

微小型计算机

ISSN1001-8786

1997

5

开发与应用

MICRO-MINICOMPUTER DEVELOPMENT & APPLICATION

曙光信息产业有限公司
天津技术服务中心



天津市电子计算机研究所

地址：天津市河西区友谊路宾馆南道5号

ISSN 1001-8786



电话:28353894
电传:28358069
邮编:300061

欢迎订阅

《微小型计算机开发与应用》
集技术与实用性为一体，是读者的良师益友



本刊 16 开 48 页，双月刊，国内统一刊号 $\frac{\text{ISSN}1001-8786}{\text{CN}12-1122/\text{TP}}$

邮发代号 6-87 每期定价 2.40 元

编辑部地址：天津市河西区友谊路宾馆南道 5 号

邮政编码：300061

邮局代号 6-87 津工商广字 0146 国内统一刊号 $\frac{\text{ISSN}1001-8786}{\text{CN}12-1122/\text{TP}}$

定价：2.40 元

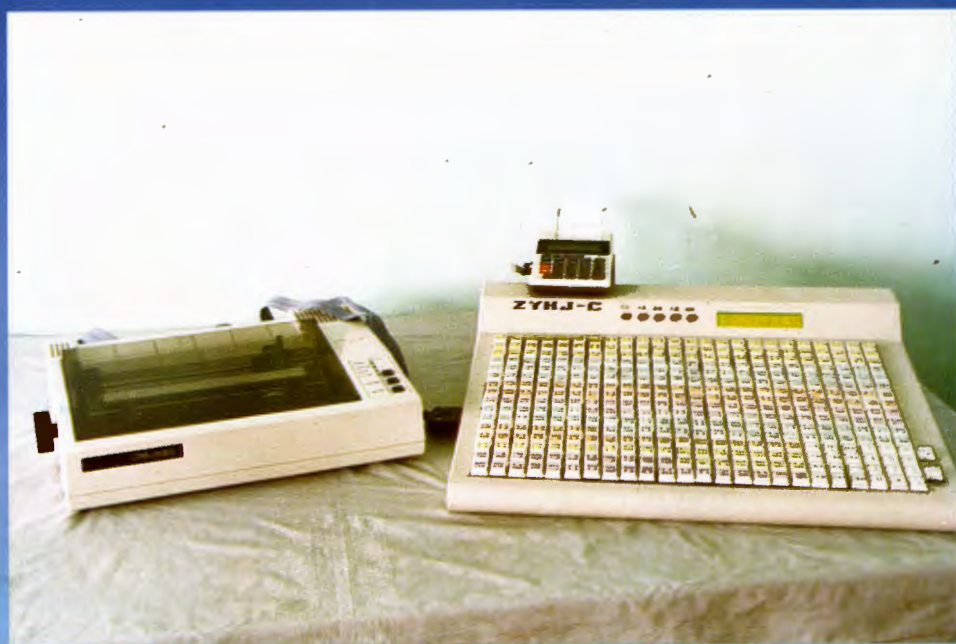
ADEPT

ZH系列打印機



天津市紅星工廠

地址：天津市河北区万柳村大街3号
电话：(022)26431738 26432821-5
电挂：4832 邮政编码：300241



ZYHJ-C 中型西药华价机简介

该机有 360 个“药名键”可记忆 720 味中西药的单价及库存量，关机后单价及库存量记忆不丢失，该机设置了“缺货键”。如某药在键盘上没有，可自行输入单价进行划价。

划价时，只要依次按：

药名键，数字键、药名键，数字键、…… 药名键，数字键、“单方结”键即可显示该方价钱，再按下打印键就能打印出药方、单价及划价金额。

改写各味药的单价及库存量操作十分方便，每味药只需 1-2 秒钟。

该机计算准确无误，可精确到小数点后第五位，单方划价结束时，厘位和毫位作四舍五入，显示到分位。

该机有八位数码管显示，单方最大计价达 9999 元。

该机具有金额连续累积功能、药品量的入库、出库、查库及结算等功能。

该机具有标准的 232 通讯口可和 PC 进行通讯。

该机具有并行打印口，可联接 CP-80 式 MX-80 打印机。

天津市电子计算机研究所

地址：天津市河西区友谊路宾馆南道 5 号

电话：(022)28137605

邮编：300061

联系人：胥大方 于金城

微小型计算机开发与应用
Weixiaoxing Jisuanji
Kaifa Yu Yingyong

1997 年第 5 期目次

双月刊 (总第 91 期)
(1981 年创刊)

专题论述

- Plug and play 原理及开发 于长云 孙 静(2)
建设现代列车通信网 刘 竞(7)

应用软件

- 防火墙技术在网络计费系统中的应用
..... 尹春华 陈 雷(11)

- 用 MFC 类库实现 Winsock 编程
..... 施 雷 王 纤(14)

- 关于 C++ 程序设计特点的探讨
..... 程为曙 刘 璟 俞 瑛(18)

- 客户/服务器模式在海洋地理信息系统中的
应用 张纪成 刘 炜(21)

实用技术

- 工业锅炉单片机监测调节和辅助管理系统
..... 张毓瑾 尹蓉华 蒋静敏(24)

- 一种由 DSP 芯片 TMS320C30 与计算机
构成的声频实时分析系统 ... 柴晓冬 李晓辉(29)

- 气体弹簧测试系统的设计 张亚新(32)

- 单片机在压力测控仪中的应用 朱一纶(34)

- 微机实时定位控制 周富国(38)

经验点滴

- 谈微机的优化内存管理 李笑梅(41)

- 简单易行通用安装盘的制作
..... 孙国锋 叶念渝(46)

- 送你一把键盘锁 罗加导(48)

主办:天津市电子计算机研究所
编辑:《微小型计算机开发与应用》编辑部
出版:天津市电子计算机研究所
天津市电子计算机学会
地址:天津市河西区友谊路宾馆南道 5 号
邮编:300061
发行:天津市邮局
印刷:天津市武清县长宏印刷厂
订购处:全国各地邮局

CONTENTS

SPECIAL TOPIC

- Principle and Exploitation of Plug and play
..... Yu Changyun Sun Jing(2)

- Building up Communication Network for Modern
Trains
..... Liu jing(7)

SOFTWARE

- The Application of File Wall Technology in Network
Accounting system
..... Yin Chunhua Chen Lei(11)

- Winsock Programming With MFC library
..... Shi Lei Wang Xian(14)

- The Discussion on The Feature of C++ Program-
ming
..... Cheng Weishu Liu Jing Yu Ying(18)

- Application of Client/Server in Marine Chart Infor-
mation System
..... Zhang Jicheng Liu Wei(21)

- Zhang Jicheng Liu Wei(21)

APPLICATION

- A Single-Chip Microcomputer scan, Automatic con-
trol and Aided Management System for Industrial
Boiler
..... Zhang Yujin Yin Ronghua(24)

- A Real Time Sound Spectrum Analysis system com-
posed With TMS 320C30 and Microcomputer
..... Chai Xiaodong Li Xiaohui(29)

- Design of air Spring Testing System
..... Zhang Yaxin(32)

- Application of Single chip Microcomputer in Measur-
ing and controlling Pressure
..... Zhu Yilun(34)

- Computer Real-time Location Control
..... Zhou Fuguo(38)

- Zhou Fuguo(38)

- Zhou Fuguo(38)

LEARNING COLUMN

- Optimal Memory Management of Microcomputer
..... Li Xiaomei(41)

- Simple Formahing of General Installation Disk
..... Sun Guofeng Ye Nianyu(46)

- Providing You A key-Lock
..... Luo Jiadao(48)

Plug and Play 原理及开发

天津理工学院计算中心 于长云 孙 静

摘 要 Plug and Play 包含硬件及软件标准。本文总结 Plug and Play 的硬件机制,介绍 Configuration Manager 的管理功能,并且简述了如何开发 Windows95 的 VxD 虚拟驱动程序。

关键词 即插即用 VxD 设备驱动程序

1 总线工作机制

由于设计上缺乏远见,遵循 ISA (Industry Standard Architecture) 标准的 PC 扩展卡往往附设 DIP 开关或跳线开关,为了使扩展卡能够重新配置,用户必须手工改动 DIP 或跳线开关以便指定 IRQ, DMA, 端口号及内存地址等。由于提高性能的需要,新的总线标准不断推出,同时识别及配置扩展卡的方案也相应出台。

EISA (Extended Industry Standard Architecture)、MCA (Micro Channel Adapter)、PCI (Peripheral Component Interconnect) 这三种主流总线标准都各自为政拥有不同的扩展卡识别方法。其中一个重要的识别信息是指定卡类型及制造商的代码。

实际上任何软件,包括操作系统、BIOS 及配置实用工具等,都可以读取该标识码,各种软件通过为扩展卡预留的 I/O 口地址与 EISA 卡通讯。因此,在 2 号槽的 EISA 卡使用口地址 2000—2FFF,通过读取 2C80—2C83 范围每个口的一个字节,软件可获得 32 位的标识码,该标识码包含用于指定制造商的 3 个 ASCII 字符以及一个标识设备的 16 进制数。(ASCII 字符仅用低 5 位加以区别,示例所示均为大写字母低 5 位二进制代码)。

软件通过扩展卡标识码来确定扩展槽中的插卡。用户第一次安装某个 EISA 卡时,必

运行实用程序配置该卡。配置程序根据随卡提供的配置文件为插卡分派资源,同时把配置信息完整地写入 CMOS 内存中。下一次引导机器时,系统 BIOS 根据 CMOS 中的信息初始化所有 EISA 卡。

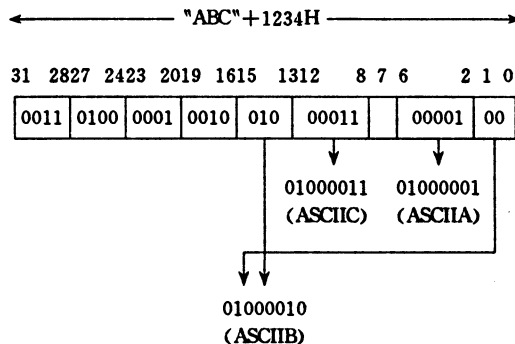


图 1 EISA 标识码示例

MCA 型扩展卡的工作机制类似。每个 MCA 卡有 8 个 POS (Programmable Option Select) 寄存器,软件可通过 I/O 口 100—107 访问。标识某个 MCA 卡时,软件可通过端口 096H 的适配器激活寄存器选择该卡(见图 2)。然后从口地址 100—101 获取一个 16 位卡标识码。同 EISA 卡一样,用户通过配置实用工具和配置文件将配置信息写入 COMS 中,系统 BIOS 在引导时初始化各 MCA 扩展卡。

PCI 总线标准提供了扩展卡的自标识方法。每个 PCI 卡有 256 字节的配置空间,用于标识该卡的制造商和类型,同时也指明该卡的资源需求。软件通过口地址 CF8 选择某个寄存器来读取其中的 32 位值。下面是 PCI 卡中配置信息的数据结构。

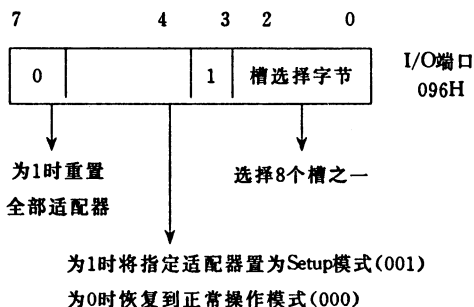


图2 MCA 激活寄存器

typedef struct tagCONFIG

```
{
WORD vendorid;      //00 制造商 ID
WORD deviceid;      //02 设备 ID
WORD command;       //04 命令寄存器
WORD status;        //06 状态寄存器
BYTE revision;       //08 修订 ID
BYTE class[3];       //09 类码
BYTE cachesize;      //0C cache 尺寸
BYTE latency;        //0D 内含计时器
BYTE hdrtype;        //0E 头信息类型
BYTE bist;           //0F 预置自检控制
DWORD baseaddr[6]    //10 基地址寄存器
DWORD cis;           //28 CardBus CIS 指针
WORD subvendor;      //2C 子系统制造商 ID
WORD subsystem;      //2E 子系统
DWORD romaddr;       //30 扩展 ROM 基地址
DWORD reserved[2];   //34 预留
BYTE irq;            //3C 中断线
```

```
BYTE ipin;           //3D 中断引脚
BYTE min_gnt;        //3E 33mhz 总线最小突发
BYTE max_lat;        //3F 最大突发
BYTE device[196];    //40 设备相关信息
}CONFIG, * PCONFIG;
```

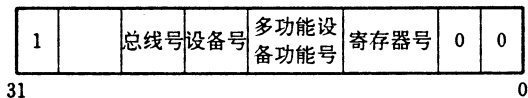


图3 写入口 CF8 的 PCI 配置地址

由于 EISA 和 MCA 总线标准只标识扩展卡的制造商和类型,用户依赖于配置软件及配置信息文件,对用户而言,理想的情况是只需将硬件插入 PC,操作系统即自动识别新硬件及其资源需求。而这正是 ISA Plug and Play 的目标。

2 ISA Plug and Play 插卡

ISA PnP (Plug and Play) 卡中包含了对 I/O 口、DMA 通道、IRQ 及映射内存位置需求的完整描述。配置 PnP 卡时,软件首先给卡指派一个唯一标识该卡的 CSN (Card Select Number)。机器启动时所有 PnP 卡的 CSN 为 0。软件往 I/O 口 279 写入一个 32 字节初始化键,将所有 PnP 卡设成被称为睡眠状态的初始化配置状态,然后各卡被指定一个 CSN。通过各不相同的 CSN,系统可自动读取各卡的资源需求。

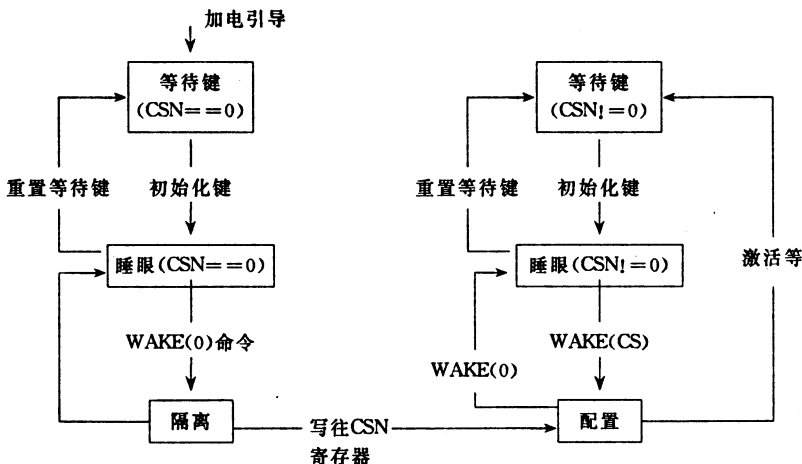


图4 ISA PnP 卡的状态简图

每个 PnP 卡的标识码中有 32 位标识制造商,32 位标识卡本身,另有 8 位是校验码,这 72 位只是隔离阶段期间读操作所涉及数据的一半,在第 n 和 $n+1$ 读操作之间,处于隔离阶段的卡检测位其 72 位标识码的位 $n/2$,如果该位为 1,PnP 卡在读取偶字节时往数据总线写入 55H,在读取奇字节时往数据总线写入 AAH。如果该位为 0,PnP 卡检查数据总线的最低 2 位,看是否有任何其它卡对 55H 及 AAH 作出反应。如果没有发现,则 0 位卡保持隔离状态,否则 0 位卡切换至睡眠阶段。最终,只有一个卡处于隔离状态,软件此时可分派一个非零 CSN。该卡不再参与往后的检查及分派过程,这样所有 PnP 卡都将先后获得一个非零 CSN。

如果 PC 中装有新版 Plug and play BIOS,即使在 MS-DOS 下也可实现自动配置。Windows95 则通过自动软件配置,进一步拓展了 Plug and play 概念。

3 即插即用

Windows95 的配置管理器 (CONFIGMG.VXD) 是一个 Ring0 级 VxD,负责标识及配置所有硬件并维护驱动程序,CONFIGMG.VXD 几乎直接与 Device Manager 和 Device Installer Control Pane 交互 Windows-95 每次引导时都要启动 CONFIGMG.VXD,后者能对加电和硬件拆卸等动态事件作出反应。

Plug and Play BIOS 根据它探知的硬件特性得出一个“驳接 ID”,而一个称为 CONFIG\$ 的 MS-DOS7.0 实模式驱动程序扫描 Windows95 信息库 (Registry database),创建一个匹配该 ID 的硬件特性文件的名称。用户通过 Windows95 Device Manager 创建及修改硬件特性文件。

开发 Plug and Play VxD 需使用 Windows95 Device Driver Kit,无论是编写设备加

载器还是设备驱动程序,实际上都是编写 Ring0 VxD。这种 VxD 通常可动态加载,它们必须处理 PNP_New_Devnode 系统控制消息。

通过 Microsoft DDK 用 C 语言开发 VxD 时,仍然需要用汇编语言写一小段程序。假设我们开发一个 School Bus Driver,其汇编语言文件包含 Declare_Virtual_Device 宏以及 VxD 开发中必不可少的设备控制例程,设备控制例程中的语句只是将系统控制信息传递给 C 函数。例如,下列语句处理 PNP_New_Devnode 消息:

```
Control_Dispatch PNP_New_Devnode,-
OnPnpNewDevnode,\cCall,<ebx,edx>

#define WANTVXDWRAPS
#include <basedef.h>
#include <vmem.h>
#include <vxdwraps.h>
#include <configmg.h>
CONFIGRET OnPnpNewDevnode(DEVNODE
devnode,DWORD loadtype)
{
    switch(loadtype)
    {
        :
    }
    return CR_DEFAULT;
}
```

大部分与 Configuration Manager 交互的驱动程序都是可动态加载的。在 DEF 文件的 VXD 语句部分加上 DYNAMIC 关键字即可使驱动程序动态可加载。

VXD SCHOOL DYNAMIC

同其它驱动程序一样, Sys_Dynamic_Device_Init 和 Sys_Dynamic_Device_Exit 系统控制消息必须处理,并且返回成功值。在 C 语言驱动程序中,这意味着采用标准的 Control_Dispatch 宏激活一个返回 TRUE 值的 C 函数。C 驱动程序通过分析 VXDWRAPS.H 头文件确定哪些服务可用,VtoolsD 扩展了

可调用的服务的数目。一个驱动程序可以象下面的例子那样登记自己：

```
CONFIGRET code = CM_Register_Enumerator (devnode, onEnumerate, CM_REGISTER_ENUMERATOR_HARDWARE);
```

DDK 文档中的 VxD 服务如 CONFIGMG_Register_Enumerator 等是由一个称为 CONFIGMG 的 VxD 输出,但这些服务需避免直接调用,应该使用在 CONFIGMG.H 中

```
CONFIGRET OnPnpNewDevnode(DEVNODE devnode,DWORD loadtype)
```

```
{
    //onPnpNewDevnode
    switch(loadtype)
    {
        case DLVXD_LOAD_DEVLOADER:
            {
                //DLVXD_LOAD_DEVLOADER
                CONFIGRET code;
                char dvrname[REGSTR_MAX_VALUE_LENGTH];
                DWORD lname=sizeof(dvrname);
                code=CM_Read_Registry_Value(devnode,NULL,"PortDriver",
                                           REG_SZ,dvrname,&lname,CM_REGISTRY_SOFTWARE);
                if (code!=CR_SUCCESS)
                    return code;
                code=CM_Load_DLVxDs(devnode,dvrname,DLVXD_LOAD_DRIVER,0);
                return code;
            }
            //DLVXD_LOAD_DEVLOADER
        }
        return CR_DEFAULT;
    }
    //OnPnpNewDevnode
```

