

中华学习机实用大全 ● ② ● ● ● ● ● ● ●

汉字处理 与数据库技术

韩仲清 主编



电子工业出版社



封面设计：阎欢玲

ISBN 7-5053-1047-X/TP · 170 定价：4.60 元

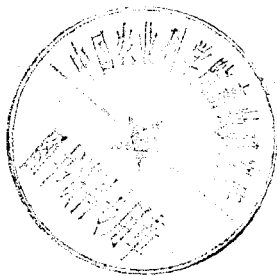
TP273

中华学习机实用大全②

014350

汉字处理与数据库技术

韩仲清 主编



电子工业出版社

内 容 提 要

本书从实用的角度出发，详细介绍了中华学习机CEC-I的汉字处理和数据库技术。主要包括：

键盘输入指法训练；汉字输入技术，其中还特别介绍了汉字的五笔字型输入法；汉字文稿编辑软件WORDSTAR；西文软件的汉化原理与方法；cdBASE数据库的基本概念、安装和运行；数据库文件的建立、检索、插入、修改和删除的方法；命令文件和报表格式的设计；cdBASE程序设计的方法与技巧。

本书的最大特点是内容充实、具体、实用、易学，最适宜于广大青少年、中小学生、家长、各类办公人员、企业管理人员和计算机爱好者自学；也可以作为高等院校非计算机专业、培训班、函授班、职业学校、中专和中小学等学生计算机课程的教材；还可供从事计算机研究和应用的人员使用。

中华学习机实用大全②

汉字处理与数据库技术

韩仲清 主编

责任编辑 施玉新

电子工业出版社出版（北京市万寿路）

电子工业出版社发行 各地新华书店经销

中国科学院印刷厂印刷

开本：787×1092毫米1/32 印张：11.625 字数：270千字

1990年12月第1版 1990年12月第1次印刷

印数：14000册 定价：4.60元

ISBN7—5053—1047—X/TP·170

前 言

中华学习机以前所未有的速度进入寻常人家，成为人们工作、学习和生活的得力助手，尤其是在开发青少年的智力方面，已经显示出了强大的威力。

为满足广大青少年、中小學生、家长和计算机爱好者对中华学习机知识的渴求，我们组撰了这套《中华学习机实用大全》。该书内容丰富，具体、实用；把中华学习机的最新软件以及最实用、最急需的技术、技巧和方法毫无保留地介绍给读者，使初学者很快入门，入门者进一步提高，学到知识，掌握技术，增长才干，启迪智慧，得到力量，增强解决实际问题的能力。

《中华学习机实用大全》分为七册：

1. BASIC和LOGO语言
2. 汉字处理与数据库技术
3. 操作系统
4. FORTRAN与PASCAL语言
5. 汇编语言程序设计
6. 游戏与绘图
7. 硬件维修与经验技巧

如何使用这套书？为便于阅读和使用，每册内容彼此均是独立的，读者可以从任何一本书开始阅读。但是，如果读者是计算机技术的初学者，那么最好按顺序阅读。当然，每本书中可以只选学自己感兴趣的那部分内容。

《中华学习机实用大全》在内容安排上，由浅入深，循序渐进。既考虑到初学者很快入门，又考虑到让入门者进一步提高，还考虑了应用者能够实用。书中有较多实例，读者可以边读边学边用边想边写（写自己的程序）。在结构安排上，既便于自学自用，又可以作为教材。在文字叙述上，力求浅显、通俗、易懂。在选材上，突出实用性技术。

本书是《中华学习机使用大全》的第二册，主要介绍了中华学习机的汉字信息处理和数据库技术，CWORDSTAR和汉字dBASE II的使用方法及其技巧。利用中华学习机进行汉字处理和数据库管理，这两部分均是全新的内容，目前还没有这方面的书籍出版。

欢迎读者对本书进行品评，指出疏漏和错误，我们将甚为感谢！

在编写本书的过程中，电子工业出版社和电子报社的编辑们给予了指导和帮助，特别是王有春、孙萌老师提出了许多宝贵的修改意见；为调试和运行示例程序，成都三开元电脑部经理舒新生无偿地提供了CEC-I中华学习机及其软件；张陞楷副教授审阅了全部书稿。在此一并表示感谢！

参加本书编写的有：韩仲清，刘元社，廖兴祥，丁正铨，揭金良，黄金姬。全书由韩仲清统稿。

编著者

1989年11月14日于四川大学

目 录

第一部分 汉字信息处理技术

第一章 键盘输入指法训练	(2)
1.1 键盘的指法分区管理	(2)
1.2 击键的姿势和方法	(5)
1.3 字符的分组输入练习	(8)
1.4 字符输入的综合练习	(16)
1.5 输入速度的强化训练	(21)
习题一	(22)
第二章 汉字的编码输入方法	(23)
2.1 汉语拼音输入法	(23)
2.2 区位码输入法	(26)
2.3 五笔字型输入法	(27)
2.3.1 五笔字型软件的装入	(27)
2.3.2 汉字的结构分析	(28)
2.3.3 末笔字型交叉识别码	(35)
2.3.4 单体结构拆分原则	(38)
2.3.5 基本字根及其排列	(42)
2.3.6 五笔字型编码输入	(48)
2.3.7 重码与容错码的处理	(56)
习题二	(58)
第三章 文稿编辑软件WORDSTAR	(60)
3.1 如何启动汉字WORDSTAR	(60)

3.1.1	WORDSTAR需要的硬件和软件环境	(60)
3.1.2	如何进入功能主菜单	(61)
3.1.3	功能主菜单简介	(62)
3.2	文稿文件的建立方法	(63)
3.2.1	文稿的输入	(63)
3.2.2	文稿的存盘	(65)
3.2.3	存盘命令小结	(67)
3.3	光标的移动	(67)
3.4	文稿的修改	(68)
3.4.1	插入字符	(68)
3.4.2	删去字符	(70)
3.4.3	更改字符	(72)
3.4.4	修改命令小结	(75)
3.5	字块及其操作	(76)
3.5.1	字块的设定方法	(76)
3.5.2	字块的操作	(77)
3.5.3	字块命令小结	(79)
3.6	文稿的排版和打印	(80)
3.6.1	打印字体的种类	(80)
3.6.2	打印字体的设置方法	(81)
3.6.3	文稿版面设计实例	(83)
3.6.4	文稿的打印	(85)
3.7	WORDSTAR的其它功能	(87)
	习题三	(89)
第四章	汉字信息处理系统简介	(90)
4.1	汉字信息处理系统	(90)
4.2	汉字信息处理系统中的代码标准	(92)
4.2.1	西文输入编码	(92)
4.2.2	汉字信息交换码	(92)

4.3	汉字信息处理过程中的代码变换	(93)
4.4	关于计算机内码的讨论	(96)
	习题四	(99)
第五章	软件汉化的原理与方法	(100)
5.1	什么叫软件汉化	(100)
5.2	系统软件的汉化过程	(101)
5.2.1	汉字的编码	(101)
5.2.2	编写汉字识别系统程序	(106)
5.2.3	系统的调试与修改	(108)
5.2.4	运行汉字识别系统	(108)
5.3	实用、应用软件的汉化问题	(108)
5.4	软件的汉化方法	(109)
5.5	汉字输入法概述	(110)
5.6	汉字库	(115)
5.7	汉字输出技术	(119)
	习题五	(121)

第二部分 汉字数据库cdBASE技术

第六章	cdBASE的基本概念	(123)
6.1	什么是数据库	(123)
6.1.1	数据库的特点	(123)
6.1.2	数据库的种类	(126)
6.1.3	数据库的组成	(130)
6.2	cdBASE文件概述	(133)
6.2.1	文件类型及其扩展名	(134)
6.2.2	使用符号约定	(135)
6.3	常量、变量、函数及数据类型	(136)
6.3.1	常量、变量和函数	(136)

6.3.2 数据类型	(137)
6.4 表达式	(138)
习题六	(140)
第七章 cdBASE的安装和运行	(141)
7.1 系统要求	(141)
7.1.1 硬件环境	(141)
7.1.2 软件环境	(142)
7.1.3 系统文件及其作用	(142)
7.1.4 Z80卡的插入方法	(143)
7.2 cdBASE的启动	(144)
7.2.1 冷启动	(144)
7.2.2 热启动	(145)
7.3 简单命令的用法	(146)
7.4 技术规格和操作总则	(151)
7.4.1 技术规格	(151)
7.4.2 操作总则	(152)
习题七	(152)
第八章 数据库文件及其操作	(154)
8.1 生成数据库文件的基本方法	(154)
8.1.1 数据库文件的创立	(154)
8.1.2 选取工作区和打开数据库文件	(161)
8.1.3 复制数据库文件	(163)
8.2 查看数据库文件	(167)
8.2.1 显示数据库文件	(168)
8.2.2 列表数据库文件	(170)
8.2.3 浏览数据库文件记录	(172)
8.3 定位记录指针	(173)
8.3.1 转到指定记录位置	(173)
8.3.2 跳跃定位指针	(174)

8.3.3	带条件的定位命令	(176)
8.4	数据库文件的修改	(177)
8.4.1	修改数据库记录	(177)
8.4.2	修改数据库文件结构	(181)
8.5	添加记录	(185)
8.6	删除数据库文件记录	(189)
8.7	统计命令	(192)
8.8	两个数据库文件的连接	(195)
	习题八	(198)
第九章	函数与运行特征	(201)
9.1	函数及其用法	(201)
9.1.1	数值函数	(201)
9.1.2	字符处理函数	(203)
9.1.3	逻辑函数	(206)
9.2	运行特征的设置方法	(207)
9.2.1	SET (参数) ON/OFF的用法	(208)
9.2.2	SET (参数1) TO (参数2)的用法	(213)
	习题九	(214)
第十章	排序和索引	(215)
10.1	排序命令	(215)
10.2	索引文件及其操作	(217)
10.2.1	索引文件的建立	(217)
10.2.2	对索引文件的操作	(221)
10.3	对有序文件的操作	(223)
	习题十	(224)
第十一章	命令文件和报表格式设计	(229)
11.1	命令文件	(229)
11.1.1	命令文件的建立方式	(229)

11.1.2	命令文件的执行	(231)
11.1.3	命令文件的修改	(236)
11.2	专用于程序的命令	(237)
11.2.1	键盘输入命令	(238)
11.2.2	条件语句	(240)
11.2.3	循环	(244)
11.3	报表格式设计	(250)
11.3.1	报表格式文件	(250)
11.3.2	格式文件的建立和调用	(257)
11.3.3	实用报表设计	(269)
	习题十一	(266)
第十二章	程序设计方法与技巧	(268)
12.1	程序设计概述	(268)
12.1.1	程序和程序设计	(268)
12.1.2	程序设计的基本技术	(270)
12.1.3	结构程序设计简介	(273)
12.1.4	程序质量	(276)
12.2	cdBASE应用程序设计	(277)
12.2.1	应用程序设计的步骤	(277)
12.2.2	cdBASE程序及语句结构	(281)
12.3	cdBASE实用技巧	(284)
12.3.1	巧用数据库文件	(284)
12.3.2	检查键盘输入	(287)
12.3.3	宏替换的使用技巧	(290)
12.3.4	巧建屏幕格式	(293)
12.3.5	构造数组	(294)
12.3.6	设计口令	(296)
12.3.7	建立专用词库	(299)
12.3.8	巧用TOTAL命令	(301)

12.3.9	美化屏幕显示	(302)
12.3.10	如何调试cdBASE程序	(305)
第十三章	cdBASE与BASIC文件的数据共享	(319)
13.1	数据共享的基本方法	(319)
13.1.1	数据库和文本文件之间的类型转换	(310)
13.1.2	SDF和DELIMITED的区别	(311)
13.2	如何在BASIC中使用数据库文件	(313)
13.3	如何在cdBASE中使用BASIC文件	(314)
习题十三	(316)
第十四章	cdBASE应用实例	(318)
14.1	快速盘点系统	(318)
14.1.1	快速盘点数据库文件 (KSPD.DBF) 结构	(318)
14.1.2	程序功能结构图	(319)
14.1.3	源程序清单及解释	(319)
14.1.4	说明	(326)
14.2	住房管理系统	(327)
14.2.1	住房管理数据库 (ZFGLK.DBF) 结构	(327)
14.2.2	住房管理软件功能结构图	(328)
14.2.3	源程序清单	(329)
附录	dBASE命令和出错提示信息	(335)
一	dBASE命令一览表	(335)
二	dBASE出错提示信息	(343)

第一部分

汉字信息处理技术

本篇介绍中华学习机 CEC-I 的汉字信息处理技术，主要包括：

键盘输入指法训练；汉字的编码输入方法，特别是最为流行和实用的五笔字型输入法；文稿编辑软件汉字WORD-STAR的用法；汉字信息处理系统简介以及西文软件的汉化原理与方法。

第一章 键盘输入指法训练

中华学习机CEC-I使用的是标准键盘，除了可用于程序输入外，还可以用于进行中西文打字训练。本章专门介绍一种称为“触觉法”的字符快速输入技术。这种技术不但适用于西文字符的输入，也适用于中文字符的输入。

1.1 键盘的指法分区管理

如何才能快速准确地输入字符或汉字？除了多练以外，还必须讲求训练的技术和方法。

1. 键位和指头的分区

一个人两只手，十个指头有长短，这是先天确定的。一个CEC-I键盘共有69个键（其它微机键盘可能多于69键），这是预先固定的。如何恰当地运用我们的两只手十个指头来管理这些键？最好的办法是“包产到户”，“责任到手”。对两只手十个指头进行恰当的分工，这就是键盘的指法分区管理。

为了键盘输入的高效和准确，我们采用键位指头分区管理的办法，将键盘分成左右两大部分，参见图1.1所示。从图中可见，在键盘正中有一条“S”形的粗线，它将整个键盘分成左右两部分，这就是左右手的管辖范围。左端由左手负责控制，右端由右手负责管理。同样，为了将键位落实到每一个指头，又分别将左右端由三条线各自分成四个小部分，

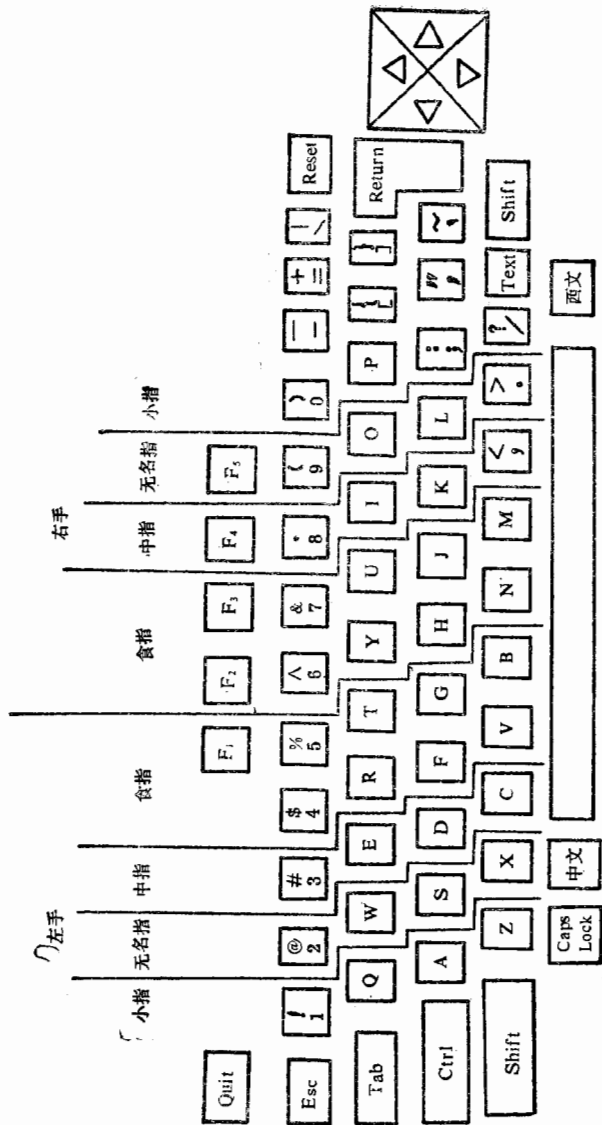


图1.1 键位指法分区图示

每一个小部分由相应的一个指头负责进行管理。可能读者已经发现了，为什么大拇指没有安排工作？但同时又会发现，空格键并没有分配任何指头给予管理，这份工作当然留给大拇指了。

实践证明，这种分工是非常恰当和正确的。因此，我们在训练字符输入时，必须严格地按照这个图示的分工范围进行动作，任何指头的“互相帮助”和“助人为乐”都会造成指法训练的混乱和影响输入字符的速度。

2. 基准键位

从图1.1中的指法分工可见：一个指头要负责管理上下四排（除空格键外）的键位，那么在这四排中，哪一排是最重要的呢？如果将空格键位计算在内，键盘上最基本的字符就分成五排。由于指头的分工管理是按上下行移动的，因此我们手指头处于待命的位置，当然应该是在键盘的中间行上最好。因为这样指头上下移动的平均距离最短，响应的平均速度最快。由于移动距离短，指头错位的可能性也相应较小。于是，我们将ASDF…JKL；字符所在的那一行确定为基准行。在该行中，为使左右移动方便，又将ASDF和JKL；这八个键位确定为基准键位。所谓基准键位就是我们指头的常驻键位，当指头敲击了其它键之后必须立即缩回到基准键位上待命，以便敲击其它键。基准键位和指头的对应关系如图1.2所示。

现在要求读者记住的就是分区图和基准键位。如果我们能够不看键盘和任何图示，就能够在脑海里将键位分区和基准键位同手指头的对应关系浮现出来，那是最好不过的了，因为这样可以帮助我们找准键位。

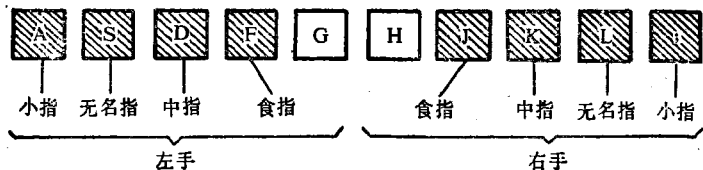


图1.2 基准键位示意图

1.2 击键的姿势和方法

在体育比赛中，运动员优美的姿势和漂亮的动作，会激起观众的阵阵掌声。特别是体操和跳水等运动项目，运动员的姿势和动作成了水平高低的主要指标。在计算机键盘输入中也是如此。特别是专门的录入员，击键姿势的正确与否，将直接影响到输入的速度和正确性。因此，初学者一开始就要注意击键的姿势和动作，严格要求自己，养成良好的习惯。

1. 正确的击键姿势

正确的击键姿势不但美观大方，而且能提高击键的速度和准确率。因此，我们必须十分注意。正确的姿势应该是：

①将监视器放在键盘的正后方，键盘向右稍微移动，左边留出放置输入原稿，以便于阅读。

②姿势端正，腰背挺直，两脚平放在地上，身体稍微倾向于键盘的右方。

③将全身的重量置于椅子上。坐椅的高低应调到适当的位置，以便于手指操作。

④两肘轻轻贴于腋边，手指轻放于基准键位上，手腕悬空平直。

⑤视线集中于键盘左端的输入原稿上。

有些初学者不大注意坐姿，随随便便，松松跨跨。以下姿势都是不正确的，也是有害的。

胡乱地站立在键盘前边或不规范的坐在椅子上。如：一个腿压在另一腿上（俗称翘起二郎腿）；或蹲在椅子上；或趴在桌子上；或过于歪在键盘的一边。

2. 正确的击键方法

有了正确的坐姿，还要有正确的击键方法。初学者必须做到：

①严格按手指划分的范围和分工击键。击键时要敲键，而不要压或按键。

②手腕要平直，手臂保持静止，全部动作只限于手指部分，身体其它部位不能接触工作台和键盘。

③从手腕到指尖的手指形成弧形，指头的第一关节与键位成垂直角度（指甲要修平），参见图1.3（a）。

④输入字符时，手抬起，只有要击键的手指才伸出去击键，击键后立即缩回到基准键位。

⑤击键时以指尖垂直向键位使用冲击力，力量要在瞬间爆发出来，并立即反弹回去，见图1.3（b）。可以说，能否体会和掌握这个要领，是练习计算机键盘输入和英文打字技术的关键。

⑥击键时不可用力太猛，也不可太轻，必须适度，敲键时要声音清脆，有节拍感。

⑦操作计算机键盘时主要的用力部位是指关节，而不是手腕用力，更不是胳膊和全身用力。当练到手指敏感度较强时，可以结合使用腕力。

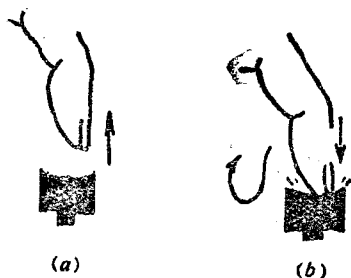


图1.3 击键示意图

在击键练习中，初学者容易发生的错误是，

- ①击键时手指形态变形，手指翘起或往里勾，掌握不住手形。
- ②不是击键而是按键或压键，没有弹性，敲击声迟钝，没有节拍感。
- ③手腕与手指跳动配合不好，既影响手形又影响击键的速度。
- ④左（右）手击键时，右（左）手离开了基准键位，搁在机器或桌子上。
- ⑤打字时手腕不能悬空，依靠在桌子上。
- ⑥小指和无名指没有力量，控制不住，容易带动其它键位。因而要特别注重对这两个指头的训练。
- ⑦眼睛老是盯着键盘，或用触摸的办法找键位。这些都会影响击键的正确性。
- ⑧击键后不习惯于立即缩回到基准键位。
- ⑨急于求成，这是所有初学者最普遍和最大的毛病。

当我们第一次练习敲键时，就应该严格按照要求操作，有意识地克服各种毛病。冰冻三尺，非一日之寒。只有长期坚

持下去，才能收到良好的效果。

1.3 字符的分组输入练习

在了解了指法训练的初步规则以后，就严格按照训练规程，进行实际的键盘输入字符练习。为了循序渐进地掌握各个字符的输入方法和技巧，我们将分组介绍。建议读者严格按照这个顺序进行训练。

1. ASDF和JKL 的练习

这是基准键位上的八个键，应该首先练习并熟练掌握之。训练方法如下：

①按基准键位要求，将左右手的指头轻放在对应的八个键位上。（一定要按规定对位，切记！）

②指头从左到右，再从右到左，依次敲击各个键。（指头不能错位，切记！）

③击键时要注意用力适度和击键的声音。从敲第一个键开始，就要注意击键的节拍感。

④手下盲打（不看键盘），眼睛注视屏幕或稿件。如果一时不习惯而乱了方寸，千万不要惊慌，将指法和键位关系调整正确后再继续练习。（不要眼睛看一个键，指头才击一次键，而是要凭键的位置来找键，切记！）

为了练习手指的位置感觉，一个字符可以反复练多次。

例1. 小指	无名指	中指	食指
AAAA	SSSS	DDDD	FFFF

左手

食指	中指	无名指	小指
JJJJ	KKKK	LLLL	; ; ; ;
<div style="text-align: center; margin-top: 5px;">右手</div>			

请将例1按从左（即左手的小指开始）到右的顺序敲10遍，再按从右（即右手的小指开始）到左敲10遍。再默记一下手指头和键位的对应关系。以下例子各练10遍，字母间的空格可以暂时不管。

例2. ASDFJKL; ASDFJKL; ASDFJKL;

例3. ;LKJFDSA ;LKJFDSA ;LKJFDSA

例4. ASDFJLK; ASDFJLK; ASDFJKL;

例5. LJ; KDAFS LJ; KDAFS LJ; KDAFS

例6. ADD; ALL; AS; ASK; SAD; SALAD;

DAD; LAD; FALL; KALL; JASF; KAD;

LASS; KLLJ;

2. EI和GH的练习

E和I字键在键盘的第三排，根据键位分区，E应该由左手的中指去管理，I由右手的中指管理。EI的敲键方法是：

①将指头放在基准键位上。（这相当于预备动作）

②左手中指上移敲E，右手中指上移敲I。

③击键完成后左右手的中指必须缩回到基准键位上。

例1.

DEDEDEDEDED	KIKIKIKIKIK
左手	右手

例2. DEDKIK KIKDED KIKEDK

KDEDIK

例3. AELI; SELI; FEJI;

例4. A LAD FELL; A LAD SELLS SKIL,
SAID

例5. FAIL FAIL JAIL LAID SAID SAID
DISK SAFE LIFE LEAK SKILL
SLAKE FIELDS DIKES

G和H字键夹在八个基准键位之间，G由左手的食指管理，H由右手的食指控制。

例1. FGFGFGFG JHJHJHJH
 └──────────┘ └──────────┘
 左手 右手

例2. FLAGE HIDE HIS HAFE; GESS
HING

例3. ASDFG HJKLA LKJDH GFDSA,
GIKDSJLHA

3. 空格键和回车键的练习

空格在键盘的最下方，它由大拇指控制。击键方法是：

①手指处于基准键位上。

②右手从基准键位垂直上抬1~2厘米，大拇指横着向下击空格键。

③击键完毕立即缩回。一个空格击一次键。例如：
SALL┌SAILED; ┌FALL┌JAFE┌SAFES┌LIKES
其中┌代表空格键，以后书写时，空格就用一个空白位置来表示。注意：以后输入示例中应该留空格的地方一定要记住敲空格键。

回车键在键盘上用 Return 来表示，为简单，我们用↵表示，它应该由右手的小手指来控制。击键方法是：

①手指处于基准键位上待命。

②抬右手，伸小指击↙键。

③击键完毕立即回归基准键。回归过程中小指要提前弯曲，以免带动其它键而造成错误。例如：ASDF↙ JKL↙
EIGH↙ ASGI↙ LIGH↙

在基础练习阶段，要把指法操作的正确性放在第一位，不要急于盲目追求输入速度。自己不太熟悉的击键动作要反复训练。

4. RTUY和QWPO的练习

RTUY和QWPO这八字符在键盘上的第三排上，位置如图1.1所示。

RT和UY分别由左右手的食指来管理。

例1. 食指 食指 食指 食指
 RRRRR UUUUU TTTTT YYYYY
 左手 右手 左手 右手

例2. FRFRFRF JUJUJUJ FTFTFTF
 JYJYJYJYJ

例3. FRJU GHTY FUJR GUHR HRFJ
 GHTY↙

例4. JUST LADY LARK LASH LAST
 GRAY HALT↙ DUSK DUSTY

例5. AT A FUTURE; DATE; THE JUDGE
IS JUST↙ AT LEAST A YEAR USE THE
REGULAR RATE; REST A LITTLE; AFTER
THAT DATE A SAFE RIDE; FREE RIDES↙
A FIRST TALKER; TAKE A LETTER A
LETTER; SUGGEST FURTHER TESTS;

QW分别由左手的小指和无名指来控制，而PQ分别由右手的小指和无名指来管理。

例1. PQPQPQPQP WOWOWOWOW
 小指 无名指

例2. QWQWQWQW POPOPOPQ QOQOQOQO
PWPWPW↙

例3. FOLLOW THE PATH AS FAR AS IT
GOES; IT IS QUITE↙SHORT; YOU ARE
AWARE; DETER; THAT THE JEWELS ARE
↙ OUR PROPERTY↙

例4. A GOOD DEAL A LOT OF A PAIR
OF ADD TO↙ ALLOW FOR AS FOLLOWS
AS FOR AS IF AS; LET↙ US GO TO THE
PARK; IT IS GOOD FOR THE HEALTH↙
TO GET UP EARLY↙

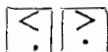
5.; ·shift<>的练习


Shift键的作用是用于控制换档。在计算机键盘上,如果一个键位上有两个字符,那么当需要输入上端字符时就必须

先压住 shift 键,再敲上端字符所在的键。如: <, 当需要输入“<”(小于)符号时,必须先压住 shift 键不松手,敲 < 键以后再松开,那么屏幕上就出现<符号。

由键位分区图可见:shift是由小指控制的。为使操作起来方便,键盘的左右两端均设有一个shift键。如果待输入的字符是由左手控制的,那么事先必须用右手的小指压住shift键,再用左手的相应指头击上端字符键;如果待输入的

字符是右手控制的字键，那么事先必须用左手的小指压住 shift 键，再用右手的相应指头击上端字符键。只有上端字符击键完毕后左右手的指头才能缩回到基准键位上待命。

 分别由右手的中指和无名指控制。

例1. 中指 无名指


例2. shift/<<<<, shift>>>>。

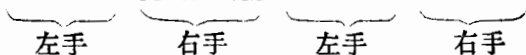
例3. S = (H + I) * L, F > A · OR · S < J

例4. "I LEFT ALL THE KEYS AT THE STAGE."

例5. IF A (I) > K GOTO PRO

6. VBMN 和 ZXC 的练习

VB 和 MN 分别属于左右手的食指管理。

例1. 食指 食指 食指 食指
 VVVVV MMMMM BBBBB NNNNN
 

例2. VMVMVM BNBNNBNB MBMBMB
 NVNVNV ↙

例3. FVFVF JMJM FBFBF JNJNF ↙

例4. I AM A STUDENT. I GET UP AT
 HALF PAST ↙ SEVEN EVERY MORNING. WE
 SHALL HAVE SUMMER ↙ HOLIDAYS FROM
 TOMORROW. THE SNOW IS ABOUT ↙ TWO
 FEET DEEP ↙

ZXC 位于键盘的左端，应该由左手的小指、无名指和中

指来控制。

例1.	小指	无名指	中指
	ZZZZZ	XXXXX	CCCCC

左 手

例2. AZAZAZAZ SXSXSXSX DCDCDCDC ✓

例3. ZOOMZOOMZOOM SEXSEXSEX

CODECODECODE ✓

例4. HOW OLD ARE YOU. YOUR SEX IS...
THE ✓ CHILDREN WERE SITTING ON THE
BRANCHES OF ✓ THE TREE THE BUTTEE-
RFLIES WERE FLYING ABOUT ✓ MERRILY
✓

7. 数码键1~0的练习

数字码1 2 3 4 5 6 7 8 9 0在键盘的上方。由键盘分区图可见，10个数字码分成了左右两大部分，10个数码分别对应左右手的各个手指。根据数据的出现情况可采取两种不同的击键方式。

①通用式击键输入

所谓通用式击键输入，就是象前面介绍的字符一样，按规定指法击键，既有准备阶段，又有回归阶段。这种方式适用于数字和字符混合出现的情况。输入数码时，必须从基准键出发，击键完毕后再回到基准键。

例1. A1A1A1A1A1A : 0; 0; 0; 0 S2S2S2
L9L9L9L ✓ D3D3D3 K8K8K8K8 F4F4F4F4
J7J7J7J7J7 ✓ F5F5F5F5 J6J6J6J6J

例2. 12357924680 3691472580

例3. 1ST 2ND 3RD 4TH 5TH 6TH 7TH

8TH 9TH 10TH✓ 21ST 22ND 23RD 34TH 45TH
56TH 67TH 78TH✓ 89TH 90TH✓

例4. JANUARY 15, 1919 MARC3, 1960 MAY
5, 1854 ✓ JULY 20, 1489 BUS NO.6 ROOM 369
PLATFORM17✓

②基准式击键输入

所谓基准式击键输入，就是将数字1234和7890作基准键位处理。输入数码时，我们可以象基准键那样，将手指轻放在对应的数码键位上，敲完一个数字后不必缩回到原定的字母基准键位，而只需回归到这里的数字基准键位上，这样可以提高输入数字的速度，但指法的对应关系和操作动作还必须按基准键的要求办理。这种方式适用于成批的数字数据输入。

例1. 10293847268360 11223344556677889900✓

例2. 1238 1475 1831 1291 1374 1659 1934✓

2121 2157 2468 2035 2779 2918 2316✓

3472 4238 8765 9610 9173 8501 7296✓

对于数字键的输入，重点还是应该放到通用式击键输入法的练习上。

8. 其它字符的输入练习

除了字母和数字键以外，键盘上还有其它一些字符，如：+ - * / () # ! @ ? & ; ' , \$ % ↑ ↓ ← → ctrl 等等。这些字符的输入也必须按它们各自的分区，用相应的手指按规则击键输入。只要我们熟悉了字母和shift符号的击键原则和方法，那么这些字符的输入是不难体会和掌握的。

例1. + + + + - - - - * * * * * () () ()
() # # # ! ! ! ! @ @ @ @ & & & & ? ? ? ?

... , , , , \$ \$ \$ \$ \$

例2. $A+B * X / (B * * 2 - 4 * A * C) > 100 \cdot OR \cdot S =$
“ABC” ↙

例3. IF A (I) = B (X + 5) THEN ↙

T = A (I) : A (I) = B (I) : B (I) = T

ELSE PRINT “AL” . I ; “=” ; A (I + 1) ; ↙

至此，键盘上的主要字符的输入方法介绍完毕。对于其它字符，亦可参照相应原则进行练习。读者一定要结合自己的实际情况，反复练习，反复体会和琢磨，才能真正掌握这门技术。

1.4 字符输入的综合练习

经过前面的基础训练，读者已经掌握了触觉输入技术的基本方法。现在必须进行综合训练。

在综合训练中，如果遇到一个单词中有多个相同的字符，可以不必击一键就回到基准键，而可以连续敲入后再回到基准键待命。另外，如果下一个待输入的字符离刚击的键最近，也可以不必回到基准键而直接击键；在比较熟练的情况下，还可以跳过基准键而直接转到另一行去击键；同一手指管辖的相邻字键也可以连击。

例1. MUST MUST MUST MUST MUST
MUST ↙

例2. ONCE ONCE ONCE ONCE ONCE
ONCE ↙

例3. LOOP LOOP LOOP LOOP LOOP
LOOP ↙

例4. ABCDEFGHIJKLMNOPQRSTUVWXYZ 连击
10遍。

例5. + - * / () # ! & @ \$ = > < , . : ; ↑ ↓ → ← ↵
连击10遍。

例6. 反复练习BASIC命令和函数字符的输入。

CALL COLOR= $\langle n \rangle$ DATA DEF FN DIM ↵
DRAW n [AT X, Y] FLASH FOR... TO...
STOP... ↵ NEXT FRE (X) GET GOTO GOSUB
GR HGR ↵ HCOLOR= $\langle n \rangle$ HGR2 HIMEN, X
HLIN X1, ↵ X2 AT Y HPLOTX, Y HPLOT
TO X, Y ↵ HPLOT X1, Y1 TO X2 Y2 (TO
Xi, Yi)...) ↵ HTAB (...) IF... GOTO... IF...
THEN... INPUT ↵ INVERSE LET... =... LOMEM
: X MUSIC X, ↵ Y NORMAL NOTRACE ON
 $\langle n \rangle$ GOSUB... ON ↵ $\langle n \rangle$ GOTO... ONEER ONERR
GOTO PLAY ↵ PLOT X, Y POKE X, Y
POP PRINT READ ↵ RESTORE RETURN ROT
= $\langle m \rangle$ SCALE= $\langle n \rangle$ SPEED= $\langle \dots \rangle$ TEXT TRACE
VTAB (...) VLIN AT WAIT XDRAW n (AT
X, Y) ABS (...) ASC ATN (...) CHR \$ (...)
COS (...) EXP (...) INT (X) LEFT \$ (A \$, n)
LEN (A \$) LOG (X) MID \$ (A \$, n I, M)
PEEK (X) POS (...) RIGHT \$ (A \$, n) RND
(X) SGN (X) SIN (X) SORN (XX) SPC (...)
SQR (X) STR \$ (...) TAB TAN USR (X)
CATALOG CONT CTRL-C CTRL-S
DELETE END HOME IN# n LIST LOAD NEW

PR # n PRINT APPEND OPEN CLOSE WRITE
RUN STOP CHR \$ (4)

例7.cdBASE命令输入练习

? /? ? @...SAY...[US/NG PIC...][GET...]
APPEPT...TO...APPEND [BLANK] CANCEL
APPEND FROM... [SDF/DELIMITED] FOR
BROWSE CHANGE...FIELDS... FOR... CLEAR
CONTU COPY CONTINVE STRUCTURE TO
EXTENDED COUNT CREATE DELETE(FILE)
DISPLAY [STRUCTURE] DISPLAY MEMORY
DO WHILE ENDDO EDIT EJECT ERASE
FIND GOTO {RECORD(n)/TOP/BOTTOM}IF...
ELSE...ENDIF INDEX ON...TO...INPUT...TO...
INSERT [BLANK] [BEFORE] FOF JOIN TO...
FOR...(FILD...)FIELD LIST FOR STRUCTURE
LOCATE...FOR...LOOP MODIFY COMMAND
NOTE/* PACK QUIF READ RECALL
RELEASE REMARK RENAME...TO...REPLACE
...WITH...REPORT [FORM...] SAVE SELECT
PRIMARY/SECONDARY SET ALTERNATE
ON/OFF BELL CARRY CONFIRM COLON
CONSOL DATE DEBUG DEFAULT ECHO
ESCAPE EXACT FORMAT HEADING INTE-
NSITY LINKAGE MARGIN RAW SCREEN
STEP TALK SKIP SORT ON ASCENDING
DESCENDING STORE SUM TOTAL ON
UPDATE USE WAIT

例8. 输入下列程序。这是两个人进行四则运算的一个竞赛程序。反复练习10遍，每输完一遍记下输入时间，再进行比较，分析有何进步。

```
10 INPUT "输入甲名字:" , A$
15 INPUT "输入乙名字:" , B$
20 IF INT(RND(1)*10)>5 THEN 35
30 C$=A$ : A$=B$ : B$=C$
35 HOME
40 PRINT "现在该" , A$ : "竞赛."
50 C$=A$
60 GOSUB 200
70 K1=K
75 HOME
80 PRINT "现在该" , B$ : "竞赛."
90 C$=B$
100 GOSUB 200
110 K2=K
120 IF K1>K2 THEN PRINT A$ : "胜" : B$
130 IF K2>K1 THEN PRINT B$ : "胜" : A$
140 IF K1=K2 THEN PRINT "***平局***"
150 END
200 K=0
210 FOR I=1 TO 10
220 J=INT(RND(1)*4)
230 IF J<2 THEN 410
240 R=INT(RND(1)*1E4)
250 IF R<1E3 THEN 240
260 S=INT(RND(1)*10)
270 IF S=0 THEN 260
280 IF J=2 THEN 370
290 PRINT "<" , I , ">" , " " , R , "/" , S , "=" ,
```



```

300 INPUT W
310 PRINT
340 IF W=INT(R/S) THEN 550
350 IF W=INT(R/S*10+0.5)/10 THEN 550
360 GOTO 530
370 PRINT "<" , I, ">" , " " , R, "*" , S, "-" ,
380 INPUT W
385 PRINT
390 IF W=R*S THEN 550
400 GOTO 530
410 R=INT(RND(1)*1E4)
420 IF R<1E3 THEN 410
430 S=INT(RND(1)*1E4)
440 IF S<1E3 THEN 430
450 IF J=1 THEN 500
460 PRINT "<" , I, ">" , " " , R, "-" , S, "-"
470 INPUT W
475 PRINT
480 IF W=R-S THEN 550
490 GOTO 530
500 PRINT "<" , I, ">" , " " , R, "+" , S, "-" ,
510 INPUT W
515 PRINT
520 IF W=R+S THEN 550
530 PRINT "错误, 再试. 总分: " , J
540 GOTO 570
550 PRINT "正确, 很好! 总分: " , J
560 K=K+10
570 PRINT K
580 NEXT I
590 RETURN

```

1.5 输入速度的强化训练

介绍输入技术的目的是为了**提高输入字符的准确率和速度**。在综合训练的基础上，我们要有意识的对输入速度进行强化训练。具体措施是：

1. 集中精力，排除干扰，静下心来进行输入训练。

2. 反复练习输入一篇稿子，直到达到一定速度并完全没有错误为止。训练中每练一遍都记下时间并进行比较。容易出错的地方反复练习。

3. 使用综合训练中用到的指法击键，有意识的训练手指的灵活性和提高动作频率，以形成条件反射。

4. 注意观摩和学习别人的好姿势，好动作和好经验。

5. 以文稿的意群为单位进行快速阅读，要把心智技能同动作技能的练习结合起来。当手指输入第一个单词时，眼睛就应该注视到下一个单词了，这样做到忙而不乱，细而不断。

6. 多记多想多练。力争每天能够按严格要求训练一小时，坚持一百天，定能大大的提高输入速度。

如何衡量我们的输入速度是否合乎要求呢？对一般打字员来说，要求每分钟能够正确输入200个字符（包括空格，回车）就行了。但作为一个优秀的录入员，则要求每分钟至少能够正确输入650~700个英文字符（折合英文单词约130~140个）。根据这个计算标准，训练一百天后就可以考查我们的输入速度属于哪一档次了。

以上介绍的方法也适用于汉字的编码输入。如果我们能够达到每分钟输入60~100个汉字也是可以的。利用五笔字

型输入，任给一篇稿子每分钟可输入140余个汉字，这是相当可观的速度了。

习 题 一

用本书中所介绍的程序进行练习。要求每天完全正确地输入一例，记录操作时间，注意速度和准确性。如果输入有误，则再重复练习。直到完全正确后才换另一程序来练习。请如此坚持100天。

第二章 汉字的编码输入方法

汉字历史悠久，源远流长，是我国人民交流思想、传递信息的主要工具。要利用计算机进行汉字处理，非得解决汉字的输入/输出技术不可。从目前的情况来看，最关键的又是汉字的计算机输入问题。本章介绍三种已被实践证明是行之有效的汉字输入方法，这些输入方法的指法要求同第一章完全一样。

2.1 汉语拼音输入法


对于年轻的读者来说，恐怕汉语拼音输入法是最简单的一种汉字输入法了。因为只要我们能够用汉语拼音来拼写汉字，就很容易掌握汉字的输入方法。


1. 汉字的输入

我们要输入汉字，首先必须进入中华学习机的汉字系统。最简单的方法是按下键盘上的“中文”键。在汉字系统状态下，键入“F2”键（或CTRL-L），屏幕提示行（在显示屏的下面）显示“拼音”字样，表示进入了拼音输入方式。

中华学习机的拼音输入是全拼音输入方式，即要查找一个汉字，必须将该汉字的拼音按声母、韵母依次键入（至多前6个字母），系统根据输入的字母在状态提示行上显示出所有的同音节汉字，包括不同声调的汉字。而在IBM PC等

微机上，通常使用压缩码，使得汉语拼音输入法的敲键个数尽可能的少。中华学习机不具备这一功能，但这对汉字的拼音输入没有任何本质的影响。

使用26个字母键作为输入码，查找一个汉字最多键入6个字母，大小写字母均可。当敲入汉字拼音的第一个字母后屏幕上提示出以该字母打头的音节中最常用的6个汉字供用户选择。如果所需汉字已经出现在提示行上，就直接键入该汉字左边的编号数字，这样，该汉字就取到当前光标位置上。若没有，则继续依次键入后继的字母。当键入所有字母（至多6个）后，屏幕提示行中仍然没有出现所需汉字，则敲  键继续查找后面的同音汉字，直到计算机发出“嘟嘟”声，表示同音节汉字全部提示完毕。

若提示的汉字走过了头而需要返回查看前幕的汉字时，则敲  键，即显示前一幕提示的6个汉字，如此反复操作可一直返回到同音节字中的第一幕汉字。

在向前或向后查找中，一旦提示行中出现了用户需要的汉字，就键入此汉字前面的序号，该汉字就会自动取到内存中并在屏幕上的光标处显示出来。若需要连续取字，可只管敲入汉字的序号就行了。

在输入过程中，若拼音码敲错了，可敲“<”键回退抹掉，使状态提示行上最后敲错的拼音字母被删除，或敲空格键，可删除已输入到状态行的所有拼音字母。再敲“<”键，将已键入到内存中的最后一个汉字删除掉。

例如：输入中华学习机的“中”，操作步骤是：

- ①进入汉字系统后敲F2键进入拼音输入方式。
- ②敲入“中”字的汉语拼音字母“ZH”，屏幕提示：

拼音：ZH 1 这 2 主 3 中 4 种 5 着 6 争

显然，我们需要的“中”字已经出现了。

③敲序号取字。这时敲入数字3，“中”字便取到了。

注意：①凡是要用拼音输入汉字，事先必须敲 F2 进入拼音输入方式。

②凡是要输入汉字以外的符号，如英文字母，必须敲 F1 键进入字符输入方式。

2. 特殊符号的输入

在实际应用中，经常涉及到一些特殊符号，如：①②③， \geq Σ 以及 \perp \top \vdash \dashv 等。这些已作为图形符号固化在机器中。查找这些符号通常只能用区位码方法。中华学习机中提供了比较简便的方法，使我们在拼音方式下也能够方便地取到这些图形符号。具体查找方法是：

在选择“F2”进入拼音方式后，键入-（减号）、=（等号）、/（斜线）可以选择输入标点符号、算术运算符号和制表符等。这些特殊符号如下：

敲入-（减号）可输入的符号有：

， 。 、 : ? ! ; “ ” ‘ ’ …

- · [< > 《 [[]] [[{ }] }

敲入=（等号）可以输入的符号有：

¥ % + - × ÷ ± = ≈ > < ≥ ≤ / ([)] Σ ~ ①②③④⑤⑥

⑦⑧⑨⑩

敲入/（斜线）可输入的符号有：

- | [] \top \vdash \dashv \perp \top +

2.2 区位码输入法

在汉字系统状态下，敲“F3”键（或CTRL-W）后，屏幕状态提示行出现“区位”字样，表示当前处于区位码输入方式。

区位码输入法使用国家标准GB2381—80中规定的区码和位码值来输入汉字。这种输入方式的特点是：仅使用0~9共10个数字键；每个汉字和图形字符只需用4位数字表示，前二位表示汉字所在的区号，后二位表示汉字所在区号中的位号。国家标准规定，00~10区为图形字符，16~55区为国家标准一级汉字，56~87区为国家标准二级汉字，每区有94个汉字或字符，其位号为01~94。用区位码来输入汉字是简单的。只要在区位码输入方式下，敲入某汉字的区位码值，所需汉字就自动取到屏幕光标处。

在区位码输入汉字的过程中，如果输入码敲错，可用“◁”键来删除。如果输入的区位码值不在国标码所规定的范围内，系统会发出“嘟嘟”声警告用户，并拒绝接收这样的区位码，等待用户重新键入或删除修改。

我们仍然以中华学习机的“中”字为例，说明如何用区位码来输入汉字。

①敲F3键进入区位码输入方式。

②“中”字在第54区中的第48位，因而其区位码值是5448。

③敲入“中”字的区位码值5448，则屏幕上立即出现“中”字。

注意：要利用区位码来输入汉字，那么我们必须记得汉字的

区位码值。

2.3 五笔字型输入法

以上介绍了拼音、区位两种汉字编码输入方法。拼音输入法受发音和音调的限制，输入速度受到影响。区位码输入法要求用户记住汉字的区位码值，记忆量太大。这些缺点使这两种输入方法的实用价值受到一定限制。现介绍一种编码简单、记忆量少、规律性强、处理字多、输入速度快的汉字编码输入方法——五笔字型输入法。它是目前四百多种汉字编码输入方案中最受欢迎的输入方法之一。各种微机、M-2401打字机都可用五笔字型输入法。CEC-I中华学习机上也实现了汉字的五笔字型输入。这就使更多的政府机关、学校、企事业单位都能用微机进行汉字的编辑处理。

2.3.1 五笔字型软件的装入

拼音、区位两种输入方法是在中文方式下键入F2及F3即可进入。由于早先的中华学习机汉字系统还没有定义专门的功能键进入五笔字型方式，所以五笔字型的系统软件直接受软件命令的控制。它的装入有两种方式：

①将五笔字型系统盘插入驱动器，开机启动。

②在BASIC提示符“_”下，键入PR#6命令来启动。

这两种方式均可以进入五笔字型所需要的汉字系统。在汉字系统状态下，击“F4”或“F5”键后，屏幕状态提示行出现“五笔”字样，表示当前已处于五笔字型输入方式。汉字WORDSTAR中可以直接使用五笔字型输入法来输入汉字，用法请参见第三章。

2.3.2 汉字的结构分析

要利用五笔字型方法来输入汉字，那么必须懂得汉字的编码规则以及如何把一个汉字拆分成五笔字型所能识别的形式。为此，我们先来看看汉字的笔划结构。

对于一种具体的汉字输入编码方案来说，其编码的基本要求是：键数少、码长短、输入效率高、重码少、编码规律性强、规则简明。为达此目的，五笔字型输入法将汉字分为三个层次：即笔划、字根、单字。所谓笔划就是人们书写汉字时连续写成的一笔。所谓字根就是由若干笔划复合连接交叉形成相对不变的一种汉字结构称为字根。字根又称为字素、部件、字元或构件等等。

如果汉字的分解以笔划为编码单位，那么一个汉字最多的笔划可能有二十多笔，就要击键二十多次才能完成一个汉字的输入，这样的输入速度无疑是太慢了。如果以字根为分解单位，那么平均最多只击四次键即可输入一个汉字。我们以“湖”字为例来看汉字的分解：

汉 字	字 根	笔 划
湖	讠	、 丶 丨
	古	一 丨 丨 丨 一
	月	丨 丨 一 一

由上例可见，若分解到笔划，则编码输入“湖”字要击12次键；若分解到字根，则只须击键三次即可。

汉字的部件（字根）有几百种，可以归并为几十类，正

好相当于拼音文字的几十个字母，正好适用于国际通用微机键盘，正好发挥双手十个指头的作用。尽管汉字是由点、横、撇、捺、折等若干笔划组成的，但分拆到部件一级基本上就能如意地输入汉字了。它基本上解决了击键少、码长短、输入效率高、重码少的问题。

1. 汉字笔划

我们把书写汉字时不间断地一次连续写成的一笔叫做汉字的笔划。对汉字进行分析后，可将其笔划归并为横、竖、撇、捺、折五种。为了便于记忆和应用，根据使用频率的高低依次用1、2、3、4、5作为这五种笔划的代号。如表2.1所示。

表2.1 笔划及其代号

代号	笔划名称	笔划走向	笔划及其变形
1	横	左→右	— — ㄠ
2	竖	上→下	丄
3	撇	右上→左下	ノ ノ
4	捺	左上→右下	ㄨ ㄨ
5	折	带转折	乙 ㄥ ㄣ ㄩ ㄚ ㄜ ㄝ ㄞ ㄟ

从表中可以看出笔划及其代号，除基本笔划外，还对基本笔划可能变化的形态结构进行了归并。即是说在汉字的形态结构中，其基本笔划“一、丨、ノ、ㄨ、乙”常因笔势和结构上的匀称关系而产生某些变形：或一带笔而变成钩，如：

“丨”变成“J”，“一”变成“一”；或多了一个转折如：“了”、“乚”；或基本笔划的长短不一致等等。但这些都影响拆字的规则。因此，把“、”归于“\”，“丿”归为“一”看成是运笔方向相同而笔形稍有变异的结果，所以把它们归为同一类是恰当的。

掌握笔划间的关系，对非基本字根的拆分和识别码的取法是非常有用的。

2. 字根结构

一切汉字都可以由基本字根拼合而成。根据字根中五种笔划彼此之间的关系，可分为单、散、连、交四类。

单——指五笔划自身，即一丨ノ\乙。

散——指构成字根的笔划之间有一定的距离。如：沙、二、八等。

连——指构成字根的笔划是连接的，可以单笔与单笔相连，也可以笔笔相连。如：丁、人、弓等。

交——指组成字根的笔划是互相交叉的，如：十、力、又、车等。

五笔字型键盘字根请参见图2.1。

3. 汉字的字型结构

根据汉字的整体轮廓和组成汉字的各个部分之间的关系，我们可以把成千上万的汉字分为三种组成类型：左右型、上下型、杂合型。并视各型汉字的使用频率的多少分别给它们编写为1、2、3。如表2.2所示。

由表可见：所谓左右型，是指汉字基本上是左右部分组成；上下型则是汉字由上下部分组成；而杂合型是指没有明显的上下或左右结构的那种汉字。表中的杂合型又叫独体字。左右、上下型为合体字。两部分合在一起的汉字叫双合

表2.2 汉字组成类型

字 型	字型代号	字 例
左 右	1	江 浏 给 到
上 下	2	字 空 花 型
杂 合	3	因 凶 道 乘 太 重 天 且

字、三部分合在一起的汉字叫三合字。一般合体字只分到三合字。由于二、三合体字的字根较少，不足四码，所以有必要补加一个字型信息。对于可以拆分为四个（至多四个）部分的汉字，其信息量已足够，就不必考虑字型信息了。

1型：左右型汉字

左右型汉字有以下两种情况：

①双合汉字的左右结构。左右字根间有一定的距离。如：明、现、肚、理、咽。而咽的右边是杂合型，但整个汉字是左右型结构的。

②三合汉字的左中右结构。这种汉字由左中右三部分构成，如：侧、哪。另一种三合汉字，虽然是三个部分构成，但并不是左中右结构。如：别、谈。这两种三合汉字的字型结构都归为左右型结构。

2型：上下型汉字

上下型汉字也分为双合字和三合字：

①双合字的上下结构。字根的上下间有一定距离。如：军、声、字、节、旦等。

②三合字的上、下结构。这种汉字通常由上中下三部分

或者三部分按上下排列而构成。如：意、想、花。

3型：杂合（单体、内外、包围型）

三型汉字是指汉字没有明显的左右或上下关系。如：团、同、这、斗、周、建、飞等。

汉字的字型特征可以作为识别汉字的一个重要依据。如：吧、邑，它们的字根一样，但结构不一样。在输入不足四码的汉字时，除字根输入外，还必须输入字根的排列方式，这样才能准确的识别汉字。

由前述可知：笔划可以组成字根，字根可以组成汉字。笔划和字根之间存在单、散、连、交的关系。同样，字根与汉字之间也存在单、散、连、交的情况。

基本字根在组成汉字时，其字根间的位置关系也分为单、散、连、交四类：

单——指组成汉字的基本字根本身就是一个独立的汉字。如：口、木、山、田。

散——指构成汉字的基本字根之间可以保持一定的距离。如：吕、足、汉、汗、照。

连——指一个基本字根连单笔划构成汉字。如：“自”由“丿”连“目”而成，同理：千、且、久。

交——指几个基本字根交叉套迭之后构成的汉字。如：申、里、韦。

4. 基本字根

五笔字型汉字编码方案采用字根拼形输入方式，将字根象搭积木一样，即可拼合成汉字。例如：“木子”拼合成“李”，“日月”拼合成“明”，三人为“众”等汉字都由几个字根拼合而成。把字根科学地安排在计算机键盘上，就形成了字根键盘，如图2.1所示。例如：

