

面向对象的分布式测控系统

付长冬 王勇 潘清 (国防科工委指挥技术学院 101407)

摘要:本文首先介绍分布式测控系统的组成情况、软件功能模块、面向对象程序设计的基本概念;然后着重介绍基于面向对象的分布式测控系统总控软件的总体设计、系统中的基本类及其层次关系等内容。

关键词:面向对象 分布式 测控系统 继承

随着我国航天试验任务的需要,航天飞行器地面测试与控制系统由集中式处理系统发展为分布式处理系统。分布式测控系统是通过网络按客户/服务器结构将各分系统特殊测试设备(SCOE)、遥控设备(TC)、遥测设备(TM)与总控设备(OCOE)互连,各设备与总控设备相互通信,传输数据,并由总控设备集中处理各种数据,发送各种命令,监视系统状态,管理用户测试,实现统一的实时监控、测试、控制航天飞行器的地面测试控制系统。分布式测控系统软件是分布式测控系统的控制核心,由它统一管理、调度、控制系统的运行。本文首先介绍分布式测控系统的设备组成,然后说明分布式测控系统总控软件的功能模块,并介绍面向对象概念及方法,最后阐述分布式测控系统总控软件基于面向对象的实现情况。

COE)、分系统特殊测试设备(SCOE)、遥测设备(TM)、遥控设备(TC)、以太网、外围设备等设备组成。总控设备由主测试计算机(MTP)和测试控制台(TCC)等设备组成。

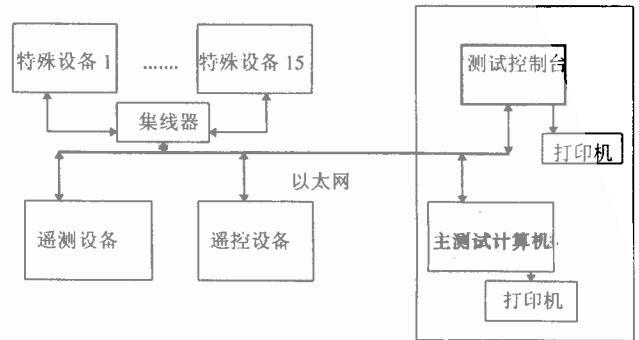


图1 分布式测控系统设备组成图

1. 分布式测控系统设备组成

如图1,分布式测控系统硬件环境由总控设备(O-

编辑按语

测控系统是计算机应用系统的一个重要方面,它的应用领域十分广泛,从航空航天等高新技术装置直至日常的民用电器设施都要涉及。本期的系统建设栏目中有几篇文章介绍了测控系统的有关技术问题。“面向对象的分布式测控系统”一文介绍了一种用于航天飞行器地面测控系统,它采用面向对象的方法进行总控软件的设计。“基于层次式并行处理机制的分面式实时测控系统”一文着重介绍了层次式并行处理机制的概念,以便在单片机环境下提供用户尽可能大的时空余量,提高实时处理能力。“洗衣机智能模糊控制系统”则是民用电器设施使用计算机控制的一个例子。

本期的通信网络栏目中“一种新的协议——XTP”一文介绍了一种网络通信中的新协议,它具有超过以往协议能适应网络发展的若干优越性能并具备许多新的功能。或许在今后得到推广,读者不妨给予适当关注。

分布式测控系统设备采用总线形式通过以太网相互连接。主测试计算机(MTP)是分布式测控系统的数据处理、测试、控制中心,被用在测试准备、确认、执行、监视和回放的全部活动中,它具有操作管理卫星系统级电气功能测试的功能。因为主测试计算机是总控软件运行和数据存储的基本平台,因而它必须是高性能的计算机。它和其他设备在功能上是客户/服务器的关系,它为其他设备提供所需要的服务。测试控制台(TCC)是分布式测控系统中提供人机接口的计算机,一方面显示来自遥测设备和分系统特殊设备的实时数据,显示报警信息,显示各种命令的确认信息,显示测试程序的运行情况,显示相关参数的信息以及其他数据。另一方面提供测试控制手段。它和主测试计算机组成分布式测控系统的总控设备(OCOE)。

遥测计算机具有遥测数据获取和遥测数据模拟的功能,遥测数据模拟功能包括编程方式数据模拟、透明方式数据模拟。遥控前端设备是在测试台的控制下将遥控指令(PSK/PCM)信息发送给卫星的设备,它可与主测试计算机(MTP)和本地控制台相连实现其功能。

每个分系统设备装入卫星之前必须进行测试以验证其功能是否正确。这些测试需要以计算机为基础的测试设备,我们把这些设备称做专用测试设备(SCOE)。

2. 总控软件功能模块

分布式测控系统软件包括总控设备上的总控软件和各前端设备的控制处理软件。总控软件是分布式测控系统软件的核心,是系统的控制中心;各前端设备的控制处理软件通过网络和总控软件通信、相互传输数据,具体处理前端设备所要处理的事务。本节主要介绍总控软件的组成模块及它们的功能特点。

