

汉字 FOXBASE⁺程序设计

(2.0~2.1 版)

张国焯 郑卫东

编著 郭盈发 审

王小华 李菊莲



西安电子科技大学出版社

501120



90501120

汉字 FOXBASE⁺程序设计

张国焯 郑建卫 王小华 李菊莲 编

郭盈发 审



陕西电子编辑部

* * * * *

汉字 FOXBASE⁺程序设计

张国焯 郑卫东 王小华 李菊莲 编著

* * * * *

西安电子科技大学出版社

陕西法学会印刷厂印刷

开本: 787× 1092 1/16 印张: 20.25 字数: 486 千字

1993 年 3 月第一版 1993 年 3 月第一次印刷

印数 1—5000 册

ISBN7—5606—0272—X / TP · 0096

定价: 12.00 元

前言

多用户汉字 FOXBASE⁺是根据美国 Fox Software 公司的 FOXBASE⁺2.00 关系型数据库管理系统汉化而成的。汉字 FOXBASE⁺与汉字 dBASEⅢ, dBASEⅢ plus 完全兼容,在功能上有许多重要的扩充。FOXBASE⁺比 dBASEⅢ plus 快 6.74 倍。汉字 FOXBASE⁺是继汉字 dBASEⅢ plus 之后推出的又一深受广大用户欢迎的数据库管理系统,它已成为开发管理信息系统等应用领域的有力工具。

本书共分十四章。第一章和第二章:介绍数据库的一般知识,汉字 FOXBASE⁺的特点及运行环境,汉字 FOXBASE⁺的基本概念。第三章:介绍汉字 FOXBASE⁺的函数。第四章:介绍汉字 FOXBASE⁺数据库文件结构的建立和数据录入的几种不同途径,以及数据库文件结构与数据的输出。第五章:详细介绍数据库的各种操作功能,索引文件的建立,以及记录的查询与统计汇总。第六章:主要介绍数据库操作的其它辅助命令。第七章至第九章重点介绍汉字 FOXBASE⁺的程序设计方法,是前面几章内容的综合应用。第七章介绍汉字 FOXBASE⁺的程序执行方式,具体叙述了结构化程序设计的基本概念,及程序的顺序,选择和循环三种基本的结构形式。同时,还举例说明了命令文件的建立和执行。第八章:介绍输入输出的设计方法,以便建立更为友好的用户界面。第九章介绍子程序的调用和过程文件的设计。第十章:介绍更为复杂的多重数据库文件的操作。阐述了工作区的基本概念,数据库文件之间联系的建立,以及它们之间数据的更新与连接。第十一章:介绍参数的设置和系统的配置,以改善系统的环境性能和开发软件的质量。第十二章:介绍汉字 FOXBASE⁺与其它软件的接口技术,实现汉字 FOXBASE⁺同其它语言之间的数据交换。通过数据交换,可用其它语言来弥补汉字 FOXBASE⁺在数值运算、作图等功能上的不足。第十三章:介绍汉字 FOXBASE⁺的编程技巧,以便更加有效地编出高质量的程序。第十四章:介绍多用户汉字 FOXBASE⁺的系统要求、特点和功能。

本书第一章至第三章由王小华编写,第四章、第五章由张国焯编写,第六章、第十章至第十四章由郑建卫编写,第七章至第九章由李菊莲编写。张国焯负责全书的编排和定稿工作。郭盈发负责全书的审阅工作。

在本书的编写过程中得到杭州电子工业学院领导、计算机系领导的关心和大力支持。叶又新教授、章鉴汀教授为本书提了许多宝贵的意见。徐兵为第六章、第十一章作了大量的整理工作。在此表示衷心的感谢!

