

程序设计初步

会计电算化的浪潮一浪高过一浪，迫切要求每一位会计工作者跟上时代发展的步伐，尽快学习电算化知识，为财会软件的操作、会计电算化软件的开发奠定基础。首先介绍了操作系统与汉字系统，将有助于进入会计电算化程序设计阶段。接着介绍了 C 语言的基本要素，旨在帮助未接触过 C 语言的读者打好基础，也是进行 C 程序设计的指南。最后介绍了 FoxBASE + 的功能与基本操作，它是开发会计电算化软件的主要编程语言。这对初学者来说，无疑也是速成 FoxBASE + 程序设计的一条捷径。

1 程序设计基础知识

1.1 操作系统简介

1.1.1 磁盘操作系统的基本操作

MS-DOS 是微软公司为 PC 机开发的磁盘操作系统，它在我国的计算机系统中，拥有绝大部分市场。目前，最为流行的是 MS-DOS 6.22 它除了具有一般 DOS 所具有的功能外，还提供了磁盘倍增空间、删除树、微软诊断、反病毒与病毒预防等功能，确实可称得上是一种比较完善的操作系统。

MS-DOS 6.22 有 100 多条可用命令，但用得比较多的命令几乎只占一半，其详细的用法与功能，可以利用系统提供的 HELP 命令查看，或者查阅 MS-DOS 6.22 使用手册。

1.1.2 Windows98 操作系统简介

众所周知，DOS 操作系统必须记住 DOS 命令的格式及其复杂的参数，随着计算机的普及和推广，人们对操作系统的要求也越来越高。美国 Microsoft 公司开发的窗口软件 Windows 是一个多任务的图形操作系统，可以说是操作系统的一场工业革命。同以前的 Windows 版本相比，Windows98 中文版具有使用更方便、可靠性更高、速度更快、真正的 Web 集成、更具娱乐性等特点。

进入 Windows98：当用户安装 Windows98 之后，只需打开计算机的电源开关即可启动 Windows98 系统，输入用户名与密码后即可进入 Windows98 系统。

Windows98 的关闭：在关闭或重新启动计算机之前，必须先关闭 Windows98，否则可能会丢失信息。

在结束了 Windows98 中的工作之后，可通过单击“开始”按钮，然后单击“关闭系统”，激活“关闭 Windows”对话框，然后选“关闭计算机”。

在非正常关闭 Windows98 中，如果不得不使用关闭电源的方法来关闭 Windows98 系统的话，当再次开机后，系统首先会自动进行磁盘修复，然后启动 Windows98 系统。当你进入 Windows98 系统后，请注意执行“系统工具”中的“磁盘扫描程序”，以修复长文件名等问题。

1.1.3 Windows NT 5.0 网络操作系统简介

Windows NT 5.0 是目前最容易安装、管理和使用的网络桌面操作系统，是继 Windows98 后的新一代全新网络桌面操作系统，也是未来的主流操作系统，号称消除了所有的 BUG 和不稳定的因素，以其精美的界面、超强的功能迅速受到全世界网络管理员与用户的青睐，是唯一由美国政府认证的网络操作系统。

——启动 Windows NT 5.0

安装 Windows NT 5.0 之后 就可以使用 Windows NT 了。

(1) 开机。

(2) 出现引导程序菜单。如果有两个或多个引导项，则刚安装完成的 Windows NT 项高亮显示出现在列表中的第一项，请按 <Enter> 键。

(3) 当出现登录消息时，按 <Ctrl + Alt + Del> 登录。

(4) 在登录信息对话框中，输入密码，单击“Ok”。如果已是某个域的成员，在“From”后面的文本框中选择要登录的域。

进入 Windows NT 5.0 之后，可以在它的引导之下实现 Windows NT 5.0 的各项功能，通往信息世界。

——退出 Windows NT 5.0

在结束了 Windows NT 5.0 中的工作之后，可打开“Start”菜单，单击“Shut Down...”选项，系统弹出“Shut Down Windows”对话框 如果确认要退出 则单击“Ok”。

1.1.4 UC DOS 7.0 希望汉字系统简介

随着 UC DOS 7.0 进入各行各业、千家万户，为了使读者能更快、更好地掌握这套优秀的软件，很有必要在这里作一简单的介绍。

(1) 启动 UC DOS 7.0

UC DOS 7.0 在安装完毕后，安装程序已在 UC DOS 子目录中建立了两个批处理文件：UCDOS.BAT 和 UP.BAT。运行这两个批处理文件之一，均可启动 UC DOS 7.0。

