

FOXBASE 和 FOXPRO 与汇编语言的参数传递

朱锦堃

(中山大学物理学系, 广州 510275)

摘 要 研究在 FOXBASE 或 FOXPRO 环境下, 把参数传递到汇编语言, 并以制造精美的立体界面为例, 说明参数传递的基本原理, CPU 寄存器在这个传递过程中的作用, 及扩展程序的编写方法.

关键词 参数传递, BX 寄存器, 汇编语言接口, 立体界面

分类号 TP311.52

随着电脑软件的发展, WINDOWS, WINBASE 等在图形界面上都有重大的突破, 特别是画面的凹凸形结构设计, 使人们产生立体和活动的感觉, 很受广大用户欢迎. 令人遗憾的是 FOXBASE 没有这个功能. 但这两个软件都留有与汇编语言的接口, 只要把 FOXBASE 的参数传递到汇编语言, 由汇编语言完成 FOXBASE 所不能完成的功能.

1 扩展汇编语言接口的基本原理⁽¹⁾

FOXBASE 和 FOXPRO 有下面 3 条命令是开辟汇编语言接口的.

LOAD<文件名>

CALL<文件名> WITH“<参数表>”

RELEASE MODULE<文件名>

1.1 LOAD 命令

该命令是将二进制程序文件从磁盘读入内存, 在内存中间同时最多装入 16 个二进制程序文件, 每个二进制程序文件最大 32KB.

二进制程序文件扩展名默认为 .BIN. 如果 LOAD 命令读入的文件除扩展名外和前次读入的二进制程序文件同名的话, 新读入的二进制程序文件将覆盖前次读入的文件.

下面以汇编语言程序为例, 说明生成二进制程序文件的步骤:

(1) 使用 MASM 对汇编语言源程序进行编译, 生成 OBJ 为扩展名的目标代码文件.

(2) 使用 LINK 对目标代码文件进行连接, 生成以 EXE 为扩展名的可执行文件.

收稿日期: 1995-06-20 朱锦堃, 男, 55 岁, 副教授

(3) 使用 EXE2BIN 将可执行文件转换为 BIN 扩展名的二进制程序文件。

在生成二进制程序文件应符合如下要求:

(1) 在偏移量 0000H 处, 应存放可执行指令。

(2) LOAD 根据二进制程序文件的大小决定分配内存容量, 因此二进制程序文件执行期间占用的内存不能超过本身的文件长度。

(3) 二进制程序文件执行期间不能破坏 FOXBASE 建立的内存变量。

(4) 控制返回 FOXBASE 之前, 应恢复代码段(CS)寄存器和栈段(SS)寄存器。

(5) 当汇编语言程序遇到 RET 时, 即返回 FOXBASE。

(6) 编写汇编语言程序时, 不能设有堆栈。

1.2 CALL 命令

该命令是在 FOXBASE 下运行已装入到内存的二进制程序文件, 用 WITH 短语与二进制文件进行参数传递, 可以使用字符型表达式和任意类型的内容变量交换数据。一旦执行 CALL 命令, 代码段(CS)寄存器则指向二进制程序的起始地址处。数据段(DS)寄存器和 BX 寄存器用于存放由 WITH 短语传递数据的第 1 个字节的地址。如果没有指定 WITH 短语, BX 寄存器的值为 0。

1.3 RELEASE MODULE 命令

该命令用于释放由 LOAD 命令装入到内存所占用的空间。

1.4 原理性举例^[2]

本例是把 FOXBASE 的 7 个参数(ASCII 码)传送到汇编语言, 并把第 1 个参数作加 3 处理, 第 2 个参数作加 5 处理, 其余不作处理, 看它们返回到 FOXBASE 的参数是什么结果, 便可知道数据传递的关系。

汇编语言的源程序: 文件名为 WB. ASM

1. PROG SEGMENT BYTE	7. WOV [SI], AL	12. MOV [DI], AL
2. ASSUME CS:PROG	8. INC BX	13. RET
3. GETPATH PROC FAR	9. MOV AX, [BX]	14. GETPATH ENDP
4. MOV AX, [BX]	10. MOV DI, BX	15. PROG ENDS
5. MOV SI, BX	11. ADD AL, 5	16. END
6. ADD AL, 3		

程序中的第 4 行 BX 是 FOXBASE 的第 1 个参数存放的地址, 本语句的意义是把 FOXBASE 送来的第 1 个参数值放入 AX 中。第 5 行是把返回 FOXBASE 的第 1 个参数地址放在 SI 中, 第 8 行是指向第 2 个参数, 第 10 行是把返回 FOXBASE 的第 2 个参数地址放在 DI 中, 也可以放在 SI 的下一个地址中, 第 13 行当程序运行到 RET 时, 则把 SI 和 DI 所指向的地址内容的第 1, 2 个参数返回 FOXBASE。