由于编者水平有限,时间仓促,难免有错误和不妥之处,敬请读者批评指正。

编者

一九九〇年十月十五日

内 容 简 介

本书全面系统地介绍了汉字 FOXBASE+关系型数据库管理系统的特点、功能、程序设计、编程技巧、与其它软件的接口技术,以及参数设置,系统配置和多用户情况下的系统要求、特点与功能。

全书共分十四章，各章均有大量实例，章末附有习题，内容深入浅出，通俗易懂，便于自学。

本书可作为大专院校和各类计算机应用培训班的教材或教学参考书。也适合中等以上文化程度的读者自学,对于广大计算机应用开发技术人员也有着重要的参考价值。

目 录

第一章 汉字 FOXBASE⁺概述

1-1 计算机基础知识	(1)
1-2 数据库简介	(3)
1-3 汉字 FOXBASE ⁺ 数据库系统	(5)
1-4 汉字 FOXBASE ⁺ 特点	(13)
1-5 汉字 FOXBASE ⁺ 的运行环境	(17)
1-6 汉字 FOXBASE ⁺ 的运行	(18)
习题	(18)

第二章 汉字 FOXBASE⁺基础知识

2-1 符号的约定	(19)
2-2 汉字 FOXBASE ⁺ 的语法规则	(19)
2-3 全屏幕编辑及其光标控制	(24)
2-4 出错信息	(27)
习题	(28)

第三章 汉字 FOXBASE⁺函数

3-1 数值函数	(29)
3-2 日期和时间函数	(31)
3-3 字符串函数	(32)
3-4 类型转换函数	(35)
3-5 库操作函数	(37)
3-6 环境函数	(40)
3-7 杂类函数	(44)
习题	(45)

第四章 建立数据库

4-1 数据库文件的建立	(47)
4-1-1 数据库文件结构的建立—CREATE	(47)
4-1-2 数据库文件的打开和数据的录入—USE、APPEND	(52)
4-1-3 数据库文件的关闭—USE	(54)
4-2 数据库文件结构和数据的输出	(55)
4-2-1 分页列表输出—DISPLAY、STRUCTURE、DISPLAY	(55)
4-2-2 连续列表输出—LIST STRUCTURE、LIST	(59)
4-3 用其它方法建立数据库文件	(60)

4-3-1	数据库文件结构的建立—CREATE、FROM、 COPY STRUCTURE、MODIFY STRUCTURE	(61)
4-3-2	数据库文件数据的录入—APPEND FROM	(65)
4-3-3	由旧数据库文件来建立新数据库文件—COPY	(70)
	习题.....	(77)
第五章 数据库操作		
5-1	数据库文件记录的编辑和修改	(80)
5-1-1	记录指针及其操作—GO、SKIP	(80)
5-1-2	记录的编辑—EDIT	(82)
5-1-3	记录的插入—INSERT	(82)
5-1-4	记录的翻阅—BROWSE.....	(84)
5-1-5	记录的修改—CHANGE	(87)
5-1-6	记录的替换—REPLACE	(87)
5-2	数据库文件记录的删除和恢复	(89)
5-2-1	DELETE 命令	(89)
5-2-2	PACK 命令	(91)
5-2-3	ZAP 命令	(92)
5-2-4	RECALL 命令	(92)
5-3	索引文件	(93)
5-3-1	索引文件的建立—INDEX	(94)
5-3-2	索引文件的打开—USE	(95)
5-3-3	索引文件的重建—REINDEX	(97)
5-4	数据库文件记录的查询	(98)
5-4-1	记录的直接查找—LOCATE	(98)
5-4-2	记录的继续查找—CONTINUE	(99)
5-4-3	记录的索引查找—FIND、SEEK	(100)
5-5	统计汇总	(103)
5-5-1	统计记录数—COUNT	(103)
5-5-2	求平均值—AVERAGE	(104)
5-5-3	求和—SUM	(105)
5-5-4	求分类和—TOTAL	(105)
5-6	数据库文件的排序—SORT	(108)
	习题	(109)
第六章 数据库操作的其它辅助命令		
6-1	有关内存变量的操作	(112)