(2) 退出 UC DOS 7.0

UC DOS 7.0 常驻内存后，可以使用两种方法退出 UC DOS 7.0。

运行系统退出程序 QUIT.COM

在 DOS 提示符下运行 QUIT.COM 即可彻底退出 UC DOS 7.0 完全释放 UC DOS 所占用的所有系统资源。

使用 Ctrl - F5 退出 UC DOS 7.0

在任何时候，均可按 Ctrl - F5 选择功能 4 退出 UC DOS 7.0。

1.1.5 文字编辑工具 WPS 简介

WPS (Word Processing System——字处理系统) 是一个集编辑、修改、复制、打印于一体的汉字处理系统。它提供了丰富的全屏幕编辑功能和输出格式控制及打印功能，为应用文的编写提供了方便。

1.2 C语言初步

C语言是一种通用的结构化程序设计语言，它具有程序执行速度快、可移植性好、源程序书写灵活、图形功能强等优点。而FoxBASE+系统本身就是用C语言编写的，用C语言和FoxBASE+混合编程会感到比其它高级语言更方便，因此，学习C语言尤为重要。

Turbo C 2.0 是使用很广的一种C语言版本，它功能丰富，表达能力强，使用灵活方便。由于其推出的时间较早，在功能上不如近期出现的各种C++版本强大，但是，Turbo C短小精悍，一套完整的系统不超过2Mb。对C++中有而Turbo C中没有的一些感兴趣的C函数，可以采用库维护工具软件Tlib.exe将它们从C++中添加到Turbo C中去，以取人之长，补己之短。本章简介Turbo C 2.0有关更详细的说明，请查阅Turbo C 2.0使用手册。

源程序经过编译后生成目标代码，再经过连接后生成可执行程序。目标代码的后缀是.OBJ，可执行文件的后缀是.COM或.EXE。

源程序可在集成环境下按下<Alt> + <F9>键编译，编译通过后，按<F9>键连接成为可执行文件。也可以采用菜单命令进行编译、连接。

另外，Turbo C 2.0还为用户提供了一个标准的命令行编译程序。用户应根据命令行编译程序TCC.EXE，并指定选择项和文件，即可编译和连接。

通常，用户编译连接好的图形程序文件还不是能完全独立运行的文件，它还需要图形驱动程序的支持，这是很不方便的。用户应该借助BGIOBJ的图形驱动程序和字体的使用程序，将驱动程序转换为目标文件，把驱动程序和字体文件的目标模块加入到GRAPHICS.LIB中。这样，当连接图形子程序时，连接程序就能定位它们。只有这样，才能生成完全独立运行的C图形程序文件。

1.3 FoxBASE+入门

随着微机在管理中应用的不断深入，关系型数据库管理系统FoxBASE+语言以其很强的数据处理能力、简单易学、操作方便、占用磁盘空间与内存小而深受广大用户的喜爱。因此，程序员在开发管理系统时，FoxBASE+就成为他们首选的、主要的编程语言。

1.3.1 进入和退出FoxBASE+的操作

(1) 进入FoxBASE+的操作

先进入DOS状态

启动CCDOS或MS-DOS后，进入中国龙或希望汉字操作系统。

再进入FoxBASE+系统

按要求安装好FoxBASE+系统，键入FOX回车，屏幕显示“.”表示已进入FoxBASE+状态，即可输入FoxBASE+命令进行操作了。

(2) 退出FoxBASE+状态的操作

在点提示符“.”后面键入QUIT回车，即可回到DOS操作系统状态。

1.3.2 命令文件源程序的编译

在FoxBASE+系统中，为了加快程序的运行速度，可把扩展名为PRG文件编译并加密生成扩展名为FOX文件。通过FOXPCOMP.EXE编译*.PRG产生后缀为.FOX的同名文件，实现对源程序的保密和加速运行，保护源程序作者的合法权益。

编译方法 FOXPCOMP -E *.PRG

会计电算化程序设计技术

具备程序设计的基础知识之后，注意力就应该转移到如何进行会计电算化程序设计这个关键问题上。从第四章开始，以医院为例，重点讨论工资核算、物资核算、会计核算、财务分析、科室核算及奖金分配等经济管理中主要组成部分的程序设计。其中，大部分程序具有通用性，小部分程序也只需稍加修改就可以适用于其他企事业单位。

2 会计电算化主控菜单程序设计

菜单技术是实现应用系统程序模块化的工具之一，也是软件人机交换的窗口。如何实现这一人机界面的方便友好，是软件设计的一个极其重要的方面，也是用户衡量软件优劣的重要依据之一。

会计电算化软件的菜单设计采用“弹跳式”方法更适宜，菜单选择采用↑↓→←彩条驱动，〈Esc〉键返回上级菜单，减少了误操作的可能性。整个菜单层次分明、设计简炼、显示迅速、易于维护、画面活跃。

由于菜单程序是系统的控制核心，管理系统开始运行时，首先进入菜单程序，它显示系统所具备的各项功能名称，给出必要的提示信息，因此，会计电算化各级菜单的设计十分重要。它不仅要求设计者具备一定的计算机知识，而且要求具有相当程度的财务会计管理水平。

