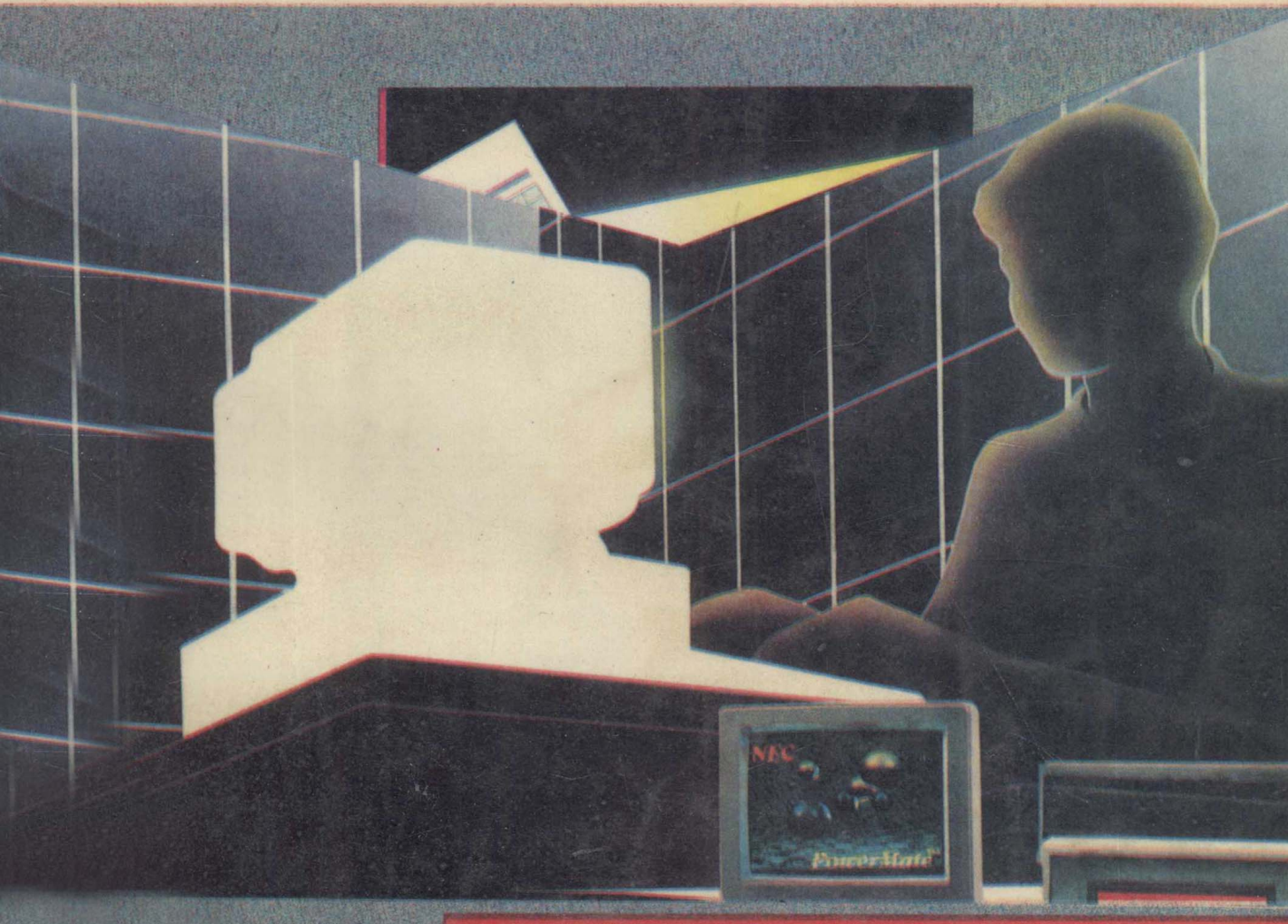




计算机知识普及系列丛书

dBASE III 培训教程

王路敬 编著



学苑出版社

dBASE III 培训教程

王路敬 编著

路 扬 审校

学苑出版社

1994.