6-2 磁盘操作命令	(117)
6-3 其它辅助命令	(119)
习题	(121)
第七章 汉字 FOXBASE+ 程序设计(I)	
7-1 结构化程序设计基本思想	(122)
7-2 汉字 FOXBASE+ 命令文件的建立和执行	(123)
7-3 顺序执行程序设计	(125)
7-4 分支程序设计	(129)
7-4-1 两个分支的程序设计—IF...ELSE...ENDIF	(129)
7-4-2 多个分支的程序设计—DO CASE...ENDCASE	(132)
7-5 循环程序设计	(136)
7-5-1 循环语句—DO WHILE...LOOP...EXIT...ENDDO	(137)
7-5-2 单循环应用举例	(137)
7-5-3 多重循环及其应用	(140)
7-5-4 循环语句与其它分支语句的联合使用	(142)
习题	(142)
第八章 输入输出程序设计	
8-1 屏幕输入输出程序设计	(145)
8-1-1 屏幕格式设计命令	(145)
8-1-2 屏幕输入格式文件的建立与应用	(152)
8-2 标准的报表格式文件 .FRM	(154)
8-2-1 标准报表格式文件的建立和改 —CREATE REPORT、MODIFY REPORT	(156)
8-2-2 标准报表的生成及输出 —REPORT FROM	(157)
8-3 标准的标签格式文件 .LBL	(159)
8-3-1 标准标签格式文件的建立和修改 —CREATE LABEL、MODIFY LABEL	(159)
8-3-2 标准标签的生成及输出—LABEL FROM	(160)
8-4 报表格式程序设计	(160)
8-4-1 设计报表程序的一般方法	(160)
8-4-2 控制报表输出去向命令—SET DEVICE	(162)
8-4-3 控制打印机走纸命令—EJECT	(162)
习题	(163)
第九章 汉字 FOXBASE+ 程序设计(II)	

9-1 主程序	(164)
9-2 子程序	(167)
9-2-1 有关调用子程序的命令—DO、RETURN、RETRY	(167)
9-2-2 子程序调用的嵌套	(169)
9-3 过程和过程文件	(169)
9-3-1 过程文件编辑器—FOXBIND	(169)
9-3-2 使用过程文件	(170)
9-3-3 参数说明—PUBLIC、PRIVATE	(171)
9-3-4 参数传递—DO...WITH...PARAMETERS	(173)
9-4 ON 命令	(174)
习题	(176)

第十章 多重数据库文件操作

10-1 工作区的基本概念—USE、SELECT	(177)
10-2 数据库文件之间联系的建立和操作—SET RELATION	(181)
10-3 用一个数据库文件对另一个数据库文件的更新—UPDATE	(185)
10-4 两个数据库文件的连接—JOIN	(187)
习题	(190)

第十一章 参数设置和系统配置

11-1 参数设置命令—SET	(192)
11-2 系统状态显示命令—LIST / DISPLAY STATUS	(204)
11-3 汉字 FOXBASE ⁺ 的配置文件	(206)
习题	(210)

第十二章 汉字 FOXBASE⁺与其它软件的接口技术

12-1 用汉字 FOXBASE ⁺ 文本文件接口实现与其它软件的数据共享	(211)
12-2 其它软件直接存取数据库文件数据的方法	(221)
12-3 汉字 FOXBASE ⁺ 调用外部目标的接口 —RUN!、LOAD、CALL、RELEASE	(225)
习题	(227)

第十三章 汉字 FOXBASE⁺编程技巧

13-1 优化汉字 FOXBASE ⁺ 运行环境	(228)
13-2 常见问题及解决方法	(229)
13-3 汉字 FOXBASE ⁺ 编程技巧	(231)
习题	(246)

第十四章 多用户汉字 FOXBASE⁺

14-1 系统运行环境	(247)
-------------------	-------

14-2	多用户汉字 FOXBASE+ 基础知识	(248)
14-3	多用户对数据的共享与控制	(249)
14-4	加锁方法	(250)
14-5	加锁失败的测试和处理	(255)
14-6	多用户命令	(258)
14-7	多用户函数	(261)
14-8	多用户编程死锁的预防	(265)
14-9	常见的多用户出错信息	(268)
	习题	(269)