2.1 会计电算化管理系统应用程序总控结构

会计电算化管理系统应用程序主要是对工资核算、物资核算、会计核算、财务分析、科室核算、奖金分配等进行计算机管理，以提高工作效率和工作质量，因此，其总控结构可以概括为工资核算、物资核算、财务处理、奖金处理、综合服务等 5 个部分 如图 2-1 所示。限于篇幅 本书省略了工资核算、物资核算、系统安装、会计核算等部分的程序设计内容，留给读者自己完成。如图 2-1 所示。

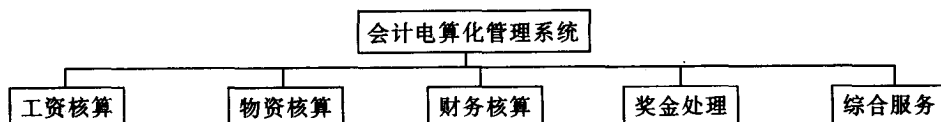


图 2-1 会计电算化管理系统应用程序总控结构图

2.2 采用 FoxBASE + 设计会计电算化主控程序

每种软件下的程序都应该有一个主控程序，它是系统的核心，在主控程序中根据需要选择相应的功能项，则转去执行相应的文件。

2.2.1 会计电算化总调入程序

为了防止非法用户进入系统，在系统程序的入口应设置口令，如果口令输入错误，则用户不能进入操作系统，并返回到 DOS 提示符状态。在程序中，除了设有 ON ERROR DO \EDPA\ERRORPRG 语句，使系统在运行过程中出现错误码时，能执行错误处理程序之外 还设有 ON KEY = 315 DO \EDPA\ZHFV\ZHFV27 语句，使得用户每当按下〈F1〉键时，即提供有关如何操作本系统软件及其它信息的描述。顺便说一句，如果把标题用大一点的字体写出，再加一些图案美化，无疑会大大地增加软件的艺术效果。

注意：如果用户的计算机已经安装 Windows98 等 Windows 操作系统，在“开始”程序”—“MS-DOS 方式”下运行 EDPA 会计电算化管理系统过程中，个别程序运行时可能会显示“Sharing violation reading drive C” 此时 需要退出系统回到 DOS 提示符 并使用“EXIT”命令返回到 Windows 桌面，然后进行以下操作：

选择“开始”按钮之后，选择“关闭系统”，然后选择“重新启动计算机并切换到 MS-DOS 方式 (M)”；

进入汉字系统，如希望汉字系统等；

此后 就可以顺利运行 EDPA 会计电算化管理系统了。需要注意的是：如要进行图形分析打印，请修改 FoxBASE + 的配置文件 CONFIG.FX 中的内存大小设置或者不运行汉字打印驱动程序。否则，可能会出现内存不够，或者请不要运行汉字打印驱动程序，否则，将会出现内存不够。

其程序清单见 EDPAMAIN.PRG。

```
* * * EDPAMAIN.PRG * * * * *
```

```
SET TALK OFF
CLEAR ALL
CLOSE ALL
CLEAR PROG
SET ECHO OFF
SET DOHI OFF
SET STAT OFF
SET ESCA OFF
SET CENT ON
SET DATE ANSI
SET SAFE OFF
SET SCOR OFF
SET EXAC OFF
SET BELL OFF
SET CONF OFF
SET CLEAR OFF
SET DEVI TO SCREEN
```

```

SET PRINT OFF
SET PRINT TO PRN
SET MARGIN TO 0
SET MESS TO 24
ON ERROR DO \EDPA\ERRORPRG
ON KEY = 315 DO \EDPA\ZHFV\ZHFV27
SET PROC TO \EDPA\EDPAMAIN
IF VAL(SYS(12)) <75 * 1024
    @ 10, 20 SAY " 内存太少 "
    ?REPL(CHR(7), 6)
    DE = INKEY(4)
    CLOSE ALL
    QUIT
ENDIF
CLEAR TYPE
DO \EDPA\WINDOW WITH 20, 46, 10, 16, "/N", "W/GB +
PUBL YFKL, PA1, PA2, PA3, PA4, PA5, PA6, PA7, PA8
USE \EDPA\ZHFV\EDPAPASS
GO TOP
IF ! EOF( )
    YFMC = SPACE(8)
    @ 11, 30 SAY "EDPAS"
    @ 13, 22 SAY " 请输入用户名 : " GET YFMC
    READ
ELSE
    DO \EDPA\WINDOW WITH 8, 64, 10, 18, "/N", "W/RB"
    @ 11, 20 SAY " 首次进入会计电算化管理系统设置 "
    BU00 = SPACE(8)
    DWMCSR = SPACE(30)
    @ 13, 16 SAY " 请输入会计主管 : " GET BU00
    @ 15, 16 SAY " 请输入单位名称 : " GET DWMCSR
    BU10 = SPACE(4)
    READ
    IF LEN(TRIM(BU00))= 0
        USE
        QUIT
    ENDIF
    STORE CHR(ASC(SUBS(BU00, 1, 1 )) - 30) + CHR(ASC(SUBS(BU00, 2, 1) ) - 25) +
    CHR(ASC(SUBS(BU00, 3, 1 )) - 20) + CHR(ASC(SUBS(BU00, 4, 1 )) - 15) +
    CHR(ASC(SUBS(BU00, 5, 1)) - 10) + CHR(ASC(SUBS(BU00, 6, 1 )) - 5) +
    CHR(ASC(SUBS(BU00, 7, 1) ) - 3) + CHR(ASC(SUBS(BU00, 8, 1) ) - 2) TO BU000
    STORE CHR(ASC(SUBS(RIGHT(BU10, 4), 1, 1) ) - 30) + CHR(ASC(SUBS(RIGHT(BU10, 4), 2, 1) )
- 25);
    + CHR(ASC(SUBS(RIGHT(BU10, 4), 3, 1)) - 20) + CHR(ASC(SUBS(RIGHT(BU10, 4), 4, 1) ) -