S键上有“木”，B键上有“子”，击S和B再加一个空格键，就输入了一个“李”字。

W键上有“人”，击三个WWW再加一个“U”键就可以输入“众”字。这里的“众”字的“U”键就用到了字型和末笔编码。

读者可能已经发现了，字根的个数很多，如果所有字根都安排在键盘上，必将远远超过键盘的“容量”。因而必须进行筛选。我们把那些组字能力强（即组字频度高），日常应用中出现次数多（实用频度高）的字根挑选出来作为基本字根。而凡是不属于基本字根的字根就必须按规则拆分成基本字根的集合。这样，字根的个数就大大减少了。在五笔字型编码输入法中，优选出了130种基本字根，形成了《五笔字型汉字编码方案》的字根总表，这130种基本字根已在图2.1中给出。凡是以后涉及到字根这个术语，均是指基本字根。例如在拆分：补、祥、促、社、玻、护、都不能按一型的二合汉字来拆分。因为“衤、衤、足、皮、户”都不是基本字根。所以：

补：拆为“衤 亻 卜”，而不能拆成“衤 卜”。

玻：拆为“王 户 又”，而不能拆成“王 皮”。

护：拆为“扌 尸”，而不能拆成“扌 户”。

促：拆为“亻 口 心”，而不能拆成“亻 足”。

社：拆为“衤 土”，而不能拆成“衤 土”。

对于汉字的结构，归纳起来要记住以下几点：

①只有“散”的汉字才可能分为上下、左右型。

②“连”与“交”的汉字只能属于杂合型（3型）。

③不能分上下、左右的汉字一律属于杂合型（3）。

④凡不是基本字根的“字根”都必须拆分成基本字根。

2.3.3 末笔字型交叉识别码


字根是组成汉字的基本单位。但汉字中往往存在这种情况：同样的字根，但由于相对位置关系不同而组成不同的汉字。如：同样是字根“日”和“九”，可以组成晃和旭两个不同的汉字：“口”和“八”可组成：只和叭。这样，如果只是敲入字根而不指出它们的位置关系，那么就不能唯一地确定一个汉字。为此，我们在敲入字根的同时，必须告诉机器字根的排列方式（是上下型，还是左右型），即再补充敲入一个字型代号（左右型代号为1，上下型代号为2）。补敲的这个字型代号称为字型识别码（即左右型码还是上下型的识别标志）。

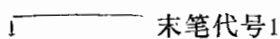
采用字型识别码解决了部分问题，但还没有解决全部问题。从图2.1的键盘字根图就会发现：同一个键位上排列了多个字根，也就是说，这些字根的键码是相同的，如：丁、西、木三个字根都排列在S键上。彳排在I键上。如果我们输入：汀、洒、沐三个字，那么都应该敲字母IS。显然，又撞车了，而且用字型识别码也无法解决。如何办呢？仔细分析和观察“丁、西、木”这三个字根就会发现，它们的最大差别在于末尾笔划。因而，对这类字可以补敲一个末尾笔划的代号，称为末笔识别码。如果一个字既要加字型代号又要加末尾识别码，就太麻烦了，因此将两种代号合并成一个二位数，用个位表示字型代号，十位表示末笔代号，这样统称为“末笔字型交叉识别码”，使两者各起各的作用。在五笔字型中，笔划有五种，字型有三种，末笔字型编码为 $5 \times 3 = 15$ 种。这样，重码率大大降低。追加识别码后，码长从二位加到三位或三位加到四位，相当于原码扩大了25倍。参见表2.3。

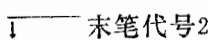
表2.3 末笔字型交叉识别码


字 型 笔 形	左右型 1	上下型 2	杂合型 3
横 1	11 G	12 F	13 D
竖 2	21 H	22 J	23 K
撇 3	31 T	32 R	33 E
捺 4	41 Y	42 U	43 I
折 5	51 N	52 B	53 V

我们来看下面的例子：

例：汉：4 3 5 4 4 1 (ICY, Y为第4区第1个键)
 末笔代号4
 字型代号1 (左右型)

字：4 5 5 2 1 2 (PBF, F为第1区第2个键)
 末笔代号1
 字型代号2 (上下型)

华：3 4 5 5 1 2 2 2 (WXFJ, J为第2区第2个键)
 末笔代号2
 字型代号2 (上下型)

同：2 5 1 1 2 3 1 3 (MGKD, D为第1区第3个键)
 末笔代号1
 字型代号3 (杂合型)

↓ _____ 末笔代号1

本：1 4 1 1 1 3 (SGD, D为第1区第3个键)

↑ _____ 字型代号3 (杂合型)

↓ _____ 末笔代号4

东：1 5 4 3 4 3 (AII, I为第4区第3个键)

↑ _____ 字型代号3 (杂合型)

对上面的例子，我们给出如下说明：

汉：它的字型结构为左右型，字根为“彳、又”，它的末笔为“丶”，末笔字号为4，字型代号为1，“41”就是其所组成的末笔字型交叉识别码，对应的键名字母为“Y”，所以“汉”应键入字母“ICY”。

字：拆分为“宀”和“子”，末笔为“一”，代号为1，字型结构为上下型，代号为2，“12”编码为对应的末笔字型交叉识别码，它的键盘字母为“F”，所以，“字”的五笔字型编码对应的输入字母为“PBF”。

本：它的末笔为“一”，代号为1。它的字根与单笔划相连，字型结构为杂合型，代号为3，“13”识别码对应的键盘字母为“D”，所以，“本”的五笔字型编码对应的输入字母为“SGD”。

有的汉字字根不足四个时，就要键入末笔字型代码以区别字根相同而字型和末笔不同的汉字。如：邑、吧，它们的字根相同，但字型不同，为了区分这两个汉字，必须用末笔字型交叉识别码来判别。

邑：拆分为“口巴”，末笔为“丨”，代号为5，其字型为上下结构，代号为2，所以末笔字型交叉识别码为“52”，对应的字母为“K”。五笔字型编码为“KCB”。

吧：拆分为“口巴”，末笔也是“丨”，代号为5，其

字型结构为左右型，代号为1，所以末笔字型交叉识别码为“51”，对应的字母为“N”。五笔字型编码为“KCN”。

2.3.4 单体结构拆分原则

由基本字根通过连或交的关系可以形成汉字的“单体结构”。那么倒过来，如何把单体结构的汉字拆分成字根呢？拆分的原则是：

①对于连笔结构，拆分为单笔与基本字根。如：“自”拆成为“丿”和“目”、“天”拆成为“一”和“大”。

②对于交叉结构和交连混合结构，按书写顺序拆成几个已知的最大字根，以增加一笔就不能构成为字根来决定笔划分组。如：

朱：只能拆成“亠小”而不能拆成“牛、八”。因为“亠”多一笔就是“牛”，牛不是基本字根。也不能拆成“亠、丨、小”，这样把“丿”笔划割断了，并且不是最大字根。只要属于连笔结构时，就不能按交、连结构拆分。在具体拆字过程中要掌握以下四点：

1. 能散不连

如果一个单体结构可认为是几个字根“散”的结构，就不要视为连的结构。如：“百”拆分成“丿日”就比拆分成“一白”好。“午”拆分成“亠十”，都不是单笔划，为上、下结构。“非”拆分成“彳 丨 彳”，不是单笔划，为左右关系。

除单笔与基本字根存在连的关系外，一般不存在连的关系。

2. 兼顾直观

拆分时尽量照顾直观性。如：“羊”拆分成“丩羊”比

拆分成“亠丰”要直观得多。自：‘目，舟：‘舟、生：‘主等，都比较直观。

3.能连不交

一个单体结构能按连的结构拆分，就不要按交的结构拆分。如：

天：拆成“一大”，不能拆成“二人”。

失：拆成“丿夫”，不能拆成“宀大”。

丑：拆成“乙土”，不能拆成“刀二”。

4.取大优先（能大不小）

在各种可能拆法中，按书写顺序，每次都拆分出尽可能大的字根，以保证拆分出的字根数最少。如：

缶：拆分为“亠山”，不能拆为“宀一匚”。

平：拆分为“一亠丨；”，不能拆分为“一乚十；”。

重：拆分为“丿一日土”，不能拆分为“千日二”。

总之，取大优先以拆出的字根数最少的那种拆法为正确。在拆出字根相等的条件下，散比连优先，连比交优先。为了让用户尽快地掌握拆分方法，列出常见基本字根拆分示例。如表2.4所示。在输入汉字编码时，经常需要查阅此表。

在汉字的拆分中独体字是难点，它不象有的二合型汉字比较直观。有的独体字拆分成字根时容易产生错觉。如：

兔：拆分为“𠂆口儿”，不能拆分成“𠂆口儿”。

东：拆分为“匕小”，不能拆分成“匕丨八”。

在拆分时一定要注意字的结构和掌握四点拆分原则，只要对单体结构的拆分原则掌握好了，五笔字型输入就不成问题了。掌握了上表所列常见基本字根拆分方法后，对独体字的拆分也就容易掌握了。

注意：按汉字的通常书写笔划顺序拆分出来的字根不一

表2.4 常用字拆分示例

横起笔类		丙: 一 冂 人		戈: 七 丿		冉: 冂 土	
手: 三十	由: 一 冂 人	臣: 亡 丨 冂 丨	冎: 一 冂 丨	匹: 冂 儿	巾: 冂 丨	央: 冂 大	冎: 冂 丨
彳: 丰 冂	本: 木 一	巨: 冂 冂	束: 一 口 小	瓦: 一 乙 乙	央: 冂 大	四: 四 土	果: 日 木
子: 一 寸	束: 一 口 小	无: 冂 儿	束: 一 冂 小	无: 冂 儿	日: 日 十	史: 口 又	里: 日 土
夫: 二 人	木: 木 乙	牙: 冂 丨 丿	平: 一 艹 丨	戒: 戈 卩	至: 一 丿 幸	虫: 口 丨 一	皇: 口 丰
无: 二 儿	来: 一 米 人	死: 一 夕 匕	巫: 工 人 人	歹: 一 夕 匕	死: 一 夕 匕	电: 日 乙	鬼: 白 匕
正: 一 止	世: 甘 乙	爽: 大 又 又 又	甘: 甘 二	于: 一 十 人	爽: 大 又 又 又	申: 日 丨	禺: 日 冂 丨
酉: 酉 一	其: 甘 三	夫: 一 艹 人	革: 甘 手	与: 一 乙 一	夫: 一 艹 人	少: 小 丿	冎: 冂 冂
下: 一 卜	辰: 厂 二 乙	屯: 一 口 乙 止	灭: 一 火	屯: 一 口 乙 止	屯: 一 口 乙 止	冎: 冂 冂	见: 冂 儿
击: 二 山	灭: 一 火	夷: 一 弓 人 厂	太: 大 丶	夷: 一 弓 人 厂	夷: 一 弓 人 厂	免: 口 儿	
未: 二 小 木	夫: 大 丶	严: 一 丩 厂	文: 大 丿 乙	严: 一 丩 厂	严: 一 丩 厂		
末: 一 小 木	兀: 一 儿	互: 一 互 一	尤: 犬 乙	互: 一 互 一	互: 一 互 一		
丰: 二 丨 一	尤: 犬 乙	豕: 豕 乙 冂	万: 厂 乙	豕: 豕 乙 冂	豕: 豕 乙 冂		
井: 二 月 丨	万: 厂 乙	成: 厂 乙 乙 冂	页: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
韦: 二 乙 丨	页: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
玉: 干 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
戔: 十 戈	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
来: 三 小 三	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
非: 三 丨 三	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
考: 土 丿 一 乙	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
止: 十 乙	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
才: 十 丿	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
求: 十 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
疋: 冂 止 乙	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
丐: 一 卜 乙	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
亚: 一 乙 一	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
事: 一 口 冂 丨	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
吏: 一 口 又 一	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
出: 一 口 丨 一	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
再: 一 冂 土 乙	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
西: 一 冂 土 乙	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		
市: 一 冂 丨	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂	成: 厂 乙 乙 冂		

撇起笔类

矢: 丿 大
失: 乙 人
千: 丿 十
壬: 丿 土
丢: 丿 土 乙
重: 丿 一 四 土
重: 丿 一 日 土
垂: 丿 一 冂 土
牛: 乙 丨
彳: 乙 止
帝: 乙 冂 丨

竖起笔类

鹵: 卜 口 又
卩: 丨 乙 冂
鬼: 月 乙
且: 月 一
冎: 冂 冂 三
县: 月 一 乙
命: 冂 冂
丹: 冂 乙
册: 冂 冂 一

定完全正确；拆分出来的每个部分一定要是基本字根。

2.3.5 基本字根及其排列

五笔字型有130个基本字根，那么这些字根如何排列在计算机键盘上呢？排列的规律和原则又是什么呢？本节就来讨论这些问题。

1. 键名及其排列

从图2.1中已经看到了，五笔字型方案中130种字根分别排列在25个英文字母键上，每个键位上安排了2—6个字根。这130种基本字根，按照其首笔笔划代号，兼顾键位设计的需要，共分成5个区，每个区又分成5个位，这样得到：11—15，21—25，31—35，41—45，51—55共25个键位。

我们把11—55这种数字称为键位代码，又在同一键位上的所有字根中选出一个具有代表性的字根作为该键的户主，称为键名，参见图2.2。各区的键名如下，

1区：王土大木工

2区：目日口田山

3区：禾白月人金

4区：言立水火之

5区：已子女又夕

仔细观察各区的字根就会发现：1区所有字根的首笔划都是以横开始的，而“横”的代号刚好也是1。2区内所有字根的首笔划都是以竖为开始的，“竖”的代号为2。其余类推。这样，各区字根的规律性也就知道了。注意：

①当键名字作为字根组字使用时，仍和其它字根一样对待。如果把键名字作为输入的汉字，那么连击四次即可。如“金”连击“QQQQ”、“人”连击“WWWW”。

金 35 Q	人 34 W	月 33 E	白 32 R	禾 31 T	官 41 Y	立 42 U	水 43 I	火 44 O	之 45 P
工 15 A	木 14 S	火 13 D	土 12 F	王 11 G	目 21 H	日 22 J	口 23 K	田 24 L	； ；
乙	纟 55 X	又 54 C	女 53 V	予 52 B	己 51 N	山 25 M	< ,	> .	? /

图2.2 键名与键位

如果作为字根时，仍击相应字母，如：“杜”击“SF”加一个空格即可。

②键名字作为字根时，不能再拆分。

2. 字根的分区

五笔字型输入法将130个字根助记词分为5区，每区5个键位。为了帮助记忆，下面给出五笔字型编码方案的创始人王永民先生所写的助记词。

第一区的字根键位排列如图2.3所示，其字根键位助记词为：

11G 王旁青头戈（兼）五一。（借同音转义）

12F 土土二千十寸雨。

13D 大犬古石厂，（指羊字底“年”）

14S 木丁西

15A 工戈草头右框七（草头“艹”，右框“]”）



图2.3 第一区字根键位

第二区的字根键位排列如图2.4所示，其字根键位的助记词为：

21H 目具上止卜虎皮（“具上”指具字的上部“且”）

22J 日早两竖与虫依。

23K 口与川，字根稀，

24L 田甲方框四车力。（方框指“口”）

25M 山由贝，下框几。

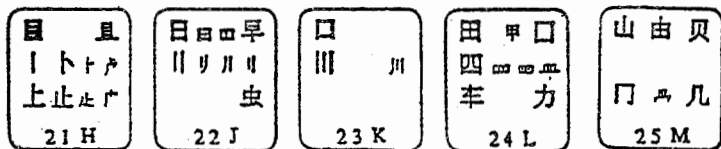


图2.4 第二区字根键位

第三区的字根键位排列如图2.5所示。其字根在键位上排列的助记词为：

- 31T 禾竹一撇双人立（“双人立”指“彳”）
 反文条头共三一（“条头”指“攴”）
 32R 白手看头三二斤，（三二指键位32）
 33E 月彡（衫）乃用家衣底。（家衣底指“豕氏”）
 34W 人和入，三四星，（三四指34）
 35Q 金勺缺点无尾鱼，（指“勺、鱼”）犬旁留×儿一
 点夕，（指“犴、×、儿夕”）
 氏无七（妻）。（“氏”去掉“七”为“氏”）



图2.5 第三区字根键位

第四区字根在键位上的排列如图2.6所示，它们的助记词为：

- 41Y 言文方广在四一，
 高头一捺谁人去，（高头指“古”，谁人去指“讠、圭”）

42U 立兴两点六门病

43I 水旁兴头小倒立。(“シ、ツ、小”)

44O 大业头，四点米。(“业、米”)

45P 之字军盖建道底，(“宀、宀、之、彡”)

摘ネ(示)ネ(衣) (“ネ、ネ”摘除右边的点为“ネ”)

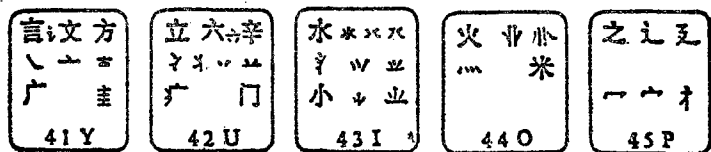


图2.6 第四区字根键位

第五区字根的键位排列如图2.7所示，其中字根的助记词为：

51N 已半已满不出己，

左框析尸心和羽(左框指“尸”)

52B 子耳了也框向上(框向上指“尸”)

53V 女刀九白山朝西。(山朝西指“ヨ”)

54C 又巴马，丢矢矣，(“矣”丢掉“矢”为“又”)

55X 慈母无心弓和匕，(母无心指“母”)幼无力(“幼”去掉“力”为“么”)

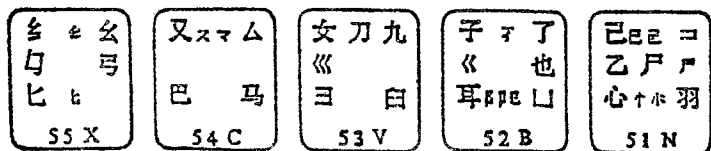


图2.7 第五区字根键位

理解和熟背五个分区的助记词，就能掌握每个键位上的字根。

3. 字根的键位特征

根据图2.1的字根键盘不难看出，键位上的每组字根，不管个数多少，都具有三种特征：

①除字根的首笔代号与其所在的“区号”保持一致外，还有一部分字根的第二笔代号与“位号”一致。

如：王、主，它们的第一笔为横，代号与区号一致，第二笔还是横，代号为1，与位号一致，这种字根的区位号或字根代码为11（G）。

又如：文、方，它们的首笔是捺，代号为4，次笔为横，代号为1，所以区位码或字代码为41（Y）。

②与键名字根外形相近的字根。

如：“王”键位上有相似的字根“主”和“王”；“日”键位上有相似的字根“日”和“口”。

③键位代码还表示组成字根的单笔划的种类和数目。

如：点的代号为4，那么41代表一点“丶”，42表示两点“㇇”，43表示“シ”与“ツ”，44表示四点“灺”等。依此类推，11键表示“一”，“12”键表示“二”，13键表“三”等等。

掌握了以上三个字根键位特征，对字根键位的记忆就不难了。

查找字根时应注意以下几个问题：

①定区：对任何字根，第一笔的笔形代码（代号）就唯一确定了它的键盘上的区号。

②定位：对位号的确定有三种方法：

第一，字根的第二笔代号确定位号，在130种字根中，

第二笔代号确定位号的占90%左右。

第二，位号与键位上的复合散笔字根的笔划数目一致。
如：

一区：11键位表示“一”，12表示“二”，13表示“三”。

二区：21键位表示“1”，22表示“リ”，23表示“川”。

三区：31键位表示“ノ”，32表示“リ”，33表示“彡”。

四区：41键位表示“、”，42表示“ン”，43表示“シ”，44表示“…”。

五区：51键位表示“乙”，52表示“《”，53键位表示“<<<”以及与它们相似的字根。

第三：将基本笔划形状相似的归类在一个键位上。

如：“四方”形字根和“力”都归在24键位上；带两点的“立六辛彡ン彡”都归在42键位上。如表2.5所示。

根据以上确定字根键位的规律和特征，就能将字根的键位唯一确定。

2.3.6 五笔字型编码输入

1. 键名字的输入

从五笔字型键盘字根总图（图2.1）可以看到，每个键位左上角的汉字是键名字，它的特点是组字频度高，形体上有一定的代表性。要输入这些汉字时，只需连击该键四次即可。也可以和其它字根一样用于拼成汉字。

2. 高频字的输入

五笔字型字根键盘上，从11—55共25个键位代码，每个

表2.5 笔划相似的字根

区	位	字母	代码	笔画	键名 基本字根
1 横 起 类	1	G	11	一	王主戈五一
	2	F	12	二	土土二千十甲寸雨
	3	D	13	三	犬犬三羊手长古石厂アナナ
	4	S	14		木丁西
	5	A	15		工戈升廿卅匚七七
2 竖 起 类	1	H	21	丨	目且上止止卜卜尸尸
	2	J	22	丨丨	日日四早リリリ 虫
	3	K	23	川	口川
	4	L	24	川	田甲口四四皿四车力
	5	M	25		山由贝门凡
3 撇 起 类	1	T	31	丿	禾禾竹ノノ夕夕
	2	R	32	夕	白手持夕夕夕夕斤斤
	3	E	33	夕	月月舟夕夕夕乃用豕夕夕
	4	W	34		人亻八八八
	5	Q	35		金钅勺勺勺又儿夕夕夕
4 捺 起 类	1	Y	41	丶	言讠文方广广广 圭
	2	U	42	丶	立辛辛夕夕夕六六六 广
	3	I	43	丶	水灬灬灬夕夕夕 小中业
	4	O	44	灬	火业 灬米
	5	P	45		之之之夕夕夕
5 折 起 类	1	N	51	乙	巳巳巳コ尸尸乙心忄小羽
	2	B	52	乙	子孑耳卩卩卩 了也口
	3	V	53	乙	女刀九白日乙
	4	C	54		又又又巴马ム
	5	X	55		纟纟弓匕匕纟

键位对应一个最常用的汉字，这种汉字称为高频汉字，又称一级简码，它与键名字一一对应。但高频字在键盘上没有直接标出，而只能通过敲键名字和空格键才能显示出来25个高频字对应的键位如图2.8所示。

利用一级简码，要输入如下的汉字：我是中国人、中国的土地、中国的、人民等也就简单多了。例如输入：我是中国人，只需敲入：Q┐J┐K┐L┐W┐就行了。（“┐”代表空格键）

3. 二级简码的输入

二级简码由单字全码的前两个字根代码组成。具有二级简码的汉字，键入前两个字根码再加一个空格键即可。例如：“李”，拆分成“木子”，击“SB”后再敲一个空格就准确地取到一个“李”字。

4. 三级简码的输入

三个简码由单字的前三个字根组成。如果一个汉字的前三个字根码在整个编码体系中是唯一的，一般都选用三级简码。要输入三级简码汉字只要击前三个字根代码，再加一个空格键即可。显然要击四次键，但减少了字根和交叉识别码的判定，这样可以提高编码输入的速度。如“爸、宝、兵”等都是三级简码字。若要输入“爸”，只敲入：“WQC”就可以了，但也可以敲“WQCB”。使用简码输入时需要注意：

有的汉字具有高频字码、二级简码、三级简码、全码四种输入编码，为了提高速度，当然是用高频字码输入最快。如：“经”字，它可以用这四种编码输入中的任何一种方式输入。

5. Z键的用处

我 35 Q	人 34 W	有 33 E	的 32 R	和 31 T	主 41 Y	产 42 U	不 43 I	为 44 O	这 45 P
工 15 A	要 14 S	在 13 D	地 12 F	一 11 G	上 21 H	是 22 J	中 23 K	国 26 L	； ；
乙	经 55 X	以 54 C	发 53 V	了 52 B	民 51 N	同 25 M	< ,	> 。	? /

图2.8 高频字

Z 键叫辅助学习键。如果对五笔字型字根不太熟悉或者对某一汉字拆分难以确定，或者输入一个汉字时，不管拆分未知字根在汉字编码中排在第几，都可以用“Z”键来代替。如果你对汉字字根一无所知，那么将四个代码敲成“ZZZZ”，机器就会自动将6763个汉字从头到尾依次分组显示出来，一行显示5个汉字。现用“型”和“苗”来说明如何利用Z键进行输入。

型：拆分成“一+Z土”四个字根，如果编码没有重码，当敲入“GAZF”时，提示行上显示“GAJF”编码，光标处立刻显示汉字“型”。如果第一个字根不知道，则可敲入“ZAJF”。若第二个字根不清楚，则敲入“GZJF”。其余类似。若四个字根都不清楚，则敲入“ZZZZ”，从头至尾顺序查找。总之，拆分时哪个字根不清楚，就用“Z”代替那个字符。

苗：拆分成“艹田”如果对末笔字型识别码不清楚，也用“Z”代替，敲入“ALZ”后，若没有重码进行选择，那么马上在提示行上显示“苗”的编码“ALF”，光标处立即出现汉字“苗”。

有的又称“Z”键输入法为选择式易学输入法，这给五笔字型输入法的自学者带来了方便。

6. 成字字根的编码输入

五笔字型字根键盘的每个键位上，除键名字根外，还有一些独立的汉字，我们称这些汉字为成字字根。成字字根的输入用公式表示为：键名代码+首字码+次笔码+末笔码如图2.9所示。如：

文：键名41、首笔代码41、次笔代码11、末笔代码41，对应键盘字母应敲：YYGY。

键名：把所在键连打四下

例：言：言言言言，又：又又又又，山：山山山山
 YYYY CCCC MMMM

成字根：报户口 加打第一、二、末单笔画（不足四码，补空格键）

例：西：西-|一，方：方'-乙，厂：厂-|（空格）
 SGHG YYGN DGT

书写顺序

例：
 新：立木斤 (正)
 立斤木 (误)

取大优先

例：
 平：一-| (正)
 一-|十 (误)

兼顾直观

例：
 自：| 日 (正)
 | 日三 (误)

能连不交

例：
 天：一-大 (正)
 二-人 (误)

汉字

键面有

键面无

拆

超过四码：取1、2、3、末字根编码

例：赣：立早女贝；攀：木××手

刚好四码：依次键入即可

例：照：日刀口；到：一-△土

不足四码：字根键入完后，补打“末笔字型识别码”
 （仍不足四码，补空格键）

末笔字型识别码表

字型 \ 末笔形	左右型	上下型	杂合型
横 1	11 (一)	12 (二)	13 (三)
竖 2	21 (丨)	22 (丨)	23 (丨)
撇 3	31 (丿)	32 (丿)	33 (丿)
捺 4	41 (㇇)	42 (㇇)	43 (㇇)
折 5	51 (乙)	52 (乙)	53 (乙)

例：
 汉：43 54 41

又：又

ICY

例：
 会：34 12 54 41

人二△；

WF C U

必：51 31 33

心 J 多

N T E

图2.9 汉字拆分原则

西：键名代码14、首笔代码11、次笔代码21、末笔代码11、对应键盘字母应敲：SGHG。

总之，键位上除键名外，还有成字字根，要输入这种汉字时，只需按成字字根输入公式敲入相应字母即可。

注意：如果字根只有二笔划，则用空格键结束。

7. 识别码输入

除键名字、高频字、成字字根汉字、四个字根汉字输入外，还有不足四个字根汉字的输入，对于这种情况要输入识别码才能确定汉字。

识别码就是末笔字型交差识别码。识别码由该字末笔代号与它的字型代号结合而成。如：“英”，末笔代号为4，字型代号为2，识别码为42，对应的键字母为U。在用识别码时，应注意末笔的掌握。

①所有包围汉字中的末笔，规定取被包围的那一部分的末笔。例如：

“远”，其末笔为“乙”代号为5，字型代号为杂合型3，它的识别码为53，对应的键字母为V。

“国”，其末笔为“、”，代号为4，字型代号为3，识别码为43，对应的键字母为I。

②凡是除用“刁、九、力、七”字根作为末笔字根识别的以外，其它一律为“折笔”识别。例如：

仇：34 53 51 (WVN)

化：34 55 51 (WXN)

识别码的确定，应注意字型结构和书写的笔顺，准确确定末笔代码。

8. 词汇的编码输入

笔划构成了字根，字根构成单字，以单字为基本单位组

成成千上万的词汇。五笔字型方案体现了中文汉字这一特点，以单字代码为基础，完全依据字形组成了与单字代码码型一致相容的大量词汇代码。好象“中共中央”代表“中国共产党中央委员会”，或英文中的DOS代表 Disk Operation System一样，每一个词条汉字的前一两个字根组成一个新的代码，来代表整个词汇。这些词汇全部来自汉字的字形，一般不用死背便能“见词知码”。中华学习机的五笔字型输入方法版本较多，各版本的词条有所不同，但它们的组词编码基本一致。有的五笔字型版本根本没有词条输入。

所有的词汇码都是四码，其码型与单字码型完全一样。词汇代码的取码规则分为以下四种情况。

①双字词

双字词在汉语词汇中有相当大的比例。双字词的编码取法为：分别取两字的前两个字根代码，共四码组成。如：

计算：言十竹目（YFTH）；

机器：木几口口（SMKK）；

汉字：彳又宀子（ICPB）；

实践：宀彡口止（PUKH）。

②三字词

前两字各取其第一码；最后一个字取其前两码，共四码组成。如：

计算机：言竹木几（YTSM）；

解放军：宀方宀车（QYQL）；

共青团：甘圭口十（AGLF）；

共产党：艹立小宀（AUIP）。

③四字词

每字各取其第一码，共为四码。如，

汉字编码：シ宀纟石 (IPXD)；

程序设计：禾广言言 (TYYY)；

科学技术：禾小才木 (TIRS)；

五笔字型：五竹宀一 (GTPG)。

④多字词

取第一、第二、第三及最末一个字的第一个码，共为四码，就能构成多字词的词汇编码。如：

电子计算机：日子言木 (JBYS)；

中华人民共和国：口イ人口 (KWVL)；

中国人民解放军：口口人宀 (KLWP)。

由以上例子可以看出，词汇码与单字码相比，不用任何特殊标记。那么五笔字型怎样使大量的字和词的编码能相容共存呢？因为在五笔字型的键数及码长的条件下，共能容纳 $25^4 \approx 39$ 万个可能的编码。其中汉字单字码及其简码占用 12000 余个，还有着大量的空闲码位。但中华学习机内存和驱动器受限制，各版的五笔字型词汇编码数量不同。总之：当你输入常用的词汇编码时，若没有该词汇，则马上改为单字编码输入。单字码与词汇码有着不相同的分布规律，二者混合使用不用换档。绝大部分情况下不会发生冲突。有了这样的词汇输入，操作起来就方便多了。

2.3.7 重码与容错码的处理

所谓重码，就是敲入键码后出现了多个汉字与之对应。一个好的编码方案，当然重码越少越好。在五笔字型方案中，对重码字是按使用频度分级处理的。若有重码输入时，重码字同时显示在提示行中，而使用频度高的排在第一个位置上。如第一个字就是所需的字，只要继续输入下文，该字

就自动取到光标处。若所需的字不在第一个位置，则敲入该字的位置序号，汉字同样可以取到光标处（类似拼音输入法的取字规则）。这样，重码字只要级别不同，就象没有重码一样，不会影响汉字的输入速度。

为了进一步解决重码问题，将一级字库中出现的不常用的那些重码字的最后一码改用L（24）代替，作为该重码字的容错码。这样处理后，常用字按原码输入，不常用的字按容错码输入，两者都不作任何特殊处理和增加击键次数。在一级汉字中，就基本上解决了重码的问题。

读者可能会问：究竟什么叫容错码？所谓容错码实际就是容易搞错的码或者容许搞错的码，五笔字型方案有四种类型的容错：

1. 拆分容错

个别汉字由于书写顺序因人而异，从而造成拆分顺序容易弄错，如：

长：丨匕、43（正确码）；

长：匕丨、43（容错码）；

长：丨一丨、（容错码）；

长：一丨丨、（容错码）。

动：一云力51（容错码）；

动：二厶力51（正确码）。

秉：丿一ヨ小（正确码）；

秉：禾ヨ43（容错码）。

2. 字型容错

有的汉字的字型分类很难确定，如，

占：丨口12（正确码）；

占：丨口14（容错码）。

3. 定义后缀码容错

所谓定义后缀就是把最后一码修改为24(L)的汉字。主要定义一级字库中频度较低的那个字，使之成为唯一的外码。外码就是将原码的最后一位改为L后的码。如：“喜”和“嘉”的编码为“FKUK”，是一对一的重码，“喜”在提示行的第一位置，不作任何处理就可将其取到光标处，但“嘉”必须敲“Z”才能将它调到光标处。现将“嘉”的原码“FKUK”改为“FKUL”就不重码了。FKUL就是“嘉”的唯一的外码。

4. 五笔字型新旧版本容错

五笔字型的版本不同，它的字根设计和分区也略有不同。为顾及旧用户使用新版本，有关汉字均以容错方式加以解决。

习 题 二

1. 什么是部件（字根）？
2. 五笔字型有多少种字根？
3. 汉字笔划通过分析后，归为几种？其代号是什么？
4. 字型有几种类型？其代号各是什么？
5. 五笔划组成字根时，笔划间的关系可分为多少种？
6. 怎样确定末笔字型交叉识别码？
7. 单体结构分析原则有几种？
8. 拆分以下汉字，并说明归于什么类型？分别写出末笔字型交叉识别码。

申身电失重牛朱天龙书在中斤儿已及交
尸禾关个更丰飞凡反斗丁方丑舟尺才中

9. 什么叫键名字, 键几次才能键入键名字? 字根键盘分几区?

10. 字根的特征有几种? 查字根时应注意什么?

11. 什么叫成字字根? 怎样确定成字字根代码?

12. 什么叫简码? 什么是高频字? 怎样输入高频字? 怎样输入二级、三级简码汉字。

13. 请选一篇文章反复练习输入。试一试自己的输入速度(每分钟能够输入多少个字)。

第三章 文稿编辑软件WORDSTAR

中华学习机CEC-I不但可以用于数据值计算（如：BASIC语言、FORTRAN语言和PASCAL语言）和数据处理（如：cdBASE），还可以用于文稿编辑。本章介绍目前流行的汉字处理软件WORDSTAR在CEC-I机上的使用方法。

3.1 如何启动汉字WORDSTAR

3.1.1 WORDSTAR需要的硬件和软件环境

要利用汉字WORDSTAR来编辑和打印文稿，必须提供一定的硬件和软件环境。

1. 硬件环境

- ①CEC-I中华学习机。
- ②显示器或家用电视机。
- ③至少一个软盘驱动器。
- ④如果需要打印文稿，那么还需要一台打印机（当然要适用于CEC-I且能打印汉字）。

2. 软件环境

CEC-I用到的汉字WORDSTAR保存在一张软盘上，它能够提供汉字的拼音、区位码以及五笔字型输入方式，同时能够提供汉字的编辑和打印等一系列功能。它不需要另外

的操作系统盘，启动之后可以取出系统软盘而插入工作盘，这对于单驱动器情况下的使用来说特别方便。

3.1.2 如何进入功能主菜单

我们要使用 WORDSTAR，必须首先进入系统提供的功能菜单（或称为起始命令）状态。这在打开屏幕电源后有两种方法进入功能主菜单：

1. 冷启动

①将拷贝有汉字WORDSTAR的软盘插入到驱动器，并关好驱动器的“门”。

②打开主机电源开关。稍候片刻，屏幕显示功能主菜单：

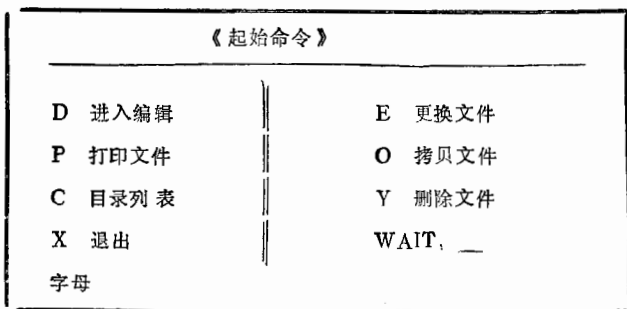



图3.1 功能主菜单

这时光标在“_”位置闪烁，等待我们输入功能代码。

2. 热启动

①打开主机电源使机器处于BASIC提示符“]”状态。

②将WORDSTAR软盘插入驱动器，关好“门”。

③敲入热启动命令PR # 6 ↵（事先必须压下  键使字母处于大写状态），屏幕显示为：

] PR #6 ↙

稍候，屏幕仍然显示如图3.1所示的功能主菜单。

建议最好用热启动，系统不容易出错。

3.1.3 功能主菜单简介

启动WORDSTAR后，屏幕显示《起始命令》，这就是系统的功能主菜单。下面简单介绍一下各菜单项的含义。

D进入编辑——这个菜单项的功能最强。从文稿的建立、插入、删改到编辑排版等一系列操作，都必须首先进入该状态后才能进行后继操作。敲入字母D后系统所处的状态称为编辑状态，其主要功能是对文稿进行编辑处理。

P打印文件——如果文稿需要在宽行纸上打印出来，就可以敲入字母P，从而选用WORDSTAR的打印功能，使文稿按需要的格式打印出来。

C目录列表——列出当前软盘上全部文件的名称及其类型。

E更换文件——更换软盘上指定文件的名称。

O拷贝文件——复制磁盘文件。

Y删除文件——删去磁盘上指定的文件。

X退出——退出WORDSTAR，回到BASIC的提示符状态“]”。

WAIT:- ——光标在此闪烁，等待用户敲入功能代码（菜单项前面的字母）。

这些功能的详细用法将在下面逐一介绍。

3.2 文稿文件的建立方法

一段文字、一篇文章、一个通知,在汉字WORDSTAR中,我们都可以称为文稿或文书。文稿不论长短,在计算机上处理时都必须有一个名字,这种名字称为文件名或文稿文件名。文件名可以用1~4个汉字来表示,也可以用英文字母和数字符号来表示,还可以汉字、字母、数字符号混用。但要注意:文件名字字符个数不能多于8个英文字符或4个汉字。如:

开会通知, 报告, ABC, A1, BA2 都是正确的文件名称。

3.2.1 文稿的输入

文稿文件的建立方法是:

- 1.在主菜单下选择D进入编辑状态
屏幕显示:

D

使用本命令建立新文书文件或更改现存文件。文件名取1—4个汉字,文件名是:

文件名? —

光标在“?”之后闪动,要求用户提供文稿的名字。

- 2.用户回答文稿文件名字

假设希望文稿用“通知”作为文件名,那么我们就敲入“通知”这两个汉字(F₂表示汉语拼音, F₃表示区位码, F₄或F₅表示五笔字型输入方式,我们可以用任何一种方式

来输入汉字)，即文件名?通知↵，屏幕显示：

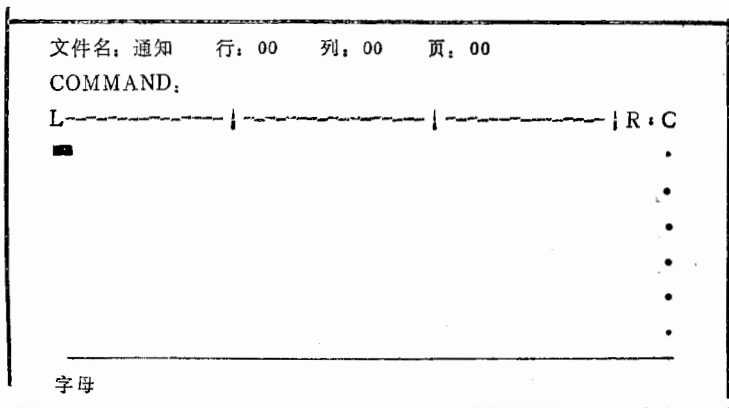


图 3.2

其中：

第一行称为状态行。它给出了当前系统所处的状态信息。

文件名：通知——表示当前要建立的文件的名称是“通知”。

行：00——光标正指向的那一行。由于文稿还没正式建立，所以行号为零。

列：00——光标正指向的那一列。

页：00——目前屏幕上显示的是第几页。由于还没有输入文稿字符，所以这里的行、列、页的值均为零。随着文稿的操作以及光标的移动，这些状态值会随之而动态变化。

COMMAND：——用于显示文稿的操作命令。用户敲入什么命令，这里就提示什么命令。

第三行称为标尺行，它给出了屏幕上显示字符的左右边界，其中：

L——左边界，即输入和显示字符的开始位置。

R——右边界，即输入和显示字符不得超过的右边界。

|——定位点标志。

——光标，表示输入字符显示的位置。

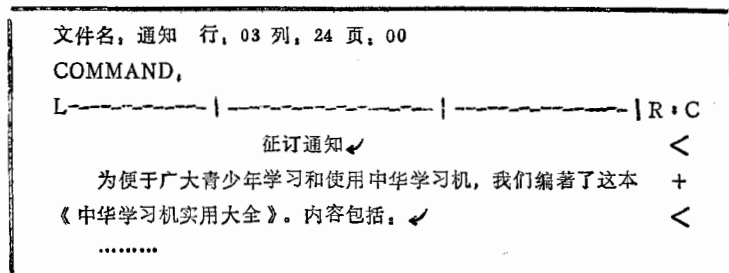
·——空白行标志，表示该行还没有书写任何字符。屏幕上还可能出现的其它标志有：

<——表示一段文稿结束。只有敲入回车键以后才会显示这种段结束标志。

+——表示一行未完，后行是该行的续行。

3. 敲入文稿字符

当回答文件名字以后，系统显示“新文件？”，询问刚敲入的文件名是否是新文件（即磁盘目录中还没有的文件）。我们回答一个“Y”或回车键均可，屏幕显示图 3.2，等待用户敲入文稿。我们以征订通知为例，敲入通知的内容如下所示：



这是输完最后行冒号: ✓后显示的状态。

3.2.2 文稿的存盘

如果文稿的输入已经告一段落（对长文稿来说），或全部输入完毕，那么我们必须将这些字符存盘，以便将来使

用。WORDSTAR提供了如下几个存盘命令：

^SD——将当前文件（正在处理的那个文件称为当前文件，下同）存盘后退出编辑状态，回到主菜单。（^是`ctrl`的简写，以下同）

^SS——将当前文件存盘后回到的处理状态。如果先前是在输入字符，那么敲入^SS 就开始将文件存盘，然后再回到先前的输入状态，这时可以继续输入后继字符。对于输入一篇长的稿件来说，最好用此命令，输入一部分就存盘一部分，以免发生故障（如：突然断电）时前功尽弃。

^SX——将当前文件存盘后退出 WORDSTAR 字处理，回到BASIC的提示符“]”状态。如果我们本次操作完结而需要退出汉字处理系统，就可以用此命令。本命令同主菜单中的×命令作用相同。

^SQ——这表示撤消对文稿的修改，直接回到主菜单，如果先前对文稿进行了修改，现在确定不修改，希望仍然保留原来的内容，那么可以用该命令。

在上面的输入中，培训通知内容已全部输入，因而可敲^SD命令将“通知”内容存盘。

一篇文稿不论长短均是用这种方法来建立。但实际使用中要注意以下几点：

①如果输入字符过程中发现有错误，如：字符敲错了，那么可以用`<`键回退到出错处，再敲入正确的字符。也可以暂时不管，待全部文稿输完后再用专门的修改命令更正之。

②一个段落输入完毕，必须敲入一个回车键，这称为硬回车，表示后继字符显示在一个新行上。如果一段很长而屏幕上一行显示不完，我们可以只顾敲入字符，而由系统自动换行，这种换行称为软回车。硬回车的屏幕标记是“<”，

软回车标志是“+”。

③如果字符间需要保留空格(白)，可以敲空格键。如果一篇文稿每段的开头需要右退两个汉字，那么我们可以敲四次空格键。

④如果需要输入表格，那么可以使用图形符号。见第二章拼音输入法中介绍的特殊字符输入法。

3.2.3 存盘命令小结

命令	功能
^SD	存盘后回到主菜单。
^SQ	不存盘直接回到主菜单。
^SS	存盘后回到先前的处理状态。
^SX	存盘后退出字处理回到“]”状态。

3.3 光标的移动

我们要熟练地掌握WORDSTAR提供的各种编辑方法，首先得知道光标的移动命令。

1. 光标在行间的移动

WORDSTAR 提供了一些命令(又称光标控制键)，它们可以使得光标在屏幕的字符行间上下移动。

^K——光标上移一行，列位置不变。

^J——光标下移一行，列位置不变。

^M——光标移到下行的开头。↵有相同功能。

^C——光标下移7行(汉字)。

^R——光标上移7行(汉字)。

^X——光标移到文件的末尾。

^E——光标移到文件的开头。

2. 光标在列间的移动

^H——光标左移一个字符位置。◀键与此功能相同。

^U——光标右移一个字符位置，▶键与此功能相同。

注意：这里的光标控制键必须在编辑状态下才是有效的。

3.4 文稿的修改

一篇文稿在输入过程中不免出错，特别是长篇稿件的输入，出错更是难免。因而，WORDSTAR 提供了很强的修改功能。文稿的修改包括三个方面：一是添加字符；二是删去字符；三是更改字符。下面分别介绍之。

3.4.1 插入字符

所谓插入字符就是在文稿的字符间插入一个或多个新字符。操作方法是：

①在主菜单下敲入字母D进入编辑状态，然后回答文件名。

②将光标移到待插字符位置。

③敲入^I或|Tab|，使系统处于允许插入状态（屏幕显示标志为：COMMAND；插入（INSERTON），同时光标右端及以后各行为空，等待用户输入新字符。

④敲入新字符后，再敲|Esc|键退出插入状态，屏幕显示全部（包括刚插入的）字符。

⑤新字符插入完毕后用^SS或^SD命令存盘。

我们用例子来说明这种操作方法。

④修改完毕用 \wedge SD 存盘，然后回到主菜单（当敲入 \wedge SD后，显示COMMAND；正在存盘，如果发生错误，系统给出提示信息）。

顺便提及，如果要在文稿中插入一个空白行，那么可以用两种方法来实现：

一是按照前面介绍的插入字符的办法，将光标移到待插空白行的位置，将输入字符换成输入一系列的空白就行了。二是将光标移到待插入空白行位置的下一行，敲 \wedge I 进入插入方式，再敲 \swarrow 键，再敲Esc键，一个空白行就生成了。我们要强调以下几点：

①待插入（或称为添加）的字符只有在系统处于允许插入状态时敲入才会有效。（允许插入的标志是敲入 \wedge I 命令或Tab键后有“COMMAND：插入（INSERT ON）”提示出现在屏幕的上端）

②本软件以汉字为单位来移动光标，不论如何移动，光标始终能够指向汉字的前半部分，因而插入新字符时不存在会将一个汉字辟为两半的情况。

③输入的字符总是从光标处开始插入。

④插入字符完毕必须用Esc键退出（解除插入状态）。

3.4.2 删去字符

在编辑文稿中如果发现有多余的字符，可以用WORDSTAR提供的删除命令进行处理。删除字符的操作可以在两种状态下进行：

1. \wedge D删除命令的用法

在 \wedge D命令的控制状态下，可以删除屏幕上少量的字符，控制键有：

^D——使系统处于删除状态。

↵——删去光标所在行右端的全部字符。

▷——删去一个字符。

◁恢复被↵和▷去掉的字符。

这些命令的操作步骤是：

①在主菜单下敲字母D进入编辑状态。

②调入文件，将光标移到待删字符处。

③敲入^D使系统处于删除状态。（在屏幕上方出现的标志是“COMMAND：删除（DELETE）”）

④用↵或▷键删除字符。

⑤用Esc键退出（解除删去状态）。

使用中需要注意：

①事先必须将光标移到待删字符上，再敲^D进入删除状态。如果进入^D后再来移动光标就困难了。

②当用↵或▷删去字符而还没有退出^D状态（即没有敲Esc键）时，用◁键可以依次恢复消失的字符（这时被↵或▷去掉的字符又重现在屏幕上）。

2. ^Z清库命令的用法

要清除屏幕上少量的字符（一行或一个），用^D下的↵或▷是方便的。如果要清除若干行以至于全部文稿字符，用^Z清库命令下的控制键更为方便。

^Z——使系统进入清内存字符状态。

^ZC——清除内存中的全部字符。

^ZT——清除光标以上的字符（包括未在屏幕上显示出来的部分）。

^ZB——清除光标以下的字符（包括未在屏幕上显示出来的部分）。

^ZL——清除光标所在行的全部字符。

操作方法是：

①首先使文稿处于编辑状态。

②将光标移到待删位置处。

③用 ^Z命令进入清库状态。（屏幕上端提示“COMM
AND：清库：^Z”）

④用C、T、B或L清除字符。

⑤用 ^SS或 ^SD存盘。

这些命令的使用均较简单。请注意以下几点：

①清库操作必须在 ^Z命令控制下进行。

②在用C、T、B操作时，系统要在“COMMAND：”
位置上提醒用户并要求作出选择。如：

COMMAND：清除全部内容？

这是使用C时的提示，要求用户回答Y或N（下同）。

COMMAND：清除光标以上的内容？

这是使用T时的提示信息。B的提示是：

COMMAND：清除光标以下的内容？

③C、T、B操作成功后自动退出清库状态，因而不必再
敲Esc键。

④如果操作中途需要回到编辑状态可敲Esc键。

⑤删改完成后可用 ^SS或 ^SD存盘。

3.4.3 更改字符

所谓更改字符就是将当前的字符换成另一个字符。如：
“三”换成“四”，“X”换成“Y”。事实上，利用前面
介绍的删除再结合插入的方法也可以对文稿字符进行更改，
但WORDSTAR提供了更为简便的办法。

1. 直接更改字符

所谓直接更改就是在编辑状态下用某个(些)字符来代替另一个(些)字符。操作方法是:

①首先进入文稿的编辑状态。

②将光标移到待改字符处(可用<、>、^K和^J控制命令来移动光标)。

③敲入新字符。

④改写完毕后存盘(实际使用中可以多处修改完毕后一起存盘)。

2. ^G查找替换命令的用法

在文稿修改中往往有这种情况:把文稿中的某个字符(如:我们)全部换成另一个字符(如:大家),或者不同的地方换成不同的字符。在WORDSTAR中可用^G查找替换命令来实现。下面先介绍几个控制键命令:

^G——在编辑方式下使系统处于查找替换状态。

^GR——从文稿开头开始查找指定字符。

^GT——从光标的当前位置开始查找指定字符。

^GY——将查找到的字符自动换成指定字符。

^GU——仅自动替换光标所处的字符。

操作步骤是:

①首先进入文稿编辑状态。

②敲入^G使系统处于自动查找替换状态。

③用R、T、Y或U进行查找替换。

④替换操作完成后存盘。

我们再来看看各个命令的具体用法。

①^GR命令的用法

在文稿编辑状态下敲入^G后再敲R, 屏幕提示:

COMMAND: 找?学习↵

意思是询问用户需要查找文稿中的什么字符，如回答“学习”（用下划线标明回答信息）。屏幕又显示：

COMMAND: ①连续②顺序③反查?__

这里给出了可选择的三种查找方式，需用哪种方式就敲入相应的数字代码。

“连续”的意思是从文稿开头依次找出所有的“学习”字样。每找到一个“学习”，光标就停留在此，等待用户敲入替换单词，然后继续，直到全文找完或敲Esc退出为止。继续查找的提示标志是：

COMMAND: <SPACE>继续!