对 CM_Read_Registry_Value 的调用从信息库的软件分枝中读取对应于 DEVNODE 的 PortDriver 值。程序员不必知道每个关键字是如何命名的,只需提供 DEVNODE 指针和 CM_REGISTRY_SOFTWARE 标志,Configuration Manager 将接手随后的事情。关键字的长度并无限制,本例中的 256 字节缓存足以容纳 Windows95 下的长文件名。如果 CM_Read_Registry_Value 参数为 NULL,则函数将提供实际值占用空间的尺寸,这时可调用 HeapAllocate 动态分配内

用宏生成的带 CM_前缀的名称。

4 设备加载程序

设备加载程序(device loader)都有一个 PNP_New_Devnode 处理例程,用来决定加载哪一个驱动程序并且随后动态加载之。寻找正确的驱动程序的通常办法是在信息库中读取 CM_Read_Registry_Value 软件关键字,其中 PortDriver 的值即指明了正确的驱动程序,下面是读键的代码示例:

存,需要注意的是最后要用 HeapFree 释放动态空间。

对 CM_Load_DLVxDs 的调用动态加载参数 2 指定的驱动程序。这个参数将包含驱动程序文件名,可以是多个文件名,分隔符是逗号或空格。该 API 利用 VXDldr 虚拟设备动态分配指定的驱动程序,如果未指定路径,VXDldr 象 Windows OpenFile 函数那样在 Windows 和系统目录中搜索文件,然后在当前路径中搜索。注意某些系统构件期望在特定目录中寻找文件,例如,I/O 子系统采用系统目录的 IOSUBSYS 子目录作为许多

驱动程序的存放地点。当然大部分驱动程序都在系统目录中以便于搜索。

有许多特殊情况需应付,最常见的 COM1 串口即是一个例子,实际上 VCOMM 为每个串口机并口将自己登记为 Device Driver。当 Configuration Manager 调用 VCOMM 承当设备加载程序的角色时, VCOMM 将完成登记过程。VCOMM 并不实际动态加载 SERIAL. VXD,直到某个应用程序打算使用 COM1. SERIAL. VXD 实际是一个 VCOMM 的“port-driver”,而不是常规的 Plug and Play 设备驱动程序。

另一个例子是静态 VxD 也是一个设备加载程序。VMOUSE 等静态 VxD 对 Device-Init 系统控制消息的反应是完成大部分初始化工作,但大多数 VxD 的初始化发生在运行 Configuration Manager 之后。由于总线清点 (bus enumeration) 发生在 Configuration Manager 的 Device-Init 过程中,静态 VxD 可能在准备好处理之前收到 PNP-New-Devnode (DLVXD-LOAD-DEVLOADER) 消息。这种情况下,静态 VxD 对早到的 PNP-New-Devnode 消息返回 CR-DEVLOADER-NOT-READY 作为回答。一切就绪之后,静态 VxD 调用 CM-Register-DevLoader,这时,Configuration Manager 将发出另一个 PNP-New-Devnode 消息以恢复设备加载过程。

5 总线清点程序

根清点程序要掌握的设备之一是 PC 中的硬件总线。PCI 总线的硬件关键字是 Root*PNP0A03\0000。总线清点程序的核心是其清点函数,Configuration Manager 调用这个函数应付各种情况。

```
CONFIGRET_cdecl OnEnumerate  
(CONFIGFUNC cf,  
SUBCONFIGFUNC scf,
```

```
DEVNODE tonode,  
DEVNODE aboutnode,  
ULONG flags)
```

这里 cf 参数是关键所在,scf 参数代表子功能索引,tonode 参数指出总线本身的设备节点,而 aboutnode 参数或者指出总线的 DEVNODE,或者指出挂接设备的 DEVNODE,flags 参数包含标志位,目前为 0。cf 参数的含义如下:

CONFIG-APM	电源管理事件发生
CONFIG-CALLBACK	调用 CM-CallBack-Enumerator
CONFIG-ENUMERATE	清点中层子节点
CONFIG-FILTER	限制逻辑配置
CONFIG-PRESHUTDOWN	系统将关机
CONFIG-READY	设备已安装
CONFIG-REMOVE	设备正从系统中拆卸
CONFIG-SETUP	设备已安装
CONFIG-SHUTDOWN	系统正在关机
CONFIG-START	开始采用分派的配置
CONFIG-STOP	停止使用当前配置
CONFIG-TEST	查询
CONFIG-TEST-FAILED	前一个 CONFIG-TEST 失败
CONFIG-TEST-SUCCEDED	前一个 CONFIG-TEST 成功
CONFIG-VERIFY-DEIFY-DEVICE	确认设备是否存在

Root Enumerator 为包括总线在内的所有继承设备创建 DEVNODE。每个总线清点程序然后为关联的硬件创建 DEVNODE。一些关联设备(例如 SCSI 控制器)可以再关联自己的设备,因此它们的设备驱动程序也包含清点程序以便创建附加的 DEVNODE。Configuration Manager 然后通过调用这些清点程序来构造硬件树。实际的清点程序有一些方法去发现物理设备。例如 ISAPNP 采用隔离方案去寻找关联于 ISA 总线的 Plug and play 设备。清点程序将确定唯一的设备标识码并创建设备节点,下面是一个示例:

建设现代列车通信网

上海铁道大学机电工程学院 刘 竞

摘 要 现代列车通信网有着重要的地位和广阔的发展前景。本文就列车数据传输系统的特点,阐述了列车通信网的结构和通信协议,并对当前我国研制列车通信网的问题和难点,探讨了其中的关键技术。

关键词 列车通信网 局域网 光纤分布式数字接口 异步传输模式

作为大运量、高密度的客运有效手段,开发现代高速列车是世界铁路的发展趋势。为实现对列车的运行,机车和车辆的控制、状态监测、故障诊断以及对旅客信息服务系统进行综合处理,就要采用经济可靠的计算机控制技术。而多机控制的现代列车上,由于配置多个实现不同功能的装置,势必要求建立一个连接网络或通信网络,通过数据交换进行控制和监视,并把多种信息以透明的形式提供给司机和维修人员。因此,列车通信网是现代列车控制、检测与诊断系统的核心部分,它保证了列车的安全、可靠、舒适运行。同时,它也是整个铁路自动化工程的重要子系统,和

车站现场网、调度监督系统(CTC)、铁路管理信息系统(TMIS)等构成的广域网,便于从全局方面来调配和控制工程的发展。

1 列车通信网的任务

列车通信网的功能以及任务主要包括四个部分:

1.1 机车传动控制

由司机台手柄位或列车自动控制装置,可以得到机车牵引力或制动力的给定值。根据这一控制主指令,机车控制可分为三级:列车控制级、机车控制级和传动设置控制级。

```
DEVNODE device;
CONFIGRET code = CM_Create_DevNode
(&device, "SCHOOL\\WCO1234\\0000", tonode,
0);
if (code == CR_ALREADY_SUCH_DEVN-
ODE)
return CR_SUCCESS;
:
return CR_SUCCESS;
```

在这个例子中,CM_Create_DevNode 为设备创建了一个新的设备节点作为总线的子节点。CM_Create_DevNode 的第二个参数是一个设备标识码,它分为三个部分。第一部分与总线名相同,第二部分通过制造商和类型标识设备本身,第三部分标识该设备在计算

机中的一个实例。Configuration Manager 并不细究标识码的内部,它只要求设备标识码在系统中是唯一的,同时它为对应的 DEVNODE 命名 HKLM\\Enum 的硬件子关键字。

6 结语

Plug and Play 是一项使得硬件可智能配置的新技术,目前已在 Windows95 中实现。关于 PnP 的编程主要是如何编写 VxD 虚拟驱动程序的问题,这其中包含许多复杂的实现细节。掌握 PnP 开发需仔细研究 PnP 技术规范并了解硬件运作机制,本文在此权作抛砖引玉。

a. 列车控制级包括列车控制装置和总线管理器,主要处理来自司机台或自动控制装置的信息,产生与整个列车有关的控制变量并实现对总线的控制。

b. 机车控制级包括防空转和防滑装置、电空联合制动和保护逻辑,主要根据列车控制级输出的指令和给定值实现对本机车的控制功能如:优化黏着利用,给不同制动系统分配制动力,控制列车的加速度变化率,监控主要设备的电气和热状态及气动设备的压力等。

c. 传动控制级主要任务是对主变流器实现控制从而控制牵引电动机的转速和转矩,同时还反馈实际状态信息。

1.2 中央监视

用于存储大量的故障信息及相关环境数据,即过程数据,故障发生前的状态和伴随出现的有关状态。这些故障信息由列车各计算机子系统或传感器送来,经过中央监视装置处理后分为三个显示层次;第一层是对司机的常态数据和故障的发生信息和应急措施;第二层是对机车上的维修人员进行小修理提供有关故障情况记录;第三层是对地面检修部门进行大维修提供故障原因深入查找和诊断结果统计分析。

1.3 车厢控制与监视

用于旅客车厢的计算机控制与监视,主要对本车厢的服务功能(如门控、空调、旅客信息系统)进行控制,并对这些功能的主要特征量进行检测和监视,诊断结果传送到中央监视装置中。

1.4 旅客信息服务

向旅客提供运行信息、娱乐、通信和其他服务。各车厢的计算机显示器、音响、电视、服务员呼唤装置、无线电通信装置、故障处理装置都与车厢控制、监视中心相接,再由后者接向旅客信息系统操作中心。

2 列车通信网的特点

与地面计算机通信网比较,现代列车通信网的传输距离短,不超过 2000 米,所以它很象一个局域网。由于列车的干扰状况和运行环境的恶劣程度是其他工业场合不可比拟的,所以列车对通信网的可靠性要求极高。同时,又要求通信网具有实时性,卓越的事件驱动能力和时间驱动能力,即在规定的采样周期内,及时响应操作命令,及时采集机车的工况参数,及时给出控制指令。

国际电工委员会 IEC 和国际铁路联盟 UIC 联合成立的专门的工作组已在研究制定适用于列车控制系统的通信网络规范—TCN(Train Communication Network)标准,它把列车控制系统中传输的信息分为以下六种:

- 列车运行的控制命令。
- 运行中的各车厢(包括有动力装置的机车和无动力装置的旅客车辆)的状态信息。
- 故障诊断所需的信息及结果。
- 显示单元所要显示和提示的各种信息。
- 通信网的管理信息。
- 其他与列车运行,安全及服务有关的需要在列车内互相传递的信息。

根据信息的性质和实时性的要求,我们认为列车网络通道中传送的主要有三种类型数据:

a. 过程数据

过程数据表示列车上的状态,如时间、速度、列车当前运行状态、司机命令等。过程数据的特点是长度短而时间性强,如紧急的过程数据必须在 50ms 以内通过列车总线。因此传输的时间必须是确定和有界的,为保证时延,这些数据周期性地发送。

b. 消息数据

消息数据是一些非紧迫的信息,例如诊

断功能,旅客服务功能的消息。消息数据的长度可能较长,因此可以由若干帧组成。消息数据以命令方式传送,因此是非周期的,传输时间不能有确定的上界,允许随负荷情况而变化。

c. 管理数据

管理数据是同一总线内主设备对从设备的状态校验、检测以及列车初运行和其他管理所用的数据,特点是帧很短。

列车运行期间,网络通道主要是用于过程数据的传输。而对于这些短而频繁的紧迫数据的传输在 ISO 等标准中尚未考虑,车厢中许多小设备也不适合用复杂的软件,因此根据用户之间的公约可对通信网的通信作很大简化。例如,所有设备都使用由公约所规定的相同格式时,无需由表示层来协调数据格式。

总的说来,列车通信网类似于局域网,但有其特殊性,即:网络拓扑结构简单,传输的报文短,报文实时性要求高,抗恶劣环境能力强等。

3 列车通信网的结构

本着开放和标准化的原则,列车通信网符合 ISO 的 OSI 模型。按照网络拓扑设计中的分群设计原理,对于节点数目不多的列车通信网,建立两级结构较为适合:车厢局域网连接机车或车厢内部各种终端装置,较高级的列车主干网连接机车和车厢各网络节点。TCN 标准中也将列车主干网和车厢局域网称为列车总线和车厢总线。有时也在车厢总线下设第三级总线,如连接传感器的总线或连接执行单元的控制总线,可把这些总线认为是车厢总线的一部分。列车通信网两级结构如图 1 所示:

列车主干网的结构主要有总线型或环型。我国试研制的高速列车采用总线型拓扑结构,由于 20 个车厢及头、尾两台机车的编

组方式,传输长度已达 860 米极限,因此需减少各车接插件中引起的耗损及长线传输带来的干扰。列车主干网采用半双工总线呼叫方式,操作端的主机采用分时方式实现一点对多点通信。采用光纤传输可提高传输频率,扩大传输信息量,且抗电磁干扰能力强,此种网络结构一般采用双工环型。由于目前我国机车、车辆之间的光纤接口未达到研制水平,列车主干网仍用双绞屏蔽线。

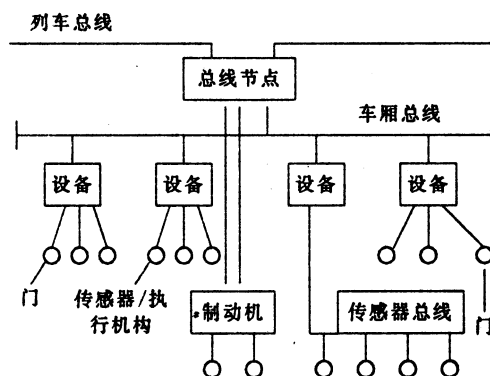


图 1 列车通信网结构

车厢局域网采用星型结构用于分散控制,根据数据交换方式分为全总线结构和部分总线结构。全总线即所有有计算机控制的部件全挂在总线上,通过串行通信进行数据交换,部分总线即关键部件挂在总线上,其余则通过输入输出交换信息。机车的控制比一般车辆较为复杂重要,所需交换信息较多,其车厢总线尽可能采用全总线,车厢内部光缆接口市场也有货源,可考虑用光缆。

4 网络技术发展

近年来网络结构演进极其迅速。一种方案尚未普及,新的应用观念和技术已推陈出新。我们列车通信网的研制原则是采用先进、成熟、有发展前途的技术,并基于标准能对铁路其它信息网创造高速联网的基础。目前最有前途的网络技术就是:光纤分布式数据接口(FDDI),异步传输模式(ATM)。

4.1 FDDI

FDDI 是目前技术上最成熟的高速网络,作为列车主干网可提供分布式应用的必备带宽和恶劣环境下的可靠数据传输,已在德国 ICE 高速列车和日本新干线上得到成功应用。

FDDI 的主要特点如下:

- 多数据帧处理方式提供了高环路利用率;
- 100Mbps 的高传输速率;
- 多模光纤 2Km 单模光纤 25~100Km 的站间距离提供大的网络覆盖范围;
- 限定令牌可满足特殊传输要求;
- 完善的站级分布式管理体制 SMT;
- 双环拓扑冗余的高可靠性和良好的互操作性。

FDDI 的标准共分四个子层,即 MAC, PHY, PMD 和 SMT,它们和 ISO 的 OSI 网络协议的对应关系如表 1:

表 1 FDDI 和 OSI 的协议层次

OSI	FDDI	
数据链路层 (DLL)	LLC	
	介质访问控制层(MAC)	SMT 站 管理
物理层 (PL)	物理协议层(PHY)	
	物理介质相关子层(PMD)	

FDDI 采用令牌传递模式(token-passing scheme)来决定工作站的存取访问。进入环的第一个工作站可建立并测试与环的连接,然后产生一个在工作站间传递的令牌。当一个工作站接收到令牌时,它才能传送一组固定的数据帧。

FDDI 有几个通过带宽分配来实现的优先机制。第一个叫同步带宽分配(Synchronous Bandwidth Allocation, SBA)机制,它可以将一定量的带宽分配给一确定的工作站或一些工作站,让它们有更大的捕捉令牌机会。第二个机制叫异步服务,它占用未通过 SBA 分配的带宽并将这部分带宽等分给环

上的工作站。异步服务中,每个工作站有一个令牌旋转时间标记(TokenRotation, Timer, TRT),可告诉工作站何时可得到令牌,当令牌出现时,工作站将 TRT 或期望的到达时间比作目标令牌旋转时间(Target Token Totation Time, TTRT),TTRT 是令牌旋转一周的预先设定时间,通常为 8ms。如果 TRT 低于 TTRT,工作站就能捕获令牌并传出异步数据帧。如果 TRT 大于 TTRT,则令牌来得太迟,只能进行异步服务工作站必须跟随 SBA 功能的工作站。

FDDI 的反向旋转的双环拓扑结构,产生介质及相应的控制设备保证了它的高可靠性。FDDI 的双环包括主环和副环。主环用来进行正常工作,当出现故障时,副环可用来构成通路。如果是电缆故障,两边的站将信息流从主环绕到副环,以避免主环的故障。

4.2 ATM

ATM 是一种以信元(Cell)为基础的高速分包多工交换(Multiplexing and Swiching)技术,提供了与现有各种网络的连接能力。规模从几十 Kbps 到几个 Gbps 的带宽范围任选,这些不同速度的选择允许 ATM 连接台式机、服务器、主干网和广域网。ATM-LAN,是一种新的复用与交换机制,将列车局域网无缝集成进入广域网,因此是列车通信网和外部广域网理想的网络互连模式。

ATM 协议层次结构如表 2:

表 2 ATM 协议层次

管 理	控制高层	用户高层
	ATM 适配器(AAL)	
	异步传输层(ATM 层)	
	传输聚合层(TC 层)	
	物理介质层(PLCG)	

ATM 用于列车通信网中,能提供数据、语音和视像的服务,同时不受所支持设备数目的限制,具有可延展性和灵活性。延展性是

防火墙技术在网络计费系统中的应用

辽宁大学计算中心 尹春华 陈 雷

摘 要 本文首先论述了目前网络安全管理中常用的防火墙技术的种类及其安全策略,然后讨论了防火墙技术在网络管理计费系统中的应用。

关键词 防火墙 计费系统 网络

1 引 言

在计算机网络,尤其是在 Internet 网迅猛发展的今天,网络系统的安全问题,越来越得到了人们的关注。目前在开放的环境下,系统安全的含义和维护与过去相比有了根本的

指当需要新的应用和新设备时,网络可支持更高速率的能力;灵活性是指网络中各种设备速率可混合使用,以支持特定网络部件的需求。

按需分配带宽是其区别与现有网络的一大优点。这是通过呼叫允许控制 CAC(Call Admission Control)来实现的。其含义是在连接建立之前,申请者经过一谈判过程获得所需的频宽。当然,这一频宽是申请者所能忍受的,即获得一定的业务质量保证。有了这一保证,申请者的信息便不会受网上繁忙程度的影响。在 ATM 中,大小不同的数据包均分成 48 字节一段,然后加上 5 字节的头信息,头信息包含数据的类型及该包来源和目的地。使用头信息可以给对时间敏感的数据赋以适当的特性,以便进行多种数据的无缝传输。

ATM 网络中有四种业务类型:1)恒定比特率 CBR(Constant bit rate);2)可变比特率 VBR(Variable bit rate);3)可用比特率 ABR(Available bit rate);4)无定义比特率

变化。联网为人们带来了许多便利,但是在享受这种便利的同时也必须承担相应的风险。因为一些不合法的使用者可能会随时闯入,使用户遭受损失。因此在这种形势下,如何采用有效的安全防范技术,保证网络安全可靠的运行,已成为一个现实而迫切的问题,防火墙的概念也由此应运而生。

