

· 微机软件 ·

基于 xPC Target 的 Windows2000 实时数据传输研究*

陈 岚¹, 万国春², 彭健飞¹

(1. 江西科技师范学院应用物理系, 南昌 330013 ; 2. 同济大学电子与信息工程学院, 上海 200092)

摘 要: 鉴于 xPC Target 平台在实时数据传输方面的缺点, 本文提出了一种重叠 Scope 技术的设计思想, 有效地解决了这个问题。

关键词: xPC Target 平台; 实时数据传输; 重叠 Scope 技术

中图分类号: TP316.7 **文献标识码:** A **文章编号:** 1002-2279(2004)05-0022-03

The Research of Windows 2000 for Real-time Data Transmission Based on xPC Target

CHEN Lan¹, WAN Guo-chun², PENG Jian-fei¹

(1. Application Physics Department, Jiangxi science and technology teachers college, Nanchang 330013, China ;

2. School of Electronic & Information Engineering, Tongji University, Shanghai 200092, China)

Abstract: In view of xPC Target workshop's shortcoming at real-time data transmission, a kind of design idea is put forward which can efficiently solve the problem.

Key words: xPC Target Workshop; Real-Time data transmission; Overlapped Scope technology

1 引 言

xPC Target 是由美国 MathWorks 公司开发的一个用于构建硬实时应用系统的 32 位平台, 它可以运行于 PC 机或者与 PC 机兼容的计算机系统上。一般来说, 由 xPC Target 平台构建的硬实时应用系统需要两台计算机, xPC Target 是一个单任务实时平台, 目前 xPC Target 平台只支持与 NE2000 或者 SMC91C9X 兼容的以太网卡, 支持 TCP/IP 通信协议^[1]。

在实时应用系统中, 为用户提供一种机制——将实时任务在运行过程中的有关数据保存为磁盘文件以便今后进行离线分析——是非常重要的。例如: 在用于产品考核的实时应用系统中, 通过这种机制, 用户就可以通过不断地在线试验、离线分析来达到完善产品设计方案的目的, 这对于产品设计人员来说是非常重要的。

xPC Target 平台提供了一种用于以上目的的数据记录(Data Logging)机制。主要特点是在实时任务的开发阶段静态地指定希望记录的有关数据以及缓冲区大小, 在实时任务的运行阶段希望记录的数据将依次被保存在相应的缓冲区中; 当实时任务停止运行时, 用户就可以通过运行于主机上的监控软件请求将以上记录的这些数据从目标机传输到主机

上, 并根据需要将它们保存为磁盘文件, 以便今后进行离线分析。但是实践当中 xPC Target 平台提供的这种机制难以满足我们的应用需要, 因为它存在着以下主要缺点^[2]: ①随着系统的运行, 所记录的数据大小将逐渐超过缓冲区的大小, 这时将会不可避免地造成数据的丢失。②在实时任务停止运行之前, 用户无法请求数据传输。

鉴于 xPC Target 平台在实时数据传输方面所存在的缺点, 本文提出了一种称为重叠 Scope 技术的设计思想来有效地解决这个问题。

2 基于 xPC Target 平台的应用程序开发与实时应用系统基本结构

2.1 xPC Target 平台应用程序开发的基本流程

xPC Target 平台没有任何开发能力, 所有的工作都应该在主机上进行。在主机上为 xPC Target 平台进行应用程序开发时, 相应的开发环境应该包括以下一些开发工具: Matlab、Simulink、Real-Time Workshop、一个第三方的 C 编译器。

一般来说, 为 xPC Target 平台开发应用程序的基本流程如图 1 所示。

2.2 基于 xPC Target 平台/Windows NT 环境的实时应用系统基本结构

* 基金项目: 江西省科技厅重点工业攻关项目(200210030)

