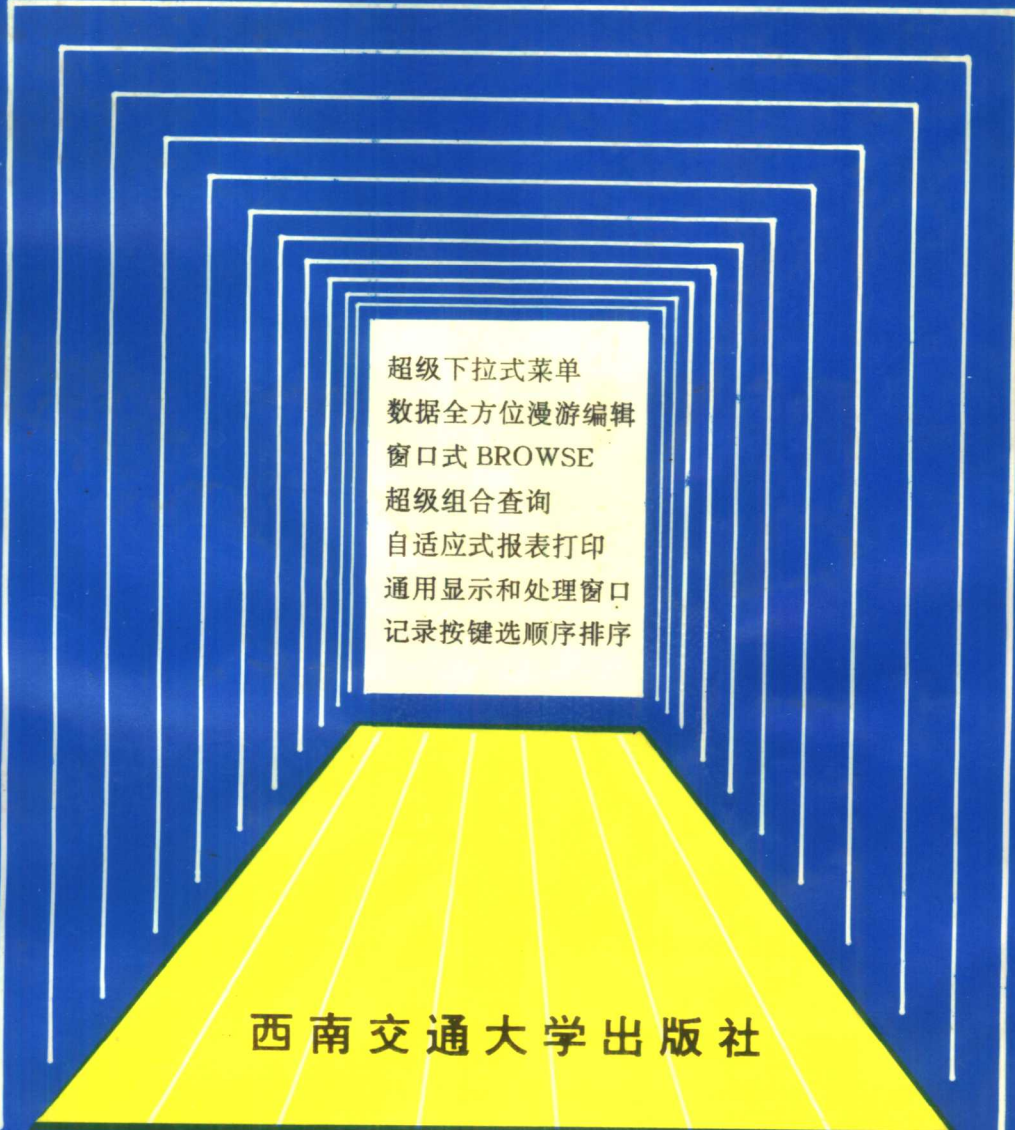


FoxBASE+ 及 FoxPro 实用编程新技法

张长虹 张长梅 编著



超级下拉式菜单
数据全方位漫游编辑
窗口式 BROWSE
超级组合查询
自适应式报表打印
通用显示和处理窗口
记录按键选顺序排序

西南交通大学出版社

FoxBASE⁺及 FoxPro

实用编程新技法

张长虹 张长梅

西南交通大学出版社

(川)新登字 018 号

内 容 提 要

本书着重讨论了目前国内广泛使用的关系型数据库管理系统 FoxBASE⁺ 及 FoxPro 的结构化程序设计方法和技巧;纠正了以往教程中一些以讹传讹的错误概念;深入讲解了一些重要的实用知识和大量最新通用的程序模块(如超级下拉式菜单、窗口式 BROWSE、数据全方位漫游编辑、超级组合查询、记录按键选顺序排序、通用显示及处理窗口、文本文件和数据库阅读模块、自适应式报表打印……)。

