石油赛伯数据采集仿真系统的实时性解决方案

李 巍 王 雷

(北京航空航天大学 计算机学院, 北京 100083)

摘 要: 为了保留石油工业赛伯(Cyber)数据采集系统的软硬件成果,在现代的高性能硬件平台上采用软件仿真的方法构造了一个仿真系统,以独立硬件设备模块为单位对实际系统进行建模,形成了以指令集仿真器为核心,通信总线仿真模块为桥梁,连接各个设备仿真模块的仿真系统. 为了满足仿真系统的期限,从系统平均性能要求和最坏性能要求两个方面对实时性能需求进行了详细的分析,提取出影响系统实时性的性能参数,根据这些性能参数的要求提出了一个实时性解决方案,解决了仿真系统与实际的数据采集设备之间的实时通信问题,协同完成了数据采集任务,在此基础上给出了一个基于 RTLinux(Real-Time Linux)实时操作系统的原型系统.

关 键 词: 指令集仿真器; 期限; RTLinux; 实时操作系统

中图分类号: TP 391.9

文献标识码: A **文 章 编 号**: 1001-5965(2006)02-0228-05

Real-time solution for Cyber data acquisition simulation system in petroleum industry

Li Wei Wang Lei

(School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: In order to reserve the valuable software and hardware fruits of the Cyber data acquisition system, the method of software simulation was adopted to build a simulation system on a modern high performance hardware platform. Petroleum industry's Cyber data acquisition simulation system was modeled by taking individual real devices as fundamental items and it shaped an architecture which contained a instruction-set simulator as a core, a communication bus simulation module as a bridge that connected each device simulation module. In order to meet the deadline of simulation system, its real-time performance was analyzed at length according to the average performance requirement and the worst performance requirement, and serveral performance parameters were extracted, then a real-time solution which based on a real-time operating system RTLinux was provided to solve the real-time communication problem between the simulation system and the real data acquisition devices.

Key words: instruction-set simulator; deadline; RTLinux; real-time operating systems

中国石油工业赛伯数据采集系统是以 PDP 11 计算机[1]为核心的一套数据采集系统,它包括计算机设备,井下仪器和测井软件系统. 此系统的功能是通过计算机控制井下仪器采集油井的地质特征数据,然后利用计算机对采集到的数据进行处理,接着将相关的输出信息发往数据显示设备和打印设备,并根据需要将信息记录在存储介质上;系统也可以从存储介质读入以前记录的数据,加以分析.

由于系统的某些硬件部件超出了使用年限或者由于操作不当而遭到损坏,甚至无法再次投入使用,这就必须投入相当大的人力财力物力对这

收稿日期: 2005-01-25

作者简介: 李 巍(1979-),男,广西梧州人,硕士生,waynee@buaa.edu.cn.

些硬件部件进行维修和保养甚至重新购买,以致增加了生产成本,加大了生产管理的复杂度.

为了降低硬件维护成本、增强生产效益,提高信息管理水平,实现油田改革的目标,设计并实现了赛伯数据采集仿真系统,完成了对 PDP11 中央处理器、系统引导 ROM、系统时钟、键盘、终端、磁带机的软件仿真,并且能够通过基于 CompatPCI 总线的 I/O 适配卡对井下仪器进行控制以及完成数据采集功能.

但是,实际的测井软件系统需要实时地对外设中断进行处理, 基于通用 Linux 操作系统所设计的赛伯数据采集仿真系统无法满足这种实时性要求,因此本文对赛伯数据采集仿真系统的实时性进行了分析与研究, 开发了一个基于 RTLinux[2] 的原型系统. 本文首先从总体上描述了赛伯数据采集仿真系统的软硬件功能;然后对仿真系统的实时性问题进行了详细的分析与研究;接着给出了整个仿真系统的实时性解决方案;最后对全文进行了总结.

1　仿真系统总体描述

1.1　赛伯数据采集系统的硬件结构

图 1 描述的是赛伯数据采集系统的硬件结构.