意思是敲空格键就继续查找文稿中指定的字符。

“顺序”的意思是从文稿开头找到第一个指定的字符。

“反查”是从文稿末尾开始依次向开头查找。

在^GR 状态下，不管是连续或顺序还是反查，系统都只能进行查找而不能自动进行替换。

注意：^GR 的敲键动作是先敲入^G 放开后再敲R。
^GT、^GY、^GU亦如此。

②^GT命令的用法

^GT同^GR用法相似，区别仅在于：

^GT查找的起点位置是光标先前的位置。如果敲^GT之前光标已指向文稿的某个字符，那么敲^GT后就从该字符（具体起点就是光标的位置）开始查找（查找方式也有连续、顺序、反查）。而^GR总是以文稿开头作为查找的起点（反查除外）。

③^GY命令的用法

^GY命令能够从文稿的开头依次将旧字符全部自动换

成新字符，不论先前光标的位置在文稿的什么地方均如此。
使用该命令时屏幕的提示标志是：

COMMAND: ^GY (敲入^GY命令)

COMMAND: 找?学习↵ (旧字符: 学习)

COMMAND: 换为?实用↵ (换成新字符: 实用)

COMMAND: (自动替换后回到编辑状态)

④^GU命令的用法

^GU同^GY用法相似。区别仅在于: ^GU以光标位置作为起点依次向后进行自动查找和替换。而^GY始终以文稿的开始作为查找起点。

3.4.4 修改命令小结

命令	功能
^I或Tab	进入插入状态。
^D	进入删除状态。
^D↵	删去光标右端字符。
^D▷	删去一个字符。
^D◁	重现由↵或▷删去的字符。
^Z	进入清库命令状态。
^ZC	清除内存中全部文稿字符。
^ZT	清除光标以上的全部字符。
^ZB	清除光标以下的全部字符。
^ZL	清除光标行的全部字符。
^G	进入查找替换状态。
^GR	从文稿头开始查找字符。
^GT	从光标位置开始查找字符。
^GY	从文稿头开始查找并替换字符。

- ^GU 从光标位置开始查找并替换字符。
Esc 退出^I、^D、^Z、^G以及^B、^P状态。

3.5 字块及其操作

在文稿的编辑过程中，有时需要将一句话、一段文章甚至几页从文稿的一个地方插入到另一个地方，或者需要将其删除掉。这可用字块操作来实现。

3.5.1 字块的设定方法

1. 什么叫字块

所谓字块就是用“【”和“】”括起来的一段字符。“【”称为块首标志，简称首标；“】”称为块尾标志，简称尾标。

字块中包含的字符可多可少，少到一个字符，一句话，多到几行，甚至整个文稿文件。

2. 字块首尾标的设置

设置字块的首标和尾标可用如下命令：

^BB——设置字块首标【。

^BK——设置字块尾标】。

具体操作方法是：

①将光标定位在待设字块的开始位置。

②敲入^BB，屏幕上立刻显示出字块首标【。

③再将光标移到待设字块的末尾。

④敲入^BK，屏幕上立刻出现字块尾标】。

⑤敲入^SS或^SD存盘（如果只是在屏幕上看看，可以不必存盘）。

实际使用中应注意：

①设定字块标志时，是先设置首标还是尾标无关紧要，但首尾标志的顺序不能颠倒。如：【…】是正确的，而】…【是错误的。

②同一个文件中只能设置一个字块。

3.5.2 字块的操作

设定字块以后，我们可以对字块进行搬迁、复制或者删除。

1. 字块的搬迁

在文稿修改中有时需要把某一段话插入到另一个位置，这可以利用字块的搬迁来达到目的。

^BV——将字块搬迁到当前光标的位置。

字块搬迁的操作方法是：

- ①进入文稿的编辑状态。
- ②用^BB和^BK来确定字块。
- ③将光标移到待插入字块的位置。
- ④敲入^BV进行搬迁。

例2. 假设有如下的诗句：

处处闻啼鸟，春眠不觉晓，
夜来风雨声，花落知多少。

一看便知第一二句的位置颠倒了，我们可以用^BV命令进行搬移。具体操作是：

①将“处处闻啼鸟，”设置成一个字块，方法是将光标移到第一个“处”字上，敲入^B。屏幕提示：COMMAND；字块操作^B，再敲入字母B，文稿显示为：【处处闻啼鸟，

…

②再将光标移到第一个逗号，之后，敲入^B后再敲K，整个字块设置完成，显示为：

【处处闻啼鸟，】…

③将光标移到“不觉晓，”之后，敲入^B后再敲字母V，整个字块的搬移就算完成。屏幕显示为：

春眠不觉晓，处处闻啼鸟，

夜来风雨声，花落知多少。

^BV命令对于长篇稿件中的字句移动来说最为方便。

2. 字块的复制

文稿中如果经常重复出现某一段字符，那么输入时可以只敲入一次，其余重复出现的字符可以用字块的复制来得到。

^BO——将字块复制到当前的光标位置。

^BO命令的操作方法是：

①将待重复的字符设置成一个字块。

②将光标移到需重复出现字块内容的位置。

③用^BO命令进行复制。

例3. 下面是一则笑话：

岂有此理

从前，有一个人喜欢学别人讲话。听到别人讲了一句“岂有此理”，他就念念不忘。

有一天过河，他还在念“岂有此理”。由于人多，等到上船后，早把“岂有此理”给忘了。他非常着急，便挤到船仓到处寻找。船工问他找什么，他说丢了一句话。船工说：话都会丢失吗？真是岂有此理。这人一听马上转忧为喜，带着抱怨的口气说：你早知道“岂有此理”，为什么不还给我“岂有此理”。

这则笑话中一共出现了7次“岂有此理”。在输入这段文稿时，我们可以只敲入一次，其它位置上的“岂有此理”都可以用 \wedge BO复制而得到。请读者自己完成。（参见3.6节）

注意字块的搬迁与复制之间的区别：

\wedge BV是把字块中的字符从一个地方搬到另一个地方，操作成功后原字块位置上的字符就不存在了，因而我们用了“搬迁”这种说法。而 \wedge BO则是将字块中的字符复制到另一个地方，操作成功后，原字块位置上的字符仍然存在。

3. 字块的删除

要删除文稿中的字符，除了可以用 \wedge D和 \wedge Z命令以外，也可以用字块删除命令。

\wedge BD——将字块中的字符以及标志一起删去。

如果要删除文稿中的某些段落，可先设置成一个字块，再用 \wedge BD命令删除之。

4. 删除字块标志

\wedge BH——删去字块标志。

注意： \wedge BH只是删去了字块标志，字块中的字符并没有被删除。

3.5.3 字块命令小结

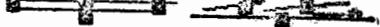
命令	功能
\wedge BB	设定字块首标【.】
\wedge BK	设定字块尾标】。】
\wedge BV	字块搬迁。
\wedge BO	字块复制。
\wedge BD	删除字块
\wedge BH	删除字块标志。

3.6 文稿的排版和打印

经过修改编辑的文稿，总希望用某种版面格式打印出来，以便阅读或保存。CEC-WORDSTAR提供了简易的排版和打印功能。

3.6.1 打印字体的种类

中华学习机CEC-I提供了A~O共15种不同型号的字体，示例如下：

- A 中华学习机
- B 中华学习机
- C 中华学习机
- D 中华学习机
- E 中华学习机
- F 中华学习机
- G 中华学习机
- H 中华学习机
- I 中华学习机
- J 中华学习机
- K 中华
- L 
- M 
- N 
- O 

在这15种字体中，最常用到的是A、B、C、D、E、F、G7种。在排版文稿时，可以根据不同的版面要求来选用不同型号的字体。

3.6.2 打印字体的设置方法

按照文章编辑排版的要求，对不同重要程度的文字符号，可以选用不同型号的打印字体。字体的版面设置方式有两种：一是整个文稿选用一种字体；二是在编辑状态下用建立命令来设置多种字体。

1. 整个文稿选用一种打印字体

如果整个文稿只选用一种打印字体，那么只需在主菜单状态下敲入字母P，系统就提示用户，要求回答打印字体（A~O）。如：敲入A，就表示文稿按A型字打印。这种设置方式的最大缺点是文稿版面太单调。

2. 同一文稿中选用多种字体

对一篇文章的排版来说，通常的字体要求是：文章题目字体最大；标题次之；正文再次之；脚注内容的字体最小。如何按段落的版面设计要求来选择不同的打印字体呢？可以用建立命令来实现。操作步骤是：

①在主菜单下敲入字母D，使文稿处于编辑状态。

②将光标移到文稿某行的开始位置，以便对该行字符指定打印字符。

③进入版面设计处理。敲入^P，屏幕提示：

COMMAND: ^P (G: 建立 Y: 取消)?_

G: 建立——表示需要版面设计。要求用户提供的信息有：打印字体（A~O）；纸的宽度（纸宽），即一行上可以打印多少个字符；每行左端留多少个空格（空移）；上下

行之间的距离（行距），每两个字符之间的距离（列距）。

Y：取消——去掉光标所在行原有的版面设计格式。

一般来说：如果文稿版面事先没有设计好，那么最好用G来建立。如果希望去掉原来的版面格式，那么可以用Y命令。

④选择字型。根据需要敲入字母G，屏幕提示：
COMMAND：打印字型（A—O）？

这个提示要求用户选择打印字体。

⑤选择纸宽和左边空格数。敲入字型代码，如：A，屏幕提示：

COMMAND：纸宽？__空移？__

纸宽表示每行打印多少个字符；空移表示左端留多少个空格。纸宽和空移都可以在0~255之间选值。

⑥选择行距和列距。敲入纸宽和空移数值后，屏幕最后提示：

COMMAND：行距？__列距？__

行距是指文稿每两行的距离；列距是指每两个字符之间的距离值。对于不同的行列要求，我们可以给出不同的行距和列距。对于文字稿件来说，行距可以是8，列距可以是1。对于表格，上下行之间可以挤紧一些，因此可以是小于5的正整数。一个较为理想的版面格式，可能要经过多次实验才能取得满意的打印效果。

以上操作完成后，该文稿行的版面格式就算设计好了，用同样的方法可以设计其它行的版面打印格式。在屏幕上版面格式的标记是：

〈字型T纸宽V空移Y行距X列距〉

假设：字型为A，纸宽为80，空移为5，行距为8，列距为1，则屏幕显示版面格式为：

3.6.3 文稿版面设计实例

我们仍以笑话“岂有此理”为例，说明如何进行文稿的版面设计。

1. 建立文稿文件

① 输入文稿字符。

由于笑话文稿中先后出现了7次“岂有此理”，因而可以只敲入一次，其余要用到的6个“岂有此理”用字块的复制来得到。下面请用汉语拼音或五笔字型编码方式输入（假设文稿名为：笑话）：

岂有此理

从前，有一个人喜欢学别人讲话，听到别人讲了一句“”，他就念念不忘。

有一天过河，他还在念“”，由于人多，等到上船后，早把“”给忘了，他非常着急，便挤到船仓到处寻找。船工问他找什么，他说丢了一句话，船工说：话都会丢失吗？真是“”。这人一听马上转忧为喜，带着抱怨的口气说：你早知道“”，为什么不还给我“”。

这段文稿中只敲了第一个“岂有此理”，其余6个暂时省去了。

② 在文稿上定义字块。

对于不熟悉汉字输入的人来说，用字块定义方法来得到重复的句子或段落是最为方便的。在这个笑话文稿中，把第一个“岂有此理”定义成字块。将光标移到“岂”字之前，敲入^BB，屏幕显示字块首标。再将光标移到“理”字之后，敲入^BK，屏幕立刻显示出字块尾标。完整的字块显示

为：

【岂有此理】

③进行字块复制。

将光标移到文稿正文第一行的引号“之后，敲入^BO，则“岂有此理”四个汉字便被复制到该位置。其余5个的复制与此类似。操作完成后文稿全文是：

【岂有此理】

从前，有一个人喜欢学别人讲话，听到别人讲了一句“岂有此理”，他就念念不忘。

有一天过河，他还在念“岂有此理”，由于人多，等到上船后，早把“岂有此理”给忘了，他非常着急，便挤到船仓到处寻找，船工问他找什么，他说丢了一句话，船工说：话都会丢失吗？真是岂有此理，这人一听马上转忧为喜，带着抱怨的口气说：你早知道“岂有此理”，为什么不还给我“岂有此理”。

在实用中注意：如果文稿较长，复制的段落较多，最好复制成功一段就用^SD命令存盘一次，以免发生故障（如：断电）时前功尽弃。待全部复制操作成功后再用^BH命令删去字块标志。

④对文稿进行必要的编辑和修改。

2.版面格式设计

①确定各段落的打印字体、纸宽、左空格、行距和列距。

在这个具体例子中，假设题目用F型体，纸宽为50，空移为10，行距为15，列距为1。正文通常比题目字体小，这里用A号字体，纸宽仍为50，空移为4，行距为8，列距为1。

②设计标题行

将光标移到标题行的开始处，敲入 $\wedge P$ ，屏幕显示一系列提问，我们一一回答如下：

COMMAND: $\wedge P$ (G: 建立 Y: 取消) ? G

COMMAND: 纸宽? 50 空移? 10

COMMAND: 行距? 15 列距? 1

操作完成后屏幕上显示的标题行是：

◀FT50V10Y15X1▶ 岂有此理 <

③设计正文。

将光标移到正文第一行的开头，敲入 $\wedge P$ ，重复上述步骤，且将版面格式的参数换成正文所需的数值就行了。操作完成后屏幕显示为：

◀FT50V10Y15X1▶ 岂有此理 <

◀AT50V4Y8X1▶ 从前，有一个人 +

..... <

④敲入 $\wedge SD$ ，将版面格式信息存盘，以后就可以直接用来打印了。

注意：①文稿的同一行上只能用同一种字体，但不同的行可以用不同的字体。

②版面格式的效果只有在宽行纸上打印出来以后才能明显看出。

3.6.4 文稿的打印

在主菜单中有一个P命令，其作用是打印文稿文件。操作步骤是：

①回答文件名字。在主菜单状态下敲入字母P，回答系统提问。屏幕提示：

P

文件名笑话↵

这里要求回答待打印文稿的名字，敲入：笑话↵

②是否建立打印格式？敲入文稿名字以后，屏幕提示：
需建立打印格式吗？N

由于打印格式事先已经建好，因而回答敲入：N。如果回答Y，则系统要求用户提供：字型、纸宽、空移、行距和列距等参数值。这就是整个文稿用一种打印字体的设计情况。

③回答打印份数。屏幕提示：

此文件拷贝数？1↵

如果需要打印多份，当打印完第1份文稿后，屏幕提示：

终止打印按Y，继续打印按N，暂停按P

根据实际情况，可以敲入相应的操作代码。

④是否分页，屏幕提示：

采用自动分页？Y

⑤每页行数是多少？屏幕提示：

每页/行数？10↵

⑥是否需要打印页码？屏幕提示：

消除分页符？Y

⑦换页时是否需要暂停？屏幕提示：

换页是否停顿一下？Y

⑧正式打印，屏幕提示：

打印机准备好后，按回车？↵

这时将文稿按版面设计格式打印出来，如下：

岂有此理

从前，有一个人喜欢学别人讲话，听到别人讲了一句“岂有此理”，他就念念不忘。

有一天过河，他还在念“岂有此理”，由于人多，等到上船后，早把“岂有此理”给忘了。他非常着急，便挤到船仓到处寻找，船工问他找什么，他说丢了一句话。船工说：话都会丢失吗？真是岂有此理。这人一听马上转忧为喜，带着抱怨的口气说：你早知道“岂有此理”。为什么不还给我“岂有此理”。

注意：

①如果用户认为某些提问并不重要，也可以敲↵作为回答。

②敲入^P可以暂停打印。

③敲Esc键可以中断正在进行的打印，无条件返回到系统主菜单。

3.7 WORDSTAR的其它功能

在主菜单中，我们已经介绍了D和P命令的用法。下面再介绍C、E、O、Y和X命令。

1. 列出磁盘文件目录

在主菜单状态下敲入字母C，便可列出磁盘上的文件目录。如：敲入C，列出目录为：

C

列目录

WORDSTAR V-254

B 019 说明书

A 004 HAN

B 004 字型

B 003 笑话

END OF FLLES:

敲入↵，系统回到主菜单。

2. 更换文件名字

在主菜单下敲入字母 E 便可对文件更名。如将文件名“HAN”换成“ZHONG”，敲入字母E，屏幕提示：

E

文件名字? HAN↵

名字换成? ZHONG↵

操作成功后系统回到主菜单。这时，用C列出目录可见，HAN已被ZHONG代替了。

3. 拷贝文件

在主菜单下敲入字母 O 可以把文件复制到另一个软盘上。如：敲入字母O，屏幕提示：

O

文件名字? 笑话↵

文件名拷贝到? XH↵

操作成功后回到主菜单。

4. 删除文件

在主菜单下敲入字母Y，屏幕提示：Y

Y

文件名字? ZHONG↵

操作成功后系统回到主菜单。

5. 退出字处理

在主菜单下敲入字母X，便退出WORDSTAR而回到BASIC状态。如：敲入X，屏幕提示：

X

谢谢使用CEC-1 WORDSTAR

习 题 三

- 1.请自己输入一段短文，练习字符的插入、删除、修改、字块定义和字块复制等操作。
- 2.对上述短文进行版面设计并打印输出以考察设计效果。
- 3.对表格的版面设计来说，行、列距以多少值较好？实验几次后进行比较。

第四章 汉字信息处理系统简介

4.1 汉字信息处理系统

能够实现汉字或汉语输出、输入、存储、处理与传输的系统都称为汉字信息处理系统。目前所有的汉字信息处理系统几乎都是以计算机为中心，再配备各种汉字或语音的输入、变换、输出和传输的设备而构成的。最简易通用的汉字信息处理系统是由通用的微型计算机系统，它包括主机，软件、硬盘驱动器，键盘，CRT显字器，打印机等，再配置汉字的输入、处理、输出软件而构成的。

计算机内部信息的存储、处理都是以二进制代码为基础的。英文字母采用二进制编码输入，在绝大多数的微型计算机中采用美国标准信息交换码即ASCII码。从信息处理的角度来讲，中文和西文并没有本质的不同，但是实际上由于汉字的数目特多，字形构造复杂，输入、输出、存储、处理都有自身的特殊性。因此汉字信息处理系统的关键问题就在于：

1. 输入问题

如何把汉字或语音转换成二进制数据输入到计算机。

2. 输出问题

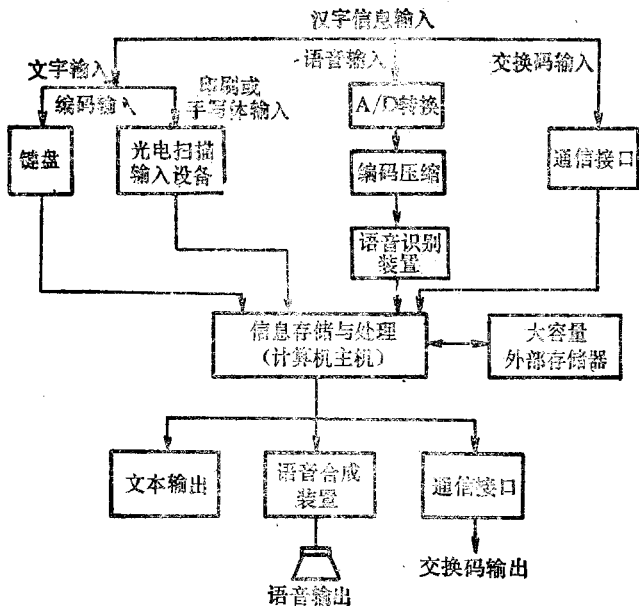
如何把二进制形式的汉字编码转换成汉字或汉语。

3. 代码系统问题

考虑到目前计算机，特别是微型计算机的发展水平，以

及中西文兼容的要求，汉字在机器内部存储、运算与传输采用怎样的代码形式即代码系统问题。

由于文字或语音的输入方式较多，一个比较完整的汉字信息处理系统如图4.1所示。



在一个比较完善的汉字信息处理系统中，能够把任何一种汉字输入信息，转换成所需的任何一种形式的汉字输出信息。例如把语音转换成文字；文字转换成语音等。但目前这种技术还不能完全实用。

4.2 汉字信息处理系统中的代码标准

4.2.1 西文输入编码

在绝大多数的微型计算机中以美国信息交换标准代码即ASCII码作为输入码，把26个英文字母，标点符号，10个数字以及一些常用的符号通过键盘输入到计算机内部。但是在IBM大中型计算机中却多以扩展的BCD码即EBCDIC码作为输入编码。在计算机内部，对不同的数据类型，其编码形式是不同的。例如整数常用16位补码表示；实数采用由阶码和位数共同构成的浮点数形式表示；字符串采用与输入码相同的形式表示。而且数据的存储和处理还可以采用不同的形式，例如规格化浮点的第一位尾数，因为在用原码表示的尾数中总是为1，因此在存储时不予保留，而只在运算时恢复它。在ASCII中，最高位留出作为奇偶校验位，这样剩下的7位可以区别128个字符。

4.2.2 汉字信息交换码

汉字信息交换代码是用于汉字信息处理系统之间进行信息交换的代码。我国目前计算机中所使用的交换码其标准主要有二个：①关于字母、数字和符号的GB1988《信息处理交换用的七位编码字符集》，它与国际通用的基本代码集相同，只是把货币符号\$变成人民币的标识符¥。②GB2312《信息交换用汉字编码字符集—基本集》GB2312规定了进行一般汉字信息交换用的6763个汉字和682个非汉字图形字符。每个汉字（包括图形字符）用二个字节表示，每个字节

为7位二进制编码，其取值从21（16进制）~FE（16进制）。即每个汉字都是由GB1988字符集中的二个图形字符编码（除SP，DEL外）组合而成的。在中华学习机上，已将GB2312中规定的6763个汉字固化在主机板上了。

4.3 汉字信息处理过程中的代码变换

由于汉字的数量大，如果只能按GB2312中规定的区位码值来输入，这就使人难于记忆，不利于汉字信息处理系统的推广，因此产生了各种各样的汉字编码输入技术。在这种编码输入的系统，代码的变换过程如图4.2所示。

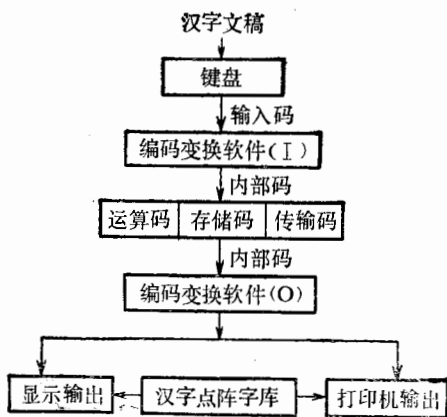


图4.2 编码输入过程中代码变换

1. 汉字输入码

汉字输入码也称外码，是为了将汉字通过键盘输入到计算机而采用的编码。对于它们的基本要求是编码规则简单，易学易记，操作方便，输入速度快，重码率低。输入码一般都比较长，要3—4个字节才能代表一个字，而且码长不等，有些字一个字节，有些字三、四个字节。因此不便于汉字的存储与处理。所以输入码还要转换成长度相等且较短的内部码。目前比较流行的汉字编码输入方法估计已有几十种，比较典型的如：区位的输入法；五笔字型编码输入法；宏观字形编码（钱码）输入法；拼音码输入法；首尾码输入法；通俗码输入法；电报码输入法等。这些输入方法中，象区位码输入法，电报码输入法，一个四位数字代表一个汉字，汉字和它们的编码是一一对应的，一个输入码只对应着一个汉字，这类编码称为无重码字编码。上述的其他类型编码，例如拼音码等，往往同一个编码对应着好几个汉字，这类编码称为有重码字的编码。在使用这类编码方法输入汉字时，计算机显示所有的重码字，并使你能从中选择出所需要的字。

2. 汉字内部码

在汉字信息处理系统内部为汉字的存储、运算与传输所编制的代码称为内部码。存储、运算与传输可以采用不同的内部码形式。在决定内部码时应考虑下列因素：

①码尽量短，能代表的汉字尽量多。

②与现行的汉字交换码（即国标2312汉字信息交换基本集等）有比较明确的对应关系，转换起来方便。

③尽量与高级语言中字符串类型相一致，使得高级语言中对字符串的处理都适用于对汉字的处理。这样，若字符串能作为文件名称，变量标识符等，那么汉字也能作为文件名

称，变量标识符。

编码变换软件（I）的作用就是把输入码转换成内部码；而编码变换软件（O）的作用则把内部码转换成对应的点阵信息使该字显示在屏幕上，或通过打印机打印出来。

3. 汉字字形点阵信息

汉字的显示和打印都是利用汉字字形点阵信息（它在点阵字型库中，例如：CEC-I的固化字库），借助于计算机的图形显示和打印功能而实现的。

汉字点阵点数的多少直接影响汉字的造型和质量，点数越多，汉字的质量可做到越高。点数的多少以横向点数乘纵向点数表示。目前在微型计算机中普遍采用 15×16 ， 24×24 ， 32×32 的字型点阵；对于象轻印刷系统，激光照排系统等使用更精密的点阵。一个 24×24 点的汉字字模如图4.3所示。它需要72个字节的存储容量。

目前 15×16 ， 24×24 ， 32×32 点阵的字型库都已经标准化，有相应的国家字形点阵标准，分别称为GB5199，GB5007，GB6345等。目前的汉字点阵字型库中包括GB2312中规定的6千多个汉字以及各种图形符号。对 16×16 点阵而言，由于每个汉字点内存32个字节，GB2312总共8836个汉字（包括符号）共占内存约280k字节，在CCDOS中，在启动时自动装入内存。对于一些分辨率较高的字型点阵，例如： 64×64 点阵字模，每字占512字节的存储容量，8836个单元共需要4.5M字节左右的存储容量，因此它们都只能驻留在硬盘上。目前 15×16 ， 24×24 ， 32×32 点阵信息，往往写入高密度的PROM芯片中，制作成汉卡的形式，再插入计算机或打印机中作为字库使用。

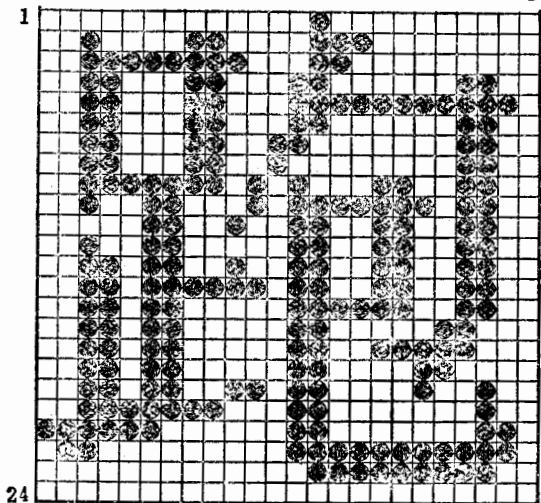


图4.3 24×24点阵的汉字字模“跑”

4.4 关于计算机内码的讨论

汉字机内码主要指在计算机内部汉字进行存储、运算（合并、增删等操作）与传输所使用的码，但不包括字型点阵信息。微型计算机在英文文本工作方式时，输入码，交换码，存储运算和传输码都是ASCII代码。但对于汉字来讲，理想的情况希望这些码都是一致的。但是对于汉字信息处理系统，问题就比较复杂，例如GB2312国标汉字信息交换码

是用二字节七位ASCII字符图形码表示的，在中英文兼容的系统中，一字节的西文字符随时都可以出现在汉字之间，这就要求系统能够正确区分出是英文字符还是汉字。为此必须对汉字交换码作出某种标识。经过标识以后的交换码就成为机内码了。目前国内采用的机内码大约30多种，主要有以下几种方法：

1. 用国标码最高位置1的方式

如图4.4所示。



图4.4 高位置1方式的机内码

它把双字节汉字交换码的二个字节的最高位都置成“1”，以示和ASCII码相区别。它的最大优点是机内码表示简单，和交换码之间有明显的对应关系。不足之处在于它只能表示94个区，94个位共计8836个汉字。若要在GB2312基本集以外增加辅助集，只有另辟途径，例如用第一字节高位是1，接着一个字节高位是零表示是另外一个汉字集等。这种机内码表示法，在以单字符为基本编辑单元的汉字文本编辑软件中，有时会出现错位现象。例如：汉字“交换”二字，区位码是2927 2727，现在希望把“换”字改成“还”字，“还”字的区位码是2725。如果采用以单字符操作为基础的查找替换命令，查找完成时光标会移到“交”字的第二个字节上，替换后的结果是2927 2527，它是“交够”二字的区位码，这样由于定位的错误，达不到正确修改的目的。这种机内码表示方式还要求原系统能够接收高位为1的ASCII码，否则要修改系统软件。

2. 用专门的图形字符标识汉字

在汉字交换码之前加一个系统基本上不用的字符以标识汉字。这种方式 and 英文的兼容性好，和交换码的对应关系也比较清楚，但是开销较大，每一个汉字都要在前面增加一个字节的汉字标识符，而且对不同的系统很难选择到同一个字符来作任何标识字符。

3. 用首尾二个标识符括住一串汉字

如图4.5所示。



图4.5 用二个标识符括住一串汉字作机内码

这种方法比上一种方法开销小些，不必每个汉字都加上标识符，但对汉字串作截断处理时往往容易发生混乱。其它和上法差不多。

4. 用字母和数字的组合来表示汉字机内码

此方法不是用交换码来表示汉字，而是用字母数字的组合来表示汉字。例如：用26个小写字母，26个大写字母，10个数字来对汉字进行编码，当然前面还要加汉字标识符，如图4.6所示。



图4.6 字母数字组合机内码

这种方式中西文兼容性好，不需要修改系统软件。但是和交换码的关系不直接。

目前在微型计算机的CCDOS汉字操作系统中，采用上述第一种方式，即交换码高位置“1”的方式表示机内码的较多，输入码经编码软件变换以后转换成机内码，供存储、运算、传输使用。在显示和打印时，再把机内码转换成相应

的点阵信息，在屏幕上显示出来或通过点阵打印机打印出来。

习 题 四

1. 什么叫汉字信息处理系统？
2. 简述汉字信息处理系统处理汉字的工作过程。
3. 汉字信息交换码同西文码有什么不同？
4. 什么是汉字输入码？内部码？确定内部码应考虑到哪些因素？
5. 什么是机内码？有几种表示方式？各自的优缺点是什么？你认为还可有什么方式更好？

第五章 软件汉化的原理与方法

对只能用于西文信息处理的计算机软件进行汉化是一项复杂而费时的工作。但在我国，若不解决计算机的汉字信息处理问题，就谈不上计算机技术的大力推广、普及和应用。本章讨论软件汉化的一般原理和方法，供有兴趣的读者参考。

5.1 什么叫软件汉化

所谓软件的汉化就是指为计算机系统设计一个能够识别汉字的软件，使得该软件能够输入、显示和打印汉字。在设计识别汉字的软件时，可将汉字同西文一样作为字符来处理，并且设计的汉字软件可中西文兼容，即可以输入输出汉字，同时也可以输入输出西文字符。实现这样一个系统，就叫做软件汉化或简称汉化。如果是一个操作系统，经过汉化处理后可识别汉字和西文字符，则可称它为中西文操作系统。一般的中西文操作系统的构成如图5.1所示：

在中西文操作系统支持下的实用、应用程序以及一系列的工具体软件，如编辑程序、各种高级语言的编译程序、管理软件dBASE等也都应进行汉化处理，得到中西文文本编辑、可识别中西文的高级语言以及各种中西文管理软件。人们通过这些软件进一步开发中文应用软件。对软件的汉化可分为：系统软件的汉化和实用、应用软件的汉化。

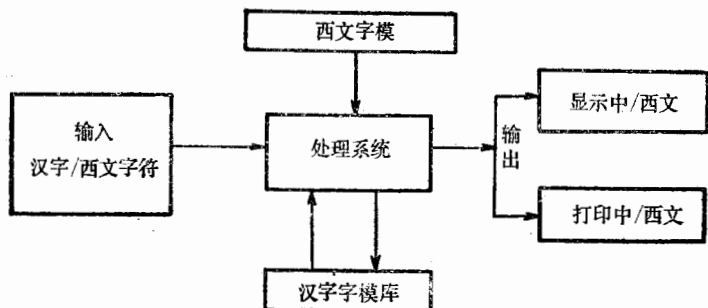


图5.1 汉字系统简示图

对系统软件汉化工作可分为：汉字编码、程序设计、系统调试和修改以及运行四个阶段。

5.2 系统软件的汉化过程

5.2.1 汉字的编码

第一阶段是进行汉字的编码工作。目前国内流行的中西文操作系统均基于《信息交换用汉字编码字符集—基本集》(GB2312—80)编码，这个标准中有94行94列，共有 94×94 个表项，我们把行称为区，即94个区，把列称为位，亦有94个位。其中1~15区为字母、数字图形符号以及其它一些记号。从16区开始到55区为一级汉字区域，第55区只有89个汉字，因此，一级汉字区域共有 $40 \times 94 - 5$ （即3755个）汉字。从56区开始到87区为二级汉字区域，共有32个区，拥有3008个汉字。

本标准拥有汉字6763个，一级汉字区域可用汉语拼音直接进行检索，提取所需要的汉字。二级汉字区则只能按部首与笔画顺序进行检索。在西文操作系统中用一个字节来表示

字符的ASCII码。实现汉字的存取用一个字节容易与西文字符混淆，因此，我们用区和位码来定位一个汉字，区码和位码合在一起的编码值将占用七位，为了以示区别，用两个字节分别表示区、位码，第一字节表示区码，第二字节表示位码，如图5.2所示。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x

图5.2 区、位码的表示

为了实现中西文兼容的操作系统，我们规定将区、位码的高位置“1”，后七位为区、位码值，这样的编码作为中西文系统的汉字机内码，如图5.3所示。

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	x	x	x	x	x	x	x	1	x	x	x	x	x	x	x

图5.3 汉字机内码的表示

确定编码以后，则是生成汉字的字模，将汉字信息表中的字母、数字、图形符号、汉字等全部信息按照一定的数据结构，建立汉字字模信息库，以供中西文系统调用，并作为显示、打印的信息。

我们知道，在图形方式下，西文的ASCII码字符是以8×8的点阵显示的（如图5.4所示），其中右边一列和下边的一行作为字符的行、列的间隔。如字母“A”和“a”的数据分别为：

030H, 078H, 0CCH, 0CCH, 0FCH, 0CCH, 0CCH, 000H和

000H, 000H, 078H, 00CH, 07CH, 0CCH, 076H, 000H

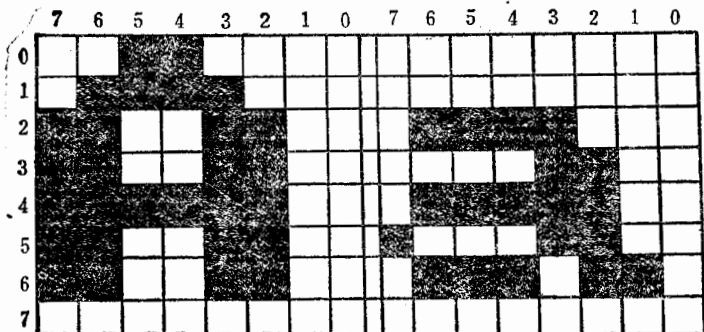


图5.4 ASCII码点阵图象

这种点阵数据的字符可在图形方式下显示。在 640×200 的图形方式下可显示 80×25 个，在 320×200 图形方式下可显示 40×25 个，同时，在显示、打印之前可进行放大、缩小及旋转等变换。

由于汉字的结构极其复杂，如果表示成ASCII码那样大小的点阵，则对于笔画多的汉字就无法表示出来；若用 16×16 位的点阵来表示，那么在显示时，一个汉字需占用16线16列。而在 640×200 的图形方式下，可显示 40×11 个汉字；在 320×200 的图形方式下，可显示 20×11 个汉字。

汉字字模的点阵数据也象ASCII码字模数据一样存入内存存储器中，由于汉字点阵数据太多，仅一个汉字字模的点阵就得占用32个字节，表中全部汉字点阵数据共占用216416字节（图形符号等的点阵数据未计入）。

按上述方法设计的汉字字模可供显示和打印输出用，也可编写一个程序在绘图机上绘制汉字，并都可使其放大，缩小以及旋转等。

目前流行的中西文操作系统多数以16点阵的字模库为

主，且存放于随机存取存储器中，而开发的如 24×24 、 32×32 、 48×48 、 64×64 、 128×128 等点阵的字母库，由于文件较大，一般都驻留于磁盘上。如CCDOS 2.13A中西文操作系统，拥有 24×24 的宋体、仿宋体、黑体、楷体四种字母库和图形符号库，每种字体的汉字库均为48936字节大小。又如一般的排版系统也拥有宋、仿宋、黑、楷体四种字库，其字模为 64×64 点阵，每个汉字占用512字节。因此，汉字库均为3462656字节大小，这些中、高品质的字模一般只供打印输出用，而在显示器上显示的仍为低品质（ 16×16 ）字模。

上述编码方案只解决了显示、打印和汉字机内编码的技术问题。下面将讨论如何输入汉字的问题，怎样才能找到我们所需要的汉字呢，这就要规定一种或多种汉字输入码。在我们查阅《新华字典》的过程中，采用了拼音查字法和部首查字法；在电讯部门规定了电报码的查字法；要从GB2312—80汉字信息表中查找一个汉字，只要知道区、位码值，就可以找到所需要的汉字，我们称它为区位码查字法，还可例举许多查找汉字的方法。

目前的汉字系统中，用拼音码、区位码、首位码，快速码输入汉字的较多。因此，系统设计上还应考虑汉字输入的编码，下面以常用的这四种汉字输入法为例进行编码。我们规定：

拼音码最多不超过三个小写字母键；

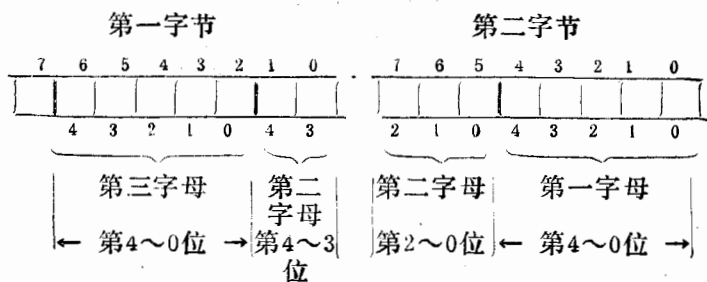
区位码为四位无符号十进制数字符号键；

首尾码为三键，第一、二键分别是汉字左上角与右下角的笔形码对应的小写英文字母，第三键是汉字的汉语拼音第一个字母键。

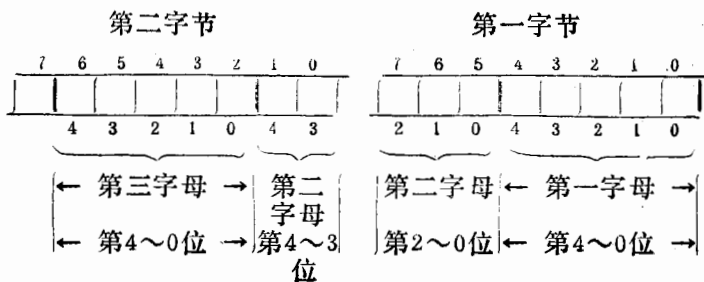
快速码不超过四键，前两键为首尾码的首字母和尾字母，后两个为拼音码的头两个字母。

这四种汉字输入码中，用区位码输入汉字时，通过区位码的直接计算可获得所需要的汉字。而其余三种输入方法则需要建立它的输入码对照表，因为规定用26个小写英文字母：a~z，则我们可给字母a~z按顺序分别编号为00001~11010，也就是说序号最大为26，用5位二进制数字即可表示一键，三种输入码中最长用四个字母键，所以最多用20位。我们可分别将它们表示为：

拼音码：

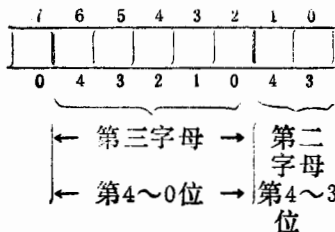


首尾码：

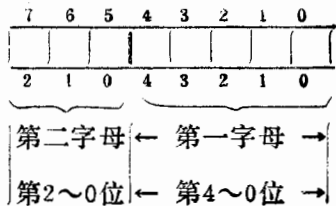


快速码：

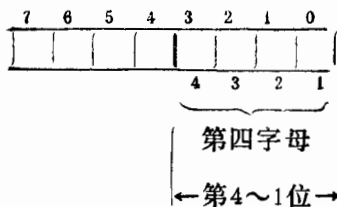
第二字节



第一字节



第三字节



如果将它们分别编成独立的输入码对照表，则汉字的输入码对照表占用47341字节。根据三种输入码之间相互关系，我们将表示一个汉字的三张表项合并成一张表，合并的方法是将其公有的部分合并，取其中一份可表示为如图5.5所示。

这样合并以后，一个汉字的输入码对照表项只占用四个字节，减少了存储器的开销，合并以后的输入码对照表的大小为27052字节。

5.2.2 编写汉字识别系统程序

第二阶段是编写识别汉字的系统程序，这是软件汉化的中心环节。在这里编写的程序要求即可识别汉字又可识别西文字符，这就给编程工作带来了很大的困难。这一阶段着重解决的是键盘的管理和显示器的管理以及打印机的管理。

第四字节				第三字节				第二字节				第一字节			
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
4	3	2	1	0	4	3	2	1	0	4	3	2	1	0	4
拼音码 第三字符				拼音码 第二字符				拼音码 第一字符				首尾码 第一字符			
快速码 第四字符				快速码 第三字符				快速码 第二字符				快速码 第一字符			

图5.5 汉字输入码对照表项结构

即实现汉字的输入，并根据输入码字母去查找与之匹配的汉字输入码对照表项（图5.5所示），再转换成汉字的机内码。最后，通过汉字机内码去提取汉字字模点阵数据供显示和打印输出。

5.2.3 系统的调试与修改

第三个阶段是系统的调试、修改。这里首先要建立中西文操作系统所需要的表格信息、汉字字模数据库等，并将编写的程序建立成文件。其次是调试识别程序和辅助程序，在程序调试的过程中，对原设计不合理以及未考虑到的地方重新设计、调整，同时对表格信息、汉字字模数据库中的错误进行修改、再调试等，实现中西文操作系统。

5.2.4 运行汉字识别系统

第四个阶段是运行汉化后的系统。这一阶段是一个长期的对系统检验考核的过程，也是用户对系统设计思想，程序设计、系统质量、系统的配置等的鉴定的阶段，设计者有义务解决运行过程中发现的问题并且提供改进型的中西文系统版本。

5.3 实用、应用程序的汉化问题

对系统软件（如操作系统等）的汉化大致要经历以上四个阶段的工作。而实用、应用程序的汉化是指对中西文操作系统下运行的实用、应用程序的汉化。这些软件可分为两大类：一类是在中西文操作系统的管理下，系统处于中文显示方式时运行这些程序后，系统显示方式被改变成了西文显示方式，退到操作系统后仍处于西文显示方式，有的在中文显

示方式下根本就不能执行。另一类是在中文显示方式下可正确地运行并能得到正确的结果，而保持操作系统原来的显示方式不变。因此，汉化时要分别对待。

汉化第一类软件，首先要解决的是将显示器置成图形方式，并且保持在图形方式下运行。其次是将屏幕提示信息串改成相应的汉字串。其长度应不超过西文信息串的长度。最后，若有输入功能，可考虑是否需要输入汉字字符，如果不需要输入汉字，则不必处理。如果需要输入汉字，则还要改造这部分程序。

汉化另一类软件，只是不必考虑屏幕的方式，其他的汉化工作与第一类软件大致相同。

5.4 软件的汉化方法

实现汉字中西文兼容的常用方法有软件、固件和硬件法三种。软件法和固件法多数应用于微机系统。而在多用户超级微机以及大、中、小型计算机系统，则多采用硬件法。

软件法是将操作系统加以扩充，增加汉字信息加工软件，如检索、同码处理、字形点阵旋转、字形点阵数调度等，以及增加汉字处理支持软件，如屏幕编辑、汉字字典生成、汉字库建立与维护等。实际上是对原系统中管理外部设备的那部分程序加以扩充改造，使它能支持汉字处理。这样，各种程序设计语言都能调用和处理汉字，原操作系统的功能全保留，使系统成为中西文兼容系统。例如 IBM-PC 的 CCDOS 操作系统，就是将原有的 PC DOS 操作系统的 BIOS 改造为 CCBIOS，使之支持汉字系统。

固件法是将操作系统中扩充的程序和字库固化在汉卡的

EPROM或ROM中，这就可能在基本不修改操作系统的情况下，使大多数原来只能处理西文的软件能直接用来处理汉字。

硬件法采用汉字终端，在终端设备上配有汉字库和中文的输入输出处理程序。由终端直接对操作员输入的汉字和西文进行识别，经编辑处理后，全部以ASCII码的形式输入主机。输出时，终端接收主机送来的汉字和西文混合信息，加以区分后转换成点阵信息进行显示或打印。这种终端汉字和西文都是以ASCII码形式出现，主机可以统一处理。主机将代码转换，字形处理等例行工作交给终端去做，不用修改原有操作系统，使原来的西文计算机不作任何改进就能实现中西文兼容。这种终端既不占用户的内存空间，又大大提高了主机处理中文信息的效率。终端和主机连接时通过标准的串行接口。

以上三种汉化技术在我国均已比较成熟，汉化效果好，得到广泛应用。

要使计算机具有处理汉字的功能，需要解决汉字输入、汉字库和汉字输出三个主要问题。汉字处理的大致过程是：用户通过键盘把汉字编码或词汇编码输入到计算机内，汉字处理软件通过“汉字字典”查找汉字或词汇的内部编码，然后对内部码进行处理加工，输出时，再根据汉字的内部码去汉字库找出相应的汉字字形信息，送往显示器显示或点阵打印机去打印。

5.5 汉字输入法概述

英文是拼音文字，每个单词的字母是按照自左至右的顺

序排列的，只要按照单词的字母顺序击键，就能得到它们的编码。英文的字符，包括大写字母，数字和常用符号加在一起总数不超过128个。因此采用键与字母一一对应的键盘，就能方便地将英文信息送入计算机。汉字是方块图形文字，字数多，字形复杂，加之简、繁、正、异各体，总共不下六万个。这给汉字信息的输入、输出和字形存储带来了巨大困难。

汉字输入根据键入技术来分类，大致分为整字、字素和编码三大类。

采用一字一键的输入方法，这就是大键盘整字输入方式。这种方法在通用汉字系统上几乎不用，只在专业系统（例如，电子照相排版）等某些特定的场合使用。

字素键入是直接利用ASCII码标准的小键盘输入汉字信息。字素键入是将字素（几十到一百多个）刻在键帽上，将汉字分解后按一般的笔顺键入。

编码键入也是直接利用ASCII码标准键盘输入汉字。编码键入是将汉字的语音、字形等特征转换成ASCII符号，从标准键盘输入汉字。这种方法用得最普遍。

汉字输入的核心就是编码方案。目前国内外提出的汉字编码方案不下500多种，这些方案各有千秋。由于码长、规则等因素决定了不同编码方案的优劣和速度。最常用的汉字输入方法只有十几种。比如国家标准代码、大众码、五笔字型码、前三末一码、国际区位码、汉字拼音码、双拼码、声韵部形码、笔形码、见字识码、首尾码、仓颉码、电报码、手写输入、长城笔形码以及智能化的联想式汉字输入等。这些编码中根据其特点，又可以大致划分为拼音码、形码和音形码三类。

①拼音码是指每个汉字都用拼音字母输入。汉语普通话发音有四百多种音节，可以由22个声母、37个韵母拼合而成。用拼音方法表示发音，不可避免地会出现大量同音字，这些同音字具有相同的编码。为了进一步识别这些同音字采用了“拼音——提示”编码方案，即可在计算机屏幕上提示同音字供操作者选择。

③形码是根据汉字的笔形、角形用英文字母或数字进行编码，或者从汉字拼形出发划分为字元（码元）进行编码。汉字总数虽多，但是组成全部汉字的部件和笔划都是有限的。汉字部件的大部分与字典的部首相同，因此，可以把汉字的笔划、部件等看作汉字的字母，按“先高后低，先左后右”的原则键入。例如长城笔形码是一种把汉字笔画看成汉字的字母，按照笔画给汉字编码的输入方法。它仅用1~8共八个码元代表汉字结构中的八种笔形，通过不太复杂的编码规则可以编出覆盖国际汉字集全部汉字的输入代码。按照“先上后下，先左后右”的原则取汉字笔形。各个部件中取笔形编码时，最多取前面的三码，整字取六位码，其余信息作为冗余码处理。因此，笔形码最大的码长为六位。对最常用的汉字定出简码，以及利用词汇码提高编码效率。在此基础上，长城笔形码最近又推出了新版本，新的笔形码具有音形结合，字词结合的特点。可以把中文字、词按声韵加笔形编码。形码、音码无需切换，字词键入不要间隔，并可以按笔形码的原则给任何外文字包括不认识的文字进行编码。

王安电脑公司较早采用的三角编码法，就是一种角形编码。它使用基本字形作为每个角的编码，也是按照“先上后下，先左后右”的原则取三个角的字形进行编码，以此区别各个汉字，而很少出现重码。

③音形码是根据汉字的声与形或声韵、部、形、义、频度等相结合进行的编码。例如见字识码就是拼音与字元相结合的一种等长编码。它是先将每个汉字分解为4个字元组成的字元串，然后每个字元可找到一个关系字，取关系字汉字拼音的第一个字母为标记，将字元串变成字母串，即可得编码。

除上述三类编码方法外，还有国际码、四位数字的标准电报码等，都是与字形和字音无关的编码方案。

不论采用何种编码输入方法，操作者都需要熟记编码的规则，这就大大影响了键入速度。因此各种键入法都应尽量利用人工智能技术来提高键入速度以减轻操作人员的负担。智能化联想式汉字信息处理系统，就是比较全面和完善地解决了汉字信息处理的各个难题。联想式汉字系统适用于所有的输入方法，它以计算机辅助输入技术，比一般人工编码高一层，应用了人工智能原理。它容纳了五万条以上的词汇，就使快速、易学的编码可以迅速建立。今后的汉字编码方法充分利用词汇输入的功能会超过现有的各种编码。

由于汉字的特殊性，实践证明，要想将来出现一种标准的汉字键入法是很难的。对于专职人员，速度是首位；而对于一般用户，易学才是头等之事，这两个指标是相互矛盾的。因此，一个通用的汉字系统，一般都允许用多种编码方案输入汉字。从键盘得到的代码称作外部码。编码方案不同，外部码的长短和格式等也不相同，彼此相差很大，必须通过键盘解释程序把它翻译成统一的机内代码，即内部码。在计算机内，汉字信息的处理是以内部码的形式进行的，因此必须先将键盘送来的一组外部码放在键码暂存区，然后通过汉字编码解释程序查找“输入字典”，找到相应的内部代码。汉字输入的一般方法如图5.6所示。

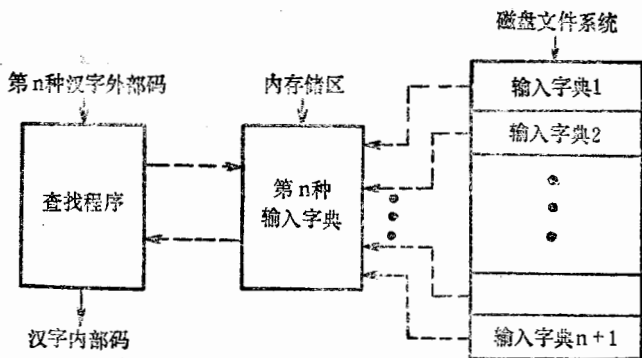


图5.6 汉字输入的一般方法

“输入字典”实际上是张键盘码和内部码的对照表，表的结构不同，查找方法也不同。常用的有三种方法：

①顺序法。对照表中的键盘码是任意排列的。查找时从第一个键码开始，对码值逐个进行比较，直到两者相等，就得到对应的内部码。这种方法速度慢。

②折半法。键盘码按大小排列，取中间值码和需要转换的键盘码值加以比较，判定应在哪一半范围内查找，再取新的中间码值，折半缩小查找范围，直到查出。

③散列计算法。把数值上离散很大的键盘码值，经简单的算法把它压缩到一个连续的地址空间，可直接算出有关的地址值，快速查找到对应的内部码。“输入字典”一般放在磁盘中，查找时再把它的一部分调入内存储区。对于一个汉字系统，有多少种汉字编码输入方法，就要建立多少种“输入字典”和检索程序。

内部码是设计中西文兼容的关键问题。不同汉字系统往

往采用不同的内码。目前最常见的内码有以下三种。

①双字节表示法。将双字节汉字交换码的两个字节的最高位置1，以区别ASCII码。英文字母是用ASCII码的形式存放，每个字母用高位为0的一个字节表示。其形式如：

	8	7	6	5	4	3	2	1
字符	0	×	×	×	×	×	×	×
汉字	1	×	×	×	×	×	×	×
	1	×	×	×	×	×	×	×

②图形字符标识法。它是将系统中不用的某个图形符号作为汉字标识符，有些将这个标识符用于汉字串的首尾。

③三字节表示法是指用三个字节表示一个汉字，常用的有ADD（A表示A—Z26个字段，D表示0—9十个数字）和AaA（A表示26个大写字段，a表示26个小写字段）等。

一般地说，双字节表示法信息冗余度小，且便于转换为交换码；三字节表示法则与西文兼容性较好。

5.6 汉字库

计算机中的汉字内码如同ASCII代码一样，仅是一种代码，并不能表示汉字的字形。要使计算机的终端显示汉字或打印机打印汉字，还必须将内部码转换成汉字的字形信息，又称字模。在计算机系统中，字模以图形方式预先存入计算机的汉字库中。表示汉字字形的常用方法为点阵法。点阵法以0、1二进制表示字形的一个白点或黑点，它不仅表示

汉字，也可以表示汉字以外的任何图形符号，包括外文字母和数字。由于点阵法的汉字库易于生成、修改和管理，结构简单等优点而被广泛采用。

点阵法字库中的汉字一般用 18×16 、 16×15 、 24×24 和 32×32 的点阵，而对于西文字符，常用 7×7 、 7×9 和 9×9 的点阵。点阵法的点阵数目越多，表现的笔划精细，汉字输出质量越高，但需要的存储量也越大。一个 18×16 的点阵汉字字模，要占用36个字节（ $18 \times 16 = 288$ 个点，每点占一位， $288 \div 8 = 36$ ）。 24×24 的点阵则要占用72个字节。如果用 18×16 的点阵直接表示国标一级汉字，则要占用140K字节（一级字库数 $3755 \times$ 一个字模占36个字节 ≈ 140 K字节）。

为了提高读取字形点阵的信息速度，常用汉字的字形存放在ROM或EPROM存储器中，这种ROM或EPROM被称作硬汉字库或汉卡。为了不占用系统的存储空间，一般都是通过并行接口片子从ROM或EPROM中读取汉字字形的。为了表示国标一级汉字字库，使用不同集成度的ROM片子，字形点阵数不同所用的ROM片子数是不同的，它们之间的关系如表5.1。

为满足在不增加设备的情况下，仍然使用汉字库的这部分用户的需要，将汉字字形点阵信息存放在软磁盘中，这种汉字库被称为软字库，当然这是以降低汉字输出速度为代价的。硬字库中的字形点阵一般按汉字内部大小顺序存放。软字库中的字形点阵可按不同的排序方法放在软盘的特定系统区，或者作为文件放在软盘上。不论是硬字库还是软字库，汉字的内码都应与汉字点阵构成一一对应的关系。

由表5.1可知，为存放国标一级汉字，使用128K位的ROM片子构造成 24×24 点阵字形的字库，需要使用18个片

表5.1 汉字ROM的构成

容量/片	构成	片数(一级汉字)
1M单位	18×16	1
	16×15	1
	24×24	2
	32×32	4
128K位	18×16	10
	24×24	18

子。但一般情况下不是把一个汉字的字形存放在一个ROM片子上，而是分开存放在九个片子上的。把这18个片子分成A和B两组。A组的片号为00~08，可以构成一个汉字，其中每个片子存放字形九分之一部分的八个字节；B组的片号为09~17，由它们构成另一个汉字。

由于汉字的内码是用两个字节表示，用它对国标约44个一级汉字进行实际寻址时，显然存在着冗余信息，可以采用硬件或软件方法对汉字内码进行压缩，组成同ROM地址相对的12位地址信息($a_1 \sim a_{12}$)。如图5.7所示。

压缩码的 a_{12} 用来对ROM的组别进行选择， a_{12} 是“0”时选择A组； a_{12} 是“1”时选择B组。压缩码 $a_1 \sim a_{11}$ 是送到ROM片子的引脚 $A_3 \sim A_{13}$ (地址线)，用来选取某组中的一个汉字。