```

```

15);
  TO BU100
  APPE BLAN
  REPL CZXM WITH BU000, PAS1 WITH BU100, PAS2 WITH BU100, PAS3 WITH BU100
  REPL PAS4 WITH BU100, PAS5 WITH BU100, PAS6 WITH BU100, PAS7 WITH BU100
  REPL PAS8 WITH BU100, DWMCS WITH DWMCSR
  ??REPL(CHR(7), 10)
  @ 17, 12 SAY“按任一键后 你就有权进入 EDPA 系统了( 尚未设置口令 )!”
  `WAIT `
  QUIT
ENDIF
STORE CHR(ASC(SUBS(YFMC, 1, 1 ))- 30) + CHR(ASC(SUBS(YFMC, 2, 1 ))- 25) + ;
CHR(ASC(SUBS(YFMC, 3, 1 ))- 20) + CHR(ASC(SUBS(YFMC, 4, 1 ))- 15) + ;
CHR(ASC(SUBS(YFMC, 5, 1 ))- 10) + CHR(ASC(SUBS(YFMC, 6, 1 ))- 5) + ;
CHR(ASC(SUBS(YFMC, 7, 1 ))- 3) + CHR(ASC(SUBS(YFMC, 8, 1 ))- 2) TO YFMC
LOCATE FOR CZXM = YFMC
IF EOF()
  @ 14, 24 SAY“ 无此用户 !”
  “WAIT `
  SET COLO TO W/N
  CLEAR
  QUIT
ENDIF
PA1 = PAS1
PA2 = PAS2
PA3 = PAS3
PA4 = PAS4
PA5 = PAS5
PA6 = PAS6
PA7 = PAS7
PA8 = PAS8
BU1 = SPACE(4)
@ 14, 22 SAY“请输入用户口令 :”
SET COLO TO W/N, GB/GB
@ 14, 38 GET BU1 PICT“! !!! ”
READ
STORE CHR(ASC(SUBS(BU1, 1, 1 ))- 30) + CHR(ASC(SUBS(BU1, 2, 1 ))- 25);
+ CHR(ASC(SUBS(BU1, 3, 1 ))- 20) + CHR(ASC(SUBS(BU1, 4, 1 ))- 15) TO YFKL
IF PA1#YFKL
  SET COLO TO W/N, W/RB
  @ 15, 24 SAY “ 无此权限 !”
  “WAIT `
  SET COLO TO W/N
  CLEAR

```

```

QUIT
ENDIF
USE
DO \EDPA\EDPAMENU
RETU

```

用户信息数据库结构如下：

Structure for database: \EDPA\ZHFW\EDPAPASS.DBF

Field	Field Name	Type	Width	Dec
1	CZXM	Character	8	
2	PAS1	Character	4	
3	PAS2	Character	4	
4	PAS3	Character	4	
5	PAS4	Character	4	
6	PAS5	Character	4	
7	PAS6	Character	4	
8	PAS7	Character	4	
9	PAS8	Character	4	
10	DWMC	Character	30	

错误处理程序如下：

```

* * * * ERRORPRG. PRG
SET TALK OFF
SET COLO TO W/B
CLEAR
DO \EDPA\WINDOW WITH 4, 74, 10, 17, "/N", "W/R"
@ 11, 30 SAY " * * * * 错误信息 * * * *"
DO CASE
    CASE ERROR( ) = 108. OR. ERROR( ) = 109
        @ 14, 20 SAY "文件正在被其他用户使用，或者记录正在被其他用户使用！"
        RETRY
    CASE ERROR( ) = 26. OR. ERROR( ) = 20
        @ 14, 20 SAY "数据库未被索引或记录未被索引"
        REINDEX
    CASE ERROR( ) = 148
        @ 14, 20 SAY "网络服务器正在忙碌！"
        RETRY
    OTHER
        @ 14, 20 SAY MESSAGE( )
QUIT