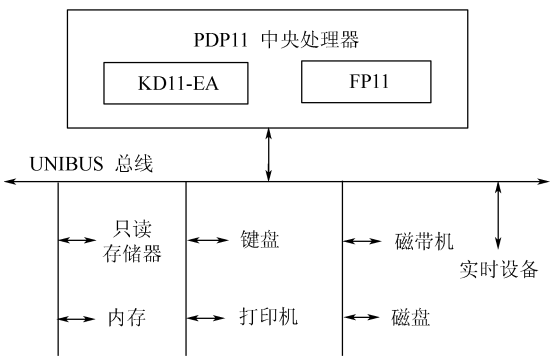


图 1　测井系统硬件结构

赛伯数据采集系统是由 PDP11 中央处理器完成计算和控制任务的,其中包括一个整数处理器 KD11-EA 和一个浮点处理器 FP11;PDP11 中央处理器通过 UNIBUS 总线与外设进行通信;外设模块包括只读存储器、内存、键盘、打印机、磁带机、硬盘和实时设备,其中实时设备负责控制和处理从井下仪器采集上来的油井地质数据,它包括光学电子设备、通用电子设备和仪器连接模块等[3].

1.2　仿真系统的运行平台

赛伯数据采集仿真系统用软件仿真的方法将赛伯数据采集系统中陈旧甚至报废的硬件部件替换成软件,这样不仅减少了硬件维护的成本,而且能够保留原系统中的测井软件知识成果,使其不加改变地运行在仿真系统之上. 图 2 给出了仿真系统的运行平台配置图.

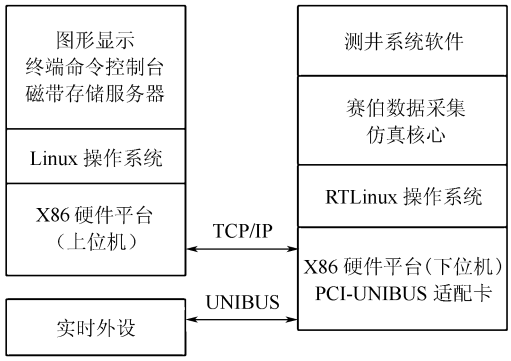


图 2　仿真系统运行平台配置图

如图 2 所示,仿真系统的运行平台采取上下位机结构,上下位机之间使用点对点对等网络连接,采用 TCP/IP 协议进行通信,下面分别阐述上下位机配置.

1) 下位机配置

下位机采用 X86 为仿真核心硬件运行支撑平台,而仿真系统上运行的测井系统软件需要连接实际的外围测井设备,系统必须能够实时地与外设交互. 由于 PDP 11 计算机是通过 UNIBUS 与实时设备进行通信的,而支撑仿真核心运行的 X86 硬件平台采用的是 PCI 总线. 因此,为了实现实时设备与仿真系统的交互,仿真系统通过“PCI-UNIBUS 适配卡”将 X86 硬件平台与实际的外围设备连接起来.

RTLinux 操作系统为仿真核心提供了软件运行支撑平台. 仿真核心通过基于 RTLinux 的 PCI-UNIBUS 适配卡驱动程序所提供的例程来对外围测井设备进行监控.

仿真核心为测井系统软件提供了一个虚拟的 PDP11 计算机运行平台,利用硬件平台提供的高性能计算,提高测井系统的运行效率.

2) 上位机配置

上位机采用的是 X86 兼容的 PC 机,利用了现代 PC 机提供的高性能,为在其上运行的 Linux 操作系统以及应用程序提供强大的硬件运算平台.

仿真系统所需要的图形显示、终端命令控制台以及磁带机存储服务器等功能都运行在上位机之上,通过 TCP/IP 协议与下位机的仿真核心进

行数据通信.

1.3 仿真核心的软件架构

由图1可以看出,如果将PDP11中央处理器也视为一个设备,那么系统呈现出一个由BUS连接的设备集合的状态.考虑到BUS的纯通信作用,系统甚至可以简化为一组定义了公共通信接口的设备对象.根据这种思想,在X86硬件平台上,基于通用Linux系统实现了一个PDP11的仿真系统[4].

图3中的仿真控制模块是仿真系统向用户提供的命令控制接口,用来控制整个仿真系统以完成各种系统功能(如系统引导、外设的装载和卸载、设置内存大小等等).CPU模块中包括了指令仿真、内存管理和中断处理三个仿真子模块,其中指令仿真子模块是在保持语义一致性的前提下,将PDP11的整型指令和浮点型指令翻译成X86指令来解释执行;中断处理子模块是仿真PDP11 CPU的中断处理功能;内存管理子模块是仿真PDP11 CPU的地址映射机制,为程序提供虚和实两种访存模式,并给系统软件提供访存权限控制.