(京)新登字 151 号

内 容 提 要

本书是作者在多年培训班教材的基础上经过加工整理编写而成。该书实用性强、应用程序实例多,是各种微机培训班的好教材。

欲购本书的用户,请直接与北京 8721 信箱联系,电话:2562329,邮码:100080。

计算机知识普及系列丛书

dBASE II 培训教程

编 著:王路敬
审 校:路 扬
责任编辑:甄国宪
出版发行:学苑出版社 邮政编码:100036
社 址:北京市海淀区万寿路西街 11 号
印 刷:双青印刷厂
开 本:787×1092 1/16
印 张:22.625 字 数:536 千字
印 数:1~4000 册
版 次:1994 年 4 月北京第 1 版第 1 次
ISBN7-5077-0821-7/TP·19
本册定价:15.00 元

学苑版图书印、装错误可随时退换

前 言

深受用户欢迎在IBM-PC及其兼容机长城系列微机上运行的关系型数据库管理系统dBASEIII,在企事业事务管理的各个方面得到了较广泛的应用,近几年来,在我们举办的多期微机技术培训中,dBASEIII这一软件做为教学的主要内容加以推广,在应用推广中,很多用户,很多学员,尤其是初学者,普遍反映缺少编程方法,感到编程困难,一些有一定基础的用户则迫切希望把编程思路拓宽,掌握一些编程技巧,也有相当多的用户求能通过具体应用程序设计实例的分析,得到实用而有价值的参考模型,缩短编程时间,加速应用步伐,为此编写了这本书奉献给广大读者。

本书的编写具有两个显著特点:(1)实用性强,该书每一章节素材的选取来自教学的实践,工作的实践或根据用户愿望与提出要求,经过分析、整理,充实和系统化,有效地缩短了教与学,学与用之间的距离,书的初稿已作为多期培训班的主要教材使用,受到学员的欢迎和好评。(2)应用程序实例多,全书几乎所有问题的说明都是给出具体的程序,通过程序实例体现编写应用程序的基本方法和技巧,而很多程序实例可以不加修改或只要稍加修改就可用到相应的问题中,具有相当的通用性。

全书共分三章,第一章是dBASEIII编程方法,在明确dBASEIII应用程序设计的基本思路前提下,着重说明菜单程序的一般结构形式,设计步骤和处理方法,该章分为主菜单程序的设计以及实现主菜单功能的其他功能程序的设计,诸如建立数据库文件程序的设计,增加和插入记录,删除记录,查询,修改数据,统计运算,打印等等程序的设计;第二章是dBASEIII编程技巧,该章以具体的程序实例,编写应用程序可借鉴的一些技巧,例如,保密口令字的设置方法,加快数据,程序输入方法,提高程序执行速度的措施,打印实线报表,宽行报表以及改变汉字字型等等;第三章是实用程序设计实例,该章列举了五个实用管理程序设计实例,例如人事管理,工资发放管理,单位固定资产管理,农作物品种资源管理以及小型事务综合管理等。实例不但给出了系统的数据库结构,系统功能框图,程序流程图,还对程序的设计作了必要的分析说明,并给出了所使用的程序清单,向读者提供了有实用价值的参考程序,本书既可作为dBASEIII应用培训的参考教材,也可以作初学者,甚至有一定基础的人进行编程的自学资料。

本书的编写工作得到了中国科学院希望高级电脑技术公司的大力支持,编者在此表示衷心感谢。编者期望本书能受到读者的欢迎,但由于编者水平及对这一软件应用实践的限制,书中难免有不当或错误之处,恳请读者批评指正,不胜感激。

目 录

第一章 dBASE III 编程方法

- 1.1 dBASE III 应用程序设计的基本思路 (1)
- 1.2 dBASE III 菜单程序的设计 (2)
 - 1.2.1 主菜单程序的设计 (8)
 - 1.2.2 建立数据库文件的程序设计 (23)
 - 1.2.3 增加和插入记录的程序设计 (28)
 - 1.2.4 删除记录的程序设计 (37)
 - 1.2.5 查询程序的设计 (51)
 - 1.2.6 修改数据程序的设计 (97)
 - 1.2.7 统计程序的设计 (115)
 - 1.2.8 打印程序的设计 (130)
- 1.3 dBASE III 程序调试中常见错误信息及产生原因 (144)