```

ENDCASE

具有立体投影效果的窗口自定义函数如下：

```

* * * * WINDOW. PRG
PARA X_WINDOW_1, X_WINDOW_2, Y_WINDOW_1, Y_WINDOW_2, COLOR_1, COLOR_2
SET COLO TO W/B
CLEAR

```

```

SET COLOR TO &COLOR 1
@ Y_WINDOW_1 + 1, X_WINDOW_1 + 2 CLEAR TO Y_WINDOW_2 + 1, X_WINDOW_2 + 3
SET COLOR TO &COLOR 2
@ Y_WINDOW_1, X_WINDOW_1 CLEAR TO Y_WINDOW_2, X_WINDOW_2
@ Y_WINDOW_1, X_WINDOW_1 SAY 'r' + REPL('-', (X_WINDOW_2 - X_WINDOW_1) / 2 - 1) + 'r'
@ Y_WINDOW_2, X_WINDOW_1 SAY 'L' + REPL('-', (X_WINDOW_2 - X_WINDOW_1) / 2 - 1) + 'L'
X = Y_WINDOW_1 + 1
L = 1
DO WHILE L < Y_WINDOW_2 - Y_WINDOW_1
    @ X, X_WINDOW_1 SAY ' | '
    @ X, X_WINDOW_2 SAY ' | '
    X = X + 1
    L = L + 1
ENDDO
@ Y_WINDOW_1 + 2, X_WINDOW_1 + 2 SAY REPL(CHR(205), X_WINDOW_2 - X_WINDOW_1 - 2)
RETU

```

如果要让用户更简便、更快地进入操作系统，那么，建立一个批处理文件是行之有效的。批处理文件建立之后，用户只需在 EDPA.BAT 所在目录下输入 EDPA 并回车即可进入系统。

```

REM EDPA.BAT
@ ECHO OFF
FOX \EDPA\EDPAMAIN
ECHO ON

```

当输入 EDPA 并回车 尔后输入正确的用户名、口令之后 屏幕显示 EDPAS 的主菜单如下：

EDPAS 会计电算化管理系统

主 菜 单
W——工资核算
M——物资核算
F——财务处理
P——奖金处理
S——综合服务
X——退出系统

2.2.2 系统主菜单程序

根据会计电算化系统 EDPAS(Electron Date Processing Accounting System) 的特点 设计系统的主菜单如下：

```

* * * * EDPAMENU.PRG
SET TALK OFF
IF PA1#YFKL
    DO \EDPA\WINDOW WITH 20, 40, 12, 15, "/N", "W/GB +
    @ 13, 24 SAY " 无此权限！"
    "WAIT"

```

```

SET COLO TO W/N
CLEAR
QUIT
ENDIF
MENZCL = "W — 工资核算 M — 物资核算 F — 财务处理 P — 奖金处理 ;
S — 综合服务 X — 退出系统 "
SWITHTF = . T.
PRZCL = 1
DO WHILE . T.
    IF SWITHTF
        SET COLO TO W/B
        CLEAR
        SET COLO TO /N
        @ 5, 12 TO 16, 51 CLEAR
        SET COLO TO / + W
        @ 6, 10 TO 15, 49 CLEAR
        SET COLO TO / * R
        @ 7, 11 TO 14, 48 CLEAR
        SET COLO TO + W / * R
        @ 8, 11 SAY REPL( " — ", 19)
        SET COLO TO + R / B
        @ 4, 4 SAY " ┌───┐ "
        @ 5, 4 SAY " │   │ "
        @ 6, 4 SAY " └───┘ "
        @ 8, 54 SAY " 版权所有复制必究 "
        SET COLO TO G/B
        @ 10, 54 SAY " 作者 :
        SET COLO TO W / + R
        @ 4, 10 SAY " ┌─┐" + REPL( " — ", 18) + " ┌─┐ "
        @ 5, 10 SAY " │ │ " + SPACE(36) + " │ │ "
        @ 6, 10 SAY " └─┘" + REPL( " — ", 18) + " └─┘ "
        @ 5, 20 SAY "EDPAS 会计电算化管理系统 "
        SET COLO TO * GR + / R
        @ 7, 24 SAY " 主 菜 单
    ENDIF
    SET COLO TO * GR + / R
    L = 1
    DO WHILE L <= 6
        @ L + 8, 20 SAY SUBSTR(MENZCL, (L - 1) * 16 + 1, 16)
        L = L + 1
    ENDDO
    SET COLO TO * GR + / + 1
    @ PRZCL + 8, 11 SAY SPAC(9) + SUBS(MENZCL, (PRZCL - 1) * 16 + 1, 16) + SPACE(13)
    INK = 0

