信息,代码如下:

```
char szFrMem[ MAX_MEM];
char szMaxAllcMem[ MAX_MEM];
char szBuffer[ MAX_DATA];
void * pFreeAddr;
void * pAllocAddr;
.....
/* 创建保存内存信息的文件 */
fd = open( "/ram0/meminfo", O_CREAT|O_RDWR,0x644);
/* 输出重定向到指定的文件 */
ioTaskStdSet( taskIdSelf(), STD_OUT, fd);
/* 打印内存信息 */
memShow( 0);
.....
/* 将内存信息读取到 szBuffer */
read( fd, szBuffer, MAX_DATA);
.....
/* 提取可用内存和最大使用内存信息 */
pFreeAddr = strstr( szBuffer, "free" );
pAllocAddr = strstr( szBuffer, "alloc" );
memcpy( szFrMem, pFreeAddr + 5, 10);
memcpy( szMaxAllcMem, pAllocAddr + 6, 10);
```

(3) 异常处理模块

VxWorks 提供了较为完备的异常处理机制,对于与硬件体系结构相关的异常处理,可以使用 Vx-Works 提供的接口函数来替换异常处理程序,直接获取异常硬件上下文。对于通用的异常处理, Vx-Works 提供了处理异常的通用初始化工具库,用来捕获和报告由程序错误产生的异常^[5]。VxWorks 的 excLib 库提供了 excHookAdd() 函数接口,用于将用户指定的异常处理钩子函数绑定到系统的异常处理流程中,当发生异常时由系统自动调用该函数。

为了实现对嵌入式系统异常信息的捕获和发送,系统在初始化阶段即绑定异常处理钩子函数 excHandle(),同时创建异常守护任务。异常发生后, excHandle() 首先保存异常现场信息,包括异常任务

ID、异常向量号等,然后释放信号量激活异常守护任务,由异常守护任务将异常现场信息发送到监控端,异常处理流程如图 2 所示。

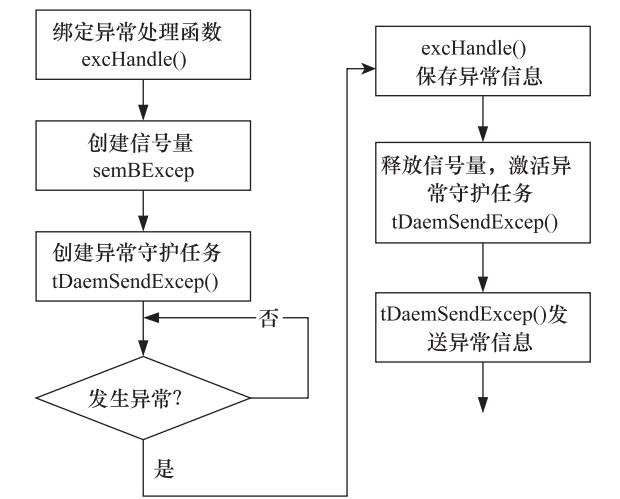


图 2 异常处理流程

系统还设计了对资源使用状况进行监视的机制,在系统初始化阶段发起一个独立的资源监视任务,定期对资源使用状况进行分析,当发现资源占用超过由监控端配置的告警阈值时,激活异常守护任务向远程监控端发送告警信息。另外,对于一些关键性任务,结合 VxWorks 提供的任务删除钩子函数 taskDeleteHookAdd(),当这些任务被异常删除时,监控系统立即采取指定处理措施,如重启该任务或向监控端发送异常信息等。

2.2.2 信息收发模块

为了保证监视信息和控制命令的可靠传输,系统基于 TCP 协议在 VxWorks 端和监控端之间建立通信连接。为了使通信双方能够正确地接收和识别对方传送的信息,二者必须遵循统一的通信规范,系统设计的监控帧格式如表 1 所示。