第二章 dBASE III 编程技巧

- 2.1 口令字的设置 (147)
 - 2.1.1 在程序的总入口处设置口令字 (147)
 - 2.1.2 在修改数据入口处设置口令字 (149)
- 2.2 随机建立、打开数据库, 修改结构 (151)
 - 2.2.1 随机建立和打开数据库的一种处理方法 (151)
 - 2.2.2 在应用系统中修改库结构的一种操作方法 (152)
- 2.3 明细型字段的使用技巧 (153)
- 2.4 提高dBASE III程序及数据的输入速度 (154)
 - 2.4.1 加快输入程序的措施 (154)
 - 2.4.2 加快数据输入的措施 (154)
- 2.5 加快数据统计的方法 (155)
- 2.6 提高程序执行速度的措施 (157)
- 2.7 dBASE III的打印技巧 (162)
 - 2.7.1 打印命令的选取 (162)
 - 2.7.2 打印实线报表的操作 (164)
 - 2.7.3 改变汉字字型的操作 (166)
 - 2.7.4 宽行报表的输出技巧 (168)
 - 2.7.5 加宽打印和显示方法 (169)
 - 2.7.6 数值为零的字段不打印方法 (169)
- 2.8 使喇叭鸣叫与在提示行中显示信息 (169)

2.9 屏幕颜色的设置	(170)
-------------------	-------

第三章 实用程序设计实例

3.1 人事管理应用程序实例.....	(174)
3.1.1 一个简单的人事档案管理系统的的设计	(174)
1. 数据库结构	(174)
2. 人事档案管理系统功能的实现	(175)
3.1.2 一般人事档案管理系统的的设计	(182)
1. 数据库结构	(182)
2. 系统总体逻辑框图	(185)
3. 实现系统功能的程序清单	(185)
3.2 工资发放管理应用程序实例.....	(217)
3.2.1 一个简单工资发放管理应用程序的设计	(217)
1. 工资数据库结构及框图	(217)
2. 工资发放管理程序清单	(218)
3.2.2 一般工资发放微机管理应用程序设计	(229)
1. 工资数据库结构	(229)
2. 工资发放管理程序总体功能结构框图	(230)
3. 系统组成及程序清单	(231)
3.3 固定资产管理应用程序设计实例.....	(250)
1. 固定资产数据库结构	(250)
2. 固定资产应用程序清单	(251)
3.4 农业生物种质资源管理应用程序设计实例.....	(285)
1. 大豆品种资源数据库结构	(285)
2. 系统结构逻辑框图	(288)
3. 大豆品种资源管理程序清单	(290)
3.5 小型综合事务管理程序设计实例.....	(341)
1. 管理程序的组成	(341)
2. 管理程序的总体逻辑框图	(342)
3. 数据库结构	(342)
4. 程序清单	(344)

第一章 dBASE III 编程方法

dBASE III 是广大微机用户比较喜欢的用于事务管理方面的软件。它丰富的语句自成体系，易学易懂，使用方便；灵活的人机会话功能，使得计算机操作人员可以随时通过键盘和显示屏幕与正在进行的程序对话，输入需要的数据，选择不同的程序功能，干预程序的执行，使程序按照不同操作人员的不同要求进行相应的工作；先进的结构式程序设计功能，使程序逻辑组织模块化。怎样利用 dBASE III 提供这一套语言系统开发出解决较为复杂一些问题的应用程序呢？本章从事务管理微机实用程序设计的基本方法入手进行介绍，以便使初学 dBASE III 程序设计的人能较快的进入实用程序的开发与设计。