如图2所示,主测试计算机的测试控制处理软件是总控软件的核心部分,包括以下子系统:(1)总控子系统;(2)控制台子系统;(3)遥测子系统;(4)遥控子系统;(5)特殊设备子系统。总控子系统又分成:系统初始化子系统、遥测数据处理子系统、命令分析与处理子系统、测试序列管理子系统等。

主测试计算机软件部分还包括后台准备软件。它包括:监视参数表文件生成程序、监视曲线表文件生成程序、遥控指令表文件生成程序、遥控块指令表文件生成程序、系统消息检查程序、系统共享内存检查程序、系统启动批处理程序、测试应用程序准备程序等。

测试控制台计算机软件包括:(1)键盘语法命令检查程序;(2)多窗口显示管理程序;(3)人机会话界面程序;(4)窗口图形显示程序;(5)网络通信程序;(6)后台准备程序;包括系统启动程序,系统检查程序等。

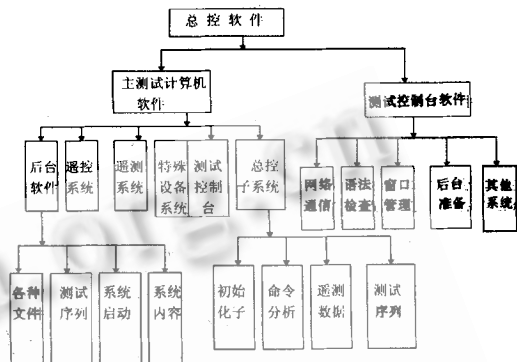


图2 总控软件包模块图

3. 面向对象程序设计

基于数据结构问题求解方法的主要欠缺是关键数据结构的作用域和可视性。在这样的系统中,许多重要的过程和函数的实现严格依赖于数据结构。如果这些关键数据结构的一个或几个有所变化,那么软件系统的结构的变化,可以扩散到一部分程序甚至整个软件系统,许多过程和函数必须重写。在这种情况下,有可能因为几个关键数据结构变化,造成软件系统整个结构崩溃。

面向对象程序设计是软件系统的设计与实现的新方法。一个面向对象的软件系统的结构是建立在一系列类上,这些类刻画了所有要处理的基本数据类型的行为。定义一个面向对象软件系统的结构框架和构成一个高层系统设计,最终只表现为一系列类(子类),它们的定义和对象。对象控制整个系统并向其他对象发送消息。用户可以通过准确定义方法的集合向该对象发送消息来处理这个对象。一个面向对象软件系统的构造不依赖于对象的内部结构,而依赖于定义这个方法,方法能在对象内部数据上处理。

面向对象程序设计是以对象和类为中心的,它将软件系统分解成基本数据类或者子类,以及每个基于类和子类的特殊定义。它的基本步骤是:

- 确定类;
- 定义属性和方法;
- 找出类间的关系;

- 把类安排出层次结构;
- 以类和对象为中心构造整个系统。

面向对象软件开发方法采用“喷泉”软件生命周期,系统一切各个阶段(分析、设计和实现)都是统一的按照相互作用的对象来组织的,各个阶段的过渡比较平滑,不会出现“瀑布”型的结构化分析和设计中从数据流图到模块图那样大的跳跃。它将软件开发过程划分为:概念模型分析、系统设计、对象设计、对象实现和系统组装等五个阶段。

概念模型分析的主要目标是建立系统模型,具体包括系统分解和对象识别。在分析阶段面向问题域建立对象模型和过程模型,进入系统设计阶段,就从问题域转入程序域,给出模型对象和过程的规范描述。在系统组装时,采用过程式程序设计语言的控制结构包括顺序、选择和重复等基本形式来构造面向对象的软件。

4. 基于面向对象的系统设计

文献[1]分析说明了分布式测控系统软件的功能要求和系统设计,本节将介绍基于面向对象的分布式测控系统的总控软件设计情况。

(1)总体设计。面向对象的分布式测控系统软件设计首先考虑的是如何确定基本类,用它实现系统的分布性、与本地子系统的数据通信和相互同步,构成满足一个用户功能性能要求的系统。

测试控制总控系统的子系统利用消息类和共享内存类进行数据通信和同步,利用网络通信类和前端设备的前端系统进行通信。消息类和共享内存类实际上是利用UNIX IPC机制实现其方法集合的,而网络通信类利用TCP/IP协议实现其内部方法的。

总控测试控制系统通过使用消息类,共享内存类进行多进程、多任务间的进程通信。由于每个子系统以独立的任务形式存在,因而每个子系统可能使用类的不同对象,但必须保证不同进程识别同一消息队列和共享缓冲区。

TCC子系统向TC子系统发送一命令消息: TCC子系统向本任务内的TC消息对象发送消息, TC子系统从本任务内的TC消息对象接收消息,而TCC子任务的TC消息对象和TC子任务的TC消息对象是不同的,如何实现他们之间的统一和数据通信?图3是消息对象通信图。