点阵汉字库可以利用EPROM进行复制，也可以利用计算机系统的字库管理程序来建立，也可直接利用国家颁布的15×16，24×24点阵字模和数据集的国家标准，以避免大量重复的劳动。

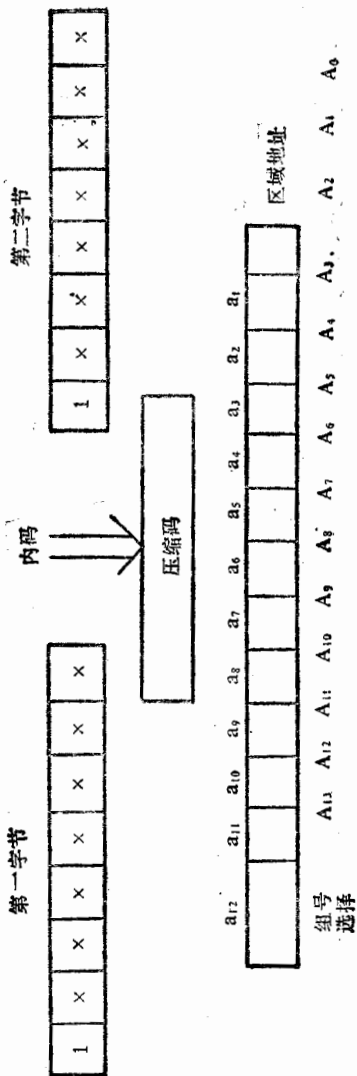


图5.7 内码转换为字库地址的方法

5.7 汉字输出技术

从内部代码到点阵代码的转换是通过函数或输出字典实现的，这种转换称之为检字。在硬字库中，汉字点阵代码的地址一般是按大小排序的，当内部是等长且从头到尾连号紧凑排列时，则可把内码作为汉字库的地址，直接找到相应的字形点阵。如果内码按序但不是紧凑排列，可以用“对分法”找到相应的字形点阵。软字库检字比较复杂，涉及到读盘操作。磁盘中的汉字库一般按内码大小顺序排列，以文件形式存放，并预先在内存储器中存放磁盘字库文件的目录，同时建立一张字库所在盘的磁盘参数表。检字时，先将汉字内码转换成文件记录号，再将字库目录和磁盘参数表转换成磁盘地址（磁道号、扇区号和块号），最后按磁盘地址读出相应的字形点阵码。对于非压缩字库，可直接得到汉字的点阵码，而对于压缩字库还得采用还原算法才能产生汉字点阵码。

目前我国汉字信息处理系统的输出设备主要有打印机与显示器两种。

打印机多以点阵式的针打为主，打印速度一般为40~100汉字/秒，字形可变，有些打印机还配有汉字字库。根据点阵打印机的工作方式，应考虑字形点阵数据的传送方式。

CRT显示器的屏幕分辨率为640×200或640×400，字体多数为16×16点阵，也有24×24或32×32。高分辨率显示25行，中分辨率显示11行，每行40个汉字。

汉字输出到显示器有文本和图象两种方式。文本方式的输出原理如图5.8所示。它由同步计数器、显示RAM、汉字

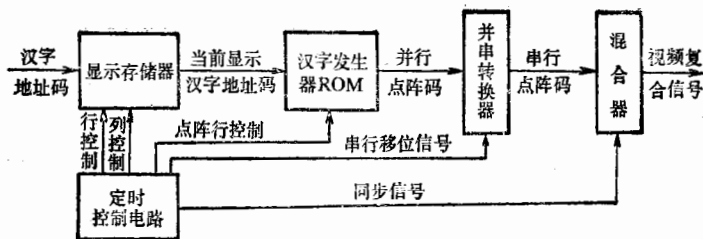


图5.8 文本方式的输出框图

发生器、并串转换器和混合器等组成。显示输出时，主机先将字模的地址依次放入RAM，然后由同步计数器控制RAM按顺序读出，送到汉字发生器，读取要输出的汉字字模点阵。同步计数器控制RAM每次只读出一个汉字点阵的一行，并行送入并串转换器，将其变为串行信息，作为视频信号与行扫、帧扫同步信号混合成复合视频信号送往CRT。

常用的汉字输入方式是图象方式，其原理如图5.9所示。首先建立一个屏幕显示缓冲区，使缓冲区中每个字的位与屏幕上的每个显示点一一对应。改变缓冲区中字位的0或1值，屏幕上相应点的亮暗也随之变化，只要将显示的汉字、数字或字符点阵信息按某种规律送入屏幕缓冲区的适当单元，就能在CRT屏幕的某个位置上显示出相应的字。

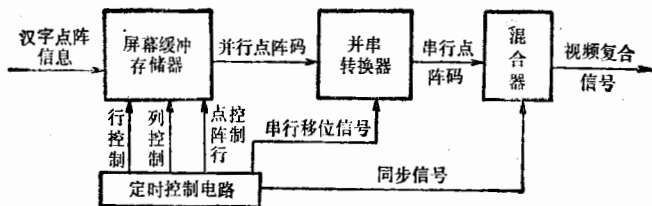


图5.9 图象输出方式框图

习 题 五

1. 软件汉化要作哪些工作？
2. 区位码中的区和位是如何规定的？为什么要把汉字分为两级？
3. 通用汉字系统采用什么技术？
4. 汉字中西文兼容的常用方法有哪些？各方法的主要特点是什么？
5. 一个好的汉字输入法应采用什么技术？
6. 输入字典实质是什么？
7. CCDOS操作系统的内码采用哪种方法表示？
8. 汉字字形信息以什么方式存入计算机？表示字形信息的常用方法是什么？
9. CCDOS操作系统的汉字输出是在什么方式下工作的？

第二部分

汉字数据库cdBASE技术

数据库技术的应用已广泛渗透到国民经济的各个部门。然而在人们的心目中，认为数据库只有在大型机上才可以使用。的确，最初的数据库系统是建立在大型计算机的基础之上的。但随着计算机技术的发展，现在就象中华学习机这类计算机家族中的“小不点”，也可以使用数据库技术了。这就是本篇将要介绍的汉字dBASE II，我们简称cdBASE或dBASE。这是世界上应用最为广泛的微机数据库系统之一，有“大众数据库”的美誉。

第一章 cdBASE的基本概念

6.1 什么是数据库

提到数据库 (DataBasc, 缩写为DB) 这个术语, 可能读者并不陌生, 因为很多广告、新闻报道中经常提到。然而, 要确切而严格的说出“什么是数据库”又是困难的。因为它不象数学公式和物理定律, 三言两语就能说明白。我们从以下几方面来说说“什么是数据库”。

6.1.1 数据库的特点

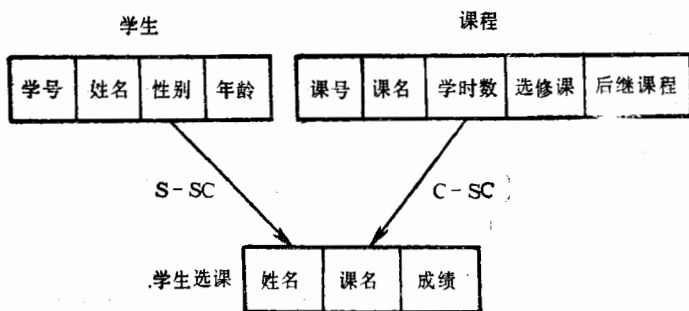
由于要正面回答“什么是数据库”这个问题比较困难, 因而很多资料都采用先“谈谈数据库应该具备一些什么特点”的办法来讨论数据库。我们也可如法炮制。归纳起来说, 数据库有如下一些特点:

1. 数据的结构化

所谓数据的结构化, 是指数据库能够描述的不单是某一个独立的、零散的事物, 而且还可以表达一个事物和其它事物之间的联系。例如: 学生是一个事物, 刻划学生的特征可以用: 学号、姓名、性别、年龄等等。数据库当然可以描述学号、姓名、性别、年龄之间的联系。这种联系称为事物的内部联系。课程又是另一个事物, 其特征可以是: 课号、课名、学时数、先修课程、后继课程等等。同样, 课号、课名、学时数、先修课、后继课程刻划了“课程”这一事物的

内部联系。对这一内部联系，数据库也能表达。学生和课程虽然是性质不同的两种事物，而事实上它们之间却有一种天然的联系，这种联系就是：一个学生可以选修若干课程；一个课程可有若干学生选修。学生和课程之间存在一种称为“学生选课”的联系，这种一个事物和其它事物之间的联系称为外部联系。数据库技术除了能够描述事物的内部联系以外，还能够描述如象“学生选课”这类外部联系。目前流行的高级语言（如：BASIC、FORTRAN、COBOL等）的文件系统，通常只能描述事物的内部联系。数据库与高级语言文件系统的根本区别也就在于此。

学生和课程的数据，经过结构化抽象以后，如图6.1所示。



16.1 数据结构化示意图

这就说明，如果我们要用数据库来解决某单位（工厂、机关、学校、厂矿或医院等）的数据处理问题（如工资、人事、产品等），那么首先必须抽象出单位的数据结构，这种结构也称为企业的整体逻辑结构。

从上可以看出，数据结构实际是企业中同类数据的一个

框架。通过数据的结构化处理就把各个框架联系在一起了。

2. 数据的独立性

数据的独立性是指数据库中的数据不依赖于具体的应用程序。数据和应用程序分家，这是数据库技术的又一个特点。其好处在于：如果应用程序改变了，存放的数据仍然可以不变；如果数据改变了，应用程序也可以不变。通常，高级语言中程序和数据是紧密相联的，数据为某种应用而存在，程序为某些数据而存在，彼此不能分开。这种方式带来的缺点是显然的。

3. 数据的共享性

所谓数据的共享性是指：凡是数据库中存放的数据，从原则上来说，各个用户都可以采用各自的语言和方式来使用它。例如：学生的数据，除供选课使用外，还可供与之有关的教师、教务部门、后勤部门等使用。彼此的使用可以互不干扰。但需注意，共享决不是随心所欲的乱用，而必须遵循一定的原则。

4. 数据的安全性、完整性和并发控制

由于数据的共享性，相应地带来了数据的安全性和并发控制等问题。

安全性是指保证数据库中数据的安全，防止未经授权的人存取或破坏或泄露。为此，采取了一些安全指施。如：规定口令，使用密码，限制存取权限等。

完整性就是要保证数据库中数据始终是正确的。数据库技术采取了一些校验措施来保证数据的完整性。

并发控制对各个用户存取数据的优先顺序和使用情况提供一种协调措施，以保证各个用户存取数据的正确和避免发生冲突。这是由数据库的并发机制来实现的。

5. 最小冗余度

所谓最小冗余度，就是指数据中重复数据最少。从理论上来说，可以做到完全没有重复数据，但出于对效率的考虑，允许少量重复。即是说，数据库中数据的冗余是有控制的。

6.1.2 数据库的种类

目前比较流行的数据库有三种：一种是层次模型的；二是网状模型的；第三种是关系模型的。三种模型中应用最广、最方便的是关系模型。

1. 层次模型

先看图6.2所示的层次关系。

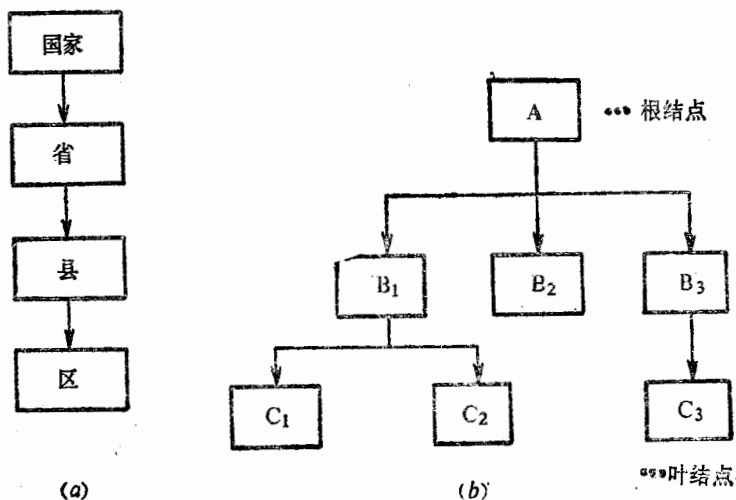


图6.2 层次模型示意图

在(a)中，一个国家下属省，省下属县，县下属区，

这是一级管一级，典型的层次结构。在(b)中，A下属 B_1 、 B_2 、 B_3 ， B_1 又分为 C_1 、 C_2 ， B_3 又分为 C_3 ，这也是一种层次结构。我们将每一个方框称为一个结点。最上面的结点称为根结点。不能更分枝的结点称为叶结点。从上图看出，层次模型符合如下两条：

①只有一个结点无双亲（结点之上不再有管辖的结点，如A）。

②其它结点有且仅有一个双亲。

由于层次模型类似自然界中一棵倒立的树，因而也称为树形结构。由此，我们不难理解根结点和叶结点的含义。有的也把箭头上方的结点称为双亲结点，下方的结点称为子女结点。层次模型也类似家族结构。

比较典型而又有代表性的层次模型的数据库是IMS (Information Management System) 系统。它是1969年由美国IBM公司推出的。

2. 网状模型

我们先看图6.3。

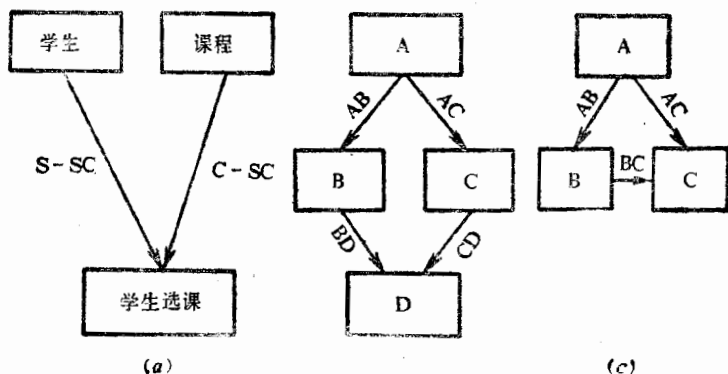


图6.3 网状模型示意图

仔细观察图6.3就会发现结点的组成有以下规律：

①可以有多个结点无双亲。（如图6.3（a））

②至少有一个结点有两个和两个以上的双亲。（如：学生选课结点有学生和课程两个双亲）

满足以上条件的模型称为网状模型。在这种模型中，箭头是极为重要的，它代表了存取数据的方向，称为存取路径。

网状模型的系统研究始于1969年美国数据系统语言协会下属的数据库任务组（Data Base Task Group），她提出了有关网状模型所应该具备的功能和特征的报告（通常称为DBTG报告）。这个报告几经修改，它对数据库，特别是网状模型的数据库的研究与发展起到了很好的促进作用。以后，人们又在DBTG报告的基础上研制了一系列的数据库系统。

3. 关系模型

层次模型和网状模型都是在文件管理系统的基础上发展起来的，因而其操作带有浓厚的文件系统的色彩。故而有些资料又把这两种模型统称为格式化模型或叫做传统的格式化模型。虽然格式化模型比起文件系统来说有了很大的进步，但对非计算机专业人员来说仍然是复杂的，这就妨碍了数据库技术的广泛普及和应用。

1970年，美国IBM公司的研究员E.F.Codd提出了关系数据库的概念，从而开拓了数据库研究和实践的新方向。

在关系模型中，所有的结点（事物）都用二维表格的形式给出。如前面图6.1的学生和课程模型，用关系表示如下。

学生

学号	姓名	性别	年龄
89123	张珊	男	15
89112	李实	女	14
...

课程

课号	课名	学时数	先修课	后继课程
89001	三角	80	代数	立体几何
89005	数据库	72	BASIC	专家系统
...

学生选课

姓名	课名	成绩
张珊	数据库	90
李实	代数	85
...

不论一个企业的逻辑结构有多么复杂，用这种表格总是可以描述出来的。上面的表格具有如下一些特点：

①表格的名字——数据库文件名

表格的名字是整个表格的一个代表。如：“学生”，“课程”，“学生选课”。这个名字是用户自己取定的。一

张表格又称为一个关系。在dBASE数据库中，表格名又称数据库文件名。

②表格的栏目——字段名

表格中最上面的一栏是栏目。如：姓名、性别，年龄等是“学生”表格的栏目，它共有四个栏目。在dBASE中，表格的栏目称为字段名。如：“姓名”是一个字段名，“性别”又是另一个字段名。在关系数据库中，栏目或字段名又称为属性。一个表格可以由多个栏目（字段）组成。

③表格的内容——记录

表格中有若干横行，每一个横行叫做一个记录。如：“89123 张珊 男 15”就是“学生”表格的一个记录。在关系数据库中，一个横行又称为一个元组。一般说来，一个表格中可以有若干个记录。

在以后的叙述中，我们把表格和数据库文件、栏目和字段不加区别的等同使用。

自1970年以来，关系数据库的研究与应用发展十分迅速。最著名的关系数据库是System R，它是为大型机研制的一个关系数据库，其性能与网状模型不相上下。此外，适宜于微型机上使用的关系数据库有dBASE II/III、FoxBASE等。目前国内应用最为广泛的数据库是dBASE II/III和FoxBASE，这两种数据库在用法上大同小异，其命令格式基本上是相同的。因此，只要掌握了这两种数据库中的任何一种，另一种也就会用了。

6.1.3 数据库的组成

“数据库”这个词前面已用到多次，但严格说来，可能是不恰当的，因为“数据库”还包含有很多内容。为了全面

了解数据库，我们还要引入以下几个概念：

1. 数据库系统

所谓数据库系统是指一个计算机系统中配备数据库以后所需人员、硬件和软件的全体。

数据库系统 (Data Base System, 缩写为DBS) 涉及的人员有两类：一是设计和维护数据库的骨干人员，称为数据库管理员 (Data Base Administrator, 缩写为DBA)。对于大型机器来说，通常配备DBA (微机通常不需要)。要求DBA有比较全面的知识，既不但精通计算机，还得熟悉企业管理，DBA常常由一个专家小组来担任。其职责是：

- ①建立数据库并负责装入数据。
- ②监督和控制数据库的使用、运行。
- ③维护和改进数据库。

另一类人员是数据库的用户。这通常是企业的领导人和工作人员。我们在设计一个具体的数据库时，必须考虑到用户的技术水平。

数据库需要一定的硬件系统的支持。由于各个系统不同，所需硬件要求也不尽相同。总起来说，内外存容量大、机器运行速度快对建立数据库大有好处。

数据库系统涉及到的软件有操作系统 (Operating System, 缩写为OS)、数据库管理系统 (Data Base Management System, 缩写为DBMS) 和应用程序。操作系统用于管理、调度和控制整个计算机系统 (包括硬件和软件)。没有操作系统的支持，DBMS不能正常运行。因而在使用数据库之前，一定要先装入操作系统。应用程序是用户用于检索、插入、删除和修改数据库中数据的一组软件。

2. 数据库管理系统

DBMS是一组非常复杂的软件，它是用户和机器之间的一个“翻译”，它的主要功能是：

- ①建立数据库
- ②管理数据库
- ③维护数据库
- ④数据通讯

DBMS四种功能都是通过高级语言的形式提供给用户或DBA的。语言分为两类：一是专门用于建立数据库的语言，称为数据定义语言(Data Description Language, 缩写为DDL)；另一类是专门用于检索数据库的语言，称为数据操纵语言(Data Manipulation Language, 缩写为DML)。

3. 数据库

按“数据库”本来的含义，它是指企业中有联系的数据的集合。这些数据通常保存在计算机的外存(如磁盘)上。我们可以用DBMS提供的方法和规则来存取。

数据库、数据库系统、数据库管理系统是互有联系而又有区别的三个概念。严格地说，数据库应当是指“数据”；数据库系统人员、软硬件的全体；数据库管理系统是用于管理“数据”的一组软件。我们通常所说的学习数据库，实际是学习数据库管理系统的各种语句和语法规则，从而用于设计企业所需要的数据库。

数据库的定义是各种各样的。

J.D.Uuman指出：数据库是存储在计算机系统 中的数据。

C.J.Date说：数据库只不过是一个计算机化了的簿记。

系统。

J.Martin说得比较具体：数据库是以有控制的冗余度存储在一起的、内部有联系的数据集合，它以最佳方式为一个或多个应用服务；数据的存储与应用程序无关；在数据库中增添新数据、修改和检索现有数据用的是一种公共的、有控制的方法。

苏岳威（美籍华人）给数据库下了这样的定义：数据库是从语义上和结构上与任何企业的用户相关联的、且本身又存在相互联系的操作数据的集合。

凡此种种，不一而足。归纳起来，我们可以这样来理解数据库：数据库既表示企业中数据的集合，又表示它还能对数据进行组织、存储和管理，并提供一套专门的技术和方法，使人们能够简便而迅速地检索、插入、删除和修改库中数据。

如果对这些概念还不清楚的话，可以先学习使用dBASE，获得一些感性知识后再来温故。

6.2 dBASE文件概述

1981年5月，ASHTON-TATE公司研制成功dBASE I，以后又经多次修改和扩充，形成了dBASE II的多种版本。由于dBASE II的简便和实用，受到广大用户的普遍欢迎，被誉为大众数据库，1984年7月，ASHTON-TATE公司在征求广大用户意见的基础上，对dBASE II进行了更大的改进，推出了功能更强的dBASE III。目前dBASE的最高版本是dBASE IV。dBASE II的各种版本在用法上大同小异，为了简便，我们统称为dBASE。

国内使用的dBASE有纯西文和中西文两种。一般把前者称为dBASE，后者称为cdBASE。但两种dBASE功能相似，用法相同。

dBASE处理的对象只能是表格。表格的名字称为数据库文件名，栏目的名字称为字段名，表格的内容听做记录。一般来说，一个数据库文件有一个名字、若干个字段和若干个记录。

6.2.1 文件类型及其扩展名

dBASE提供了7类文件，

文件种类	扩展名
数据库文件	· DBF
索引文件	· NDX
存储文件	· MEM
报表格式文件	· FRM
命令文件	· PRG (· CMD)
格式文件	· FMT
文本输出文件	· TXT

每个文件必须有一个文件名。文件名是字母或数字的序列，字符个数不能超过八个，字符中可以嵌入冒号。注意：

①构成文件名的字符只能是字母或数字，第一个字符必须是字母（英文），后面可以是字母，也可以是数字。也可以直接用汉字作文件名，但不能超过4个汉字，因为一个汉字要占两个英文字符位置。

②文件名的各字符之间不能夹杂空格（白）。为区别各类文件，对不同类的文件给出了不同的内部标志，这种标志称为扩展名。

数据库文件是dBASE中最基本的文件，它用于存储数据库的基本信息。索引文件是在数据库文件的基础上建立起来的一个有序文件，其作用是便于对数据库文件进行快速查找。存储文件是把内存变量组织成一个文件，以便永久性地保存内存变量的值。报表格式文件包含了打印报表所需要的格式信息。命令文件允许把若干命令组织在一起，从而生成一个文件或程序。格式文件是为了建立常规屏幕格式以便数据的使用和输出。文本文件可用于系统通讯和记载dBASE的处理活动。

我们介绍dBASE，实际上就是介绍这7种文件的建立方式以及与之有关的各种操作方法。如果能够熟练地掌握这种文件的建立、打开、操作和关闭命令，那么在实际应用中就会更加得心应手，要编写出质量较高的应用程序就指日可待了。

6.2.2 使用符号约定

为叙述的方便，我们约定以下符号的含义。

{...}花括弧中的内容必选一项，亦可选多项。

[...]方括弧中的内容可根据需要取舍。

<...>尖括弧中的内容由用户提供。

注意：在机器上练习各种语句时，花括号、方括号、尖括号并不敲入机内，而只敲入括号中的实际内容。

/斜线代表或者的意思。

↵代表回车键，即键盘上的↵或RETURN或ENTER键。

^代表键盘上的CTL键。

^W表示先按住CTL键不放手，再敲W键。

ESC代表键盘上的ESC键。

在实习dBASE之前，请参见键盘上的符号键。

6.3 常量、变量、函数及数据类型

6.3.1 常量、变量和函数

常量就是其值固定不能变化的量，也称常数。常量包括数值常量、字符常量和逻辑常量。数值常量与数学中的常数相同。字符串常量是用引号将字符序列括起来，如：“ABC”，“我是学生”。逻辑常量是逻辑值.T.或.F.（表示真或假）。

变量就是其值是可以变化的量。变量包括两种：一种是与数据库文件结构有关的字段名，这种变量称为字段变量。如前面所述的学号、姓名、性别，都是字段名。另一种变量类似数学中的X、Y一类的未知数，它用于存放数据库操作中一些临时性的结果，称为内存变量。

变量名的构成方法同文件名一样，区别仅在于变量名最多允许10个字符。

函数是具有专门功能的一些符号。dBASE提供了如下一些函数：

函数	功能
INT (〈数值型表达式〉)	取出值的整数部分。
TYPE (〈表达式〉)	测定表达式的类型。
VAL (〈字符型表达式〉)	将仅由数字组成的字符转换成数值。
STR (〈算术表达式〉)	将数值型转换成字符型

〈长度〉, [〈小数位数〉])

LEN (〈字符型表达式〉) 求字符串的长度, 即字符个数。

\$ (〈字符型表达式〉, 〈开始位置〉, 〈长度〉)

取子字符串。

@ (〈字符表达式-1〉, 〈字符表达式-2〉)

求字符表达式-1的值在字符表达式-2中的开始位置。

I (〈字符型表达式〉) 将小写字母转换成大写字母。

TRIM (〈字符串〉) 去掉字符串末尾的空白(格)。

DATE () 求系统日期。

求当前记录号。

* 测定记录是否注有删除标记。

EOF 判断文件是否处于结束。

& 替换出字符型变量的值。

6.3.2 数据类型

数据类型是指常量、变量或函数的值所具有的类型。中华学习机上的dBASE提供了三种数据类型: 数值型、字符型和逻辑型。凡是取值为算术值的变量或常数, 其数据类型均为数值型。凡是取值为字符的变量或常数, 其数据类型均定为字符型。凡是取值为“真”或“假”的变量或常数, 其数据类型都指定为逻辑型。

常量的数据类型是按常量的书写形式来确定的。如果按照数值常量的形式来书写, 那么其类型是数值型的, 其余类推。

字段的数据类型是根据字段名的实际取值情况, 由用户

在创建文件结构时用下面的字母来指定的：

C——代表字符型 (Character)。如果某个字段名取值是字符 (英文字符或汉字或空格)，那么必须用C来指明其字段类型。如：姓名字段，其数据类型就一定是C。

N——代表数值型 (Numeric)。如果字段值要参与加、减、乘、除等算术运算，那么该字段就一定用N指明。如：年龄，奖学金等字段。

L——代表逻辑型 (Logical)。如果字段取值.T.(真)或.F.(假)，那么就必须用L指明。

注意：有些字段既可指定为N，也可指定为C，例如：学号字段就是如此。判定的准则是：若学号可能参与算术运算，则一定为N型。否则，尽管学员全部是数字，也仍然指定为C型。

在定义字段时，要涉及到字段宽度这个概念。所谓字段宽度是指字段名所取值能够容纳的最大字符个数。例如：“姓名”字段的宽度若为8，那么就表示数据库中保存的所有人的名字不能超过8个英文字符(即4个汉字)。字段的类型和宽度一经确定，以后就只能按照这种类型和宽度来存取数据的值。当然，根据需要也可以对字段类型和宽度进行修改。

6.4 表 达 式

dBASE表达式可分为：算术表达式、关系表达式、逻辑表达式和字符串表达式。前三类表达式类同于BASIC语言。

算术运算符有：+，-，*，/用于构成算术表达式。
加，减，乘，除

关系运算符有：>， >=， <， <=，
大于， 大于等于， 小于， 小于等于，
< >或#。

不等

用于构成关系表达式。

逻辑运算符有：.NOT., .AND., .OR.用于构成逻辑
非， 与， 或

表达式。

字符串运算符：+， -， \$。用于构成字符串表达式。

字符串表达式是指：字符串常数或字符串变量或字符串
函数，或用字符串运算符把上述各项连结而成的式子，都可
以叫做字符串表达式。

+——把两个字符串直接连串起来，生成一个新的字
符串。

-——作用同+，区别仅在于：如果“-”号左边的字
符串末尾有空白，则把空白移到结果的末尾。如：“abc
” + “def” ⇒ “abcdef”，而“abc” - “def”
⇒ “abcdef” 其中“ ”表示一个空白。

\$——测定字符串中是否含有子串。规则是：若\$的左
端包含在右端之中，则运算的结果为真，否则为假。如
“ABC” \$ “XYABCW”的结果为.T.(真)。

注意：

②可以用圆括号“(”和“)”来改变运算的优先顺
序。

②表达式的值所具有的数据类型决定了表达式的类型。

③如果表达式中包含了多种运算，其运算的优先顺序
是：算术运算，关系运算、字符串运算和逻辑运算。

习 题 六

1. 举例说明数据库的特点。
2. 在图6.3(b)(c)中, 哪些结点有两个双亲? 各自的双亲是哪些结点?
3. 举一个表格的实例, 并说明其文件名、字段名以及它的记录。
4. 结合3题, 说明字段类型和字段宽度的含义。
5. 如何给dBASE的文件和字段取名? 为什么文件名和字段名的字符间不能夹杂空格(白)?
6. 常量、变量以及表达式的数据类型是如何确定的? 分别举例说明之。
7. 求下列字符串运算后的结果。
 - ① "XYZ" \$ "XYZWZYX"
 - ② "XYZWXYZ" \$ "XYZ"
 - ③ "ABCD_ _" + "EFG"
 - ④ "ABCD_ _" - "EFG"

第七章 cdBASE的安装和运行

本章介绍如何在中华学习机 CEC-I 上安装汉字 dBASE 以及一些简单命令的用法。

7.1 系统要求

在中华学习机上运行 cdBASE, 它需要一定的硬件和软件支持。

7.1.1 硬件环境

在中华学习机上运行 CdBASE, 必须满足如下的硬件环境:

1. 基于8080、8085或Z80的微处理机系统。
在CEC—I上, 必须插上一个Z80卡。
2. 至少64K字节的内存。
对西文dBASE来说, 至少需要48K字节内存。
3. 显示器(可以用家用电视机代替)。
4. 至少一个软盘驱动器。
5. 最好有打印机。

出厂的中华学习机只留有一个插件槽, 用于插Z80卡后就不能接打印机。这时可选配五槽口扩展接口板解决这个问题。

7.1.2 软件环境

除了硬件以外，还必须有软件的支持。

1. 中文CP/M操作系统。

通常用APPLE II CP/M Ver 2.20B代替。

2. cdBASE数据库管理系统。

目前，中华学习机CEC—I上使用的CP/M操作系统和cdBASE拷贝在同一张软盘上，由于CEC—I内存容量有限，只复制了CP/M最核心的部分。这样作的目的是为了给用户留下较多的磁盘空间。

7.1.3 系统文件及其作用

我们把拷贝有CP/M操作系统和cdBASE文件的软盘称为系统盘，其文件称为系统文件。要正确运用cdBASE，系统盘上必须有如下一些系统文件：

文件名	扩展名	占用空间	作用
DBS	.COM	25K	主程序。
DBASEMAI	.OVR	7K	主覆盖程序。
DBASEMSG	.COM	8K	处理出错提出信息。
DBASEAPP	.OVR	4K	处理添加记录。
DBASEBRO	.OVR	2K	处理浏览记录。
DBASEJOI	.OVR	1K	处理连接记录。
DBASEMOD	.OVR	3K	处理文件结构的修改。
DBASERPG	.OVR	4K	处理报表文件。
DBASESRT	.OVR	2K	处理文件排序。
DBASETTL	.OVR	1K	处理合并细项。
DBASEUPD	.OVR	1K	处理更新文件。

DBASEMSC .OVR 4K 其它。

PIP .COM 操作系统程序。

在整个系统文件中，PIP.COM是操作系统的 一个命令，其作用是负责装入外围交换程序，供以后的磁盘文件和外围传送操作之用。其余12个程序均是与dBASE有关的程序模块。dBASE系统成功的采用了覆盖技术，使得在如象中华学习机这种内存容量不太大的机器上仍然可以使用dBASE数据库。从上表已经看到，12个模块中就有10个模块是用覆盖技术来实现的（程序扩展名为.OVR的是覆盖文件）。在这些覆盖模块中，最重要的是主覆盖模块——DBASEMAI.OVR，它由DBS.COM调入，作为常驻模块，负责对输入的dBASE命令进行解释。如果输入进来的命令DBASEMAI.OVR能够处理，那么就立即进行处理。否则，立刻调用其它功能模块覆盖主模块，并即刻处理之，处理结束后再恢复被覆盖的主模块DBASEMAI.OVR。如果输入或执行命令过程中发生了错误，则由DBASEMSG.COM调出相应的出错信息进行处理。其余模块的作用是显然的。在我们的使用中，已将上述文件复制到一张软盘上了。

7.1.4 Z80卡的插入方法

中华学习机CEC—I虽然具有64K字节的内存容量，但仍然不能直接在其上运行dBASE。因为cdBASE要求主机芯片是8080、8085或Z80的，而中华学习机CEC—I不满足这一要求（APPLEII也如此）。为了不改动CEC-I的硬件，可以采用插上一个Z80卡的方法来使之满足要求。如何插入Z80卡？

1. 小心打开中华学习机CEC—I的面板盖。（事先一定要拔下电源插头，断开电源）。

2. 将一个Z80卡对准插件，然后平稳地插入到槽口内（Z80卡上有指示灯的那一面对准用户。切记不要把方向弄反了）。

3. 由于Z80卡占据面板盖位置，因而使用期间不能上面板盖。对于选配了五槽口扩展接口板的用户，不必取下面板盖。

7.2 cdBASE的启动

在使用cdBASE之前，必须先将cdBASE装入机内。根据主机电源事先是否已打分开为冷启动和热启动两种方式。

7.2.1 冷启动

所谓冷启动就是事先没有打开主机电源的情况下来启动系统。

1. 请事先准备好系统软盘。其上有cdBASE的12个系统程序和CP/M操作系统。

2. 把系统盘插入到A驱动器中，如果只有一个驱动器，把系统盘直接插入就行了，关上驱动器的“门”。

3. 打开屏幕电源后再打开主机电源。稍候，屏幕显示：

```
APPLE II                CP/M
56K          Ver.       2.20B
(C)1980      MICROSOFT
A>
```

这是有关版权的提示，A>是操作系统提示符，它的出现表示CP/M操作系统安装成功。

4. 在A>提示下，敲入：DBS↵。这时可以看见Z₈₀卡的指示灯在闪亮，同时听到读磁盘的声音。稍候，屏幕显示：

```
请输入日期 (MM/DD/YY) 或回车:
```

这时系统要求用户敲入日期或回车键。如果需要输入日期，必须按月/日/年的形式输入。也可以直接敲回车键↵。假设我们敲入回车键↵，屏幕显示：

```
请输入日期 (MM/DD/YY) 或回车: ↵  
汉字DBASE I V1.0
```

出现一个小圆点，表示cdBASE安装成功，这时就可以开始使用CdBASE的各个命令了。如果有两个驱动器，可用SET DEFAULT TO B: ↵来设置默认工作盘。

7.2.2 热启动

所谓热启动就是在主机电源已经打开的情况下再来启动系统。热启动和冷启动类似，唯一的区别是将冷启动的第3步：打开主机电源改成：敲：PR #6↵，其效果和屏幕提示与冷启动相同。

注意：PR #6是CEC—I机上热启动的操作标志。字母必须用大写（压下Caps Lock键后显示的字母均是大写形式）。

2. 保存表达式的值

如果要把一个表达式的值保存起来，以备后继命令中使用，就可用存储命令STORE。

格式：STORE(表达式)TO(内存变量名)

作用：将表达式的值传送到内存变量名中保存起来，以备后用。

这里的表达式可以是一个具体的常数，或一个有值的变量名，也可以是一个表达式。常数、变量名、函数都是表达式中的特殊情形。

例3. 将 $5 * 6$ 的值传送到内存变量X中。

·STORE 5 * 6 TO X ↵

30 (系统将 $5 * 6$ 的值30自动显示在屏幕上)

例4. 求 $b^2 - 4ac$ 的值并送到D中。设， $a=3$ ， $b=4$ ， $c=5$ 。

·STORE 3 TO A ↵

·STORE 4 TO B ↵

·STORE 5 TO C ↵

·STORE B * B - 4 * A * C TO D ↵

·? D ↵

例5. 将字符串保存在一个内存变量W中。

·STORE “学习和使用数据库技术” TO W ↵

·? W ↵

学习和使用数据库技术。

为了简洁起见，在以后的书写中，凡是命令中要敲空格的地方，我们都留出一个空白而不再用_标出，例如：STORE 5 * 6 TO X ↵。读者在上机实习时，注意留出空白的地方

必须至少敲一次空格键。

通过STORE命令就可以把表达式的值保存到内存变量中。一个内存变量一旦得到了值，就有了定义，也叫做定义内存变量。除了用STORE命令可以定义内存变量名以外，还有其它方法，以后将会看到。内存变量只能在内存中起作用，一旦断电，内存变量及其值也就不复存在了。在同一个dBASE程序中，不同名的内存变量至多64个。

3. 释放内存变量

如果程序或对话中用到不同的内存变量名超过了64个，就会妨碍对内存变量操作的继续进行。解决内存变量个数超界的办法有三：一是重复使用变量名，即先前已定义的内存变量名，再重新赋与新的值；二是去掉一些内存变量名，这就是所谓的释放内存变量；三是将内存变量组织成文件。

格式：RELEASE[(<内存变量表>)/ALL]

作用：释放指定的变量名。

其中：内存变量表由一个或多个变量名构成。如果是多个变量名，那么各变量名彼此间要用逗号隔开。ALL表示所有的意思，命令中若选用了ALL就表示释放所有的内存变量名。

例6. 释放内存变量X, A, B的值。

```
·RELEASE X, A B↵
```

·? X, A, B↵ (由于X, A, B已释放掉，就不能再显示其值)

变量没找到

例7. 释放所有内存变量。

```
·RELEASE ALL↵
```

注意：方括号中的项称为待选项，根据用户需要而确定

取舍。在释放命令中，如果不选用待选项，也表示释放内存变量名。内存变量一经释放，其名其值都不存在了。

4. 存储文件的建立和调用

前已述及，内存变量只能在内存中存在，一旦断电，所有值均会全部去掉。如果我们希望长期保存内存变量的值，就必须把内存变量名及其值组织成文件。在计算机中，只有用文件形式组织起来的内容才能脱离内存而永不消失。内存变量文件（或称为存储器文件）是我们所要介绍的7种文件中最简单的一种，其建立的命令格式如下：

格式：SAVE TO 〈文件名〉

作用：将已有定义的内存变量组织成一个文件。

例8. 将内存变量R、X和Y组织到存储文件中。

```
·STORE 5 TO R↵
```

5

```
·STORE 5*8 TO X↵
```

40

```
·STORE “学习数据库” TO Y↵
```

学习数据库

```
·SAVE TO SFILE↵ （SFILE是存储文件的名字）
```

通过SAVE命令，就把X和Y的值连同它们的名字一起保存到磁盘上的 SFILE 文件中去了。以后需要再用到X和Y的值，就可从 SFILE 文件中调用。调用存储文件的命令是：

格式：RESTORE FROM 〈文件名〉

作用：将指定存储文件中的内存变量名及其值调入内存。

例9. 将SFILF文件中的内存变量调入内存。

·RESTORE FROM SFILF ✓ (调用存储文件)

如何知道存储文件中内存变量已调入到内存呢? 可用显示内存变量的命令显示之。

格式: DISPLAY/LIST MEMORY

作用: 在屏幕上显示内存变量的名和值。

例10. 把从SFILF文件中调入的内存变量显示在屏幕上。

·DISPLAY MEMORY ✓ (以下是屏幕显示)

R (N) 5 (变量名, 类型, 值)

X (N) 40

Y (C) 学习数据库

合计 03 变量被使用 (用了3个变量)

00022 字节被使用 (占用22个字节)

5. 清屏幕

经过一系列的对话操作后, 屏幕上的字符可能很零乱, 我们很希望象黑板上的字一样全部擦掉, 这可以用清屏命令。

格式: ERASE

作用: 清除屏幕上的所有字符, 并将光标定位在屏幕的左上角(0行0列位置)。

例11. 清除屏幕上的所有字符。

·ERASE ✓ (字符清除了, 光标定位在左上角)

6. 退出dBASE, 回到操作系统状态

如果我们这次上机结束, 或者希望回到操作系统状态之下, 就要用到退出命令。

格式: QUIT

作用: 无条件回到操作系统提示状态。

例12. 退出数据库, 回到A>状态。

·QUIT↵ (以下是屏幕提示)

***END RUN CDBASE I ***

A> (结束运行 CDBASE I, 回到操作系统)

注意: 当退出cdBASE时必须用QUIT命令。这时系统自动关闭所有打开的文件, 并将必要信息登录到磁盘上。如果退出dBASE回到A>后又想立刻进入dBASE, 那么只需敲入: DBS↵就行了。

7.4 技术规格和操作总则

dBASE用起来简单和方便, 但仍然有一些限制, 实际应用中必须遵守。

7.4.1 技术规格

1. 每个数据库文件至多 65535 个记录。由于出厂的 CEC—I 只有一个驱动器, 系统盘上所剩空间不多, 实际上容纳不了这么多个记录 (可选用双驱动器接口板, 扩充为两个驱动器)。

2. 每个记录至多 32 个字段。

3. 每个记录至多 1000 个字符 (或 500 个汉字)

4. 每个字符型变量至多容纳 254 个字符 (或 127 个汉字)。

5. 数值型数据至多 10 位有效数字。

6. 数值的表示范围是 $\pm 1.0 \times 10^{-63} \sim \pm 1.8 \times 10^{63}$ 。

7. 至多64个内存变量，占用总空间1536字节。
8. 报表标题最长254个字符（或127个汉字）。
9. 在求和（SUM）命令中，至多5个数值型字段。
10. 可以同时打开16个文件，其中至多两个数据库文件。
11. 一个数据库文件上至多允许建立7个索引文件。

7.4.2 操作总则

1. 每个命令必须以命令动词作为开头。
2. 每个命令的书写必须符合语法格式。
3. 命令中的待选子句的顺序是任意的。
4. 命令行的最大长度是254个字符（包括空格）。
5. 若命令动词等关键字多于4个字符，可以只写前面的4个字符。但拼写必须正确。如DISPLAY可以写成DISP。
6. 一个命令一行敲不完，可以用分号结尾，以示下行为续行。
7. 最好不用命令动词和关键字作文件名或变量名。
8. 在机器上敲入命令行时，命令动词或关键字前后必须至少留一个空格且都必须用大写字母。

习 题 七

1. 请面对一台中华学习机，练习cdBASE的启动。
2. 练习下列字符的输入。
学习和应用cdBASE数据库技术。
3. 说明？和STORE两个命令的异同点。

4. 自己定义若干个内存变量，要求变量的数据类型分别包括数值、字符和逻辑型。再建立成存储文件并调用之。

5. 释放内存变量的命令RELEASE对存储文件中的变量是否起作用？为什么？请用例子来试试看。

6. 有人说：清屏幕命令ERASE执行后，不但把屏幕上的字符去掉了，同时也把内存变量清除了。这种说法对吗？为什么？

7. 有人说：用QUIT命令来退出dBASE，对磁盘上的文件来说是最安全的。这是真的吗？为什么？试试看。

第八章 数据库文件及其操作

数据库文件是cdBASE最基本而又最重要的一种文件，很多文件和命令都和数据库文件直接有关。可以说，把数据库文件的建立、操作和使用方法搞清楚了，其它文件也就不难弄懂了。

8.1 生成数据库文件的基本方法

对于不同的情况，可以用不同的方法来建立和生成数据库文件。数据库建立后，可以马上输入数据。

8.1.1 数据库文件的创立

1. 创建数据库文件

我们要利用计算机来检索信息，那么首先必须存储信息，而存储信息的最好形式就是组织成数据库文件，用到的命令（也可以称为语句）格式如下：

格式：CREATE 〈文件名〉

作用：允许用对话的方式建立一个数据库文件，并将文件名字自动登录在磁盘目录中。

其中：CREATE 是建立数据库文件的命令标志，其中文含义是创立创建的意思。文件名是用户给数据库文件取的名字。下面用一个例子来说明CREATE命令的用法。

例1. 将如下的商品信息建立一个数据库文件。

商品信息

货号	品名	单价	型号
120	自行车	179.93	环球-2
145	电冰箱	1030.96	北极星
138	电视机	1140.00	美乐-9
127	收录机	715.74	神笛牌
158	缝纫机	161.26	家佳乐

在用计算机来建立有关商品信息的数据库之前，首先要确定以下几件事情：

①数据库文件的名称

文件的名称既要取得短小又要能够体现存储信息的意义，使人一目了然。本例就用“商品”为数据库文件名。

②数据库文件的字段名

通常可以直接用表格的栏目名称作为字段名。但要注意，字段名太长超过10个英文字符或5个汉字时必须用缩写形式。本例就直接用栏目：货号、品名、单价、型号作字段名。

③字段名的类型

所谓字段名的类型是指字段对应的那一列取什么样的值：是数值吗还是字符？或者是逻辑值。这必须根据实际情况和以后的应用来确定。在本例中，货号确定为字符型(C)，理由是以后对货号不可能进行加、减、乘、除运算；品名只能确定为字符型(C)；单价必须确定为数值型(N)，且有2位小数；型号也只能是字符型(C)。本例中没有用到逻辑型(L)。

④字段宽度

前面第一章已经说过，字段宽度是指字段值中所能容许

的最大字符个数。这通常要根据实际应用来确定。本例中假定：货号最多3位；品名最长6个字符（3个汉字）；单价最多7位数字，其中包括1个小数点位，而小数点之后有2位小数，实际只有4位整数；型号最多6位（3个汉字）。系统规定：字符型字段的最大宽度为254，数值型字段的最大宽度为19，逻辑型字段的宽度固定为1。

需要强调的是：如果我们是为应用单位设计数据库，那么必须征求用户意见，以用户要求为准。如果用户要求与cdBASE的规定有矛盾，必须进行协调。

现在假设系统已处于点状态，可直接敲入，CREATE 商品↵（以下是屏幕显示的情况）

·CREATE 商品↵（以下是提示信息，要求用户回答）

输入记录结构如下：

字段 名字，类型，宽度，小数位数

001 —

光标在“—”处闪动，要求我们回答：第1个字段的名称、类型、宽度（长度），如果有小数，还需要回答小数位数（即小数点之后有几位小数）。我们可以按照前面约定，用敲键的方式进行回答。如敲入：货号，C，3↵后系统又要求输入第2个字段，屏幕显示（汉字输入方法请参见第二章2.3节）如下：

字段 名字，类型，宽度，小数位数

001 货号，C，3↵

002 —

我们再按上述方式回答第2个字段，……。如敲入：品名，C，6↵……，直到全部字段定义完毕。为使读者看清

敲键的全部过程，我们再把定义过程显示如下：

·CREATE 商品↵

输入记录结构如下：

字段 名字，类型，宽度，小数位数

001 货号，C，3↵

002 品名，C，6↵

003 单价，N，7，2↵

004 型号，C，6↵

005 ↵ （定义字段结束）

注意：“字段，名字，类型，宽度，小数点位置”以及字段这一列下面的数字“001、002、003、004”等都是系统给出的提示信息，不要敲入，也不能由用户敲入。而“货号，C，3↵”等是必须由用户敲入的。目前，中华学习机CEC—I上使用的cdBASE的提示信息还没有全部汉化，为使不熟悉专业英文的读者看懂提示信息，我们已将相应的英文提示信息全部译成了中文，并提示在原英文的位置上。

2. 装入数据

当光标处于一个新字段时，敲入回车键就表示字段定义完毕。本例中定义了4个字段，当光标在“005”后闪动时敲入↵，表示字段定义结束，屏幕就提示：

现在就输入数据吗？Y/N

询问是否立刻输入数据。如果敲入字母Y，就表示现在需要输入数据。如果敲入其它字符就表示不立刻输入数据，系统将文件结构保存起来后回到点状态，此时我们可以进行其它操作。本例中敲入Y，表示需要立刻输入数据，屏幕显示为：

RECORD# 00001 （记录号，系统自动提示）

货号	:	—	:
品名	:		:
单价	:		:
型号	:		:

光标在第1个字段的第1个字符位置上闪动，这里我们可以依次输入记录值。冒号之间的位置表示了字段的宽度。现在依次敲入：

120自行车179.93环球-2

第1个记录就算输入完毕，再依次输入第2个记录，…直到所需记录全部输完为止。

在输入数据时，如果敲入的字符占满了字段宽度所给的位置（冒号之间的位置），系统把光标自动跳到下一个字段，开始接收下一个字段的数据；如果冒号之间的位置没有被输入字符占满，那么必须敲入一个↵键，强行使光标跳到下一个字段。光标所处的位置就是当前需要处理的位置。

当光标处于一个新记录的第1个字段的第1个字符位置时，敲入↵或^W，就可以结束数据的输入。这时系统自动将输入的数据及整个文件结构保存在磁盘上，然后回到点状态。这样，一个数据库文件就算建成了。本例中我们一共要输入5个记录，光标处在第6个记录时，敲入回车键↵，系统回到点状态。

如何才能知道输入的数据是否正确呢？可以用显示命令显示之。

· USE	商品↵	(打开文件)
· DISPLAY	ALL↵	(显示文件记录)
00001	120自行车	179.93 环球-2
00002	145电冰箱	1030.96 北极星

00003	138电视机	1140.00	美乐-9
00004	127收录机	715.74	神笛牌
00005	158缝纫机	163.26	家佳乐

至此，一个完整的数据库文件就建成了。用同样的方法，可以创建其它数据库文件。

3. 误操作的纠正方法

在建库和装入数据的过程中，难免不敲错键码，我们把敲键和输入数据过程中所发生的错误统称为误操作。根据作者的教学经验，初学者在以下几方面最容易出错：

①在文件名字中夹杂空格（白）

如将数据库文件名“商品”，误敲为“商 品”，用户的想法是在“商”和“品”之间留一个空格，看起来美观些，但好看未必中用。违反了文件的取名规则。改正的办法是重新敲入正确的文件名，系统将既往不咎。

②在字段名字中夹杂空格（白）

这也是系统规则所不允许的。发生这类错误，系统将给出更正错误的机会。例如：将“货号”误敲为“货 号”，屏幕提示：

字段	名字，类型，宽度，	小数位数
001	货 号，C， 6	（货号间有一个空格）
错误字段		（提示字段有错）
001	—	（更正的机会）

我们重新敲入：货号，C，6就行了。凡是定义字段中发生的错误，如：应该敲逗号的地方误敲为其它符号，定义类型不用字母C、N、L，宽度超界等等，系统都给出这种提示。

③小数位数给得不合适

系统要求含有小数的数值型字段的宽度至少比小数位数要多2倍。例如：如果单价字段的小数是2位，那么字段宽度应大于或等于4。下面的定义都是错误的：

003 单价, N, 4, 3↙ 或003 单价, N,
4, 4↙

顺便再次强调：各类字段的最大宽度必须在系统规定的范围内。如果实际应用中某个字符型字段的宽度要超过254个字符，那该怎么办？请读者考虑。

④在回答“现在就输入数据吗？”时误敲Y以外的键

我们的本意是想敲字母Y以便立刻输入数据，但由于初次上机，心情激动以至于手指都有点不听使唤，无意间碰到了Y以外的键，因而造成给系统以违心的回答，系统便回到点状态，挽回局面的最好办法是在点状态下打开文件后再敲入：APPEND↙，例如：

· USE 商品↙

· APPEND↙ (以下就可以按正常操作输入数据)

实际上，凡是希望对某一个数据库文件继续输入数据，均可用此形式。

⑤敲入的数据有误

当我们敲入后面的数据时才发现前面的字符有错，这是常有的事，不必惊慌。改正的办法是利用系统提供的控制键将光标移到出错处，重新输入正确字符就行了。

在APPEND命令下可用的控制键有：

^E——光标移到上一个字段。

^X——光标移到下一个字段。

^S——光标左移一个字符位置。

^D——光标右移一个字符位置。

^V——允许在光标处插入字符。

◀——光标向左回退一个字符。

^Q——正在操作的文件不存盘，回到点状态。

^W——文件存盘后回到点状态。

这些控制键，为我们上下左右移动光标提供了方便，同时也为修改数据失误提供了方便。但要注意：这些控制键只有在输入数据的过程中才有效，如果已经退出或回到点状态下，那么就“离柜不认”了，这种情况只好用专门的命令来修改，后面将会看到。

8.1.2 选取工作区和打开数据库文件

cdBASE的任何操作，都只有在一定的内存区域中才能进行。系统将内存分成主区和辅区两个部分，每一部分称为一个工作区。

1. 选取工作区

如果在进行操作之前，我们希望将要进行的操作在某一个工作区内进行，那么就可以事先选定区域，这就是工作区的选取。

格式：SELECT <PRIMARY/SECONDARY>

作用：选择主区(PRIMARY)或辅区(SECONDARY)作为操作命令的工作区。

例2. 选择主区为工作区。

· SELECT PRIMARY ✓

(以后的操作就在主区内进行)

注意：

①如果我们没有明确选定工作区，那么就隐含说明使用的是主区。

②工作区一经选定，只要没有重新选择，其后的操作就一直在该区内进行。

③对于涉及到两个数据库文件的操作命令，如，JOIN、UPDATE（后面介绍）等必须选择工作区。

实际上，选取工作区就是为其后的cdBASE操作划定一个活动场所。刚选定的工作区或正在使用的工作区称为活化工作区或当前工作区。

2. 打开数据库文件

任何文件在使用之前都必须先打开，但各种文件的打开方式并不完全相同，打开数据库文件可用USE命令。

格式：USE[〈文件名〉]

作用：打开指定的数据库文件并把记录指针指向文件的第1个记录位置。

记录指针是cdBASE中极为重要的概念，它是系统用来标识记录位置的一个内部计数器。随着命令的执行和记录位置的推移，指针的计数值也在随之变化。因而，我们把指针正指向的那个记录称为当前记录或活化记录。

例3. 在主区内打开“商品”文件。

· SELECT PRIMARY ✓ (选取主区)

· USE 商品 ✓ (打开文件)

· DISPLAY ✓ (显示第1个记录)

00001 120 自行车 179.93 环球-2

注意：

①一个数据库文件一经打开，就可对它进行操作，操作次数不限。

②在同一工作区内只能打开一个数据库文件，如果又打开了另一个数据库文件，先前的数据库就会自动关闭。

③若USE之后不带数据库文件名，则表示关闭当前打开的数据库文件。

④刚打开或正在使用（先前已打开）的文件称为活化文件。

8.1.3 复制数据库文件

如果先前一个文件也没有，那么要得到数据库文件就只好用CREATE命令来建立。如果事先建立了一些数据库文件，现在希望将某个数据库文件的某些行（记录）和列（字段）抽取出来，从而组成一个新数据库文件，那么可以用复制命令。

1. 复制数据库记录

格式：COPY TO <目标文件名> [<范围>] [FOR <条件>] [FIELDS <字段表>]

作用：将活化数据库文件复制到目标文件中。

其中：

<目标文件名>就是要新建立的数据库文件名。

<范围>用于指出数据库文件的哪些记录要复制到新文件中，它提供了三种可选用的方式：

〈范围〉	ALL	表示所有记录
	NEXT _n	从当前记录开始的下面n个记录。
	RECORD _n	第n个记录。

在一个具体应用中，三种情况只能给出一种。

POR <条件>对记录给出限制条件，表示满足条件的记录才被选取。条件可以是一个关系表达式或者逻辑表达式。

FIELDS <字段表>表示旧文件中的哪些字段要复制到新文件中。如果是多个字段的话，字段彼此间要用逗号隔开。

这里对<范围>、<条件>和<字段表>的解释，也适用于其它命令。看看下面的例子。

例4. 把价格低于500元的商品复制到DSP文件中（DSP是新文件的名字）。

· USE 商品↵

· COPY TO DSP ALL FOR 单价<500↵

00002记录被拷贝 （提示信息，表示有两个记录被复制了）

· USE DSP↵

· DISPLAY ALL↵

00001	120	自行车	179.93	环球-2
00002	158	缝纫机	163.26	家佳乐

例5. 生成一个高价商品文件GSP，文件中只需要品名及其单价。（假设价格大于1000元的为高价商品）

· USE 商品↵

· COPY TO GSP FOR 单价>1000 FIELD
品名, 单价↵

· 00002 记录被拷贝

· USE GSP↵

· DISPLAY ALL↵

00001	电冰箱	1030.96
00002	电视机	1140.00

例6. 生成一个家用电器文件JD.