1.1 dBASE III 应用程序设计的基本思路

利用 dBASE III 设计一个应用程序时，对于初学程序设计的人来说，感到困难的是缺乏程序设计的方法。而设计一个应用程序采用什么方法又往往是相当重要的，如果方法不当，会导致事倍功半的结果。dBASE III 提供了一套先进的结构式应用开发语言，它不但语句丰富、自成体系，具有灵活的人机会话功能，而且它的语言体现了结构式程序设计的特征，用 dBASE III 语言编写的程序是一个结构式的程序。通常一个 dBASE III 的实用程序由一个主控程序和若干个子程序组成的。一个稍微复杂的应用管理系统中有许多大小不同的程序，这里就存在从何处入手的问题。传统的方法是先从下层入手，也就是说先实现下层程序（模块），因为下层某一程序功能比较具体，可以实现一部分就调试一部分。但是这种方法包含着很大的危险性，因为它不是从整个系统的功能出发，使用这种从下往上的方法，很可能出现各个模块单独调试运行都可通过，但是系统联调试时却产生通不过的现象，其主要毛病产生于整体结构与各程序块间的接口处。

dBASE III 实用程序的设计值得提倡的基本方法是：“从上到下，先粗后细，化整为零，分而治之”。这种方法的基本思路是：

1. 先实现上层，后实现下层。具体来说就是首先调试整个系统的结构和各个程序之间的接口，在确保全系统结构及各模块间接口正确无误之后，再实现下层程序的功能。

2. 在实现上层模块时，下层模块只保留其名字及输入参数和输出参数。而模块本身的处理先空着，或者只保留一两处象征性的语句或简单的显示信息。这样，上层模块可以调用下层模块，虽然其具体处理功能还未实现，可以验证调用是否成功，输入输出的传输是否正确等。但是这样就可以调试系统结构，保证其正确性。

3. 化整为零就是先部分后全部。即把整个系统分成若干“版本”，先实现一个只反映系统骨架的版本，而后不断扩充，最后实现系统全部功能。

采用从上到下的方法既能有效地解决接口问题，又便于尽早发现系统结构整体上存在的问题，从而收到理想的效果。

实现上述方法的操作可分为以下七步进行：

- (1) 明确管理系统应具有的功能，按这些功能划分程序模块，画出系统总体框图。
- (2) 要管理的信息由哪几种数据类型组成，字段多少，长度怎样，设计确定数据库

的结构，并输入一定数量的数据。

(3) 根据程序的功能画出程序的细框图。每一功能子程序的框图描述应尽量清晰详细，把每一步要做什么都在框图上标明。

(4) 若需输出报表，则要设计报表格式。若屏幕输入数据，则设计屏幕格式文件。

(5) 检查程序框图没有错误，便可动手分模块编写程序。若几个人共同编程序，要按人分模块。完成功能比较完整的模块，不要将一个功能模块分给几个人来写，以免引起混乱。

(6) 程序编写完以后，先分模块进行调试。调试的目的是检查该模块功能上是否达到了预定的要求，并排除错误。当各模块都通过分调之后，再联起来进行统调。

(7) 统调通过后试运行，试运行无误后便可投入正常使用。

在进行程序设计时提倡多模块短程序。这样可以增强程序的灵活性，既便于查错，又便于改动。

1.2 dBASE III 菜单程序的设计

上节主要讲述了dBASE III实用程序设计的基本方法，而在实施这种设计思想时，菜单方式是经常使用的，因为实用程序主要是为了使用，而用户很多是不懂计算机的，不可能要求他们都会打入dBASE III的命令，如果利用菜单方式，即将程序的功能全部在屏幕上显示，并将每一功能对应一个数字或字母，使用户象点菜一样，按某一键即可完成某一功能。菜单分大菜单和小菜单，大菜单即为主模块（或主控模块），小菜单即为子程序（子模块）。菜单程序的一般结构框图如图1.1所示。