表 1 监控帧格式

1Byte		1Byte	<i>n</i> Byte	1Byte
类型		数据长度	有效数据 (控制命令或监视信息)	校验

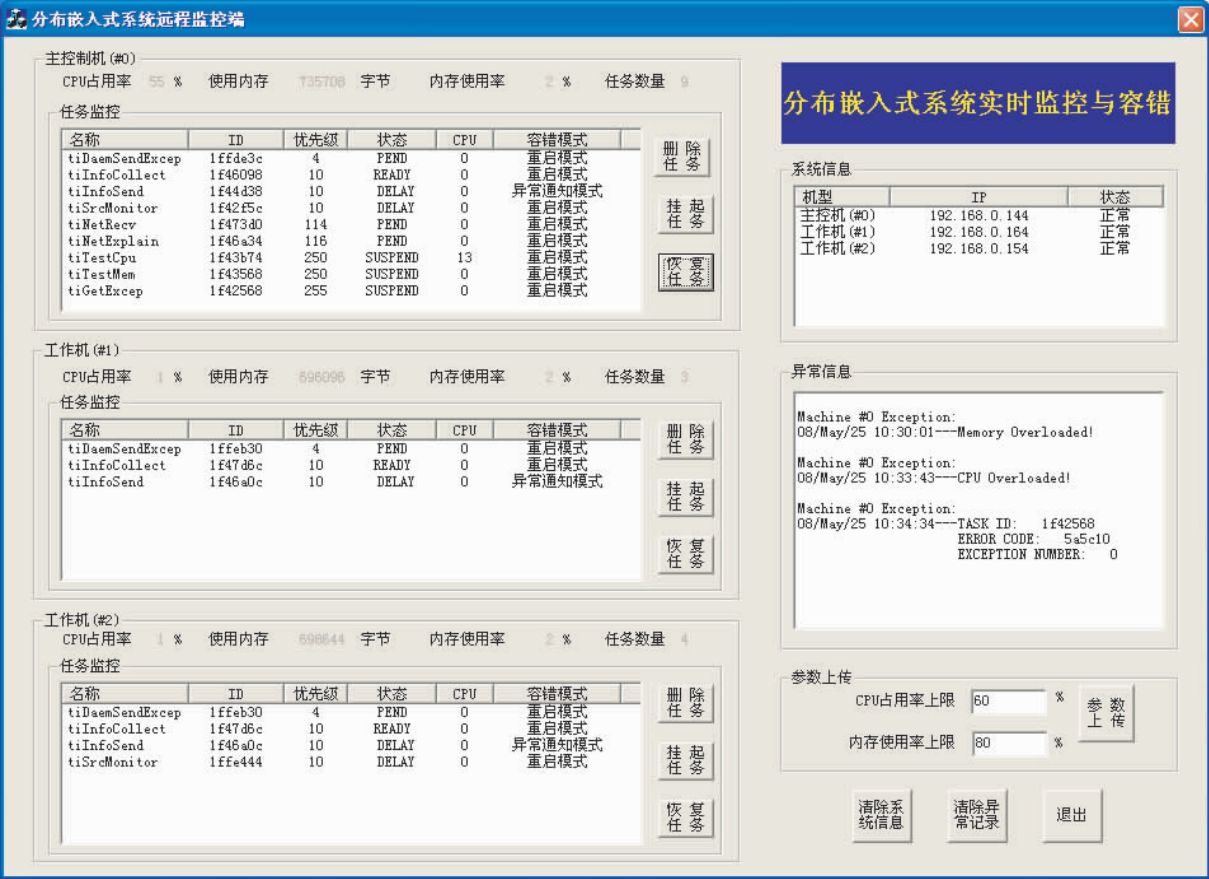


图3 监控端软件运行主界面

监控帧的第一个字段占用1字节,表示帧的类型,系统设定0x00~0x03分别代表控制命令帧、资源信息帧、任务信息帧和异常信息帧;帧的第二个字段表示有效数据的总长度,占用1字节;有效数据存放在第三个字段,系统设定:如果该帧为控制命令帧,则有效数据的第1个字节代表命令号,0x00表示参数配置命令,0x01表示重启异常任务,0x02表示终止异常任务;后续数据表示参数信息或者异常任务的ID。最后一个字段存放该帧有效数据的校验码。由于每一组数据前2字节的帧头格式固定,协议约定监控端在每次接收数据时,首先接收2字节的帧头信息,从中获取帧类型和有效数据长度,然后接收有效数据信息和校验信息。

