

(培训讲义资料)

关系数据库管理系统

dBASE(II、III)程序 设计实用指南

尹彦芝 施振川 孙凤霞编

科海培训中心
中国科学院
计算所十七室

一九八八年十月

前 言

dBASE 自问世以来,受到广大用户的欢迎,现已成为广泛流行的一种数据库管理语言。

本书既不是作为dBASE的教科书,也不是作为学习dBASE的启蒙读物。它是面向这样一些读者,他们对dBASE语言已经有了一些基本了解,现在想着手用dBASE语言去编写程序。或者,他们已经编写过一些程序,现在想成为一名dBASE高级程序员。但是,由于dBASE语言其固有的易学易用的特点,所以哪怕是一名初学者,也可以凭借此书学会使用dBASE。

本书结合实例,深入地阐述了dBASE的某些概念,详细地讲解了dBASE程序设计各方面的技巧、和容易出现的问题及其解决办法。书中的大量例题是集许多人的多年经验而成,对于程序员来说,是一份非常珍贵的借鉴资料。

中国科学院计算技术研究所第十七研究室主任董洪皋同志对本书的编写提了宝贵的意见,并对全书进行了审校,借此向他表示衷心感谢。我们非常希望广大读者对于书中的不足和错误之处坦率地提出宝贵意见,以使它不断完善,更好地为大家服务。

作者

通用符号定义

符号	名字	用法
*	星号	表示是一个dBASE命令行
::=	赋值号	对语法成份或元变量作出定义或解释
...	省略号	①省略有关部分 ②用在命令行的结尾，表示其它意义要参照随后的程序
;	分号	用在命令行的结尾，表示此命令行在下一物理行内继续
< >	尖括号	用在命令语法说明中，表示其中的内容在实际程序中是由程序员填写的
[]	方括号	①用在命令语法说明中，表示其中的内容是可选择的 ②在字面字符串中用作定界符，此时与单引号和双引号同义
{ }	大括号	用在命令语法说明中，表示其中的内容是可以重复的，允许重复的次数由有关文本说明
	竖杠	用于分离各选择项，意义相当于“或者”
^	上尖号	①用在文本中，表示是控制键，如^S就等于Ctrl—S ②用在程序中，表示是指数运算

目 录

前言	(1)
通用符号定义	(10)
第一章 系统环境和准备工作	(1)
1.1 dBASE的各种版本	(1)
1.2 支持dBASE的操作系统	(1)
1.3 运行dBASE所需的文件	(2)
1.4 操作系统配置所需的改变	(3)
1.5 dBASE缓冲区	(3)
1.6 dBASE运行时的特性和配置	(4)
1.7 安装	(6)
1.8 硬盘系统	(6)
第二章 基本概念	(7)
2.1 文件名	(7)
2.2 别名	(8)
2.3 字段名	(8)
2.4 内存变量名	(9)
2.5 dBASE数据库文件的类型	(9)
2.6 全屏幕操作控制键	(10)
第三章 数据类型	(12)
3.1 字符型数据	(12)
3.2 数值型数据	(14)
3.2.1 数的范围和精度	(14)
3.2.2 数值数据的操作符和函数	(14)
3.2.3 数值数据的内部表示法	(15)
3.3 逻辑型数据	(16)
3.4 日期型数据	(16)
3.5 MEMO字段	(17)
第四章 数据结构	(18)
4.1 概述	(18)
4.2 数据文件的结构	(18)
4.3 索引文件的结构	(22)
4.3.1 二进制树	(22)
4.3.2 B ⁻ 树	(23)
4.3.3 B ⁺ 树	(24)