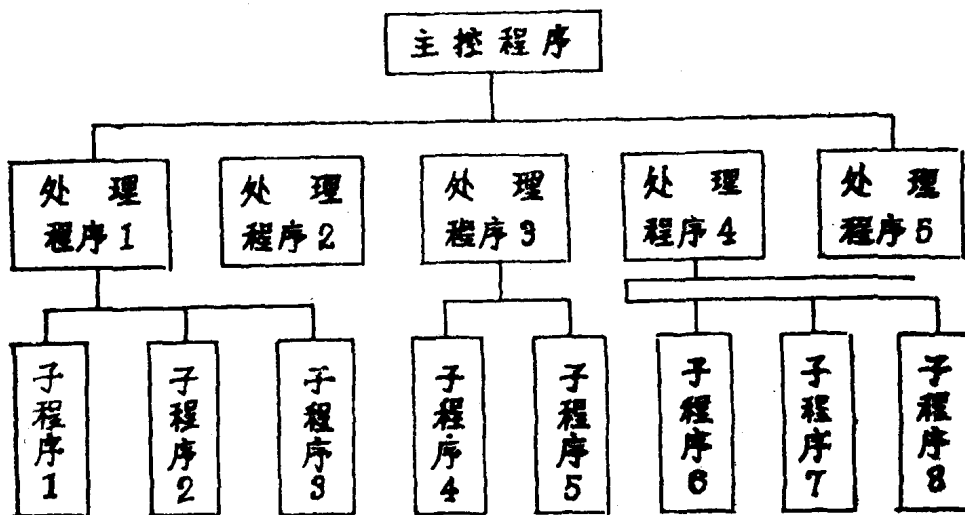


图1.1 菜单程序逻辑结构框图

在一个比较完善的微机管理系统中，往往包含了数十个，甚至上百个菜单程序，因此，菜单程序如何设计，对系统的功能起着举足轻重的作用。众所周知，dBASE III命令功能很强，往往一条命令就可以独立完成一类复杂的操作，例如数据处理中常用的分类、排序、检索、显示、打印报表等操作，都可以由一条命令完成，如果再加上范围、条件、

函数等短语，其功能就更强了。因此，操作使用人员几乎可以不必考虑操作功能如何实现，而把主要精力放在菜单程序设计上，只要把一个个菜单设计好，整个程序设计的问题也就迎刃而解了。

dBASE III的菜单程序通常有如下特点：

(1) 多分支选择结构。“菜单”说到底是一个多分支程序，即程序执行到此先暂停，等待使用者决定下边执行哪个分支程序，而后继续执行。dBASE III的条件转移命令有两个，IF和CASE。IF命令要求ENDIF与IF成对使用，一般在分支较少时可以使用，而在分支较多时，使用IF远不如使用CASE编程简洁，在菜单程序中，大多使用CASE命令。

(2) 单入口单出口结构。尽管菜单程序是多分支的，但其入口和出口是唯一的，它是由单支发散为多支，再由多支汇总为单支，程序在DO CASE后发散，在ENDCASE时汇总。通常把“一道菜单”程序编为一个独立的“子命令文件”，需要时调用，执行某分支后由唯一的RETURN返回主程序。

(3) 采取了“容错”设施。实践表明，在微机系统运行中，绝大多数错误来自于操作失误。为了使微机系统在操作使用人员对“菜单”提问回答错误时仍能继续运行，并给操作人员以错误提示信息和改正错误的机会，而不至“锁死”或停止运行。在菜单程序中，一般都含有循环程序，在选择错误时，用LOOP命令返回程序开关，再显示该“菜单”，要求操作人员重新选择。

按选择的内容菜单程序编写可分为五类，它们是：

(1) 选择文件菜单。这里所指的文件，主要是指dBASE的命令文件（以·CMD或·PRG为扩展名），也可以是数据库文件（以·DBF为扩展名），或报表（以·FRM为扩展名）等其他文件。通过选择相应的文件完成相应的功能。

(2) 选择记录菜单。这种选择的实质是划定选择条件的上下限，使以后的操作只对满足该条件的记录起作用，这种菜单程序也比较简单。

(3) 选择字段菜单。这种选择方式的实质是在数据库横的方向对字段范围作出选择。在数据库应用中，常常希望选择字段表，例如打印报表时，常常需要仅把我们关心的数个字段内容打印出来，因此必须在命令中给出字段名；又如在检索或统计时，常常要以字段名作为条件，例如查询职称是高级工程师，家住北京东城区的，年龄在45岁至50岁之间的教师，其dBASE III的命令条件为：FOR职称 = “高级工程师” · AND · 家庭住址 = “北京东城区” · AND · (年龄 >= 45 · AND · 年龄 <= 50)，这里的“职称”、家庭住址、年龄等字段名绝不能输错，若输入错，计算机将无法查找。但是要求所有操作人员把每个数据库的字段名都记熟，显然是不可能的，而且每次都输入字段名也很麻烦。为此在进行系统设计时，在凡是需要对字段名作出选择的地方，都编写选择字段菜单，可以大大提高工作效率和系统运行的可靠性。