FOXBASE 的源程序

1. LOAD WB. BIN	4. ?M
2. M = "1234567"	5. RELEASE MODULE WBZ
3. CALL WB WITH M	

程序中的第 1 行是把汇编源程序生成文件名为 WB. BIN 并装入内存, 第 2 行是从 FOXBASE 带入汇编的 7 个参数, 最左边为第 1 个, 第 3 行是进入汇编语言执行 WB 的二进制程序文件, 由于有 WITH 短语, 所以有参数传递关系, 第 4 行是从汇编返回到 FOXBASE 的参数, 其结果是“4734567”, 其中第一个参数是 FOXBASE 带入为 1 经汇编加 3 处理后返回为 4, 第 2 个参数从 FOXBASE 带入为 2, 经汇编加 5 处理后返回为 7. 其余没有处理则保持原样. 第 5 行为释放 WB 文件所占的内存. 如果该文件需要重复使用, 那么第 5 行可以删去.

2 立体框的设计

FOXBASE 和 FOXPRO 是没有立体框的功能的, 为了使 FOXBASE 能达到 WINDOWS 的立体界面, 作者设计了两个应用程序, 命名为 ZHU-1. BIN 和 ZHU-2. BIN. 当你需要画立体框时, 通过 LOAD 命令把它调入内存, 再配合 FOXBASE 程序的参数传递, 便可生成精美的立体界面.

2.1 ZHU-1. BIN 的立体框制作^[3]

制作立体框的基本原理是在一个色框的基础上, 再加上直角形的亮条和直角形的黑条, 再配合底框的颜色, 便可得到醒目的立体框效果. 色框是由 FOXBASE 程序完成的, 而亮条和黑条是由汇编语言完成的, 它是从色框的行列参数传递到汇编语言, 再调用中断完成划白色线条和黑色线条, 白色线条有如光线射入, 黑色线条有如阴影部分.

ZHU-1. BIN 文件内有 A ~ R 共 18 种不同图形选择, 可制成各种不同形式的主体框, 其中 R 是扩大阴影部分, 有效增强立体感, 18 种的图形选择, 在调用 CALL 命令时应注明是哪一个参数.

(1) 简单立体框设计. 俗想在萤屏建立一个凸框, 框的对角线行列坐标为 (3, 5) 和 (22, 75), 衬底颜色为绿色, 色框为淡蓝色. 下面是生成这个效果的应用程序.

```

1. LOAD ZHU-1. BIN           5. @ 3, 5 CLEAR T 22, 75
2. SET COLO TO N/G          6. CALL ZHU-1 WITH "03052275J"
3. CLEAR                    7. CALL ZHU-1 WITH "03052275R"
4. SET COLO TO N/BG

```

程序中的第 1 行是装入划立体框的二进制程序文件, 第 5 行是在行列坐标 (3, 5)(22, 75) 位置上划出一个色框, 第 6 行是划出一个凸形框, 其中数字值为传递到汇编语言的行列参数值, J 为凸框, 第 7 行的 R 为扩大阴影部分. 如果你不需要扩大阴影部分, 第 7 行可以省去. 当你想划凹形框, 第 6 行可以选择参数 I.

(2) 多层立体框设计. 应用上述程序, 修改不同的行列坐标, 就可以得到多层的立体框. 各个小框可分布在大框的各个部位, 采用不同的凹凸形设计, 其效果令人满意.

(3) 按钮设计. 按钮的设计与立体框的方法完全相同, 只是行列坐标的尺寸较小, 从视觉上产生一个按钮的感觉. 它们又分为两类, 一类是凹形按钮, 另一类是凸形按钮. 如果使用程序调节凹凸之间变化时间为 0.2 ~ 0.3 s, 就会得到活动按钮的效果.

2.2 ZHU-2. BIN 立体框的制作^[3]

ZHU-2. BIN 所生成的立体框要比 ZHU-1 版本要高, 再加上在 FOXBASE 或 FOXPRO 方面充分发挥编程技巧, 可以使立体边框的宽度和颜色可调, 而且亮边与黑边交角处形成 45° 角增强型立体感, 使整个界面的设计提高了几个档次.