附录

- 附录一: ASCII 字符编码表
- 附录二: FOXBASE+ 函数集 (字母序)
- 附录三: FOXBASE+ 命令集 (字母序)
- 附录四: FOXBASE+ 错误信息
- 附录五: 图形字符代码表
- 参考文献

第十五章 汉字 FOXBASE+ 2.10 版

15-1	筛选索引	(295)
15-2	视图 (VIEW) 文件	(295)
15-3	菜单功能	(296)
15-4	其它	(299)
	习题	(301)

第十六章 汉字 FoxPro 关系数据库

16-1	FoxPro 的主要特点	(302)
16-2	FoxPro 新增加的命令	(310)
16-3	FoxPro 新增加的函数	(317)
16-4	FoxPro 新增加的系统内存变量	(322)

第一章 汉字 FOXBASE⁺概述

当今人类社会已进入信息化时代,计算机早已不仅仅是用于数值计算,而是更为广泛地应用于信息处理领域,因此,计算机在某种意义上已被人们称之为信息处理机。数据处理问题的特点是数据量大、类型多、结构复杂,同时,对数据的贮存、检索、分类、统计等处理的要求较高。为了适应这一需求,把数据从过去附属于程序的做法改变为数据与程序相对独立;对数据加以组织与管理,使之能为更多不同的程序所共享。这就是“数据库系统”的基本特点之一。

由于数据库技术的出现,数据处理的成本大大降低,效率越来越高,可靠性不断增加,从而推动了计算机应用的普及。

数据库系统有四个重要的组成部分,即经过组织,可供多方面使用的数据集合——数据库;支持数据贮存和数据操作的计算机系统;介于数据库和应用程序之间的数据库管理系统;提供给用户使用的各种数据库应用程序。我们这里介绍的是一个能在 DOS 和 XENIX 操作系统环境下运行的单用户或多用户 FOXBASE⁺数据库管理系统,以及如何建立数据库与编制 FOXBASE⁺应用程序的方法与技巧。

本章主要对汉字 FOXBASE⁺做一简单的介绍。

1-1 计算机基础知识

冯·诺依曼 (John Von Neumann) 于 1946 年提出了“存贮程序计算机”的设想,实现了真正的自动计算,这一设计思想对计算机的发展产生了巨大的影响。尽管近半个世纪以来,计算机的硬件制造技术以令人难以置信的高速度向前发展,其性能/价格比几乎以每四年一个数量级的速率变化,新的计算机产品令人眼花缭乱,目不暇接,但是,绝大部分产品仍采用“冯·诺依曼”结构。

当人们使用计算机时,就必须把计算机要完成的计算任务分解为计算机能执行的各种基本操作。也就是说,在使用计算机之前,要把求解的复杂问题按照机器能够识别的形式设计计算步骤,即编制程序的过程。程序的每一步骤称为一条指令。每条指令相当于对计算机下达的一个工作命令,它规定计算机应该进行哪一种基本操作,以及操作时所需要的数据。计算机的工作归结为执行程序的过程。我们要求机器能自动执行这些程序,就必须把程序预先存放到一个具有“记忆”功能的装置中保存起来。程序中每条指令通常是按一定顺序一条条存放的,计算机工作时,只要能知道程序中第一条指令放在什么地方,它就能按照顺序,取出每条指令加以识别,然后按照每条指令规定的功能执行相应的操作。这些都是由计算机中担任指挥功能的电子部件——控制器来完成的。整个运算任务实现的前提是在运算之前,必须把编制好的程序和所需要的数据送到有记忆功能的部件——存贮器中保存起来,然后由控制器控制程序的执行。这就是计算机自动连续工作的基础——存贮程序的原理。