作者简介: 陈岚(1968-), 女, 江西南昌人, 副教授, 硕士, 主要研究方向: 信号与信息处理。

收稿日期: 2003-11-20

基于 xPC Target 平台/Windows NT 环境的实时应用系统基本结构,如图 2 所示。

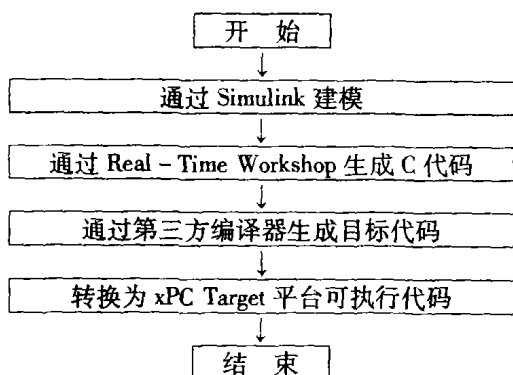


图 1 xPC Target 平台应用程序开发基本流程

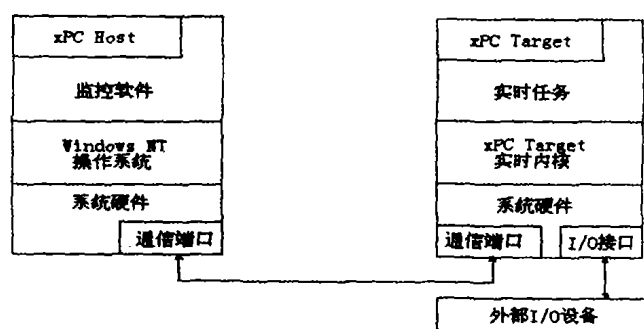


图 2 基于 xPC Target 平台/Windows NT 环境的实时应用系统基本结构

这是一个分布式实时应用系统,其基本思想就是通过一个软实时节点来管理一个硬实时节点:这种结构既可以保证实时任务在硬实时节点(xPC Target 平台)上获得足够的硬实时响应性能,又可以充分利用软实时节点(Windows NT 操作系统)所提供的其它强大功能。

实现基于 xPC Target 平台/Windows NT 环境的实时应用系统的主要优点是:系统软硬件成本比较低,实时应用程序开发方便快捷。其主要缺点就是:系统性能往往受到主机与目标机之间通信速率的限制。

3 基于 xPC Target 平台的实时数据传输的改进策略

3.1 重叠 Scope 技术

xPC Target 平台提供了一个称为 Scope 的对象,Scope 对象有两个重要特征:

(1)Scope 对象用于在实时任务的运行过程中收集数据,而且数据源以及缓冲区大小可以由运行于主机上的监控软件动态指定。

(2)Scope 对象可以由运行于主机上的监控软件请求启动和停止。

Scope 对象具有以下几种状态:SCST_WAIT-TOSTART(等待启动)、SCST_WAITFORTRIG(等待

触发)、SCST_ACQUIRING(正在收集数据)、SCST_FINISHED(数据收集完成)、SCST_INTERRUPTED(停止/中断)^[3,5]。

重叠 Scope 技术的基本思想就是通过两个重叠的 Scope 对象来实现无数据丢失的实时数据传输:

1)首先启动第一个 Scope 对象,使得它开始从指定的数据源收集数据,即进入 SCST_ACQUIRING(正在收集数据)状态。

2)在第一个 Scope 对象进入 SCST_FINISHED(数据收集完成)状态之前,主机启动第二个 Scope 对象从同样的数据源收集数据。

3)在第二个 Scope 对象收集数据期间,即处于 SCST_ACQUIRING(正在收集数据)状态期间,主机请求将第一个 Scope 对象所收集的数据从目标机传输到主机上。

4)在第二个 Scope 对象进入 SCST_FINISHED(数据收集完成)状态之前,主机重新启动第一个 Scope 对象,使得它重新开始从指定的数据源收集数据。

5)在第一个 Scope 对象收集数据期间,即处于 SCST_ACQUIRING(正在收集数据)状态期间,主机请求将第二个 Scope 对象所收集的数据从目标机传输到主机上。

6)依次循环,在主机上,监控软件负责将以上两个 Scope 对象所收集的数据正确地合并为原始数据。因为以上两个 Scope 对象所收集的数据在时间上存在着重叠区,因此我们把这种实现技术称为重叠 Scope 技术。如图 3 所示为重叠 Scope 技术的基本原理。

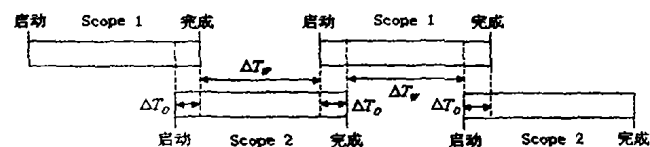


图 3 重叠 Scope 技术基本原理

3.2 数据合并算法实现流程

重叠 Scope 技术中的另外一个重要问题就是如何将两个 Scope 对象所收集的数据正确地合并为原始数据^[5],我们设想:如果在 Scope 对象的缓冲区中存在着的一组标志可以用来标识数据样本之间时序关系的话,那问题就解决了。由此,我们提出以下基本思想:在 Scope 对象的缓冲区中插入时间信息数据作为时标。实现这种思想的技术非常明确:将一个时钟信号分别指定为两个 Scope 对象的数据源之一,该时钟信号即作为标识数据样本之间时序关系的时标,如图 4。通常,作为时标的时钟信号可以为