UBR。其中,前两类业务已有成熟应用。总的来说,ATM 是新技术,尚未形成统一的产品标准,且价格昂贵。当前使用 ATM 需采用局域网仿真(LAN Emulation),使用 ATM 的优越性难以充分体现。但它代表了铁路大信息网的发展方向。

5 结束语

当前我国准高速列车的研制已经进入实施阶段,列车通信网还是我国铁路事业的新领域。应该看到,国外各大工业集团已有了各自成功的应用系统,而列车通信网的统一国际标准尚未完善,解决我国列车通信问题不能一蹴而就,应及时对计算机网络通信的各种技术作深入了解和研究,充分考虑每一项技术的特征。在引进和研制时尽量采用先进和成熟的技术,走标准化和开放系统的道路,发展我国现代列车网络体系,为实现铁路自动化打好坚实的基础。

2 防火墙的分类及使用的安全策略

所谓防火墙,从广义上说,它是一个系统或系统组,它在两个网络之间实施相应的访问控制策略。特别意义上说,防火墙是部件和系统的汇集器,它置于两个网络之间,并且具有如下特征:a. 所有从内部通向外部或从外部通向内部的通信业务都必须经过它;b. 只有经过授权的通信业务才允许通过它;c. 系统自身对入侵是免疫的。从另外意义上说,防火墙是被用于保护一个可信网络的装置,以防止来自一个不可信网络的攻击,如图 1 所示。

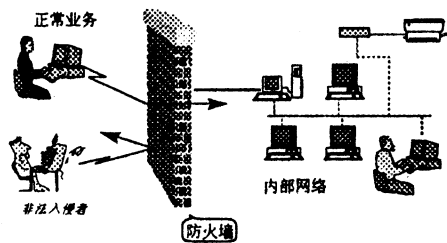


图 1 防火墙示意图

目前根据防火墙在网络上的物理位置,在 OSI 七层协议中的逻辑位置以及所具备的功能,可分为以下两大类:

a. 基本型防火墙。主要包括

- 线路级网关(Circuit-level Gateway): 在此类型中,防火墙放置在 Internet 网和私有网之间,逻辑地工作在 OSI 协议的下三层,他根据客户的要求,开通或封闭基于 TCP/IP 的连接,它适合于网络内部对外部的访问限制,但不能实施强验证和协议过滤。

- 包过滤路由器(Packet Filters Router): 这是一个检查通过它的数据包的路由器,主要运用 IP 地址和端口号进行限定处理。它是在七层协议的下三层实现的,包的类型可以进行拦截和登录,因此它比其它类型的防火

墙更易于实现。

- 应用网关(Application Gateway): 它的物理位置与包过滤路由器一样,但它的逻辑位置是在 OSI 七层协议的应用层上,所以它采用应用协议代理服务(Proxy Services)的工作方式实施安全策略。

b. 复合型防火墙。主要包括

- 双重基地型网关(Dual-Homed Gateway): 它使用一个含有两个网络接口的应用网关,并将其接在内部网络和包过滤路由器之间,信息服务器接在两者之间。

- 屏蔽主机防火墙(Screened Host Firewall): 它的应用网关只需要单个网络端口,物理地连接在包过滤路由器的网络总线上。但其逻辑功能仍工作在应用层,所有业务都要通过它来代理服务。

- 屏蔽子网防火墙(Screened Subnet Firewall): 它使用两个包过滤路由器,形成一个子网的态势。

- 集中控制调制解调器(Modem Pools): 它要求网内各用户不能直接通过调制解调器与网外连接。只允许通过安装在防火墙适当位置的集中控制的调制解调器与网外联系。

防火墙的安全保护能力在逐步提高,它一般采用了以下安全策略。

- 网络服务访问控制策略: 包括登录控制、先进的鉴别机制,如时间同步鉴别、挑战/响应鉴别。

- 信息完整性校验策略和信息加密策略。

- 跟踪审计策略

- 可信计算机策略等。

3 防火墙技术在计费系统中的应用

做为网络管理的五大功能之一的计费管理是大家比较关注的问题。目前国内针对不同类型的网络服务,计费的策略也各不相同:

一般来说,话音服务以时间为依据;数据服务则采用以时间、通讯量为标准的复式计费;而在信息服务中,通常是统计网络流量,如以 IP 字节作为基本计费单位。

一个完整的网络计费系统,主要由两部分组成:一个是数据采集子系统,另一个是数据处理子系统。其中数据处理子系统主要是完成数据的统计、查询和对报表的处理,可用一个数据库系统来实现完成。

计费系统实现的关键是数据采集子系统,而数据采集子系统的实现目前主要分成基于主机和基于路由器两种。基于主机的数据采集可直接利用系统本身提供的日志功能,来获得详细的计费数据。但由于基于主机的计费功能只能记录本机上的用户,这就要求用户只能使用在计费主机上为其设立的用户帐号,这将无论对于帐号管理还是主机的系统开销都是一个沉重的负担。基于路由器的数据采集可以得到通过该路由器的所有 IP 地址的数据流量,我们可以根据网络中每个 IP 地址的进出流量,以及在不同时间段,对不同费用的网络的使用情况等,进行统计,给出记帐依据。在基于 TCP/IP 的互联网络中,所有应用的数据都是在 IP 数据包上进行的。因此 IP 的数据流量反映了用户对网络的实际使用情况。但是此种方法中,基于路由器的数据采集无法将 IP 地址和使用该 IP 地址的每个用户联系起来,因此它无法得到每个用户具体的使用情况。而且还容易出现非法用户冒用他人的 IP 地址或未分配的 IP 地址。另外基于路由器的数据采集只能收集到每个 IP 地址对外访问的数据流量,无法得到其它信息。

通过以上分析,我们看到这两种数据采集方式均有弱点。而且目前实现防火墙的两种手段之一的基于代理技术,即在应用层网关上实现防火墙功能。这种基于代理方式技术的防火墙不仅可以加强系统的网络安全,还可用它完成计费服务,而这种计费手段恰

巧能克服以上两种数据采集的缺陷,是一种折中解决方案,首先,它是基于主机实现的,可完成基于主机的数据采集的全部功能,通过对于用户使用情况的日志,提供详细的计费数据。通过设置用户 ID 和口令,可以区别每个使用的用户。其次,它运用的是代理技术,与在主机上为每一个用户分别建立帐号相比,大大减轻了系统开销,可在同一台主机上支持更多的用户。但此种技术也有本身固有的缺陷,即以减少易用性作为代价。

使用代理型防火墙进行计费的另一个好处是简化了对 PPP/SLIP 用户的计费管理。对于 PPP/SLIP 用户,由于其在拨通之后,可以不登录到网控中心的任何机器上,所以无法利用主机系统为其计费。尽管可以通过在路由器上为每个用户设置帐号来缓解矛盾,但路由器的密码数量也有一定限度。当使用防火墙进行计费时,可以在路由器上使用一个公共帐号,然后要求所有用户必须从防火墙上通过,从而完成计费功能。

近年来 Internet 在全球迅猛发展,我国以中国教育科研网(CERNET)为代表的大批专业性网络正在建设之中。当前网络发展的特点是,网络规模越来越大,复杂性和异构性越来越高,这使得网络的安全性问题变得日益重要。在这种形式下,如何采用有效的网络管理技术,保障网络安全可靠地运行,已成为一个现实而迫切的问题。

参考文献

- 1 Marcus J. Ranam. "Thinking about Firewall", Proceeding of second International Conference on Systems and Network Security and Management (SANS - I), April 1993
- 2 赵战生. "Internet 防火墙". 中国计算机用户. 1996
- 3 董学章, 赵宏. "防火墙技术及其在校园网建设中的应用". 中国教育和科研计算机网的研究与发展. 第一卷. 华中理工大学出版社, 1996

用 MFC 类库实现 Winsock 编程

上海交通大学计算机系 施 雷 王 纤

摘 要 本文介绍 sockets 的网络编程原理,同时结合 MSVC4.2 基本类库中基于 Windows Sockets 的 CSocket 类的编程经验,给出一个例程说明。

关键词 流方式 数据报文方式 客户机 服务器 Socket

随着计算机网络的飞速发展,许多软件商开始面向较完善的 TCP/IP 网络通讯协议开发一些网络通信接口。目前流行的 Internet 网是在 UNIX 系统上发展起来的,sockets 是其中较为成熟的网络编程接口。socket 在 UNIX 上的成功应用使 Windows 下的 sockets 程序设计标准很快出台。Windows Sockets API 是 Microsoft Windows 的网络程序接口,它包括一个标准的 Berkeley Sockets 功能调用的集合以及为 Windows 所作的重要扩充,这些扩充有利于程序开发者更好地利用 Windows 的消息驱动特性,设计出高性能的网络程序。MFC 自 4.0 版本开发了两个基本类 CAsyncSocket 和 CSocket,为网络通信编程提供了两种模式。它们支持数据报文方式传输和流方式数据传输,并同 MFC 中档案对象 CArchive 相联系,是网络编程的高层抽象。CSocket 类的封装、开放和标准化使网络通讯变得透明了。它提供了一些网络通讯的标准成员函数及与文档类联系的模式,使得编程人员不必了解底层的网络通信,即可方便地在网上不同应用程序间交换数据,体现了网络编程面向对象的特点。

1 Sockets 的工作模式

Socket 原意为插座,代表通信的一端,通常作为一个网络节点与网络通信范围内的其它 Sockets 交换数据。Socket 有三种通信方

式。第一种是流方式(Stream Socket)它提供了一个面向连接的服务,数据可靠的、无错无重复地发送,且数据按发送顺序接收,内设流量控制,避免数据流超限;数据被看作是字节流,无长度限制。在这种方式下,两个通讯的应用程序之间要先建立一种虚拟的连接。它对应 TCP 协议。流方式的特点是通讯可靠,对数据有校验和重发的机制,通常用作数据文件的传输,如文件传输协议(FTP)和电子邮件(TELNET)。第二种叫做数据报文方式(Datagram Socket),提供一个无连接服务,它对应 UDP 协议。数据报以独立包形式被发送,不提供无错保证,数据可能丢失或重复,并且接收顺序混乱。此方式由于取消了重发报文机制,能够达到较高的通讯速率,可以用作一些对数据可靠性要求不高的通讯,如实时的语音、图像传送,广播消息等。第三种为原始方式(Raw Socket),该接口允许对较低层协议,如 IP、ICMP 直接访问。常用于检验新的协议实现或访问现有服务中配置的新设备。

通常 Sockets 工作于全双工方式,即数据流可以同时两个方向上传输。Sockets 至少可用于三种以上的通信环境:客户/服务器体系结构、点对点方式及远过程调用(RPC)。在 TCP/IP 网络协议中,通信的两进程间相互作用的主要模式是客户/服务器模式(Client/Server Model)。即客户向服务器发出服务请求,服务器接收到服务请求后,提供相

应的服务。因为是进程通信,同一个主机即可作为服务器也可同时作为客户,但服务进程一般是先于客户请求而启动。

2 CSocket 类及编程模式

在微软的 MFC4.0 (Microsoft Foundation Class Library) 中,增加了支持 Winsock 编程的类,其中主要有 CAsyncSocket 类和 CSocket 类

在 CAsyncSocket 类封装了部分 Windows Sockets API。CAsyncSocket 类主要针对那些具有一定网络编程经验,希望同时拥有 socket API 编程的灵活性和类库编程的便利性的开发者。CAsyncSocket 类的主要功能是把某些基于 socket 的 Windows 消息进行转化,利用其回调函数进行处理。

CSocket 类是 CAsyncSocket 类的继承类。这个类比 CAsyncSocket 类具有更高的抽象化。并且如果结合 CArchive 类使用 CSocket 类,进行网络编程就象读写文件一样简单。更重要的是,CSocket 提供了阻塞管理,而这对 CArchive 类的同步是特别重要的。CSocket 类支持流方式和数据报文方式的数据通信。一般情况下使用较多的是使用流方式的数据通信。

通常情况下,使用 CSocket 类对象总是涉及到创建对象本身和把它与其它 MFC 类对象相连。流方式下,CSocket 类与档案类 CArchive 相连,CArchive 对象作为数据流接送缓冲区。CArchive 对象又与一个 CSocketFile 对象相连。CSocketFile 类是 CFile 的继承类,专用于经由 Windows Sockets 网络数据接收和发送。数据在网络上的流向如图 1 示

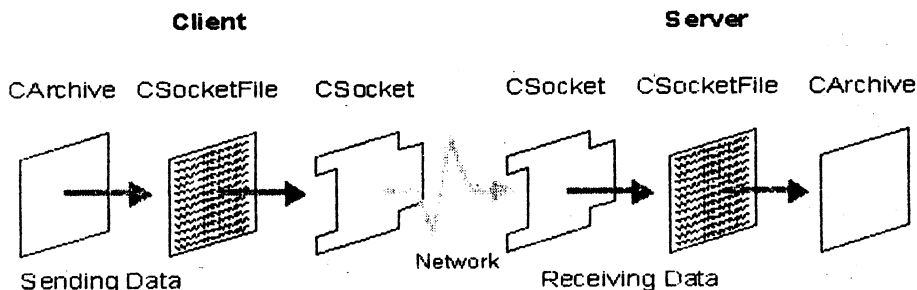


图 1 数据发送和接收流程图

在服务器和客户机建立数据通讯之前,总是服务器应用程序先开始执行以使之能倾听到客户应用程序请求连接的信号。两边应用程序分别在服务器端和客户机端作如下工作:

- 构造一个 CSocket 对象。
- 为 CSocket 对象创建一个指向底层 SOCKET 的句柄。对一个客户 CSocket 对象而言,可使用缺省值 (SOCK_STREAM) 创建;而在服务器端 CSocket 调用类成员函数 Create 创建时必须指明一个端口号。

- 在客户机端,调用 CAsyncSocket::Connect 建立此 CSocket 对象与服务器 CSocket 对象的连接。在服务器端,调用 CAsyncSocket::Listen 开始倾听客户端的连接请求;当接收到一个连接请求时,调用 CAsyncSocket::Accept 接收,注意在调用 Accept 之前必须构造一个新的空 CSocket 对象作为调用参数。

- 创建一个 CSocketFile 对象,将 CSocket 对象与之相连。

- 分别创建一个 CArchive 对象作装入

(接收)和存储(发送)用,将 CArchive 对象与 CSocketFile 对象相连。

- 用 CArchive 对象在客户机和服务器的 sockets 之间传递数据。

- 删除 archive、socketfile 和 socket 对象。

为了支持这种与 CArchive 对象相连的

Server

```
Csocket sockSrvr;  
sockSrvr.Create(nPort);  
sockSrvr.Listen();  
  
Csocket sockRecv;  
sockSrvr.Accept(&sockRecv);  
CSocketFile file(&sockRecv);  
CArchive arIn(&file,CArchive::load);  
CArchive arOut(&file,CArchive::store);  
arIn>>dwValue;  
arOut<<dwValue;
```

socket 工作方式,CArchive 中特别提供一个成员函数 IsBufferEmpty()。此函数用以判断接收信息的 CArchive 对象内部缓冲区是否为空。如果缓冲区中有多个消息数据,则需不断地读数据、清数据直至缓冲区为空。

下面用代码描述流方式下建立客户机与服务器的连接。

Client

```
Csocket sockClient;  
sockClient.Create();  
  
sockClient.Connect(strAddr,nPort);  
  
CSocketFile file(&sockClient);  
CArchive arIn(&file,CArchive::load);  
CArchive arOut(&file,CArchive::load);  
arIn>>dwValue;  
arOut<<dwValue;
```

3 使用 Csocket 类的网络通信例程

使用 CSocket 类的应用程序需在 CWinApp::InitInstance()中调用 AfxSocketInit 来初始化 Windows Sockets DLL。基本类库自动作好调用 Windows Sockets DLL 中的初始化函数 WSASStartup(),并在应用程序

//应用文档类头文件

```
#include "mysock.h"  
class CMyDoc:public CDocument  
{.....  
public:  
CString m_strHandle;           //请求建立连接方的标识句柄  
CMySocket* m_pSocket;          //请求建立连接方的 socket 对象指针  
CSocketFile* m_pFile;          //请求建立连接方的 socketfile 对象指针  
CArchive* m_pArchiveIn;        //用作客户方的接收数据缓冲区  
CArchive* m_pArchiveOut;       //用作客户方的发送数据缓冲区
```

结束之前调用 WSACleanup()结束对 Windows Sockets DLL 的使用。MFC 中,文档用来保存应用数据,视用作向用户展示这些数据,它们之间的相互作用不再赘述。仅从应用文档类的类成员数据和类成员函数就可看到应用数据是如何通信的。于是用户可以把有关数据通信的主要实现放入应用文档类。


```

public:
    BOOL ConnectSocket(LPCTSTR lpszHandle, LPCTSTR lpszAddress, UINT nPort);
        //调用 CSocket::Connect 建立连接
    void SendMsg(CString& strText);        //发送消息
    void ReceiveMsg();        //接收消息
    //{AFX_VIRTUAL(CMyDoc)
    public:
        virtual BOOL OnNewDocument();        //建立连接并初始化内部成员数据
        virtual void DeleteContents();        //断连并删除相关对象
    }AFX_VIRTUAL
    .....
};

```

数据通信的单位是消息,用户可以自定义消息的格式和内容。如:

```

class CMsg, public CObject
{
    .....
public:
    CString m_strText;
    BOOL m_bClose;
public:
    virtual void Serialize(CArchive& ar);
    //根据消息内容重新定义文件读写
    .....
};

```

利用 MFC 中 CSocket 类与 CArchive 类的联合工作方式发送数据和接收数据变得相当简单。

```

void CChatDoc::SendMsg(CString& strText)
{
    if(m_pArchiveOut != NULL)
    {
        CMsg msg;
        msg.m_strText = m_strHandle + _T(",") + strText;
        TRY
        {
            msg.Serialize(*m_pArchiveOut);
            m_pArchiveOut->Flush();        //发送缓冲区写 CSocketFile 对象, MFC 自动调用
        }
        //相关函数完成发送数据
        CATCH(CFileException, e)
        {
            //发送失败的一些处理
        }
        END_CATCH
    }
}

void CChatDoc::ReceiveMsg()
{
    CMsg msg;
    do
    {
        TRY

```

从 C 到 C++ 关于 C++ 程序设计特点的探讨

天津轻工业学院计算中心 程为曙 刘 璟 俞 瑛

摘要 通过对 C++ 特点的分析,说明 C++ 程序设计的核心问题是“类”的设计,要充分利用 C++ 这方面的优势。

关键词 程序设计 C 语言 Borland C++

C 语言是通用程序设计语言,适于编制各种系统软件和应用软件,在计算机科学的发展过程中,C 语言也不断地发展和扩充,C++ 就是这一发展的结果,和 C 相比较,C++ 在功能方面有较大的提高,程序设计风格有较大的变化。如何使用 C++ 开发软件,充分发挥它的优势,是一个值得探讨的问题。

1 C++ 的主要特点

1.1 提供了类(Class)的结构

这是 C++ 的精华。类是与程序有关的“资源”实体的集成,它提供了组织在逻辑上有关的数据、函数及其对象的技巧,为编写大

```
msg.Serialize(*m_pArchiveIn); //从缓冲区中获取数据
CATCH(CFileException,e)
{
    ..... //获取数据失败的一些必要处理,如是连接中断则
           //设制中断布尔量
}
END_CATCH
if(msg.m_bClose) //是否为连接中断?
{
    delete m_pArchiveIn; m_pArchiveIn=NULL;
    delete m_pArchiveOut; m_pArchiveOut=NULL;
    delete m_pFile; m_pFile=NULL;
    delete m_pSocket; m_pSocket=NULL;
}
if(m_pSocket==NULL)
    return;
}
while(! m_pArchiveIn->IsBufferEmpty()); //循环直到缓冲区为空
}
```

由此可见,MFC 类库中 CSocket 的使用使得网络编程变得异常简单,用户在网上发送和接收数据就好像只是在一个文件缓冲区送取数据。CSocket 类还替用户管理阻塞,使用户不再担心由于资源共享或阻塞模式引起的死锁。

参考文献

- 1 王国印译. VC++ 技术内幕. 北京清华大学出版社.
- 2 Books Online of MSVC4.2
- 3 蒋东兴、林鄂华 编著, Windows Sockets 网络程序设计指南, 清华大学出版社.