图 1-1 是冯·诺依曼型计算机的基本组成原理图。

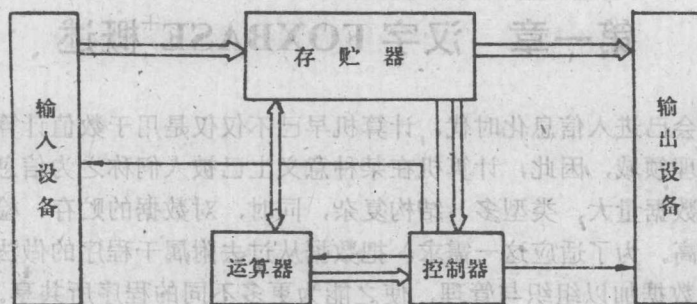


图 1-1 计算机的基本组成

图中的双实线表示数据通路，细实线表示控制命令信息通路。

(1) 存储器

存储器是实现程序与数据存储功能的部件。它接收外界送来的程序与数据，在处理过程中提供指令与数据信息，并保存运算结果。

(2) 运算器

运算器是对数据进行各种运算的部件。这里所指的运算除了加、减、乘、除等基本算术运算之外，还包括与、或、非、异或等基本逻辑运算，以及数据的传送，移位等操作。

(3) 控制器

控制器是根据存储器中存储的程序，发出执行程序中指令所需的全部控制命令，来控制计算机中存储器、运算器、输入/输出设备的工作，安排好计算机工作的顺序，对可能遇到的各种情况进行处理。控制器从存储器中指定的单元开始，逐条取出指令，经过分析后，按要求向各部件发出相应的控制信号，完成指令所规定的操作。

在控制器工作过程中，还要接受执行部件的反馈信息。例如，运算器送来的运算结果、状态等。这些反馈信息为控制器判断下一步如何工作提供了依据。

存储器、运算器和控制器组成了计算机的主机，而运算器和控制器合在一起通常被称为中央处理器，简称 CPU (Central Processing Unit)。

构成主机的存储器通常亦称为内存。

(4) 输入设备

用于接收外界的数据和人们编制的程序与命令等信息。

(5) 输出设备

用于把信息（或结果）以适当的形式输出。

输入设备和输出设备统称为外部设备。

在冯·诺依曼型计算机中，指令和数据均以二进制形式存储。通过指令序列的顺序执行来实现程序所安排的任务。

以上介绍的内容都是针对实际的物理装置的，用计算机的术语来讲，即是所谓的“硬件”。一台没有相应“软件”支持的计算机通常称为“裸机”。一台裸机在功能上往往很有局限，用户使用极不方便。所谓的“软件”是指由计算机中完成一定功能的所有程序与数据的集合。软件能对硬件的功能加以扩充和完善。举例来讲，硬件本身只能识别由“0”与“1”组

成的二进制代码，用户想要在裸机上输入一个数据，自己就得编写许多条指令组成的一段程序，而且，二进制代码难记难学，用户会感到十分不便。如果为裸机配上汇编程序、编译程序、操作系统、数据管理系统及应用程序等软件，那么输入一个数据，即使用户有更多更高的要求，也是非常容易实现的。

因此，一台裸机每加上一层软件，就会变成一台功能更强的机器。通常把这种功能更强的机器称为“虚拟机”。图 1-2 反映了硬软件之间的这种关系。



图 1-2 硬、软件层次关系

数据库管理系统就是这样一种软件，用户通过它与计算机打交道，从而使得操纵数据变得极为方便与容易，大大减少了用户的工作量。

1-2 数据库简介

数据管理指的是对数据的组织、编目、存贮、检索和维护等。它是数据处理的中心问题。随着计算机技术的发展，数据管理也经历了由低级向高级的发展过程。大体上，可以分为三个阶段：

(1) 人工管理阶段（五十年代中期以前）

(2) 文件管理阶段（五十年代后期至六十年代后期）

(3) 数据库系统阶段（七十年代初以后）

文件管理阶段与人工管理阶段相比，由于它通过文件系统来管理和使用各种设备介质上的信息，把信息的逻辑结构映象成设备介质上的物理结构。这样就可以使用户不必过多地考虑物理细节，而将精力集中于算法。

