

用 C 构造 FOXPRO 函数

陈建平 (上海财经大学 计算中心)

1. 引言

FOXPRO 提供了很好的开发环境和高级的数据库管理工具,但是对硬件的访问,如 A/D 转换器、I/O 口及串行通讯口等,未提供相应的功能和函数,FOXPRO 的应用程序设计界面 API(Application programming interface)允许使用 C 语言和汇编语言去完成一定的功能,即用 C 构造 FOXPRO 函数,用 API 还可增加图形、加密和网络等功能。

2. FOXPRO API 结构

FOXPRO 与 C 是通过传值和传地址方式二种类型传递参数数值结构。

```
typedef struct {
    char ev-type;
    char ev-padding;
    short ev-width;
    unsigned short ev-length;
    long ev-long;
    double ev-real;
    MHANDLE ev-handle;
}
```

地址结构:

```
typedef struct {
    char l-type;
    short l-where;
    l-NTI,
    l-offset,
    l-sub1, l-sub2;
} Locator;
```

FoxInfo 结构:

FoxInfo 是连结 FoxPro 和 C 的桥梁,它包含以下信息:

- 被 FOXPRO 调用的函数名
- C 函数名
- 参数的个数和类型

3. 开发准备

必须具备 Microsoft 的库构造工具集 LCK (Library Construction Kit)

Microsoft C/C++ 7.0 或 Watcom C 9.0

4. 举例:构造软盘检测函数 CHKDISK

本例主要完成对软驱的检测。因为在开发 FOXPRO 应用程序时,经常要做系统备份,但如果软驱中无盘或软盘写保护或未被格式化,则系统会出现:

Not ready reading drive A

Abort, Retry, Fail ?

这样会破坏 FOXPRO 应用程序的界面,本函数主要完成对软驱的检测。

源程序 chkdisk.c:

```
#include <direct.h>
#include <dos.h>
#include <bios.h>
#include <c:\foxlck\pro-ext.h>

void --far hhandler( unsigned devert, unsigned doserr, unsigned --
far *hdr );
FAR chkdisk(ParamBlk FAR *parm)
{
    /* Install our hard error handler. */
    -harderr( hhandler );
    if( -mkdir( "a:\ \testdrive" ) )
    {
        -PutStr( "\
nFailed" );
    }
    else
    {
        -rmdir( "a:testdrive" );
    }
}

void --far hhandler( unsigned devert, unsigned doserr, unsigned --
far *hdr )
{
    -RetInt( devert, 10 );
    -hardretn( doserr );
}

FoxInfo myFoxInfo[] = {
```

```

    { "CHKDISKA", (FPFI) chkdiska, 0, "" },
};

FoxTable -FoxTable = {
    (FoxTable FAR * ) 0,
    sizeof( myFoxInfo ) / sizeof( FoxInfo ),
    myFoxInfo
};

工程文件 dosc7.mak:

#
# Sample MS C/C++ 7 makefile for building a DOS API from
a single C
# module. Also works with Microsoft's Visual C++ . Usage;
#
# >nmake PLBNAME= <source without extension> MODEL
= <model> /F dosc7.mak
#
#
C7DIR = c:\c700 # where Microsoft C7 is installed
FOXDIR = c:\foxlck # location of FoxPro API
CC = cl
#
# Uppercase MODEL because cl options are case sensitive
#
! IF "$ (MODEL)" == "s"
MM = S
! ELSE IF "$ (MODEL)" == "m"
MM = M
! ELSE IF "$ (MODEL)" == "l"
MM = L
! ELSE IF "$ (MODEL)" == "" # default to large if memo-
ry model not spec'd
MM = L
! ELSE
MM = $ (MODEL)
! ENDIF
CFLAGS = /A $ (MM) w /Zp /GW /I $ (FOXDIR) /I
$ (C7DIR) \include /Fs $ (PLBNAME).lst

LINKER = $ (C7DIR) \bin \link
# LFLAGS = /ONERROR:NOEXE /NOF
LFLAGS = /ONERROR:NOEXE /NOE /NONULLS

```

```

OBJS = $ (FOXDIR) \api-m $ (MM).obj $ (PLBNAME).
obj
LIBS = $ (FOXDIR) \api-m $ (MM).lib
MAPFILE = NUL
DEPENDS = $ (PLBNAME).obj

all: $ (PLBNAME).plb

$ (PLBNAME).obj : $ (PLBNAME).C $ (FOXDIR) \ pro-
ext.h
    set PATH = $ (C7DIR) \ bin; % PATH%
    @ $ (CC) /c $ (CFLAGS) /Fo $ (PLBNAME). obj
$ (PLBNAME).C
$ (PLBNAME).plb : $ (DEPENDS)
    set PATH = $ (C7DIR) \ bin; % PATH%
    set LIB = $ (C7DIR) \ lib
    echo > NUL @<< $ (PLBNAME).lrf
        $ (OBJS; = + ^
    )
    $ @
    $ (MAPFILE)
    $ (LIBS; = + ^
    )
    $ (LFLAGS)
<<
    $ (LINKER) @ $ (PLBNAME).lrf
    设 C/C++ 目录为 C700
    LCK 目录为 FOXLCK
    用下列命令生成库:

    NMAKE PLBNAME = CHKDISK MODEL = S /F DOSC7.
    MAK

```

5. 应用

在 FOXPRO 环境下:

set libr to chkdisk

x = chkdisk()

x 得到函数的返回码

.t. 正确

6656 软驱门未关好或软盘未被格式化

7424 软盘写保护