4.3.4	dBASE II 索引文件结构	(24)
4.3.5	dBASE II 中的索引搜索算法	(25)
第五章	命令文件的处理	(26)
5.1	概述	(26)
5.2	命令文件的基本框架	(27)
5.2.1	命令文件的头	(27)
5.2.2	命令文件的尾	(27)
5.2.3	命令文件的体	(28)
5.3	程序结构	(29)
5.3.1	IF 命令	(29)
5.3.2	CASE 命令	(29)
5.3.3	循环结构	(31)
5.3.4	过程结构	(33)
5.4	命令文件间数据的变换	(34)
5.5	算法	(36)
5.5.1	记录文件的细节	(36)
5.5.2	结构程序命令中的注解	(37)
5.5.3	子程序的使用	(38)
5.5.4	递归	(41)
5.5.5	DO.....WITH 命令的改进	(43)
5.5.6	dBASE 的加密版本	(43)
5.5.7	应注意的问题	(43)
第六章	工作环境参数的设置	(45)
6.1	概述	(45)
6.2	环境参数简表	(45)
6.3	算法	(47)
6.3.1	环境参数设置举例	(47)
第七章	数据的处理	(49)
7.1	概述	(49)
7.1.1	直接和间接	(49)
7.1.2	变量	(49)
7.1.3	操作符	(49)
7.1.4	函数	(50)
7.1.5	其它符号	(52)
7.1.6	表达式	(53)
7.1.7	关于 MEMO 字段	(54)
7.1.8	算法	(54)
7.2	字符数据的处理	(57)
7.2.1	子字符串	(57)

7.2.2	字符串的相等	(58)
7.2.3	算法	(59)
7.3	日期数据的处理	(62)
7.3.1	日期格式	(62)
7.3.2	日期的计算	(63)
7.3.3	算法	(63)
7.4	逻辑数据的处理	(66)
7.4.1	逻辑表达式	(66)
7.4.2	算法	(66)
7.5	数值数据的处理	(68)
7.5.1	数的范围	(68)
7.5.2	数的精度	(68)
7.5.3	数值运算的函数	(68)
7.5.4	数值运算中的错误	(68)
7.5.5	算法	(69)
第八章	内存变量	(71)
8.1	概述	(71)
8.2	内存变量区和内存变量的大小	(72)
8.3	内存变量的特性	(72)
8.4	宏代替	(73)
8.5	优先权	(75)
8.6	内存变量文件	(75)
8.7	内存变量的管理	(76)
8.8	空内存变量	(76)
8.9	算法	(76)
8.9.1	在DO WHILE...ENDDO循环中的宏	(76)
8.9.2	内存变量的类型的转换	(76)
8.9.3	数组	(76)
8.9.4	应注意的问题	(81)
第九章	屏幕处理	(82)
9.1	概述	(82)
9.2	屏幕输出处理	(82)
9.2.1	格式化方式	(82)
9.2.2	非格式化方式	(84)
9.2.3	颜色的控制	(85)
9.2.4	算法	(91)
9.3	屏幕输入处理	(95)
9.3.1	屏幕输入处理的方式	(95)
9.3.2	格式化输入	(69)

9.3.3	格式文件	(101)
9.3.4	算法	(102)
第十章	数据文件的处理	(109)
10.1	概述	(109)
10.1.1	建立文件	(110)
10.1.2	改变文件的结构	(111)
10.1.3	优先权	(113)
10.1.4	算法	(113)
10.2	文件的打开和关闭	(118)
10.2.1	概述	(118)
10.2.2	算法	(120)
10.3	添加数据	(121)
10.3.1	从文件中输入	(121)
10.3.2	从键盘上输入	(121)
10.3.3	算法	(123)
10.4	数据的排序	(124)
10.4.1	概述	(124)
10.4.2	分类法	(124)
10.4.3	索引法	(125)
10.4.4	算法	(126)
10.5	移动数据和寻找数据	(130)
10.5.1	概述	(130)
10.5.2	按数据进行移动	(132)
10.5.3	按记录进行移动	(134)
10.5.4	算法	(136)
10.6	修改数据	(141)
10.6.1	用数据文件来修改数据	(141)
10.6.2	从键盘上修改数据	(141)
10.6.3	应注意的几个问题	(142)
10.7	显示数据	(143)
10.7.1	概述	(143)
10.7.2	算法	(143)
10.8	拷贝到另一个数据文件中	(150)
10.8.1	概述	(150)
10.8.2	算法	(151)
10.9	使用多个文件和多个盘片	(151)
10.9.1	使用多个文件	(151)
10.9.2	使用多个盘片	(154)
10.9.3	算法	(154)