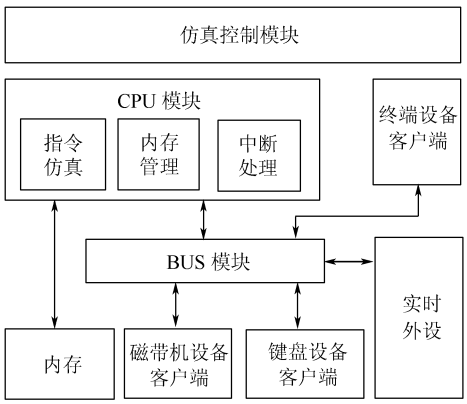


图3 仿真系统总体结构

仿真系统中的内存模块是以全局共享数组结构存在的,所以访存指令可以通过地址映射后直接访问内存,而不用通过UNIBUS,这样大大加快了访存速度.

终端、键盘、磁带机仿真设备都是以软件形式存在,为了减少下位机的处理负担,仿真系统采用客户端/服务器端(C/S)模式,在上位机实现终端显示、键盘输入以及磁带机存储服务器,而在下位机实现终端、键盘、磁带机设备的客户端,客户端和服务器之间采用TCP方式进行通信.

2 仿真系统实时性分析

根据Donald Gillies的定义,实时系统是指计算的正确性不仅与计算的逻辑正确性有关,而且与产生计算结果的时间有关.

性能约束是用术语"期限"来表示的.如果系统必须在一个时间点上,或者是一段时间间隔内对外部事件作出反应,那么把这个时间点,或者时间间隔称为期限.

2.1 仿真系统平均性能要求分析

赛伯数据采集系统中PDP11系列中央处理器的主频范围是8~15 MHz,整个系统使用单个中央处理器完成计算和控制功能,其他外设(如内存、磁带机、键盘等)都是通过UNIBUS与中央处理器进行通信.UNIBUS是PDP11系列机的一种总线通信协议标准,其中包括一个中断仲裁模块,负责管理外设中断请求的功能.在系统中有一个时钟控制中断产生的频率,使外设中断以最高为1 kHz的频率产生,所以系统要实时处理这些外设中断,如果中断未能得到及时响应,而且超出系统的容错范围的话,会导致系统崩溃.由此看出,此系统要求满足硬期限的性能约束,属于硬实时系统.

要保证仿真系统的正确性,在仿真系统设计阶段就必须考虑上述的系统性能约束.

CPU的性能是使用MIPS(Million Instructions Per Second,每秒百万次指令)来衡量的[4],假设仿真CPU的性能为P,则仿真一条PDP11的指令需要的平均时间是$1/P$.假设外设中断请求频率是FHz,因此可以求出中断请求的间隔是$1/F$.假设中断处理的平均时间是T_F,可以得出以下平均性能计算模型(见图4).

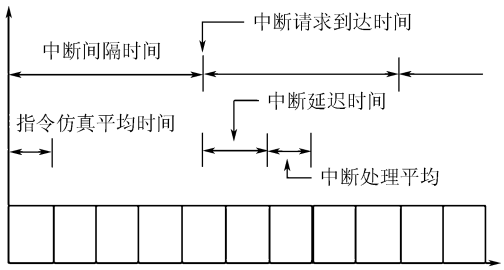


图4 平均性能计算模型

由图4可以看出,当中断请求到达的时候,系统不一定能够马上响应中断,而是首先完成中断到达时刻正在执行的仿真指令,然后再进行中断处理,因此指令仿真执行平均时间和中断处理平均时间之和不能超过中断间隔时间,这里的指令仿真执行平均时间包括了对访问外设的指令执行时间的平均,可以得到公式

$$1/P + T_F < 1/F \tag{1}$$

满足公式(1)才能保证系统的平均性能得以满足. 其中 T_F 是可以使用万用表测量出来的,F 是系统的硬件特性,而只有 P 参数需要计算. 显然仿真 CPU 性能 P 跟仿真系统硬件平台 CPU 的性能 P' 相关,P 与 P' 之间的关系如何确定?

无论使用何种软件仿真方法来进行指令仿真,都是把一条仿真 CPU 的指令解释成 N 条宿主 CPU 的指令来执行,所以执行一条仿真指令的平均时间等于执行 N 条宿主计算机指令的平均时间,可以得出公式

$1/P = N \times 1/P'$ (其中 N 是可以估算的)

或

$$P = P'/N \tag{2}$$

由式(2)代入式(1)可以得出

$$N/P' + T_F < 1/F \tag{3}$$

对于赛伯数据采集仿真系统来说,F 等于 1 000 Hz,经过测量得出 T_F 约为 50 μs,指令集仿真器平均使用 200 条指令仿真一条 PDP11 的指令,根据公式(3)可以计算出 P' 必须大于 211 MIPS. 根据这个性能指标选择了 PentiumIV 1.26 GHz 作为宿主平台的 CPU.