线性时钟信号或锯齿波信号等。

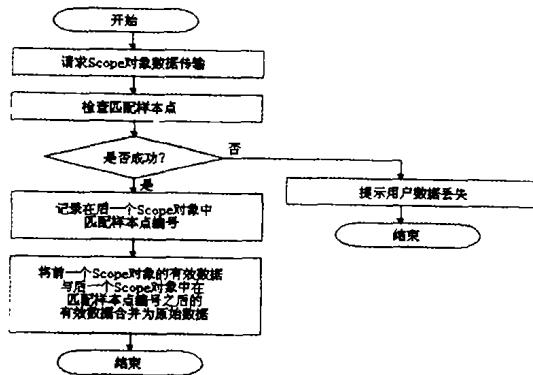


图4 数据合并算法实现流程

3.3 编程实现技术

为了提高系统的并行性和性能,在实时数据传输的实现过程中,我们采用了多线程技术;即:除主线程之外,我们专门创建另外一个线程用于管理实时数据传输以及实现相应的数据合并算法;并在必要时通过调用 Windows NT/2000 操作系统所提供的相关调度函数提高相关线程的优先级,以提高系统的响应速度,如下所示:

```

DWORD WINAPI RealtimeDataTransmissionAndProcessing
(
    LPVOID lpThreadParam
); //线程入口函数。

DWORD dwThreadId; //线程 ID 返回值。
  
```

(上接第 21 页)

所以产品资源仍须存储在传统的数据库中,这使得必须在不改变库结构的情况下集成企业原有系统的数据库,甚至是跨平台问题。解决上述问题,本方案使用 JDBC 数据库连接技术将原有数据库集成到现有的系统中。JDBC 是基于 Java 开发的数据库连接技术,它可以实现跨平台,对各种关系型数据库的访问提供了统一的 API 函数。

通过以上对企业资源的分析描述,利用现有的新兴技术,给出了可以实现对资源远程访问的解决方案,下面具体给出基于 J2EE 平台下的网络化制造中的资源管理技术模型。如图 2 所示。

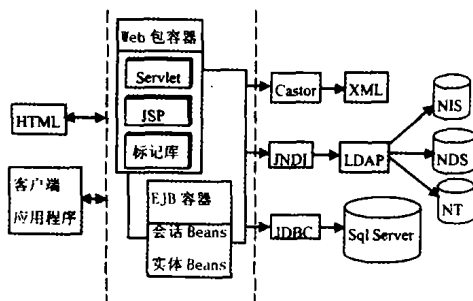


图2 网络化制造资源管理技术模型
在模型中最左面一层是客户端,可以是浏览器

```

HANDLE hThreadHandle = CreateThread( NULL, 0,
    RealtimeDataTransmissionAndProcessing,
    NULL, 0, &ThreadID ); //创建一个新线程。
SetPriorityClass( GetCurrentProcess(),
    REALTIME_PRIORITY_CLASS ); //将进程基本优先级设置
    为实时级。
SetThreadPriority( hThreadHandle,
    THREAD_PRIORITY_TIME_CRITICAL ); //将线程优先级设置
    为时间关键级。
  
```

4 结束语

本文讨论了 Windows 2000 在实时数据传输方面的改进策略,这些策略能够有效地解决实际问题且易于实现。

参考文献:

- [1] 尤晋元,史美林,陈向群,等. Windows 操作系统原理 [M]. 北京:机械工业出版社,2002.
- [2] David A. Solomon, Mark E. Russinovich 著,詹剑锋译. Windows 2000 内部揭秘 [M]. 北京:机械工业出版社,2001.
- [3] 孟毅,冯志彪. Windows2000 实时应用研究 [J]. 微机发展,2003;13(11):3-5.
- [4] Peter G. Viscarola, W. Anthony Mason 著,新智工作室译. Windows NT 与 Windows2000 设备驱动及开发 [M]. 北京:电子工业出版社,2000.
- [5] Baker A, Lozano J. Windows 著,施诺译. 2000 设备驱动程序程序设计指南 [M]. 北京:机械工业出版社,2001.

和应用程序两种形式。中间一层是包括 Servlet、JSP、标记库、EJB 等技术的 Web 服务器和应用服务器,可以采用其中的一种或几种技术来实现对后台数据库的调用。最右面一层是本文主要介绍的对资源进行描述和存储的各项技术。这里数据库采用的是 SQL Server,当然也可以采用其它任何类型的数据库,或直接调用企业现存的数据库。

5 结束语

本文给出的实现企业资源描述和共享的方案,不仅适合企业已存在某些应用系统的情况,也适合企业重新开发一套企业资源管理的软件。另外通过本方案结合 SOAP、UDDI 和 WSDL 等技术可以用来构成一个 Web 服务。

参考文献:

- [1] 严隽琪. 数字化与网络化制造 [J]. Http: www. e - works - net. cn. ,2002 - 06 - 12.
- [2] 赵博,范玉顺. 网络化制造环境下的动态企业建模 [J]. 机械工程学报,2002;38(6):31-35.
- [3] Subrahmanyam Allamaraju. J2EE 编程指南 [M]. 北京:电子工业出版社,2002.
- [4] 伊晓强. J2EE 全实例教程 [M]. 北京:北京希望电子出版社,2002.