本书的主要特色在于新颖、通用、完整,是 FoxBASE⁺ 程序设计的最新成果,所有程序模块均能直接引用,可缩短程序开发周期,设计出具有良好用户界面的应用系统。

本书适合于从事计算机数据管理、应用及软件开发人员和大中专院校师生阅读。同时,对应用其它数据库管理系统的程序员也会有帮助和启迪。

FoxBASE⁺ 及 FoxPro 实用编程新技法

张长红 张长德

西南交通大学出版社出版发行

(成都·九里堤)

新华书店经销

郫县犀浦印刷厂印刷

开本: 787×1092 1/16 印张: 18

字数: 460 千字 印数: 5001—10000 册

1994 年 8 月第 1 版 1995 年 8 月第 2 次印刷

ISBN 7-81022-712-2/T·125

定价: 16.20 元

前 言

*FoxBASE+*和*FoxPro*数据库管理系统目前已广泛应用于管理信息系统的开发中。它们不但与*dBASE III plus*和*dBASEN*兼容,运行速度要快得多,增加了百数十条命令和函数,在性能上大大优于*dBASE III plus*和*dBASEN*,使我们能更方便灵活地设计应用程序。

本书的例子全部在286、386、486微机和汉字*FoxBASE+2.10*、*FoxPro2.5*环境下调试通过,所有程序均能直接引用或套用。本书的例子由浅入深,简明实用,使读者能够边学习、边上机对照实践,从而快速掌握*FoxBASE+*和*FoxPro*数据库管理系统,成为具有综合应用设计能力的程序员。

需要特别指出的是,书中列举的程序都是作者几年来实践的结晶,均为首次公开发表。程序设计立意新颖,通用性强,简明实用,对从事计算机管理、应用及软件开发人员都会有启迪和帮助。

本书的第一章深入细致地讲解了编程中常用命令、函数的使用方法和应用技巧;第二章主要介绍了很多新颖实用的程序模块,它们涉及了管理信息系统中的大部分内容,考虑到初学者,大部分模块都给出了单独调用示范程序,以便于读者尽快理解其中的设计技巧,提高学习兴趣。第三章是一个综合应用实例:RB工资管理系统V3.0(设计风格与以往的工资系统完全不同),因为我们每个人都对工资这一问题比较清楚,所以用工资作为例子便于读者理解程序设计技巧;同时,工资管理系统包括了管理信息系统的基本任务,有利于提高读者的综合程序设计能力。

由于水平有限,书中难免存在缺陷,恳请读者不吝赐教。

在本书的编写过程中,作者得到了马建华、徐武、盛强、贾效虎等同志的指导和帮助,在此一并致谢!

编 者

1994年5月

目 录

第一章 程序设计方法与技巧	1
1.1 结构化程序设计方法	1
一、程序质量的衡量标准	1
二、结构化程序设计的基础	2
三、结构化程序设计的方法	4
1.2 构造运行环境	6
一、加快系统的启动	6
二、WPS 在 Fox 下的直接调用	6
三、2.13H 特显及打印功能的共享	8
四、提高应用程序运行速度的技巧	10
五、在 640KB 内存、汉字系统下运行 FoxPro	13
1.3 灵活运用宏替换函数 &	14
一、利用 & 函数替换文件名	14
二、利用 & 函数替换 Fox 命令	14
三、利用 & 函数减少表达式的重复书写	15
四、利用 & 函数替换字段名	15
五、利用 & 函数简化菜单设计	16
六、利用 & 函数计算表达式的值	16
七、利用 & 函数构造数组	17
1.4 READKEY()函数的应用技巧	17
一、基本知识	17
二、记录编辑翻页技巧	18
三、历史编辑管理模块的实现	19
1.5 INKEY()函数的应用技巧	21
一、基本知识	21
二、INKEY()码的获取	22
三、系统的自动退出	23

四、以打字机敲打方式输出信息	23
五、字幕的平滑移动	25
六、宽行滚动显示模块	26
1.6 灵活运用 KEYBOARD 命令	29
一、自动演示技术	29
二、自动创建数据库文件	30
三、简化程序设计,减少冗余	30
1.7 数组应用技巧	31
一、数组的基本知识	31
二、新型工资面额张数计算方法	32
三、数组与库记录之间的数据交换	34
四、数组元素的删除、增加与排序	35
1.8 GETENV()函数的应用技巧	38
一、DOS 环境变量的概念	38
二、DOS 命令模块的实现	39
三、提供一种加密方法	39
1.9 公用子程序的设计技巧	40
1.10 设置错误和键盘陷阱	41
一、错误陷阱(ON ERROR)	41
二、键盘陷阱(ON ESCAPE)	43
三、键盘陷阱(ON KEY)	45
1.11 应用 TYPE()增强程序的容错率和通用性	48
1.12 发挥 TRANSFORM()函数的作用	49
1.13 SELECT 与 SELECT()	50
1.14 学会运用现场文件	51
1.15 其 它	52
一、自定义三角函数	52
二、随机函数的实现	53
三、系统状态信息显示模块	55
四、计算器模块	56
五、过程组合器 FOXBIND 和过程分解器 UNBIND	58