下面某单位职工基本情况菜单程序的实例就是一种比较典型的选择字段菜单方式。

```
A>TYPE ZDS.PRG
      PUBL NN, SS TT
      SET TALK OFF
```

```

AA = . T .
DO WHILE AA
NN = " "
SS = " "
CLEAR
@ 1, 1 SAY " "
TEXT
    1. 编号  2. 姓名  3. 单位  4. 性别
    5. 民族  6. 出生年月  7. 党团  8. 学历
    9. 毕业学校  10. 外语水平  11. 工作时间
    12. 工作地点  13. 业务专长  14. 职称
    15. 行政职务  16. 基本工资  17. 备注
ENDTEXT
BB = . T .
DO WHILE BB
CC = 0
@ 10, 10 SAY "请输入字段名代码: " GET CC
READ
DO CASE
CASE CC = 1
    TT = "编号"
CASE CC = 2
    TT = "姓名"
CASE CC = 3
    TT = "单位"
CASE CC = 4
    TT = "性别"
CASE CC = 5
    TT = "民族"
CASE CC = 6
    TT = "出生年月"
CASE CC = 7
    TT = "党团"
CASE CC = 8
    TT = "学历"
CASE CC = 9
    TT = "毕业学校"
CASE CC = 10

```

```

        TT = “外语水平”
CASE CC=11
        TT = “工作时间”
CASE CC=12
        TT = “工作地点”
CASE CC=13
        TT = “业务专长”
CASE CC=14
        TT = “职称”
CASE CC=15
        TT = “行政职务”
CASE CC=16
        TT = “基本工资”
CASE CC=17
        TT = “备注”
    OTHERWISE
    @11, 10 SAY “按键错, 请重新选择!”
    LOOP
    ENDCASE CC
    @12, 0 SAY “ ”
    @13, 15 SAY TT
    DD = “ ”
    @13, 25 SAY “请输入限制条件: ” GET DD
    READ
    IF AT( “$”, DD) <> 0
    MA = TRIM( DD ) + TT
    ELSE
    MA = TT + TRIM( DD )
    ENDIF DD
    NN = NN + MA
    @14, 0 SAY NN
    @15, 0 SAY “ ”
    @15, 30 SAY “还有其他条件吗(Y/N)? ”
    WAIT TO EE
    @15, 20 SAY “ ”
    @16, 0 SAY “ ”
    IF UPPE( EE ) = “Y”
    NN = NN + “ . AND . ”

```

```

        SS = SS + TT + " , "
    ELSE
        SS = SS + TT
        BB = . F .
    ENDIF EE
ENDDO BB
AA = . F .
ENDDO AA
RETURN

```

在程序中TT为选中的字段名，DD为限制域，NN为条件，SS为字段名表。该程序有两层循环，第一层循环变量为AA，其作用是当操作失误，即键入了“菜单”中没有的数字时，重复显示该“菜单”，让操作人员重新输入正确的数字。第二层循环变量是BB，用于输入复合条件，并将各条件用·AND·运算符连接起来，作为查询或统计的条件，存于变量NN之中，同时将字段名用“，”号间隔开，作为选择显示或打印的字段名表，存于SS中。