2.2　仿真系统最坏性能要求分析

赛伯数据采集系统在满足系统平均性能约束的基础上,还必须满足系统的期限,也就是要充分分析系统性能的最坏情况,并使用适当的方法来达到系统最苛刻的性能要求.

对于实际的赛伯数据采集系统来说,外设处理是异步的,不占用 CPU 的计算时间,只是在需要 CPU 处理外设数据的时候,给 CPU 发出中断,请求 CPU 的处理外设事件. 但是在仿真系统中,已经用软件方法将外设仿真成软件模块,而且仿真系统硬件运行平台是单 CPU 体系结构的,因此外设仿真模块(如磁带机仿真设备),需要占用宿主 CPU 计算时间;再者,仿真系统采取的是上下位机结构,上下位机之间是通过对等网络进行通信的,这样还要将网络通信的开销考虑在内.

假设仿真一条 PDP11 的指令需要的最长时间是 $T_{\max}$. 外设中断请求频率是 F Hz,因此中断请求的间隔仍是 $1/F$. 假设中断处理的最长时间是 $T_{F\max}$,仿真设备的最长处理时间是 $T_{D\max}$,网络通信延迟最大值是 $T_{N\max}$,可以得出以下的最坏性能计算模型(见图 5).

由图 5 可以看出,由于设备处理占用了很长的时间,所以大大增加了中断延迟时间,根据 UNIBUS 设计手册[3] 的描述可知,如果一次中断

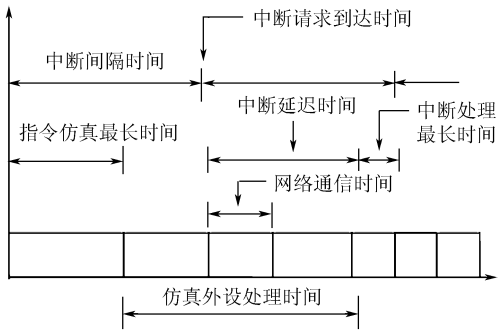


图 5　最坏性能计算模型

在请求到达之后超过 150 μs 没有得到响应的话,总线中断仲裁器将终止此次中断,而进入下一次中断请求处理. 这就意味着中断延迟时间过长会导致有效中断丢失,产生系统错误,甚至系统失效.

问题的主要矛盾是仿真设备的最长处理时间 $T_{D\max}$ 过长,影响了仿真系统的中断处理性能. 考虑到中断处理的起始时间只是依赖于上一条仿真指令的执行结束时间,而与仿真设备处理无关. 既然中断处理是仿真系统最关键的任务,因此首先要保证中断处理的实时响应. 根据这种设计思想,选取 RTLinux 实时操作系统提供的基于静态优先级的多实时任务调度功能来解决这个实时性问题.

3　基于 RTLinux 的系统设计

3.1　RTLinux 架构的概述

RTLinux 是一个遵循 POSIX 1003.13 PRE51(实时操作系统标准最小集)的实时操作系统. 它在标准 Linux 内核与计算机硬件之间增加了虚拟硬件层,标准 Linux 内核把这个虚拟硬件层看成真实的硬件. RTLinux 实现了一个完整而且可预测的实时内核,这个内核不会受 Linux 内核干扰. RTLinux 线程是通过一个基于静态优先级的调度器来调度执行的,而整个 Linux 内核和所有的普通 Linux 进程作为这个调度器的后台任务来进行调度. 因此整个通用 Linux 操作系统作为最低优先级的任务运行在 RTLinux 的实时内核之上(如图 6 所示).

3.2　仿真系统实时性实现

根据 RTLinux 实时操作系统提供的基于静态优先级的多实时任务调度机制,可以将图 5 中串行执行的任务分离出来,然后根据任务执行的紧急程度,给各个任务分配适当的优先级,接着选择适当的调度策略来控制任务的执行.