第十一章 外部文件和已关闭的文件	(156)
11.1 外部文件	(156)
11.2 对已关闭文件的操作	(160)
11.3 算法	(161)
第十二章 报表生成和打印	(163)
12.1 概述	(163)
12.2 格式化输出	(164)
12.3 表格生成和输出	(165)
12.4 关于打印机的一些问题	(167)
12.5 算法	(168)
12.5.1 屏幕拷贝	(168)
12.5.2 页计数	(168)
12.5.3 列格式化	(171)
12.5.4 在报表文件中用空格代替0值	(172)
12.5.5 改进数据输出	(174)
12.5.6 应注意的几个问题	(175)
第十三章 汇编语言接口	(176)
13.1 概述	(176)
13.2 8位系统	(177)
13.3 16位系统	(178)
13.4 算法	(179)
13.4.1 日期测试	(179)
13.4.2 改变用户区	(183)
第十四章 多用户环境	(185)
14.1 概述	(185)
14.2 算法	(187)
第十五章 调试	(189)
15.1 引起错误的常见原因	(189)
15.1.1 语法错	(189)
15.1.2 结构错	(189)
15.1.3 逻辑错	(190)
15.2 调试技术	(191)
15.2.1 检查系统	(191)
15.2.2 缩小范围	(192)
15.2.3 临时去掉程序的某些部分	(193)
15.2.4 纠正错误	(193)
15.3 算法	(194)
15.3.1 程序优化	(194)
第十六章 交互方式	(196)

16.1	概述	(196)
16.2	MODIFY COMMAND命令	(196)
16.3	外部MEMO字段	(197)
16.4	功能键	(197)
16.5	算法	(198)
16.5.1	命令文件园点提示符	(198)
16.5.2	长命令行	(200)
附录A	错误信息	(201)
附录B	子程序	(212)
B.1	日期子程序	(212)
B.1.1	使用日期子程序的说明程序	(212)
B.1.2	设置dBASE II 系统日期	(214)
B.1.3	核实日期的有效性	(215)
B.1.4	把日历变为Julian周期	(217)
B.1.5	把Julian周期变为日历	(217)
B.1.6	从日历求星期几	(219)
B.1.7	列出两个日期之间的记录	(220)
B.1.8	把普通日历日期变为Julian周期日	(221)
B.2	通用子程序	(223)
B.2.1	把美元数转变成等价的字符	(223)
B.3	数学子程序	(227)
B.3.1	使用数学子程序的说明程序	(227)
B.3.2	开方子程序	(229)
B.3.3	指数子程序	(230)
B.3.4	自然对数子程序	(231)
B.3.5	三角函数子程序的使用说明	(233)
B.3.6	三角函数子程序	(235)
附录C	程序	(240)
C.1	命令模拟程序	(240)
C.1.1	UPDATE模拟程序	(240)
C.1.2	JOIN模拟程序	(241)
C.1.3	TOTAL模拟程序	(242)
C.1.4	QUIT TO模拟程序	(243)
C.2	通用程序	(245)
C.2.1	打印 邮政标签	(245)
C.2.2	计算索引文件的大小	(248)
C.2.3	从一个字段中把名和姓分开	(249)
C.2.4	把大写字符字段变为仅第一个字符大写	(252)
C.2.5	把一个句子分段	(255)