(4) 应用宏代换函数方法

在菜单的选择项较多，即分支较多的情况应用宏代换的方法编写菜单程序较方便。

在分支程序命名时，把某一层次的程序，即某一菜单下的程序，用同一前缀和不同后缀来表示，后缀即选用菜单上供选择的功能序号，也就是说，把文件名分为两部分，前面部分相同，而序号不同，用户选择不同的功能后，利用宏代换把前缀与后缀组合成一完整的程序名，再执行之，这样便可以较短的程序实现分支转移。例如一个简单工资管理程序的几个子命令文件分别命名为PROG 1，PROG 2，……PROGn等后，即可把程序改写成下面的形式：

```

clear all
clear
a = 1
@ 0, 15 say "*****"
@ 1, 15 say "*"
@ 2, 15 say "*简单财务工资管理系统主控程序*"
@ 3, 15 say "*"
@ 4, 15 say "*****"
set console off
set talk off
clear
use gzdbf
set console on
text

```

```

        × × × × × × × × × × × × × × × × × × ×
×   1. 修改固定项   2. 修改活动项   ×
×   3. 打印工资总表 4. 打印工资条   ×
×   5. 打印汇总表   6. 查询信息     ×
×   7. 退出                               ×
        × × × × × × × × × × × × × × × × × × ×
endtext
input "请您输入(1—7)选择功能:" get a rang 1, 7
read
bx="PROG"+str(a, 1, 0)
do & bx
return

```

这种编写方法清晰简明，虽然由于使用序号命令降低了一些可读性，但用前缀标明了程序的层次。

(5) 建立参数数据库方法

利用这种方法编制菜单程序，是为菜单选择建立一个参数数据库文件。该文件有两个字段，一个是参数，如分支程序名，范围，查询字段名等；一个是菜单选择中的功能代码。在用户选定功能后，以其选择的功能代码作为关键字，利用LOCATE等命令在这个参数表文件中即可找到对应的参数。下面这个菜单程序就是使用这种方法的一例。

```

A>TYPE EXAMPLE.PRG
SET TALK OFF
DO WHILE .T.
CLEAR
@ 1, 10 SAY "*****"
@ 2, 10 SAY "* 某单位人事资料管理操作内容 *"
@ 3, 10 SAY "* 1 ……显示      2 ……查询   *"
@ 4, 10 SAY "* 3 ……统计汇总  4 ……编辑修改*"
@ 5, 10 SAY "* 5 ……打印报表    6 ……退出   *"
@ 6, 10 SAY "*                               *"
@ 7, 10 SAY "*****"
A=""
INPUT "请选择(1—6)" GET A
READ
IF A = 6
    RETURN
ENDIF
USE <参数库名>
LOCATE FOR XH=A

```

```

IF EOF ( )
    ? "选择错误, 请重新输入"
ELSE
    PP = CS
    DO & PP
ENDIF
ENDDO

```

其中〈参数库名〉中的记录为:

```

USE    <参数据名>
LIST   OFF
      XH   CS
      1   XS
      2   CA
      3   DJ
      4   BJ
      5   DY

```

其中XS·PRG是显示子模块, CA·PRG是查询子模块, DJ·PRG为统计汇总子模块, BJ·PRG为编辑功能子模块, DY·PRG为打印功能子模块。

1.2.1 主菜单程序的设计

主菜单程序又称主控模块,它是管理系统的控制核心,它要调用和运行子程序,要为整个程序建立工作环境,在返回圆点提示符之前还要关闭各类文件。管理系统运行时,首先进入主菜单程序,主菜单程序应显示出系统所具备的各项功能名称,给出必要的提示信息。每一种功能子程序有其自己的编号和名称,操作者根据自己的需要键入相应的功能编号,使其控制转向功能号所对应的子程序,执行完后返回主菜单程序。

主菜单程序应该包括下面的基本内容:

1. 设置工作环境

首先要用CLEAR获得一个清晰的初始状态,再逐个设置所需要的SET状态。例如,控制台响铃,命令的对话状态,目录路径,小数位数及功能键的设置等。

2. 初始化内存变量

要定义整个程序中使用的内存变量,给出它们的初始值。

3. 设计主菜单

为了保证单入口单出口的程序结构,一般都采用循环来设计主控程序。在可控程序中通常必须有如下四种类型的语句:

(1) DO WHILE<条件>语句。一开始应确保条件为“真”,以保证执行主控程序时能进入主菜单。通常选用DO WHILE·T·的形式,而在“退出”选择项下使用EXIT退出主控程序。

