

一种试验任务集中控制模式数据通信的 设计与实现^①

王 华 刘焕敏 顾 培 罗海艇 (中国卫星海上测控部 江苏 江阴 214431)

摘 要: 试验任务集中控制模式的数据通信, 可实现统一的数据通信接口、实战任务中对其它各部分统一调度指挥和工作流的程序化, 该模式节省人力资源、优化工作模式、有效提高任务执行的准确性和可靠性。详细对该模式数据通信的进程间通信、进程间同步、内存映射、远程代理机制和基于 XML 文档的数据通信接口等技术进行了分析探讨。

关键词: 集中控制模式; 数据通信接口; 进程间通信; 进程间同步; 内存映射; 代理机制

Design and Implementation of the Satellite Launching Mission Monitor and Display System of Data Communication in Centralized Control Pattern

WANG Hua, LIU Huan-Min, GU Pei, LUO Hai-Ting

(China Satellite Maritime Tracking and Control Department, Jiangyin 214431, China)

Abstract: Data communication in Centralized control pattern is a new method in the experimental information monitor and display software system, which can realize the functions including consolidated data-communicated interface, consolidated control, and the procedure of task flow. This method can save manpower resource, optimize task mode and improve the veracity and reliability in the satellite launching mission. This article expounds the key technology. Furthermore, process communications, process synchronization, agent mechanism and memory mapping are also discussed in detail.

Keywords: center control pattern; data communication; process communications; process synchronization; agent mechanism; memory mapping

1 引言

海上测控部试验指挥所, 早期试验任务数据通信是由数传、数据接收转发和数据显示三部分组成, 特点是各部分功能集中, 每次任务数据通信接口发生变化, 需进行软件代码维护和测试, 任务准备周期较长, 由于修改频繁的问题, 安全隐患较大。随着任务强度的增大和任务准备周期的缩小以及指挥所需求的变化, 任务实战时需对试验任务数据监视系统实时控制操作, 需将实战任务数据、首区实况和三维模拟演示上指挥所大屏投影系统, 需要对 6 台大屏投影微机进行实时控制、备份和操作。因此指挥所试验任务数据

通信采用了新型的集中控制的数据通信模式。

2 集中控制的数据通信架构

试验指挥所试验任务集中控制数据通信模式, 具备数据通信信息统一配置、数据通信接口自动生成、对初始化理论数据统一装订、对数据库统一管理及通过集中控制流在实战任务中对其它各部分统一调度指挥等特点, 可实现工作流的程序化。

集中控制数据通信分为数据流和控制流两部分。数据流从结构上分为四个部分: 数据通信、数据处理、数据显示和集中控制。前端数据通信接收从中心由

^① 收稿时间:2009-09-15;收到修改稿时间:2009-11-02

HDLC 链路发过来的数据帧,在完成校验处理后交由数据处理进行解析,转换成内部约定的数据帧格式再以广播的方式发送到内部网络上,数据显示以哑终端的形式接收该数据,并以一定的方式显示。

集中控制数据通信架构如图 1 所示。

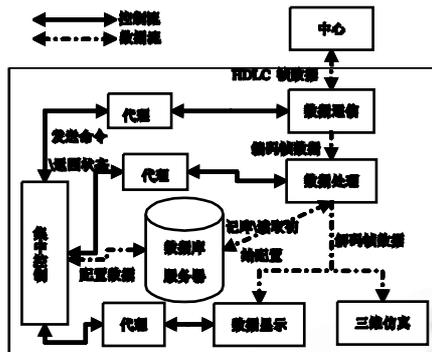


图 1 集中控制数据通信架构

控制流在任务实战过程中,集中控制通过代理服务发送控制命令,可以完成对其它部分的远程自动控制、调度指挥,比如对数据显示程序可以远程控制软件的启动、停止、重启、系统总清、画面情况、显示画面切换,还可以要求数据显示返回当前状态进行监控等等。

数据流通过数据通信和数据处理两级过滤,很好的屏蔽了底层链路数据格式的差异,给数据显示提供了统一的内部数据帧格式;如果今后底层数据链路从 HDLC 过渡到 TCP/IP 协议,或者数据编码方式发生改变,都不会影响到数据显示^[1]。