C.2.6	计算标准误差	(257)
C.2.7	以随机字符记录建立一个数据文件	(259)
C.2.8	16进制变为10进制	(261)
C.2.9	10进制变为16进制	(262)
C.3	系统开发程序	(264)
C.3.1	打印系统信息	(264)
C.3.2	建立变量的交叉索引表	(271)
C.3.3	接受通行字到应用系统	(279)
C.3.4	退出应用系统前的转贮程序	(281)
C.3.5	测定一条dBASE III命令的执行时间	(287)
C.3.6	测定几条dBASE III命令的执行时间	(290)
C.4	生成程序的程序	(293)
C.4.1	生成主菜单和子菜单的程序	(293)
C.4.2	生成报表命令文件	(302)
C.4.3	生成dBASE II中的STORE和REPLACE语句	(316)
C.4.4	生成dBASE III中的STORE和REPLACE语句	(320)
附录D	汇编语言子程序	(324)
D.1	汇编码接口程序	(324)
D.1.1	把INTEL 16进制文件变为dBASE II POKE序列(1)	(324)
D.1.2	把INTEL 16进制文件变为dBASE II POKE序列(2)	(327)
D.1.3	从一个二进制文件中建立一个POKE序列	(330)
D.2	CP/M—80程序	(333)
D.2.1	CP/M—80程序的说明程序	(333)
D.2.2	等待控制台输入或延时结束	(337)
D.2.3	CP/M 2.2上的Inkey函数	(338)
D.2.4	用前置0代替前置空格	(340)
D.2.5	左对齐字符串	(341)
D.2.6	用空格代替文件的控制字符	(342)
D.2.7	求盘上剩余的容量	(349)
D.3	CP/M—86程序	(357)
D.3.1	用空格代替文件的控制字符	(357)
D.3.2	求盘上剩余的容量	(364)
D.3.3	取盘驱动器的状态	(366)
D.4	DOS 2.0程序	(370)
D.4.1	DOS 2.0程序的说明程序	(370)
D.4.2	MS—DOS 2.0X目录拷贝程序	(373)
D.4.3	取IBM PC系统日期	(377)
D.4.4	改变MS—DOS 2.0X上的目录	(377)
D.4.5	求盘上的剩余的容量	(381)

D.4.6	取IBM PC系统时间	(385)
D.4.7	置IBM PC系统日期	(387)
D.4.8	用空格代替文件的控制字符	(388)
D.4.9	在菜单的指定位置显示系统时间	(396)
附录E	各种表格	(402)
E.1	操作符表	(402)
E.2	环境参数表	(403)
E.3	函数表	(409)
E.4	命令表	(413)
E.5	其它符号表	(429)
E.6	术语定义表	(430)
E.7	ASC II码表	(432)
附录F	技术参考资料	(435)
F.1	@...SAY...GET [dBASE II 2.3B, 2.4]	(435)
F.2	使用多个索引文件时应注意的问题 [2.3B, 2.4]	(437)
F.3	MODIFY COMMAND命令中的问题	(439)
F.4	SET ALTERNATE TO [<文件>] [dBASE II 2.3B, 2.4]	(440)
F.5	在dBASE II中使用索引文件时的问题 [2.3, 2.4]	(441)
F.6	禁止分号符的功能 [2.3B, 2.4]	(443)
F.7	禁止换页 [2.3, 2.4, 2.41]	(444)
F.8	矫正dBASE II数据文件头中的记录计数 [2.3B, 2.4]	(445)
F.9	在MP/M操作系统下使用dBASE II [2.3B, 2.4]	(446)
F.10	重建dBASE II数据文件头 [2.3, 2.4]	(448)
F.11	支持国际字符集 [2.4版, 16位系统]	(449)
F.12	防止dBASE II截掉一个命令过程的最后几行	(450)

第一章 系统环境和准备工作

1.1 dBASE的各种版本

自从dBASE问世以来,已经有许多次的改进,有了许多版本。毫无疑问,新的dBASE版本还会不断诞生。在这本书里,重点介绍的各种版本有:

dBAB II: 2.3, 2.3A, 2.3B, 2.3D, 2.4, 2.41, 2.42

dBASE III: 1.00

在dBAS II中,2.42版是2.41版的国际版本,差别是2.41版的数值精度只有10位数字,而2.42版的数值精度有16位数字,这就使得2.42版的运行速度慢一些。

在这本书中,为了给这些版本分组,我们用了符号“×”。例如,2.4×指的是所有2.4版本,包括2.4, 2.41和2.42。

还有其它一些版本,本书未作重点介绍,这些版本是:

dBASE II: 2.02, 2.3C, 2.4E

2.02版是早期的版本,后来被2.3版代替。2.3版成了dBASE II的最为广泛流行的版本。

2.3C版本本来是想代替2.3B版的,但是在发行后又几乎立刻收回去了。

2.4E是dBASE III的先声,是为AT & T公司的UNIX操作系统环境研制的。

1.2 支持dBASE的操作系统

dBASE可以在好几种操作系统支持下运行,下面这张表格列出了这些操作系统。

操作系统名称	操作系统类型	dBASE 版本
CP/M-80	8位 单用户	2.3b, 2.4, 2.41
CP/M-86	16位 单用户	2.3d, 2.4, 2.41
Concurrent CP/M	16位 单用户	2.3d, 2.4, 2.41
Concurrent DOS	16位 单用户	2.3d, 2.4, 2.41
PC-DOS	16位 单用户	2.3d, 2.4, 2.41, dBASE III
MS-DOS	16位 单用户	2.3d, 2.4, 2.41, dBASE III
TurboDOS	8位 多用户	2.3b, 2.4, 2.41 (单用户)
MP/M- II	8位 多用户	2.3b, 2.4 (单用户)
MP/M-86	16位 多用户	2.3d, 2.4, 2.41 单用户
EPC/OS	8/16位 多用户	2.3b, 2.4, 2.41 (单用户)
MMCSI	8位 多用户	2.3b, 2.4, 2.41 (单用户)
3-COM Ethernet	16位 多用户	2.3d, 2.4, 2.41 多用户
PC-NET	16位 多用户	2.3d, 2.4, 2.41 (单用户)
在3BX机上的UNIX系统V	32位 多用户	2.4e, dBASE III

1.3 运行dBASE所需要的文件

当dBASE运行时，并不要求dBASE所提供的全部文件都存在。在任何情况下，那些实际上不属于dBASE的实用程序文件（例如INSTALL和dCONVERT），可以不在盘上。

dBASE运行时所用的文件可以分为两类。一类是必需的，这一类文件在运行时必须存在；另一类是可选的，这一类文件只有当运行过程中需要用到它们时才必须存在。

	必须始终存在的文件	可以在需要时再调入的文件
dBASE III 1.00	dBASE.EXE, 主程序文件 dBASE.OVL, 主复盖文件	HELP.DBS, 联机资料文件 ASSIST.HLP, 菜单驱动的辅助文件
dBASE II 2.4×	dBASE.COM, 主程序文件 dBASEOVR.COM, 主复盖文件	dBASEMSG.TXT, 联机资料文件
dBASE II 2.3×	dBASE.COM, 主程序文件 dBASEMSG.COM, 信息 dBASEMAI.OVR, 主复盖文件 dBASEAPP.OVR, APPEND命令 dBASEMSC.OVR, 其它复盖文件	INSTALL.COM, 安装程序 dBASEBRO.OVR, BROWSE命令 dBASEJOL.OVR, JOIN命令 dBASEMOD.OVR, MODIFY命令 dBASEROG.OVR, REPORT FORM命令 dBASESRT.OVR, SORT命令 dBASETTL.OVR, TOTAL命令 dBASEUPD.OVR, UPDATE命令

主程序文件在第一次被调用时就被装入内存，因此，它可以放在某一台驱动器上或某一驱动器/目录上（不要求是当前驱动器，也不要求是当前目录）。为了调用它，在打入文件名dBASE以前，必须打入驱动器名，如果操作系统支持路径的话，还需要打入路径名。

复盖文件是由dBASE主程序文件调用的。在dBASE II中，复盖文件必须放在当前驱动器/目录上，因为dBASE II只会到当前驱动器/目录上寻找复盖文件。在dBASE III，复盖文件可以放在非当前驱动器/目录上，只要预先正确设置了路径名，告诉dBASE到什么地方去找复盖文件就可以了。你的操作系统是否支持路径名的概念以及如何使用路径名，请参考你的操作系统手册。

在dBASE II，所有其它的文件（数据库文件，命令文件，索引文件，内存变量文件等等）可以放在另一台驱动器上，因为dBASE II在文件名前可以接受驱动器名。但是，这些文件不能驻留在同一驱动器的不同目录下面，因为dBASE II不接受路径名。dBASE III没有这个限制，所有这些文件可以放在你乐意存放它们的任何地方。

❶ 软件出版商们普遍遵守的一个原则就是常常在销售的盘片上提供一个名为Read.me的ASCII文本文件。如果你有一张具有这个文件的盘片，在使用那个软件前先读读这个文件。

1.4 操作系统配置所需的改变

在MS-DOS和PC-DOS操作系统下运行dBASE III以前,必须对操作系统的两个缺省值进行改变。在一个名为Config.sys的ASCII文本文件中重新设定两个系统参数就可以做到这一点。