```

```

SET COLO TO +G/W
DO WHILE INK = 0 . OR. INK#27 . AND. INK#5 . AND. INK#24 . AND. ;
    INK#13 . AND. INK#88 . AND. INK#119 . AND. INK#77 . AND. INK#109 ;
    . AND. INK#70 . AND. INK#102 . AND. INK#80 . AND. INK#112;
    . AND. INK#83 . AND. INK = 115 . AND. INK = 88 . AND. INK = 120
    INK = INKEY()
ENDDO
DO CASE
    CASE INK = 5
        SWITHTF = . F.
        SET COLO TO N/*R
        @ PRZCL + 8, 11 SAY SPACE(9) + SUBS(MENZCL, (PRZCL - 1) * 16 + 1, 16) + SPACE(13)
        PRZCL = PRZCL - 1
        IF PRZCL <1
            PRZCL = 6
        ENDIF
    CASE INK = 24
        SET COLO TO N/*R
        SWITHTF = . F.
        @ PRZCL + 8, 11 SAY SPAC(9) + SUBSTR(MENZCL, (PRZCL - 1) * 16 + 1, 16) + SPACE(13)
        PRZCL = PRZCL + 1
        IF PRZCL > 6
            PRZCL = 1
        ENDIF
    CASE INK = 87 . OR. INK = 119
        PRZCL = 1
        SWITHTF = . F.
        SET COLO TO N/*R
        @ 9, 11 TO 14, 19 CLEAR
        @ 9, 36 TO 14, 48 CLEAR
        @ PRZCL + 8, 11 SAY SPAC(9) + SUBSTR(MENZCL, (PRZCL - 1) * 16 + 1, 16) + SPACE(13)
    CASE INK = 77 . OR. INK = 109
        PRZCL = 2
        SWITHTF = . F.
        SET COLO TO N/*R
        @ 9, 11 TO 14, 19 CLEAR
        @ 9, 36 TO 14, 48 CLEAR
        @ PRZCL + 8, 11 SAY SPAC(9) + SUBSTR(MENZCL, (PRZCL - 1) * 16 + 1, 16) + SPACE(13)
    CASE INK = 70 . OR. INK = 102
        PRZCL = 3
        SWITHTF = . F.
        SET COLO TO N/*R
        @ 9, 11 TO 14, 19 CLEAR
        @ 9, 36 TO 14, 48 CLEAR

```

```

    @ PRZCL + 8, 11 SAY SPAC(9) + SUBSTR(MENZCL, (PRZCL - 1) * 16 + 1, 16) + SPACE(13)
CASE INK = 80 . OR. INK = 112
    PRZCL = 4
    SWITHTF = . F.
    SET COLO TO N / * R
    @ 9, 11 TO 14, 19 CLEAR
    @ 9, 36 TO 14, 48 CLEAR
    @ PRZCL + 8, 11 SAY SPAC(9) + SUBSTR(MENZCL, (PRZCL - 1) * 16 + 1, 16) + SPACE(13)
CASE INK = 83 . OR. INK = 115
    PRZCL = 5
    SWITHTF = . F.
    SET COLO TO N / * R
    @ 9, 11 TO 14, 19 CLEAR
    @ 9, 36 TO 14, 48 CLEAR
    @ PRZCL + 8, 11 SAY SPAC(9) + SUBSTR(MENZCL, (PRZCL - 1) * 16 + 1, 16) + SPACE(13)
CASE INK = 13
    SET COLO TO W / B, N / W
    PROG = " \ EDPA \ GZHS \ GZHSZX \ EDPA \ WZHS \ WZHSZX \ EDPA \ CWCL \ CWCLZX \ ED-
PA;
JJCL \ JJCLZX \ EDPA \ ZHFW \ ZHFWZX \ EDPA \ TCXT \ TCXTZS"
    PROG1 = SUBS(PROG, (PRZCL - 1) * 17 + 1, 17)
    IF PROG1 = " \ EDPA \ TCXT \ TCXTZS"
        EXIT
    ENDIF
    DO &PROG1
    SWITHTF = . T.
CASE INK = 88 . OR. INK = 120
    EXIT
ENDCASE
ENDDO
CLOSE ALL
RUN DEL TEMP * . *
SET COLO TO W / N
CLEAR
QUIT

```

3 综合服务子系统程序设计

本章从系统维护和其它服务两个方面分别给出各个模块的程序设计，其中揉进了不少的设计技巧。它们既为用户提供了非常重要的有关信息，又为读者详细介绍了很多实用程序的编程技术。