文件系统中的文件基本上是对应于一个或几个应用程序，或者说数据是面向应用的。它仍然是一个不具有弹性的无结构信息集合，存在以下几个方面的问题：

• 冗余度大

文件系统下的用户各自建立自己的文件，相互之间数据不能共享，造成数据大量重复存储。不仅浪费存储空间，更严重的是容易造成数据的不一致性。

• 数据独立性较差

数据和程序相互之间的依赖仍较严重。

• 数据无集中管理

各个文件没有统一的管理机构，其安全性和完整性等无法得到保证。

所有这些问题，文件系统本身无法解决，这严重地阻碍了数据处理技术的发展，同时，也成为数据库技术产生的原动力和背景。

数据库系统的目标首先就是克服文件系统的这些弊病，用一个软件来集中管理所有的文件，以实现数据的共享，保证数据的完整性、安全性。

数据库系统的核心是数据库管理系统 DBMS (Database Management Systems)。在 DBMS 的集中管理下，将具有较高的数据独立性及较低的冗余度，并且相互之间有联系的文件集合叫做数据库。

基本的数据库模型有三大类，即层次模型系统，网络模型系统与关系模型系统。

目前，在微机上实现的 DBMS 软件，如 dBASE、INFORMIX、UNIFY、INGRES、ORACLE、FOXBASE 等几乎都是关系型数据库管理系统。下面介绍有关关系型数据库的概念。

一般来讲，一个关系型数据库可以看作为一张或几张由若干行、列所组成的表格，每一张这样的表格称为一个“关系”，它们在数据库系统中用文件的形式存放，称为“数据库文件”。表格中的行称为“记录”，列称为“字段”。每个字段都有一个名称，叫做“字段名”，这相当于表格中的栏目。关系型数据库管理系统 (RDBMS) 具有提供数据库文件之间（即表格和表格之间，或关系和关系之间）相互对照、相互引用的能力。换言之，关系模型中，存在着两类数据，一类是组成数据库文件的记录，另一类则反映数据库文件间的联系。

例 1. 图 1-3 中的表格 1、表格 2 及表格 3 分别反映学生、课程与学生学习的情况。

各表格中的第一行为各个字段名，其余各行都为记录，各列为字段。

显然，通过各个表格可以了解到一些基本情况，如，学号为 875202 的林国荷是一位女同学，生于 1967 年 12 月 29 日，不在职，国家每月给予 35 元的贷款。又如，“数据库概论”这门课程号为 51002，第六学期开课，学时数为 50，学分为 3 分。还可以通过这些表格，来得到某些统计信息。如，1967 年出生的学生有哪些，第五学期开设了哪些课程，课程号为 51002 的选修学生的平均成绩等等。

回答上述这些问题仅涉及其中的某一张表格的内容，即仅用了前述的第一类数据。对于象林国荷同学所选修的“数据结构”这门课程的成绩是多少，或林国荷同学第六学期的平均成绩为多少等这些问题，不是一张表格中的信息所能回答的，而是通过几张表格之间的相互对照才能回答。这是第二类数据要解决的问题。

顺便提一下，能否将上述的三张表格合并为一张表格，即在一个记录中同时反映学生的基本情况、所选修的课程情况和学习成绩等，请读者自己从数据冗余的角度出发，细心地考虑一下，就不难作出正确的回答。

最后要指出的是，关系模型把数据之间的复杂联系代换成表格之间的关系，但表格之间应保持某种联系的信息。如，表格 1 和表格 3 之间是通过“学号”来保持相互联系的。要是去掉表格 1 或表格 3 中的“学号”字段，则它们相互之间的联系便中断了。