· USE 商品↙

· COPY TO JD ALL FOR “电” \$品名↙

00002记录被拷贝

· USE JD↙

· DISPLAY ALL↙

00001 145 电冰箱 1030.96 北极星

00002 138 电视机 1140.00 美乐-9

注意以下几点:

①如果没有选用范围、条件,那么系统将全部记录复制到目标文件中。当然可以只选用范围或者条件,也可以两者同时选用。

②如果不选用字段表,则所有字段复制到目标文件中,即是说,新文件和活化数据库文件有完全相同的字段。

③在一个命令中,待选项彼此的先后位置顺序是无所谓的。但必须在命令动词后面。

2. 复制文件结构

如果我们并不需要数据库文件中的数据,而是欣赏其文件的结构,那么可以用复制文件结构的命令。

格式: COPY STRUCTURE TO <目标文件名>

作用: 把活化数据库文件的结构复制到目标文件中去。

例7. 把商品文件的结构复制到SSP中。

· USE 商品↙

· COPY STRUCTURE TO SSP↙

当然,如果需要的话,也可以用APPEND命令把数据添加到SSP文件中去。

3. 生成结构文件

用CREATE命令建立的数据库文件，字段名是用户根据实际表格取定的，记录也是根据表格内容输入的，这是地地道道的数据库。cdBASE中还可以生成另一种数据库文件，其字段名是固定的且只有4个字段，这4个字段分别是：字段名、类型、宽度和小数位数，而每一个记录就是一个数据库中实际的字段名、类型、宽度和小数位数这些具体的值。这种数据库文件专门称为结构数据库文件，简称结构文件。产生结构文件的命令是：

格式：COPY TO(文件名) STRUCTURE
EXTENDED

作用：用活化数据库文件的结构来生成结构数据库文件。

例8. 用商品文件的结构来生成一个结构数据库文件。

· USE 商品 ↙

· COPY TO CSP STRUCTURE EXTENDED ↙

00004记录被拷贝

· USE CSP ↙ (打开结构文件)

· DISPLAY ALL ↙ (以下是结构文件CSP的

记录)

00001	货号	C	3	
00002	品名	C	6	
00003	单价	N	7	2
00004	型号	C	6	

从这个例子可以看到：商品文件的各个字段名、类型、宽度、小数位数成了CSP文件的记录，记录的个数刚好是商品文件的字段个数。由于CSP是一种数据库文件，因而，

凡是适用于数据库文件的操作对结构文件也适用。我们可以用这种方法来修改数据库文件的结构，例如增加一些字段或减少一些字段或更改字段。

4. 结构文件转换成数据库文件结构

要把结构文件转换成数据库文件结构，可用命令：

格式：CREATE 〈文件名〉 FROM 〈结构文件名〉

作用：将结构文件的记录转换为数据库文件结构。

例9. 把结构文件CSP的记录转换成数据库文件DSP。

```
· CREATE    DSP    FROM    CSP ↵
```

这就把DSP建成一个数据库文件，只是它还没有记录而已。我们可用APPEND来添加记录。如：

```
· USE    DSP ↵
```

```
· APPEND ↵
```

```
RECORD #          00001
```

```
货号      :      :
```

```
品名      :      :
```

```
单价      :      :
```

```
型号      :      :
```

这时我们就可以按照前面介绍的方法输入数据。

从前面介绍的命令可以看出，建立和生成数据库文件有多种方式，以后还有别的方式，究竟什么命令好，要根据具体情况来确定。总的原则是：凡是能够根据已有文件生成的数据库文件，就不必再用CREATE命令去从头建立。

8.2 查看数据库文件

数据库建好以后，可以用显示命令或列表命令查看其内

容和结构。

8.2.1 显示数据库文件

前面示例中我们已经用到了DISPLAY，这就是显示命令。

1. 显示数据库记录

格式：DISPLAY [〈范围〉] [FOR 〈条件〉] [OFF]
[FIELD〈字段表〉]

作用：在屏幕上显示活化数据库文件的记录。

其中范围、条件和字段表的含义同前面讨论的相同。若选用OFF，就表示不显示记录号码。

例10. 找出价格高于500元的商品。

· USE 商品 ↙

· DISPLAY	ALL	FOR	单价 > 500 ↙
00002	145	电冰箱	1030.96 北极星
00003	138	电视机	1140.00 美乐-9
00004	127	收录机	715.74 神笛牌

每个记录前面的序号（如：00002，00003，00004）就是记录号，这是用户输入数据时由系统按输入记录的先后顺序而编定的，记录号也可以看成是记录在磁盘上的地址。对一个数据库文件来说，只要我们知道了记录号，也就能够很快找到整个记录了。

例11. 找出价格高于500元且带“电”的商品，不显示记录号码。

· USE 商品 ↙

· DISPLAY ALL FOR 单价 > 500 . AND . “电” \$

品名 OFF ↙

145	电冰箱	1030.96	北极星
138	电视机	1140.00	美乐-9

注意:

①如果命令中不选用范围或者条件,那么就只显示当前记录。

例12. 不带任何待选项的显示。

· USE 商品 ↙ (当前记录是第1个记录)
· DISPLAY ↙

00001	120	自行车	179.93	环球-2
-------	-----	-----	--------	------

②如果使用了OFF,则不显示记录号码(见例11)。

③屏幕上每次至多显示20个记录(西文 dBASE),中文 cdBASE至多显示9个记录后暂停,等待敲入任何键后又继续显示后继记录。如果记录个数太少就看不出这个特征。

④如果需要将显示的内容同时打印在宽行纸上,可以在显示命令之后跟上TO PRINT。

例13. · USE 商品 ↙
· DISPLAY ALL TO PRINT ↙

如果用了TO PRINT,事先一定要连通打印机,否则会锁住系统。以至敲任何键均不起作用。如果出现这种情况,只好关机后重新启动。

2. 显示数据库文件结构

如果只显示数据库文件结构,那么可以用显示文件结构的命令。

格式: DISPLAY STRUCTURE

作用：显示活化数据库文件的结构。

例14. 显示商品文件的结构。

· USE 商品 ✓

· DISPLAY STRUCTURE ✓ (以下是显示内容)

文件的结构：商品·DBF

记录个数： 00005

最后更改的日期：06/23/89

主区使用数据库

字段	名字	类型	宽度	小数
001	货号	C	003	
002	品名	C	006	
003	单价	N	007	002
004	型号	C	006	
合计			00023	

注意：这里，总的宽度比定义文件时多了一个字节。本来应该22 ($3+6+7+6=22$) 个字节，现在显示的却是23，多出的那个位置系统留作删除标记用（后面将要介绍删除标记*）。

8.2.2 列表数据库文件

cdBASE除了用DISPLAY显示数据库记录或结构外，还可以用LIST命令。

格式1：LIST[<范围>][FOR <条件>][FIELDS <字段表>][OFF]

作用：用卷动方式列出活化数据库文件的记录。

格式2：LIST STRUCTURE

作用：用卷动方式列出活化数据库文件的结构。

其实，LIST同DISPLAY的作用和用法是类似的，格式中除了命令动词LIST和DISPLAY以外，其它部分完全相同。它们在用法上的区别仅在于：

1. LIST 用卷动的方式不停地把所有记录显示完毕。停留在屏幕上的只是最后的几个记录。
2. LIST没有周期性的暂停。

当然，可以用敲^S键的办法使之暂停，再敲任何键又继续列表。

3. 如果既不用范围也不用条件，那么LIST显示的是数据库中全部记录。

例15. 找出价格高于500元且带“电”的商品。

· USE 商品↙

· LIST ALL FOR 单价>500 .AND.“电”\$品名↙

00002	145	电冰箱	1030.96	北极星
00003	138	电视机	1140.00	美乐-9

例16. 列出全部商品。

· USE 商品↙

· LIST↘

00001	120	自行车	179.93	环球-2
00002	145	电冰箱	1030.96	北极星
00003	138	电视机	1140.00	美乐-9
00004	127	收录机	715.74	神笛牌
00005	158	缝纫机	163.26	家佳乐

请读者自己给出LIST STRUCTURE的例子。

8.2.3 浏览数据库文件记录

DISPLAY和LIST命令最大缺点是显示或列表记录时人工无法干预，因而无法进行修改。但系统提供的浏览命令弥补了这种不足。

格式：BROWSE

作用：浏览活化数据库文件的记录并可以进行修改。

例17. 浏览商品文件的记录并将电视机的货号改为333。

· USE 商品 ↙

· BROWSE ↙ (以下是屏幕显示)

RECORD# : 00005 (当前记录号)

货号	品名	单价	型号
158	缝纫机	163.26	家佳乐

我们要修改电视机的货号，就敲^R控制键使光标跳到电视机所在的第3号记录，再用^E（左移一个字段）或^X（右移一个字段）把光标移到138处，将138改为333就行了。这时屏幕显示：

RECORD# : 00003

货号	品名--	单价---	型号--
333	电视机	1140.00	美乐-9

利用下面的光标控制键，我们可以随心所欲的浏览或修改记录。

^E——光标左移一个字段，若已到最左端，则移到上一个记录。

^X——光标右移一个字段，若已到最后字段，则移到下一个记录的开头。

↙——光标右移一个字段，功能同^X

- ∧R——光标跳到上一个记录。
- ∧C——光标跳到下一个记录。
- ∧S——光标左移一个字符位置。
- ◀ ——光标回退一个字符，功能同∧S。
- ∧D——光标右移一个字符位置。
- ∧W——将当前数据库文件存盘后回到点状态。
- ∧Q——当前数据库文件不存盘，即撤消修改回到点状态。

灵活运用这些控制键，可以在屏幕上迅速地把光标上下左右移动，以便于浏览和修改。

8.3 定位记录指针

BROWSE命令的优越性在于我们能够方便地移动光标，其实际效果相当于拨动了记录指针。但BROWSE中拨动记录指针是由敲∧R（向上拨动）和∧C（向下拨动）控制键来完成的，能否由专门的命令来控制？当然可以，这就是定位记录指针命令。

8.3.1 转到指定记录位置

利用转向命令，可以将指针定位到希望的记录上。

格式：GOTO{<记录号>/TOP/BOTTOM}

作用：将指针定位在活化数据库文件的指定记录位置上。

其中：

记录号——可以是数据库文件中的一个具体的记录号码，也可以是一个具有算术值的表达式，表达式的值就代表

记录号。如：GOTO 3表示指针定位到第3号记录上，使之成为当前记录。

TOP——表示数据库文件的头，即第1个记录。

BOTTOM——表示数据库文件的尾，即最后那个记录。

例18. 把指针定位到商品文件的末尾。

```
· USE 商品↵  
· GOTO BOTTOM↵  
· DISPLAY↵
```

00005 158 缝纫机 163.26 家佳乐

例19. 在上题基础上，把指针定位在数据库文件的开头。

```
· GO TOP↵ (GOTO可简写为GO)  
· DISPLAY↵
```

00001 120 自行车 179.93 环球-2

例20. 将第3、4两个记录显示出来。

```
· USE 商品↵  
· GO 3↵ (GO 3也可省写成3)  
· DISPLAY NEXT 2↵
```

00003 333 电视机 1140.00 美乐-9

00004 127 收录机 715.74 神笛牌

8.3.2 跳跃定位指针

GO的优点是可以转向任何记录。除此之外我们还希望能够以某个记录为准而前后移动指针。跳跃命令即可达此

目的。

格式: SKIP[[+/-]<算术表达式>]

作用: 在活化数据库文件中, 指针按当前记录的位置前进(记录号码增大的方向)或回退(记录号减小的方向)若干个记录。

其中:

+——表示前进。这里的前进是指记录指针朝记录号码增大的方向拨动。

-——表示回退, 即向记录号码减小的方向拨动指针。

<算术表达式>——表示前进或回退的记录个数。它一定是能够计算出数值的一个算术表达式。

+ (正负号)可以省略。如果不带正负号, 那么也表示前进方向。

例21 · USE 商品↵

· SKIP +2↵ (前进2个记录位置)

记录: 00003 (当前记录是第3号记录)

· DISPLAY↵ (显示当前记录)

00003 333 电视机 1140.00 美乐-9

· SKIP -2↵ (回退2个记录)

记录: 00001 (回退到第1个记录)

· DISPLAY↵

00001 120 自行车 179.93 环球-2

· SKIP↵ (前进到下一个记录)

记录: 00002

· DISPLAY↵

00002 145 电冰箱 1030.96 北极星

注意：若不带任何待选项，即只有SKIP，则表示前进到下个记录 +或-仅表示前进或回退的方向。

8.3.3 带条件的定位命令

GOTO和SKIP都是一种无条件定位。但实际应用中往往希望按照某种条件来拨动指针，这就要用到条件定位。

1. 条件定位

格式：LOCATE[<范围>]FOR<条件>

作用：在活化数据库文件中，把指针定位到满足给定条件的第1个记录位置上。

例22. 将指针指向价格高于500元的第一种商品。

· USE 商品 ↙

· LOCATE ALL FOR 单价 > 500 ↙

记录：00002

· DISPLAY ↙

00002 145 电冰箱 1030.96 北极星

注意：这个命令找到的仅是满足条件的第一个（不一定是文件的第1个记录）记录。如果需要找第2个满足条件的记录，那么必须用下面介绍的继续查找命令。如果给定范围内都没有满足条件的记录，那么系统提示“超出范围”或“没有找到”。

2. 继续条件定位

格式：CONTINUE

作用：继续查找活化数据库文件中满足LOCATE命令所给条件的下一个记录。

例23. 找出价格高于500元的第二种商品。

· USE 商品↵

· LOCATE ALL FOR 单价>500↵

记录: 00002

· DISPLAY↵

00002 145 电冰箱 1030.96 北极星

· CONTINUE↵ (继续查找)

记录: 00003

· DISPLAY↵

00003 333 电视机 1140.00 美乐-9

注意: 四个定位记录指针的命令 (GO, SKIP, LOCATE和CONTINUE) 都只能定出记录指针, 不能显示出记录值。如果需要把记录值显示出来, 还得用DISPLAY命令 (读者想想: 用LIST是否恰当, 为什么?)。

8.4 数据库文件的修改

读者可能还记得“离柜不认”的情景吧! 如果在输入数据时没有把错误排除干净, 那么只好用专门的修改命令了。对数据库文件的修改包括两个方面: 一是修改数据; 二是修改文件结构。

8.4.1 修改数据库记录

修改数据库记录的命令很多, 如: EDIT, CHANGE, BROWSE, REPLACE。它们各有本领。

1. 编辑命令

如果我们认定某个记录有错, 那么用编辑命令也许是方

便的。

格式：EDIT〈记录号〉

作用：允许用户用对话方式修改活化数据库中指定的记录数据。

例24. 将商品文件的第5个记录的价格改为169.75元。

· USE 商品↵

· EDIT 5↵ (以下是屏幕显示)

RECORD #: 00005

货号 : 158;

品名 : 缝纫机;

单价 : 163.26;

型号 : 家佳乐;

修改过程是：用^X控制键或↵将光标下移到“单价”字段，再用^D将光标右移到“163.26”中的“3”处，将“3.26”改为“9.75”，敲^W将修改数据存盘后回到点状态。

在修改过程中，可以用如下的光标控制键：

^E——光标上移一个字段，光标移到第一个字段时，则把当前记录存盘后再转到上一个记录继续编辑。

^X——光标下移一个字段，光标移到最后字段时，则把当前记录存盘后再转到下一个记录继续编辑。↵也具有同样功能。

^S——光标左移一个字符。◀具有同样功能。

^D——光标右移一个字符。

^V——允许在光标处插入字符。

^R——将当前记录存盘后转到上一个记录继续编辑。

^C——将当前记录存盘后转到下一个记录继续进行

编辑。

^G——删去光标所处的字符。

^W——将修改内容存盘后回到点状态。

^Q——撤消修改不存盘回到点状态。

2. 改变命令

EDIT不方便之处有二：一是不能指定字段，二是不能带条件。CHANGE命令恰好克服了这两个毛病。

格式：CHANGE[<范围>] FIELDS <字段表>[FOR <条件>]

作用：允许用对话的方式修改活化数据库文件的记录。

例25. 修改商品文件的货号。

· USE 商品 ↵

· CHANGE ALL FIELDS 货号 ↵

记录：00001 (第一个记录)

货号：120 (原来的货号)

CHANGE? 120 ↵ (把货号120修改成115)

TO 115 ↵ (120和115都由用户敲入)

货号：115 (修改后的货号为115)

CHANGE? ↵ (询问是否还有修改，↵停止修改)

操作过程是：系统自动显示修改字段原来的值(120)，CHANGE?...TO...的意思是把...修改成...。上例中“CHANGE? 120 ↵ TO 115 ↵”意思是“把‘120’改成‘115’”。CHANGE?意思是询问用户还有修改吗？如果有，则重复上述过程，若无则敲↵结束修改。

现在再来显示商品文件的记录，可以清楚看见货号已被修改了。

· USE 商品 ↵

· LIST ↵

00001	115	自行车	179.93	环球-2
00002	145	电冰箱	1030.96	北极星
00003	333	电视机	1140.00	美乐-9
00004	127	收录机	715.74	神笛牌
00005	158	缝纫机	169.75	家佳乐

注意：如果命令中没有使用范围和条件，那么系统依次显示所有记录，等待用户修改。如果使用了范围或条件，那么系统自动提示满足条件的记录，等待修改后又继续提示满足条件的下一个记录，直到满足条件的记录被修改完或用↵强行结束修改为止。注意：命令中必须给出被修改的字段名。

3. 浏览中修改

该命令的格式和用法请参见8.2.3。

4. 替换命令

对于大批量且又有规律的修改，例如：所有商品降价5%，再用EDIT或BROWSE来修改显然就不方便了。对于这种情况，cdBASE提供了自动替换命令。

格式：REPLACE[<范围>]<字段1>WITH<表达式1>
[, <字段2>WITH<表达式2>...][FOR<条件>]

作用：在活化数据库文件中，系统自动用给定的表达式的值来替换相应字段（WITH左边的字段）的值。

例26. 把所有商品降价5%。

· USE 商品 ↵

· REPLACE ALL 单价 WITH 单价 * 0.95 ↵

00005条记录被替换

(提示信息)

· LIST ↙

00001	115	自行车	170.93	环球-2
00002	145	电冰箱	979.41	北极星
00003	333	电视机	1083.00	美乐-9
00004	127	收录机	679.95	神笛牌
00005	158	缝纫机	161.26	家佳乐

注意：如果不用范围和条件，那么就只替换当前记录；凡是能够用一公式表示出来的情形，都可以用REPLACE来实现；表达式中可以包含字段名，也可完全没有字段名；表达式的类型必须与相应字段类型相同。

8.4.2 修改数据库文件结构

修改文件结构有三种情况：一是添加新字段；二是去掉字段；三是更改字段名。用到的命令是：

格式：MODIFY STRUCTURE

作用：允许用对话的方式来修改活化数据库文件的结构。

这个命令有一个最大的缺点是修改文件结构时要破坏数据库文件中的记录，造成数据丢失。为了克服这一问题，可以这样来解决：

①首先将待修改文件复制一个新的结构备份，这个新结构是用于进行修改的。

②用MODIFY STRUCTURE对新结构进行修改。

③将旧文件（待修改文件）的记录添加到新结构中生成新文件。再把新文件复制到旧文件。

④删去新文件。这时旧文件的结构就被修改好了且仍然保留了正确的数据。

1. 添加新字段

如果数据库文件结构中需要新增加若干个字段，鉴于单独用 MODIFY STRUCTURE 来修改文件结构要丢失数据，因而必须将下面命令联用。

格式：USE 〈旧文件〉（打开待修改的文件）

COPY TO〈新文件〉 STRUCTURE

USE 〈新文件〉

MODIFY STRUCTURE

〈允许修改新文件结构〉

APPEND FROM 〈旧文件〉（把旧文件记录添加到新文件中）

COPY TO 〈旧文件〉

DELETE FILE 〈新文件.DBF〉（删去新文件）

我们用一个例子来说明这几个命令的联合用法。

例27. 在商品文件中添加一个“备注”字段。

· USE 商品↙

· COPY STRUCTURE TO NEWF↙

（复制结构）

· USE NEWF↙

· MODIFY STRUCTURE↙（修改结构）

修改将删掉所有的数据记录 * * * * * 继续？

（Y/N）Y

	名字	类型	长度	小数
字段01	·货号	C	003	000

字段02	:品名	C	006	000:
字段03	:单价	N	007	002:
字段04	:型号	C	006	000:
字段05	:备注	C	006↵	:

修改过程是：敲入 MODIFY STRUCTURE 后，屏幕提示“修改将删掉所有的数据记录 * * * * * 继续?(Y/N)”，意思是修改要破坏数据库中数据是否继续修改操作，回答Y就继续修改操作，回答N就不修改而回到点状态。由于我们先将数据库文件保护起来了，因而敲入字母Y。这时屏幕显示出文件结构，光标处在“货号”位置。由于我们要添加字段，因而用↵或^X将光标下移到“字段05”，再敲入“备注└C└006↵”，新添加字段定义完毕，敲^W结束定义，回到点状态。在敲入新字段名、类型和宽度时，彼此之间要留至少一个空格。

紧接以上操作，将商品文件的数据添加到新文件中。

· APPEND FROM 商品↵

00005记录被添加

· COPY TO 商品↵ (用新文件的结构和内容取代旧文件)

00005记录被拷贝

· USE↵ (关闭NEW.F文件)

· DELETE FILE NEW.F.DBF↵ (删去中间文件NEW.F)

商品文件是否修改了?我们可列出结构来看看。

· USE 商品↵

· LIST STRUCTURE↵

文件的结构: 商品.DBF

记录个数: 00005

最近更新日期: 06/23/89

主区使用数据库

字段	名字	类型	宽度	小数
001	货号	C	003	
002	品名	C	006	
003	单价	N	007	002
004	型号	C	006	
005	备注	C	006	
总计			00029	

从这里可以看到, 字段“备注”加上了。当然, 我们也可以把旧文件复制到一个新文件中保存起来, 再修改旧文件(虽然失去了数据, 但已经事先由新文件保存起来了, 不必担心), 等旧文件结构修改好了以后, 再把新文件中保存的数据复制到旧文件中, 其效果完全相同。请读者自己试试。

如果添加的字段不是在最后, 而是在字段之间插入一个新字段, 例如, 要把“备注”字段插入到“单价”和“型号”之间, 那么可以将光标移到“型号”字段, 敲入N, 屏幕空出一行, 这时敲入新字段名就行了。

MODIFY STRUCTURE命令中可用的光标控制键有:

^E——光标上移一个字段。

^X——光标下移一下字段。也可用↵键。

^S——光标左移一个字符。也可用键◀。

^D——光标右移一个字符。

^G——删除光标所处字符。

^V——允许在光标处插入字符。

^R——满屏下移8个字段。

^C——满屏上移8个字段。

^T——删去光标所在行。

^N——将光标所在行下移，留出一空行，允许用户插入新字段。

^W——存盘后回到点状态。

^Q——撤消修改回到点状态。

2. 删除字段

方法同上，将光标移到待删字段，敲^T键，字段就被去掉，后继字段自动上移。请读者将添加的“备注”字段去掉。

3. 更改字段名

在应用中，有时希望更改字段名中的某些字符。方法是，将光标移到待改字段，再把光标移到待改字符，重新敲入正确字符就行了。

需要特别强调的是：修改文件结构时一定要有效的保护好数据，在修改字段名时，不能同时改变字段的位置和宽度。因为一旦把位置和宽度变更，以后复制回去的数据和字段名就错位了。

对于字段的修改，除了使用MODIFY STRUCTURE命令以外，也可以用结构文件的方式来完成。请读者自己试试。

8.5 添加记录

添加记录有三种情形：一是在已知记录之间插入一个新记录；二是在文件末尾添加若干新记录；三是把一个文件接

在另一个文件的末尾。下面分别介绍之。

1. 插入记录

格式: INSERT [BEFORE] [BLANK]

作用: 允许在活化数据库文件的当前位置插入一个新记录。

其中: BEFORE是在...之前的意思, BLANK表示空格(白)。

例28. 在商品文件的第2个记录之后插入一个新记录:

```
267 电饭锅 38.42 快熟牌
· USE 商品 ✓
· GO 2 ✓      (定位在第2个记录)
· INSERT ✓    (以下是屏幕提示)
RECORD#      00003
货号         :267:
品名         :电饭锅:
单价         :38.42:
型号         :快熟牌:
· LIST ✓
00001  115   自行车   170.93   环球-2
00002  145   电冰箱   979.41   北极星
00003  267   电饭锅   38.42    快熟牌
00004  333   电视机  1083.00   美乐-9
00005  127   收录机   679.95   神笛牌
00006  158   缝纫机   161.26   家佳乐
```

插入记录的过程是: 键入INSERT命令后, 光标处在“货号”字段的第一个字符位置, 我们按装入数据时所用的

方法，敲入所需数据就行了。输入结束后敲↵键系统自动把新记录存盘。注意：①INSERT中可用的光标控制键有：

^E——光标上移一个字段。

^X——光标下移一个字段。

^S——光标左移一个字符。也可用键◀。

^D——光标右移一个字符。

^V——允许在光标处插入字符。

^R——把当前记录存盘后回到点状态。也可用^W。

^Q——不存盘回到点状态。

②若选用BEFORE，则表示在当前记录位置上插入，原记录自动下推；否则在当前记录的下一个记录位置上插入。

③若选用BLANK，则由系统自动插入一个空白记录。

④若只有一个记录，INSERT无效。

⑤INSERT不能用于在文件末尾添加记录。

2. 添加记录

格式：APPEND [BLANK]

作用：允许用对话的方式在活化数据库文件的末尾添加若干新记录。

例29. 在商品文件末尾添加新记录。

· USE 商品↵

· APPEND↵

RECORD# 00007

货号 :149:

品名 :电热褥:

单价 :88.77:

型号 :助温牌:

这同输入数据完全一样。用APPEND添加数据非常方便。当光标处在一个新记录的第一个字段的第一个字符位置时敲↵或↵W就结束添加数据而回到点状态。

APPEND BLANK含义是自动添加一个空白记录，这在格式修改数据库文件时常用到。APPEND命令中所用控制键同INSERT相同。

3. 添加一个文件

格式：APPEND FROM<文件名>

作用：将指定文件添加到活化数据库文件的末尾。

例30. 假设采购了一些新商品，其信息保存在NEWSP文件中，现在希望将NEWSP文件的记录连接在商品文件的末尾。假设NEWSP文件中的记录如下所示：

货号	品名	单价	型号
139	电子琴	357.29	自学—4
183	钢琴	2300.00	新星—1

· USE 商品↵

· APPEND FROM NEWSP↵

00002条记录被加入

· LIST↵

00001	115	自行车	170.93	环球-2
00002	145	电冰箱	979.41	北极星
00003	267	电饭锅	38.42	快熟牌
00004	333	电视机	1083.00	美乐-9
00005	127	收录机	679.95	神笛牌

00006	158	缝纫机	161.26	家佳乐
00007	149	电热褥	88.77	助温牌
00008	139	电子琴	357.29	自学-4
00009	183	钢琴	2300.00	新星-1

由上可以看到，NEWSP文件的记录已连接到商品文件中去了。需要注意：NEWSP文件和商品文件必须有相同的结构（字段同名、同类型、同宽度）。如果结构不完全相同，那么只有同名同类字段值被连接。命令执行后，NEWSP文件的记录仍然存在。

8.6 删除数据库文件记录

数据库经过长时间使用以后，可能有些数据已不再需要了，这就可以删除之。

1. 注删除标志*

对数据库中的数据，即使暂时不用，也不宜立刻去掉。因为一经去掉就不能再恢复，到这时就追悔莫及了。因而通常采用注删除标志的办法，使之有再“生”的机会。

格式：DELETE[<范围>][FOR <条件>]

作用：对活化数据库文件中指定的记录注上删除标志

。

例31. 给商品文件中带“琴”的商品注上删除标记。

· USE 商品↙

· DELETE ALL FOR “琴” \$品名↙

00002条删除

· LIST↙

00001	115	自行车	170.93	环球-2
00002	145	电冰箱	979.41	北极星
00003	267	电饭锅	38.42	快熟牌
00004	333	电视机	1083.00	美乐-9
00005	127	收录机	679.95	神笛牌
00006	158	缝纫机	161.26	家佳乐
00007	149	电热褥	88.77	助温牌
00008	* 139	电子琴	357.29	自学-4
00009	* 183	钢琴	2300.00	新星-1

货号139和183之前的“*”就是删除标志。

注意：如果不带范围和条件，则只对当前记录作删除标志。

2. 去掉删除标志

如果我们希望恢复注有删除标志的记录，则可以用恢复命令。

格式：RECALL [(范围)][FOR (条件)]

作用：去掉活化数据库文件中记录注有的删除标志*。

例32. 去掉商品文件中记录之前注有的删除标志*。

```
· USE 商品↵
· RECALL ALL↵
00002条记录被恢复
· LIST↵
```

00001	115	自行车	170.93	环球-2
00002	145	电冰箱	979.41	北极星
00003	267	电饭锅	38.42	快熟牌
00004	333	电视机	1083.00	美乐-9

00005	127	收录机	679.95	神笛牌
00006	158	缝纫机	161.26	家佳乐
00007	149	电热褥	88.77	助温牌
00008	139	电子琴	357.29	自学-4
00009	183	钢琴	2300.00	新星-1

执行恢复命令后，删除标志不见了。注意：RECALL命令的执行并不是去掉注有删除标志的记录，仅是把*符号去掉了。

3. 压缩记录

如果真的希望把无用的记录去掉，那么可以用压缩记录的命令。

格式：PACK

作用：将活化数据库文件中注有删除标志的记录从库中清除掉。

例33. 把带“琴”的商品去掉。

· USE 商品↵

· DELETE ALL FOR “琴” \$ 品名↵

00002条删除

· PACK↵

压缩完成00007记录被拷贝

· LIST↵

00001	115	自行车	170.93	环球-2
00002	145	电冰箱	979.41	北极星
00003	267	电饭锅	38.42	快熟牌
00004	333	电视机	1083.00	美乐-9
00005	127	收录机	679.95	神笛牌

00006	158	缝纫机	161.26	家佳乐
00007	149	电热褥	88.77	助温牌

用PACK压缩掉的记录，在数据库文件中就再也找不到了。因此，使用PACK命令必须十分谨慎。

4. 删除文件

如果整个文件都要去掉，那么可以用删除文件的命令。

格式：DELETE FILE〈文件名.扩展名〉

作用：删去指定的文件。

例34. 删去NEWSP文件。

```
· DELETE FILE NEWSP.DBF ✓
```

这样，NEWSP文件就被删除了。需要注意的是：

删除的文件不能是打开的文件：文件必须带扩展名：用该命令可以删除磁盘上的任何文件。实际上，系统在删除文件时并不是一个一个地去掉记录，而是仅抹掉磁盘目录中的文件名就行了。

8.7 统计命令

在应用中，我们经常要统计记录的个数或将某一系列的值全部累加起来，就可用统计命令。

1. 计数

格式：COUNT[〈范围〉][FOR 〈条件〉][TO 〈内存变量〉]

作用：统计活化数据库文件中记录的个数。

例35. 统计第3~6个记录之间价格高于500元的商品

种数。

· USE 商品 ✓

· 3 ✓ (定位当前记录)

· COUNT NEXT 4 FOR 单价 > 500 TO W ✓

计数 = 00002

· ?W ✓ (将统计个数显示出来)

2

.

注意：COUNT命令是按行来计数的，统计的结果可以保存在内存变量中，这也是定义内存变量的一种方法。命令中如果不带条件和范围，则表示对所有记录计数（相当于用ALL）。

2. 按列求和

如果我们希望把数据库文件中某字段的所有数值累加起来，可以用求和命令。

格式：SUM (数值型字段表) [(范围)] [FOR (条件)] [TO (内存变量表)]

作用：对活化数据库文件中记录的字段按列求代数和。

例36. 假设有如下的工资表格：

姓名	基本工资	福利费	奖金	房租费	水电费	实发工资
张森林	73.46	2.53	7.00	1.89	2.13	
徐荣策	68.35	3.10	7.50	1.45	1.55	
吕速才	68.92	3.00	8.00	2.11	1.98	
王锡刚	88.87	3.56	8.88	1.48	2.60	
陈力	125.6	3.55	8.66	1.99	2.88	
总计						
平均						

要求用cdBASE来求实发工资、各项总计值和平均值并填到相应的表格位置。

分析：先按表格给出的栏目建立工资数据库文件，输入数据，实发工资部分敲↵跳过，以后用REPLACE命令来替换。最后两个记录只输一个“总计”和“平均”，其它位置敲↵（假设工资文件已按要求建好，注意字段宽度）。

下面给出求实发工资及总计和平均的命令（内存变量 S_1 、 S_2 、 S_3 、 S_4 、 S_5 、 S_6 分别用于存放所有人的基本工资、福利费、奖金、房租费、水电费、实发工资的累加和）。

· USE 工资↵

· REPLACE ALL 实发工资 WITH 基本工资
+福利费+奖金-房租费-水电费↵

00005条替换了

· GO 1↵

· SUM NEXT 5 基本工资, 福利费, 奖金 TO
 S_1 , S_2 , S_3 ↵

· GO 1↵

· SUM NEXT 5 房租费, 水电费, 实发工资 TO
 S_4 , S_5 , S_6 ↵

· GO 1↵

· COUNT NEXT 5 TO C↵

· GO BOTTOM↵

· REPLACE 基本工资 WITH S_1/C , 福利费
WITH S_2/C , 奖金 WITH S_3/C ↵

· REPLACE 房租费 WITH S_4/C , 水电费
WITH S_5/C , 实发工资 WITH S_6/C ↵

00001条替换了

· SKIP -1↙

· REPLACE 基本工资 WITH S₁, 福利费
WITH S₂, 奖金 WITH S₃↙

· REPLACE 房租费 WITH S₄, 水电费
WITH S₅, 实发工资 WITH S₆↙

00001条替换了

· LIST OFF↙

张森林	73.46	2.53	7.00	1.89	2.13	78.97
徐荣策	68.35	3.10	7.50	1.45	1.55	75.95
吕速才	68.92	3.00	8.80	2.11	1.98	76.63
王锡刚	88.87	3.56	8.88	1.48	2.60	97.23
陈力	125.67	3.55	8.66	1.99	2.88	133.01
总计	425.27	15.74	40.84	8.92	11.14	461.79
平均	85.05	3.14	8.16	1.78	2.22	92.35

请读者认真理解这个实例中用到的各个命令。在 SUM 中为什么要用 NEXT 5? 如果换成 ALL, 那么达到同样目的要作哪些修改? 请读者自己来完成。

需要强调的是: 凡是按横行求和只能用 REPLACE 命令; 凡是按列求和可以用 SUM 命令; 所有的求和都只能针对数值型字段。

8.8 两个数据库文件的连接

如果我们希望把两个有联系的数据库文件的某些行和列拼成一个新的文件, 那么可以用 cdBASE 提供的连接命令。

格式: JOIN TO <文件名> FOR <条件>
 [FIELD <字段表>]

作用: 将主区(P.)和辅区(S.)内已经打开的数据库文件按条件进行连接, 从而生成一个新的数据库文件。

例37. 假设有学生和成绩两个数据库文件, 其格式和数据如下。

学生文件

学号	姓名	性别	年龄	籍贯
8910	张正	男	16	成都
8908	李杨	女	15	北京
8903	兰兰	女	16	长春
8909	冬冬	男	14	成都

成绩文件

姓名	数学	物理	化学	语文	政治	总分
李杨	85	80	75	83	80	
冬冬	90	88	78	83	85	
张正	80	85	83	89	82	
兰兰	75	72	84	88	84	

假定两文件均建立且装入了数据。成绩文件的总分字段值输入时可敲/跳过。现在要求找出成都籍学生各自的姓

名、性别和总成绩。

· USE 成绩↙

· REPLACE ALL 总分 WITH 数学+物理+
化学+语文+政治↙

00004条记录替换了

· SELECT SECONDARY↙ (选择辅区)

· USE 成绩↙

· SELECT PRIMARY↙ (选择主区)

· USE 学生↙

· JOIN TO 学生成绩 FOR 姓名=S.姓名.AND.
籍贯="成都" FIELD姓名, 性别, S.总分↙

· USE 学生成绩↙

· LIST↙

00001 张正 男 419

00002 冬冬 男 424

JOIN命令的执行过程是这样的：

学生文件的第一个记录依次同成绩文件的每一个记录按给定的条件进行比较，如果条件成立，那么就把满足条件的两记录合起来生成一个新记录，作为新文件的第一个记录；如果条件不成立，则学生文件的第一个记录继续同成绩文件的下一记录比较，…，这样直到成绩文件的最后那个记录。在比较过程中，条件成立则生成新记录，不成立就继续比较。然后，学生文件的第二个记录又同成绩文件的每一个记录进行比较…，如此重复以上过程，直到学生文件的最后记录。这就是说，两文件的记录每两两都要比较完。如学生文件有4个记录，成绩文件也有4个记录，那么JOIN命令要比

较 $4 \times 4 = 16$ 次。由此可见，执行JOIN命令是很费时的。

JOIN命令的执行情况如图8.1所示。

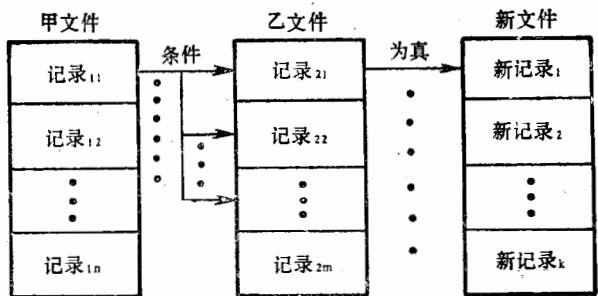


图8.1文件连接示意图

使用中需要注意：

①由于两个文件的每两个记录彼此都要进行比较，如果条件放得太松，产生的新文件记录个数有可能超过文件允许的限度。

②如果主区和辅区内使用相同的字段名，那么必须在同名之前缀以S.或P.，以示区别。通常在主区中引用辅区内的字段名，则字段名之前缀以S.（如：S.姓名）。在辅区中引用主区内的字段名则前缀以P.。

③如果命令中省略了FIELD待选项，那么新文件中的字段由主区和辅区两个文件的字段共同组成。选用原则是：首先使用主区文件的字段名，再使用辅区文件的字段名，直到最多32个字段为止。

习 题 八

1. 建立如下三个数据库文件（自己给出字段类型、宽

度和小数位数)，并装入适当的数据。

学生（学号，姓名，性别，年龄，籍贯，奖学金）

课程（课号，课名，学时数，先修课）

学生选课（姓名，课名，成绩）

2. 在上题装入数据的过程中，考察8.1.1给出的光标控制键。哪些键在你的机器上是有效的？

3. 举例说明必须选择工作区的情况。

4. CREATE和COPY命令都可用于建立数据库文件，它们在用法上有什么不同？

5. 什么是结构文件？它的实际用处是什么？

6. COPY命令的各种不同形式之间有什么区别？它们各适用于什么场合？

7. LIST和DISPLAY命令的作用是相同的，都是用于显示数据库文件的记录或结构，因而LIST命令可以取代DISPLAY，这个说法是否正确？为什么？举一个实例来说明。

8. BROWSE也可用于显示记录，它的最大优点是什么？

9. GOTO、SKIP和LOCATE都可用于定位记录指针，三者在使用法上有什么不同？

10. EDIT、CHANGE和REPLACE各适用于哪些情况？

11. 在课程文件中增加一个字段：任课教师，请写出语句（命令）。

12. INSERT和APPEND的区别是什么？APPEND和APPEND FROM〈文件名〉的区别又在哪里？

13. 任何两个数据库文件都可用APPEND FROM〈文

件名)命令连接在一起吗?试试看有什么问题?想想这是什么原因?

14. 为什么cdBASE中删除记录要采取先注标记 * 后用PACK来删去记录的办法, 而不立刻直接删去记录?

15. 在哪些情况下宜使用DELETE FILE命令?

16. 在cdBASE中, 按行求和用什么命令?按列求和用什么命令?举一个实例来说明并写出语句。

17. 在第一题建立的数据库文件中, 找出选修了数据库的学生名字以及数据库课程的先修课。请用语句写出来。

18. 如果对数据库文件希望按行来操作, 那么相应命令中可以给出哪两个待选项?按列来操作呢?

第九章 函数与运行特征

cdBASE中配备了多种函数。按照函数值的类型可分为数值函数，字符处理函数以及逻辑函数。另外，本章还要介绍为方便用户而提供的运行特征控制。

9.1 函数及其用法

cdBASE的函数可以实现通常表达式难以实现或不能实现的运算。函数本身就是一种表达式。当然，函数也可以和常数、变量一起组成更为复杂的表达式。函数经过运算以后也一定能够得出一个值。

9.1.1 数值函数

所谓数值函数就是函数运算后得出的值是一个数值，这类函数称为数值函数。

1. 取整函数INT

格式：INT (〈数值型表达式〉)

作用：取出数值型表达式值的整数部分，把小数部分全去掉。

例1. ? INT (158.723) ↵

158 (只取整数部分)

· STORE 123.748 TO X ↵

· ? INT (X) ↵

· ? INT (-5 * 6 / 4) ✓

-7

注意：INT函数不是四舍五入取整。

2. 字符串变换成数值的函数VAL

格式：VAL (<字符型表达式>)

作用：将字符型表达式值中左边的连续数字符号转换成数值。

例2. · STORE "1989.6" TO X ✓

1989.6 (字符 1989.6)

· ? VAL (X) ✓

1989.6 (数值 1989.6)

· ? VAL ("35AB43") ✓

35

· ? VAL ("123456") ✓

123456

注意：VAL函数只能转换仅由数字组成的字符串。

3. 求字符串长度函数LEN

格式：LEN (<字符型表达式>)

作用：求出字符型表达式值中包含了多少个字符。

例3. · ? LEN ("ABCDE") ✓

5

· STORE "123456" TO X ✓

· ? LEN (X) ✓

6

4. 求子串位置函数@

格式：@ (<子字符串> , <主字符串>)

作用：求出子字符串在主字符串中的开始位置。

例4. · ? @ ("ABC" , "STWABCXY") ✓

4

· ? @ ("数据库" , "cdBASE 数据库技术") ✓

7

· STORE "我们的祖国象花园" TO X ✓

· STORE "祖国" TO Y ✓

· ? @ (Y, X) ✓

7

5. 求当前记录号函数

格式：#

作用：求当前记录的号码。

例5. · USE 商品 ✓

· ? # ✓

1 (当前是1号记录)

· GOTO BOTTOM ✓

· ? # ✓

7 (当前是第7号记录)

注意：数学函数运算后得到的值均是数值，可以直接参与算术运算。

9.1.2 字符处理函数

字符处理函数的最大特点就是经过函数运算后得出的值都是字符。

1. 将数值转换成字符串的函数STR

格式：STR (<数值型表达式> , <长度> [, <小数位

数>))

作用：将数值型表达式的值按指定的长度转换成字符串。

其中：小数位数表示转换成字符串以后串中应保留的小数位。

例6. · ? STR (123.456, 7, 3) ↙

123.456 (这是字符串123.456, 相当于
“123.456”)

· STORE 5 * 8 TO W ↙

· ? STR (W * 2.21, 5, 2) ↙

88.40 (字符串)

2. 将数值变成ASCII码字符的函数CHR

格式：CHR (<数值型表达式>)

作用：将数值表达式的值转换成相应的ASCII码字符。

例7. · ? CHR (78) ↙

N (字母N的ASCII码是78)

· ? CHR (10 * 8) ↙

P (字母P的ASCII码是80)

3. 取子串函数 \$

格式：\$ (<字符型表达式>, <开始位置>, <个数>)

作用：把字符型表达式值中从<开始位置>起的指定<个数>的字符取出来，生成一个子串。

例8. · ? \$ (“ABCDEFGH” , 4, 3) ↙

DEF

· STORE “我们的祖国象花园” TO W ↙

· ? \$ (W, 7, 4) ↙

祖国

4. 日期函数DATE

格式: DATE ()

作用: 产生一个载有日期信息的字符串。

当我们启动cdBASE时, 系统要求我们按月/日/年的形式输入日期, 这个日期值就是保存在DATE () 函数中的。

例9. · ? DATE () ✓

06/23/89 (表示89年6月23日)

5. 测试数据类型函数 TYPE

格式: TYPE (<表达式 >)

作用: 测试表达式值的类型。C——字符型, N——数值型, L——逻辑型。

例10. · ? TYPE ("AB" + "CD") ✓

C (表示 "AB" + "CD" 的值 "ABCD" 是字符型)

· STORE 5*8 TO S ✓

· ? TYPE (S) ✓

N (数值型)

6. 删除末尾空格函数TRIM

格式: TRIM (<字符型表达式 >)

作用: 去掉字符型表达式末尾的空格。

例11. · ? LEN (TRIM ("ABC ")) ✓

3 (ABC之后的4个空格被去掉了)

7. 宏替换函数 &

格式: &<字符型内存变量名>

作用: 替换出内存变量的值。

例12. · STORE "商品" TO B ✓

```

· USE &B↵ (相当于直接写“商品”)
· DISPLAY↵
00001 115 自行车 170.93 环球-2
· STORE “*” TO CC↵
· ? “75&CC 80” ↵ (CC和80之间留一个
  空格)
75*80 (将CC中的“*”替换出来)
· STORE “3333” TO C↵
· ? 4444+ &C↵
7777 (数值)

```

这个示例说明：&不但能够将字符替换出来放到所在位置，而且能够将全由数字组成的字符串替换出来后转换成数值并参与算术运算。注意，&和其后的变量名之间不能留空格。如 & C是错误的。

9.1.3 逻辑函数

逻辑函数的运算结果一定是逻辑值真(用.T.表示)或假(用.F.表示)。

1. 检查文件是否结束函数EOF

格式：EOF

作用：若记录指针指向数据库文件最后那个记录之后的位置时，EOF的值为真，否则为假。

例13. USE 商品↵

```

· ? EOF↵
.F. (当前记录是1号，所以EOF的值为假)
· GO BOTTOM↵
· SKIP↵ (下推一个记录位置)

```

·? EOF ↙

.T. (文件已经结束)

EOF是End of File(文件结束)的缩写, 这个函数在对文件的循环处理中很有用。

2. 判断文件是否存在函数FILE

格式: FILE(〈字符型表达式〉)

作用: 若字符型表达式的值代表的文件已在磁盘上, 则函数的值为真, 否则为假。

例14. ·? TYPE(“商品”) ↙

.T. (表示商品文件已在磁盘上)

·? TYPE(“ABCD”) ↙

.F. (表示没有ABCD文件)

9.2 运行特征的设置方法

在执行cdBASE命令的过程中, 用户可以根据自己 的需要, 设置一些工作状态, 如: 是否响铃、是否暂停等等, 这就叫做设置运行特征。如果用户没有明确地指出运行特征, 那么系统就隐含一种特征, 这称为标准状态。

系统用于设置运行特征的方式有两种:

格式1: SET 〈参数〉 ON/OFF

作用: 将某种状态设置为开(ON)或关(OFF)的状态。

格式2: SET 〈参数1〉 TO 〈参数2〉

作用: 将参数1置为参数2。

灵活地设置运行特征, 将会给我们的应用带来极大的方便。

9.2.1 SET (参数) ON/OFF的用法

SET (参数) ON/OFF形式的运行特征共有17个, 现分别介绍之。

1. SET BELL ON/OFF——是否响铃

数据填满字段的宽度时是否响铃。ON 表示要响铃, OFF表示不响铃。如果在输入和修改数据时, 敲入的数据字符填满给定字段宽度(即两个冒号: 之间的位置)时, 希望系统用声音提醒我们, 那么在操作之前就设置SET BELL ON。系统隐含的标准特征是ON。

2. SET CARRY ON/OFF——是否将前一记录复制到下一记录

在输入数据的过程中, 往往有这种情况: 后一个记录和前一个记录的内容大部分相同, 这时不必重输后一个记录, 而用SET CARRY ON特征将前一个记录内容自动复制到后一个记录中, 再来修改不同部分。如用APPEND来添加数据, 如果记录的某些部分是相同的, 那么就用SET CARRY ON。

具体写法是:

- SET CARRY ON (置到开状态)
- USE 人事 ↙
- APPEND ↙ (自动将前一记录内容复制到下一记录中)

…… (不同的部分由人工进行修改)

系统的隐含状态是OFF(关闭)。

3. SET COLON ON/OFF——是否用冒号定界

在使用APPEND和EDIT命令时已经看到, 字段宽度是

由冒号(;)定出来的。如果我们希望取消冒号定界,那么可以在此之前设置:

· SET COLON OFF

表示APPEND和EDIT中不用冒号定界。系统隐含特征是ON(要用冒号定界)。

4. SET CONFIRM OFF/ON——光标是否自动下跳

在用APPEND、EDIT、INSERT、CREATE输入和修改数据库数据的过程中,当输入数据填满字段宽度时,如果希望敲↵键后允许光标跳到下一个字段(或记录),那么在该命令使用之前必须设置:

· SET CONFIRM ON (取消光标自动下跳)

系统的标准状态是OFF(填满字段宽度时光标自动下跳)

5. SET CONSOLE ON/OFF——是否输出到屏幕

在应用中,如果希望程序、命令、执行结果以及系统的提示信息等均不要在屏幕上显示出来,那么可以设置:

· SET CONSOLE OFF (屏幕不显示任何信息)

这通常用于键入加密口令等不允许人家看到自己的程序运行情况的时候。如果要恢复到屏幕显示,那么把OFF换成ON。系统的标准状态是ON。

6. SET DEBUG ON/OFF——是否将ECHO和STEP执行的结果送到打印机

为了分析和排除程序中的错误,有时需要将ECHO和STEP命令执行的结果传送到打印机上打印出来。可以设置:

· SET DEBUG ON

系统的标准状态是OFF。DEBUG ECHO STEP和TALK命令均用于程序的调试中。

7. SET ECHO ON/OFF——是否显示命令行

在调试程序中，如果我们希望在程序执行的同时，又在屏幕上显示出正在执行的程序，这样有利于程序的调试，那么可以在程序执行之前设置：

· SET ECHO ON

系统的标准状态是OFF（不显示）。

8. SET EJECT ON/OFF——是否换页

如果打印完一张报表后需要换到另一页上再打印另一张报表，那么可以设置换页标志：

· SET EJECT ON

系统的标准状态是ON（自动换页）。但需注意，这个命令对有些打印机可能不起作用。

9. SET ESCAPE ON/OFF——ESC 中断是否有效

在cdBASE中，不管处于任何情况，敲ESC键后系统就无条件回到点状态。如果要取消这一功能，那么设置：

· SET ESCAPE OFF

这时，敲ESC键就不能回到点状态了。系统提供的标准功能是ON。

10. SET EXACT ON/OFF——是否严格比较

所谓严格比较是指：若两个字符串比较相等，只有当两个字符串的字符和个数都对应相同时才称为相等。如：“ABC” = “ABC”是相等的，而“ABC” = “ABCDE”是不等的。

非严格比较是指：只要较短的字符串包含在较长的字符串之中且处于较长字符串的开头就行了。如：“ABC” = “ABC” 是相等的，而 “ABC” = “ABCDE” 也是相等的。

系统的标准状态是OFF（即非严格比较）。

11. SET INTENSITY ON/OFF ——是否反相显示

在使用APPEND、BROWSE和EDIT命令中，字段名除了用冒号定界外，我们看到定界的部分更加光亮，以便衬托出需要的部分，这就是反相显示（即白底黑字）。这种显示方式可以控制，当不需要反相显示时，可以设置：

· SET INTENSITY OFF （取消反相显示）

系统的标准状态是ON（需要反相显示）

12. SET LINKAGE ON/OFF——是否双区操作
如果我们希望把主区和辅区的记录联系起来，则可以设置双区联合操作。

· SET LINKAGE ON

这时当使用LIST，DISPLAY，SUM和REPORT命令时，主区和辅区两个数据库文件均起作用。而且字段个数可以放宽到64个，记录的字符个数也可以达到2000个字符。

