

基于虚拟机技术的 DCS 仿真系统设计与实现

王 平

(华北计算机系统工程研究所,北京 100083)

摘 要: 提出了基于虚拟机技术的 DCS 仿真系统的实现方式,描述了虚拟控制器的具体实现方法及虚拟机技术的其他应用。

关键词: 集散式控制器系统;仿真;虚拟机;指令集虚拟机

中图分类号: TP391.9

文献标识码: A

Design and implementation of DCS simulation system based on virtual machine technology

WANG Ping

(National Computer System Engineering Research Institute of China, Beijing 100083, China)

Abstract: This paper proposed DCS simulation system approach based on virtual machine technology, and described the implementation of the virtual controller. At last, it discussed other application of virtual machine technology.

Key words: distributed control system; simulation; virtual machine; instruction set architecture

随着工业的发展,DCS(Distributed Control System)的应用越来越广,但是由于 DCS 本身的分散控制集中管理的特点必然导致其存在一定的复杂性,不会像普通系统那样容易使用,所以在 DCS 产生的同时,相应的仿真系统也应运而生。

1 现有 DCS 仿真系统

DCS 仿真系统是将真实 DCS 在非 DCS 的计算机系统中重现(纯软件),不会涉及到底层硬件,完全可以运行在普通的 PC 环境中。基于这种与硬件无关的特性,仿真系统可应用于操作人员的培训和前期的工程调试。所以,DCS 仿真系统的仿真程度直接影响到后期的实际操作。

目前主要存在两种方式:(1)保留原有的集散控制系统软件和硬件,接入一个只限于实现过程仿真的仿真计算机,即激励(Stimulation)方式;(2)全范围各个部分的仿真,用软件模拟硬件的方式仿真其人机界面,而不必复制其所有功能,即仿真(Simulation)方式^[1]。

上述两种方式中,激励方式的软硬件仿真程度较高,但实现复杂,与模型连接也很困难。相对来说仿真方式比较容易实现,软件实现整个过程灵活性较强,可以运用于不同的环境,同时便于系统升级。

2 基于虚拟机技术的 DCS 仿真系统

在各个环节使用软件实现对 DCS 的仿真,然而 DCS 中很多软件已经运行在 PC 机上,尤其是操作人员使用的上位机软件,这部分完全可以直接使用,不额外仿真,具体操作在实际 DCS 中与 DCS 仿真系统中都完全一样,并使得硬件及其相关硬件也能得到完全模拟。基于虚拟机技术的 DCS 系统实现的是硬件以及与硬件相关部分的仿真,即可以在 PC 环境中再现 DCS。图 1 所示为 DCS 结构图。

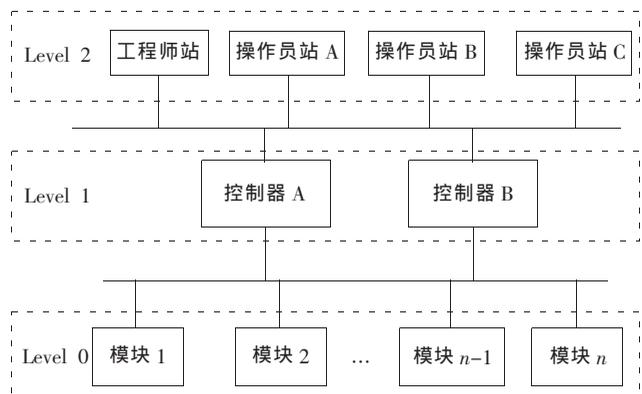


图 1 DCS 结构图

技术与方法 Technique and Method

图 1 中 Level 2 的工程师站、操作员站均运行在 PC 环境中,Level 1 及 Level 0 是硬件相关的部分。控制器是整个 DCS 中最为重要的环节,对上起着接受操作命令和反馈现场数据的作用;对下起着传递命令和采集现场数据的作用。目前最简单的控制器仿真是将控制器程序移植到 PC 环境中,这样不仅改变了控制器程序,也丢失了一部分硬件特性,同时会随着控制器程序的不断升级而频繁升级,并需要时刻保持与控制器程序对齐。

而基于虚拟机技术的 DCS 仿真系统则是把仿真目标放在了控制器本身,而不是控制器程序。在 PC 中实现一个虚拟的主控制器,控制器程序可以直接运行在虚拟控制器中,使得控制器程序的升级只需要简单的替换镜像文件。只有当硬件出现升级的时候,才需要对虚拟控制器进行升级。硬件稳定性相对较高,不会出现频繁地升级。

Level 0 的硬件模块内部逻辑比较简单,可以使用 LabVIEW 实现虚拟设备。虚拟控制器将输出值传送给虚拟设备,最终虚拟设备根据输出值在显示面板中作相应的显示,同时虚拟设备也可以将自身的某些值传给虚拟控制器。

综上所述,整个基于虚拟机技术的 DCS 仿真系统,是由 Level 2 的实际软件、Level 1 虚拟控制器以及实际的控制器软件和 Level 0 虚拟设备模块组成。由此可以看出,难点是虚拟控制器的实现。