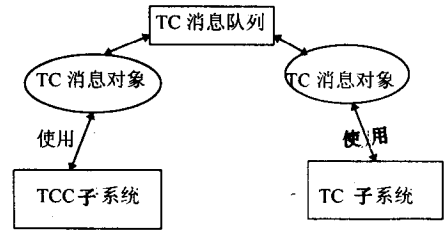


图3 消息对象通信图

通过以下方法解决上述问题:

不同任务内的同一类型消息对象使用UNIX系统的同一消息队列,在对象初始化时实现这一点,这样虽然不同对象但使用UNIX内核的同一队列,对象中的私有数据msgid, key, flags保存了系统消息队列的特征值,对象中的其他方法利用这些值进行处理。

```

message-class::message-class(exterkey)
{
    msgid = msgget(exterkey, 0);
    if (msgid <= 0)
        msgid = msgget(exterkey, IPC-CREAT|0777);
    key = exterkey;
    flags = 1;
}
else{
    key = exterkey;
    flags = 1;
}
}
    
```

共享内存类实现和消息类实现的机制是相同的,同类的不同对象通过构造函数实现指向内核中的同一共享内存区,在对象中可以用私有变量addr, shmId等标识。

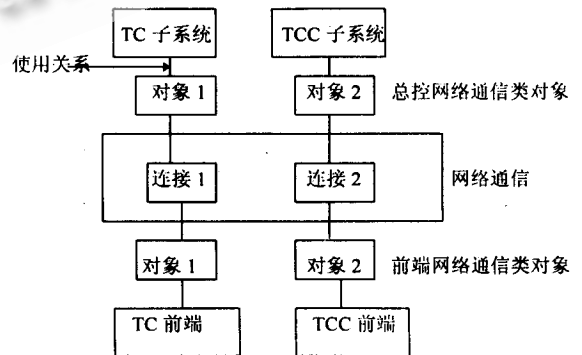


图4 网络通信类对象和子系统的关系图

对于网络通信类,它利用UNIX网络协议服务和套

接字接口实现类中的方法,同套接字接口比较,类中的方法比较简单。总控测试控制系统的子系统对象使用网络通信类对象,每个子系统有一个网络通信类对象,通过设置对象中的本地主机名,端口名标识子系统的连接地址。而前端系统通过设置网络通信类对象的远程主机名和远程端口号标识前端子系统的连接地址。这样实现的客户服务器连接是唯一的。他们的关系如图4所示。

(2)基本类及其层次关系

①准备数据类。准备数据类关系图如图5所示。

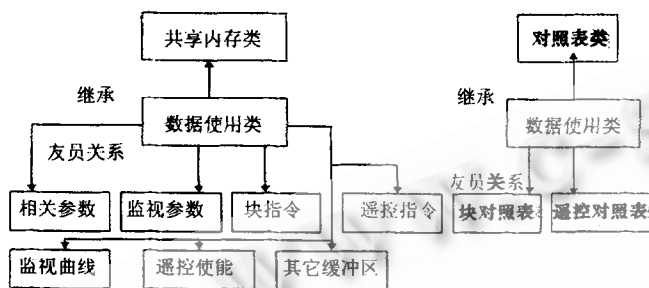


图5 准备数据类关系图

②子系统类如图6所示。

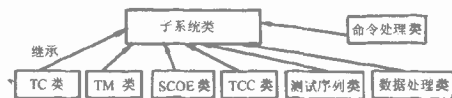


图6

每个具体子系统类它和消息类、共享内存类、网络通信类的关系如图7所示。

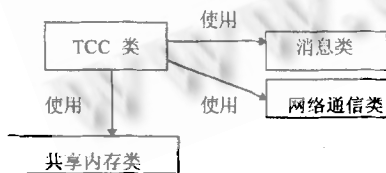


图7

5. 系统组装

在确定系统的类以及将类组织成层次结构,实现类的方法后,我们开始实现系统的功能。

组装的原则是:总控测试控制系统的子系统采用单独任务的形式实现,每个子系统采用多进程体系结构实现,具体实现按照过程设计步骤和方法。下面以TCC子系统实现为例概念性说明面向对象的多进程的系统实现。

TCC子系统处理流程:

```

|tee-class tcc-obj1;
message-class tccmessage(key1);
.....
if (fork() == 0) {
    for(;;) {
        give a message for receiving data to tccmessage
        object;
        give a send-data message to net-class object;
    }
}
for(;;) {
    give a message for receiving data from network to
    net-class object;
    give a message for sending data to 命令处理系统
    对象 to commmessage;
}
.....}
    
```

参考文献

- [1] “分布式测控系统软件的设计与实现” 付长冬, 硕士论文, 国防科工委指挥技术学院, 1996.12
- [2] “基于消息的分布式通用平台的实现” 付长冬, 计算机工程与应用, 待发表
- [3] “面向对象的软件构造” 冯玉琳等, 软件学报, 1996.3
- [4] “事件驱动的主动对象模型” 朱冰等, 软件学报, 1996.3

(来稿时间:1997年1月)