系统的标准状态是OFF（即主区和辅区彼此独立）。

13. SET PRINT ON/OFF——是否连通打印机

如果我们希望把操作过程中显示的内容，数据、程序以及提示信息在打印机上打印出来，那么可以事先连通打印机。

· SET PRINT ON （事先要安装好打印机并打开电源）

如果中华学习机上没有安装打印机，那么就不能用这条

命令。

系统的标准状态是OFF（不接通打印机）。

14. SET RAW OFF/ON——字段间是否留空格

如果希望DISPLAY或LIST显示的数据字段彼此间不留空格，那么可以设置：

· SET RAW ON↙（字段间不留空格）

· USE 商品↙

· DISPLAY↙（字段值间无空格）

00001 115自行车170.93环球-2

· SET RAW OFF↙（字段间要留空格）

· DISPLAY↙（字段值间有空格）

00001 115 自行车 170.93 环球-2

系统的标准状态是OFF（字段值彼此间要留空格）。

15. SET SCREEN ON/OFF——是否需要全屏幕

操作

所谓全屏幕操作是指：在CREATE、APPEND、EDIT、INSERT和BROWSE命令的执行中，光标可以在屏幕中上下左右移动。如果希望关闭这种功能，那么可以设置状态为

· SET SCREEN OFF

系统的标准状态是ON（需要全屏幕功能）。

16. SET STEP ON/OFF——是否暂停

在程序调试中，有时需要执行一条命令后就暂停，以便分析程序中的错误。这时可以设置状态为：

· SET STEP ON

当系统执行一条命令后暂停，并提示用户选择后继的操作：

Y: 单步 N: 键盘命令 ESC: 退出

如果敲入字母Y, 表示继续一条一条命令的调试。

如果敲入字母N, 表示从键盘输入命令。

如果敲ESC键, 表示退出命令文件的执行。

系统的标准状态为OFF (即连续操作)

17. SET TALK ON/OFF——是否显示执行结果

如果我们不希望命令执行的结果显示在屏幕上, 就可设置状态为:

- SET TALK OFF ✓
- STORE 5*8 TO X ✓
- ? X ✓

40

系统的标准状态为ON (要显示命令执行的结果), 因而前面很多命令执行的结果都自动地显示出来了。

9.2.2 SET <参数1> TO <参数2> 的用法

1. SET FORMAT TO SCREEN/PRINT

将@命令执行的结果送到屏幕或打印机上输出。如果以后用@命令建立的报表不仅要在屏幕上显示出来, 而且还要在打印纸上打印出来的话, 那么一定要事先设置:

- SET FORMAT TO PRINT

注意: 没有配备打印机的系统不能用该命令。没有明确说明在打印机上输出时, 系统自动在屏幕上显示出来。

2. SET HEADING TO <字符串>

在输出用REPORT命令建立的报表时, 可以在报表标题之前加上一个小标题。如设置:

- SET HEADING TO “学习情况统计”

那么以后在调用报表文件打印报表时就自动在报表之前打印一个表头：学习情况统计。

3. SET MARGIN TO <n>

设置打印纸左边所留的空白边。其中 $n < 132$ 。如：

```
· SET MARGIN TO 10
```

表示打印纸左边空10个字符的位置后开始打印有效字符。

习 题 九

1. 如果要利用INT进行四舍五入取整，那么该如何办？请写出命令来。

2. @函数和\$函数有什么区别？举例说明。

3. STR和CHR有什么区别？它们各自在哪种情况下使用？

```
4. · STORE 5*8 TO X ✓
```

```
· ? &X ✓
```

40

这是否正确？为什么？

5. 请上机实习SET CARRY ON的功能。

6. 如果在APPEND中输入数据，我们希望消除光标的自动下跳功能，应该设置什么状态？写出这个命令。

7. 请想一个实例，希望字段间的显示刚好不留空格。写出命令并显示出结果。

第十章 排序和索引

在第三章实习显示命令和定位命令时，如果记录个数较多，那么用户将明显地发觉，系统查找一个记录的速度太慢了。这是由于系统使用的查找方法引起的。以前使用的方法是一种顺序查找，即系统根据查找条件，一个记录一个记录依次比较，如果记录个数很多，而满足条件的记录又恰巧在数据库文件的尾部，那么花费的时间自然就长了。为了解决这个问题，cdBASE 引入了另一种组织形式——排序和索引，使得记录的查找速度大大加快。

10.1 排序命令

所谓排序就是把数据库文件的记录按某些字段值递升或递降的次序重新排列的过程，从而生成一个有序的文件。

1. 排序命令

格式：SORT ON 〈排序字段〉 TO 〈文件名〉
〔ASCENDING/DESCENDING〕

作用：将活化数据库文件的记录按指定字段值递升或递降的次序重新排列，从而生成一个新的数据库文件。

例1. 将商品文件按单价由小到大排序。

- USE 商品 ↙
- SORT ON 单价 TO SSP ↙

排序完成

· USE SSP↵

· LIST↵

00001	267	电饭锅	38.42	快熟牌
00002	149	电热褥	88.77	助温牌
00003	158	缝纫机	161.26	家佳乐
00004	115	自行车	170.93	环球-2
00005	127	收录机	679.95	神笛牌
00006	145	电冰箱	979.41	北极星
00007	333	电视机	1083.00	美乐-9

例2. 按品名中字母降序来排列各种商品(字母降序是: Z Y X……C B A, 升序是: A B C……X Y Z)。

· USE 商品↵

· SORT ON 品名 TO SPM DESCENDING↵

排序完成

· USE SPM↵

· LIST↵

00001	115	自行车	170.93	环球-2
00002	127	收录机	679.95	神笛牌
00003	158	缝纫机	161.26	家佳乐
00004	333	电视机	1083.00	美乐-9
00005	149	电热褥	88.77	助温牌
00006	267	电饭锅	38.42	快熟牌
00007	145	电冰箱	979.41	北极星

注意:

①排序后生成的新文件名不能是排序文件本身。

例如: 我们将商品文件排序, 那么短语“TO”之后的文件名一定不能又是“商品”。排序并不影响待排序文件

(商品)。

②ASCENDING表示递升(由小到大), DESCENDING表示递降(由大到小)。如果都省略, 那么隐含按递升排序。汉字的升序或降序实际是按拼音字母顺序。

③排序后生成的新文件仍然是数据库文件。

2. 对排序新文件的操作

由于排序后生成的新文件仍然是一个数据库文件, 因而, 凡是对数据库文件能够进行的操作均可用于排序后的新文件。

10.2 索引文件及其操作

索引是不同于排序的又一种文件组织形式。其组织形式和作用类似于一本书的目录。当我们要查找一本书的内容时, 经验告诉我们: 先查看目录, 根据目录找到页码, 再根据页码找到具体的内容。从查找速度来说, 这种方式肯定是最快的。现在的问题是要先建立目录, 这就是索引命令的作用。

10.2.1 索引文件的建立

索引文件是通过INDEX命令来建立的。

格式: INDEX ON (索引字段表达式) TO (文件名)

作用: 在活化数据库文件的基础上, 按给定的索引字段值递升的次序建立一个新的有序文件。

例3. 按品名来建立一个索引文件。

· USE 商品 ↙

INDEX ON 品名 TO ISP↵

00007条记录被索引

· USE 商品 INDEX ISP↵ (打开索引文件
ISP)

· LIST↵

00002	145	电冰箱	979.41	北极星
00003	267	电饭锅	38.42	快熟牌
00007	149	电热褥	88.77	助温牌
00004	333	电视机	1083.00	美乐-9
00006	158	缝纫机	161.26	家佳乐
00005	127	收录机	679.95	神笛牌
00001	115	自行车	170.93	环球-2

例4. 按两个相同类型的字段来建立索引。现有一个销售文件，其结构和数据如下：

销售文件

记录号	姓名	品名	单价	数量	金额
00001	张森林	电冰箱	979.41	2	1958.82
00002	张森林	收录机	679.95	1	679.95
00003	徐荣策	自行车	170.93	5	854.65
00004	徐荣策	电视机	1083.00	3	3249.00
00005	吕速才	电冰箱	979.41	2	1958.82
00006	吕速才	收录机	679.95	3	2039.85

要求先按品名索引，如果品名相同，再按售货员姓名索引。

· USE 销售↵

· INDEX ON 品名+姓名 TO IXS↵

00006条记录被索引

· USE 销售 INDEX IXS↵ (打开索引文件)

· LIST↵

00005	吕速才	电冰箱	979.41	2	1958.82
00001	张森林	电冰箱	979.41	2	1958.82
00004	徐荣策	电视机	1083.00	3	3249.00
00006	吕速才	收录机	679.95	3	2039.85
00002	张森林	收录机	679.95	1	679.95
00003	徐荣策	自行车	170.93	5	854.65

例5.按两个不同类型字段来建立索引。我们仍然以销售文件为例。先按售货员姓名索引，姓名相同再按单价索引。

· USE 销售↵

· INDEX ON 姓名+STR(单价, 7, 2) TO
IXD↵

00006条记录被索引

· USE 销售 INDEX IXD↵

· LIST↵

00006	吕速才	收录机	679.95	3	2039.85
00005	吕速才	电冰箱	979.41	2	1958.82
00003	徐荣策	自行车	170.93	5	854.65
00004	徐荣策	电视机	1083.00	3	3249.00
00002	张森林	收录机	679.95	1	679.95
00001	张森林	电冰箱	979.41	2	1958.82

在具体实现索引时，系统需要建立两个区（不是前面所说的工作区）；一个是存放数据的，称为数据区（将就数据库文件来充当）；另一个是索引字段的值和记录号的一个对

照表，称为索引区。建立索引的过程就是生成索引区。

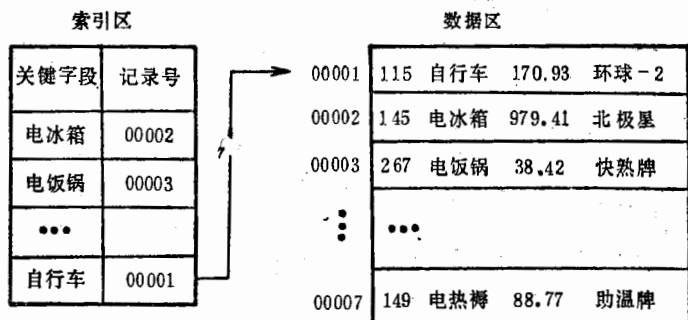


图10.1 索引示意图

当给定一个索引字段值来检索一个记录时，系统首先在索引区内查找。因为索引区只有两项，相对来说比较小，而且索引字段值是有序的，因而很快就能查到给定的索引字段值（当然该值对应的记录要存在于数据库中），按照对应的记录号，马上就可以把记录取出来，因而查找速度是相当地快。从这里也可以看出，索引区是依赖于数据库文件的，没有数据库文件，索引区内指定的记录就无法找到，因为索引区内并不实际存放数据库中的记录。

使用中需要注意以下几点：

- ①索引字段表达式可以用“+”连接起来的多个字段名；如果其中有数值型字段，则必须用STR函数转换成字符型。参见例5。
- ②最重要的索引字段必须先写。如：品名+姓名。
- ③索引总是递升的。索引文件的扩展名是·NDX。
- ④逻辑型字段不能作索引字段。
- ⑤索引字段表达式中字符个数<100个。

⑥同一数据库文件上至多可以建立7个不同的索引文件。

10.2.2 对索引文件的操作

建立索引文件的目的是为了快速地查找数据库的记录。

1. 打开索引文件

格式: USE <数据库文件名> INDEX <索引文件名字表>

作用: 打开基于数据库文件的索引文件。

其中: 索引文件名字表是基于数据库文件上建立的一个或多个索引文件。如果是多个索引文件, 彼此间要用逗号隔开。

例6. . USE 销售INDEX IXS, IXD ↙

这就打开了建立在销售文件上的IXS和IXD索引文件。排列在索引文件名字表中的第一个索引文件 (IXS) 称为主索引文件。当用USE打开索引文件时, 指针指向主索引文件的第一个记录 (索引区的第一个记录, 不一定是数据库文件的第一个记录), 随着命令的执行, 指针可以来回在主索引区上拨动。

注意:

①一个USE一次可以打开建立在同一数据库文件上的至多7个索引文件。

②若使用GOTO和SKIP命令, 那么定位的是索引的位置, 而不是数据库文件的位置。

③如果索引文件是打开的, 那么用APPEND、EDIT、REPLACE和PACK命令引起的变化将自动反映到索引中来。

2. 查找命令FIND

格式: FIND (字符序列)

作用: 查找索引文件中索引字段表达式的值同给定的字符序列相匹配的第一个记录。

例7. · USE 销售 INDEX IXD ↵

· FIND 张森林 679.95 ↵

· DISPLAY ↵

00002 张森林 收录机 679.95 1 679.95

注意:

①字符序列不能带引号。如: FIND “张森林715.74”是错误的。

②若索引字段值包含了若干个前导空格(即值的左边有若干个空格), 则字符序列也必须包含相同个数的空格且用引号将空格括起来。

③字符序列必须是索引字段表达式值的全部或开头连续的若干个字符才有效, 不能从值中间抽出部分作为字符序列。如: FIND 森林或 FIND 679 都是不正确的。而 FIND 张或 FIND 张森或 FIND 张森林是可以的。

④当字符序列是一个内存变量的值时, 必须用宏替换 &。如:

例8. · STORE “张森林” TO W ↵

· FIND & W ↵

· DISPLAY ↵

00002 张森林 收录机 679.95 1 679.95

⑤FIND只能定位满足条件的第一个记录。

10.3 对有序文件的操作

我们把排序后得到的新文件和索引文件统称为有序文件。把用于排序和索引的字段称为关键字段。对有序文件，cdBASE提供了一些专门的命令，下面介绍之。

1. 更新命令UPDATE

如果我们希望用一个文件的内容来更新另一个文件的内容时，就可用更新命令。

格式：UPDATE FROM 〈文件名〉 ON 〈关键字段〉
〔ADD〈字段表〉〕 〔REPLACE〈字段表〉〕

作用：用指定文件的内容来修改活化数据库文件的相应字段内容。

其中：关键字段用于构成修改的条件。即是说：如果两个文件（指定的文件和活化数据库文件）在关键字段上的值相等，那么就可以进行修改。ADD和REPLACE指出了修改的方式。ADD表示累加，REPLACE表示替换。

我们还是先来看一个例子。

例9. 假设库存文件和货单文件都是有序文件，它们的关键字段是品名（即按品名排序）。

库存文件

货单文件

记录号	品名	数量	记录号	品名	数量
00001	电冰箱	55	00001	电冰箱	158
00002	电饭锅	231	00002	收录机	166
00003	电热褥	233	00003	自行车	120
00004	电视机	150			
00005	缝纫机	144			

00006 收录机 23

00007 自行车 38

现在需要把新货入库，即把货单中商品的数量值累加到库存文件的同类商品数量值中。可用如下命令来实现：

- SELECT SECONDARY ↵
- USE 货单 ↵
- SELECT PRIMARY ↵
- USE 库存 ↵
- UPDATE FROM 货单 ON 品名 ADD 数量 ↵
- LIST ↵

00001 电冰箱 213

00002 电饭锅 231

00003 电热褥 233

00004 电视机 150

00005 缝纫机 144

00006 收录机 189

00007 自行车 158

通过执行UPDATE命令就把货单文件的数量值累加到库存文件的相应数量值上去了。

UPDATE的执行情况如图10.2所示：

系统执行UPDATE命令的过程是这样的：

按库存文件的记录顺序，用品名的值同货单文件的对应字段（品名）的值进行比较。如果字段值相等，就把货单文件的指定字段值累加到库存文件的数量值上，然后推到库存文件的下一个记录，重复进行比较、累加。如果比较后发现值（品名的值）不等，库存文件自动跳到下一个记录，再重复比较和累加的过程，直到库存文件记录完毕为止。

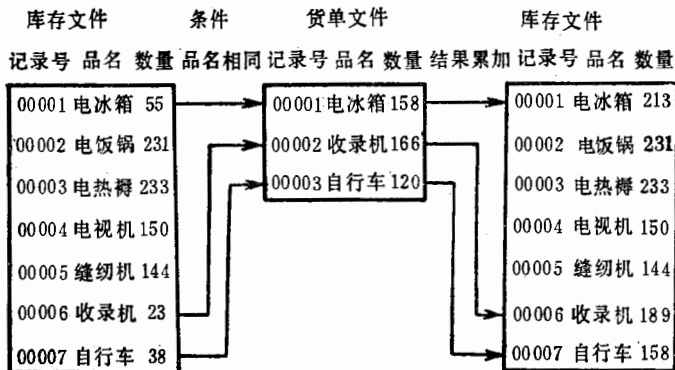


图10.2 更新文件示意图

使用时需要注意：

- ①两个文件都必须按关键字段排序或索引。
- ②只有数值型字段才能用累加（ADD）方式。
- ③当关键字段相同的记录多于一个（即关键字段值对应的记录不唯一）时，只对第一个记录进行累加或替换。
- ④如果同时累加（ADD）或替换（REPLACE）多个字段，则各字段间要用逗号隔开。

2. 合并细项TOTAL

在实际应用中，经常要进行象流水帐合计之类的工作，这可用合并细项命令来实现。

格式：TOTAL ON 〈关键字段〉 TO 〈文件名〉

 [〈范围〉] [FOR 〈条件〉] [FIELD 〈字段表〉]

作用：对活化有序文件按关键字段值分组求和。

其中：FIELD 〈字段表〉指出了需要合并细项的字段名。我们还是用例子来说明这个命令的用法。

例10. 假设我们需要按售货员的姓名来统计每个人的销售金额，实际就是将各人销售商品的金额加起来。（假定销售文件按姓名索引后为销售 I）

· USE 销售 INDEX 销售 I ↵

· TOTAL ON 姓名 TO TFILE FIELD 金额
ALL ↵

00003条记录被拷贝

· USE TFILE ↵

· COPY TO 营业额 FIELD 姓名, 金额 ↵

00003条记录被拷贝

· USE 营业额 ↵

· LIST ↵

00001 吕速才 3998.67

00002 徐荣策 4103.65

00003 张森林 2638.77

TOTAL命令的执行情况如下所示:

记录号	姓名	品名	单价	数量	金额	记录号	姓名	金额
00001	吕速才	电视机	979.41	2	1958.82	00001	吕速才	3998.67
00002	吕速才	收录机	679.95	3	2039.85			
00003	徐荣策	自行车	170.93	5	854.65	00002	徐荣策	4103.65
00004	徐荣策	电视机	1083.00	3	3249.00			
00005	张森林	电冰箱	979.41	2	1958.82	00003	张森林	2638.77
00006	张森林	收录机	679.95	1	679.95			

由此可以看到：TOTAL命令实际就是将按姓名分组的记录的金额值加起来就行了。

需要注意的是：

①如果不用范围和条件，那么就对所有记录合并细项。

②如果不用FIELD指出合并的字段，那么将对所有数值型字段合并细项。

③合并细项后得到的新文件同待合并细项的老文件有相同的结构。新文件除了合并项（例如金额字段）的值是新值以外，其它值如：姓名、品名、单价、数量均是旧文件相应分组的头个记录值。

④由于新旧文件有相同的结构，如果新文件的字段宽度（如：金额字段的宽度小于7）不足以容纳合并值，则系统在合并字段（金额）中放置星号*，作为数据溢出标志。因而，凡是要进行分组合并的数据库文件，在定义其合并字段宽度时应适当加大。

习 题 十

1.举出一个需要用到排序和索引的例子，并分别用SORT和INDEX进行排序和索引。

2.SORT和INDEX的区别是什么？

3.如果一个数据库文件要按多个字段排序，如：学生高考成绩文件，首先按总分由高到低排序，总分相同再按数学、物理、化学、语文等分数由高到低排序，请写出实现这种排序的语句。

4.USE〈数据库文件〉INDEX〈索引文件名字表〉这个命令虽然可以打开多个索引文件，但指针却只能作用于主索引文件。那么，这时打开多个索引文件还有什么意义？请举一个实例来说明，最好写出语句。

5.对索引文件能否用LOCATE来查找（定位）？对排序文件能否用FIND进行检索（定位）？

6. 对索引文件要增加或减少一些记录怎么办? 用语句来完成。

7. 请举一个例子, 它适宜于用UPDATE命令来更新且必须用到替换(REPLACE)。写出语句。

8. 分别建立销售和商品两个数据库文件并装入适当数据。文件结构是:

销售(姓名、品名、单价、数量、金额) 商品(品名、型号、产地) 写出求解下列问题的命令。

① 找出销售了张珊所售的某一种商品的其他售货员的名字。

② 找出畅销商品的名称和产地。所谓畅销商品是指销量在1000件以上且累计金额不低于20万元的商品。

第十一章 命令文件和报表格式设计

前几章介绍了cdBASE命令在对话方式下的用法。显然，学起来容易，用起来简便。缺点是速度太慢，自动化程度太低，这就直接影响到cdBASE的实际应用价值。为解决这一问题，系统又提供了程序的工作方式，或称为cdBASE的程序设计，这是通过命令文件来实现的。

数据库技术最适宜于进行数据处理。在数据处理中经常要用到各种报表，cdBASE也提供设计实用报表的能力和办法。可以说，最有用的报表格式也是通过命令文件的形式表现出来的。因此，本章我们先介绍命令文件。

11.1 命令文件

所谓命令文件就是根据实际应用要求，一定规则和顺序排列的cdBASE命令序列。它以文件的形式保存在磁盘上，以后可以用专门命令来执行。命令文件也叫做程序，以后可以不加区别地使用这两个概念。命令文件的最大好处是可以使我们的操作自动化，程序规范化。

11.1.1 命令文件的建立方式

从理论上来说，命令文件的建立方法是简单的。

格式：MODIFY COMMAND <文件名>
<命令序列>

RETURN

作用：允许用对话的方式来建立和修改一个命令文件。

其中：〈文件名〉就是命令文件的名称，它的扩展名是 .COM。命令文件建立成功以后，磁盘上保存的就是这个名字。

〈命令序列〉就是完成某种或某些操作而要用到的一个或多个命令。这些命令要按有关规则和顺序写出，一个命令占一行，每个命令之后用↵换行。

RETURN可以看成是命令文件的结束标志，在子程序中是必要的。

我们来看一个命令文件实例。

例1. 在第五章例10中，要求按售货员的姓名来统计每个人的营业额。那里是用对话方式来实现的。现在用命令文件写出就是：

· MODIFY COMMAND MCF1 (命令文件名是 MCF1)

新文件 (系统提示，表明建立新文件，以下用户敲入)

```
USE 销售↵
INDEX ON 姓名 TO 销售 I ↵
USE 销售 INDEX 销售 I ↵
TOTAL ON 姓名 TO TFILE FIELD 金额
ALL↵
USE TFILE↵
COPY TO 营业额 FIELD 姓名, 金额↵
USE 营业额↵
LIST↵
RETURN (敲^W将命令文件存盘)
```

至此，一个命令文件就建好了。建立过程是这样的：

①首先在点状态下敲入建立命令文件的标志：

MODIFY COMMAND以及文件名MCF1。

②屏幕上用“新文件”作为提示，意思是用户现在建立的是一个新的命令文件。屏幕空出，等待用户敲入各条命令。

③从上到下，从左到右依次敲入各条命令，每条命令输入完毕要敲↵键。从上面可以看到，命令之前没有出现小圆点了，这是程序和对话工作方式的差异之处。

④全部命令输入完毕，敲^W，其作用就是告诉系统，现在需要把输入的内容保存在磁盘上。敲^W存盘后又回到点状态。这时我们又可以进行其它操作。

建立命令文件时需要注意：

①一个命令占一行。如果一个命令较长，一行容纳不下，可以由系统自动换行（不用敲↵），或在适当的位置敲入一个分号；表示后继行为续行。

②每一行输完以后一定要敲↵键，表示该行结束。为简洁，以后程序行末尾我们不再给出↵，但用户输入命令时是不能少掉的。切记！

③整个命令文件输入完毕，一定要敲^W记盘。如果是大型程序，为防止机器故障而前功尽弃，最好输入一部分就敲^W记盘后再接着输入。

④经验告诉我们：用CREATE等命令来建立数据库文件的操作最好在点状态下用对话方式来完成。而对文件的检索等操作最好用命令文件的形式来完成。

11.1.2 命令文件的执行

从上例已经看到，在敲入命令后，系统并不立刻执行。

这是程序和对话工作方式的又一不同之处。命令文件建好以后，必须用专门的执行命令来运行。

1. 执行命令DO

格式：DO 〈命令文件名〉

作用：打开并执行指定的命令文件。

例2. 调用例1中建立的命令文件MCF1。

· DO MCF1 ✓ (以下是运行提示和结果)

00006条记录被索引

00001吕速才 3998.67

00002徐荣策 4103.65

00003张森林 2638.77

由于是在点状态下调用，因而运行结束后仍然回到点状态。

2. 命令文件的多重调用

由命令文件的建立和执行，我们自然会联想到：既然命令文件是由语句（语句和命令两个术语经常等同使用）序列组成的，那么这些语句序列中能否包含DO语句呢？即一个命令文件中又调用另一个命令文件。回答是肯定的。由此，引出了命令文件的多重调用问题。

有两个概念需要事先解释一下。

主程序和子程序——如果甲程序（命令文件和程序这两个概念也可以等同使用）中调用了乙程序，那么称甲程序为主程序，而乙程序称为子程序。主程序和子程序是相对的。在某种场合是子程序，在另一种场合（如乙调用丙）又可能是主程序。这如同四世同堂中的老子和儿子的关系一样。但通常的习惯却是把不被调用的程序称为主程序。凡是被其它

程序调用的程序都叫做子程序。主程序和子程序的调用关系如图11.1所示。

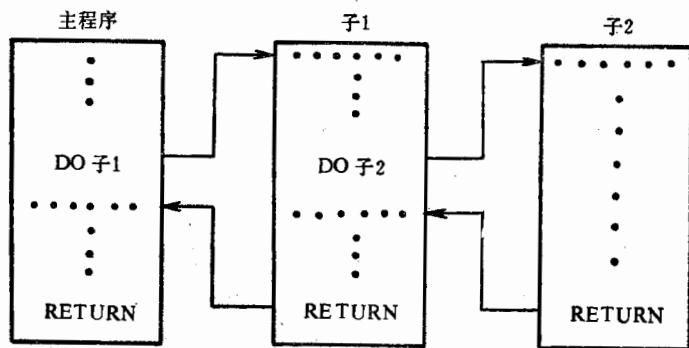


图11.1 主子程序的调用示意图

调用过程是：主程序执行到“DO 子1”时，就调用“子1”程序，相当于转到“子1”程序的第一条命令。当“子1”执行到“DO 子2”时，就转到“子2”的入口，继续执行。当遇到“子2”的“RETURN”语句时，控制返回到“子1”中“DO 子2”的下一条语句，再继续执行。当遇到“子1”中的“RETURN”时，又自动转到“主程序”中“DO 子1”的下一条命令继续执行，直到主程序结束为止。整个调用过程遵循的是“哪来哪去”的原则，即调用了某个子程序，执行完后还得回到调用处的下一个语句，这样形成一个闭合回路。

从理论上来说，这种一重一重的嵌套调用关系还可以一直延续下去。但由于具体硬件的限制，最多可以嵌套16层。一个程序中多次调用同一个子程序，当然也可以调用多个子程序，但总的调用层次不能超过16层。因为cdBASE规定同时打开的文件个数不能超过16个。

例2. 用主子程序的形式来改写例1。

- MODIFY COMMAND MCF2↙
 - DO MCF11 (调用子程序MCF11)
 - USE 营业额
 - LIST
 - RETURN (敲^W存盘)
- MODIFY COMMAND MCF11↙
 - USE 销售
 - INDEX ON 姓名 TO 销售I
 - USE 销售 INDEX 销售I
 - TOTAL ON 姓名 TO TFILE FIELD
 - 金额 ALL
 - USE TFILE
 - COPY TO 营业额 FIELD 姓名, 金额
 - RETURN (^W)
- DO MCF2↙ (执行主程序, 结果同例1)

这个例子说明主程序MCF2中调用了子程序MCF11。

例3. 我们也可以把MCF11子程序改写成主子程序的形式。

- MODIFY COMMAND MCF111↙ (子程序)
 - USE 销售
 - INDEX ON 姓名 TO 销售I
 - DO MCF112
 - USE TFILE
 - COPY TO 营业额 FIELD 姓名, 金额
 - RETURN (^W)
- MODIFY COMMAND MCF112↙ (子程

序)

USE 销售 INDEX 销售 I

TOTAL ON 姓名 TO TFILE FIELD 金額

ALL

RETURN (^W)

· MODIFY COMMAND MCF3↙ (主程序)

DO MCF111

USE 營業額

LIST

RETURN (^W)

· DO MCF3↙ (运行主程序)

执行过程是：MCF3中调用MCF111，MCF111中又调用MCF112。我们可以把调用简单地理解为将子程序搬到调用语句的位置来执行。由上可以看到，子程序也是用命令文件来实现的。子程序为编写和调试大型程序提供了有效的处理方法。在实际应用中，我们可以将一个大型问题分为若干块，每一块用一个子程序来实现。当然，子程序中还可以再套子程序。一般说来，一种相对独立的操作（可能要用多个命令才能实现）就可以作成一个子程序。然后用一个总控程序将各个子程序串起来，组成解决实际问题的一个应用程序（参见后面的应用实例程序）。

在编写和调用子程序时注意：①子程序中至少有一个RETURN（返回）语句。②除了用DO来调用程序以外，也可以在操作系统下调用，格式为：A>DBS(文件名)，这时系统在启动cdBASE后自动执行命令文件，运行完毕后回到操作系统状态。

11.1.3 命令文件的修改

命令文件的修改包括：添加、减少或更改命令以及因敲键失误而造成的字符错误。无论哪种情况，都可以用控制键将光标移到出错处，再更正之。

MODIFY COMMAND中允许使用的 光标 控制 键 有：

^E——光标上移一行。

^X——光标下移一行。↵具有相同功能。

^S——光标左移一个字符。◀有同种功能。

^D——光标右移一个字符。

^T——删去光标所在行，后面的行依次递推上移。

^N——将光标所在行及后面所有行依次下推，留出一个空白行，以便插入新行。

^G——删去光标所在字符。

^V——允许在光标左边插入字符。

^W——文件存盘后回到点状态。

^Q——文件不存盘回到点状态。

例4. 修改命令文件MCF3，要求在显示营业额之前加一个提示信息。

· MODIFY COMMAND MCF3↵

DO MCF111

USE 营业额

LIST

RETURN

现在希望在“USE营业额”和“LIST”之间插入一个新行。方法是：当我们敲入MODIFY COMMAND

MCF3↵以后，屏幕显示出原来输入的命令，光标处在第一个命令行“DO MCF111”的第一个字符位置。敲↵或↵X使光标移到“LIST”上，再敲↵N。这时屏幕上空出一行（LIST自动下推），我们就敲入添加的命令。屏幕显示如下，

```
DO MCF111
```

```
USE营业额
```

```
？“下面开始显示数据”（这是新插入的命令行）
```

```
LIST
```

```
RETURN
```

同时还可以修改发生的其它错误。敲↵W就把修改后的内容存盘（代替原文件的内容）。如果只希望在屏幕上修改而不改变磁盘原来文件的内容，那就不敲↵W而敲↵Q。↵Q的作用是撤消修改后回到点状态。

注意：命令文件的建立和修改方式相同，即建立和修改都用MODIFY COMMAND(文件名)。那么系统如何判断现在是要建立新文件呢还是要修改旧文件？如果文件名是磁盘上已经存在的命令文件名，系统就认为是要进行修改，因而在屏幕上显示出命令行可能一屏显示不完。否则就是建立新文件，因而给出空白屏幕等待用户输入程序。

11.2 专用于程序的命令

程序中除了可以用对话方式下那些命令以外，还有一些专用的命令。所谓专用是指离开了程序，这些命令就没有什么意义了。再者，个别命令的完整形式本身就是一段程序。

11.2.1 键盘输入命令

为了增强与程序的交互作用，使得在运行中可以方便地干预程序的执行，系统提供了一组键盘输入命令。

1. 等待命令WAIT

格式：WAIT [〈提示〉] [TO〈内存变量名〉]

作用：暂停程序的执行，等待从键盘上敲入任何一个字符后又继续后继命令的执行。

其中：〈提示〉是用户自己根据需要给出的。如果使用了TO待选项，那么键盘上敲入的值可以保存在内存变量中。

例5. WAIT “请输入数据：” TO W

执行该命令后屏幕提示：

请输入数据：

这时我们可以从键盘上直接敲入数据。

注意：

该命令只能接收一个字符。而且，不论我们敲入的是一个字母还是数字，都作为字符来接收。一般的dBASE不具备〔〈提示〉〕功能。

2. 接收命令ACCEPT

格式：ACCEPT [〈提示〉] TO〈内存变量〉

作用：允许从键盘上接收一串字符并送到内存变量中保存起来。

例6. ACCEPT “请输入姓名：” TO NAME

执行该命令后，屏幕提示“请输入姓名：”并等待我们从键盘上敲入姓名值。

请输入姓名：张珊↵

系统自动将键盘上敲入的“张珊”保存到内存变量NA-

ME中。

注意：ACCEPT允许接收一个或多个字符并作为字符型数据保存在内存变量中。输入字符之后一定要用↵作为结束标志；输入字符也不能带引号。

3. 输入命令INPUT

格式：INPUT [(提示)] TO(内存变量)

作用：允许从键盘上接收一个表达式的值到内存变量中。

例7. INPUT “输入数据：” TO H

执行该命令后屏幕显示：

输入数据：5+3*8↵

系统将5+3*8的值29送到H中。

注意：INPUT接收的是表达式的值而不是表达式这个字符串。表达式可以是算术、关系、逻辑或字符串表达式。

4. 读入命令READ

格式：@(<行, 列>)SAY<表达式>GET<变量名>

...

READ

作用：允许从键盘上读入一个值到变量中。

这是一种格式输入输出命令，其中的<行, 列>是指屏幕的坐标位置。SAY<表达式>表示输出，GET<变量名>表示输入。

例8. 用屏幕格式输入输出来修改商品文件的第3号记录。

• MODIFY COMMAND MCF8↵

USE 商品

GO 3

ERASE

@2, 10 SAY "修改单价:" GET 单价

@3, 10 SAY "修改型号:" GET 型号

READ

@4, 8 SAY "修改后的记录如下:"

DISPLAY

RETURN

· DO MCF8 ✓

修改单价: 34.89

修改型号: 乐天牌

修改后的记录如下:

00003 267 电饭锅 34.89 乐天牌

程序的执行是这样的: 机器在屏幕的第2行第10列显示“修改单价:”, 第3行第10列显示“修改型号:”。光标处在单价的冒号之后, 这时敲入: “34.89”以及“乐天牌”, 系统将值分别送到单价和型号字段中。这两个值是通过READ来得到的。需要注意的是:

①GET和READ必须联用。如果GET之后是内存变量名, 那么必须事先赋初值。

②利用这种命令, 我们可以构造一个直观的输出输入格式。

11.2.2 条件语句

在应用中经常需要根据条件来选择执行某些操作动作, 这就是条件语句的作用。

1. 如果语句IF

格式: IF <条件>

<命令序列1>

[ELSE

<命令序列2>]

ENDIF

作用：如果条件成立，则执行命令序列1，否则执行命令序列2。然后转到ENDIF的后继命令去执行。

例9. 如果W的值是Y那么就打开商品文件，然后显示之。否则就打开销售文件，再显示之。用IF语句来描述就是：

```
· MODIFY COMMAND MCF0  
  WAIT "请输入数据：" TO W  
  IF W = "Y"  
    USE 商品  
    LIST  
  ELSE  
    USE 销售  
    LIST  
  ENDIF  
  ? "操作结束"  
RETURN ( ^W )
```

显然，IF……ELSE……ENDIF用于两个分支选择是非常方便的。其执行流程如图11.2。

从图中看出：不论条件是否成立，后继命令总是要执行的。

2. 情形语句CASE

如果应用中要进行两个分支以上的选择，那么用DO CASE或许更为方便。

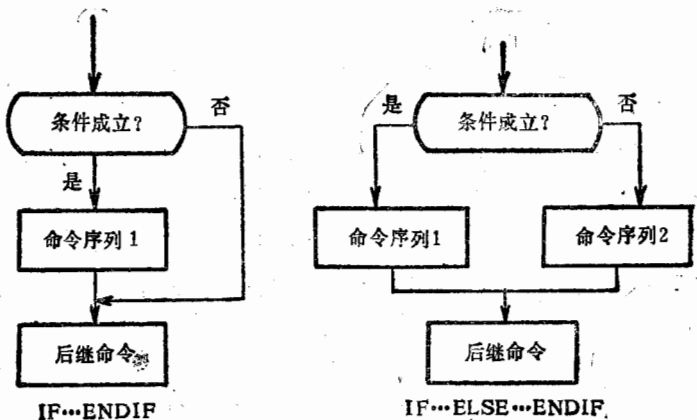


图11.2 如果语句的执行流程

格式: DO CASE
CASE<条件1>
 <命令序列1>
CASE <条件2>
 <命令序列2>
...
[OTHERWISE
 <命令序列n>]
ENDCASE

作用: 构成多分支选择。如果条件1成立, 则执行相应的命令序列1。否则, 依次检查条件, 哪一个条件成立就执行与那个条件相应的命令序列。如果给出的所有条件均不成立而又带有OTHERWISE项, 那么就执行命令序列n。所有命令序列执行后均转到ENDCASE的后继命令去执行。

例10. 假设需要在不同的情况下显示不同文件的记录, 我们可以用CASE命令来实现这种选择。

• MODIFY COMMAND MCF10 ↙

? “这是一个选择不同文件的程序。”

ACCEPT “请输入文件名:” TO FNAME

DO CASE

CASE FNAME = “商品”

USE 商品

LIST FOR “电” \$品名 FIELD 品名

CASE FNAME = “销售”

USE 销售

LIST FOR 数量 > 2 FIELD 姓名

CASE FNAME = “营业额”

USE 营业额

LIST FOR 金额 > 4000 FIELD 姓名

OTHERWISE

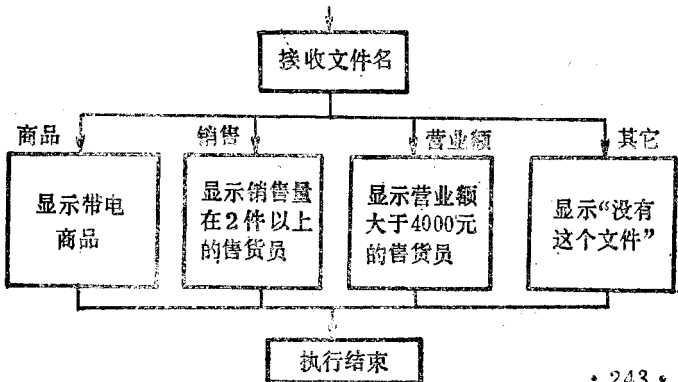
? “没有这个文件。”

ENDCASE

? “执行结束。”

RETURN (^W)

这个程序的执行如下框图所示:



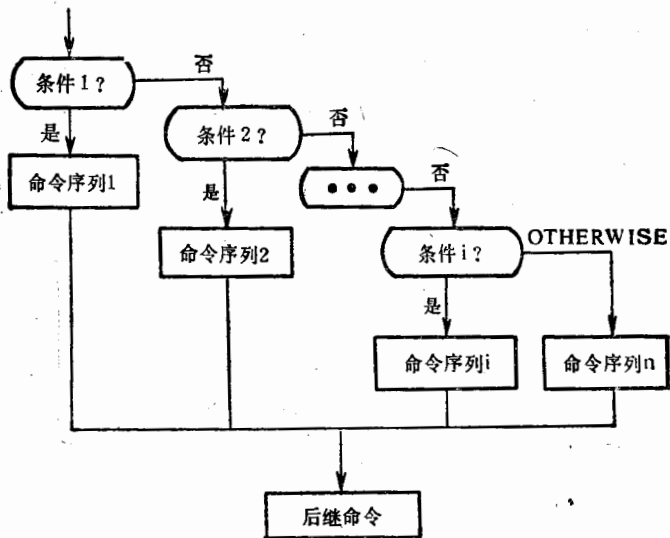


图11.3 CASE执行流程图

注意：在所有分支情形中，一旦有一个条件为真，就执行相应的命令序列。如果后面的条件也为真，系统也不会再去执行。只有前面条件不成立时才会进行下一个选择。其执行流程如图11.3所示。

11.2.3 循环

重复执行某些操作，但每次重复却有不同的内容，这种情况可以用循环恰当地描述出来。

1. 循环语句DO WHILE

格式：DO WHILE〈条件〉

〈命令序列〉

ENDDO

作用：当条件为真时，反复执行指定的命令序列，直到条件为假后就转到ENDDO的后继命令去执行。

其中：〈条件〉可以是一个关系表达式或逻辑表达式，它的值决定了是否循环。如果其值为真，则循环，否则就不循环而转到ENDDO之后。

〈命令序列〉又称为循环体，它是需要循环执行的部分。通常由一个或多个命令组成。

循环语句的执行过程是：

①计算条件的值，判断其是否为真。

②若条件值为真，执行循环体后又转第一步。

③若条件值为假，跳出循环转到ENDDO的后继命令去执行。

例11. 假设有一个工资文件，其结构如下：

工资（姓名，基本工资，奖金，福利费，房租费，水电费，实发工资）现要求把基本工资<70元的增加10元，其余增加8元。用程序来实现就是：

```
· MODIFY COMMAND MCF11 ↵  
  USE 工资  
  DO WHILE .NOT. EOF  
    IF 基本工资 < 70  
      REPLACE 基本工资 WITH 基本工  
        资+10, 实发工资 WITH 实发工资  
        +10  
    ELSE  
      REPLACE 基本工资 WITH 基本工资+  
        8 实发工资 WITH 实发工资+8  
    ENDIF
```

```

SKIP
ENDDO
LIST
RETURN (^W)
· DO MCF11 ↙

```

运行结果略去了。循环语句的执行流程如图11.4所示。

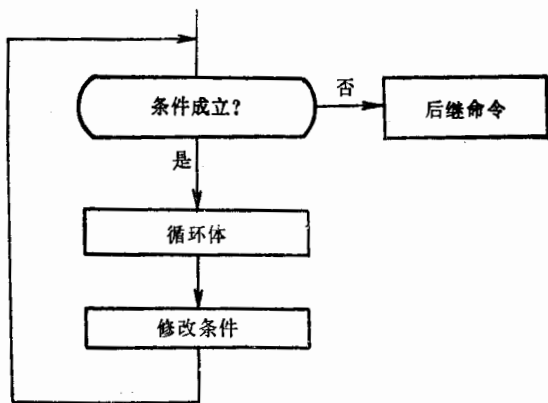


图11.4 循环语句流程示意图

2. 回路语句LOOP

格式: LOOP

作用: 无条件转到循环的开头, 准备进行下一次循环。

例12. 找出一次销售金额就超过3000元的售货员姓名、品名及其销售金额。

```

· MODIFY COMMAND MCF12 ↙
USE 销售
DO WHILE .NOT. EOF

```

IF 金额 <= 3000

SKIP

LOOP

ENDIF

DISPLAY 姓名, 品名, 金额

SKIP

ENDDO

RETURN (^W)

· DO MCF12 ✓

00004 徐荣策 电视机 3249.00

程序的执行过程是：打开销售文件，记录指针指向第一个记录。判断循环语句条件 .NOT.EOF 的值是否为真。由于记录没有结束，所以 EOF 的值为假，.NOT.EOF 表示非假即为真。因此得出循环条件 .NOT.EOF 的值为真，所以执行循环。如果金额值小于或等于 3000，那么就把记录的指针下推后执行 LOOP 而无条件进行下次循环。如果金额值大于 3000，就跳到 ENDIF 之后，显示满足要求的记录后再检查下一个记录。这样直到文件结束为止。

注意：LOOP 通常和 IF 语句联用以跳过某些语句。

3. 多重循环

如果循环语句的循环体中又包含有循环语句，这就构成所谓的多重循环。多重循环的一般形式如下：

```
DO WHILE<条件1>
...
外循环 { 内循环 { DO WHILE<条件2>
           <循环体>
           ENDDO
           ...
         }
       }
ENDDO
```

从这种嵌套结构的二重循环可以看到，外循环执行一次，内循环就要执行多次。

例13. 假设工资文件中增加了一项单位字段，表示某人是属于哪个部门的。完整结构如下：

工资（姓名，单位，基本工资，奖金，福利费，房租费，水电费，实发工资）现要求按单位统计实发工资（即将同一个单位职工实发工资累加起来）。

解决该问题的一种方法是先按单位排序文件，然后再累加同单位职工的实发工资。这个程序如下：

```
· MODIFY COMMAND MCF13 ↵  
  USE 工资  
  SORT ON 单位 TO GZS      (按单位  
                             排序)  
  USE GZS  
  STORE 单位 TO W      (第一个单位送W)  
  DO WHILE .NOT.EOF (外循环判断文件是否  
                             结束)  
    STORE 0 TO S      (S用于累加实发工  
                             资)  
    STORE W TO K      (K用于构成内循环条  
                             件)  
    DO WHILE K=单位      (相等表示是同  
                             一单位)  
      STORE S+实发工资 TO S  
      (累加实发工资)  
      DISPLAY      (显示记录)  
      SKIP      (下移一个记录)
```

IF K = 单位 (比较前后两人是否
同单位)
LOOP (相同则再转内循环)
ELSE (不相同则换单位)
STORE 单位 TO W (将下一单位保
存)

ENDIF

ENDDO

? “小计”, S (显示小计值)

ENDDO

RETURN

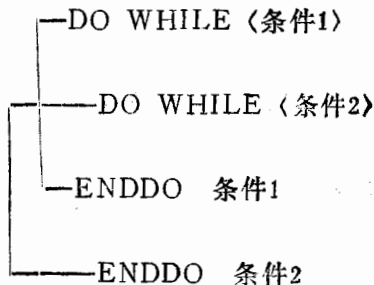
· DO MCF13 ✓

输出数据略。这里主要说明两重循环的用法。对于这个具体问题，还有更为简便的解决方法，请读者自己考虑。

对循环语句来说，要注意以下几点：

① DO WHILE 和 ENDDO 必须一一对应。

② 一个循环可以嵌套另一个循环，或两个循环并列。不允许两个循环互相交叉，如：



这是错误的循环结构。

③ 这里的循环同其它高级语言（如：BASIC，

FORTRAN等)的循环不完全相同。循环体内必须改变条件以期朝着循环结束的方向前进。

11.3 报表格式设计

在cdBASE数据库中,实现报表格式设计有两个途径。一是利用REPORT命令由对话方式来建立;二是用@...SAY...命令以程序的方式来实现。前者建立的是报表格式(.FRM)文件,后者建立的是格式(.FMT)文件。

11.3.1 报表格式文件

这种文件专用于提供输出数据的格式信息,是用对话方式来建立的。

1. 建立方式

格式: REPORT 〈文件名〉

作用: 允许用对话方式为活化数据库文件建立一个输出报表格式。

REPORT命令可以建立两种报表格式:一是不带小计的报表;另一种是带有小计的报表。下面分别用例子来说明。

例14. 对销售文件的数据,按如下的报表格式打印出来。

销售情况统计表

姓名	品名	单价	数量	金额
张森林	电冰箱	979.41	2	1958.82

.....

操作过程是:

· USE 销售↵

· REPORT↵ (以下是对话内容)

输入格式报表名: 报表1↵

输入选择, M=左页边, L=行数/页, W=页宽
5,25,35↵

页标题? (Y/N) Y↵

输入页标题: **销售情况统计表**

双空格报表? (Y/N) N↵

需要总计吗? (Y/N) N

列 长度, 内容

001 6, 姓名↵

输入报表标题: 姓名↵

002 6, 品名↵

输入报表标题: 品名↵

003 7, 单价↵

输入报表标题: 单价↵

004 3, 数量↵

输入报表标题: 数量↵

005 7, 金额↵

输入报表标题: 金额↵

006↵

各提示输入项的含义是:

输入格式报表名: ——要求用户回答报表文件的名字。
本例中敲入: 报表1。

输入选择, M=左页边, L=行数/页, M=页宽——
这是要求用户回答报表左边留多少个空格, 每一页打印多少
行, 整个报表要占多少列(即页宽)。本例中输入:

5,25,35。分别对应左边留空格5个，每页打印25行，页宽是35个字符。

页标题? (Y/N) ——回答报表每页是否需要表头。本例回答：Y

输入页标题：——敲入页标题。本例是：**销售情况统计表**

双空格报表? (Y/N) ——所谓双空格报表即打印一行空一行，询问是否需要打印一行空一行。本例回答：N，即不需双空格报表。

需要总计吗? (Y/N) ——询问报表中是否需要将某些字段的值按列累加起来。本例回答：N（不需要）。

列 长度，内容——需要打印的数据库文件的字段值。可以按照定义字段时的形式进行回答。第一字段（用001表示，系统提示）的宽度是6，“内容”位置下敲入“姓名”字段。本例敲入的是：6，姓名。

输入报表标题：——要求回答对应字段“姓名”值的栏目是什么，即打印在纸上的栏目名字。这个名字可以和字段名相同，也可以不同，由用户自己确定。本例敲入：姓名。这表示字段名和格式栏目名称相同。我们也可以敲入“售货员姓名”作为报表的栏目。

其余类似。当出现一个新的字段号码提示（如：006），我们敲入/，就表示报表格式定义完毕，系统将用户定义的报表格式信息保存在磁盘上，然后回到点状态。这就建好了一个名为“报表1”的报表格式文件。

需要强调两点：一是用户必须依次正确回答系统提问；二是“长度，内容”位置只能回答相应的字段名，而“输入报表标题：”之后可以回答任何字符。

2. 报表格式文件的调用

格式: REPORT FORM (文件名) (TO PRINT)

作用: 调用报表格式文件, 格式输出数据库内容。

例15. 调用“报表1”的格式来输出销售文件的数据。

· USE 销售 ↙

· REPORT FORM 报表1 ↙

页号: 00001

06/23/89

* * 销售情况统计表 * *

姓名	品名	单价	数量	金额
张森林	电冰箱	979.41	2	1958.82
张森林	收录机	679.95	1	679.95
徐荣策	自行车	170.93	5	854.65
徐荣策	电视机	1083.00	3	3249.00
吕速才	电冰箱	979.41	2	1958.82
吕速才	收录机	679.95	3	2039.85

3. 带小计的报表格式文件

如果我们希望数据库文件中的数据在按格式输出的同时, 还能分组小计, 那么可用带分类小计的报表格式文件。要设计这种格式, 要求数据库文件必须按分类字段排序或索引。

例16. 假设销售文件中的数据按例15的格式输出, 且还要按品名来分类小计, 即将同种商品的金额加起来。建立文件用到的语句如下:

· USE 销售 ↙

· SORT ON 品名 TO SS ↙ (按品名排序)

· USE SS ↙

REPORT ↵

输入格式报表名：报表2 ↵

输入选择， M=左页边， L=行数/页， W=页宽

5,35,38 ↵

页标题？ (Y/N) Y

输入页标题：销售小计表 ↵

双空格报表？ (Y/N) N

需要总计吗？ (Y/N) Y

报表里有小计吗？ (Y/N) Y

输入需小计的字段：品名 ↵

只需简单报表？ (Y/N) N

小计后换页？ (Y/N) N

输入小计标题：商品名称： ↵

列 长度，内容

001 6,姓名 ↵

输入报表标题：姓名 ↵

002 6,品名 ↵

输入报表标题：品名 ↵

003 7,单价 ↵

输入报表标题：单价 ↵

需要总计吗？ (Y/N) N

004 3,数量 ↵

输入报表标题：数量 ↵

需要总计吗？ (Y/N) N

005 7,金额 ↵

输入报表标题：金额 ↵

需要总计吗？ (Y/N) Y

006↵

这里有几项需要解释一下：

报表里有小计吗？（Y/N）——这实际上是询问报表要按某个字段值来分组吗？回答Y表示要分组，N表示不分组。本例回答：Y（要分组）。

输入需小计的字段：——紧接上问。意思是按哪一个字段来分组（小计）。我们可以按姓名来分组，即同一个人销售的商品打印在一起，并将其金额小计。也可以按品名来分组，即将同种商品打印在一起，且将同种商品的销售金额累加起来。分组和排序（索引）的字段要相同。本例中敲入：品名（即按品名来分类小计）。

只需简单报表？（Y/N）——所谓简单报表就是说报表中只包含分类小计项，其余细目项均不给出。简单报表又称摘要报表。本例回答：N（即分类项和细目项均需要）。

小计后换页？（Y/N）——意思是每个小计项是否就作为报表的一页。本例回答：N

输入小标题：——对每一个分组给出标志。本例给的是：商品名称。

另外，在每个数值型字段之后均要求回答：

需要总计吗？（Y/N）——表示该字段是否要小计即按分组来小计。本例要求按金额小计，因而回答：Y。

其余项目的含义与例14中相同。下面调用之。

· USR SS↵

· REPORT FORM 报表2↵（以下是输出结果）

页号. 00001

06/23/89

销售小计表

姓名	品名	单价	数量	金额
* 商品名称: 电冰箱				
张森林	电冰箱	979.41	2	1958.82
吕速才	电冰箱	979.41	2	1958.82
** 小计 **				3917.64
* 商品名称: 电视机				
徐荣策	电视机	1083.00	3	3249.00
** 小计 **				3249.00
* 商品名称: 收录机				
张森林	收录机	679.95	1	679.95
吕速才	收录机	679.95	3	2039.85
** 小计 **				2719.80
* 商品名称: 自行车				
徐荣策	自行车	170.93	5	854.65
** 小计 **				854.65
** 总计 **				10741.09

请读者结合输入和输出结果，弄清各回答项的含义。使用中注意：

① 报表格式文件的修改同建立方式一样。只需将光标移到待修改的位置，重新敲入正确的内容就行了。光标控制键请参见MODIFY COMMAND命令。

② 带分类小计的报表格式文件对应的数据库文件必须在分类字段上是有序的。或者说，只有有序文件才能建立带分类小计的报表格式文件。

③ 报表格式文件建立的只能是一种简单的报表，它既不能画横线，也不能画竖线。对嵌套栏目等复杂的表格也无能

为力。

11.3.2 格式文件的建立和调用

从上节已经看到了，报表格式文件在实际应用中有很大的局限性。为了描述更为复杂的报表，cdBASE提供了更为有力的命令@...SAY...，它可以用于建立更为复杂和实用的格式文件。

格式文件(.FMT)和报表格式文件是两种不同类型的文件。它们都可以用于建立输出数据的报表格式。格式文件中用到的主要命令是@。

格式：@〈行，列〉SAY〈表达式〉〔USING〈字型描述符〉〕

作用：从给定的坐标位置开始显示（或输出）表达式的值。

其中，〈行，列〉是屏幕或打印机的坐标位置，它可以是数值常数或数值变量。〈表达式〉可以是字段名、内存变量名、常数、算术表达式、关系表达式、逻辑表达式以及字符串表达式。常用的是字段名和字符串常数。USING〈字型描述符〉用于规定表达式值的输出格式。

建立格式文件同命令文件一样，只是文件的扩展名不同而已。

例17. 将商品文件的第一个记录用格式显示出来。

·MODIFY COMMAND 格式1.FMT↵

```
@1, 0 SAY " _____"  
@2, 0 SAY "| 货号 | 品名 | 单价 | 型号 |"  
@3, 0 SAY "| _____ |"  
@4, 0 SAY "|"
```

@4, 2 SAY 货号
 @4, 8 SAY “|”
 @4, 10 SAY 品名
 @4, 16 SAY “|”
 @4, 18 SAY 单位
 @4, 26 SAY “|”
 @4, 28 SAY 型号
 @4, 34 SAY “|”
 @5, 0 SAY “_____”

(^W)

- USE 商品 ↙
- SET FORMAT TO 格式1. FMT ↙
- DISPLAY ↙

货号	品名	单价	型号
115	自行车	170.93	环球-2

这种形式的格式文件用于屏幕显示比较方便和直观，它一反系统提供的标准格式。如果再结合GET和READ，可以构造比较满意的屏幕输入格式，这对实际应用来说是有意义的。

注意：

①格式文件中只允许两个命令，一个就是@...SAY...，另一个是注释NOTE或REMARK或*其格式和作用是：

格式：NOTE/REMARK/*〈字符序列〉

作用：对程序提供注解。〈字符序列〉就是注解的内容。

②SET FORMAT TO(格式文件名·FMT)表示打开已经建立的格式文件。

③这种@...命令最适宜于构造屏幕输入或输出格式。

④用MODIFY COMMAND建立格式文件时，文件名之后必须跟上扩展名·FMT。

11.3.3 实用报表设计

在实际应用中，往往是数据和格式交织在一起，因而更多的是用命令文件来刻划报表和装填相应的数据。下面举一个例子就清楚了。

例18. 按以下格式打印工资文件。

姓 名	收 入			支 出		实发工资
	基本工资	福利费	奖 金	房租费	水电费	
张森林	73.46	2.53	7.00	1.89	2.31	78.79
徐荣策	68.35	3.10	7.15	1.45	1.55	75.95
...