型、复杂的程序提供所需的资源。从 C 的角度来看,“类”是结构的自然演化,既包括数据说明和函数说明的结构,这些函数指明了类所能完成的各种操作。使用类时必须先定义,例如:

```
Class Counter{
    long count;
public:
    Void Setvalue(long);
    lang Getvalae( );
};
```

其中关键字 Class 是类的说明符,Counter 是类的名字,{...}以内为类体,最后以“;”结束。类体中包括函数说明或定义,变量的定义。例中的 Count 是类体中定义的变量,称为成员变量,具有类作用域。Setvalue(long)和 Getvalue()是函数说明,但不是函数定义,称为成员函数。对于上述函数的定义可以写成:

```
void Counter::setvalue(long vlaue)
{
    count=value;
}
long counter::Getvalue( )
{
    return count;
}
```

用“类名::”指明函数所属的类。

类和结构一样,是一种类型,而不是对象,使用时必须在定义类的同时定义对象。例如定义 Counter 类时可定义类变量 people,表示方式为 Counter people;

对于对象的使用只能是对象的成员,与存取结构成员类似。

1.2 提供了函数名重载和操作符重载

这也是 C++ 的强大功能之一。同一函数名通过不同的参数个数和类型,实现不同的操作。重载的好处是增加了 C++ 的灵活性,它扩展了本质上相似的操作。例如函数重载,多个函数使用同一函数名,通过采用不同的参数数量和类型,得到不同的函数返回值。函数重载通常用于解决“功能相似”的操作。

例如定义一个名称为 absobj 的类,重载成员函数 abs(x)用来计算参数 x 的绝对值:

```
class absobj{
public:
    int abs(int x)
    {
        return(x>=0)? x:-x;
    }
    fcoat abs(float x)
    {
        return(x>=0)? x:-x;
    }
    long abs(long x)
    {
        return(x>=0)? x:-x;
    }
}
```

上述三个成员函数具有相同的函数体和不同的参数类型及函数返回类型,根据函数参数类型,选择所调用的函数。

1.3 C++ 支持派生类的概念

支持 OOP 中关于“继承(inheritance)”的重要特征,形成了程序中的更高级的层次关系。类是 C++ 中提供封装的逻辑单位,封装了数据和方法。C++ 的继承机制使用户可以通过增加、修改或替换给定类中的方法来对这个类进行“剪裁”,以适应不同的需要。借助生物遗传的原理,在软件设计中也可使用继承这一方法来传播代码或减少代码的复杂性,获得新的类。继承性只适用于类,变量和函数均不能继承。继承性使用户可以连续建立或扩展自己或别人建立的类,从最简单的类派生出越来越复杂的类。解决某个课题需要建立多个类,这些类一般都是通过继承,从基本、简单的类建立起来的。派生新类时,可继承部分或全部父类特征,再加入一些新的特征。同时也允许多重继承,即一个类同时继承多个类的特征。例如:

```
Class Box{
public:
    int width,height;
    void setwidth(int w){width=w;}
    void setwigh(int h){height=h;}
};
```

```

Class coloreBox:public Box{
public:
    int color;
    void setcolor(int c){color=c;}
};
coloreBox cb;
void main( )
{
    cb.setcolor(5);
    cb.setwidth(3);
    cb.setHeight(50);
}

```

类 Box 叫做类 coloreBox 的基类, coloreBox 叫做 Box 的派生类;类 coloreBox 只说明了一个函数,但从基类中继承了两个函数和两个变量。

1.4 提供了多态性功能

可以在基类和派生类中使用同样的函数名,而定义不同的操作,实现“一个接口,多种方法”。实际上是用一个相同的名字定义不同的函数,这些函数执行不同的但又类似的操作,即用同样的接口访问功能不同的函数。实际上 C++ 的函数调用机制与 C 不同, C 的编译器根据源代码调用固定的函数标识符,然后由连接器接管这些标识等,并转换成物理地址,这一过程称为先期联编。在 C++ 中,只有向具有多态性的函数传递一实际对象,该函数才能与有多种可能的函数中的一种联系起来。因此 C++ 代码可以根据运行情况执行不同的操作。

2 设计具有 C++ 特征的应用程序

在当前学习 C++ 的用户中,大都是先掌握了 C 的程序设计技术,然后在此基础上学习 C++ 的,这样,他们就可能用 C 的思路来编制 C++ 软件,写出的程序只能是 C 程序的翻版,不能体现 C++ 的特征,这是不可取的。因此程序设计时应注意以下几点:

a. C 语言是结构化程序设计语言, C++ 是面向对象的程序设计语言,前者程序设计

的核心问题是函数,后者程序设计的核心问题是“类”,这是程序设计发展的两个阶段,它们的风格和设计思路完全不同。

b. 对于 C++ 来说,采用 OOP 方法,其核心问题是处理对象之间的关系问题,既类的层次结构设计问题。 Borland C++ 提供了由包容类库包含的各种“类”,实际应用时往往还需要更多不同类的对象,这些对象可以通过类的继承性派生出我们所需的类。类的层次结构实际上是一棵倒置的树,树根是类层次结构的根类,其它类是这个根类的派生类,而派生类又可作其它类的基类,根类以外的类与根类之间的关系是继承关系。 C++ 软件开发的主要任务是开发解决某个问题的类,在程序设计中,应充分利用这种层次结构关系。

c. 设计类层次结构时,应使代码和数据的重复率最小。派生类与基类相比,仅仅是增加与基类不同的行为或基类中没有的行为。在设计合理的类库时,每个类只是在基类的基础上增加了一些有限的代码,因此,使用类库进行程序设计,对程序的调试和维护都比较容易。使用类层次结构,将大大减少程序设计的复杂性,缩短软件开发周期。

总之, C 语言使用指针进行程序设计,程序简洁、精练,技巧灵活,充分体现了设计者的能力和水平。在程序设计中,将主要精力集中在多种指针应用,存储器动态分配和位操作的使用技巧上。但这样开发的程序可读性、独立性、可重用性差,程序设计工作比较复杂。 C++ 的程序设计工作的主要精力集中在“类”及其层次结构的设计方面,因而程序的可读性、可重用性、可移植性、安全性比较好,给程序开发带来较大的方便。

参考文献

- 1 刘炳文编著. Borland C++ 实用编程技巧. 清华大学出版社, 1995
- 2 [美] Ted Faison 著, 蒋维杜等译. Borland C++ 3.1 编程指南. 清华大学出版社, 1993

客户/服务器模式在海洋地理信息系统中的应用

天津师范大学计算机系 张纪成
国家海洋信息中心 刘 炜

摘要 本文在研究 GIS 发展趋势及海洋 GIS 特点的基础上,提出了在海洋地理信息系统中采用客户/服务器模式实现系统资源共享。并结合相应的软件作了简单实例分析。

关键词 海洋地理信息系统 客户/服务器模式 进程间通信

环境和发展是当今人类面临的两大主题,海洋在其中占有重要地位,增强海洋可持续发展能力,海洋资源与环境的科学管理是当务之急。其关键环节是信息。能对大量空间信息进行存贮,更新,分析并提供海洋规划、管理辅助决策的有效工具是地理信息系统(以下简称 GIS)。因此,应用 GIS 技术建立海洋 GIS 是一项意义重大的工作。

1 概述

1.1 客户/服务器(以下简称 C/S)模式

C/S 是分布式系统的一种模式,是由多个分散的计算机经网络互连而形成的计算机系统。其中,各资源单元(物理或逻辑的)即相互协同又高度自治。具体体现在:运行于工作站上的应用程序,通过访问服务器,共享设备文件和其他资源。服务器上运行的软件能响应其它工作站的请求,提供特定的服务。

C/S 有两个明显的特征:一是系统内有大量独立的、自治的资源单元。二是对用户而言该系统就像一个单一的系统。可将这两特征归纳为分离性与透明性,分离性是系统所固有的,它要求有网络通信能力,系统管理能力和实现同一的技术;透明性是对用户隐藏分离性,使系统成为一个整体;这是决定软件设计的主要因素,也是 C/S 模式最重要的特征。

1.2 海洋 GIS

GIS 是以具有地理位置的空间为研究对象,以空间信息为核心,采用空间分析方法与空间建模方法,提供多种空间的和动态的资源与环境信息的应用技术系统。

海洋 GIS 是以 GIS 为主要技术手段,以海洋及海岸带为研究主体的地理信息系统。该系统要求不仅能完成多类海洋空间信息的复合,同化,还能及时更新,维护海洋时空数据库。

2 C/S 应用于海洋 GIS

C/S 作为分布式计算的主流,实现了真正的资源共享。将这一计算方式应用于海洋 GIS 是当今 GIS 的发展以及海洋 GIS 本身的特点所决定的。

2.1 采用 C/S 是 GIS 发展的必然

GIS 作为一门技术是地理学、计算机科学、现代信息学的集成,其发展决定于相关学科特别是计算机软、硬件技术的发展。

在计算机辅助制图、数据结构理论、计算方法理论的支持下,GIS 最早是完成某一项目的工具。

随着关系型数据库管理系统的完善,数据库能够及时更新,GIS 提供的辅助决策能够跟上现实的变化,GIS 成为某一部门决策工具;伴随着计算机网络技术的迅猛发展,GIS 达到了企业级水平即 Enterprise GIS,

它允许公用数据和应用模型在网络上供更多用户使用,从而为一个机构的多个部门提供服务,这样使得用户增多、地理信息更趋分散。而基于 C/S 模式的分布式计算不仅使计算机之间得到互操作性而且方便、灵活,易于针对性地完成工作。目前国内外业务化的 GIS 大都在这一水平上。海洋 GIS 作为 GIS 在海洋上的应用也应把起点定于这一水平上,在海洋 GIS 中应用 C/S 是 GIS 发展的必然。

2.2 海洋 GIS 的特点

a. 海洋 GIS 空间性极强。地理数据是分布在空间区域中的,可以根据行政区或自然地理区组织地理数据,采集管理和维护这些不同区域的数据并没有因果,从属关系,而是相对独立平行的。中国是一个海洋大国,海岸线漫长、海域广阔,不同海区又有各自的特点,因此一个高效的海洋 GIS 应该是分布式的。

b. 海洋 GIS 专业性强。海洋 GIS 从信息的预处理,到专题信息的提取以及多类信息的复合都是专业性极强的工作。在信息处理中,各海洋要素都有各自的业务标准及相应的数学模式,涉及到海洋学、物理学、地质学、生物学、化学等众多学科。而具体问题的解决又需要诸多要素协同完成;由于海洋 GIS 用户精力在于自身专业。因而面向特征的数据管理和操作是必须的,这样就会避开某些计算机细节,体现系统的透明性。如:在一个高档工作站上定义的某数据特征应允许用户在低档 PC 上调用来建立自己的数学模型。这也正是分布式计算的重要特征的体现。

c. 海洋 GIS 动态性、时域性强。广阔的海洋是时刻运动中的水体,这种海洋本身的运动特性决定了海洋 GIS 的时域性强。以遥感为主要数据源及时更新数据是解决这一问题的主要方法。分布式计算允许在多用户环境下进行数据更新,从而保证 GIS 决策的实效性。

d. 海洋 GIS 数据量大。海洋各要素处于广阔的动态空间中,受众多因素的影响。例如:泥沙含量受海洋要素的影响,海面风会阻止悬浮泥沙下沉,波浪又会掀起海底沉积物,潮流能使海水夹沙混合运动增强;温差会造成海水上下动;沉积物和底质会影响悬浮泥沙运动。因此,对海洋观测不能仅采用某一种手段,应形成台站、船只、飞机、卫星立体观测网,这样会使信息量大增,对于如此多的资料如果重复收集和管理就会造成数据冗余,同时,也不可能保持多个独立数据库的维护与及时更新,会产生不一致,表现在:坐标系统,比例尺,数据时效等。由此导致决策的低效与混乱。对于大量数据采用分布式管理会避免上述情况的发生。

e. 海洋 GIS 是一个开放的系统。一个单一封闭的系统不符合当今工业标准、GIS 应具有与其它专业软件如 SAS、CAD、图像处理、办公自动化等兼容的能力。通过基于 C/S 的分布式计算,GIS 可以方便灵活的与其这些业务系统协同工作,以弥补自己的不足,为辅助决策增添新形式。

综上所述,海洋 GIS 必须采用 C/S 模式,只有这样才能满足海洋 GIS 应用的要求,彻底实现信息的共享,增强辅助决策的可信度。

3 实例分析

将计算机联成网络,实现打印及文件共享并不是真正的 C/S 模式,C/S 模式除了具备资源共享的特点外,还能使任务处理由客户(C)和服务端(S)双方分担(较高级复杂任务由服务端来完成)。一个典型 C/S 模式有三个部分组成:客户、服务端以及二者之间的连接件。这种基于硬件的连接在实际应用中表现为软件通信过程,如网络协议和应用接口等。被称之为进程间通信(IAC)

3.1 C 与 S 的通信

客户与服务端进程间通讯的过程如图 1

所示。

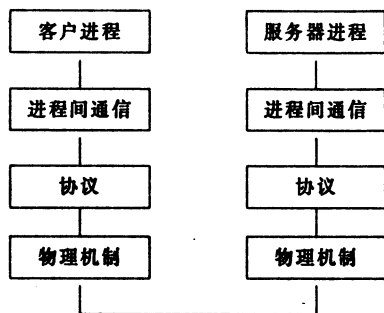


图1 客户与服务器进程间通讯的过程

由图1可以看出,在客户与服务器进程之间,基本上有三个通信层:进程间通信、协议、物理机制。分别覆盖 OSI 网络参考模型定义的 7 个标准层。其中物理机制是网络连接提供的,如网卡、网线、集线器等。协议是网络各节点之间通信规范,由网络操作系统来确定。

可见,进程间通信(IAC)是关键。进程间通信主要包括二种情形:会话,远程调用。

3.2 IAC 的实现

现以 ESRI(美国环境研究所)的 ARC/INFO 与 ARCVIEW 软件(因其用户群庞大)为基础,讨论一下 C/S 的具体实现。

RPC 远程过程调用,允许客户程序 ARCVIEW(在微机平台上,操作系统为 MS-DOS)通过网络向服务器端 ARC/INFO(在 SUN 工作站平台上,操作系统为 UNIX)发出请求,即远程任务提交,服务器端完成任务后,将结果返回客户端(ARCVIEW)。这样充分发挥了 ARC/INFO 的强大分析功能,同时满足了用户的不同要求。这种 C/S 是一对多的,即 ARC/INFO 根据先后顺序完成多个客户程序的任务,对一个大型系统来讲无疑是大有益处的。在下面的例子中以两层数据的连接为例说明这一过程。

'建立客户与服务器间的 C/S'

client=RPCClient. Make("ygs01",0x40000000,1)

'给出输出层名

```

outCov = FileDialog. Put ("outcov". AsFileName,"
* ","What is the UNION Output Coverage?")
theView=av. GetActiveDoc
tList=theView. GetActiveThemes
'获取数据层名,并产生连接命令
aiCommand=
"UNION"++
tList. Get (0). GetSrcName. GetFileName. GetFull-
Name++
tList. Get (1). GetSrcName. GetFileName. GetFull-
Name++
outCov. GetFullName++
"#NOJOIN"
MsgBox. Report (aiCommand,"ARC/INFO Com-
mand String")
'向 ygs01 服务器发出执行命令
JobID=client. Execute(1,aiCommand,String)
'等待返回结果,并设定等待时间
busyWaiting=true
while(busyWaiting)
    if (client. Execute(2,JobID,String). Extract(0)
    ="DONE")then
        busyWaiting=false
    else
        System. Execute("sleep 5")
    end
end
client. Close
'将返回结果作为新数据层(已命名过)
srcString=outCov. GetFullName++"polygon"
newSource=SrcName. Make(srcString)
newTheme=Theme. Make(newSource)
'结束
  
```

参考文献

- 1 陈述彭. 遥感大辞典. 北京:科学出版社. 1990
- 2 ESRI white paper series, Enterprise GIS — Using GIS in the Corporate Environment, May 1995
- 3 ESRI UNDERSTANDING GIS
- 4 Brian Butler. 客户/服务器实用技术指南. 北京:电子工业出版社.