3 虚拟控制器的实现

虚拟机的实现主要有两种方式:(1)主要由软件完全模拟目标平台的每一条指令,所以称为指令集虚拟机 ISA(Instruction Set Architecture),又因为是完全模拟所以又称模拟器;(2)借助于特定平台和 OS 环境,利用 VMM(Virtual Machine Monitor)程序为用户提供一个高效的虚拟执行环境,其主要特点就是执行效率高。由于使用了宿主操作系统的特性对功能进行了虚拟,所以称为功能级或者抽象级虚拟机,也属于一种仿真技术^[2-3]。

目前控制器的结构主要包括了 CPU、内存、网卡、Flash、DOC、双口 RAM、SRAM 以及各种端口。其中许多特定的硬件需要模拟,所以选择指令集虚拟机技术设计虚拟控制器。图 2 所示为虚拟控制器的结构图。

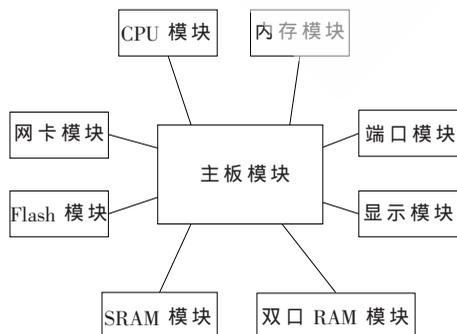


图 2 虚拟控制器结构图

图 2 中各个模块完全按照现实控制器进行划分和设计。

(1)CPU 模块包括实现指令的模拟和各个寄存器的实现。

(2)内存模块主要实现内存的模拟,其大小在虚拟控制器启动时确定,运行期间不能改变。

(3)网卡模块主要实现虚拟控制器的网卡,使得虚拟控制器可以通过网络发送和接收数据。

(4)存储模块相当于实现了虚拟机的存储介质,例如 Flash、DOC、电子盘等。

(5)SRAM 模块实现控制器用于掉电保护的 SRAM,并用于虚拟机的数据掉电保护。

(6)双口 RAM 模块主要模拟控制器与 DP 主卡交互数据的双口 RAM。

(7)端口模块实现控制器上的各个端口,虚拟控制器需要从端口读取站号、单双机状态、LED 灯状态和其他状态。

(8)显示模块主要模拟类似显示器的作用。

(9)主板模块在整个虚拟控制器起着协调作用,具有配置整个虚拟控制器的硬件特性,其中还包括定时器。

虚拟控制器在启动时,先读取硬件配置信息,如内存大小、硬盘大小、网卡数目及其配置信息。根据配置情况分配资源、初始化各个设备、读取镜像文件,然后进入指令执行阶段。此阶段为循环执行,直到虚拟控制器被执行关机操作。当有关机动作时,中断捕获后交由相应的中断处理模块。首先需要释放启动时申请的资源,然后才能退出。同时还存在其他中断,如屏幕输出、键盘输入等,具体流程如图 3 所示。

虚拟控制器不仅应用于 DCS 仿真系统中,也可以用于调试运行控制器上的嵌入式操作系统中,分析操作系统性能。只需在虚拟控制器各个模块加入统计信息,就可计算控制器程序及其嵌入式操作系统的运行情况,例如在指令集中加入统计信息,并运行操作一段时间后,即可以统计出操作系统主要指令的集中部分。这种虚拟机技术还有其他用途,比如可以做病毒分析,基于这种技术的应用(也称虚拟机技术)是一种启发式探测未知病毒的反病毒技术,能够有效地检测出未知病毒及危险代码^[4]。

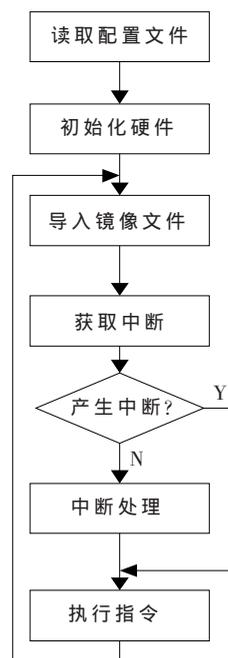


图 3 虚拟控制器运行流程图

参考文献

- [1] 王旭东. 工业过程仿真系统的 DCS 仿真交互平台设计研究[D]. 哈尔滨: 哈尔滨工程大学, 2005.
- [2] 喻强. x86 体系结构的虚拟机研究[D]. 南京: 南京工业大学, 2006.
- [3] 刘黎明, 王昭顺. 64 位虚拟机 SPANVM 的设计与实现

[J]. 计算机工程与科学, 2007, 29(2):139-141.

- [4] 曾宪伟, 张智军, 张志. 基于虚拟机的启发式扫描反病毒技术[J]. 计算机应用与软件, 2005, 22(9):125-126.

(收稿日期: 2009-09-09)

作者简介:

王平, 男, 1985 年生, 硕士研究生, 主要研究方向: 工业自动化。