3 集中控制数据通信的设计和实现

集中控制数据通信的特点之一就是增加了集中控制数据通信部分并引入了代理机制,集中控制部分的设计和实现是集中控制数据通信的关键,集中控制数据通信部分采用 C/S 模式,使系统的程序和数据被分到不同的层次中,使系统的水平伸缩性和垂直伸缩性都得到了很大的提高^[2]。集中控制为服务器端,通过代理作为一个中间数据通信层,数据显示、通信微机、数据处理微机受控微机为客户端,代理的使用可有效提高系统的松耦合性。该模式充分发挥了 C/S 结构简洁、方便、快捷的优势,同时也保证了集中控制通信的安全性。集中控制实现模式及各部分之间的数据通信接口关系如图 2 所示。

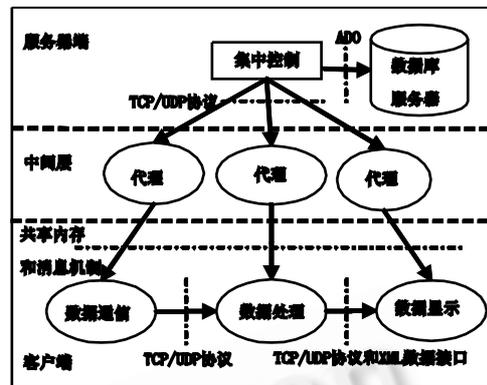


图 2 集中控制模式数据通信接口

集中控制利用 TCP/UDP 协议,通过控制代理向其他子系统发送控制命令、传递应急参数,接收其它子系统返回的状态信息,完成网络上的数据传输。集中控制代理利用共享内存和消息机制,与系统中其它子系统进行本机的控制命令和参数的传递。集中控制对数据库服务器中相关配置数据表进行维护,以供数据处理使用。对 SQL Server 数据库的访问采用 ADO 方式,数据通信部分通过 TCP/UDP 协议发送数据到数据处理,数据处理通过 TCP/UDP 协议发送数据到数据显示,以 XML 文档作为数据通信接口,进行数据通信^[3]。

4 集中控制通信关键技术点

在集中控制模式的试验任务监视显示系统数据通信中,主要使用了进程间通信、进程间同步、内存映射、远程代理机制及基于 XML 文档的数据通信接口等技术,下面将详细进行阐述。

4.1 进程间通信

Microsoft Win32 应用编程接口(Application Programming Interface,API)提供了大量支持应用程序间数据共享和交换的机制,这些机制行使的活动称为进程间通信(InterProcess Communication,IPC),进程间通信就是指不同进程间进行数据共享和数据交换^[3]。进程是装入内存并准备执行的程序,每个进程都有私有的虚拟地址空间,由代码、数据以及它可利用的系统资源(如文件、管道等)组成。每一个进程都提供了运行一个程序所必需的资源,一个进程具有 4GB 的虚拟地址空间。Windows NT Server Enterprise Edition 及 Windows 2000 Advanced Server 中低 3GB 虚拟地址空间供进程使用,高 1GB 供操作

系统的内核代码使用。Windows NT/2000 中低 2GB 供进程使用，高 2GB 供操作系统内核代码使用。Windows 9X: 0-64K 只读空间用来装入 Microsoft DOS 信息，64K-4M 装入 DOS 的兼容代码，4M-2GB 的私有空间供进程使用，2GB-3GB 的共享空间装入各种 DLL 代码，3GB-4GB 为共享的系统内核代码空间，其中共享的 2GB-4GB 的 99% 的空间是“内存无效页错误”、“General Protect Error(GPE)”，也是蓝屏的罪魁祸首。

基于 SOCKET 的 TCP、UDP 的通信，可以实现远程进程通信，它是通过规定的协议创建套接字，来实现不同机器之间的进程通信^[4]。

集中控制与代理通信，主要采用两种方式：TCP/IP、UDP 广播。TCP/IP 方式点对点通信用于对各代理软件的单独命令控制，对某一特定的机器发送命令和状态接收。UDP 方式组广播、组接收，用于对整个控制系统的命令广播。对系统中本组机器的命令发送和状态接收。