(1) 首先解剖 ZHU-2. BIN 的源程序 ZHU-2. ASM

1. * ZHU-2. ASM	18. POP AX
2. . CODE	19. INC DX
3. . MODEL SMALL	20. CMP DX, BX
4. CODE-SEG SEGMENT	21. JBE VER
5. ASSUME CS:CODE — SEG, DS:CODE — SEG	22. RET
6. ORG 100H	23. HOR:MOV AH, OCH
7. ZHU-2 PROC FAR	24. HORT:RUSH AX
8. MOV AH, [BX]	25. INT 10H
9. MOV AL, [BX + 01]	26. POP AX
10. MOV DX, [BX + 02]	27. INC CX
11. MOV CX, [BX + 04]	28. CMP CX, BX
12. MOV BX, [BX + 06]	29. JBE HORT
13. CMP AH, 0	30. RET
14. JZ HOR	31. ZHU-2 ENDP
15. MOV AH, OCH	32. CODE-SEG ENDS
16. VER:PUSH AX	33. END
17. INT 10H	

整个程序的结构是以 INT 10H 为主, 利用 INT 10H 中断调用 OCH 功能, 可以在屏幕上写彩色点. 这个彩色点的颜色只能在指定的彩色组中选择. 每个点可以是黑的或白的. 在中分辨率的方式中, 可以生成 320 个水平点乘 200 个垂直点的图象. 程序编写原则完全服从于 BIN 文件的原则.

程序中的第 8 ~ 12 行都有 BX, 其中 [BX] 是对应 FOXBASE 或 FOXPRO 传递到汇编语言的第 1 个参数, [BX + 01] 是第 2 个参数……[BX + 06] 是第 5 个参数. 另外 AH 的值是 INT 10H 用以选择行列方式, 当 AH = 0 时为行方式, AH = 1 时为列方式. AL 的值为中分辨率彩色组选择, DX 为存放垂直起点, CX 为存放行起点, BX 为存放终点. 从 FOXBASE 或 FOXPRO 传来的 5 个参数, 利用 INT 10H 从要求的起点, 逐点扫描到终点而形成一条水平线或垂直线.

(2) FOXPRO 的程序^[4] 由于篇幅所限, 不能把整个程序写出, 现只写出划水平线和垂直线的两个过程, 说明它们之间参数传递的关系, 如果读者想详细了解这个精美界面的编写方法, 欢迎互相交流.

* 划水平线: 从象素 (x_1, y_1) 到 (x_2, y_1) 颜色: LCOR

```
PROC HOR
  PARA LCOR, x1, y1, x2
  CALL ZHU-2 WITH CHR(0) + CHR(LCOR) + CHR(y1%256);
    + CHR(y1/256) + CHR(x1%256) + CHR(x1/256);
```

```
+ CHR(x2%256) + CHR(x2/256)
```

```
RETURN
```

* 划垂直线:从象素(x₁,y₁)到(x₂,y₁)颜色:LCOR

```
PROC VER
```

```
PARA LCOR, x1,y1,y2
```

```
CALL ZHU-2 WITH CHR(1) + CHR(LCOR) + CHR(y1%256);
```

```
+ CHR(y1/256) + CHR(x1%256) + CHR(x1/256);
```

```
+ CHR(y2%256) + CHR(y2/256)
```

```
RETURN
```

上面两段程序结构是相同的,只是参数不同.其中 CHR(0)是传入[BX],CHR(LCOR)是传入[BX+01],CHR(y₁%256)是传入[BX+02],CHR(x₁%256)是传入[BX+04],CHR(x₂%256)是传入[BX+06].这些参数传入到汇编语言,完成划行线功能,如果 CHR(1)则完成划垂线功能.

3 结束语

参数传递不单是应用于立体界面方面,还可以应用于其他场合,例如数据库的数据传送到硬件接口上,或从硬件接口读入数据到数据库中,或者通过电话线进行数据的传送,或者当程序完成某一工作之后,送出一段优美的乐曲等等,只要通过参数传递的原理,就不难完成 FOXBASE 或 FOXPRO 未能完成的部分,可以任你充分发挥.

参 考 文 献

- 1 刘湘,王强. FOXPRO 参考手册. 北京:学苑出版社,1993. 35 ~ 36
- 2 徐振宇. 为 dBASE III 创建汇编语言接口. 电子与电脑. 1993, 1:5 ~ 6
- 3 陈章龙,陈泽文. IBM-PC 机软硬件接口及实验. 北京:人民邮电出版社,1993. 147 ~ 148
- 4 刘彬,高峰. 汉字 FOXBASE + 2.10 编程技巧与实例,北京:海洋出版社,1992. 212 ~ 230

Researches on FOXBASE/FOXPRO and Parametric Passing of Assembly Language

Zhu Jinkun *

Abstract This paper deals with the transmission to the assembly language under the FOXBASE/FOXPRO environment. And by talking as an example the creating of exquisite stereoface, illustrates the fundamentals of parameter transmission the functions of CPU in the process of transmission and the methods of writing the extended program. As an assembly language is the kernel language of the CPU, the further research of parameter transmission makes sense.

Keywords parametric passing, assembly language interface, stereo surface

* Department of Physics, Zhongshan University, Guangzhou 510275