程序说明：

W——用于控制报表每页打印记录的个数。由于每个记录代表一个人的信息，因而也可以说是用于控制每页容许的人数。

A——屏幕或打印机的行号。用于确定输出信息所在的行号。

P——生成报表的页码。一份报表可能有多页，每一页

依次打印一个页码。

K——用于处理最后打印数据不满一页的情况，要求页号码也打印在每页的右下角。

L——统计每页人数是否为W。如果等于W，则打印页码，否则继续循环，这意味着本页还不满W个记录。

· MODIFY COMMAND MCF18↵

NOTE 这是报表的例子。

ERASE (清屏幕)

INPUT "输入每页记录数：" TO W

STORE W TO K

A=1

P=0

USE 工资

DO WHILE .NOT. EOF

L=1

P=P+1

@ A, 20 SAY "工资情况一览表"

A=A+1

@A, 59 SAY "单位：元"

A=A+1

DO LL (打印一条横线)

A=A+1

@ A, 0 SAY "| "

@ A, 10 SAY "| 收入"

@ A, 40 SAY "| 支出"

@ A, 60 SAY "| |"

A=A+1

```

@ A, 0 SAY " | 姓 名"
@ A, 10 SAY " | _____ "
@ A, 40 SAY " | _____ "
@ A, 60 SAY " | 实发工资 | "
A=A+1
@ A, 0 SAY " | _____ | "
@ A, 10 SAY " | 基本工资"
@ A, 21 SAY " | 福利费"
@ A, 30 SAY " | 奖 金"
@ A, 40 SAY " | 房租费"
@ A, 50 SAY " | 水电费"
@ A, 60 SAY " | _____ | "

```

A=A+1

DO LL

A=A+1

DO WHILE L<=W

```

@ A, 0 SAY " | "
@ A, 2 SAY 姓名
@ A, 10 SAY " | "
@ A, 12 SAY 基本工资
@ A, 21 SAY " | "
@ A, 22 SAY 福利费
@ A, 30 SAY " | "
@ A, 31 SAY 奖金
@ A, 40 SAY " | "
@ A, 41 SAY 房租费

```

@ A, 50 SAY “|”
 @ A, 51 SAY 水电费
 @ A, 60 SAY “|”
 @ A, 62 SAY 实发工资
 @ A, 70 SAY “|”

A=A+1

DO LL

L=L+1

A=A+1

SKIP

IF EOF

 STORE L TO K

 STORE W TO L

ENDIF

ENDDO

IF K<W

 A=A+W-K

ENDIF

A=A+2

@ A, 50 SAY “第”

@ A, 52 SAY P

@ A, 56 SAY “页”

A=A+4

ENDDO

RETURN (^W)

• MODIFY COMMAND LL ↙

NOT 打印一条横线

@ A, 0 SAY " _____ "

@ A, 24 SAY " _____ "

@ A, 45 SAY " _____ "

RETURN (^ W)

· SET FORMAT TO PRINT ↙ (连通打印机)

· DO MCF18 ↙

输入每页记录数: 3 ↙

工资情况一览表

单位: 元

姓名	收 入			支 出		实发工资
	基本工资	福利费	奖 金	房租费	水电费	
张森林	73.46	2.53	7.00	1.89	2.13	78.97
徐荣策	88.35	3.10	7.50	1.45	1.55	75.95
吕速才	88.92	3.00	8.80	2.11	1.98	76.63

第1页

工资情况一览表

单位: 元

姓 名	收 入			支 出		实发工资
	基本工资	福利费	奖 金	房租费	水电费	
王锡刚	88.87	3.56	8.88	1.48	2.60	97.23
陈 力	125.67	3.55	8.66	1.99	2.88	133.01

第2页

由此例可以看出，不管报表格式有多么复杂，只要用@〈行，列〉SAY精确地描述出其行列应该对应的位置就行了。用@命令还可以设计屏幕显示菜单。

例19. 假设一个大型程序的总控程序流程如下，



用菜单形式来提示总控程序的各种操作。这里假定“输入数据”对应PL1子程序，“修改数据”对应PL2，其余操作类推。程序如下：

```

· MODIFY COMMAND MCF19
  ERASE
  @ 1, 2 SAY " _____ "
  @ 2, 2 SAY " | 1.输入数据    2.修改数据 | "
  @ 3, 2 SAY " | 3.更新文件    4.统计数据 | "
  @ 4, 2 SAY " | 5.打印报表    0.退出运行 | "
  @ 5, 2 SAY " _____ "
  @ 6, 2 SAY "请选择代码： "
  WAIT TO DM
  DO CASE
DO DM= "1"
  DO PL1

```

```
CASE DM= "2"  
    DO PL2  
CASE DM= "3"  
    DO PL3  
CASE DM= "4"  
    DO PL4  
CASE DM= "5"  
    DO PL5  
CASE DM= "0"  
    QUIT  
ENDCASE  
RETURN    ( ^W )
```

当然，子程序中还可以用菜单，这样一层一层的提示用户，使得不懂计算机的人，也容易使用程序。

在设计实用报表时注意：

①尽量美观、大方、实用、上下竖线、栏目对齐。

②在@命令中屏幕上行列必须满足 $0 \leq \text{行号} \leq 9$ ， $0 \leq \text{列号} \leq 34$ 。在打印机上，行列可以是：0~254，但各种型号的打印机可能略有不同。

③在打印机上输出时，@命令中后面的行号应大于前面的行号，因为目前使用的打印机是一种“永向前”式打印机，打印头只能从上到下，从左到右依次打印。

④在打印机上输出报表之前，事先一定要打开打印机的电源，且在敲DO命令之前先设置打印机，命令是：

```
SET FORMAT TO PRINT
```

习 题 十 一

1. cdBASE的对话和程序设计两种工作方式之间有哪些区别？
2. 举一个能够用到多重调用的例子，并写出完整的程序。
3. 用于定义内存变量的命令至少有7个，请一一列出。
4. WAIT, ACCEPT和INPUT三者之间有什么区别？
5. @...SAY...能否用于定义内存变量？为什么？
6. 举一个最适宜于CASE命令的例子，写出完整的语句，再将CASE改用IF语句来实现。
7. DO WHILE和DO〈命令文件〉各有什么不同？
8. 如果要数据库文件中的数值型数据累加起来，至少有四种不同的办法来实现，请举一个例子，然后用四种方法分别写出来。
9. 自己给出一个带有数据的表格，再用两种不同的方法来生成报表。
10. 举一个屏幕使用格式文件的例子，写出格式文件并调用之。
11. 举一个至少有三层嵌套的报表的例子，用命令文件的形式写出报表格式。
12. 假设数据库文件结构如下：
销售（姓名，品名，单价，数量，金额）
商品（品名，型号，库存量，产地）
编写以下问题的应用程序。
①找出销量在1000件且销售的合计金额超过20万元的商

品名称及其产地。

②找出销售了张珊所售的某种商品的其他售货员的名字。

③找出所有滞销的产品（销量在10件以下的商品都叫滞销商品）。

13. 有哪些命令可以用来生成数据库？请写出这些命令并说明哪种情况下适用。

14. 哪些命令的操作必须要涉及到两个工作区？请用例子说明它们各自的用法。

第十二章 程序设计方法与技巧

本章以cdBASE程序设计为基础，介绍程序设计的基本概念、方法以及cdBASE的一些实用性技巧。

12.1 程序设计概述

12.1.1 程序和程序设计

什么是程序？有人说：程序是一个指令序列。还有人说：为解决某一问题而设计的一系列指令称为程序，等等。众说不一，但意思接近。通俗地讲：为完成一项工作或任务而安排的步骤均可称为程序。例如：食谱、乐谱都是程序。每个人每天都在履行一定的程序。什么是指令呢？从字义上讲指令可以解释为指示命令。对于食谱来说，加一匙食盐，放几粒花椒都是指令。对计算机指令可以理解为，说明计算机执行一种操作的一组符号。本章所述的程序一般是指计算机系统运行的程序，对于计算机这个专业领域来讲，告诉计算机如何做某一事情的一组指令序列称为程序。它可以用计算机指令或程序设计语言来编写。

什么是程序设计？简而言之，设计程序的过程称为程序设计。程序设计和程序的概念一样，并不是仅和计算机联系在一起。有烹调经验的人为生日家宴制作食谱，作曲家研制“乐谱”的过程都可以看成是程序设计。当然对于计算机领

域内所指的程序设计,是有其专业内涵的,本章所谈的程序设计是指正确、有效地组织计算机依次执行的每一条指令或语句。

程序设计是一项技术性技巧性都很强的工作,是模仿性和创造性的统一。所谓模仿性就是在进行程序设计时,所用的指令或语句,变量格式等都必须严格符合程序的语言所规定的语法。所谓创造性就是在符合程序设计语言所规定的语法前提下,程序设计者可以针对各自问题尽情发挥。

程序设计是逻辑思维。由于计算机本身是没有创造能力和普通生物的感觉器官的机器,必须向它提供清晰而无二义性的指令。所以它执行程序是严格按照指令的含义执行,不是对,就是错,或者说,不是真就是假。这就要求程序设计者尊重计算机的特性,采用逻辑思维的方法。

程序设计既复杂,又灵活,没有一种统一模式,程序设计人员可以尽情发挥。和写文章一样,一个对某种自然语言(如汉语)掌握得很好的人,不一定就能写出好文章。同样地,一个对计算机语句,命令学得很好的人,也不等于能设计出好的程序。对于同一个作文题目,十个人写的文章会十个样子;对同一个课题,十个人设计的程序也不尽相同,原因在于二者都是手工生产。程序设计不仅要满足人的要求而且还要符合机器的要求,由此说来,程序设计是一项极其复杂又十分灵活的工作。

但是,随着计算机科学技术的发展,计算机对于社会各个行业(包括人们的日常生活)产生的深远影响,将促使个人计算机及其被称为个人程序设计时代的到来。这就十分需要普及程序设计的基础知识和基本技能。

11.1.2 程序设计的基本技术

本节介绍程序设计的一些基本手段、原则、方法、步骤等基本知识。

1. 分解

应付复杂问题的一个重要手段是“分解”，即“分而治之”。将程序设计工作划分成几个阶段，把一个程序模块化等都体现了“分解”。分解的含义是将一个大问题分割成若干个较小的、易于解决的部分，然后分别处理。

随意机械的分解有可能使问题更加复杂化。在弄清问题内部存在的各种关系或联系机制后，科学地分解，可使问题简化且易于处理。程序设计中的分解应使各个阶段之间的联系尽量少，各个模块间的联系尽量少。

2. 抽象

程序设计的基本原则之一是抽象。抽象是应付复杂问题的又一手段，人的智力往往不可能一下子涉及到问题的全部细节。合理地采用抽象可使问题简化，易于理解。自顶向下逐层分解，上一层是下一层的抽象，将大量细节放在下一层，使上一层得到简化。抽象偏重于反映问题的本质属性，一般抽象在前，抽象阶段着重考虑“做什么”。而在实现阶段着重考虑“如何做”，把一个关系复杂、规模庞大、系统性强的问题，一步一步、一层一层地加以解决。

3. 逐步求精

逐步求精是一种适应性强并广泛采用的程序设计方法。逐步求精是建立在抽象原则的基础之上的，经过抽象产生抽象的模型、数据及程序。这些抽象的模型，数据及程序是不完善的，需要通过逐步求精的方法一步一步地完善，一直到最后程序满足应用实际的需要为止。

4. 确定性

程序设计的又一原则是确定性。确定性是保证程序质量的前提，开初对问题的一些含糊的直觉，要逐步变成明确的描述，使其确定化，例如：“程序质量好”要具体化成“易维护、可靠、高效、易理解”等；“模块质量好”具体化为“块内联系多，块间联系少”，象食谱中放一匙食盐这种不十分确定的指令描述，在计算机执行的程序中是不能接受的，也不应当存在。

5. 一致性

一致性表现是多方面的，在一个程序中，设计风格、变量或子程序等命名方式，都应保持一致性。在程序设计之初对所用文件名，变量名要有一个大概的划分，尽量有规律性、统一性，这样做可以使程序清晰，避免引起变量或文件名的重复、混乱，有利于程序的调试，文件的拷贝，列目录等工作。

统一化、标准化是程序成功的关键。讨论程序设计方法的目的之一，是使程序设计走上标准化的轨道，这样做对程序的易读、易维护等质量，及效率、生命周期等都将产生重大的影响。

6. 程序设计步骤

程序设计不仅对有关命令的功能要有深透的理解，更重要的是如何结合具体问题去组织这些命令。对于复杂的问题，程序设计必须分步进行，比较易于掌握且行之有效的是程序设计五步法。即：

第一步：审题。

主要是分析弄清要解决什么问题。有多大的数据信息量，对内存、外存、运行速度的要求如何？经费、时间、技

术、设备、人才等条件是否具备，用计算机解决本问题是否可行。

第二步：画出功能模块图。

根据要求和具体条件的可能，程序应具备哪些功能，按功能画出模块图。图12.1是一个最简单的功能模块图。

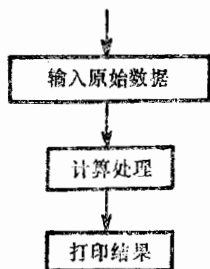


图12.1 功能模块示意图

第三步：数据结构设计。

数据的组织，直接影响到解决方法及程序设计，影响到程序质量。数据结构好，可以简化程序设计。数据的结构还会影响到存储量、处理速度、输入/输出次数、数据的冗余度、程序的可靠性等。

对于传统计算问题来讲，数据是单个的还是成组的，是数组还是文件？是随机文件还是顺序文件？等等。对于信息管理来说，数据库的数量，各个数据库的字段名，宽度，类型等等。数据结构的理论与技术比较复杂，可参看其他数据结构方面的著作。

第四步：画出程序流程图。

在分析问题、确认功能模块图、设计数据结构的基础上，画出完成各功能模块的具体流程图。这样的流程图更能

满足编程实现的需要，考虑的问题也比较深入，描绘了“如何做”的问题。

第五步：编写程序。

编写程序时，要根据第四步中的程序流程图，结合具体问题的要求，对照一些有关的实例进行。对一些与机器关系密切又无把握的地方要上机试验。最后全部通过。

以上的五步，要经反复修改，逐步求精。才能得到一个正确的程序。

12.1.3 结构程序设计简介

1. 结构程序设计

美国IBM公司的W·steven和G·myers等人提出的结构化设计(Structured Design)方法，是使用最广的程序设计方法。它运用“分解”控制程序设计工作的复杂性。“分解”就是将程序划分成一个个模块，即模块化。把一个复杂的程序设计划分成相对独立、功能单一的模块结构后，就可以单独地分工进行设计和编程，这样能够提高程序的质量并简化程序设计工作。

评价模块结构的质量标准是，块间联系最小，块内联系最大。结构程序设计的总则是使每个模块完成一个功能，模块间的传送参数尽量少。

结构程序设计首先要考虑的问题有：如何将程序划分成相对独立的模块；模块间传送一些什么信息；模块间的调用关系如何。

什么是模块？模块是指用一个名字可以调用的一段程序，它的概念类似于子程序（在cdBASE中，一个模块可以用一个命令文件来实现）。在设计的时候，一般是指这些程

序模块的原形，或称功能模块。

2. 结构图

结构设计的描述方式是结构图，它反映了程序的模块结构、块间联系以及块内联系等特性。结构图中的主要成分有：

模块——用方框表示，方框中标明反映模块功能的名字。

调用——用箭头表示模块间的调用关系，箭头的尾部模块称为调用模块，箭头指向的模块称为被调用模块。

数据——在调用箭头旁边用小箭头及字母等表示一个模块调用另一模块时的传送数据。

条件调用——在箭头的尾部用菱形表示有条件的调用。

循环调用——在箭头的尾部用弧形箭头表示循环调用。

上述符号的使用如图12.2所示。

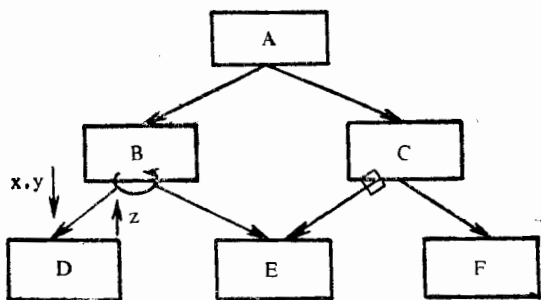


图12.2 结构图

图中模块A调用模块B和模块C；模块B循环调用模块D和E，调用模块D时传送数据X，Y，返回数据Z；模块C有条件地调用模块E，无条件调用模块F。

3. 程序设计语言的语句结构

当今的程序设计语言不胜枚举，各种机器都配上了多种语言。纵观这些流行且完善的语言，其程序的执行顺序都是由语句结构来决定的，我们在阅读他人程序或自己编程时应当有意留心结构的组成方式，最常见的语句结构如下：

① 顺序结构

顺序结构是指在程序运行时严格按语句排列的先后顺序一条一条地执行。这种结构简单明了，但不能描述某些复杂的问题。

② 判断结构

判断结构是由语句中的判断条件来决定程序执行的走向。判断结构又分为几种：①如果〈条件满足〉则〈执行语句序列〉；②如果〈条件满足〉则〈执行语句序列1〉否则〈执行语句序列2〉；③多情形判断。如：〈满足条件1〉则〈执行语句序列1〉；〈满足条件2〉则〈执行语句序列2〉；…〈满足条件n〉则〈执行语句序列n〉；〈其它情形〉则〈执行语句序列n+1〉。

③ 循环结构

循环结构就是按给定条件去重复执行一段程序。常用的是：满足〈条件〉则〈执行循环体〉，如此反复，直到不满足〈条件〉则〈执行循环的后继语句〉。循环结构能大大缩短程序的长度。

④ 调用结构

循环结构解决的是一个语句序列就地多次执行的问题，调用结构解决的是一个语句序列在不同的多个地方多次执行的问题，一般被调用的语句序列是一个独立的模块。图12.2中的B、C、D、E、F均是被调用的模块。

12.1.4 程序质量

讨论程序设计技术的目的是提高程序质量。早期，由于计算机的内存容量小，运行速度较慢，因而程序设计时特别强调程序的正确性和效率。随着计算机的普及和硬件价格的大幅度下降，加之程序规模和复杂性的增加，人们对程序的质量评价更倾向于可靠、易维护、效率、易理解等几个方面。

1. 可靠性

可靠性包括两个概念：

①正确性：指程序本身没有错误，能正确实现程序功能。

②健壮性：指在硬件发生故障或输入数据不合理时，系统仍能作出适当反映。

总而言之，可靠性是指系统在意料的情况下，能够正确地工作；在意外情况下，也能作适当处理，不会造成严重损害。

2. 易维护

通常把易阅读、易发现和纠正错误，易修改扩充归结为易维护。在设计和建立文档时都必须考虑到程序的易维护，使之易于排错或扩充程序功能。

3. 效率

效率是指程序要有效地发挥和使用计算机资源，节省时间和空间。在当今微机价格大幅度下降，不断普及且不存在机时紧张的情况下，只要不影响应用问题的解决，不必过多地去追求效率。但对那些占用机时过长，如连续运转几个小时或更长以至失去实用价值的程序，就必须千方百计提高效率，减少运行时间。对于那些可能导致容量不够的情况，应

尽量节省容量，达到在较小的资源环境下，运行较大的程序。

4. 易理解

易理解一方面是指程序内部结构清晰，易于软件人员阅读和理解；另一方面指程序的人-机界面简明清晰，使用方便。

在实际应用中，以上几个目标既有联系，又有矛盾。这就要求我们在研制经费，研制时间，可使用的资源等限制条件下，抓住主要矛盾，灵活掌握。

12.2 cdBASE应用程序设计

cdBASE是一个数据库管理系统，它包括了叫做dBASE的程序设计语言。使用dBASE程序设计语言，可以针对用户的特殊需要来开发应用程序。对于数据处理来说，用cdBASE开发程序，比用COBOL或BASIC等语言简便得多。对于计算机程序设计的初学者来说，用dBASE作为程序设计的入门，的确是一个捷径。

12.2.1 应用程序设计的步骤

1. 分析

在开始设计dBASE数据库的应用程序之前，首先应考虑下列问题：

- ①要求程序做些什么？
- ②用dBASE设计的程序能否满足需要？
- ③要求程序怎样工作？
- ④程序是否需要联机帮助？

- ⑤当对程序进行扩充和改进时，是否比较容易？
- ⑥如何建立数据库文件或其它文件？
- ⑦程序是从键盘上还是从数据库文件中获得数据？
- ⑧程序对输入和输出进行检验或修改吗？
- ⑨需要哪些报表打印和屏幕格式？

在开始设计之前，必须精确地定义出程序要做什么。从所需要的输出开始，分析输入，然后确定程序将怎样产生这些输出。例如：如果要求程序产生日、月、季和年报表，则要确定这些报表中有什么样的信息，应怎样构造并打印出这些信息。如果程序设计者不是程序的使用者，程序设计者就要与使用者一起反复商讨上述问题。这样，程序才能更好地满足实际应用的要求。

2. 设计

最常用的程序设计方法是自顶向下的结构设计方法。即从一个概括的情形开始，一步一步地工作，一直做到细节为止。在精确地定义出程序要做什么之后，用文字及图表描述整个程序设计目标，以便搞清楚它的总体结构。这时不要考虑程序的具体实现细节，重要的是确定有关的步骤。例如：要求一个程序做的事情是显示数据文件中的记录，应写出如下的程序设计概要说明。

- ①建立工作环境。
- ②用菜单提示用户该程序能够做什么。
- ③得到数据库文件名和任何有关的文件名，如索引文件名等。
- ④打开数据库文件以及有关文件。
- ⑤确定要显示的记录。
- ⑥确定要显示哪些字段。

- ⑦显示所需记录的所需字段。
- ⑧询问用户是否还要显示其它记录。
- ⑨若要，重复⑤到⑦
- ⑩若不要，关闭数据库文件并且返回。

在确定了整个程序流程之后，标识出程序要执行的基本活动，然后，逐步地把总的活动细分为越来越小的部分或模块，就象总框中的子块一样。每个模块最终应该完成一件特定的任务。对于可能性最小的问题也不能忽视。

一旦把程序的各个任务划分成了明确部分，就可以单独地为每个部分写出程序。由于每次只编写一个基本任务，所以就能很容易地写出一个程序块。把一个完整的程序设计项目划分成一个一个的逻辑上相对独立的程序设计方法称为结构程序设计，或称模块设计方法。

3. “半成品”程序

在完成了每个模块的分解之后，以文字和程序语句相结合的形式写出模块的目的。这种技术称之为伪编码。然后，在伪编码基础上写出源程序。这样作可使设计更精细，编码（写出源程序）更容易。

4. 注重调用标准模块

自顶向下和结构化程序设计技术，除了使得一个复杂的项目简化之外，还允许在任何需要的时候重复使用这些模块。在一个项目中可以在多个地方使用同一个模块，这就是软件工程中所谓的“软件重用”的新概念。这样，可以大大提高开发程序的效率。经验丰富的软件开发人员一般都积累有较多的程序模块，称之为工具箱或标准模块，在必要时拷贝下来，作一定改动即成。

5. 适当提供程序注释

源程序以及整个程序的有关文件资料是程序开发的重要内容。在源程序中提供程序段功能的注释也是十分必要的。在dBASE中用星号(*)或NOTE作为一个注释行的开始。注释有助于理解程序的逻辑,有利于测试时查找问题和错误;有利于今后的扩充、修改等维护工作。

6. 程序扩充

以模块方式设计的程序非常容易扩充。例如在cdBASE的提示“菜单”中增加选择项目,不会影响到程序的其它部分。只需在“菜单”程序中加上增加选择项目及调用语句,并增加新的子程序就可以了。

7. 多次测试

写完一个模块后,应对它进行充分的检测。这样使得出现的问题在本模块内找到原因,便于排除错误。检测时不要怕程序出错,相反,应千方百计促使程序出错,发现问题后才能解决问题。

8. 模块组装

在完成各个模块的调试之后,应逐步把各个模块在整个程序设计的意义下组装起来,进行总体调试,最后形成一个完整的程序。实际经验告诉我们:模块的组装并不轻松,即使各个模块在调试和运行中均顺利通过,组装起来后也不见得一下子就能行得通,还需要我们进一步协调。这一点,程序设计者要高度重视。

9. 程序开发步骤小结

①确定程序要做什么。

②采用自顶向下方法确定程序的整个流程和结构,把程序划分成逻辑部分或模块,再继续把这些部分一直分到每个模块完成一个特定的任务为止。画出程序设计框图。

③分别写出每个模块，先用文字，再用伪码，最后用dBASE语句。

④一边设计，一边整理设计项目和源程序的资料。

⑤单独测试每个模块。

⑥组装并对最终的程序作彻底的测试。

12.2.2 cdBASE程序及语句结构

1. 程序结构

一般来说，dBASE程序具有以下几个部分：

①程序的“前言”

程序的前言是指在程序的开头，对程序的名字、功能、作者、编辑历史及开发程序的起止时间等信息给出一个简要的说明。

②程序的运行环境设置

这部分一般跟在“前言”的后面，为程序建立操作环境。用SET命令来设置，一般是在调用程序中设置工作环境。如：SET TALK OFF等，必要时也可在相应的子程序中设置。

③程序体

程序体由完成程序功能的命令构成。诸如输入、显示、修改数据库中的信息、加工处理、输出。还有一个值得注意的问题是合理的安排中断程序及调用其它程序模块。

④关闭文件和恢复环境部分

每个程序都要做一些整理工作。例如：应保证程序在返回到点状态或操作系统之前，恢复标准特征值；为保证数据的完整性应在适当的地方关闭数据库文件等等。

2. 语句结构

①顺序结构

顺序结构是指程序运行时，按先后逐个语句执行。dBASE规定一个语句的长度不能超过254个字符，一个语句超过屏幕宽度（80个字符）时，可用分号（；）分成多行。dBASE执行时作为一个语句看待。

②判断结构

判断就是以语句中的判断条件来决定执行程序走向。

(1) 简单判断。其格式是：

```
IF<条件>  
  <语句序列>  
ENDIF
```

(2) 选择判断。其格式是：

```
IF<条件>  
  <语句序列1>  
ELSE  
  <语句序列2>  
ENDIF
```

(3) 多重选择。其格式是：

```
IF<条件1>  
  <语句序列1>  
ELSE  
  IF<条件2>  
    <语句序列2>  
  ELSE  
    IF<条件3>  
      <语句序列3>  
    ELSE
```

.....

ENDIF

ENDIF

ENDIF

(4) 结构式选择判断

DO CASE

CASE<条件1>

 <语句序列1>

 [CASE<条件2>

 <语句序列2>

.....

CASE<条件n>

 <语句序列n>]

 [OTHERWISE

 <语句序列n+1>]

ENDCASE

其中方括号中的内容表示可有可无。

③循环

循环就是按给定条件重复执行本段程序。格式是：

DO WHILE<条件>

 <语句序列1>

 [LOOP]

 <语句序列2>

ENDDO

④调用结构

其格式是：

DO <子程序名>

cdBASE的方便之处在于各个程序模块都可以根据需要进行调用。这种调用可以多至16层。这样的调用解决了多个地方多次甚至重复执行同一程序模块的问题。

12.3 cdBASE 实用技巧

本节介绍一些实用的cdBASE程序设计技巧。

12.3.1 巧用数据库文件

cdBASE数据库的设计对其程序设计的影响很大，如果能遵循数据库文件的关系模式，那么就更能体现cdBASE的优越性。对于一个应用项目，可能要建立若干个不同的，然而是有关系的数据库文件，在不同的数据库文件之间，可以用一个字段作为桥梁把它们联系起来。

1. 由源库生成目标库

由于cdBASE对于字段太多或记录数太多的数据库操作速度较低，且在断电时有可能使正在应用的数据库丢失数据或造成混乱。为了提高速度及数据安全保护的需要，在对数据库进行操作时，尽可能在目标库中进行。所谓目标库是指由原始库的数据拷贝而得到的一个新库。由cdBASE对原有的数据库产生新库十分方便。

①照源库原样产生新库

USE<源库名>

COPY TO<目标库名>

②仅复制源库的结构到目标库

USE<源库名>

COPY TO<目标库名>STRUCTURE

③仅复制源库结构中的几个字段到目标库

USE〈源库名〉

COPY TO〈目标库〉STRU FIELD〈字段名表〉

④复制部分结构连同部分数据到目标库

USE〈源库名〉

COPY TO〈目标库名〉FIELD〈字段表〉

⑤保留源库全部内容，同时需增加字段名

USE〈源库名〉

COPY TO〈目标库名〉

MODI STRU (回答Y，即消除全部数据，用键盘命令增加字段，然后用^W写盘)

APPE FROM〈目标库名〉

DELE FILE〈目标库名〉

此时的源库已增加字段名

⑥保留源库数据，并改变字段名

USE〈旧库名〉

COPY TO〈暂存外部文件名〉SDF

MODI STRU (回答Y后，修改字段，存盘)

APPE FROM〈暂存外部文件名〉·TXT SDF

DELE FILE〈暂存外部文件名〉·TXT

⑦从间接数据库产生目标库

USE〈待仿建的源库名〉

COPY TO〈间接库名〉STRU

CREA〈目标库名〉FROM〈间接库名〉

有时新的库要求：

A 原来的数据库中字段名的顺序要重新排列

B 可能要增加几个字段（用⑤的方法）

C 可能要减少几个字段（用④的方法）

对A, 可用下法:

USE OLD (例OLD. DBF中顺序为: 工号, 基本工资, 姓名……)

COPY TO TEMP STRU

USE TEMP

DISP STRU

MODI STRU (打Y, 将“基本工资”改为姓名, C, 8将“姓名”字段改为基本工资, N, 6, 2

……按^W)

APPE FROM OLD

COPY TO NEW (在NEW. DBF中, 顺序为: 工号, 姓名, 基本工资, …)

2. 裁剪数据库

为了提高效率, 根据不同的应用需要可以将数据库中的数据横向或纵向进行裁剪, 纵向裁剪就是去掉一时不用的字段, 横向裁剪就是去掉一时不用的记录。在经过裁剪之后的小数据库上操作, 可以大大提高效率。裁剪是用拷贝命令来实现的, 从源库中选出必要的字段和记录。如:

USE BRXX

COPY TO BRXX1 FIELD 疾病编号, 疗效 FOR 疾病编号>0。

这两条命令的作用是, 从一个庞大的医院住院病人信息库中, 选出供疾病分类统计用的疾病编号大于0的记录(假定疾病编号=0的住院者为产科非病人, 不属统计之列)的疾病编号和疗效两个字段。

3. 合并数据库

合并数据库也包括横向合并和纵向合并, 这可用 JOIN

命令直接进行；单独的横向合并可用APPEND命令来完成。例如：假定在某一应用项目中需要将各个下属单位软盘中的DATA. DBF文件合并成一个大库，以便在大库上加工获得整个大单位的汇总数据。对此可用如下的程序段：

```
STORE 'Y' TO HD
```

```
USE B; DATA
```

```
DO WHILE HD = 'Y'
```

```
    ? "请将下属单位软盘插入驱动器A: , 并按回车键"
```

```
    WAIT
```

```
    APPEND FROM A; DATA. DBF
```

```
    ? "继续吗? (Y/N)"
```

```
    WAIT TO HD
```

```
ENDDO
```

执行本程序段的功能是逐一录入下属单位软盘上的数据，直到回答N时，表示各下属单位的数据盘已录入完毕。实现了将各下属单位的DATA. DBF合并到整个单位B: 盘上的DATA. BDF数据库文件中。再用索引或排序及TOTAL命令可获得整个大单位的汇总数据。

12.3.2 检查键盘输入

对用户输入错误的检查，不只是在屏幕格式中，还包括在用户输入之后对程序的检查，这是程序可靠性的要求；并且还应使程序具有对用户不正确选择的处理能力。

1. 检查“菜单选择”

①假定某“菜单”具有0~9种选择，可用如下方法限制检查。

```
STORE 1 TO XZ
```

@5, 5 GET XZ PICTURE '9'

READ

如此可防止用户按字母及0~9以外的其它键。

②假定某“菜单”选择限制在1~7的范围内，则可由如下程序段实现。

STORE 0 TO XZ

DO WHILE XZ<1.OR. XZ>7

WAIT “请输入功能选择(1~7)” TO XZ

ENDDO

本程序段拒绝接收任何一个比1小或比7大的数字，如果使用者输入了不正确的数字，则要求重新输入，直接输入为1~7的数字为止。

③假定某管理软件的主控程序如下，

STORE ' ' TO N

DO WHILE .T.

ERASE

? “***×××管理系统***”

? “1. 数据输入 2. 数据修改”

? “3. 数据加工 4. 数据显示”

? “5. 报表打印 0. 退出”

@8, 9 SAY “请输入功能选择号:” GET N

READ

IF N= “0”

EXIT

ELSE

IF N<= “5”

DO PRG &N

ENDIF

ENDIF

ENDDO

本程序段比用多余CASE语句节省许多语句行。增加某些功能模块，基本上无需修改控制部分。显然，对IF N≤“5”中的5要随增加的功能模块变化，各功能的模块名都应由PRG与其序号组成。

2. 检查用户回答

在用户键入了回答之后，如何用程序检查输入的回答超出范围与否，下面是一个让用户回答（Y/N）表示是否继续的检查程序段。

```
STORE " " TO HD
```

```
DO WHILE HD# "Y" .AND. HD# "N"
```

```
  @8, 8 SAY "还继续吗? (Y/N)" GET HD  
  PICTURE "!"
```

```
  READ
```

```
ENDDO
```

本程序段不论输入大写N或Y，还是小写n或y，屏幕显示或程序接收都是大写。只要用户打入的不是Y或N，程序段请求重新输入。

3. 检查输入的表达式类型

在检索时只有输入合法的表达式才能使操作继续进行下去，这在自由的随机条件检索（即用户临时决定检索条件）情况下，更有必要。

```
STORE " " TO SR
```

```
DO WHILE TYPE(SR) = "C".OR. TYPE  
(SR) = "N"
```

```
ACCEPT "请输入检索条件" TO SR
ENDDO
```

函数TYPE产生SR中表达式的类型，在cdBASE中产生的类型是C（字符型）、N（数字型）、L（逻辑型）三种，本例中只有输入表达式的类型为C或N时才能构成循环条件。

12.3.3 宏替换的使用技巧

在cdBASE中的宏替换函数用途很广，灵活、巧妙地使用，能增加程序的灵活性、通用性、减少程序量。现将其用途归纳如下：

1. 变量代换

在管理程序中，程序长而繁，有时利用&函数可使程序精练。假设有两个数据库文件结构如下：

BA (A₁, A₂, ..., A₉), BB (B₁, B₂, ..., B₉), 且所有字段均为数值型。现要求将A₁*B₁⇒A₁, A₂*B₂⇒A₂, ..., A₉*B₉⇒A₉, 可用如下语句来实现：

```
SELECT PRIMARY
USE BA
SELECT SECONDARY
USE BB
STORE 0 TO N
DO WHILE N<=9
    N=N+1
    STORE STR(N, 1) TO X
    REPLACE ALL A &X WITH A &X * S.B &X
```

ENDDO

这比用9个REPLACE语句精简得多。

用宏替换替换内存变量的简单例子：

X=123.4

Y="X"

Z=10+&Y (Z的值133.4)

在第三条语句中，&Y替换X的值，即Z=10+123.4

2. 替换字段的内容

在程序中，有时要根据键入值确定显示当前记录某字段的内容。如：已知数据库EXA.DBF的结构由A1、A2、A3、A4共4个字段构成，则可用如下程序段显示该库当前记录某字段的内容：

ACCEPT "字段编号：" TO I

X="A"+I

? &X (显示第I(1<I<4)个字段的内容)

3. 替换字段列表

替换字段列表的方法是，将字段表(字段名间用逗号“,”隔开)以字符串形式赋给一个内存变量，然后用该变量宏替换该字段列表。仍假定字段名为A1、A2、A3、A4，其语句序列可以是：

X="A1, A2, A3"

LSIT &X (列表)

BROWSE FIELDS &X (编辑)

COPY FIELDS &X TO EX2 (生成一个新库)

4. 替换逻辑表达式

cdBASE 中有许多语句(如COPY, LOCATE, DE-

LETE等)可用一个逻辑表达式(FOR子句)指定语句的处理范围。如:

```
USE EX2
```

```
ACCEPT "输入检索条件编号:" TO I
```

```
DO CASE
```

```
  CASE I= "1"
```

```
    X= "A2>=10.AND.A3<=5"
```

```
  CASE I= "2"
```

```
    X= "A2>2.AND.A3<=10"
```

```
  CASE I= "3"
```

```
    X= "A3<5.AND.A4>0"
```

```
ENDCASE
```

```
LOCATE FOR &X
```

```
...
```

5. 替换一条语句

当一条语句在多个地方出现时,用此种方法可缩减程序长度,减少输入量。如:

```
X= "DO WHILE.NOT.EOF"
```

```
...
```

```
&X      (等价于语句: DO WHILE.NOT.EOF)
```

```
...
```

```
ENDDO
```

6. 替换一个字符序列

在字符序列中用宏替换插入内存变量(字符型)的内容:

```
X= "WANG"
```

```
Y= "HELLO &X"      (运行结果: Y=
```

“HELLO WANG”)

若宏替换的内存变量名后还有字符，则用以圆点‘.’表示内存变量名的结束。如

X = “+”

Y = “12.2 & X.4” (运行结果: Y = “12.2+4”)

总之，宏替换函数的用途非常广泛，如果运用得当，它会给程序设计带来很大方便。

12.3.4 巧建屏幕格式

cdBASE具有建立的格式文件可供多处调用的特点。

1. 建立格式文件FORM

·MODIFY COMMAND FORM.FMT

@1, 20 SAY “*****”

@2, 20 SAY “* *”

@3, 20 SAY “* DBASE 格式 *”

@4, 20 SAY “* *”

@5, 20 SAY “*****”

^W

从上面建立格式文件的过程可见，它的建立过程和命令文件的建立过程类同，区别有两点：第一点，格式文件的文件名后面要加上扩展名“·FMT”。第二点，格式文件的结束处不含“RETURN”命令。

2. 格式文件的调用

调用格式文件的两个命令是：

SET FORMAT TO <文件名>

READ

当命令文件执行到第一个命令时，就把格式文件取出

来，当第二命令即 READ 执行时，则执行格式文件中的“@”命令。

假定命令文件FILE1.CMD的内容如下。

```
·MODIF COMMAND FILE1
  SET SCREEN ON
  SET FORMAT TO FORM
  READ
  STORE 1 TO NO
  DO WHILE NO<200
  STORE NO+1 TO NO
  LOOP
  ENDDO
  ERASE
  SET FORMAT TO FORM
  READ
RETURN
```

在这个命令文件中，两次调用格式文件FORM。

12.3.5 构造数组

cdBASE不具有数组功能，使用宏代换可模拟数组功能。

下面是一个数组运算的程序，设数组A(2, 3)，数组(B3, 4)，其值为

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \\ 3 & 5 & 7 & 9 \end{pmatrix}$$

求 $C = A * B$

```
·MODIFY COMMAND SZ
```

```

STORE "1" TO I
DO WHILE &I<=2
    STORE "1" TO J
    DO WHILE &J<=3
        INPUT A &I &J
        STORE STR ( &J+1, 1 ) TO J
    ENDDO
    STORE STR ( &I+1, 1 ) TO I
ENDDO
STORE '1' TO I
DO WHILE &I<=3
    STORE '1' TO J
    DO WHILE &J<=4
        INPUT B &I &J
        STORE STR ( &J+1, 1 ) TO J
    ENDDO
    STORE STR ( &I+1, 1 ) TO I
ENDDO
STORE '1' TO I
DO WHILE &I<=2
    STOR '1' TO K
    DO WHILE &K<=4
        STORE 0 TO C &I &K
        STORE '1' TO J
        DO WHILE &J<=3
            STORE C &I &K + A &I &J * B &J &K
            TO C &I &K

```

```

        STORE STR ( &J+1, 1 ) TO J
    ENDDO
    STORE STR ( &K+1, 1 ) TO K
    ENDDO
    STORE STR ( &I+1, 1 ) TO I
    ENDDO
    LIST MEMORY
    RETURN

```

本程序，第一个二重循环，构造数组A及其数据；第二个二重循环，构造数组B及其数据。这些数据是由键盘逐一输入的。第三个三重循环求出数组C各元素的值。然后用LIST MEMORY将数组C显示出来。

12.3.6 设计口令

为了保密，或防止一些数据被误操作所破坏，在修改数据或程序启动之前可以设计口令。

1. 把口令字符串直接安排在程序中

```

·MODIFY COMMAND KL1
    ERASE
    SET TALK OFF
    STORE " " TO KL
    ? "请输入口令!"
    SET CONSOLE OFF
    ACCEPT TO KL
    SET CONSOLE ON
    IF KL<>"BMKL" .AND. KL<>"bmk1"
    ? '口令有误'

```

```
QUIT
ENDIF
RETURN
```

该程序段很简单，一旦输入的口令不是BMKL，也不是bmkl，程序控制退出dBASE。

2. 把口令放在内存变量文件中
程序如下。

①建立和修改口令的程序段

```
·MODIFY COMMAND KL.,
ACCEPT "请输入口令" TO LOCK
STORE (LOCK) TO LOCK
SAVE TO XX.MEM
RETURN
```

②审核口令的程序段

```
·MODIFY COMMAND KL.,
SET TALK OFF
RESTOR FROM XX.MEM
ERASE
? "请输入口令!"
SET CONSOLE OFF
ACCEPT TO MKL
SET CONSOLE ON
IF LOCK < > (MKL)
? "口令有误"
CANCEL
ENDIF
RETURN
```

本程序审核口令。如果口令有误，则中止程序运行；如发现失密，可以随时调用建立口令的程序段，设立新的口令。

3. 允许输入一定次数的有误口令

假定仍由2中的程序段建立口令，审核口令的程序段可设计如下：

```
·MODIFY COMMAND KL。
  RESTORE FROM XX.MEM
  SET TALK OFF
  STORE 0 TO M0
  DO WHILE M0<5
    REMARK 请键入口令！
    SET CONSOTE OFF
    ACCEPT TO M1
    SET CONSOLE ON
    IF ! (M1) = LOCK
      STORE 5 TO M0
    ELSE
      ? "口令有误"
      STORE M0+1 TO M0
    ENDIF
  ENDDO
  IF ! (M1) < > LOCK
    ? "全是有误口令，停止运行"
  CANCEL
  ENDIF
RETURN
```

本程序段允许使用者输入5次口令，若其中有一次口令正确，则往下执行程序；若5次口令都不正确，则停止程序运行。

以上介绍的方法可以在不同的程序中使用，可以根据不同密级设置多重关口。还可以采用多个口令调用同一数据库的方法，各用各的口令，把使用次数记录下来，以备检查。还可以必须两人以上，各用各的口令，口令都对上，才能进入。还可以将要保密的资料经过某种变换后保存，把变换方法的程序存入软盘，在使用数据前，必须要软盘上的有关程序变换转来才能使用。总之掌握基本方法后，就可以根据需要进行设计。

12.3.7 建立专用词库

在应用程序中，有时需要多处和多次使用某些词和短语，把这些词和短语组织起来，建成专门的词库。这样可以简化输入输出，减少用户运行程序时因输入不正确造成的出错机会，简化程序操作。

假定某单位有40多个下属单位，为方便用户操作和减少各原始库的容量，设计一个存有各下属单位代码和单位名称的词库。在用户指定单位的地方，只键入有关的代码，程序自动取出单位名供使用。在数据库DWK.DBF中第1字段DWH1为前8个下属单位的单位码，第2字段DWM1为这8个下属单位的单位名；第3字段DWH2为9~16个下属单位的单位码，第4字段DWM2为这8个单位的单位名。依此类推，直到第11字段是DWH6最后8个下属单位的单位码，第12字段DWM6为最后8个下属单位的单位名。设计8个记录为的是不翻屏幕。

其程序段如下：

·MODIFY COMMAND DW

*程序名：DW.COMD，将单位名存入DWM中

USE DWK

ERASE

LIST NEXT 8 OFF

INPUT “请输入单位码” TO DM

DO CASE

CASE DM<9

GO DM

DWM=DWM1

CASE DM>8.AND.DM<17

GO DM-8

DWM=DWM2

CASE DM>16.AND.DM<25

GO DM-16

DWM=DWM3

CASE DM>24.AND.DM<33

GO DM-24

DWM=DWM4

CASE DM>32.AND.DM<41

GO DM-32

DWM=DWM5

CASE DM>40.AND.DM<49

GO DM-40

DWM=DWM6

ENDCASE

RETURN

可见，用户在屏幕的提示下输入代码比直接输入汉字的单位名方便得多。

12.3.8 巧用TOTAL命令

cdBASE中的TOTAL命令功能很强，适用于许多分类求和的情况。

1. 按多关键字分类求和

由于TOTAL命令只能按一个关键字段进行分类求和，不能满足某些应用的需要，但只要对数据库结构改动一下就可按多关键字段分类求和。方法是把有关的字段按其主次进行字符串连接，构成一个新字段，再以这个新字段作为索引和分类求和的关键字段。如：

```
REPLACE ALL DD WITH AA+BB+CC
```

以DD字段作为索引和分类求和的关键字段（这里的AA、BB和CC是希望作为关键字段分类求和的三个字段，DD为供索引和分类求和的新字段）。这样采用：

```
TOTAL ON DD TO EE,
```

EE就是所需要的结果。

2. 用TOTAL命令代替JOIN命令

由于JOIN命令时间开销大，有时失去了实用性，巧用TOTAL命令可在短时间内达到目的。

假定A库有字段工号，基本工资，实发工资，B库有字段姓名，工号。用TOTAL命令产生一个C库，其字段为姓名，工号，基本工资，实发工资。

实现的步骤是：

第一步。将A库用MODIFY STRUCTURE增加姓名

字段，宽度与B库姓名字段相同。

第二步。将B库用MODIFY STRUCTURE增加与A库相比没有的字段，基本工资，实发工资，宽度与A库的对应字段相同。

此时，字段内容为空，B库除姓名和工号字段外，其他字段为空。

第三步。用APPEND FROM命令将A库添加到B库中。

第四步。将B库以工号关键字段（用命令SORT ON 工号 TO D）排序产生D库。

第五步。对D库以工号为关键字段分类求和（TOTAL ON 工号 TO C FIELDS 基本工资，实发工资）。用FIELDS指定求和字段，为的是提高速度。C库是一个字段为姓名，工号，基本工资，实发工资的新库。相当于JOIN命令的效果。

12.3.9 美化屏幕显示

所谓美化屏幕就是如何把屏幕的显示搞得精美好看。随着计算机应用的广泛深入，人们对美化屏幕越来越感兴趣。CdBASE在美化屏幕方面的功能比BASIC要差些，但仍可以作些努力，使屏幕生动活泼。下面给出的是一个软件的序幕程序段：

```
...  
LS=1  
DO WHILE LS<=100
```

DO CASE

CASE LS<=40

@0, 40-LS SAY “%”

@0, 39+LS SAY “%”

CASE LS>40.AND.LS<=48

@LS-40, 0 SAY “% % % %”

@LS-40, 76 SAY “% % % %”

CASE LS>48.AND.LS<=88

@8, LS-49 SAY “%”

@8, 128--LS SAY “%”

ENDCASE

LS=LS+1

ENDDO

@2, 24 SAY “中”

DO YS

@2, 28 SAY “药”

DO YS

@2, 32 SAY “识”

DO YS

@2, 36 SAY “别”

DO YS

@2, 40 SAY “专”

DO YS

@2, 44 SAY “家”

DO YS

@2, 48 SAY “系”

DO YS

@2, 52 SAY "统"

DO YS

@4, 30 SAY "作者"

DO YS

@4, 37 SAY

"XXXXXXXXXXXXXXXXXXXX"

DO YS

@6, 27 SAY "浙江科学技术出版社"

DO YS

DO YS

...

延时程序段YS,

H=20

DO WHILE H>0

H=H-1

ENDDO

RETURN

执行本程序段时，首先动态的在屏幕上构成一个边框。先从上边线的中间向两边，接着是左右边线同时进行，最后是从下边线的两头向中间布上百分符“%”，上、下为一排百分符，两边各为四列百分符构成的竖线。

然后是在第二行逐个显示：中药识别专家系统，接着在第四行显示：作者，最后在第六行显示：浙江科学技术出版社。它们之间的时间间隔是调用延时程序YS来实现的。可以将百分符改为其它喜欢的字符，修改语句H=20可改变延时的长短。

12.3.10 如何调试cdBASE程序

程序写好之后，须进行测试和纠错。

1. 可能的问题

通常有这样几类错误：①命令拼写错。②命令和表达式之间无空格。③句法错，即命令使用不正确。④命令不完整。⑤运行错。

克服拼写错和命令与表达式之间无空格的方法是，打印源程序，仔细校对。错误③或④比较难以确认，提供参考的方法有。

①检查程序中内存变量名、文件名和字段名的拼写是否出错。

②注意不匹配的数据类型。例如：用同一个@命令显示数值和字符型字段，对此用转换函数纠正；对非数值变量执行数值（函数）运算，对此将变量用数值转换予以纠正。

③仔细检查语句中的顺序是否错。

④检查语句中的选择项是否有错。

⑤命令一般可简化为4个字符。如：ENDCASE和ENDC都是可以的，而ENDCA就是错误的。

运行期间的错误多是逻辑方面的，其原因不是没按cdBASE要求书写表达式，就是逻辑上与cdBASE不一致。检查是否误用了逻辑运算符，.AND.和.OR.，特别是当它们在复杂的逻辑表达式中有时就容易出错。

2. 调试的步骤

模块化的程序设计方法，可在编写这些模块程序时，同时进行调试。从整体上说，调试程序可采用如下步骤。

①写出每一模块，并使其文件化，一旦写完就加以测

试。

②使用带有其它技术的内部调试特性来调试程序模块。

③当测试完一个模块并运行正确时，继续下一模块。

④当完成相互配合的模块后，将它们联系起来，作为一个组合程序进行测试。

⑤逐步增加新的测试好的模块，建立更大的组合程序，进行相应的测试。

⑥整个项目全部程序装配完后，同样进行测试。

⑦把程序交给非作者的其他人测试，称为 α 测试。

⑧交给用户测试，称 β 测试。

⑨交互使用。使用中遇到软件问题，还要进行维护修改。

装配时采用的方法和设计阶段采用的自顶向下刚好相反，应该采用自底向上进行。

在 α 和 β 测试中，发现的错误最好让程序作者修改，因为作者比较清楚程序的内部结构，容易避免修改中增加新的错误。在动手修改之前，应将程序和数据备份，以防调试的程序不能运行。有时，调试一个程序模块最有效的方法就是重写整个程序模块或有关部分。

3. 有关的调试命令

下面列出一些与调试有关的命令，可使用这些命令，帮助调试cdBASE程序。

①LIST STRUCTURE

作用：显示数据库的结构。

②LIST MEMORY

作用：显示内存变量的内容。

③SET TALK ON

作用：显示执行各条命令的结果。

④SET STEP ON

作用：使cdBASE执行一条语句就停一下。

⑤SET ECHO ON

作用：使执行的语句在屏幕上显示出来。

⑨SET ALTERNATE TO[〈文件名〉]

SET ALTERNATE ON

作用：执行这两条命令后，把屏幕上或打印机上出现的信息用指定的文件名存在磁盘上。

⑦SET PRINT ON

作用：接通打印机。

4. 调试方法

①单步执行

在程序执行前，执行如下的命令：

SET STEP ON

SET ECHO ON

SET TALK ON (或SET DEBUG ON)

在运行程序时，一条一条地执行，能观察执行顺序及判断结果是否正确。

②设“断点”调试

在有可能出错的位设置“断点”（用WAIT命令）及显示的提示信息，说明位置前面执行正常，请再继续，……等。

例如：

……

? “第n段程序执行正常，请按任一键继续!”

