

使 FoxPro 具有图形功能的一种方法

陶明璋 (杭州商学院计算机系 310006)

金龙法 (浙江省环保监测站)

摘要: 本文介绍了一种利用 FoxPro 的外部例行程序 API 编写 C 程序的方法, 为 FoxPro 增加图形功能, 与目前所采取的一些方法进行了比较, 叙述了用 MicroSoft C++ 7.0 编写 API 程序的方法。

一、前言

FoxPro 作为数据库处理数据其功能已相当完善, 这也是它使用面广的主要原因。但还有点小小的缺点就是无法显示图形。目前在有关计算机技术刊物上, 经常刊登一些为 FoxPro 等数据库系统添加图形功能的方法, 归纳起来有以下几种: (1) 利用象 UC DOS 等汉字系统的特显功能完成; (2) 利用 C 等具有图形处理功能的程序设计语言完成。前一种的效果倒是不错, 但移植性太差, 离开相应的汉字系统就无法运行。后一种往往作为外部程序需要 RUN 等运行, 内存消耗比较大, 并且数据之间的传递比较困难。现在笔者尝试了利用 API 功能的方法觉得不错, 跟大家讨论。

二、FoxPro API 功能简述

在 FoxPro 中, API 允许你用 C 语言编写例行子程序, 并且可用 Microsoft C/C++ 版本 7 来进行编译。可以在运行的时候转入 C 库函数并调用子函数就象它们是 FoxPro 本身的一部分。FoxPro 的 API 是一个 C 语言写成的库函数, 它允许利用 FoxPro 的数据库、事件处理器以及用户界面。使用这些功能能编写一个外部例行程序来操作设备, 读写外部文件格式, 或直接调用操作系统程序。这使你能高效地利用 C 语言编程, 它拥有 C 语言的快速和灵活性以及 FoxPro 所有的强大功能。为 FoxPro 外部例行程序 API 编写 C 程序, 必须有 Microsoft FoxPro 库结构工具箱(LCK)。工具箱包含库函数, 目标文件, 标题, 形成文件, 批文件, 用于生成你的子程序。另外, 还必须由 Watcom Group, Microsoft C/C++ 版本 7, 或 Microsoft Visual C++。本文中出现的示例子程序由 Microsoft C/C++ 版本 7 编译和测试。在例子中笔者使用了小存储模式, MS-DOS 下的 FoxPro 可用任何存储器模式。

1. FoxPro API 库的构成

对于 C 例行函数的用户, 一个 FoxPro API 库软件组

包含以下几个部分:

- FoxPro API 包含文件 (PRO-EXT.H)
- 所编写的 C 图形子函数说明
- FoxInfo 结构
- FoxTable 结构

2. API 包含文件 PRO-EXT.H

FoxPro 库结构工具箱含有 PRO-EXT.H 文件, 它包含函数模式和不断的 FoxPro API 的定义。在所编写的 C 语言例行程序中必须包含 (#include) 这个文件, 为编译程序定义 API 调用。

3. FoxInfo 结构

一个 API 库应包含一个“FoxInfo”结构, 它告诉 FoxPro 你的函数名, 带有多少参数, 要求哪种类型的参数, 这结构允许 FoxPro 检测到对函数的错误调用。FoxInfo 结构被定义在 API 首部文件 PRO-EXT.H。如下:

```
typedef struct {
    char FAR * funcName; /* 函数名(要求全部大写) */
    PPFi function; /* 函数的正确地址 */
    short parmCount; /* 参数个数或标志值 */
    char FAR * parmTypes; /* 参数类型列表 */
};
```

参数类型如下:

- C 字符(串)型
- D 日期型
- I 整型
- L 逻辑型
- N 数字型: IEEE 双格式
- R 可变或指定数组类型
- ? 允许各种类型

在文后所附的例子中作如下描述:

```
FoxInfo grFoxInfo[] = {
    {"GRSET", (PFfi)grset, 0, ""},
    {"grset", (PFfi)grset, 0, ""},
};
```

```

} "grset", (FPFI)grset, 0, ""},
} "grset", (FPFI)grset, 0, ""},
} "GRCLOSE", (FPFI)grclose, 0, ""}
};
    
```

4. FoxTable 结构

该结构向 FoxPro 报告建立你的库所需要的内存, 并且定义每个函数的地址, FoxTable 结构在 PRO-EXT.H 中被定义如下:

```

typedef struct -FoxTable{
    struct -FoxTable FAR hexLibrary;
                                /* 库的连接表 */
    short infoCount;           /* 库中的函数数目 */
    FoxInfo FAR * infoPtr;     /* 函数表 */
} FoxTable;
    在文后所附的例子中作如下描述:
    FoxTable -Foxtable = {
        (FoxTable Far *) 0, sizeof (grFoxInfo)/sizeof
        (FoxInfo), grFoxInfo};
    
```

三、C 的图形处理子函数组

(1) void far grset(ParamBlk FAR * parm)

该函数用在最前, 完成做图前的初始化工作。

(2) void far grclose(ParamBlk FAR * parm)

该函数用在最后, 完成做图后的关闭工作。

(3) void far coord(ParamBlk FAR * parm)

根据数据的个数及数据的正负情况, 画出相应的坐标系。

这里涉及到解决 0 坐标线及 X、Y 轴的刻度问题。

(4) void far bar2d(ParamBlk FAR * parm)

根据数据的次序及大小画出对应的直方图。这里涉及到根据数据的个数及次序定位直方图的坐标。

(5) void far lline(ParamBlk FAR * parm)

根据数据的次序及大小画出对应的直方图。这里涉及到根据数据的个数及次序定位折线图的坐标。

(6) void far grtitle(ParamBlk FAR * parm)

显示对应的图形标题。

四、从 FoxPro 中调用图形处理库例行程序

把文中所附例程经 Microsoft C++ 7.0 编译和 FoxPro LCK API 例程连接(Link)后生成 grProcess.PLB, 用 FoxPro 命令 SET LIBRARY TO TuProcess.PLB 来激活它。一旦库函数被激活, 用户可以调用库函数, 就好像它们是 FoxPro 的内在本身的部分函数。请看下面小程序:

```

*****
* 程序名: grtest.prg.
* 功能: 利用数据库中的数据产生图形
* 调用: 图形处理库函数 grProcess.PLB
* 日期: 1996.04.10
*****
set talk off
clea
set libr to tul
= grset()
= coord(300,5)
* 假如只有 5 个数据, 坐标点分别为
* (200), (150), (350),
* (250), (400)
= line(5,2,200,150)
= line(5,3,150,350)
= line(5,4,350,250)
= line(5,5,250,300)
= grtitle("FoxPro 例子图一")
= inkey(0)
= grclose()
= grset()
= coord(300,5)
* 假如只有 5 个数据, 坐标点分别为
* (200), (150), (350), * (250), (400)
= bar2d(5,1,200,300)
= bar2d(5,2,150,300)
= bar2d(5,3,350,300)
= bar2d(5,4,250,300)
= bar2d(5,5,100,300)
= grtitle("FoxPro 例子图二")
= inkey(0)
= grclose()
set libr to
clea
set talk on
    
```

五、小结

通过上面的叙述, 大家可以看到, 应用这样的方法使 FoxPro 具有图形功能, 使用起来很方便, 而且不存在数据传递问题。所举的例子非常简单, 只是为了说明问题, 读者在实际操作运行过程中可以加以完善。