其菜单程序清单如下：

```
* * ZHFWZX. PRG
SET TALK OFF
IF PA8#YFKL
    DO \EDPA \WINDOW WITH 18, 40, 13, 16, "/N", "W/GB +
    @ 14, 24 SAY " 无此权限 !"
    "WAIT
    RETU
ENDIF
SET COLO TO W/B
CLEAR
MEM = " 系统维护其它服务 "
MEM1 = " 设置单位名称用户权限设置用户信息   软盘格式化数据软盘备份数据软盘恢复 "
MEM2 = " 编辑器      大小写转换  ASCII  查询  彻底删除文件用户接口   系统帮助
STORE 1 TO PR, PC, BT
SWIT = . T.
DO WHILE . T.
    IF SWIT
        SET COLO TO + W/B
        CLEAR
        @ 0, 23 SAY "  " + REPL(" ", 15) + "  "
        @ 1, 23 SAY " | " + SPACE(30) + " | "
        @ 2, 23 SAY "  L" + REPL(" ", 15) + "  "
        @ 1, 32 SAY " 综合服务子系统 "
        SET COLO TO + W/GB
        @ 4, 0 TO 4, 80 CLEAR
        L = 1
        DO WHILE L <= 2
            @ 4, (L - 1) * 30 + 15 SAY SUBS(MEM, (L - 1) * 8 + 1, 8)
            L = L + 1
        ENDDO
        IF PC < 10
            CPC = STR(PC, 1)
        ELSE
            CPC = STR(PC, 2)
        ENDIF
    ENDIF
```

```

SET COLO TO /N
LLL = LEN(MEM&CPC)/12
@ 5 + 2, (PC - 1) * 30 + 15 + 1 TO LLL + 6, PC * 30 + 2 CLEAR
SET COLO TO N/W
@ 6, (PC - 1) * 30 + 15 TO LLL + 5, PC * 30 CLEAR
L = 1
DO WHILE L <= LLL
    @ L + 5, (PC - 1) * 30 + 15 SAY SPAC(2) + SUBSTR(MEM&CPC, (L - 1) * 12 + 1, 12)
    SET COLO TO + W / * R
    @ 4, (PC - 1) * 30 + 15 SAY SUBSTR(MEM, (BT - 1) * 8 + 1, 8)
    L = L + 1
    SET COLO TO N/W
ENDDO
ENDIF
SET COLO TO + W / * R
@ PR + 5, (PC - 1) * 30 + 15 SAY SPAC(2) + SUBS(MEM&CPC, (PR - 1) * 12 + 1, 12) + SPAC(2)
INK = 0
SET COLO TO N/W
DO WHILE INK = 0 . OR. INK#27 . AND. INK#5 . AND. INK#4 . AND. INK#24 . AND. ;
INK#19 . AND. INK#13
    INK = INKEY()
ENDDO
DO CASE
    CASE INK = 5
        SET COLO TO N/W
        @ PR + 5, (PC - 1) * 30 + 15 SAY SPAC(2) + SUBS(MEM&CPC, (PR - 1) * 12 + 1, 12) + SPAC
(2)
        PR = PR - 1
        IF PR < 1
            PR = INT(LEN(MEM&CPC)/12)
        ENDIF
        SWIT = . F.
    CASE INK = 24
        SET COLO TO N/W
        @ PR + 5, (PC - 1) * 30 + 15 SAY SPAC(2) + SUBS(MEM&CPC, (PR - 1) * 12 + 1, 12) + SPACE(2)
        PR = PR + 1
        IF PR > LEN(MEM&CPC)/12
            PR = 1
        ENDIF
        SWIT = . F.
    CASE INK = 19
        PR = 1
        BT = BT - 1
        IF BT < 1

```

```

    BT = 2
ENDIF
SET COLO TO W/1
@ 5, (PC - 1) * 30 + 15 TO LEN(MEM&CPC)/12 + 6, PC * 30 + 15 + 2 CLEAR
PC = PC - 1
SWIT = . T.
IF PC <1
    PC = 2
ENDIF
CASE INK = 4
PR = 1
SET COLO TO W/1
@ 5, (PC - 1) * 30 + 15 TO LEN(MEM&CPC)/12 + 6, PC * 30 + 15 + 1 + 1 CLEAR
    BT = BT + 1
IF BT > 2
    BT = 1
ENDIF
PC = PC + 1
SWIT = . T.
IF PC > 2
    PC = 1
ENDIF
CASE INK = 13
IF PR <10
    CPR = STR(PR, 1)
ELSE
    CPR = STR(PR, 2)
ENDIF
DO \EDPA \ZHFW \ZHFW&CPC. &CPR
SET COLO TO W/B
CLEAR
SWIT = . T.
CASE INK = 27
    EXIT
ENDCASE
ENDDO
RETU

```

3.1 系统维护

系统维护是保证应用系统正常运行和数据正确的基础，是每一个应用系统必备的模块之一。

3.1.1 设置单位名称

为了使应用系统具有通用性，系统允许用户修改单位名称。但是，这里没有进行加

密，当然，读者也可以将名称变换加密后存入数据库，可以保密。

```
***ZHF11.PRG
SET TALK OFF
DO \EDPA\WINDOW WITH 18, 50, 10, 14, "/N", "W/GB +
USE \EDPA\ZHF11\EDPAPASS
GO TOP
@ 11, 27 SAY " 设置单位名称 "
@ 13, 20 GET DWMC
READ
USE
RETU
```