第二章 高级编程技术	62
2.1 翻选菜单的设计及运用技巧	62
一、翻选菜单的设计	62
二、设置记录操作范围模块	64
三、设置操作条件模块	65
四、记录置空模块	66
五、动态设置功能键	67
六、数组翻选菜单的设计	68
七、字段置空模块	70
八、字段清单选择模块	71
九、辅助菜单模块	72
2.2 库结构文件的应用技巧	76
一、利用库结构文件自动创建数据库	76
二、库结构文件的扩充及应用	78
三、获取库结构数据模块	81
2.3 系统的安全性设置	84
一、隐密限时输入模块	84
二、口令与权限维护管理模块	86
2.4 超级下拉式菜单模块的设计	90
一、菜单设计原理	90
二、菜单库的设计	91
三、菜单显示程序	91
四、在线帮助的实现	98
五、页面叠压式菜单	101
2.5 全方位漫游编辑数据模块	105
2.6 超级查询模块	109
2.7 自动生成屏幕格式文件	121
2.8 命令文件缩排处理	125
2.9 窗口内记录按照键选顺序排序	137
2.10 任意调整数据库字段顺序	147
2.11 文本文件和数据库阅读器	149

2.12 通用显示及处理窗口	158
2.13 窗口式 BROWSE	168
2.14 报表打印	179
一、转换打印数据模块	179
二、多栏(横向基准)打印模块	180
三、多栏(列向基准)打印模块	181
四、自适应式报表打印模块	183
五、新型报表打印模块	186
六、WPS 打印状态表的模拟	190
七、打印机状态检测与实时打印中断模块	195
八、让打印程序适应各种型号的打印机和汉字打印系统	196

第三章 RB 通用工资管理系统 V3.0

3.1 系统模块说明	200
一、系统总体结构	200
二、系统主控模块	200
三、输入修改模块	202
四、工资计算汇总	203
五、工资查询	205
六、人员调动	206
七、部门维护	207
八、工资项目维护	207
九、系统维护	208
十、打印输出	210
3.2 数据库结构说明	213
一、工资文件	213
二、工资汇总库(HZK.DBF)	214
三、部门代码库(BMDMK.DBF)	214
四、用工形式库(YGXSK.DBF)	215
3.3 源程序清单	215
附录一 书中使用的基本数据库	275
附录二 INKEY()键值表	276
附录三 SYS(2001)函数表	278
参考文献	279

第一章 程序设计方法与技巧

在本章中,我们主要讨论 FoxBASE+ 结构化程序设计方法及一些有关程序设计技巧方面的重要知识。在 FoxBASE+ 手册或教科书中,对此很少深入细致地讨论。通过本章的学习,将对提高 FoxBASE+ 的程序设计质量起着重要的作用。

1.1 结构化程序设计方法

对于一个要用计算机处理的较为复杂的问题,如一个企业的工资管理系统,如果我们把需要完成的工资录入、计算、查询、打印等各项任务都由一个应用程序来实现,那么必然出现如下情况:

1. 程序庞大,复杂,设计十分困难。
2. 程序冗余度大,浪费人力、物力和财力。
3. 程序结构混乱,阅读、调试都极为困难。
4. 不便于修改、维护,程序哪怕有一点小小的改动,都可能牵涉到整个程序,并且也能引入新的错误。

为了避免上述情况发生,我们采用结构化程序设计方法。

一、程序质量的衡量标准

程序设计技术是和计算机硬件的发展联系在一起的。早期的计算机由于其存储容量小,运算速度慢,成本高,加上许多人有这样的认识:程序是给机器执行的,不是供人阅读的。因此,那个时期程序设计者的注意力多放在要把程序代码编写得尽量“紧凑”,尽量减少运行时间和占用的空间上,评价程序优劣的标准主要看指令是否少,使用存储空间是否省。程序设计者有时使用一个小技巧,就常常使阅读者半天绕不过弯来。程序设计像手工工艺,程序也就成了一种工艺品。这样的程序易读性很差。