工业锅炉单片机监测调节和辅助管理系统

天津理工学院

张毓瑾

天津职工冶金大学 尹蓉华 蒋静敏

摘要 本工作是用单片机构成最小系统,对工业锅炉的运行参数进行巡回检测,实现对锅筒水位、供煤量、送风量和炉膛负压的闭环控制,计算热效率,累计各运行班产出的蒸汽量和煤水电的消耗量,以提高锅炉的运行效率、安全性和管理水平,本系统已连续运行三年,工作一直正常。

关键词 单片机应用 锅炉控制

直接数字控制的常用设备是各种标准总线 and 可编程控制器。使用单片机的例子虽也已屡见不鲜,但其在高温、高湿、强电磁干扰等恶劣环境中呈现的性能尚需反复验证和摸索使用经验。此外,车间级的运行管理一般是用微型计算机作上位机,用其接收下位机传来的数据,再做相应处理。能否用同一个单片机系统既完成闭环控制任务又实现运行管理所需数据的采集、处理并自动生成,显示和打印各种报表数据呢? 这些就是本工作想要进行的尝试。

1 设计目标

1.1 样机应用环境选择

为检验系统的抗干扰能力,样机运行环境选择在钢厂内被众多大型动力设备包围着的热力车间。该车间有四台链条炉排蒸汽锅炉常年交替运行,内部环境高温、高湿。

1.2 样机功能设计

a. 因该车间的四台锅炉经常是两台运行两台备用,故本系统按同时对两台锅炉的控制和管理设计,但可随时切换到另外任意的两台。

b. 对每一台锅炉要采集的数据有九个,即锅筒水位、蒸汽压力、蒸汽流量、给水流量、给煤量、送风量、烟气含氧量、炉膛负压和电耗。其中煤耗和电耗是脉冲量,其余七个都是模拟量,故系统要有十四路模拟量输入通道和四路脉冲量输入通道。

c. 对每一台锅炉要实现给水量、给煤量、送风量和引风量的自动调节,故系统应有八路控制信号输出通道。

d. 锅炉热效率是运行管理水平的标志。对于饱和蒸汽锅炉,在不考虑蒸汽湿度、排污量和自耗汽情况下,其计算公式为^[2]:

锅炉热效率 = $(\text{蒸发量} \times (\text{蒸汽焓} - \text{给水焓})) / (\text{每小时煤耗} \times \text{煤的低发热值}) \times 100\%$
式中,每小时的蒸发量和煤耗是测量值,而蒸汽焓、给水焓与蒸汽压力有关,燃煤的低发热值与所用煤的种类有关,均有表可查^[2],故在外接 ROM 中要存入相应表格中的有关数据。

e. 因系统初始化的需要,并为实现采集和处理数据的显示和打印,以及对显示、打印和控制的人工干预,系统应配置十六键键盘和八位 LED 显示器及 TPμP 微型打印机。

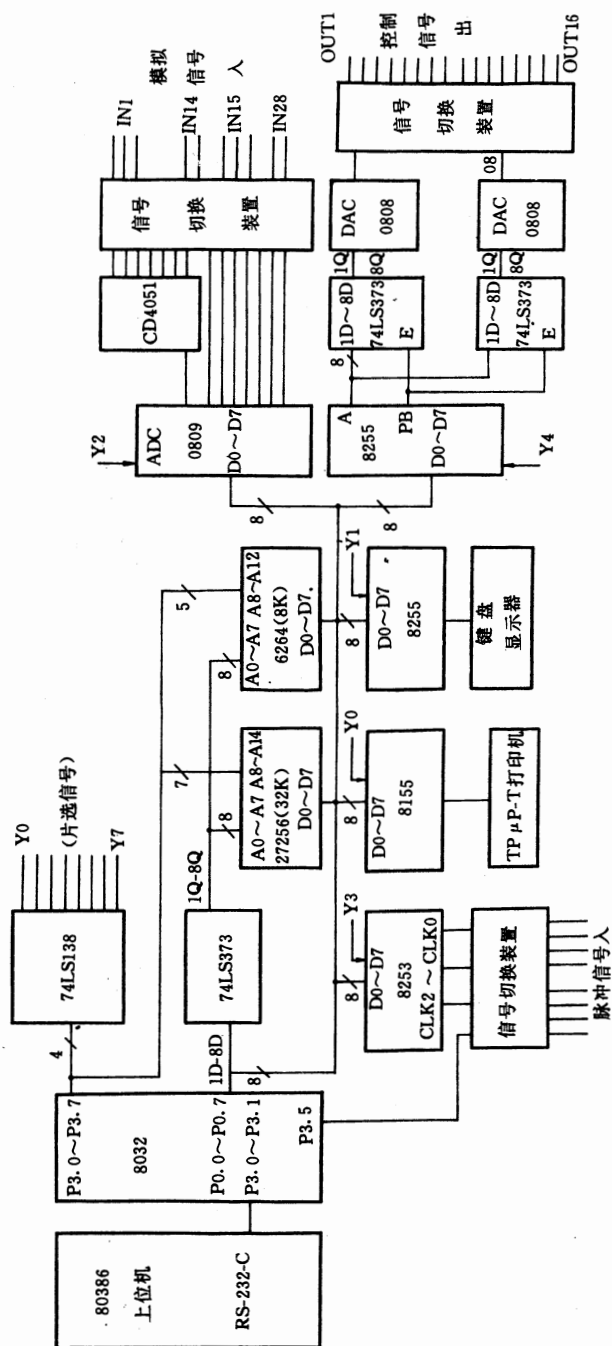
f. 因平年和闰年各月的天数不一样,故需编制软件日历和时钟程序,以实现按月按日按班和按小时的自动显示和打印。

g. 为使每天采集和整理过的数据能有更多的用途,还需设计与上位机的通信程序,并为使上位机有充分选择传送时间的灵活性,设计在外接 RAM 中安排能存放连续三十小时上述数据的缓冲区,这样,上位机在第二天上午四小时内的任意时刻接收均可。

2 系统的组成与实现

本设计的硬件系统由电源、主机、数据采

用了现成的稳压电源产品,其余七部分电路的总框图示于图1。



8253 的三个门控(GATE0 至 GATE2) 并在一起由 8032 的 P_{1.7} 信号控制, 系统启动时使其为高, 写入方式控制字, 选择工作方式 4, 再写入计数初值, 计数器即开始计数。

采样时, 先用软件置门控信号 GATE 为低, 使各计数器停止计数, 再按各计数器地址依次读出计数值; 之后, 恢复门控信号为高, 各计数器便自动恢复初值并重新开始计数。

2.2.3 数据存储器的掉电保护。

外部数据存储器 6264 使用了掉电保护电路。V_{cc}>3.6V 时, 串联的两个二极管导通, 由 V_{cc} 给 6264 供电并给电容充电; 同时通过 R 电阻给电池充电。当 V_{cc} 断电时, 电容 C 放电(为电池反向充电), 待其电压降至 3.6V 以下时, 电池开始为 6264 供电。

2.3 控制调节电路

对两台锅炉输出的控制信号共八个。因 8032 的 P₀ 口只能驱动八个 TTL 门电路^[4], 故没有采用可直接与 8032 接口的 DAC0832, 而是使用了较早的 DAC0808, 由 8255 驱动。

所用执行机构都是标准的 DDZ 电动组合单元, 所需信号均为 0~10mA 直流电流, 故均可与 0808 的输出直接相连。

2.4 键盘和显示器电路

为循环显示蒸发量和煤水电耗的采样值, 并每间隔一小时、八小时、一天和一个月上述各量的累计值的自动定时显示, 样机上设置了八个带字驱动和位驱动的 LED 显示器。另有十六个按键用于输入 0 到 9 十个数字, 完成删除、回车、复位和某些设定的功能(热效率计算、自动调节系统的人工干预等)。

2.5 打印机接口电路

样机直接带 TP_μP-T 微型打印机。用软件使其能在规定时间自启动并按设计格式打印指定数据。

8032 与打印机的接口电路示于图 4。图中, BUSY 为高电平时表示打印机正忙不能接收数据。STB 是数据选通触发脉冲输入

端, 上升沿时读入数据。

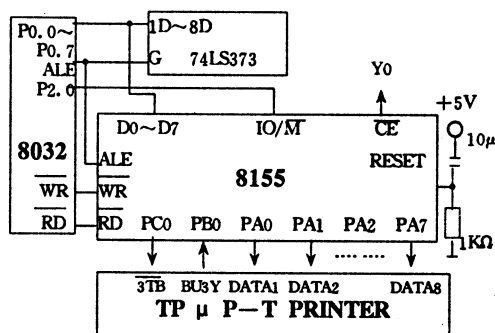


图 4 打印机接口电路

2.6 与上位机通信电路

设计为样机每天要向上位机传送一次数据, 所以在数据存储器内开辟了 1KB 的空间作为缓冲区(地址从 1200H 到 14FFH), 用于保存连续三十小时内的当值运行班编号和两台炉每小时产生的蒸汽量、煤水电耗、以及其它处理结果。工作人员在第二天上午四小时内的任何时刻启动上位机进行接收均可, 但不可再延迟, 否则该缓冲区将被新的数据覆盖, 每次传送结束时将该缓冲区清零。

2.7 抗干扰与容错

- 采用抗干扰性能优良的直流稳压电源单独为本装置供电。线路板上无强电部分, 减少了电磁干扰。

- 电路板设计制作上考虑了元件布局的合理, 采用双面走线并使走线尽可能短、尽可能宽、尽可能不在集成芯片引脚间通过, 在一些暂态陡峭电源电流的器件处增加去耦、滤波和旁路电容等措施。

- 整个系统单点接地。专用厚 3 毫米、长宽均为 0.5 米的紫铜板为接地电极, 深埋于地下 3 米处, 地线引线采用了厚 3 毫米、宽 5 厘米的铜排, 构成了本装置的一个良好的接地系统。

- 在引入脉冲输入信号时加 4584 芯片, 使输入脉冲整形并滤去干扰。

- 软件设计上对按键处理采用了去抖措施和操作错误处理、程序和数据区的保护、程

序跑飞的自恢复等容错处理。

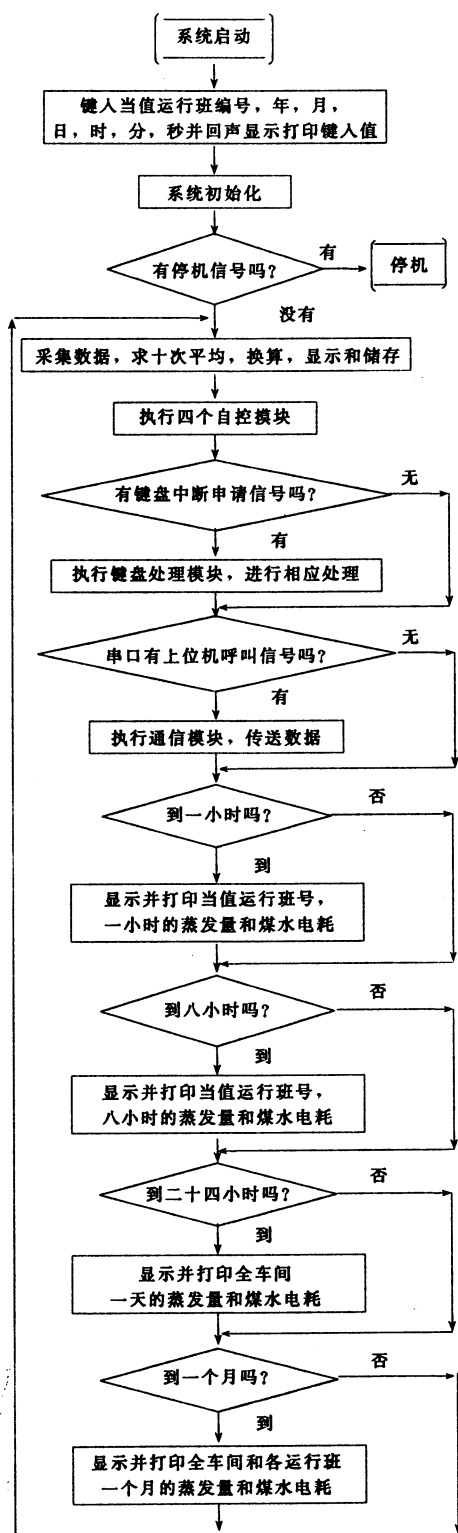


图5 应用程序的概念级流程图

3 应用程序

应用程序包括十二个模块,即主控、数据采集和处理、四个闭环控制、软件日历和时钟、数据统计、键盘处理、显示、打印和通信,全部用汇编语言写成。限于篇幅,这里只能给出概念级流程图,示于图5。

4 结束语

本装置已连续运行三年,工作一直正常,说明只要采取适当的抗干扰措施,单片机系统在恶劣环境中还是可以正常工作的。

该装置在完成四个闭环控制任务以外,还完成了一定的数据处理和自动定时输出工作,说明单片机系统可以不与上位机联机而独立承担运行控制和管理工作。

因为一天之内上位机完成与单片机通信任务只需要几秒钟,有充裕的时间可以进行其它工作,故还编写了设备管理、备品备件管理、人事管理、财务管理、双达标系统和与上级管理部门通信的程序、并都装入上位机交替运行。三年的使用情况说明这个方案还是可行的。所以,对于一个小规模的车间,只需一台386微机和一台廉价的8032单片机应用系统就可以构成完整的巡检、控制和辅助管理系统。

参考文献

- 1 何立民主编. 单片机应用文集. 北京: 航空航天大学出版社, 1993
- 2 张永照等编著. 工业锅炉. 北京: 机械工业出版社, 1985, 18~30
- 3 上海计算机技术服务公司. 国外常用集成电路简明应用手册. 上海: 上海计算机技术服务公司, 1993

一种由 DSP 芯片 TMS320C30 与计算机构成的声频实时分析系统

安徽大学电子工程与信息科学系 柴晓冬 李晓辉

摘要 本文介绍了由 TMS320C30 与计算机构成的声频实时分析系统,并在此系统上对扬声器谐波失真等参量进行了测量与分析,给出了由 TMS320C30 汇编实现的可变数字振荡器的方法

关键词 DSP 实时处理 谐波失真

1 引言

DSP 是集微电子学,数字信号处理与计算技术与一体的新器件,其广泛用于语言处理。图形图像处理以及实时信号处理等多种领域。由于 DSP 芯片采用了与通用微处理器不同的结构,即哈佛结构,使得程序存储器及其总线与数据存储器及其总线分开,使处理指令和数据可以同时进行,从而大大提高了处理速度。在数字信号处理算法中,乘法和累加是基本的大量的运算,如卷积,FFT,FIR 和 IIR,矩阵运算,矢量搜索等,都大量类似于:

$$\sum A(k)B(n-k)$$

一类的运算。通用计算机的乘法是用软件来实现的,往往需要若干个机器周期才能完成,而 DSP 中都设置了硬件乘法器和 MAC(乘法并累加)一类指令,这些操作往往可在单个

指令周期内完成,从而提高了速度。由于微机本身的运算速度受到限制,即使在声频范围内也无法达到实时分析处理,因而将 DSP 与微机结合构成系统则加快了运算速度,使一些理论和算法能实时实现。

在声频信号分析中,对扬声器各参数的测量是信号分析的主要任务之一^[1]。传统的测量分析需用到众多设备,而且精度,动态范围,稳定度等受到限制^[2]。而用 DSP 芯片构成的测量系统,其实质和微机相配合,使得在 80KHz 以下频率范围能达到实时分析处理,其性能指标达到或超过现时使用的各种信号测量分析仪器,且可将各参数的测量结果用图表形式输出。

2 系统构成及原理

测量系统的构成如图 1 所示,由换能器

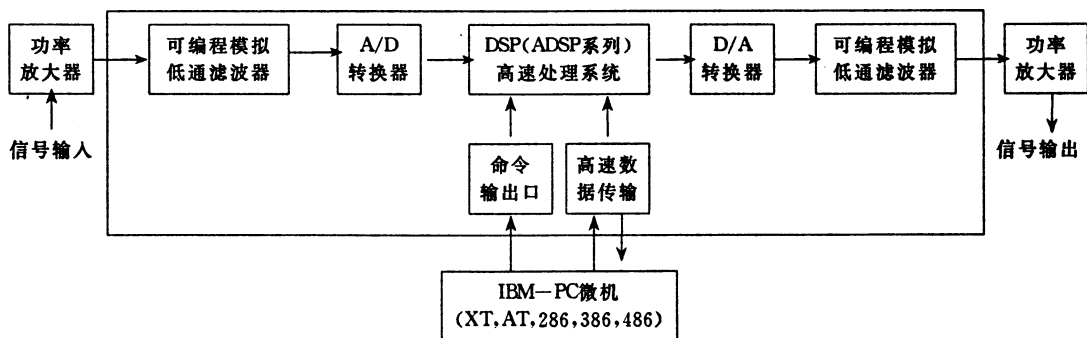


图 1

转换得到的电信号经输入放大器放大后,由防混叠滤波器作低通滤波,以防止采样的混叠效应。防混叠滤波器的截止频率根据测量分析的频率范围,由程序控制。滤波器的输出信号,经电缆送至插在微机内的数据采集板,以进行模数转换。模数转换的通道数及采样率均通过程序来设定。在每一采样周期中数据采得后,数据采集板给 TMS320C30 数字信号处理板发一中断信号,C30 则中断 CPU 的执行而执行一个中断服务子程序,将并行口中的数据读入并放在存贮器的某一地址的单元中,C30 也可通过 DMA 的功能,在不中断 CPU 的情况下并行地将该数据读入并存放在存储器某一单元中。对扬声器各参数的分析,则用 C30 汇编语言编成程序来完成,从而达到实时分析的目的。

3 微机与 TMSC30 之间主/从系统的管理和编程

TMSC30 程序存贮器由微机和 TM-SC30 共同管理,本系统采用的方法是:微型计算机用几个锁存器锁存微型计算机数据总线上的数据。使这些数据产生一个相对于 TMSC30 系统的程序存贮器的地址线或相应的控制信号。再通过其他的几个 I/O 地址读 1 写相应被选中的 TMSC30 程序存贮器的地址单元中的数据。这样微型计算机就可利用很少的资源来管理 TMSC30 的全部程序存贮器。即可进行微机与 TMSC30 之间进行控制,通信,中断等方式的进行。

下面的例子是用来说明微型计算机访问 TMSC30 程序存贮器,即往程序存贮器 0X100 至 X200 单元写数据,偶地址单元放 0X555555555555,奇地址单元放 0XAAAAAAAAAAAA,并且再读回 PC 机并显示。其 C 语言程序如下:

EXAM1.C

```
#include<stdio.h>
#include<io.h>
```

```
#include<dos.h>
#include<conio.h>
#include<malloc.h>

#define PORT9 0X369
/* CONTROL PORT ADDRESS */
#define PORT0 0X360 /* 21020 PMD 0--7 */
#define PORT1 0X361 /* 21020 PMD 8--15 */
#define PORT2 0X362
/* 21020 PMD 16--23 */
#define PORT3 0X363
/* 21020 PMD 24--31 */
#define PORT4 0X364
/* 21020 PMD 32--39 */
#define PORT5 0X365
/* 21020 PMD 40--47 */
#define PORT6 0X366 /* 21020 PMA 0--7 */
#define PORT7 0X367
/* 21020 PMA 8--15 */
#define PORT8 0X368
/* 21020 PMA 16--23 */
#define SEG 0XD000
/* PC DUAL PORT ADDRESS */
#define I 0X02
#define J 0X03
#define K 0X00
main()
{
    long int i,i1,i2,addr,in_data,*out_data;
    out_data=malloc(1024*sizeof(int));
    outp(PORT9,K);
    for (addr=0x101;addr<0x111;addr++){
        i1=addr>>8;
        i2=addr & 255;
        outp (PORT7,i1);
        outp (PORT6,i2);
        outp (PORT8,0);
        if (addr%2==0)in_data=0x55;
        elge
            {in_data=0xAA;}
        outp (PORT0,in_data);
        outp (PORT1,in_data);
        outp (PORT2,in_data);
```

```

outp (PORT3,in_data);
outp (PORT4,in_data);
outp (PORT5,in_data);
}
for (addr=0x101;addr<0x111;addr++){
i1=addr>>8;
i2=addr & 255;
outp (PORT7,i1);      /* output PMA0-7 */
outp (PORT6,i2);      /* output PMA8-15 */
outp (PORT8,0);       /* output PMA16-23 */
out_data[0]=inp(PORT0);
out_data[1]=inp(PORT1);
out_data[2]=inp(PORT2);
out_data[3]=inp(PORT3);
out_data[4]=inp(PORT4);
out_data[5]=inp(PORT5);
    for (i=0;i<6;i++){
printf("%02x", *out_data);
printf("\n");
}
free((long int *)out_data);
}

```

从此例可以看出微机访问 TMSC30 系统的程序存储器就和访问微机本身的存储器一样,只要 TMSC30 系统的存储器交给微机管理,就相当于微机的扩展存储器。

4 数字振荡器的设计

在对扬声器谐波失真等参数的测量时,都需有信号源,如正弦波信号,而该信号是由数字信号处理系统中的程序产生的。所产生的数字信号经电缆传到放大器中的重构滤波器进行低通滤波。(滤波器的截止频率则根据所产生的信号频率范围而由程序控制)由 TMSC30 构成的数字振荡器,只要改变其几个参数,就可得到不同频率的正弦波。

利用正弦波 $\sin x$ 的指数形式

$$\sin x = \frac{1}{2j} (e^{jx} - e^{-jx}) \text{ 可得正弦序列:}$$

$$x[k] = \sin K\omega T$$

将其 Z 变换,求出其复根,即可得到一个递归的差分方程:

$$y[n] = Ay[n-1] + By[n-2]$$

该方程在 $K > 2$ 以后, $y[K]$ 能用 $y[K-1]$ 和 $y[K-2]$ 算出,这是一个递归的差分方程,该序列即为 $\sin K\omega T$ 。

由于数字频率是模拟频率 f_d 与采样频率 f_s 之比,设该振荡器的频率为 $f_d = 1\text{kHz}$,采样频率为 $f_s = 10\text{kHz}$,则系数

$$A = 2\cos\omega T = 2\cos 36^\circ = 1.618033989$$

$$B = -1$$

$$C = \sin\omega T = \sin 36^\circ = 0.587785252$$

将 A、B、C 代入(6)式,可得到一个正弦序列。当改变 A、B、C 的值时,就可改变振荡频率,而不用重新编写程序,程序框图(略)