(2) ? , TEXT.....ENDTEXT或@语句。用以显示主控菜单的各个选择项。

(3) 人机交互性语句。例如, ACCEPT, INPUT, WAIT等。用以接受用户键入的选择。若各选择项的编号为数字, 则使用INPUT较为方便; 若为字母, 则应使用ACCEPT。

(4) DO CASE或IF语句。用以根据不同的选择调用不同的子模块。在选择项较多时, 使用DO CASE语句最为方便。

4. 程序结束前的内务处理

在返回圆点提示符之前, 应恢复SET设置的标准默认状态, 关闭各类文件。注意, RETURN语句并不关闭文件。除此之外还应释放全局性的内存变量。

菜单的编制方法有很多种, 常用的一种方法是利用“?”和“CASE”语句相结合, 用?在屏幕上显示各种功能及其对应选择键, 然后用CASE语句来分别执行对应功能。试看下面几个实例:

(1) 一个简单工资管理程序的主菜单程序的设计:

```
clear all
clear
@ 0, 15 say "*****"
@ 1, 15 say " * "
@ 2, 15 say " * 简单财务工资管理系统主控程序 * "
@ 3, 15 say " * "
@ 4, 15 say "*****"
set console off
set talk off
do while .t.
clear
use gzdbf
set console on
text
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    x  1. 修改固定项    2. 修改活动项  x
    x  3. 打印工资总表  4. 打印工资条  x
    x  5. 打印汇总表    6. 查询信息    x
    x  7. 退出          x
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
endtext
input "请您输入(1—7)选择功能:" to gs
do case
    case gs = 1
        do chg1
    case gs = 2
```

```

do chg 2
case gs = 3
do print 1
case gs = 4
do print 2
case gs = 5
do print 3
case gs = 6
do cax
case gs = 7
clear
@ 4, 20 say "欢迎您再次使用, 谢谢!!"
exit
otherwise
ondcase
enddo
return

```

(2) 通用人事档案管理系统的主菜单程序的设计:

* 主菜单程序 CONTROL.PRG

```

SET DEVICE TO SCREEN
SET PRINT OFF
SET TALK OFF
USE <人事档案库>
DO WHILE .T.
CLEAR
? "*****"
? " *           通用 *"
? " *           微机人事档案管理系统 *"
? " *           = = = = = *"
? " *           中 国 北 京 *"
? " * *"
? " * *"
? " * *"
? "*****"
? "           为您服务           "
? "           = = = =           "
WAIT "敲任一健开始"
CLEAR

```



```

? "          功能选择          "
? "          1 -----人事档案材料的编辑          "
? "          2 -----人事档案材料的显示          "
? "          3 -----人事档案材料的查询          "
? "          4 -----人事档案材料的统计          "
? "          5 -----人事档案材料的报表输出          "
? "          0 -----退出          "

```

```
INPUT "请按 0—5 选择功能" TO TT
```

```
DO CASE
```

```
  CASE TT = 1
```

```
    DO A:BJ
```

```
  CASE TT = 2
```

```
    DO A:XS
```

```
  CASE TT = 3
```

```
    DO A:CX
```

```
  CASE TT = 4
```

```
    DO A:DJ
```

```
  CASE TT = 5
```

```
    DO A:DY
```

```
  CASE TT = 6
```

```
    CLEAR
```

```
USE
```

```
SET TALK ON
```

```
RETURN
```

```
ENDCASE
```

```
ENDDO
```

(3) 科研仪器及固定资产管理系统主菜单程序的设计:

```
set talk off
```

```
clear all
```

```
use <仪器库>
```

```
do while .t.
```

```
clear
```

```
@ 3, 10 say "*****"
```

```
@ 4, 10 say "*  ××科研仪器管理系统××  *"
```

```
@ 5, 10 say "*.....*"
```

```
@ 6, 10 say "* 1 ——追加, 插入, 编辑, 删除 *"
```

```
@ 7, 10 say "* 2 ——数据查询, 统计 *"
```

```
@ 8, 10 say "* 3 ——数据显示 *"
```