控制中心与代理，下面是一个控制中心端建立 TCP/IP 服务端的一段代码：

```
TcpListener *listener=new TcpListener (m_
iTcpPort);
// 初始化使用端口侦听对象
listener->Start();
//开始侦听
Socket* mysock=listener->AcceptSocket();
//返回侦听到的套接字
NetworkStream * sockstream=new Network-
Stream(mysock);
//提供用于网络访问的基础数据流
BinaryReader *reader=new BinaryReader(sock-
stream);
// 初始化二进制数据流读取对象
Byte m_bSockBytes[]=new Byte[buffLength];
// 创建数据缓冲区
m_bSockBytes =reader->ReadBytes (buff-
Length);
// 读取指定字节数的数据
代理客户端代码:
TcpClient *tcpclient=new TcpClient();
// 初始化 TCP客户端对象
```

```
tcpclient ->Connect(m_slp,m_sPort);
// 建立 TCP连接
FileStream *output=this->GetStream();
//获取数据流
BinaryWriter *writer=new BinaryWriter(output);
//初始化二进制方式写数据对象
BinaryReader *reader=new BinaryReader (out-
put);
//初始化二进制方式读数据对象
.....
```

4.2 进程间同步

任务执行中，出现任何一个重要指令执行错误将导致严重的后果。因此进程间同步的有效性就显得尤为重要，进程间同步的方法很多，本文采用三种实现方法。

睡眠法。就是在当一进程 A 发送消息给进程 B 通知其取映射文件内容后，线程等待几毫秒，再发送下一条命令。经测试，使用 5 毫秒的命令间隔，未出现数据丢失现象，以现在微机的速度，处理器 1 秒可以处理几亿条指令，前端总线传输速度达千兆每秒的情况下，足以保证数据的准确准时读取。

另一种方法是采用互斥对象，互斥对象属于系统全局的内核对象，多个进程可同时访问。

第三种就是消息通知，程序 A 发送完一条命令后，等待，程序 B 接收完命令后，返回一条消息，通知程序 A 发下一条命令。此方式缺点效率较低，但满足需求。

三种方法取长补短，共同利用，保证了系统可靠性和进程同步的有效机制。

4.3 代理机制

为实现集中控制与受控软件的松耦合，使用代理机制间接对受控软件进行控制。集中控制或代理一方程序单独出现故障不会产生任何对受控软件的影响，保证任务的可靠执行。实际操作中代理软件与受控软件安装在同一微机内，集中控制发送命令通过代理软件转发给受控软件。

代理与控制程序的通信特点：一是实现代理与受控软件的松耦合，任何一方程序单独出现故障，均不会影响到另一方程序的处理。二是因为内存映射文件独立于程序而存在，所以只要有一方程序未关闭，数据便不会丢失，这使得受控软件即使重新启动后，仍

可继续访问保存在映射文件的数据。代理与受控软件的通信机制如图3所示。

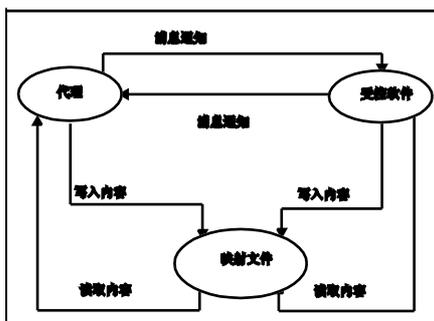


图3 代理与受控软件通信机制

4.4 内存映射

内存映射将一个进程的一段虚拟地址映射为一个文件，其它的进程可以共享该段虚拟地址，也是一种本地进程间通信的方式^[4]。Win32 API 中共享内存(Shared Memory)实际就是文件映射的一种特殊情况。进程在创建文件映射对象时用 0xFFFFFFFF 来代替文件句柄(HANDLE)。

在实现中，代理使用 0xFFFFFFFF 文件句柄创建内存映射文件，同时采用消息机制作为事件通知方式配合使用，主要是因为数据处理、监视显示实时要求高，处理器开销较大，为了减少受控软件访问共享内存的处理器开销，所以采用消息通知方式，通知程序从内存映射文件中获取数据。

内存映射创建函数如下。

```

CreateFileMapping(
(HANDLE)0xffffffff, // 内存映射文件句柄
NULL, // 内存映射文件安全属性设置为 NULL,
PAGE_READWRITE, // 映射内存对象读写模式
0, // 缓冲区高 32 位大小
iSize, // 缓冲区低 32 位大小,
lpMapFileName); // 内存映射文件名字
  