随着新的电子元件的出现,计算机硬件的功能和质量不断提高,而价格却大幅度下降,计算机的应用领域也越来越广泛,几乎涉及到了社会活动的各个领域,如工厂管理,银行事务,学校档案,图书管理,机票预订等等。这些系统的软件规模相当庞大,逻辑上很复杂,而且功能上需要不断更改和扩充,人们对软件质量的评价标准也逐渐变化。评价一个程序的好坏通常考虑以下几个方面:

1. 正确性

程序首先必须保证在执行后得到正确的结果。在计算机硬件环境正常的情况下,计算机出错的主要原因是软件的不完善,即程序设计有错误。程序不能得出正确的结果则毫无价值,因此,保证程序的正确性是第一位的。

2. 结构性

结构是指程序的组成结构或控制结构，结构性要求程序的组成结构中的成分少而精，各个组成结构独立性强。

3. 高效率

高效率是指程序运行时间要短和占用存储空间要少两个方面，即要求程序的时间效率与空间效率都高。这两者之间是互相制约的，不可兼得。一个好的程序应根据所解决问题的实际要求和条件来决定两者间的最佳配合。

4. 易读性

编好的程序使用一段时间之后可能会发现一些漏洞，也可能需要增加一些新的功能，因此对程序要进行维护，而且在程序的开发过程中也要反复地修改程序，故而只有易读、易懂、有清晰结构的程序才容易维护，才容易证明它的正确性。这是衡量程序设计好差的一个重要标准。

5. 适应性

适应性包含两方面的含义：其一是指程序适应范围的广泛性；其二是指程序对环境和需求变化的适应程度。当环境或条件发生变化时，程序运行不易被破坏，在非正常情况下应有应变能力和容错能力，使程序运行时不会产生中断、死锁等现象。因此，在程序研制时要充分考虑到各种可能的意外情况。

6. 用户友好性

程序设计出来以后是要交给用户使用的，设计程度时应该尽量从用户角度考虑，能让计算机自动完成的事情就不要让用户去做，让用户易学、易用，感觉亲切、友好。用户界面要符合自然，对每一步需要由用户干预的工作都尽量给出简洁而明确的提示。比如一些好的程序常设有帮助功能、热键功能、菜单技术、窗口技术等等，这都是为了提高用户友好性。

程序质量的标准既是评价他人程序的标准，也是我们自己设计程序时的准则。

二、结构化程序设计的基础

结构化程序设计是为了使程序具有合理的结构，以便保证和验证其正确性而规定的一套如何进行程序设计的准则。它规定由下面三种基本控制结构（顺序、分支、循环）组成：

1. 顺序结构

如图 1—1 所示，虚线框内是一个顺序结构。顺序结构的程序段被执行时，机器按照程序段中语句出现的先后顺序逐行依次往下执行。顺序结构是最简单的一种基本结构。

2. 选择结构

或称分支结构，如图 1—2 所示，虚线框内是一个选择结构。此结构是根据给定的条件，经过判断以后，在两条可能的路径中选择一条路径执行。注意，无论条件 P 是否成立，只能执行 A 操作或 B 操作之一，A 和 B 可以有一个为空，即不执行任何操作，如图 1—3 所示。

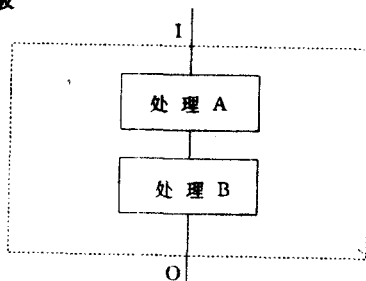


图 1-1

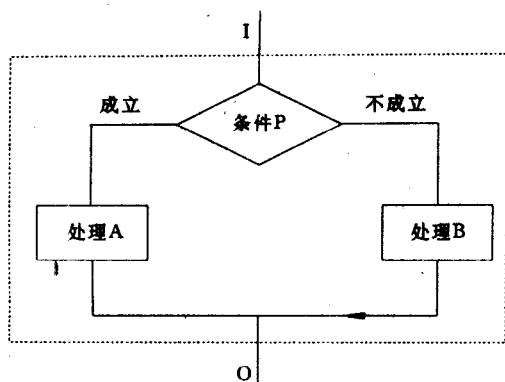


图 1—2

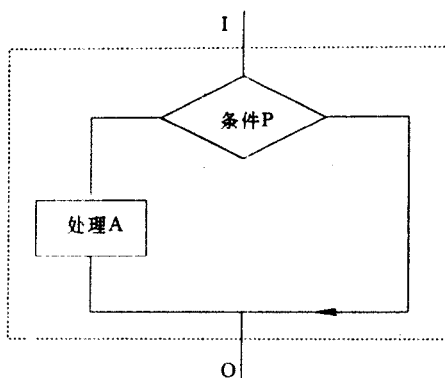


图 1—3

在 FoxBASE⁺ 中，选择结构由 IF…ELSE…ENDIF 来完成。

3. 循环结构

又称为重复结构，见图 1—4。它的功能是当给定的条件 P 成立时，执行 A 操作，执行完 A 后，再次判断条件 P 是否成立，如果仍然成立，再执行操作，……，直到某次条件 P 不成立为止，此时不执行 A 操作，而从出口 O 脱离循环。