每张表格应具有如下的性质：

(1) 在表格中，每一列的所有数据均应属于同一数据类型。

(2) 在表格中, 每一列被指定一个名字, 而且它们不允许重名。

(3) 在表格中, 行和列的次序是任意的, 这并不影响它们的信息内容。

(4) 在表格中, 每一列都是不能再分的基本信息单位。

表格 1

学号	姓名	出生年月	性别	在职否	贷款	简历
875202	林国荷	12/29/67	女	否	35.00	

表格 2

课程号	课程名	开课学期	学时数	学分
51001	数据结构	4	85	5.0
51002	数据库概论	6	50	3.0

表格 3

学号	课程号	成绩
875202	51001	94.0
875202	51002	89.0
875202	41002	89.0
875203	51003	86.0

图 1-3 学生、课程与学习情况

1-3 汉字 FOXBASE+数据库系统

FOXBASE+由 Fox Software 公司推出, 它是近年来新崛起的众多 dBASE III 兼容系统中的佼佼者。与其它同类系统相比, FOXBASE+速度极快, 不仅与 dBASE III+完全兼容, 而且对 dBASE III+进行了有益的扩充。FOXBASE+一经推出, 就立即受到广大 dBASE 用户的欢迎, dBASE 系统的用户只需花很少的时间进行学习, 就能编制和运行 FOXBASE+程序。

FOXBASE+是用 C 语言编制的, 因此便于移植, 适合于在多种硬件环境与操作系统下运行。目前主要用于 XT、AT、MAC 机以及在网络环境下运行, 并适合 DOS 与 XENIX 操作系统。FOXBASE+又分为多用户版本与单用户版本, 两类版本间百分之百兼容。汉字 FOXBASE+可以在 CCDOS 操作系统下运行。按显示器的分辨率, 又有适用于高分辨率 (25 行) 和中分辨率 (10 行) 显示器的不同 FOXBASE+系统。

FOXBASE+现有三个版本: 1.00、2.00 以及 2.10 版本。本书以 FOXBASE+2.00 版本

为基础。

1. 汉字 FOXBASE⁺的文件类型及文件管理操作

在汉字 FOXBASE⁺管理下的文件都应符合操作系统的规定：文件名字由文件名与扩展名两部分组成，一般形式为：

文件名 扩展名

例如 XK.DBF 表示文件名为 XK，扩展名为 DBF 的文件。文件名由英文字母或汉字开头，是由英文字母、汉字、数字和下划线组成的一个字符串，其宽度最多为 8 个字符（一个汉字按两个字符计算）。扩展名由 3 个字符组成，它不仅是文件名的一部分，而且经常用于区分文件类型。由于操作系统的要求，当具体指定一个文件时，还应在文件名前加上盘号以及文件路径名（不加时为系统默认值）。盘号、路径名、文件名以及扩展名中有一个不相同的二个文件，是两个不同的文件。

汉字 FOXBASE⁺共有十类不同格式的磁盘文件，用户可以利用它们存贮信息，构造自己的数据库系统。汉字 FOXBASE⁺对这十类文件都有各自不同的默认扩展名。虽然用户可以为自己的文件规定适合自己的各种扩展名，但是，我们建议，应尽量采用系统默认扩展名。这样，既明确文件的性质，又便于管理与交流。下面我们简单介绍这十类文件，并在括号中给出系统默认扩展名。

(1) 数据库文件 (DBF)

这是数据库中最基本的文件。其中，以记录和字段的形式存放着用户数据。用户可建立多个数据库文件，并可修改结构，修改、检索、统计其中的数据。

每个库文件都由两部分组成：数据库文件的数据结构及数据库文件的记录内容。

一个数据库文件最多可包含 10 亿个记录，每个记录最多由 128 个字段组成，每个字段不得超过 254 个字节，每个记录不得超过 4000 个字节。

数据库文件由 CREATE 命令建立结构；用 APPEND、INSERT、APPEND FROM 等命令输入内容；用 EDIT、BROWSE、CHANGE、REPLACE、@...SAY...GET...和 READ 等命令对其内容进行编辑修改；用 MODIFY STRUCTURE 命令修改其结构。

数据库文件用 USE 命令打开；用 USE、CLOSE DATA 等命令关闭；用 DISPLAY、LIST、@...SAY...GET 等命令显示内容。

(2) 数据库备注文件 (DBT)