5 扬声器谐波失真的测量

由于扬声器的非线性失真,在扬声器的输出的声压信号中,总会有输入信号所没有的频率成分。当扬声器输入一个正弦信号时,在该信号的频率(基频)的倍数频率上,会有一些谐波分量,由失真产生的第几次谐波声压有效值与总声压有效值之比称为第 n 次谐波失真

$$\text{总谐波失真系数 } dr = \sqrt{\frac{P_2^2 + P_3^2 + \dots}{P_1^2 + P_2^2 + P_3^2 + \dots}} \times 100\%$$

式中 P_{nf} 为第 n 次谐波的声压有效值, P_1 为基波声压的有效值。图 2 图 3 为所测结果。

由图 2 图 3 的结果可以看出,图 3 的谐波失真在频率较高时明显好于图 2,失真小。其单一频率的 n 次谐波失真也好于图 2,即第二种型号的扬声器其谐波失真要好于第一种型号的扬声器。

由于 TMS320C30 具有很强的实时处理能力,因而在声频信号分析中具有广阔的应用前景。如对扬声器的阻抗,幅频特性,频率响应以及混响时间等的测量。当然对这些参数测量时,其激励信号也要相应变化,即须产生白噪声或粉红噪声,扫频信号或窄带噪声信号作为激励信号,而这些激励信号均可由 TMS320C30 中的程序来生成。

气弹簧测试系统的设计

河海大学计算机及信息工程学院 张亚新

摘要 力—位移曲线是气弹簧性能的重要技术指标。本文介绍了用微型计算机和单片机构成的气弹簧测试系统。系统操作简便、测量精确、工作可靠。

关键词 单片机 微型机 检测

1 引言

压缩气弹簧在航空、汽车的座椅生产中被广泛使用,目前国内有许多厂家生产。由于检测设备的落后,生产厂家只能进行耐腐蚀及抗冷冲击等简单的性能测试,而决定压缩气弹簧技术性能的主要技术指标是气弹簧的动、静态力特性及活门开启特性等,其中动、静态力特性也就是气弹簧在压缩过程中的压缩力和伸展过程中的伸展力的变化情况,此特性由力—位移曲线来描述。本文所述的就

是利用计算机技术构成的专门用于气弹簧力—位移曲线测试的检测系统。

2 系统简介

图1是气弹簧检测系统的结构简图,整个系统由测试台、控制器和计算机系统三部分组成。

测试台由机架、电机、减速机、传动齿轮、齿条及测试头等零部件组成,上面装有变频器、压力传感器、光电编码器等电气器件。由

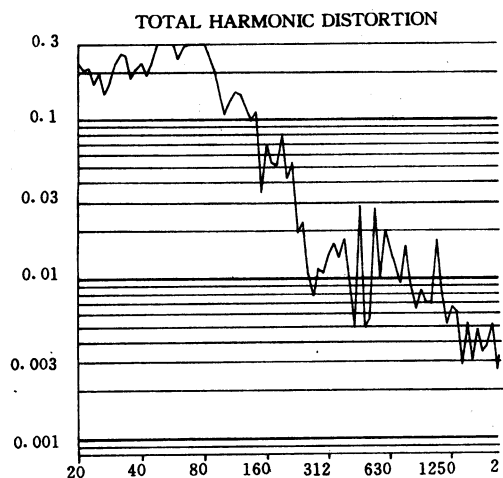


图2

参考文献

- 1 彭启琮. "DSP 实时数字信号处理". 电子科技大学出版社, 1995

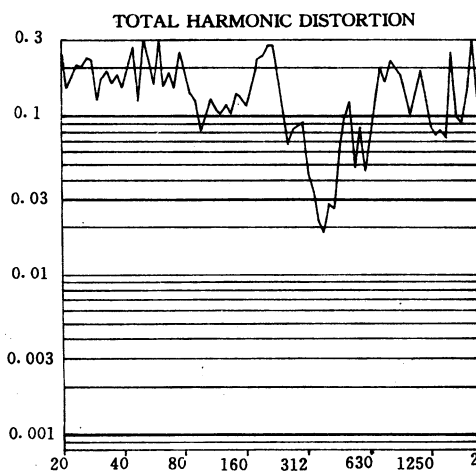


图3

- 2 语言及信号处理国际会议论文结集, 第二卷, 1996
- 3 郑君里. 信号与系统. 高等教育出版社, 1992

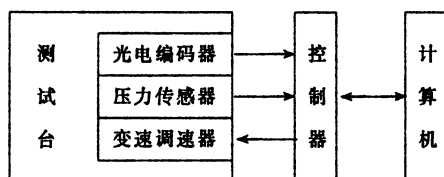


图1 系统结构简图

控制器控制测试行程及速度来采集测试数据,完成气弹簧的测试。

控制器以 8098 单片机为核心,一端与测试台上的压力传感器、光电编码器和变频器相联接,采集测试过程中获得的压力、位移等数据。另一端与计算机相联,接收由计算机设定并传送来的各项测试参数,并把测试所得数据上传给计算机。

上位计算机采用一般的个人 PC 机,外加一台 24 针点阵打印机,运行编制的软件,完成各项测试参数的设置并下载到控制器,接收控制器采集上传的测试数据,进行处理运算,画出和打印力一位移特性曲线。系统工作流程如图 2 所示。

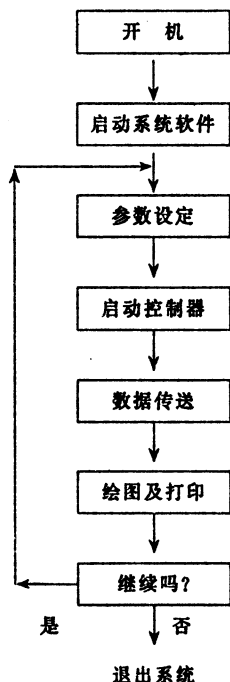


图2 系统操作流程

整个系统以 PC 计算机为用户操作界面,各种操作通过计算机键盘完成,使用方

便,测试精度高,工作可靠。

3 硬件组成

图3是本测试系统硬件接线图。

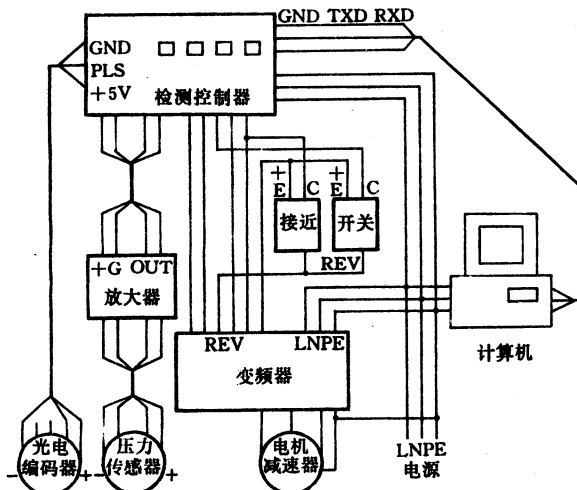


图3 系统硬件接线

安装在测试台上的光电编码器用于位移的测量,通过位移脉冲计数,确定位移行程;两个接近开关用来锁定位移行程在一定的范围内,防止电机误动作而损坏气弹簧;压力传感器和放大器电路完成气弹簧压缩力和伸展力的测量与放大;检测仪与 PC 机的通信采用串行口,工作在半双工方式,用 8098 单片机的 RXD 和 TXD 引线,通过 1488 和 1489 进行电平转换后与 PC 机串行口相连,简单实用。检测时,把气弹簧活塞杆向下利用两端连接头垂直固定于测试台上,由变频器和电机驱动顶杆,按规定的行程和速度压入气弹簧活塞杆,同时测量位移及各点的压缩力,达到行程要求后,撤去外压力,这时气弹簧活塞杆自由伸展复位,测试位移及伸展力。

4 软件设计

测试系统的软件由上位 PC 机软件及单片机软件组成。单片机软件包括系统初始化、控制器按键处理、压力信号的 A/D 转换、位

单片机在压力测控仪中的应用

南京金陵职业大学 朱一纶

摘要 介绍以 MCS-51 单片机为控制器的压力测控仪的设计原理,对系统的硬件结构及软件设计作了较详细的说明,在软件设计中采用报警时用报警号显示不同的报警原因以及通过建立信号灯的方法使程序条理清楚,功能灵活,执行速度快。

关键词 单片机 监控 报警号 信号灯

在土木工程中,需要对土壤的容许承载力,压缩(变形)模量等物理、力学指标进行测试,在试验过程中,要求对土壤试样逐渐加压并测试其内部应变力,试验时间长,我们以

8031 单片机为控制器开发设计的三轴压力测控仪根据测控规程要求编程,根据实验中土壤受压应变的情况控制加压,仪器采用数字显示,测量精度可以达到百分之零点五,且

移的计算、采集数据的存储及串行通信的处理等模块。为方便待测气弹簧的安装和测试,在控制器面板上设计有手动控制测试头上下运动的点动按键、测试启动键和复位键,通过相应的软件控制测试头点动运动。

PC 机软件用 TurboC2.0 语言编制,采用西文状态下显示汉字的技术,人机界面友好,操作方便。软件包括测试参数设定、数据传输、画动态曲线、画静态曲线等四个模块,以屏幕顶端菜单的形式显示,底行显示键盘操作说明,用户可用光标移动键左右移动彩色框来选择模块,软件同时在屏幕中央给出该模块的功能说明,按回车键则执行该模块。测试参数设定完后,自动下传给单片机。启动控制器测试完后,再执行数据传输将数据上传到 PC 机,然后就可画出气弹簧的动、静态力一位移曲线图,并可选择是否打印曲线图。

系统软件的关键部分是 PC 机与 8098 单片机的通信,为保证通信可靠和方便软件设计,让单片机工作中断方式,以 PC 机为中断源,不论是参数设定后的数据下传,还是控制器测试完成后的数据上传,都由 PC 机发出中断“握手信号”开始。数据下传时,单片

机中断处理程序每接收一个数据就回送一次“握手信号”,以示数据传送正确,PC 机再传下一个数据。数据上传时,单片机中断处理程序每发送一个数据就读一个 PC 机回送的“握手信号”,直到数据传送结束。单片机中断处理中,数据的上传与下传用不同的“握手信号”来区别。在系统设计中,采用 PC386DX40 计算机,通讯速率设定为 2400 波特,传送数据快捷可靠。

5 结束语

本文所介绍的气弹簧测试系统以单片机为核心组成测试控制器,测量精度高,采用 PC 机为操作界面,用户使用方便。是基于 PC 的集成化测试系统的设计。系统使用一年来,工作稳定可靠,满足了气弹簧生产厂家对气弹簧主要技术指标测试的要求。

参考文献

- 1 张毅刚. 8098 单片机应用设计, 电子工业出版社, 1993
- 2 王军政. Turbo C 实用高级编程技巧, 科学普及出版社, 1993

可以和微机通过串行口相连,实现自动记录测量数据,从而大大减轻了实验工作人员的劳动强度,提高了工作效率。

1 系统的构成及工作原理

三轴压力测控仪由三轴压力室,四周压力加压装置,反压力加压装置以及自动测控系统构成。见图1。测控系统通过输出信号控制低速同步电机进而控制试样周围及上方的压力。最高压力为2MPa。通过压力传感器对四周压力X3,反压力Xb及土壤的压力U进行测试,反馈信号送至测试系统进行处理。

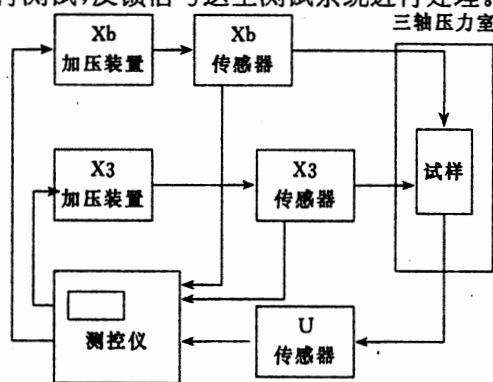


图1 测控仪结构图

2 测控系统硬件设计

测控系统硬件电路主要分成五个模块,如图2所示。

2.1 8031 控制模块

单片机控制系统由8031单片机以及74LS373,2764构成,并用一片2864保存必要的数,使测控仪具有掉电保护数据的功能。同时电路中设置了由为了扩展输入输出接口,增加了一片8155和一片8255,8255用来管理键输入和控制发光二极管与数码管显示。

8155的时钟输入端接8031的ALE端,通过对8155的计数器编程使之输出频率为250KHz的时钟脉冲信号。8155主要管理数

据采集等。为抗干扰,以MAX705构成看门狗电路。

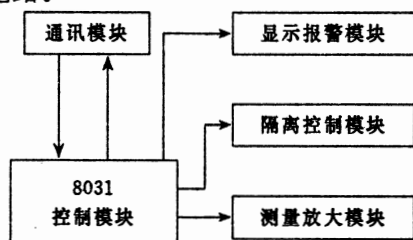


图2 硬件电路框图

2.2 测量放大模块(见图3)

要提高电路的测量精度,这个模块的设计是比较重要的。测量放大电路由LM324构成仪用放大器,对传感器送来的微弱信号进行放大,要严格挑选电阻,使仪用放大器具有很好的对称性。放大倍数可以通过调节精密电位器进行调节。

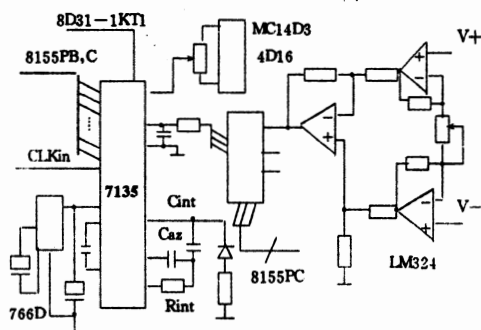


图3 测量放大电路

A/D转换采用双积分型的4(1/2)位芯片ICL7135,该芯片具有大规模集成电路的优点,只要外接高精度的积分电容,电阻和标准电源就可以完成A/D转换,输出BCD码,测量电压范围为0~2.0000V,测量分辨率为0.0001V,满足系统的精度设计要求。

积分电容要选用聚丙烯电容,积分电阻选用精密线绕电阻,通过采用8031中断查询方式采集测量数据,提高输给7135计数的时钟脉冲频率至250KHz,积分电容为0.22uF,积分电阻为100K,一秒钟可测7次左右。在设计线路时积分电容、积分电阻要靠近7135,尽可能减少线路干扰。

2.3 隔离控制模块(见图 4)

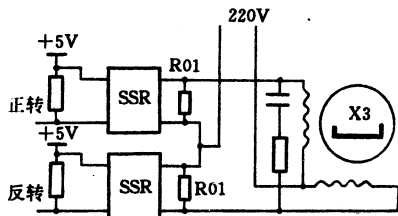


图 4 隔离控制电路

隔离控制模块采用固态继电器(输入直流电压 3—10V)作为隔离控制手段,具有反应迅速,无触点寿命长的优点。控制同步单相交流电机(转速:60 转/分)进行加压或减压。为防止过压,用行程开关控制,输出过压信号给 8031,调用中断子程序及时报警和停止电机。

2.4 通讯模块

通讯模块采用专用通讯集成块 ICL232,电路简单,测控仪通过串行口与 PC 计算机相连,计算机用 TURBO C 语言编制程序,当 8031 接收到 PC 机的命令后,设立联机标志,然后根据 PC 机传送的命令可以作两种处理:一种是由 PC 机对测控仪进行实时监控,即每次测量后把测量数据及时送给 PC 机,这种处理方式是为用一台 PC 机同时监控多台测控仪或其他仪器作准备。另一种 8031 把前一段时间(一次试验或一天)中测量获得的数据(储存在 EEPROM 2864 中)传送给 PC 机,用 PC 机对测试数据进行分析处理,绘制曲线,打印输出。

2.5 显示与报警模块

该模块属于测控仪的主要人—机界面,通过声光信号告知操作者测控仪的工作状态和测量值,报警声由 555 定时器构成多谐振荡电路,通过喇叭发出,数字显示电路采用了二—十进制寄存/译码/驱动/显示四合一集成芯片 ZCL002,具有体积小,设计方便的特点。在该电路的设计中,主要有以下特点:

- 在测控过程中,可以随时通过按键输入切换显示各测控对象测量值。

- 仪器的报警有多种原因,如:哪个测量值过大,哪个电机加压已到极限,设置值不符合要求,但面板有限,不可能设置许多报警灯,故设计了报警号。因为测量值最大为 2000KPa,针对各种情况取大于 2000 的号码编制了一套报警号,当报警时,报警灯亮,且数码管显示警号,使操作者可以迅速判断仪器报警的原因,采取相应的措施。

3 系统软件设计

测控仪的软件采用模块化设计方法,按照设计要求,主要分三部分,测控,设置和饱和试验,用键进行切换。主程序流程图如图 5 所示。

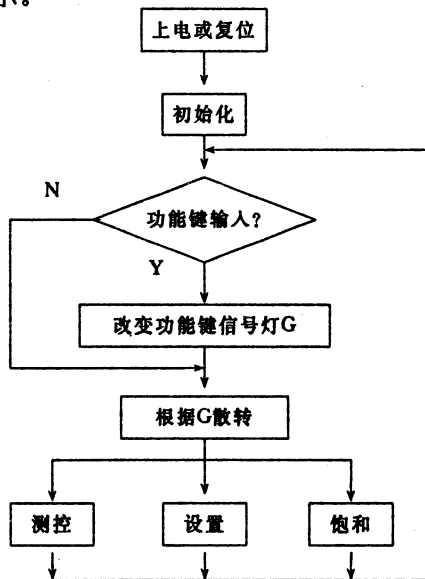


图 5 主程序流程图

采用中断方式通讯子程序,根据 PC 机命令直接送储存在 2864 中的数据或设置信号灯,进行实时监控。

因为测控仪为便携式仪器,要求体积小所以对输入键和输出显示有一定限制,我们设计了一键多义的软件程序。在软件设计中,利用建立信号灯的方法使程序简洁且功能完善。

3.1 测控子程序

测控为主要工作状态,开机若无键输入则自动进入测控状态,在测控状态,系统的工作流程如图6所示。

3.2 设置子程序

a. 设置状态用来预置各个传感器的压力与电压的变换系数,这样一旦需要更换传感器,用户可以方便地置入新的参数而不必更换软件。且设置的参数用2864芯片保存,用户一次设置后每次开机均可调用。