3.1.2 用户权限设置

通过它定义本系统的合法用户和每个用户的口令、各子系统的使用权限等，并把这些信息存放在 EDPAPASS.DBF库文件中。

每个口令均为 4 个字符。由于每个口令均采用变换加密方式，即使打开数据库也看不出其真实的用户名、口令及使用权限，其保密性非常强。如果用户的口令忘记了，那只有系统员才能在这里查到，否则，该用户将无法进入系统。

需要注意的是在设置用户权限时“总控”与“综合服务”必须进行授权，否则用户将无法进入系统。

```
****ZHF12.PRG
SET TALK OFF
USE \EDPA\ZHF12\EDPAPASS
LOCATE FOR CZXM = YFMC
IF RECNO()#1
    @ 14, 24 SAY " 无此权限！"
    "WAIT "
    RETU
ENDIF
ZHF1 = " 总控 "
ZHF2 = " 工资核算 "
ZHF3 = " 物资核算 "
ZHF4 = " 会计核算 "
ZHF5 = " 财务分析 "
ZHF6 = " 科室核算 "
ZHF7 = " 奖金分配 "
ZHF8 = " 综合服务 "
DO WHILE .T.
DO \EDPA\WINDOW WITH 18, 50, 6, 14, "/N", "W/GB +
@ 7, 22 SAY " 用户权限设置 ___ 菜单 "
CHOICE = INKEY ( ) - 48
SET COLO TO W/GB +, /W
IF CHOICE < 0 . OR. CHOICE > 4
    @ 9, 20 PROM SPACE(10) + " 增加用户 " + SPAC(12)
```

```

@ ROW() + 1, 20 PROM SPACE(10) + " 修改用户权限." + SPACE(8)
@ ROW() + 1, 20 PROM SPACE(10) + " 删除用户 " + SPACE(12)
@ ROW() + 1, 20 PROM SPACE(10) + " 用户信息 " + SPACE(12)
MENU TO CHOICE

```

```
ENDIF
```

```
DO CASE
```

```
CASE CHOICE = 1
```

```
DO \EDPA\WINDOW WITH 18, 60, 10, 15, "/N", "W/GB + "
```

```
@ 11, 30 SAY " 增加用户 "
```

```
BU0 = SPACE(8)
```

```
@ 13, 20 SAY " 请输入用户名 (最好用中文):" GET BU0
```

```
READ
```

```
IF LEN(TRIM(BU0)) #0
```

```
STORE CHR(ASC(SUBS(BU0, 1, 1)) - 30) + CHR(ASC(SUBS(BU0, 2, 1)) - 25) + ;
```

```
CHR(ASC(SUBS(BU0, 3, 1)) - 20) + CHR(ASC(SUBS(BU0, 4, 1)) - 15) + ;
```

```
CHR(ASC(SUBS(BU0, 5, 1)) - 10) + CHR(ASC(SUBS(BU0, 6, 1)) - 5) + ;
```

```
CHR(ASC(SUBS(BU0, 7, 1)) - 3) + CHR(ASC(SUBS(BU0, 8, 1)) - 2) TO BU0
```

```
LOCATE FOR CZXM = BU0
```

```
IF FOUND()
```

```
@ 14, 20 SAY " 错误 用户名已存在!"
```

```
"WAIT
```

```
ELSE
```

```
BU1 = SPACE(4)
```

```
BU2 = "6402"
```

```
@ 14, 20 SAY " 设定初始口令为:" GET BU1 PICT "!!! "
```

```
READ
```

```
IF BU1 = SPACE(4)
```

```
BU1 = BU2
```

```
ENDIF
```

```
STORE CHR(ASC(SUBS(BU1, 1, 1)) - 30) + CHR(ASC(SUBS(BU1, 2, 1)) - 25) + ;
```

```
CHR(ASC(SUBS(BU1, 3, 1)) - 20) + CHR(ASC(SUBS(BU1, 4, 1)) - 15) TO BU1
```

```
STORE CHR(ASC(SUBS(BU2, 1, 1)) - 40) + CHR(ASC(SUBS(BU2, 2, 1)) - 35) + ;
```

```
CHR(ASC(SUBS(BU2, 3, 1)) - 30) + CHR(ASC(SUBS(BU2, 4, 1)) - 25) TO BU2
```

```
APPE BLAN
```

```
REPL PAS1 WITH BU2, PAS2 WITH BU2, PAS3 WITH BU2, PAS4 WITH BU2
```

```
REPL PAS5 WITH BU2, PAS6 WITH BU2, PAS7 WITH BU2, PAS8 WITH BU2
```

```
REPL CZXM WITH BU0
```

```
@ 1, 10 SAY " 请设置用户权限 ← 奇数次选中 偶数次放弃 <ESC> 退出 )"
```

```
DIME AQ(8)
```

```
AQ = . F.
```

```
Y = 1
```

```
AOPTION = 1
```

```
DO WHILE . T.
```

```
SET COLO TO W/B
```