Config.sys是当DOS被自举时自动执行的,它是在Autoexec.bat之前执行的,必须放在计算机的自举盘上。

我们要改变的两个操作系统参数是: DOS可以使用的缓冲区的数目和DOS可以同时打开的文件数目。Config.sys文件中应包括这样两行:

```
Buffer = 15
```

```
Files = 20
```

每增加一个缓冲区就会使DOS在内存多占528个字节。因为这会使大量的RAM变得对运行软件不可用,所以在内存容量比较小的情况下应该设置比较少的缓冲区。然而,如果想充分地使用dBASE III,计算机的内存容量应该允许扩大到15个缓冲区。早期的dBASE III手册建议使用24个缓冲区,但是实践证明,10至15个缓冲区就可以运行得较快了。

DOS2.X各版本最多可打开20个文件。超过缺省值8以后,每增加一个文件就使DOS增加39个字节。在这些打开的文件中,DOS保留5个用作标准输入、标准输出、标准错误输出、辅助设备和标准打印机。这就意味着,在整个系统里各种类型的文件最多可以同时打开15个,包括dBASE在内。dBASE应用程序只可以打开13个文件,因为dBASE本身用了两个,一个是供主程序文件用,另一个是供复盖文件用。

虽然dBASE II运行时没有这个配置要求,但是当缓冲区数目增加时,它的速度也会大大地改善。文件数的设置并不影响dBASE II,因为它是直接处理文件控制块FCB,而不是使用DOS的“文件柄”(见第五章“过程结构”),这就允许在dBASE II内同时打开16个文件,不包括dBASE本身在内。

dBASE III使用全部RAM,dBASE II使用固定大小的RAM,不管实际内存有多大。各dBASE版本所用的内存如下:

dBASE 版本	所用的内容容量
dBASE II 16位	128K
dBASE II 8位	64K
dBASE II 8位 (APPLE)	56K

1.5 dBASE缓冲区

DOS的缓冲区是用于计算机的外部设备(如磁盘驱动器和键盘)和软件之间。dBASE也有缓冲区,它是用于DOS缓冲区和dBASE之间。

dBASE使用20个缓冲区,每个缓冲区512字节。这些缓冲区可分为三类:热、温、冷。当缓冲区中的数据是刚刚从盘上读入还没有被改变时,就称缓冲区是冷的。当缓冲区中的数据已被改变,且还没有保留到盘上时,就称缓冲区是热的。如果计算机正在等待键

盘输入，则热缓冲区中的数据就被自动地写到盘上。当缓冲区中的数据被写到盘上以后，就称缓冲区是温的。冷的和温的缓冲区中的数据不被写到盘上。

dBASE II 使用8个缓冲区，每个缓冲区512字节。只有当别的程序需要使用某个缓冲区时，这个缓冲区的数据才被写到盘上。因此，在文件被关闭以前，一定会有某些增添和修改的内容没有被写到盘上，容易受到破坏。

1.6 dBASE运行时的特性和配置

一个名为Config.db的可选择文件可以用来构造dBASE III，这是对构造操作系统的Config.sys文件的补充。当dBASE III被自举时，它按操作系统的路径去寻找Config.db文件，如果找到了这个文件，则dBASE将以这个文件所指定的属性来装备系统。

Config.db文件与Config.sys文件一样是一个ASCII文本文件，每一行说明一个参数。Config.db文件中只允许不带定界符的字面参数，不允许参数中出现操作符或函数。

设置参数的格式

参数的功能

ALTERNATE = <文本文件名>	这个参数既指定了替换文件的文件名，并设置ALTERNATE开关为ON。
BELL = <开关>	
BUCKET = <数字>	<数字>是1到31的整数。这一“篮子”用来保存在@...GET命令中对诸如范围和图形子句之类的参数所作的全部限制选择项（见第九章），缺省值是2K。
CARRY = <开关>	
COLOR = <颜色参数表>	见第九章的颜色表
COMMAND = <命令><命令参数>	当dBASE第一次被自举时，将首先执行这条命令。
CONFIRM = <开关>	
CONSOLE = <开关>	
DEBUG = <开关>	
DECIMALS = <数字>	<数字>为从0到14的整数
DEFAULT = <驱动器标识符>	
DELETED = <开关>	
DELIMITERS = <一个或两个字符>	
DELIMITERS = <开关>	
DEVICE = SCREEN PRINT	
ECHO = <开关>	
ESCAPE = <开关>	
EXACT = <开关>	
F2 = <命令表>	最多30个字符
F3 = <命令表>	最多30个字符
F4 = <命令表>	最多30个字符