备注文件又称记忆文件、备忘文件，是数据库文件 (DBF) 的辅助文件，用来存放 DBF 文件中的备注字段 (MEMO) 实际内容。同一个数据库文件中所有 MEMO 字段内容都存放在同一个备注文件中。一个备注字段最多可包含 4096 个字符。

若建立的数据库文件有备注字段，系统自动建立一个与此库文件同名但扩展名为 DBT 的备注文件用来存放同名库文件中所有备注字段的实际内容。而库文件中每个备注字段只占 10 个字节的空间。

含有备注型字段的库文件被打开或关闭时，同时也将自动打开或关闭同名备注文件。如果备注文件被删除，则不能打开同名库文件。只有重新建立同名备注文件才能打开此库文件。

用 DISPLAY 或 LIST 命令显示备注文件内容。

3) 索引文件 (IDX)

它由索引命令 (INDEX) 生成。一般由一个或 n 个字段组成索引关键字, 根据索引关键字值, 由小到大按索引排列库文件记录 (称为逻辑顺序)。有了索引文件, 将按记录的逻辑顺序来使用库文件内容。因此, 操作速度很快。

打开一个库文件时, 可同时打开 1~7 个有关索引文件。但只有其中的一个索引文件——主索引文件控制记录的逻辑顺序。当对一个库文件的内容进行各种修改时, 所有同时打开的索引文件都会自动地更新。对于未同时打开的索引文件, 可重新打开后, 用 REINDEX 更新。

FOXBASE+ 的索引文件与 dBASE 的索引文件尽管在功能上相同, 但在结构上是不同的。对同一库文件来说, 前者的索引文件比后者的索引文件要小得多, 且操作速度也快。当在 FOXBASE+ 环境下打开一个 dBASE 形式索引文件 (默认扩展名是 NDX) 时, 系统将立即自动建立一个等价的 FOXBASE+ 同名索引文件 (默认扩展名是 IDX)。使 FOXBASE+ 索引文件与 dBASE 索引文件共存。并在屏幕上显示:

dBASE III index—rebuilding

若在 CONFIG.FX/CONFIG 文件中加上:

INDEX=NDX

则索引文件的默认扩展名变为 NDX。但在工作中, 原 dBASE 索引文件将被同名 FOXBASE+ 索引文件复盖而被清除。

(4) 命令文件 (PRG)

命令文件, 即程序, 是由一系列完成特定应用功能的命令所构成。它是一个 ASCII 文件。用 FOXBASE+ 的 MODIFY COMMAND 命令建立, 也可在 DOS 状态下, 用 EDLIN 或其它编辑器生成。用 TYPE 命令显示其内容, 用 DO 命令启动执行命令文件。

为方便用户并加快程序执行速度, 可把多达 128 个程序组合在一起构成一个过程文件。为方便用户生成过程文件, FOXBASE+ 又提供了一个过程文件编辑器 foxbind, foxbind 可以迅速、自动地收集指定的命令文件构成一个过程文件。

用 SET PROCEDURE TO 命令打开一个过程文件。用 DO 命令执行打开过程文件中一个程序时, 速度极快, 近似为 FOXBASE+ 本身程序复盖速度。

(5) 目标文件 (FOX)

FOXBASE+ 提供编译命令 FOXPCOMP 把命令文件转换成目标文件 (原命令文件仍然存在)。目标文件是一种伪编译文件, 不能在 DOS 下执行。但比原命令文件短小, 装载速度快。可以认为目标文件是正规整理过的命令文件。其优点为:

- 节省磁盘空间
- 目标文件只占其命令文件空间的 70% 到 50%。
- 提高程序装载速度

尽管目标文件与命令文件的执行速度相同, 但目标文件装载速度显著快于命令文件, 从而加快了处理速度, 提高了系统效率。在宏观上, 提高了程序执行速度。

- 保证程序语法上的正确性

在生成目标文件时, 系统自动对各条命令检查语法, 显示其中的出错内容。从而保证了程序在语法上的正确性。