FoxBASE⁺ 中，由 DO WHILE…ENDDO 语句实现循环结构。

以上三种基本结构，有以下共同特点：

第一，只有一个入口和出口，在入口和出口之间构成了控制结构。

第二，结构内的每一部分都有机会被执行到。也就是说，对每一个操作框来说，都应当有一条从入口到出口的路径通过它。

第三，结构内不存在“死循环”。如图 1—5 就是一个死循环。

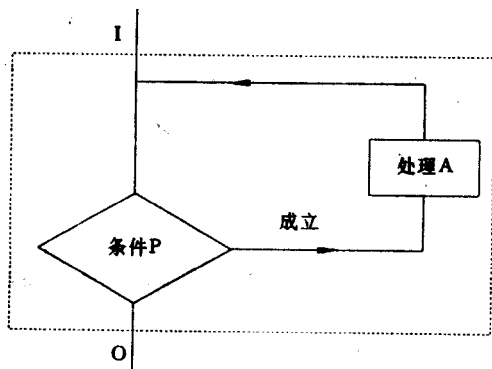


图 1—4

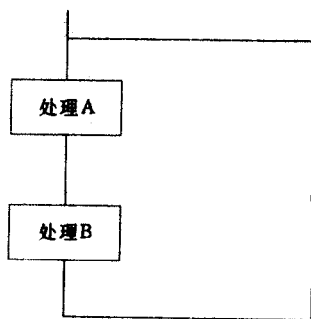


图 1—5

其实，基本结构并不一定只限于上面三种，只要具有上述三个特点的都可以作为基本结构。图 1—6 所示的是一个多分支选择结构，它根据给定表达式的值决定执行哪一个处理。可以认为此结构是由三种基本结构派生出来的。在 FoxBASE⁺ 中，多分支选择结构由 DO CASE…ENDCASE 来实现。

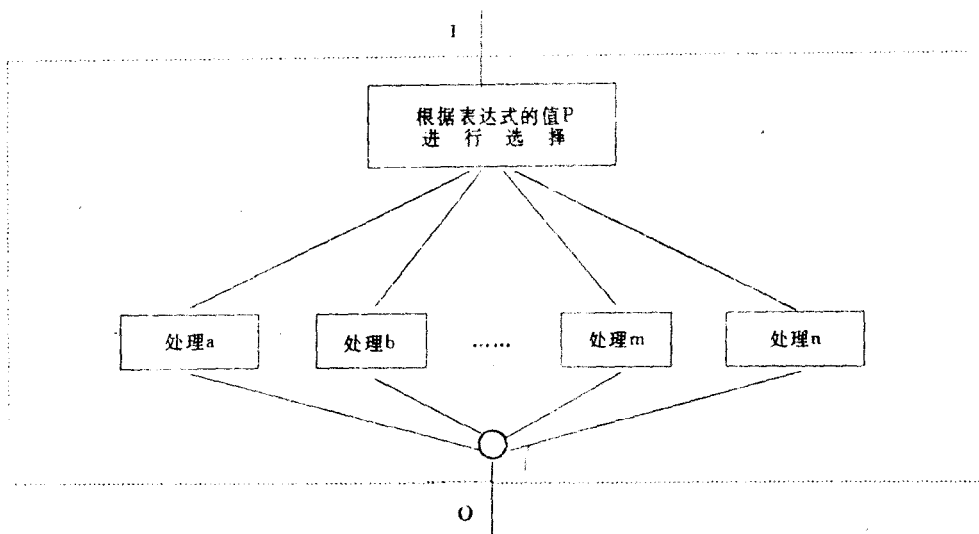


图 1-6

结构化程序设计的优点是：

(1) 单入口、单出口的结构具有相对独立性，使程序的编写和调试都十分容易。

(2) 按照单入口、单出口的原则使其方便地形成嵌套结构（即三种控制结构可以自身嵌套或互相嵌套使用）而不会破坏控制结构。

(3) 单入口、单出口的结构使程序执行时的控制流程与程序的静态描述相对应，利用静态描述的方法就可以正确地理解程序的动作。

(4) 能以控制结构为单元，从上到下顺序地阅读程序。与非结构化程序相比，结构化程序设计中不需要使用语句标号，没有无条件转移 GOTO 语句（在 FoxBASE+ 中，GOTO 是移动数据库记录指针的命令）。按照结构程序设计的观点，如果使用 GOTO 语句，将破坏控制结构单入口、单出口的原则，破坏执行程序的动态过程与程序的静态描述的一致性，程序就难以阅读和理解，容易出错并且难以检查。FoxBASE+ 避免了这种情况的发生，FoxBASE+ 的所有语句都不带标号，没有为无条件转移提供条件，FoxBASE+ 是一种非常适合于结构化程序设计的程序语言。