b. 在传感器线性度符合设计要求时,可采用设置二点定一直线方法,通常采用零点及满量程值的四分之三点进行设置,称调零和调满。

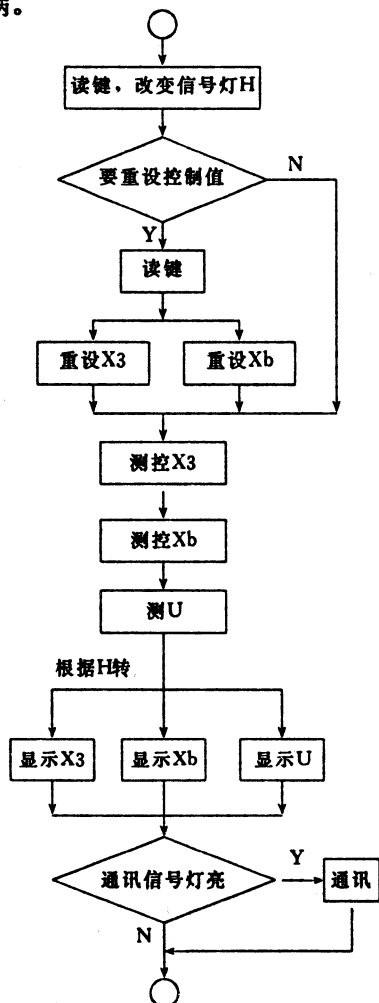


图6 测控子程序流程图

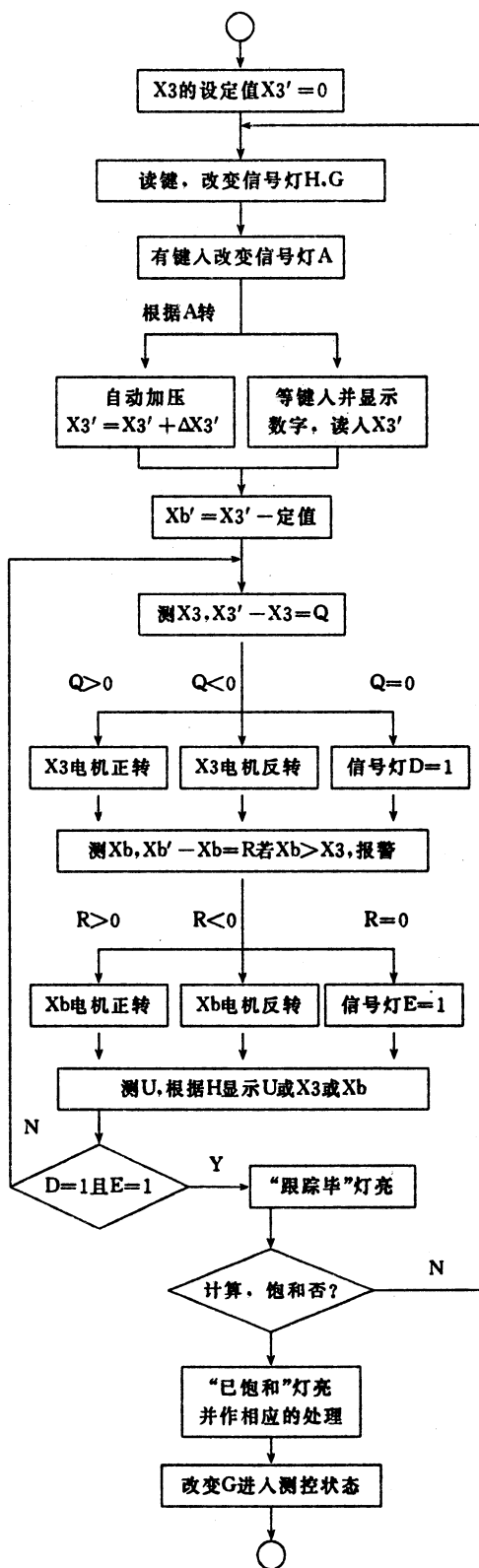


图8 饱和子程序流程图

微机实时定位控制

攀钢职教中心电气室 周富国

攀钢目前已引进了相当数量的微机,正待对老式的控制系统加以改造。定位控制是冶金、机械、化工等部门的重要控制环节。例如机械加工切割;轨梁;线材、板材、管材的生产都有着较高的尺寸精度要求,而传统的人工定尺法,误差大,生产效率低,重复加工多,材料消耗多,攀钢轨梁厂热锯目前采用的人工定尺,由于钢种、温度等因素影响,切头余量高达 50cm 左右,浪费了材料,也为后续加工带来了麻烦,严重影响着生产效率。本文就如何用微机实施自动定位控制,(热锯为例)作一初步探讨。

1 定位控制中的工艺要求及问题

产品的尺寸是严格根据用材单位的要求而定的,这就决定了钢材的计划长度,而钢坯

要加热到 1200℃左右轧制成型,整个过程温度都在不断变化,考虑到物体的热胀冷缩作用,保证足够的计划冷长度,故应留加工余量。所以实际切锯长度为:

锯切长度=计划长度+计划长度×温度系数×切时钢温

由于钢材种类不同,锯切时间不同,温度不同,钢材的膨胀量是不同的,所以就得出实时监测钢的温度,实时调整锯位,然而人工定尺是无法办到的,只能加大余量,以保证计划。

一般的大中型钢材厂家常采用多台锯同时锯切。例如攀钢轨梁厂一般是三台以上的锯同时锯。按不同的钢种、不同的温度、各台锯都应有各自的位置量和不不同的位移量,这就需要有统一的坐标原点,假定以切头端为初始原点,原三台锯的位置是 25m,50m,75m处,而根据当前的钢温要求各段进加

c. 报警极限值的设定,当系统在测控或作饱和试验时,当实测值超过满量程值 50KPa(此值可根据需要调整)时需报警,在设置时根据满量程的输入求出相应的报警极限值。设置子程序流程图如图 7 所示。

3.3 饱和子程序

饱和状态是根据对土壤试样的测试规程编程设计的。为了方便操作,在饱和试验的每一过程均有信号提示,也可以随时通过按键输入调看加压值或待测压力值。若有联机信号灯,则测试等信息均送 PC 机。饱和子程序流程如图 8 所示。

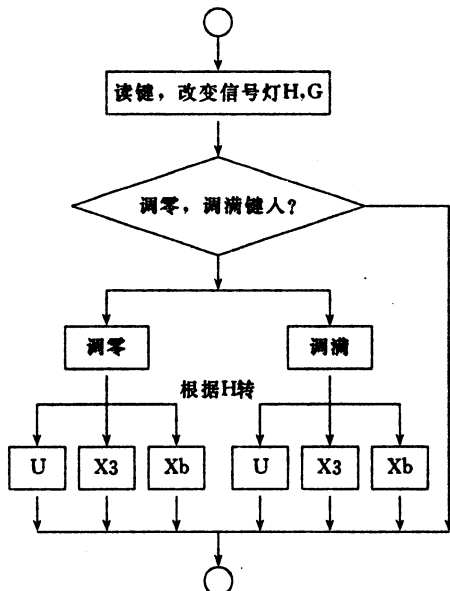


图 7 设置子程序流程图

0.3m, 则三台锯的位置应分别为: $25 + 1 \times 0.3 = 25.3\text{m}$, $50 + 2 \times 0.3 = 50.6\text{m}$, $75 + 3 \times 0.3 = 75.9\text{m}$, 各锯位移量分别为: 0.3m, 0.6m, 0.9m。

对钢温的监测采样必须实时, 要求速度快, 精度高, 否则就不能准确反映锯时钢温, 就会增加误差, 甚至出废品。在整个加工过程中, 必须随时显示系统的工作状况, 显示各锯的实际位置, 以便值班技术人员监视。由于加工轧制的钢种不同, 规格不同, 其温度系数是不同的, 这就要求系统能识别轧制的钢种和规格, 以便计算出准确的变化量。

2 微机实时定位自动控制方案

基本思想是根据红钢的温度, 用红外线测温后, 送入微机进行判断, 数据处理, 发出定量的位移指令, 去控制热锯移位。整个控制包括一个控制中心, 三条信号通道, 如图 1 所示。其中模拟输入通道完成采样功能; 数据输出完成执行驱动功能; 数据输入通道完成反馈实际位置的功能, 由单板机组成的控制中心完成处理数据、判断、及发布位移指令功能。

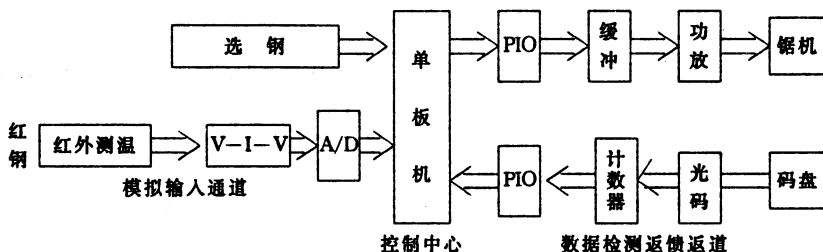


图 1

锯前, 技术人员必须将本班加工钢材型号用按钮或键盘的方式输入微机。微机识别后, 选出相应的温度系数, 计划长度等数据并存入运算准备单元。红钢到位后, 可以通过定位开关给微机一个信号, 于是微机启动模拟输入通道, 开始采样。并将采得的温度数据送入微机。为了保证精度, 可考虑采用滤波的方式来获得数据。选钢程序如图 2、3 所示。

红外线测温仪具有较高的精度和良好的可靠性, 反应速度也高, 所以能满足微机快速控制要求。由于红外线测温仪要安装在锯切现场, 而 A/D 和微机在控制室, 线路较长, 线路的干扰和损失都可能影响检测的精度, 所以在检测处加一个 V-I-V 转换器, 再经 A/D 转换器转换成数字量自动送入微机处理中心, 经滤波, 获可靠值。微机根据采样数据、原始数据、初始位置量计算出各锯的位移量, 判断前进、后退(方向)。并发出驱动指令, 通过数据输出通道去执行驱动运行。由于微

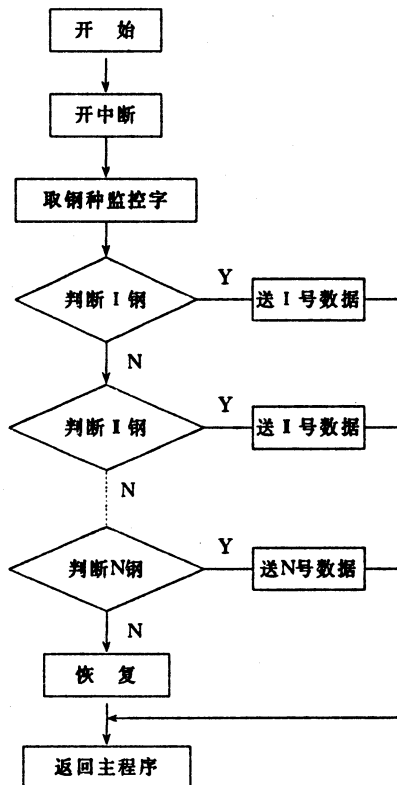


图 2 选钢程序

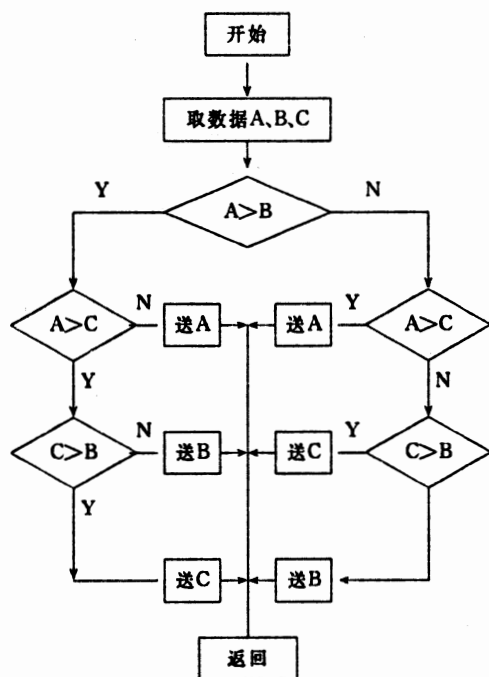


图3 采样数据滤波程序

处理机本身带的接口电路较少,所以要外扩接口 PIO 满足数据输出需要。由于红钢到位后,采样完就得及时执行,所以可考虑利用中断方式,按采样→处理→执行顺序“一体”完成。程序框图如图4所示。

若要求精度高,可将轨道传动改为链条或齿条传动,用步进电机控制。例如采用单—双六拍控制, $A \rightarrow AB \rightarrow B \rightarrow BC \rightarrow C \rightarrow CA$, 其误差不会超过一步。(如轮周长 40cm, 齿 400, 则为 $40/400=0.1\text{cm}$, 说明误差不到 0.1cm) 这就能完全满足锯切精度要求。功放部分也可以考虑采用定型的环形脉冲分配器。由微机控制中心通过软件程序控制电机的选择和每台电机的位移齿数。也可考虑由 PIO 直接输出单双六拍脉冲控制字, 经 VMOS 功放后, 驱动步进电机。驱动程序框图如图5所示。

上述开环控制虽简单,但存在误差积累问题。微机只是靠理论计算值进行控制,缺乏现场信息,所以必须设置一个反馈数字通道,用来完成对实际位置的测定。并按时报给微

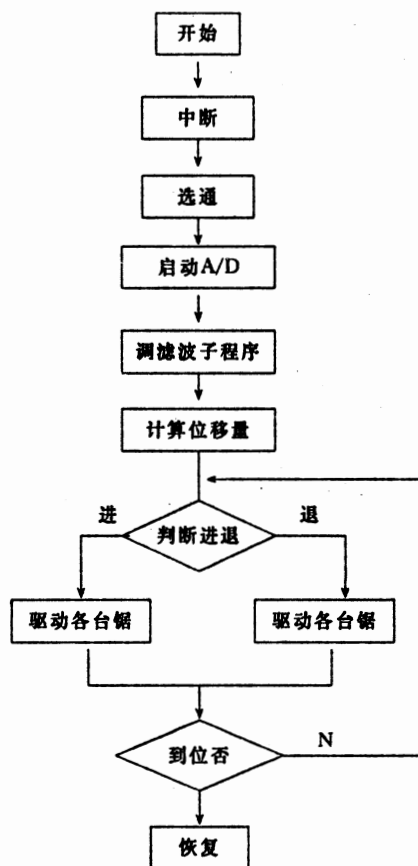


图4 处理判断程序

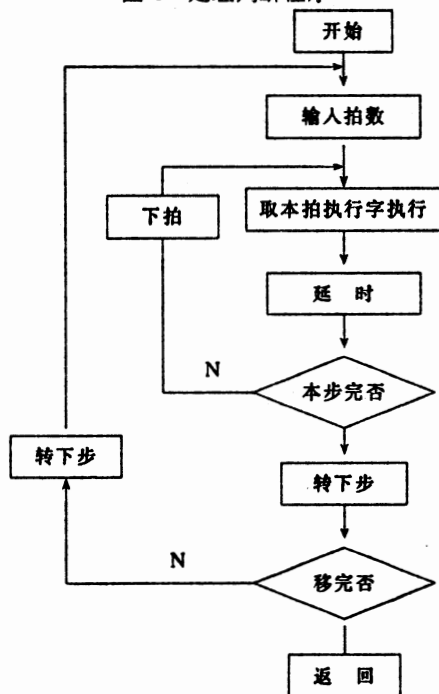


图5 步进电机驱动程序

谈微机的优化内存管理

军械工程学院 计算中心 李笑梅

摘要 优化内存管理是微机应用中的现实问题,本文从微机内存分配出发,阐述了优化内存管理的方法,并给出了优化内存管理的命令实例。

关键词 内存分配 内存管理规范 优化内存管理

1 引言

随着计算机技术的不断发展,CPU 芯片不断更新换代,由早期的 8088,发展到

80286、80386、80486,直至目前推出的 Pentium586。不同的 CPU 芯片,其寻址能力也不一样。现将 Intel 公司的主要微处理器芯片的一些技术指标列于表 1,供参考。

机中心,以校正误差。反馈检测可考虑采用与电机同步的光码盘每步一个脉冲信号送往可逆计数器。计数器由微机处理中心的“进”“退”指令,分别进行加、减计数,其反映了实际位置,可通过中断方式送给微机中心存储单元。送数程序框图如图 6 所示。

微机处理中心除了处理数据、发出控制指令外,还要负责随时显示各台锯实际位置量,以供技术人员掌握工作状况。

其显示程序如图 7 所示

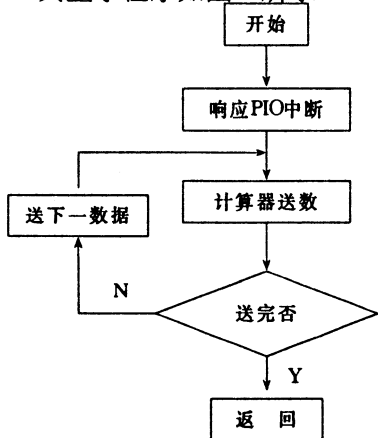


图 6 送数程序

此参考程序有往复显示功能,当中断来了(如选钢,采样执行,送数等),则微机马上

转中断执行中断程序,一旦处理完毕,就即时返回显示程序进行重复显示。

这一实时定位控制,是通过对攀钢轨梁厂现场考查后作出的,对微机改造有一定的参考价值,但由于涉及到的实际工程问题较多,在一些细节问题上,还有待探讨。

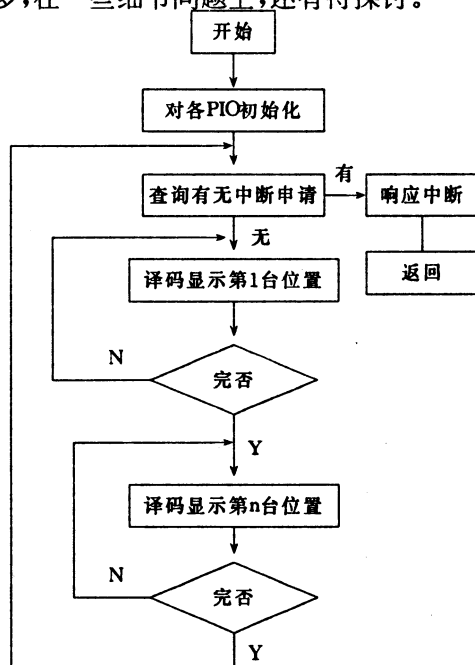


图 7 显示程序

表 1 Intel 公司的微处理器

CPU	8088	8086	80286	80386	80486	80586 Pentium
主频 MHZ	4.77/8	5/8	6/8/10	16/20/30/40	25/33/50/60/80/100	66/75/90/100/120/130
数据总线位	8	16	16	32	32	64
地址总线位	20	20	24	32	32	32
寻址能力	1M	1M	16M	4G	4G	4G

从表 1 中可以看出,由于微型机所使用的微处理器芯片不同,它的主频、数据总线位、地址总线位及寻址能力也不同,所以微型机所能管理的内存容量也随着微处理器芯片的不同而不同。早期以 8088 CPU 为对象开发的 DOS 版本,最多只能提供给用户 640KB 的内存管理,为保持 DOS 版本的向下兼容,更新换代的 CPU 继续向用户提供 640KB 的内存管理,增加的内存容量用户无法使用。高版本的 DOS 5.0 及以上版本,提供了管理高于 640KB 内存的软件,帮助用户使用这些内存空间。由于计算机的应用不断普及,应用软件的种类繁多,若不加强对内存管理,系统将经常提示“内存不够”的错误信息,使得各种应用软件不能正常运行。

2 微型机系统内存分配

DOS 系统把内存分为四类(以 DOS 6 版本为例),如图 1 所示。

a. 基本内存(Base Memory)

基本内存是系统最基本的内存也称为常规内存(Conventional Memory),从 0KB—640KB,它占用 640K 内存地址空间(由 00000H—9FFFFH)。它是 DOS 和所有程序都可以使用的内存区。

b. 保留内存(Reserved Memory)

保留内存也称上位内存(Upper Memory),位于基本内存之上的 384KB,在 A0000H—FFFFFH 内存地址空间,这个区域用于视频缓冲区(VRAM)、ROM BIOS、页

面缓冲区(页框)及其它扩充 ROM 使用。

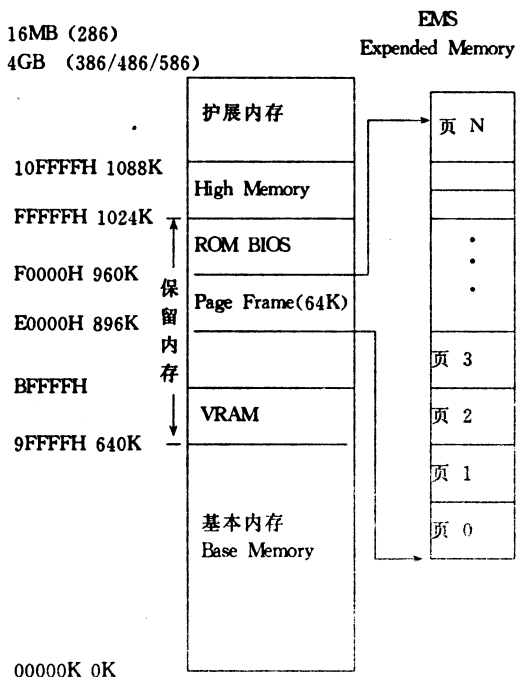


图 1 内存结构框图

其中页缓冲区专供扩充内存管理程序使用。

c. 扩展内存(Extended Memory)