仿真系统的任务可以分成仿真控制任务、指令仿真任务(包括整型和浮点型指令仿真)、中断

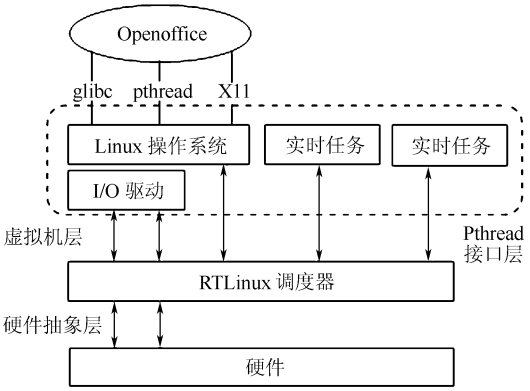


图 6 RTLinux 架构

处理任务、磁带机仿真任务、终端仿真任务、键盘仿真任务、网络通信任务,其中最紧迫的任务是指令仿真任务和中断处理任务,它们被定义成最高优先级任务,磁带机仿真任务、终端仿真任务、键盘仿真任务和网络通信任务可以被指令仿真任务和中断处理任务所打断,因此定义为次高优先级任务,而仿真控制任务是在仿真系统运行之前对系统进行配置管理的,因此可以在通用 Linux 内核上实现.

3.3 基于 RTLinux 的实现架构

下面介绍基于 RTLinux 的仿真器实时架构(如图 7 所示),“指令仿真”表示指令仿真和中断处理任务,“磁带仿真”表示磁带机仿真任务,“输入仿真”表示键盘仿真任务,“输出仿真”表示终端

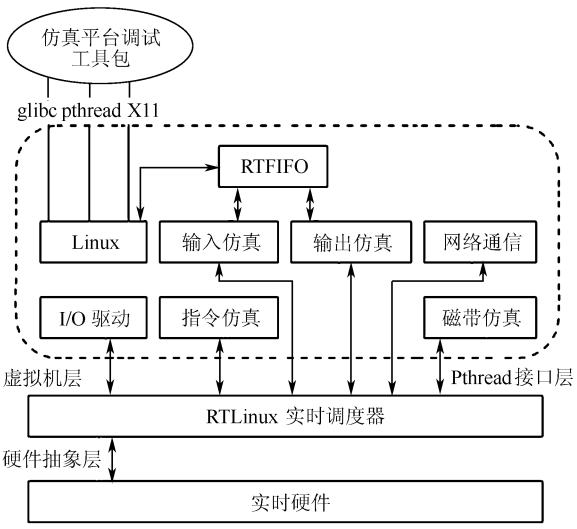


图 7 基于 RTLinux 的仿真器实时架构

仿真任务,“网络通信”表示网络通信任务,这些任务分别以 RTLinux 实时线程来实现,通过 RTLinux 提供的 pthread 接口来对任务进行创建、同步和撤销等控制.

“仿真平台调试工具包”表示仿真控制模块,它需要与键盘仿真设备和终端仿真设备进行数据通信,可以使用 RTLinux 提供的 RTFIFO[5](实时先进先出管道)建立 Linux 内核与实时任务之间的通信桥梁.

仿真系统需要与实时数据采集外设进行通信,因此必须编写基于 RTLinux 的实时设备驱动程序 RT-driver,为仿真系统提供访问实时外设的例程.

4 结 束 语

本文从总体上介绍了石油工业的赛伯数据采集系统的硬件结构;描述了赛伯数据采集仿真系统的运行平台和仿真核心的软件架构;针对仿真系统的性能约束进行详细地分析,从仿真系统中提取出影响系统性能的参数,并根据系统性能约束确定了这些参数之间的关系,得出了根据系统性能要求为运行平台选取合适处理器的方法,在此基础上研究了系统的硬期限,并提出了影响系统实时性的关键因素;最后,给出了基于 RTLinux 实时操作系统的设计与实现并且阐述了仿真器的实时架构,为解决同类的实时性问题提出了新的思路.

参考文献(References)

[1] Burnet Maxwell M, Supnik Robert M. Preserving computing's past: restoration and simulation[J]. Digital Technical Journal, 1996, 8(3): 23~38

[2] Yodaiken Victor, Barabanov Michael. A real-time linux[A]. In: Proceedings of the USENIX Conference[C]. Porland, 1997, 22(2): 6~10

[3] Wang Lei, Li Wei. The study and testing of real-time performance in a PDP11 simulator[J]. Journal of Test and Measurement Technolog, 2004, 18: 77~80

[4] Patterson David A, Hennessy John L. Measuring and reporting performance, computer architecture a quantitative approach[M]. 5th ed. Beijing: China Machine Press, 2002

[5] Dougan Cort, Sherer Matt. RTLinux POSIX API for IO on real-time FIFOs and shared memory [A]. In: The Fourth Real-Time Linux Workshop [C]. Boston, 2002