三、结构化程序设计的方法

程序设计是一个复杂的逻辑思维过程。结构化程序设计方法要求程序设计者要按照统一的步骤和方法去进行设计，它的基本内容包括：

1. 自顶向下，逐层分解

设计者拿到一个项目后，首先应从整体入手，分析这个项目要完成的主要任务是什么，充分理解项目的要求。可以把项目看成一个系统，将其划分为若干个功能各异的子系统，如果子系统仍很复杂，就把子系统进一步分解为子子系统……就这样一层层地分解下去，直到最后分解出的子系统足够清楚、简单为止。对分解出的每一层子系统，可以用文字写出必要的功能说明，把所有的说明按秩序组合在一起，就形成了整个系统的系统说明书。系统说明书要尽可能准确无误，因为客观存在既是设计者和使用者双方达成的技术协议书，也是后续工

作的基础和依据。

2. 模块化

模块化程序设计就是把一个程序系统按功能划分为若干个相对独立的、单独完成一项具体功能的程序段，这种程序段就叫做模块或功能模块。一个好的模块应有以下特点：

(1) 功能单一、明确

一个模块只完成一项具体的功能，我们可以对要完成多个功能的模块进一步划分，使每一个模块与一个具体的功能相对应。

(2) 接口简单、清楚

各模块之间（或者说一个模块与外界之间）的相互联系或交换的信息叫做“接口”。简单的接口便于记忆、程序调用。

(3) 独立性强

模块要具有独立性，必须做到：

- 模块只通过接口与外界联系。模块内的信息对外界来说是隐蔽的，不需要了解模块内部的细节。

- 模块被调用后，不能产生副作用，即不能把系统中不应该改变的变量值改变了。

- 模块可以单独命名、单独编写和单独调试。

- 避免交叉调用。

把一个复杂的系统划分为若干个功能模块，只需要对各模块的功能和接口提出明确的要求就可以分给若干人去完成，当各个模块分别调试好之后，再组合成一个完整的系统。由于模块的“独立性”，我们不必担心模块内外的互相干扰，因此，这样的系统不仅容易研制，而且容易维护、修改和扩充功能。模块化的设计方法体现了“化整为零，分而治之”的原则。在 FoxBASE⁺ 中，模块对应于一个子程序、一个过程或一个用户自定义函数。

模块划分的结果是得到一个层次结构图，它与系统结构图相同，模块间的调用关系也是纵向的。

3. 逐步求精

一个模块的功能，最初用一两句话就可以描述，成为用自然语言表达的抽象算法（伪码描述），然后对这些抽象的算法进一步分解和细化，如此下去，一直到最后的算法能够为计算机所理解为止，这种方法称为逐步求精，即最后的算法能用 FoxBASE⁺ 语言描述出来。

逐步求精最终可得详细的算法流程图及由其转化而得的程序。

可以看出，逐步求精是一种自顶向下的设计方法，但是值得注意的是，没有这样一种设计，要求先前的每一步骤都是正确的，而且是最优的，这显然是不可能达到的。对这样一个自顶向下的设计不能理解得太绝对化，原因是有时按某一方式精细化后，在以后的步骤中发现原来那样一种细化的设想并不好，或者对以后进一步求精不方便，或者有错误，或者算法不够有效，或者某些部分的算法可以合并等等。此时，必须由底向上对原来已决定的步骤进行修改。因此，对逐步求精过程正确的理解应为：它是一种为不断地由底向上的修正所补充的自顶向下的设计方法。

由于整个系统是由一些相对独立的功能模块所组成，在逐步求精过程中改变某些子问题

的解决策略，只等于改变了局部结构的内部算法，而不致于影响系统的全部结构。

1.2 构造运行环境

一、加快系统的启动

FoxBASE+ 在系统启动时，要显示一些信息：由空格和圆点组成的一个放大的 Foxbase、系列号和版本号及汉化单位。在运行应用程序时，这个显示过程显得很慢，屏幕不美观。下面是我们使用 PCTOOLS(V5.0 以下)去掉这些启动信息的操作步骤：