```

4.5 基于 XML 文档通信接口

软件不同部分或不同系统之间的交互接口、软件的人机接口或用户接口被称为软件接口。拥有良好的软件接口是系统具有良好扩展性、易用性的重要方面。

XML(Extensible Markup Language)文档具有跨平台、简单易用等特性，已成为一种被大量使用的通用数据格式是用来定义文档标记语言的框架，主要用来存储和转换数据信息，XML 存储数据时忽略数据类型，使得 XML

非常灵活^[5]。XML 存储业务数据时，通过解析 XML 得到业务数据，并对业务数据按一定的接口规则进行处理，如果在规则一定的情况下，业务数据发生变化时，用户可以修改 XML 或修改 XML 解析程序来实现业务与规则的一致，而不需修改接口程序，这使得用户非常方便^[6]。

本系统中，采用 XML 文档作为各分系统之间及内部的接口或配置文件在数据处理软件和数据显示软件的接口设计中，采用 XML 文件描述的数据接口，则通过读取数据帧处理方法配置数据库的方式由软件自动生成。在试验任务数据库中，存有各类信息帧处理方法的配置，这些配置信息包括数据帧内容，具体处理方法等。通过这些信息可以生成数据处理软件处理结果信息帧的内容和格式信息，即与显示软件的接口，将这些接口信息写入 XML 文件即完成了接口的自动生成。数据显示软件的 XML 接口文件获得数据内容和数据格式接口信息，并依据此文档进行页面显示控件的数据绑定。避免了与数据库服务器的交互，减轻了数据库服务器的负担，接口文件生成的主要过程如图4所示。从图中可见，数据使用者和数据产生者不再依赖于某份信息交换规定文件，而是依赖于数据库配置，因此软件代码无需随着信息交换规定文件的变化而频繁更动，降低了软件代码维护的成本。

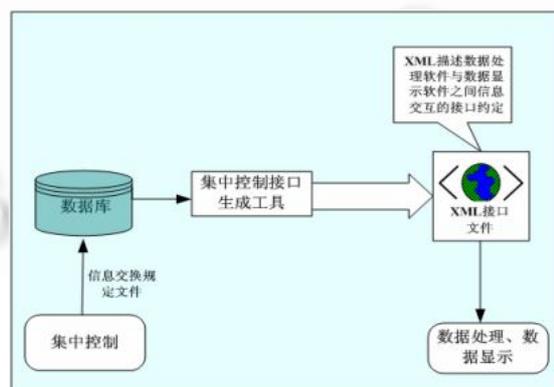


图4 基于 XML 接口

5 结语

试验任务集中控制模式的数据通信，使用代理机制实现统一的数据通信接口、实战任务中对其它各部分统一调度指挥和工作流程的程序化，经过多次实战任务证明本系统具有以下优点：实现集中控制与受控软件的松耦合，保证任务的可靠执行；在遇到未收到

(下转第 177 页)

(上接第 81 页)

特征点参数等情况下，集中控制模式可以应急输入正确的特征点，以保障任务的顺利进行；软件代码无需随着信息交换规定文件的变化而频繁更动，降低了软件代码维护的成本等等。同时这种通信模式有效的在任务实战时，对多台投影、数据通信及数据处理微机进行监控，大大节省人力资源，减少了出错率，提高了任务执行的可靠性和稳定性。

参考文献

- 1 Martin R. Agile Software Development-Principles, Patterns, and Practices.北京:清华大学出版社, 2003.
- 2 刘晓华著.精通.NET 核心技术:原理与构架.北京:电子工业出版社, 2002.
- 3 Archer T, Whitechapel A. C#技术揭密.北京:机械工业出版社, 2003.
- 4 Prosis J. Microsoft. Net 程序设计技术内幕.北京:清华大学出版社, 2004.
- 5 Dalvi D, Gray J. .NET XML 高级编程.英宇,林琪,费广正译.北京:清华大学出版社, 2002.
- 6 牛德雄,武友新.基于统一信息交换模型的信息交换研究.计算机工程与应用, 2005,41(21):185 - 197,226.

Applied Technique 应用技术 177