通信过程中,当出现网络故障或者监控端异常退出时,VxWorks端将及时处理,关闭套接字,删除信号量,重新初始化网络,然后等待新的客户端连

接请求。系统设定当VxWorks端发送或者接收数据返回失败时,立即释放信号量,激活网络监测任务进行后续处理。

2.2.3 监控端软件

监控端软件运行于Windows NT平台,对嵌入式系统的资源和任务信息进行监视,完成对异常任务信息的实时捕获、对异常任务的终止或重新启动操作,以及实现对资源占用告警阈值等参数的远程配置等,监控端界面如图3所示。监控端软件在Visual C++ 6.0下实现,基于MFC CSocket与VxWorks端进行数据传输。

3 结束语

“VxWorks的嵌入式系统远程监控系统”采用
(下转第6755页)

Research on Viscosity Reducing of Asphalt Mixture Pavement Performance

MENG Yong-jun, LU Hong-xin², HUANG Xiao-ya³

(School of Civil Engineering and Transportation, South China University of Technology¹.
Guangzhou 510640, P. R. China; Guangxi Road & Bridge Construction Co. Ltd², Nanning 530001, P. R. China;
Chongqing Pengfang Pavement Engineering Research Institute³, Chongqing 400054, P. R. China)

[Abstract] Viscosity reducing modifier is a new kind of materials which is energy saving and environmental protection one, it received widely attention from the road engineering in recent years. In terms of reducing the viscosity of asphalt binder, the mechanism of reducing viscosity based on the organic viscosity reducer, and the viscosity properties of modified asphalt are discussed, and then viscosity reducing of mix asphalt mixture is studied to prove its pavement performance.

[Key words] viscosity reducing modifier modified asphalt temperature curve pavement performance

(上接第 6750 页)

VxWorks 的 Spy 工具和异常处理机制等,实现了对嵌入式系统监视信息的采集;对异常信息的捕获和发送通过钩子函数完成,保证了异常信息的实时性;利用 socket 通信,实现了监控端对嵌入式系统的监视和控制。该系统在 Tornado 和 Visual C++ 6.0 环境下编译通过,各项功能运行正常,达到了预期的目标。下一步计划继续完善系统对任务的控制功能,实现对监控任务自身的容错控制,以及实现对任务运行模式的动态配置等。

参 考 文 献

- 1 Lauterbach TRACE 32 Concepts. Lauterbach, 2007. <http://www.lauterbach.com/main.html>.
- 2 张明,郑伟,张福仲. VxWorks 中以太网通信报文的黏连问题. 单片机与嵌入式系统应用, 2007; (4): 56—59
- 3 罗国庆. VxWorks 与嵌入式软件开发. 北京: 机械工业出版社. 2003
- 4 WindRiver System Inc. Vxworks Programmer's Guide 5.5. 2004
- 5 王泽民,芦东昕,谢鑫,等. 基于 VxWorks 的异常处理的研究和实现. 计算机工程, 2005; (13): 45—48

Design and Implementation of VxWorks-based Embedded System Remote Monitoring System

WANG Yi, GU Xiao-xia, XI Hong-ming
(China Satellite Maritime TT&C Department, Jiangyin 214431, P. R. China)

[Abstract] VxWorks is being applied to a wide variety of fields such as industrial control, aeronautics and astronautics, etc. The function and implementation of VxWorks-based embedded system remote monitoring system is introduced, this monitoring system can monitor CPU usage, memory usage and tasks status of VxWorks-based embedded system. The collection of monitoring information is implemented by spy tool and exception handling mechanism of VxWorks, the transmission of monitoring information and commands is realized basing on TCP/IP, which satisfies the requirement of real time and reliability.

[Key words] VxWorks embedded system remote monitoring exception handling