1. 去掉放大的 Foxbase

- (1) 选中文件 FOXPLUS. RSC
- (2) 按 E 键，查找 . . . (三个空格，三个圆点)
- (3) 按 E 键，输入 00 (十六进制)
- (4) 按 F5，将修改结果存盘
- (5) 按回车键返回主菜单

2. 去掉系列号

- (1) 选中文件 FOXPLUS. EXE
- (2) 按 F 键，查找 Serial
- (3) 按 G 键，指到第二个 Serial 字符串位置上
- (4) 按 E 键，输入 00
- (5) 按 F5，将修改结果存盘
- (6) 按回车键，返回主菜单

3. 去掉版本号及汉化单位信息

- (1) 选中文件 FOXPLUS. EXE
- (2) 按 F 键，查找 FoxBASE+2.1
- (3) 按 E 键，输入 00
- (4) 按 F5，将修改结果存盘
- (5) 按回车键，返回主菜单

FoxPro 的启动信息都在 FOXPRO. EXE 文件中，放大的 FoxPro 是由空格 20H 与图符 DFH 组成的，我们可以仿照上面的修改方案来屏蔽启动信息。更简单的方法是在命令行使用 -T 开关参数，即可达到关闭启动信息的目的。

二、WPS 在 Fox 下的直接调用

FoxBASE+ 提供了 MODIFY COMMAND 来建立与修改文本文件，然而 FoxBASE+ 的文本编辑器有很多不足之处，突出表现在它的编辑功能较差，特别是没有字块处理功能；另外，它最多能接纳 64 KB 的信息，由于运行在汉字系统下，使它一般只能编辑 8K 左右，用户使用时极为不便。所以，我们常常在 CONFIG. FX 文件中设置：

TEDIT=WS/CCED

WP=WS/CCED

这样，当我们在圆点提示符下执行 MODIFY COMMAND 命令时，系统自动使用 WS 或者

CCED 字处理软件来代替系统本身的文字编辑器。然而，当我们设置成 TEDIT=WPS，执行 MODIFY COMMAND 时，系统会报告错误信息：

```
program too big to fit in memory
```

即无法运行 WPS。如果我们每次要修改较长的命令文件都得退出 FoxBASE⁺，在运行 WPS 进行编辑修改后，再进入 FoxBASE⁺来调试，就会让人感到很麻烦。能不能不退出 FoxBASE⁺，在圆点提示符下直接运行 WPS 呢？

我们注意到，在 DOS 状态下，键入：

```
FoxVIEW/FoxCODE/FoxDOC/FoxGRAPH
```

能运行这些辅助工具；在 FoxBASE⁺圆点提示符下也只需直接键入：

```
FoxVIEW/FoxCODE/FoxDOC/FoxGRAPH
```

即能运行它们。这给我们一个启示：FoxBASE⁺可以调用 DOS 执行文件，在其内部必然有这些可执行文件的文件名表。使用 PCTOOLS(V5.0 以下)，我们修改了 FoxBASE⁺内部的文件名表，达到了在 FoxBASE⁺下直接运行 WPS 的目的。下面是我们的修改过程：

(1) 进入 PCTOOLS；

(2) 选中文件 FOXPLUS.EXE；

(3) 使用 F 命令，查找"FOXGRAPH"，过一会儿，屏幕显示：

```
Search argument found in relative sector 0000442 offset 0227
```

```
Press "E" to view/edit the sector or "G" to continue searching
```

选"E"，进入扇区修改状态，

输入：57(W)、50(P)、53(S)、2E(.)、45(E)、58(X)、45(E)、00(NUL)，替代原有字符，按"F5"存盘，

按回车键返回到 PCTOOLS 主菜单状态；

(4) 按 ESC 键返回；

(5) 选择 FOXPLUS.RSC；

(6) 使用 F 命令，查找"FOXGRAPH"

选"E"，

输入：57(W)、50(P)、53(S)、00(NUL)，

按"F5"存盘；

(7) 退出 PCTOOLS。

以后，我们进入 FoxBASE⁺后，在圆点提示符下直接输入 WPS，就进入高级文字处理系统 WPS 状态，可以随心所欲地编辑文本文件了。退出 WPS 后，系统又返回到 FoxBASE⁺的圆点状态。

需要注意的是：

第一，WPS.EXE 必须在 FoxBASE⁺主目录下，或者 DOS 路径说明中包含 WPS 所在的路径。

第二，如果这个文件在修改以前被运行过或者用 MODIFY COMMAND 命令打开过，则被修改前的源程序仍然被保留在程序存储缓冲区中，如果要执行该文件，必须先执行命令 CLEAR PROGRAM，以清除驻留在程序存储缓冲区中的程序。否则，执行的程序是修改前的程序。如果该程序曾被伪编译，则要删除它。