扩展内存是指 286/386/486/586 系统中 1MB 以上的 RAM 存储区。因为 DOS 是在 8088/8086 微机系统中开发的,DOS 本身只能在实模式下运行,如果要存取扩展内存时,就需要调用基本输入输出系统(BIOS)中断服务程序,将实模式换成保护模式才能由 286/386/486/586 存取,故 DOS 实模式下的用户并没真正用到扩展内存,必须切换到保护模式才能进行数据存取。但是,只有扩展内存起始的 64KB,可以通过 A20 这条地址线,

在实模式方式下被寻址,这部分扩展内存称为高内存区(HMA)。

d. 扩充内存(Expended Memory)

扩充内存是一个分离的内存池,扩充内存是以 LIM/EMS 规范为标准的内存页面管理软件的支持下使用 1MB 以外的扩充内存。内存页面管理软件将扩充内存划分为 16KB 为一页面的若干页面。所以扩充内存又称页面内存。它利用保留内存中页缓冲区,其地址是 DFFFFH—EFFFFH,共 64KB 的地址空间作为地址窗口(Page Frame),通过该窗口以页面方式访问扩充内存中的各页面。其使用方法如图 1 所示。它把页缓冲区作为切换窗口,并将 64KB 的缓冲区分为四页,把每页 16KB 中的 Page 做扩充内存页面的切换窗口,每次最多可切换四页。这样,利用页面切换技术,可以使用 CPU 在实模式下去访问超过 1MB 以上的扩充内存。

目前 LIM/EMS 规范中的 V3.2 版本和 V4.0 版本的功能有所不同,V3.2 支持 8MB 扩充内存,而 V4.0 可支持 32MB 扩充内存,V3.2 只允许数据驻留在扩充内存,而 V4.0 允许数据和程序都驻留在扩展内存中运行。

3 内存管理规范

3.1 扩展内存规范 EMS(Extended Memory Specification)

扩展内存规范 EMS 是由 Lotus、Intel、Microsoft 和 AST 公司共同制定的,先后推出有 EMS2.0 和 3.0 版本。

EMS 对三个内存区进行管理:

a. UMB(Upper Memory Block);

UMB 是上位内存块,占用保留内存所提供的地址空间。

b. HMA(High Memory Area);

HMA 是高内存区,它在扩展内存起始地址的 64KB 内存区(1024KB—1088KB)。

c. EMB(Extended Memory Block);

EMB 是扩展内存块,它是高于 1088KB 的内存区域。

EMS 的管理方式,是把扩展内存看成是基本内存之外延伸出来的存储空间,其扩展内存最大容量受 CPU 的寻址能力限制。

UMB、HMA 均在实模式下访问内存,而 EMB 必须在保护模式方式下才能由 CPU 访问内存。

3.2 扩充内存管理规范 LIM/EMS

扩充内存管理规范 LIM/EMS 是由 Lotus、Intel、Microsoft 公司共同制定的,先后推出 LIM/EMS3.2、增强 3.2 和 4.0 版本。

LIM/EMS 是一种软硬件混合设计的规范。在软件方面,它将扩充内存划分成若干页面,每个页面 16KB,它利用保留内存中 64KB 页缓冲区作为扩充内存的页面切换窗口。在扩充内存管理程序 EMM 的控制下,可以任意将四页的扩充内存空间映射到页缓冲区,供应用程序访问。在硬件方面,扩充内存的 RAM 芯片都以 LIM/EMS 卡的形式插到机器中。由于扩充内存管理规范是把 1MB 地址范围以外的物理存储空间,映射到 1MB 地址之内的页框中使用,所以扩充内存容量不受 CPU 寻址能力限制。

4 内存管理程序

为了使用扩展内存和扩充内存,在 DOS5.0 以上版本的操作系统中都提供了符合 EMS、LIM/EMS 规范的内存管理程序。

4.1 扩展内存管理程序 HIMEM.SYS

HIMEM.SYS 是 Microsoft 公司根据 XMS 扩展内存规范的标准开发的。要使用扩展内存,就必须在 CONFIG.SYS 中驱动 HIMEM.SYS 程序来管理扩展内存。

该扩展内存管理程序由 MS-DOS 系统提供,在使用时将此程序安装在 CONFIG.SYS 系统配置文件中。在 CONFIG.SYS 文件中的命令形式是:

DEVICE=C:\DOS\HIMEM.SYS

4.2 扩充内存管理程序 EMM386.EXE

EMM386.EXE 可供 DOS 在保留内存区产生上位内存块(UMB),以供驱动程序和常驻程序(STR)装入。而且,EMM386.EXE 可把系统的扩展内存模拟成符合 LIM/EMS 扩充内存规范的扩充内存,因此在 EMM386.EXE 安装前,应先安装 HIMEM.SYS。

该扩充内存管理程序由 MS-DOS 系统提供,此程度与 HIMEM.SYS 程序一起安装在 CONFIG.SYS 系统配置文件中。在 CONFIG.SYS 文件中的命令格式是:

DEVICE=C:\DOS\HIMEM.SYS

DEVICE=C:\DOS\EMM386.EXE[参数]

a. 若无参数可使用 UMB,并将扩展内存模拟为扩充内存。

b. 带 NOEMS 参数,只提供 UMB 使用,不生成模拟扩充内存,其命令格式为:

DEVICE = C: \ DOS \ EMM386.EXE
ONEMS.

c. 带 RAM 参数和 FRAME 参数,其命令格式为:

DEVICE = C: \ DOS \ EMM386.EXE
RAM[FRAME=mmmm/NOEMS].

若只有参数 RAM,与无参数的功能相同,其命令格式为:

DEVICE = C: \ DOS \ EMM386.EXE
RAM.

若选择参数 FRAME,则是用来设置页面切换窗口地址。如果不设置 FRAME 参数,EMM386.EXE 自动设置 E0000H 地址为页面切换窗口的起始地址。若设置 FRAME=NOEMS,即无页面切换窗口,使 UMB 可多出 64KB 的空间。

d. 带 I 和 X 参数的格式:

DEVICE = C: \ DOS \ EMM386.EXE [I
= mmmm-nnnn] 其中 I 指包含 mmmm -
nnnn 指定的地址空间;

X 指不包含 mmmm-nnnn 指定的地址空间;

I 和 X 参数必须配合 RAM 或 NOEMS 使用,用来指定或排除某段 UMB 的地址。

例如,将 D0000H-D7FFFH 这段内存地址空间作为硬盘保护卡的 I/O 地址,此命令格式为:

DEVICE=C:\DOS\EMM386.EXE
NOEMS X=D000-D7FF

5 内存的优化管理方法

内存的优化管理方法就是对系统配置程序进行优化管理,充分使用硬件资源,给用户提供足够大的基本内存和访问 640KB 以外的内存空间,从而提高软件的运行效率。其优化方法如下:

5.1 安装 HIMEM.SYS 扩展内存管理程序,访问高端内存,使 DOS 的核心部分驻留在 HMA 内存区,从而提供给用户较大的基本内存空间。

命令格式是在 CONFIG.SYS 中安装如下命令:

DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH

5.2 将设备驱动程序安装在 UMB 中,减少设备驱动程序占用的基本内存,从而增大用户可以使用的基本内存空间,提高用户程序的运行速度和效率。

命令格式是在 CONFIG.SYS 中安装如下命令:

DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE
DOS=UMB
DEVICE=C:\DOS\ANSI.SYS
DEVICE=C:\DOS\MOUSE.SYS

此例是将 ANSI.SYS 和 MOUSE.SYS 两个设备驱动程序安装在 UMB 中。

5.3 将常驻内存程序(STR)安装到 UMB 中,该方法的目的也是为用户提供足够的基

本内存空间,从而解决运行应用程序时,系统报告“内存不够”的问题。

命令格式是在 CONFIG.SYS 中安装如下命令:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE
DOS=UMB
LOADHIGH C:\DOS\DOSKEY.COM
LOADHIGH C:\DOS\PRINT.COM
```

此例是将 DOSKEY.COM 和 PRINT.COM 两个常驻内存程序安装在 UMB 中。

5.4 综合优化内存

其方法是腾空基本内存和充分利用 1MB 以上的内存空间,下面实例仅供参考:

在 CONFIG.SYS 文件中设置的命令是:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE
DOS=HIGH,UMB
DEVICEHIGH=C:\DOS\ANSI.SYS
LOADHIGH C:\DOS\DOSKEY.COM
LOADHIGH C:\UCDOS\MOUSE.COM
FILES=20
BUFFERS=20
```

利用上例系统配置文件启动某微机后,用 MS-DOS 6.22 提供的查看内存配置与使用情况的命令 MEM.COM,来检查剩余的基本内存、扩展内存及保留内存的使用情况时,显示如下:

Memory Type	Total =	Used +	Free
Base	640K	31K	610K
Upper	0K	0K	0K
Reserved	384K	384K	0K
Extended(XMS) *	7,168K	480	6,688K
Total memory	8,192K	895K	7,298K
Total under 1 MB	640K	31K	610K

Total Expanded(EMS) 7,488K(7,667,712 bytes)
Free Expanded(EMS) * 6,928K(7,094,272 bytes)

* EMM386 is using XMS memory to simulate EMS memory as needed.

Free EMS memory may change as free XMS memory changes.

Largest executable program size 609K(624,112 bytes)

Largest free upper memory block 0K(0 bytes)

MS-DOS is resident in the high memory area.

如果在系统配置文件 CONFIG.SYS 中没有调用内存管理程序,也没将 DOS 的基本内核装入到高内存区,却把设备驱动程序及常驻内存程序装入了基本内存,其 CONFIG.SYS 的内容如下:

```
DEVIDE=C:\DOS\ANSI.SYS
DEVIDE=C:\DOS\DOSKEY.COM
DEVIDE=C:\UCDOS\MOUSE.COM
FILES=20
BUFFERS=20
```

此时,当用 MEM.COM 检查系统内存使用的情况时,显示如下:

Memory Type	Total =	Used +	Free
Base	640K	64K	576K
Upper	0K	0K	0K
Reserved	384K	384K	0K
Extended	7,168K	7,168K	0K
Total memory	8,192K	7,616K	576K
Total under 1 MB	640K	64K	576K

Largest executable program size 576K(589,792 bytes)

Largest free upper memory block 0K(0 bytes)

从这两个显示结果可以看出,使用采用优化内存管理的 CONFIG.SYS 文件启动微机时,它的剩余的基本内存(常规内存)为 610KB,扩展内存为 6,688KB;而没使用采用优化内存管理的 CONFIG.SYS 文件启动微机时,它的剩余的基本内存为 576KB,扩展内存为 0KB。因此,只有采用优化内存管理方法才能为用户提供足够大的基本内存、1MB 以上的扩展内存和扩充内存空间。

简单易行通用安装盘的制作

华中理工大学自控系 孙国锋 叶念渝

摘要 本文旨在介绍用 Visual Basic 3.0 Setup ToolKit 制作安装盘的基本思路,一般的操作步骤。在此基础上引进了 Windows 的 Key-Value 型文本文件作为安装“参考文件”,使其通用性和可维护性大大提高,成为一个通用的安装程序。

关键词 安装程序 Key-Value 型文件 Visual Basic 3.0 for Windows

作为一个发售的应用软件,安装程序是必不可少的一部分。目前,绝大多数开发工具配备了制作安装盘的“制作向导”,使用方便,但这样制作的安装盘界面提示均为英文,且不能按用户自己的意图安排界面。笔者在实践中,利用 Visual Basic 3.0 for Windows Setup ToolKit 的基本模块,加上自己的扩充,制作了一个全中文提示,界面形象友好,可“一程多用”的通用安装程序,并实际应用于湖北省电力局的文档一体化系统,程序分售各地,安装运行良好。

该安装程序由基本功能和扩展功能两部分构成。Visual Basic 3.0 的 Setup ToolKit 提供了安装程序的基本功能模块与逻辑框架,用户根据需要作适当调整即可实现一般安装程序的功能;扩展功能则要求用户灵活应用 ToolKit 的基本功能模块并熟练掌握 Visual Basic 语言。

1 安装程序的基本构成

基本部分的核心程序有四个,Compress.exe, Setup.exe, Setup.lst 和 MySetApp.exe。其中 Compress.exe 是一个压缩工具程序,MS-Windows 提供了与 Compress.exe 相对应的解压缩函数,所以微软公司大多数发售软件均使用此压缩工具。其使用格式为:Compress-r MySetApp.exe, MySetApp.exe 是用户要压缩的文件,-r 的作用是使 MySetApp.exe 压缩后自动成为 MySetApp.ex_ 格

式,与微软公司常用安装压缩格式一致。Setup.exe 是 toolkit 提供的一个可执行程序,用户不可见其原代码,其作用有两个:一是自动检测安装原盘的盘符,如 A 或 B,并自动将此参数传给 MySetApp.exe;二是为用户的 MySetApp.exe 运行准备条件,一般来说是将 MySetApp.exe 运行所需的动态库(DLL)拷贝到 Windows System 所在的目录下,使 MySetApp.exe 运行时可动态连编,如果采用 Visual Basic 3.0 for Windows 编制程序则需把 VBRUN300.DLL 拷贝到 Windows System 目录下面。Setup.lst 是一个文本文件,它为 Setup.exe 提供必要的信息,Setup.exe 则用这些信息做初始化工作。Setup.lst 的内容通常如下:

```
MySetApp.ex_
VBRUN300.DLL
... ..
```

文件中填写一系列解压缩前的文件名,这些文件是用 Compress.exe 压缩得到的,存储在用户安装盘的第一号盘上。Setup.exe 读取这些文件并将它们压缩到 Windows System 目录下,供用户应用程序 MySetApp.exe 使用。当然用户可以推广这种用法把程序中用到的一些动态库。OLE 控件等文件以此种方式解压缩到 Windows System 目录下面,具体到本文使用的开发工具 Visual Basic 而言,则需要将用到的所有 VBX 型文件作此处理。MySetApp.exe 是用户自己编写的安装

程序的主干部分,程序的一切基本功能均在这里实现。

a. 丰富界面。在 ToolKit 提供的原代码的基础上按中国人的习惯调整界面的外观,修改字符串常数实现全汉化提示,也可适当加入位图文件,动画功能使人机界面更加友好。

b. 灵活应用基本功能。ToolKit 提供了许多安装需要的基本功能,比如自动检测 Windows 所在的目录,检测安装硬盘的大小,检测输入盘符是否合法,解压缩子过程等。这些基本功能都由 ToolKit 提供了原形函数,用户可以直接利用这些函数并按自己的要求加以组合。

c. 调整整体功能。在集成环境下编译调试,直到满足安装的一般功能后生成 EXE 可执行文件。

d. 填写 Setup.lst 文件。按上述方法填写 Setup.lst 文件,并将相应文件存储到安装盘第一号盘上。

e. 制作磁盘。用 compress.exe 将要压缩的文件压缩,并依次存于安装盘上

2 安装程序通用性的实现

通过以上步骤可以制作出一个能满足最基本要求的安装程序。在此程序中解压一个文件需要使用一次 COPYFILE 函数,也就是

```
DWORD GetPrivateprofileString(LPCTSTR lpAppName, //points to section name
```

```
LPCTSTR lpKeyName, //points to key name
```

```
LPCTSTR lpDefault, //points to default string
```

```
LPTSTR lpReturnedString, //points to destination buffer
```

```
DWORD Nsize, //size of destination buffer
```

```
LPCTSTR lpFileName //point to initialization
```

```
filename)
```

在程序中利用此过程从参考文件中获取安装的相关信息,作为参数传给 COPYFILE 函数,从而把要安装的文件维护起来。当然,安装的其它信息也可以用这种方式维护起来,比如软件开发单位名称,安装后生成的图

说要在程序中增加一条相应的语句,一方面程序变得冗长,再一方面程序无通用性,用户要增减安装文件就必须修改原代码。笔者为此原程序作了一个简易的扩展使其有了通用性,用户只需填写一个文本文件即可维护安装程序,本文中采用 Windows 的 Key-Value 型文件作为一种“参考文件”,命名为 Sptup.ini,文件的格式需要用户根据要求自己编排。比如要反应某号安装盘的信息可按如下的格式书写“参考文件”。

```
[disk1]
```

```
files=num
```

```
sfile1=解压缩前文件名 1. ## _
```

```
dpath1=目标目录 1
```

```
dfile1=解压缩后文件名 1. ## _
```

```
sfile2=解压缩前文件名 2. ## _
```

```
dpath2=目标目录 2
```

```
dfile2=解压缩后文件 2. ## _
```

```
.....
```

其中,[disk1]表明这是关于第 1 号安装盘的信息,files=num 指明本号盘中共有多少文件,以下的内容则是依次对每个文件的信息作说明。读取这些信息必须设计一个专用的子程序,过程中用到的核心函数是 Windows 的 API 函数 GetPrivateprofileString,其调用参数为:

标,文字说明,安装画面也可维护。只要用户根据需要以此思路制定出自己的“参考文件”,便可制作一个可灵活应用于不同安装目的的通用安装程序。

送你一把键盘锁

为了实现对微机的权限控制和保密目的,往往采取一些限制和防范措施,例如:在 CMOS 中设置密码,或用有关软件编一个小小的驱动器“锁”,用 Windows 的可在屏幕保护程序里设置口令,等等。这里介绍使用 DOS 用于批处理的选择命令 CHOICE 建一个简单的键盘锁,可谓方便实用。命令格式如下:

CHOICE[/C 字符][/N][/S][提示信息]

参数说明:

/C 字符 定义用作密码的字符。可以定义一个或多个字符。如果定义多个,则开锁时,只要按这字符集合中的任一个字符的对应键,即生效。最好定义上档键字符或大写字母然后加/S 参数,以增加保密性。

/N 禁止显示键入的密码字符,实际上是一个取消应答的开关。若缺省本开关,开锁时,屏幕会出现密码字符+?,让你确认,当再次按密码字符的对应键即可开锁。

/S 区分字符的大小写。

提示信息 键入要显示的如“请输入密码”之类的提示信息。

因 CHOICE 命令只能用于批处理程序,所以,要用此命令设置键盘锁,必须建立批处理文件。下面举一个取名为 JPS. BAT 的键盘锁批处理:

```
@ECHO OFF
```

```
C:\DOS\CHOICE /C> /N
```

```
CLS
```

说明:“>”就是密码字符,请注意:“/C”和“>”之间不能有空格。CLS 清屏语句的作用是,开锁时键入密码后迅速清屏,使旁人无法看清密码。

使用时,键入文件名 JPS 执行这个批处理,就可达到锁住键盘的目的。按 SHIFT 和“>”键,即可开锁。

可以推而广之,如果你不希望别人访问某个软件,可以在运行这个软件的批处理文件中加进这个键盘锁的命令,即可限制对该软件的访问权。如果把它加入自动批处理中,那么,开机自检后须开锁才会出现提示符,这样,这个键盘锁就可充当开机用密码了。当然,这个键盘锁,只能是对“东游西撞”、“乱砍滥伐”的初学者而设的小障碍。

广西田阳县教师进修学校
罗加导