F5	= <命令表>	最多30个字符
F6	= <命令表>	最多30个字符
F7	= <命令表>	最多30个字符
F8	= <命令表>	最多30个字符
F9	= <命令表>	最多30个字符
F10	= <命令表>	最多30个字符
GETS	= <数字>	<数字> 为从35到1023的整数, 缺省值为128。
HEADINGS	= <开关>	
HELP	= <开关>	
INTENSITY	= <开关>	
MARGIN	= <数字>	<数字> 为从0到254的整数
MAXMEM	= <数字>	<数字> 为从200到720的整数。这个参数用来定义当执行一个外部程序时dBASE III 将不会释放的内存容量。<数字> 所代表的是释放给系统的起始地址, 以便DOS执行一个外部的处理程序。计算机的内存容量应该为 MAXMEM + 17K + 外部处理程序所需要的内存。<数字> 的缺省值为256K。应该注意, 如果dBASE III 使用了比这个<数字> 更高的内存单元, 则将得不到任何保护, 此时若从dBASE III 内部运行一个外部程序, 则有可能破坏这些单元的内容。
MENUS	= <开关>	
MVARsiz	= <数字>	<数字> 为从1到31的整数, 它代表用K字节表示的内存变量区的容量。
PATH	= <路径表>	
PRINT	= <开关>	
PROMPT	= <字符串>	不带定界符的字面字符串
SAFETY	= <开关>	
STEP	= <开关>	
TALK	= <开关>	
TEDIT	= <文本编辑文件名> (没有扩展名)	当执行MODIFY COMMAND命令时, 将调用这个文本编辑文件来工作。
UNIQUE	= <开关>	
WP	= <字处理文件名>	当在交互方式下编辑MEMO字段时, 将调用这个字处理文件来工作。

注：①所有的参数都是不带定界符的字面值；

②〈开关〉为ON或OFF；

③F2到F10是IBM-PC机功能键的设置；

④如果在Config.db文件中对某些参数没有进行定义，则采用SET命令的参数。

见第六章“工作环境参数的设置”。

1.7 安装

dBASE III是预先安装好的，但是dBASE II在大部分计算机上要求首先运行安装程序。安装程序只需要运行一次，它的主要目的就是为一特定的计算机指定一个正确的屏幕处理办法。安装程序也提供几项其它选择，例如，改变宏代替函数所用的符号（&），选择诸如MP/M这样的多用户操作系统，改变日期的格式等等。

如果在全屏幕操作方式下屏幕不能正确地操作，那么就需要为那个特定的硬件环境安装dBASE。dBASE II 2.3D是预先为IBM PC机安装好的，所以在发货时不带安装程序。

1.8 硬盘系统

在dBASE III中，每当一个在Config.db中被指定为TEDIT或WP的外部字处理程序被使用时，或者当一个外部程序在运行时，在返回到dBASE以前，DOS将重新装入Command.com文件。DOS假定Command.com是装在自举驱动器的根目录下。微机即使有一个硬盘，也还常常从软盘A自举。从一个软盘驱动器上重新装入Command.com所需的时间比从一个硬盘驱动器上重新装入Command.com所需的时间长得得多，并且要求自举盘片是在那个驱动器里。所以，在带有硬盘的系统中，我们应该把Command.com文件放在硬盘上，并设法把这件事告诉DOS。

DOS的命令SET COMSPEC就是用来告诉DOS应该到哪台驱动器上去寻找Command.com文件，而不是一定到自举驱动器上去寻找这个文件。这条命令一般是放在Autoexec.bat文件里，其语法是：

```
SET COMSPEC = <drive>:{ \ \<subdirectory> } \Command.com
```

更详细的说明请参考你的DOS说明手册。