第三,上面的修改是将图形软件包 FOXGRAPH 替换成 WPS。使用同样的方法,可以将其它的 FoxBASE+ 默认程序改成你想在圆点提示符下直接调用的程序,如 GWBASIC 等等。

在 FoxPro 中,仿照上面的办法修改 FOXPRO.EXE 文件中三处含有 FOXGRAPH 的地方即可。

三、2.13H 特显及打印功能的共享

CCDOS2.13H 汉字系统以其丰富的特显功能、灵活方便的打印功能而深受广大用户的喜爱,在 FoxBASE+ 中可以直接调用特殊显示功能进行画点、线、矩形、圆、显示 24×24 点阵汉字等等,它可以弥补 FoxBASE+ 不能直接绘图的缺憾,为绘制各种统计曲线、直方图以及漂亮的软件封面提供了方便(在 2.13I 版本中还有音乐功能),在其环境下编有大量的应用程序。一般常使用如下特显命令:

(1) 画线命令

```
@0,0 SAY CHR(14)+"L X 坐标, Y 坐标"]
```

(2) 画框命令

```
@0,0 SAY CHR(14)+"D X 坐标, Y 坐标 B 水平长度, 垂直长度"]
```

(3) 显示特殊字体字型

```
@0,0 SAY CHR(14)+"[_X 坐标|Y 坐标 字间距 * 字符色彩 @字型 内容"]
```

在使用特显功能时,须要注意:

- 不能用 ?/?? 命令引导功能符,必须用 @...SAY 命令才能完成相应功能。
- @ 后的行、列坐标不起定位作用,只识别功能符中的以屏幕图形点数取值的坐标 (X, Y)。在 VGA 显示器、使用 CV26 显示模块情况下, X 取值 0—639, Y 取值 0—479。
- 命令行尾必须加方括号 "]", 否则可能出错,甚至死机。
- 命令行的长度不得大于 160 个字符。

虽然 2.13H 有很多优点,但也存在不足之处,突出表现在 CV26.COM 模块设计中的缺陷,其数字、字符显示不美观,色彩设置不正确,在 FoxBASE+ 中使用光条菜单、弹出式菜单时产生无法消除的黑线。为了追求美观的屏幕效果,目前我们多采用金山 SPDOS 汉字系统,它以强大方便的图文编辑功能而广泛流传开来。为了取长补短,许多用户都在机器中同时安装了这两个系统,甚至有人专门为交替运行金山 DOS 和 2.13H 的方法纂文,能不能在金山汉字系统下直接运行含有 2.13H 特显和打印控制的程序呢?通过摸索,我们完全实现了这一目的。

2.13H 的特殊显示功能是利用了西文 BIOS 中断 INT 10H 的功能 09 来实现的。在装载 2.13 汉字系统时,CCCC.COM 将 INT 78H 指向西文 INT 10H,而用自己的 INT 10H 代替了西文 INT 10H。如果在装载其它汉字系统以前,将 INT 78H 指向 INT 10H,就为我们实现 2.13 的特显功能打下了基础。下面我们用 C 语言实现上述过程:

为了能显示汉字和利用其打印功能,需要运行以下程序:

```
/*      SET78.C      */  
#include <dos.h>  
void main(void)  
{
```

```

void interrupt ( *oldhandler)(void);
oldhandler=getvect(0x10);
setvect(0x78,oldhandler);
}

```

编译 SET78. C,生成 SET78. COM;

TCC -mt SET78. C

下面是使用 DEBUG. COM 建立 SET78. COM 的过程:

C>C:\DOS\DEBUG SET78. COM

File not found

-A100

XXXX:0100 B81035 MOV AX,3510

XXXX:0103 CD21 INT 21

XXXX:0105 8CC0 MOV AX,ES

XXXX:0107 8ED8 MOV DS,AX

XXXX:0109 89DA MOV DX,BX

XXXX:010B B87825 MOV AX,2578

XXXX:010E CD21 INT 21

XXXX:0110 CD20 INT 20

XXXX:0112

-RCX

CX 0000

:12

-W

Writing 0012 bytes

-Q

:

SET78

CD\213

FILE0A

显示字库驻留硬盘

INT10H

装入特显功能

FILE16B

调入读显示字库模块

FILE 24A ISFHK

调入读 24 点阵字库模块

ZF24 3

24 点阵字符字体选择

PRTA

启动打印驱动程序

CD\

:

注意:不要运行其它汉字系统的打印驱动程序。

这样,我们就完全不必修改原来已经编好的程序,而在金山或者其它汉字系统下继续使用 2.13H 的特显和打印功能了。