WAIT

.....

? “第n+j段程序执行正常，请按任一键继续!”

WAIT

.....

③用ESC键中断运行，查看“现场”。

在执行中，若执行到某一个“断点”(WAIT)处，发现有错，或不知是否正确？想要查看“现场”，按ESC键中止程序运行，用显示或打印命令，检查内存变量、程序循环次数、数据库中的数据、记录号的值等等。

④用编译dBASE编译查错

有dBASE编译软件时，可以用编译dBASE编译dBASE程序，查出错误。编译查错一般都可以查出多个或全部错误。

第十三章 cdBASE与BASIC文件的数据共享

所谓 cdBASE 与 BASIC 文件的数据共享，是指用 cdBASE建立的数据库文件的数据能够供 BASIC文件（其它文件也如此）使用；相应地，用BASIC建立的文件的数据也可以提供给数据库文件使用。本章着重介绍cdBASE和BASIC文件间进行数据共享的方法。这些方法原则上也适用于cdBASE同其它外部文件间的数据共享。在学习这一章之前，要求读者熟悉BASIC语言及其文件系统。

13.1 数据共享的基本方法

从前面几章介绍的cdBASE中，我们已经清楚地看到：数据库文件是一种扩展名为.DBF的数据文件，而BASIC能够提供的却是扩展名为.TXT的一种文本文件（CEC-I中又称T文件，即文件是由文本字符组成的。不论是顺序文件、随机文件还是程序文件，BASIC中均称为T文件）。要使得两种类型文件的数据能够共享，最关键的问题就是要想办法使得两类文件能够互相转换，以适应双方对文件的类型要求。cdBASE提供的COPY和APPEND语句能够作到这一点，这就为数据库和外部文件间的数据共享打开了方便之门。

13.1.1 数据库和文本文件之间的类型转换

cdBASE除了便于建立、检索、插入、删除和修改数据记录以外，还能够很容易地在数据库和文本文件之间进行类型的相互转换，这为cdBASE中使用其它语言文件的数据提供了可能性。

1. 用.DBF文件来生成.TXT文件

cdBASE中最关键最重要的文件就是数据库文件，它的重要性表现在两个方面：一是保存了最重要最基本的数据记录；二是很多其它文件均必须建立在数据库文件的基础之上，例如：索引文件、报表格式文件。BASIC中保存数据记录的文件是T文件，要使得BASIC能够检索数据库中数据，必须将数据库文件转化为文本文件。

格式：COPY TO <文本文件名>[<范围>]

[FIELD<字段表>]

[FOR<条件>]{SDF/DELIMITED}

作用：用活化数据库文件来生成一个新的文本文件，原数据库文件不受任何影响。

系统在执行这条命令时，首先将数据库文件的数据复制到一个新文件中；再将新文件的类型变成文本字符型。在cdBASE中，只需执行这样一条拷贝命令，类型转换就完成了。但使用中必须注意：

①这是在cdBASE状态下来执行类型转换，因而活化数据库文件必须满足cdBASE的要求。

②这里的<范围>、FOR<条件>以及FIELD<字段表>含义同cdBASE中一样，根据需要来选用。

③SDF或DELIMITED指出转换时系统可用的转换方

式。SDF表示按记录进行复制，DELIMITED表示按字段进行复制。

④这个命令只是由数据库文件生成了一个新的文本文件，它并不破坏原数据库文件。

⑤用COPY来生成的文本文件可以供外部文件使用。

2. 用·TXT文件来生成·DBF文件

如果cdBASE中要使用外部文件的数据，那么必须将外部文件由·TXT类型转换成·DBF类型。cdBASE中提供的转换命令如下：

格式：APPEND FROM <文本文件名> [FOR <条件>]
<SDF/DELIMITED>

作用：将文本文件的数据按·DBF文件的要求添加到活化数据库文件的末尾，添加操作不影响原来的文本文件。

经过这个命令，文本文件的数据就可以为数据库所利用了。这里的FOR <条件>仅对文本文件起作用，表示文本文件中只有满足给定条件的那些记录才被添加到活化数据库文件的末尾。SDF和DELIMITED含义同前述。

有了COPY...SDF/DELIMITED和APPEND...SDF/DELIMITED命令，数据库和外部文件之间的数据共享就十分方便了。

13.1.2 SDF和DELIMITED的区别

前已述及，SDF和DELIMITED均是用于数据类型转换的，那么它们之间究竟有什么区别呢？下面我们用一个例子来说明。

例1. 假设商品文件的结构和数据如下：

字段 名字 类型 宽度 小数

001	货号	C	003	
002	品名	C	006	
003	单价	N	007	002
004	型号	C	006	

合计 00023

·USE商品↙

·LIST↙ (为简便只取前3个记录)

00001	115	自行车	170.93	·	环球-2
00002	145	电冰箱	979.41		北极星
00003	267	电饭锅	34.89		乐天牌

1.用SDF格式生成文本文件

·USE 商品↙

·COPY TO SPSDFT SDF↙

·QUIT↙ (退出cdBASE)

A>TYPE SPSDFT.TXT↙ (回到操作系统, 显示文件内容)

115	自行车	170.93	环球-2
145	电冰箱	979.41	北极星
267	电饭锅	34.89	乐天牌

这种形式同数据库显示的记录格式没有什么区别。

2.用DELIMITED格式生成的文本文件

我们仍然用商品文件为例。

·USE 商品↙

·COPY TO SPDET DELIMITED↙

·QUIT↙

A>TYPE SPDET.TXT↙

用DELIMITED格式生成的文本文件数据是：

“115”，“自行车”，170.93，“环球-2”

“145”，“电冰箱”，979.41，“北极星”

“267”，“电饭锅”，34.89，“乐天牌”

由上述可见，SDF和DELIMITED的区别是：

①用DELIMITED生成的文本文件，各字段值间用逗号隔开，且字符型数据必须用引号括起来。而SDF的各字段间由空格隔开且不带引号。

②SDF是以记录为单位来复制数据从而生成文本文件，而DELIMITED是以字段为单位来复制数据从而生成文本文件。

13.2 如何在BASIC中使用数据库文件

要在BASIC程序中使用cdBASE数据库文件提供的数
据，可分如下几步来操作：

1. 用cdBASE来生成文本文件

· USE 商品 ✓

· COPY TO SPTEXT SDF ✓

2. 退出cdBASE，进入BASIC状态

3. 用BASIC程序来检索SPTEXT文本文件

```
10 REM这是检索文本文件的例子
```

```
20 D$=CHR$(4)
```

```
30 PRINT D$; "OPEN SPTEXT" (打开文件)
```

```
40 PRINT D$; "READ SPTEXT" (读文件)
```

```
50 FOR I=1 TO 3
```

```
60 INPUT A$ (输入文件记录)
```

```

70 PRINT A $ (打印记录)
80 NEXT I
90 PRINT D $; "CLOSE SPTEXT" (关闭文件)
100 END
RUN ↙ (运行BASIC程序, 得出结果)

```

```

115 自行车 170.93 环球-2
145 电冰箱 979.41 北极星
267 电饭锅 34.89 乐天牌

```

使用中需要注意以下几点:

①数据库文件必须事先建好, 其数据字段、类型、宽度应当与BASIC文件的要求相一致。

②数据文件必须保存在一张公共软盘上。

③为用起来方便, CEC-I最好配备两个软盘驱动器。

④这里仅对生成的文本文件进行了检索, 也可以进行其它操作, 请读者自己完成, 方法与此类似。

13.3 如何在cdBASE中使用BASIC文件

我们用BASIC语言中介绍过的学生文件为例, 来说明如何在cdBASE中使用BASIC的文件。实现的操作步骤是:

1. 事先建立一个BASIC顺序文件

```
10 REM建立BASIC的文本文件
```

```
20 D $=CHR $(4)
```

```
30 PRINT D $; "OPEN B: 学生" (打开文件)
```

```
40 PRINT D $; "WRITE B: 学生" (写文件)
```

```
50 FOR I=1 TO 3
```

```
60 READ A $ (读数据)
```

```

70 PRINT A $ (写入学生文件)
80 NEXT I
90 PRINT D $, "CLOSE B, 学生" (关闭文件)
100 DATA "张 山 631 男 15 95"
105 DATA "李丽丽 632 女 13 88.5"
108 DATA "王大力 634 男 14 74.7"
110 END
RUN ✓

```

2. 退出BASIC, 进入数据库cdBASE。

3. 用cdBASE建立一个类似学生文件的数据库文件结构

• CREATE SDB ✓

字段	名字	类型	宽度	小数
001	姓名,	C,	6 ✓	
002	学号,	C,	3 ✓	
003	性别,	C,	2 ✓	
004	年龄,	N,	2 ✓	
005	成绩,	N,	4,	1 ✓
006	✓			

4. 将BASIC的学生文件的内容添加到数据库文件的末尾。

• USE SDB ✓

• APPEND FROM B, 学生 SDF ✓

• LIST ✓

```

张 山 631 男 15 95.0
李丽丽 632 女 13 88.5
王大力 634 男 14 74.7

```


这样，我们就可以用cdBASE命令如同数据库文件那样来检索、插入、删除和修改SDB文件的数据了。

一般说来，用cdBASE建立检索、修改、插入和删除数据库文件比用BASIC文件来得简单，而进行加减乘除等算术运算BASIC又更为方便。因此，我们可以用 cdBASE 来建立、装入、检索、修改数据库的数据，而用BASIC来统计、运算数据库中的数据。这样更能发挥各自的优势和长处。这里介绍的思想和方法，不但适用BASIC，也适用于其它高级语言，如：FORTRAN、PASCAL等文件同 cdBASE的数据共享。

习 题 十 三

1. 数据库文件同外部语言文件共享数据的基本思想是什么？如何来实现？

2. 假设软盘上已经有一个名字为TEXTFILE的文本文件（字段有：号码、姓名、年龄、文化程度、职称等），希望在 cdBASE中使用这一文件的内容，如何实现？请写出程序。

3. 假设软盘上已经有一个数据库文件DBFILE，希望在BASIC中检索这一文件的内容，如何实现？请写出程序。

4. 如果转换中要用到DELIMITED，那么实现时需要注意什么？请用程序来实现。

5. 如果不懂 cdBASE而懂某一种高级语言，能否使用 cdBASE数据库文件的数据？反之，如果不懂高级语言而只懂 cdBASE，那么能否应用高级语言文件的数据？应该如何

办？

6.就自己熟悉的一种语言文件进行数据共享处理，写出实现共享的程序。

第十四章 cdBASE应用实例

本章采用由小到大，由简到繁的原则，介绍几个数据库应用系统。使读者逐步掌握应用汉字dBASE数据库系统来解决实际问题的能力。

对于初学者来说，在学习掌握前面各章内容的基础上，还需注意多看一些实例，摸索编程的规律和技巧。本章介绍的各个程序都是在一些单位投入运行的实用软件。

14.1 快速盘点系统

本程序按用户经营商品的单价分别接收并储存当天的增（进货）、减（销售）数据，自动快速计算输出当日的结存数量与金额，让经营者及时掌握盈亏情况，提高经营水平和竞争能力。此外，程序还可随时提供增减商品表，未增减商品表，供经营者分析决策。

14.1.1 快速盘点数据库文件（KSPD.DBF）结构

数据库结构：A：KSPD.dbf

记录个数：6

最后更新日期：06/01/89

字段	字段名	类型	宽度	小数
1	单价	Numeric	8	4
2	昨日结存	Numeric	5	

3	本日增加	Numeric	5	
4	本日减少	Numeric	4	
5	本日结存	Numeric	5	
6	结存金额	Numeric	10	4
总计			38	

14.1.2 程序功能结构图

快速盘点程序功能结构如图14.1所示。

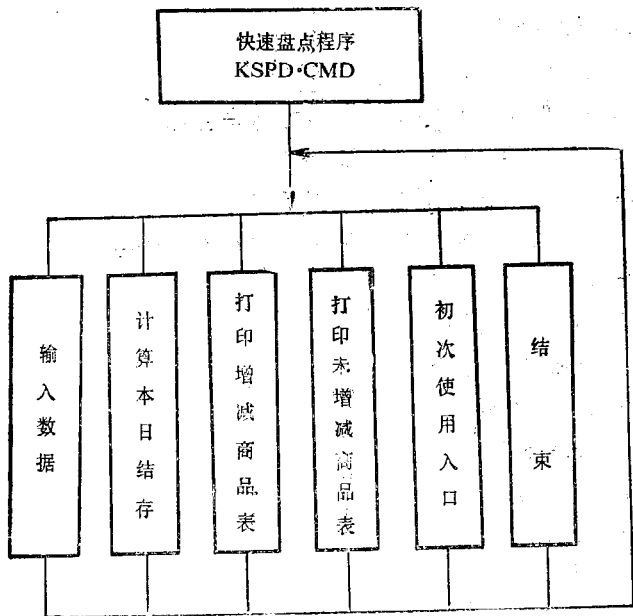


图14.1 快速盘点程序功能结构图

14.1.3 源程序清单及解释

1* 这是一个快速盘点结算程序

2 STOR "N" TO HD

```

3 DO WHILE HD= "N"
4 ERASE
5 ACCEPT "请输入日期 ( × × . × × . × × )"
   TO RQ
6 ? "日期: &RQ正确否 ( Y/N )"
7 WAIT TO HD
8 ENDDO
9 DO WHILE 1<2
10 ERASE
11 ? "1.....输入数据"
12 ? "2.....计算本日结存数量与金额"
13 ? "3.....打印增减商品表"
14 ? "4.....打印未增减商品表"
15 ? "5.....初次输入数据 ( 要输昨日结存 )"
16 ? "0.....结束"
17 ? "请选择!"
18 WAIT TO XZ
19 USE KSPD
20 ERASE
21 DO CASE
22 CASE XZ= "0"
23 USE
24 CANCEL
25 CASE XZ= "1" .OR. XZ= "5"
26 IF XZ= "5"
27 DELETE ALL
28 PACK

```

29 ENDIF
30 IF XZ= "1"
31 REPL ALL昨日结存WITH 本日结存,
 本日增加WITH 0, 本日减少WITH 0
32 ENDIF
33 COUNT TO JL!
34 STORE "Y" TO HD
35 DO WHILE HD= "Y"
36 STORE "N" TO XD
37 STORE 0 TO JC
38 STORE 0 TO DJ
39 STORE 0 TO ZJ
40 STORE 0 TO JS
41 DO WHILE XD= "N"
42 ERASE
43 IF XZ= "5"
44 @ 2, 20 SAY "请输入昨日结存" GET JC
45 ENDIF
46 @ 3, 20 SAY "请输入单价" GET DJ
47 @ 4, 20 SAY "请输入本日增加" GET ZJ
48 @ 5, 20 SAY "请输入本日减少" GET JS
49 READ
50 @6, 20 SAY "请校对, 正确否(Y/N)",
 GET XD
51 READ
52 ENDDO
53 LOCATE FOR 单价=DJ

```

54 IF EOF
55 APPEND BLANK
56 REPL 单价WITH DJ,昨日结存WITH JC
57 ENDIF
58 REPL 本日增加WITH ZJ,
    本日减少WIHT JS
59 ERASE
60 @ 7, 20 SAY "继续输入吗? (Y/N)"
    GET HD
61 READ
62 ENDDO
63 COUNT TO JL2
64 IF JL1<JL2
65 INDEX ON 单价 TO KSPDIN
66 ENDIF
67 CASE XZ= "2"
68 USE KSPD
69 REPL ALL本日结存 WITH 昨日结存+
    本日增加-本日减少
70 REPL ALL 结存金额 WITH 单价*本日结存
71 CASE XZ= "3"
72 USE KSPD INDEX KSPDIN
73 STORE "本日增加< >0.OR.本日减少
    < >0" TO TJ
74 SET PRINT ON
75 ? " "
76 ? " "

```

- 77 ? “ &RQ增减商品表”
- 78 LIST OFF 单价, 本日增加, 本日减少, 本日结
存, 结存金额 FOR &TJ
- 79 ? “ ”
- 80 SET PRINT OFF
- 81 SUM 结存金额 TO JE FOR &TJ
- 82 STORE 0 TO ZJ
- 83 STORE 0 TO JS
- 84 GO TOP
- 85 DO WHILE .NOT. EOF
- 86 STORE 单价 * 本日增加 + ZJ TO ZJ
- 87 STORE 单价 * 本日减少 + JS TO JS
- 88 SKIP
- 89 ENDDO
- 90 SET PRINT ON
- 91 ? “ ”
- 92 ? “本日增加总额”, ZJ, “减少总额”, JS
- 93 ? “增减结存总额”, JE
- 94 ? “ ”
- 95 SET PRINT OFF
- 96 CASE XZ = “4”
- 97 USE KSPD:INDEX KSPDIN
- 98 STORE “本日增加=0.AND.本日减少=0”
TO TJ
- 99 SET PRINT ON
- 100 ? “ ”
- 101 ? “ ”

- 102 ? " &RQ未增减商品表"
- 103 LIST 单价, 本日结存, 结存金额
OFF FOR &TJ.AND.本日结存< > 0
- 104 ? " "
- 105 SET PRINT OFF
- 106 SUM 结存金额 TO WJE FOR &TJ
- 107 SET PRINT ON
- 108 ? "未增减结存总额", WJE, "总计结存额",
JE+WJE
- 109 ? " "
- 110 SET PRINT OFF
- 111 ENDCASE
- 112 ENDDO
- 113 RETURN

注意：各语句前的序号是为了解释加注的，运行的程序中是不应该有的。

本程序第1句是一个注释语句，注明本程序是一个快速盘点结算程序。第2~8句提示并接受用户输入日期，采用提示用户校对输入日期的办法，保证输入日期的正确性。循环结构直到用户确认输入的日期正确，按Y键中止循环。宏替换&RQ将变量RQ接受的日期替换在语句中的&RQ处，如，输入为：89.05.11↵，则执行第6句的屏幕显示为：

日期：89.05.11正确否（Y/N）

若用户认为正确按Y键；否则按N键，再重输日期。

第9~112句是程序主体，为让用户自由选择各项功能，除“0”以外的1~5各项功能选择运行后，均重现功能显示，供用户选择执行。故使用循环语句，只有当用户选择

“0”，才结束循环并使整个程序的运行停止。第10句和前面的第4句是清屏幕的语句，作用是清除执行本语句前屏幕上的显示。第11~16句是本程序的功能菜单。17句提示用户选择11~16句中的功能。19句是打开数据库KSPD.DBF。21句是区别各种功能情形的情形语句的开头。22句判断选择，若为“0”则执行23句关闭数据库，24句终止程序运行，回到数据库系统点(·)提示符。

第25句是选择功能1或5的情形入口。26~29句完成初次使用的初置工作，即删去旧(或称表演)数据。30~32句，对通常输入数据时，首先将前一天的本日结存列入昨日结存，并将前一天的增加减少清零(以免昨天的增减数引起混乱)。33句统计并保存前一天的记录数，以备后用。34句为使35句的首次循环条件为真而给变量HD置初值。第36句同34句，37~40句为工作变量清零(0)，41句~52句为校对修改输入错误的循环，其方法与3~8句对输入日期的处理类似，其中43~45句供初次使用输入昨日结存。

第53句寻找库中单价与输入单价相等的记录。54句判断是否文件尾，若是文件尾，说明库中无本次输入单价的记录，则由55句添加一个空记录，将输入数据通过56、58句装入新记录；若不是文件尾，说明输入单价在数据库中已经存在，则由58句装入本日增加和减少。由执行60句时用户的回答(Y/N)决定36~62句是否重复执行。

第63句统计并保存输入数据之后的记录数。64句比较输入数据前后的记录数，判断是否增加了记录，若输入后的记录数JL2大于JL1，则执行65句索引数据库。便于选择3或4时按索引库打印商品表。

第71句是选择2(计算结存)的入口，69和70句完成结

存数量与金额的计算。

第71句是选择3（打印增减商品表）的入口，72句打开索引库。73句赋字符串值到变量TJ，供78句和81句作条件使用。74句连通打印机，80句断开打印机，下同。对于未经扩充，只有一个扩充槽插入Z₀₀卡使用数据库的情况下，对这种与打印机有关的语句前加一个星号“*”作为注释句。75句和76句使打印或显示空走两行。77句打印表头题目。78句打印列出本日有增减变化的记录（因每天输入前增、减通通置为零，输入后打印或显示时不等于零，就表示有增加或减少）。81句合计增、减结存额，82~89句合计增加、减少总额，92~93句打印这三个值。

第96句是选择4（打印未增减表）的入口，98句将103句和106句中用的条件送入变量TJ，103句按单价列出未增减商品的结存数量与金额。106句合计未增减商品结存额，108句打印未增减商品结存及增减和不增减商品总金额。从108句总金额由JE+WJE可见，必须先打印商品增减表（计算JE），才能再打印未增减表中的总结存额。

14.1.4 说明

本程序比较简短，比较适宜初学者学习。本程序由成都三开元电脑部组织开发并投入使用。为满足经营决策及业务需要，历经本部不断的扩充完善，已发展成为商品经营的综合管理软件，实现了从商品入库、调拨、库存、销售（批、零）、汇总等全面微机管理。

14.2 住房管理系统

本程序是应某部房管处需要开发的。适用于一般单位的职工住房数据信息的输入、修改、分类、检索及统计。可为用户在收房租、计划新建、改建和分配职工住房等工作中提供决策参考信息。

程序简单明了，适应初学者，易于维护使用。

14.2.1 住房管理数据库 (ZFGLK.DBF) 结构

数据库结构：a: zfglk.dbf

记录个数：4

最后更新日期：05/16/89

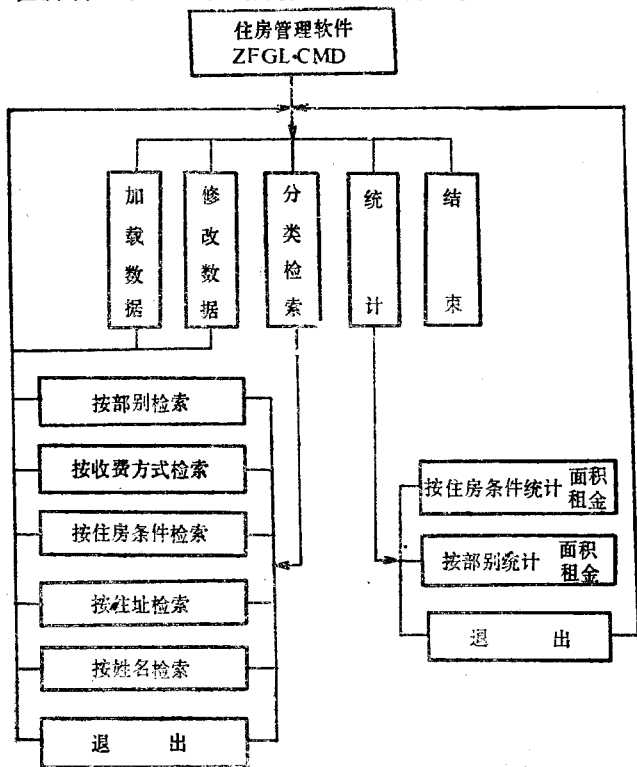
字段	字段名	类型	宽度	小数
1	住址	Character	10	
2	建面积	Numeric	5	1
3	用面积	Numeric	5	1
4	租金	Numeric	6	2
5	间	Character	2	
6	费	Character	2	
7	部别	Character	8	
8	姓名	Character	6	
9	职务	Character	8	
10	条件	Character	2	
11	进时	Character	6	
总计			61	

其中：字段住址的值为房屋的房号，由幢、单元、楼

层、门号之代码构成，共计10个字符；建面积即住房的建筑面积；用面积即住房的使用面积；费为收费方式，其值是：扣/收/非（即记录中字段费的内容是扣、收、非三者之一，扣即从工资中扣收，收即收款，非即非扣非收的其它方式）。字段条件为职工居住房屋的条件，其值可以是在、余、离、退、调、烈、转之一。进时为住户住进房屋的时间(年月)。

14.2.2 住房管理软件功能结构图

住房管理软件功能结构如图14.2所示。



14.2.3 源程序清单

```
1 *这是一个住房管理程序
2 DO WHILE 1=1
3 ERASE
4 ? "1.....录入"
5 ? "2.....修改"
6 ? "3.....检索"
7 ? "4.....统计"
8 ? "0.....结束"
9 ? "请选择"
10 WAIT TO XZ
11 STORE "Y" TO HD
12 USE ZFGLK
13 DO CASE
14 CASE XZ= "0"
15 CANCEL
16 CASE XZ= "1"
17 DO WHILE HD= "Y"
18 APPEND BLANK
19 ? "记录号" . #
20 EDIT
21 ? "继续吗(Y/N)"
22 WAIT TO HD
23 IF HD= "N"
24 USE
25 ENDIF
```

26 ENDDO
27 CASE XZ= "2"
28 DO WHILE HD= "V"
29 ERASE
30 ACCEPT "请输入修改的房号" TO FH
31 LOCATE FOR 住址= FH
32 ? "记录号" #
33 EDIT
34 ? "继续修改吗 (Y/N)"
35 WAIT TO HD
36 IF HD= "N"
37 USE
38 STORE "N" TO HD
39 ENDIF
40 ENDDO
41 CASE XZ= "3"
42 STORE "1" TO HD
43 DO WHILE HD= "1"
44 ERASE
45 ? "1.....部别"
46 ? "2.....收费方式"
47 ? "3.....住房条件"
48 ? "4.....住址"
49 ? "5.....姓名"
50 ? "0.....退出"
51 ? "请选择检索方式"
52 WAIT TO XZ2

```

53 ERASE
54 ? “检索结果是打印/显示(Y/N)”
55 WAIT TO DY
56 IF DY= “Y”
57 SET PRINT ON
58 ELSE
59 SET PRINT OFF
60 ENDIF
61 GO 1
62 DO CASE
63 CASE XZ2= “1”
64 ACCEPT “输入部别名” TO BB
65 DO WHILE .NOT. EOF
66 LIST NEXT 10 OFF FOR
    @( “&BB” , 部别 ) > 0
67 ? “按任一键继续”
68 WAIT
69 ENDDO
70 CASE XZ2= “4”
71 ACCEPT “输入住址” TO ZZ
72 GO 1
73 DO WHILE .NOT. EOF
74 LIST NEXT 10 OFF FOR @( “&ZZ” ,
    住址 ) > 0
75 ? “按任一键继续”
76 WAIT
77 ENDDO

```



```

78 CASE XZ2= "5"
79 ACCEPT "请输入姓名" TO XM
80 LIST OFF FOR @( "&XM" , 姓名 ) > 0
81 ? "按任一键继续"
82 WAIT
83 CASE XZ2= "2"
84 ACCEPT "请输入收费方式(扣/收/非)" TO
SF
85 DO WHILE .NOT. EOF
86 LIST NEXT 10 OFF FOR @( "&SF" ,
    费 ) > 0
87 ? "按任一键继续"
88 WAIT
89 ENDDO
90 CASE XZ2= "3"
91 ACCEPT "输入住房条件(在/余/离/退/调/烈/
    转)" TO TJ
92 DO WHILE .NOT. EOF
93 LIST NEXT 10 OFF FOR
    @( "&TJ" , 条件 ) > 0
94 ? "按任一键继续"
95 WAIT
96 ENDDO
97 CASE XZ2= "0"
98 STORE "0" TO HD
99 ENDCASE
100 IF DY= "Y"

```

101 SET PRINT OFF
 102 ENDIF
 103 ENDDO
 104 CASE XZ= "4"
 105 STORE "1" TO HD
 106 DO WHILE HD= "1"
 107 ERASE
 108 ? "1……按住房条件统计面积与租金"
 109 ? "2……按部别统计面积与租金"
 110 ? "0……退出"
 111 ? "请选择!"
 112 WAIT TO XZ3
 113 DO CASE
 114 CASE XZ3= "1"
 115 ACCEPT "请输入住房条件(在/余/离/退/调/
 烈/转)" TO TJ
 116 SUM建筑面积, 用面积, 租金 TO JM, SM, ZJ
 FOR @("&TJ", 条件) > 0
 117 ? "建筑面积 使用面积 租金"
 118 ? JM, SM, ZJ
 119 ? " "
 120 CASE XZ3= "2"
 121 ACCEPT "请输入部别" TO BB
 122 SUM建筑面积, 用面积, 租金 TO JM, SM, ZJ
 FOR @("&BB", 部别) > 0
 123 ? "住房部别为 &BB"
 124 ? "建筑面积 使用面积 租金"

```
125 ? JM, SM, ZJ
126 ? " "
127 CASE XZ3= "0"
128 STORE "0" TO HD
129 ENDCASE
130 IF DY= "Y"
131 SET PRINT OFF
132 ENDIF
133 ENDDO
134 ENDCASE
135 ENDDO
136 RETURN
```

本程序较简明，于此仅对可能费解的地方解释如下：

1. 第19句的作用是，当执行本句时，屏幕显示当前记录号，因为20句EDIT命令执行时，要求提供记录号，用户照执行19句时显示的记录号键入后，则按提示一个字段一个字段地输入到添加的空记录中。

2. 程序的检索部分，（情形）CASE语句没有依菜单出现的顺序出现1~5种情形，在63句是CASE XZ2= "1"，70句是CASE XZ2= "4"，83句才是CASE XZ2= "2"，这是允许的，不影响执行效果。

3. 在统计程序段中的117句是为在118句显示统计数字的上面，显示对应的名称，这些名称的位置根据试运行时118句统计数字的显示位置，用插入空格来调整。

4. 第65句至69句的程序功能是每屏显示10个符合条件的记录，用户看清后任按一键，又循环显示下一屏符合条件的记录，如此直到文件结束。

附录 dBASE命令和出错

一、dBASE命令一览表

?/?? <表达式>

显示表达式的值。

@<行, 列> [SAY <表达式>][USING <PICT子句>]]
[GET <变量>][PICT子句]]

按给定的行列位置显示表达式的值。

ACCEPT[<提示>] TO <内存变量>

允许从键盘上接收一串字符到内存变量中。

APPEND [BLANK]

允许在活化数据库文件的末尾添加若干新记录。

APPEND FROM <文件名> [SDF/DELIMITED]

[FOR <条件>]

允许将指定文件添加到活化数据库文件的末尾。

BROWSE

浏览并可修改活化数据库记录。

CANCEL

中止命令文件的执行, 无条件返回到点状态。

CHANGE [(<范围>)] FIELD <字段表>[FOR<条件>]]

允许修改活化数据库中指定字段的值。

CLEAR

清除内存变量和关闭所有打开的文件。

CONTINUE

继续定位满足LOCATE中条件的下一个记录。

COPY TO <文件名>[SDF/DELIMITED] [(范围)]
[FIELD <字段表>] [FOR<条件>]

将活化数据库文件复制到指定文件中。

COPY STRUCTURE TO <文件名>

将活化数据库文件的结构复制到指定文件中。

COPY TO <文件名> STRUCTURE EXTENDED

用活化数据库文件的结构来生成一个结构文件。

COUNT [(范围)] [TO <内存变量>] [FOR<条件>]

统计记录个数。

CREATE [(文件名)]

允许用对话方式来建立一个数据库文件。

DELETE [(范围)] [FOR <条件>]

对活化数据库文件的记录注删除标记。

DELETE FILE <文件名>

允许在数据库状态下删除磁盘文件。

DISPLAY [(范围)] [OFF] [FOR<条件>] [FIELD
<字段表>]

显示活化数据库文件的记录。

DISPLAY STRUCTURE

显示活化数据库文件的结构。

DISPLAY FILES LIKE <文件名·扩展名>

显示磁盘文件名。

DISPLAY MEMORY

显示内存变量的名称、类型及其值。

DO <文件名>

调用指定的文件。

DO WHILE 〈条件〉

 〈循环体〉

ENDDO

当条件为真时，反复执行循环体，直到条件为假。

EDIT [(〈记录号〉)]

允许修改指定记录的值。

EJECT

使打印机换页。

ERASE

清屏幕且将光标置于屏幕的左上角。

FIND 〈字符序列〉

查找索引文件中满足给定字符序列的第一个记录（定位记录指针）。

GO(TO){RECORD 〈n〉/TOP/BOTTOM}

将记录指针指向活化数据库文件的第n个记录(RECORD n)、或文件的开头(TOP)、或文件的末尾记录。

IF 〈条件〉

 〈命令序列1〉

 [ELSE

 〈命令序列2〉]

ENDIF

根据条件来选择执行命令序列1还是命令序列2，然后再执行ENDIF的后继命令。

INDEX ON 〈关键字段表达式〉 TO 〈文件名〉

在活化数据库文件的基础上，按给定的关键字段表达式来建立索引文件。

INPUT [〈提示〉] TO 〈内存变量〉

允许从键盘上输入表达式的值。

INSERT [BLANK] [BEFORE]

允许在当前记录位置上插入一个新记录。

JOIN TO〈文件名〉 FOR〈条件〉[FIELD〈字段表〉]

将两个活化数据库文件的记录按条件连接起来，从而生成一个新的数据库文件。

LIST [OFF] [FOR 〈条件〉] [FIELD 〈字段表〉]

用卷动方式显示活化数据库文件的记录。

LIST STRUCTURE

显示活化数据库文件的结构。

LIST FILE 〈文件名〉

显示磁盘文件。

LIST MEMORY

显示内存变量的名称类型及其值。

LOCATE [〈范围〉] FOR 〈条件〉

将指针定位到活化数据库中满足给定条件的第一个记录位置上。

LOOP

无条件转到本层循环 (DO WHILE...ENDDO) 的开头，以进行下一次循环。

MODIFY COMMAND 〈文件名〉

允许建立或修改命令文件。

MODIFY STRUCTURE

允许修改活化数据库文件的结构。

NOTE/* 〈字符序列〉

将NOTE或*之后的字符序列当成注解。

PACK

将活化数据库文件中注有删除标记的记录文件中去掉。

QUIT

退出dBASE II，回到操作系统状态。

READ

允许从键盘上读入变量的值。READ一定要同@〈行，列〉 GET 〈变量〉联用。

RECALL [〈范围〉] [〈FOR 〈表达式〉]

去掉活化数据库记录之前注有的删除标记*。

RELEASE [〈内存变量表〉] [ALL]

删去指定的内存变量。

REMARK 〈字符序列〉

用于在命令文件中作注解。注解内容可以显示在屏幕上。

RENAME 〈旧文件名〉 TO 〈新文件名〉

将旧文件名重新命名为新文件名。

REPLACE [〈范围〉] 〈字段1〉 WITH 〈表达式1〉

[〈，〉 〈字段2〉 WITH 〈表达式2〉]…[FOR 〈条件〉]

用给定的表达式的值替换相应字段的当前值。

REPORT 〈文件名〉

用对话方式建立打印报表需要的格式信息。

REPORT FORM 〈文件名〉 [〈范围〉] [TO PRINT]

[FOR 〈条件〉]

调用已建立的报表格式文件输出报表数据。

RESTORE FROM 〈文件名〉

调用内存存储器文件。

RETURN

将控制返回到主程序的下一个语句。

SAVE TO 〈文件名〉

将内存变量组织到文件中。

SELECT PRIMARY/SECONDARY

选择主区或辅区作为当前工作区。

(以下凡划线者为系统默认的功能状态。)

SET ALTERNATE ON/OFF

是/否输出文本字符到文本文件中。

SET ALTERNATE TO 〈文件名〉

建立一个文本文件。

SET BELL ON/OFF

输入非法数据或填满字段宽度时是/否响铃。

SET CARRY ON/OFF

在用APPEND添加数据时，上一个记录的数据是/否带到下一个记录中。

SET CONFIRM ON/OFF

按回车键后光标是/否自动跳到下一个字段。

SET COLON ON/OFF

在@命令中，是/否用冒号(：)作GET数据项的分界符。

SET CONSOL ON/OFF

是/否在屏幕上显示输出信息。

SET DATE TO mm/dd/YY

重新设置系统日期(按月/日/年的形式输入数据)。

SET DEBUG ON/OFF

是/否将ECHO和STEP命令的执行结果送到打印机上输出。

SET DEFAULT TO <驱动器标志>

设置当前工作盘所在的驱动器。

SET ECHO ON/OFF

在执行命令文件时，是/否将命令显示在屏幕上。

SET EJECT ON/OFF

在用REPORT打印报表之前，打印机是/否换页。

SET ESCAPE ON/OFF

是/否允许用 Esc 键中止命令文件的执行。

SET EXACT ON/OFF

字符串作比较时是/否要求严格匹配。

SET FORMAT TO SCREEN/PRINT

将@命令执行的结果送到屏幕/打印机上输出。

SET FORMAT TO <格式文件名>

打开格式文件。

SET HEADING TO <页标题>

用新的页标题来代替REPORT命令中原来的页标题。

SET INDEX TO <索引文件名字表>

打开指定的索引文件。

SET INTENSITY ON/OFF

全屏幕操作是/否使用高亮度屏幕字符。

SET LINKAGE ON/OFF

是/否使主区和辅区的数据库建立联系。

SET MARGIN TO <n>

设置打印报表的开始位置（左边界）。

SET PRINT ON/OFF

是/否连通打印机。

SET RAW ON/OFF

当使用DISPLAY或LIST命令时，字段之间是/否留空白。

SET SCREEN ON/OFF

对APPEND、INSERT、EDIT和CREATE命令是/否打开全屏幕操作。

SET STEP ON/OFF

在命令文件中，执行一条命令后是/否暂停。

SET TALK ON/OFF

是/否在屏幕上显示命令执行的结果。

SKIP [(+/-) (数值表达式)]

将当前的记录指针向前（记录号增大的方向）或向后（记录号减小的方向）拨动若干个记录位置。

SORT ON (字段名) TO (文件名) [ASCENDING/DESCENDING]

将活化数据库文件的记录按给定字段排序后生成一个新的有序文件。

STORE (表达式) T) (内存变量)

将表达式的值传送到内存变量中。

SUM (字段表) [TO (内存变量表)] [(范围)]

[FOR(条件)]

将活化数据库文件的记录按列求和（字段一定是数值型）。

TOTAL ON (关键字段) TO (文件名)

[FIELD(字段表)] [FOR (条件)]

将活化有序文件中关键字段值相等的所有记录进行合并。

UPDATE FROM (文件名) ON (关键字段)

[ADD <字段表>] [REPLACE <字段表>]

用一个有序文件来修改另一个有序文件。

USE <文件名> [INDEX <索引文件名字表>]

打开指定的数据库文件（以及与之有关的索引文件）。

USE

关闭所有文件。

WAIT [<提示>] [TO <内存变量>]

中断命令文件的执行，等待用户键入任何一个字符后又继续执行后继命令。

二、dBASE出错提示信息

这里的出错提示信息不仅适用于dBASE I，也适用于dBASE III。

Alias name already in use

别名已经在使用。这说明别名所代表的数据库文件在此之前已经被打开了，如果企图在另一工作区内再打开该数据库文件，就会显示这种提示信息。因为dBASE规定：同一数据库文件不能同时在多个不同的工作区内打开。针对这种情况，可以将所有数据库文件或发生问题的数据库文件关闭后再单独重新打开。

Alias not found

别名没有找到。在用SELECT命令来选取工作区时，如果不是用PRIMARY或SECONDARY符号来代表工作区；或者用到的“别名”事先没有定义，则出现该信息。

ALTERNATE Could not be opened

替换文件没有被打开。

Bad decimal width field

字段中小数位数有错误。在使用CREATE或MODIFY STRUCTURE命令时，如果字段中的小数位数给得不正确，如小数位数比字段的总宽度还要大或比字段宽度仅小1，这些都可能导致该种出错信息发生。

在dBASE II的各个不同版本中，发生这种出错信息的情况又有所不同：在2.3版本中，小数位数为零或大于254；在2.42版本中，小数位数大于63。不管哪种版本，均要求小数位数至少比字段宽度少2才是正确的。

Bad name field

字段名有误。有以下几种情况可能导致这种出错信息发生：

①在CREATE, MODIFY STRUCTURE命令中字段名已有定义；字段名中没有字母字符；字段名中字符个数多于10个；

②在TOTAL命令中使用的字段名不是源数据库文件中定义的字段名；

③UPDATE命令中给出的字段名有语法错，如字段名字拼写错了，字段名中多了字符，或者少了字符等。

Bad type field

字段类型有误。在CREATE或MODIFY STRUCTURE命令中定义的字段类型没有按要求指定为数值型(N)或字符型(C)或逻辑型(L)。

Bad width field

字段宽度有误。在CREATE或MODIFY STRUCTURE命令中，以下情况可能引起“字段宽度有误”。

①对字符型字段，宽度等于零或大于255；

②对数值型字段，宽度大于127；

③对逻辑型字段，宽度不为1。

Beginning of file encountered

遇到了文件的开始。在运行dBASE时，企图把记录指针定位在数据库文件的第一个记录之前（即0号记录），则可能发生这种出错信息。

***Beyond string

超出字符串。在选取子串函数\$中，子串的开始位置比整个字符串的宽度还大，即开始位置已经在字符串之外了，就可能发生该错误。

Cannot erase open file

不能删除打开的文件。不能用ERASE或DELETE FILE命令来删除一个打开的文件。

Cannot insert—There are no records in database file

不能插入——数据库文件中没有记录。dBASE I中的INSERT命令不能用于在一个空数据库文件（即只有文件结构没有任何记录的数据库文件）中插入记录，也不能用于在文件的末尾插入记录，而只能用于在已有记录之间插入一个新记录。如果需要在空文件或文件的末尾添加记录，可以用APPEND命令来完成。

Cannot open file

不能打开文件。在LOAD或RESTORE命令中不能打开指定的文件，则发生该出错信息。

Cannot select requested database

没有选取到需要的数据库文件。如果用SELECT命令选取的是PRIMARY和SECONDARY之外的工作区或数据库

文件，则产生该种出错提示。

Command file cannot be found

命令文件没有找到。在用DO命令来调用一个命令文件时，如果命令文件事先没有建立，或者命令文件名字拼写搞错了，或者命令文件中调用的其它命令文件发生了上述错误，都可能导致“命令文件没有找到”的出错信息。

Data item not found

数据项没有找到。在REPLACE命令中，缺少了替换短语或被替换的字段不在该文件结构中；或者将NOUPD-ATE单词误写成了NO和UPDATE两个字，均可能发生该出错信息。

Data type mismatch

数据类型不匹配。引起“数据类型不匹配”的原因有以下几种可能：

①在SORT命令中，排序字段是逻辑型(L)；

②在REPLACE命令中，替换表达式和对应字段的数据类型不一致；

③企图对不同类型的变量或表达式进行比较。

Database in use is not indexed

Database is not indexed

数据库文件没有被索引。在索引文件被打开之前，是不能使用FIND或SEEK命令的。

Directory is full

磁盘目录已满。当磁盘目录分配的存储空间已经占满时，还企图再打开或建立新文件，则可能出现该出错信息。

Disk is full

磁盘已满。企图对已满的磁盘再进行写操作，便从操作

系统返回一个出错代码。

Disk full when writing file→〈filename〉

Abort, Ignore or Delete old files(A, I, or D)?

当写〈文件名〉时磁盘已满

中断, 忽略或者删除旧文件(A, I或D)?

当工作盘已经装满而我们企图在磁盘上再建新文件时, 就可能发生该类出错信息。对于这种错误, 系统提供了三种处理办法:

①敲入字母A, 表示中断对磁盘的写操作。

②敲入字母I, 表示当前的操作无效。

③敲入字母D, 表示允许删除磁盘上用户认为无用的文件。当然, 我们也可以不删除这些文件而另换一个已格式化的空白磁盘作为用户的工作盘。

End of file found unexpectedly

End of file encountered

遇到文件结束。发生这种出错信息有以下几种原因:

①在USE, APPEND FROM, UPDATE FROM命令中, 企图打开一个不具备完整结构的数据文件;

②在dBASE II中, 读入522个字节之前发现了文件的物理结束标志;

③企图在索引文件以外或错误的索引文件上用FIND或SEEK命令进行检索操作;

④在dBASE III的REPORT, MODIFY REPORT命令中, 企图打开一个没有完整结构的报表格式文件;

⑤在dBASE III的SKIP命令中, 企图跳跃通过数据库文件的末尾;

⑥在dBASE II 2.3×和2.4版本中, 待建立的索引文件

的索引关键字段表达式超过100个字符时，出错信息变成：Too many characters（字符太多）。这和2.41，2.42版本中的同一出错信息一样了。

End of file or ERROR on keyboard input

文件结束或键盘输入有误。

Excess header lines lost

超过的标题行被丢失。

*** Execution error on +, Concatenated string too large

关于字符串运算符+的运行出错：连串的字符串太长。当对两个字符串用字符串运算符“+”进行连串运算时，如果运算的结果超过了254个字符，则出现该信息。

*** Execution error on LOG ():

关于自然对数函数LOG ()的运行出错。在调用LOG ()函数时，如果实在参数为零或负数，则发生该出错信息。

*** Execution error on sort:

有关排序 (SORT) 命令的执行有误。

*** Execution error on SPACE ():

关于取空格函数的运行出错。dBASE规定：函数的实在参数不能小于零或大于254。

*** Execution error on SQRT ():

关于求平方根函数的运行出错。SQRT ()函数的实在参数不能为负数。否则将出现该提示。

*** Execution error on STR ():

关于转换函数STR ()的运行出错。将一个数值型量转换成相应的字符型量时，如果函数中的长度参数大于19或小

数位超过了字符串长度减去1，则发生该种出错信息。

*** Execution error on SUBSTR () ; Start point out of range

关于求子串函数SUBSTR () 的运行出错：开始位置超界。在求子串函数SUBSTR (<变量名>, <开始位置>, <长度>) 中，如果给定的开始位置超出了变量值中字符的总个数，则会产生该种出错信息。

Expected ON or OFF

预置开 (ON) 或关 (OFF) 。

"FIELD" Phrase not found

"FIELD" 短语没有找到。在必须明确给出字段名的命令中，当需要FIELD短语时，语句行中却遗漏了"FIELD"，则发生该出错信息。

File already exists

文件已经存在。在RENAME命令中，TO之后的文件已经在磁盘上（即待重命名的文件与磁盘上文件同名，这是不允许的）；在CREATE命令中，欲建立的新文件与磁盘文件同名。这两种情况均导致“文件已经存在”的出错信息。

File does not exist

文件不存在。有下列几种情况导致“文件不存在”：

- ① APPEND FROM命令中使用的数据库文件没有被打开；
- ② DELETE FILE命令中欲被删去的文件没有找到；
- ③ ERASE命令中待删除的文件没有找到；
- ④ 在使用RENAME命令时旧文件没有找到；
- ⑤ 使用USE命令时不能把文件打开。

File is currently open

File is already open

文件已打开。造成这种出错信息可能有以下几种原因：

①在DELETE FILE命令中待删除的文件已经被打开（可能在别的地方打开了）；

②在dBASE的ERASE命令中，待删除文件已经被打开了；

③命令RENAME中欲重新命名的文件已在别处被打开；

④企图再打开一个已经在其它工作区内打开的数据库文件；

⑤企图在多个工作区内打开同一个数据库文件；

⑥企图复制（COPY）或排序（SORT）一个文件到它自身文件名中去；

⑦一个命令文件递归调用它自身。

File is not accessible

文件是不可用的。出现这种错误信息可能有以下几种原因，

①CREATE命令中使用的文件名包含了不允许的字符，如\`\= : + < > *`等；

②在使用MODIFY COMMAND命令时，磁盘目录已经装满，对新建文件或备份文件（.BAK）来说已经没有存储空间了；

③在COPY命令中，遗漏了目标文件名。

File too large...Some data may be lost

文件太大...某些数据有可能被丢失。在使用MODIFY COMMAND命令时打开了一个超过5000字节的文件，则引

起该提示信息。

FOR and WHILE cannot both be specified

在一个语句行中FOR和WHILE不能同时指定。

FORMAT file cannot be opened

格式文件没有打开。在SET FORMAT TO命令中没有打开指定的格式文件。

Illegal data type

非法的数据类型。

Illegal goto value

非法的转向值。在GOTO命令或任何带有RECORD短语的命令中，企图定位数据文件的指针到0号记录或65536号记录，均要发生该出错信息。

但是，在2.3D版本中，转向值为65537，在2.3B版本中，转向的记录号码为65537~65546；在2.4版本中，转向的记录号为65537，65538；在2.41版本中，转向值为65537~65565，就不显示以上信息。只要转向的记录号码大于以上指定的这些记录值，给出的信息为：Record out of range（记录号码超界）。

Illegal variable name

非法的变量名。在BROWSE, CHANGE, COPY, JOIN, RELEASE, REPLACE, SORT和其它可能出现变量名的地方，发生了语法错，如变量名事先没有定义，名字拼写有误，名字中夹杂了无效字符等，均可能导致该出错信息。

内存不足。下列情况可能引起内存不足。

①企图超过允许的GET(变量)的个数（系统默认值是128个。但可以在环境文件CONFIG.DB中修改GET默认的

Improper data type in subsubtotal expression

在子小计表达式中有不适当的数据类型。如象在 REPORT FORM命令的小计表达式中使用了逻辑型字段或备注型字段，就可能发生这种出错信息。

Index does not match database

Index file does not match database

索引文件与数据库文件不匹配。用SET INDEX TO或USE命令打开的索引文件与指定的数据库文件不匹配；存储在索引区内的关键字段表达式中的字段名同数据库文件中的字段名不匹配，都可能造成上述的出错信息。

Index file cannot be opened

索引文件没有打开。在SET INDEX TO和USE命令中指定的索引文件没有打开，或索引文件不在指定的驱动器上，都可能产生这种出错信息。

Index is too big (100 char maximum)

索引太大（至多100个字符）。在INDEX命令中给出的关键字段表达式超过了100个字符就会发生这种出错信息。对于同样性质的错误，在dBASE 2.3x和2.4版本中，出错信息为：End of file found unexpectedly（文件发生意外的结束）；在2.41和2.42版本中，出错信息是：Too many characters（字符太多）。

Insert impossible

插入是不可能的。

Insufficient memory

个数，其取值范围是35~1023）；

②在计算机内存不足（256K字节，汉字dBASE III 512K字节）的情况下，企图用RUN命令来运行一个外部程序或

用 MODIFY COMMAND 命令来建立一个外部字处理程序；

③Command.COM程序不在A驱动器上或用SET COMSPEC来载入Command.COM程序的情况下，企图运行一个外部程序或用MODIFY COMMAND命令来建立一个外部字处理程序。

Internal error——bucket overfilled

内部错误——存储桶已满。

Internal error in CREATE REPORT state machine

CREATE REPORT状态机器内部出错。

*** Interrupted ***

中断。当敲ESC键中断命令或程序的执行时，就会出现该提示。

Invalid date (hit SPACE)

无效的日期（敲空格键）。企图对日期型变量键入一个无效的日期值则发生该提示信息。系统要求敲空白键。

Invalid function argument

无效的函数自变量。函数的自变量数据类型有误或自变量的值超过了允许的范围都会发生该种出错信息。

Join attempted to generate more than 65534 records

Join命令产生的记录个数超过65534个。

Keys are not the same length

关键字段的长度不相同。在UPDATE命令中，用USE命令打开的数据文件中的关键字段与FROM之后所指数据文件中的关键字段长度不相同，因此发生该出错信息。

...keyword not found

关键字没有找到。

Line exceeds maximum of 254 characters

语句行超过了254个字符。

Macro is not a character string

宏替换 & 之后紧跟的不是字符串变量。

Memory variable file invalid

内存变量文件无效。RESTORE FROM命令中用到的内存变量文件事先必须用SAVE TO命令建好，否则发生“内存变量文件无效”的错误。

More than 5 fields to SUM

SUM命令中用于求和的字段超过5个(dBASE III中至多允许32个字段)。

More than 7 index files selected

选取的索引文件超过7个。在SET INDEX TO或USE命令中，允许同时打开至多7个索引文件。

No database file in use, enter filename:

No database is in use, enter filename:

非数据库文件在使用，键入文件名字。当数据库文件还没有打开以前，就直接使用当前工作区内的数据库文件，则发生该提示信息。

No expression to SUM

非表达式求和。在SUM命令中，不允许对不存在的字段求和。

No find

没找到。在使用FIND或SEEK命令时，关键字段表达式的值不在指定的索引文件中，则发生该提示信息。

No "FOR" phrase

非 "FOR" 短语。

No "FROM" phrase

在UPDATE命令中缺少了输入文件。

No room for heading

对于标题存储空间不够用。

Non-numeric expression

非数值表达式。

Not a character expression

非字符表达式。

Not a dBASE II database

Not a dBASE database

非dBASE数据库文件。在系统内部，dBASE对各种文件给出了一个内部标志，这个内部标志是用一个十六进制数来表示的。在dBASE II中，02代表数据库文件。在dBASE III中，83代表数据库备注文件，03代表数据库文件。

Not a logical expression

非逻辑表达式。

Not enough disk space for sort

对于排序操作没有足够的磁盘空间。

Not enough records to SORT

没有足够的记录用于排序 (SORT)。对于待排序的文件，要求至少有两个记录。

Numeric field overflow

Numeric overflow (data was lost)

数值溢出 (数据被丢失)。有下面两种原因：

①在REPLACE命令中，数值表达式的值超过了被替

换字段的宽度；

②在TOTAL命令中，目标文件的字段宽度不足以容纳合并细项后得到的值。

如果发生溢出，溢出项内用星号*填满，相应的数据被丢失。

“ON” phrase not found

“ON” 短语没有找到。

Operation with MEMO-type field invalid

对备注型字段进行操作无效。要求：

①表达式中不能出现备注型字段；

②在@...SAY...GET命令中不能出现备注型字段；

③在INDEX ON 命令的关键字段表达式中不能有备注型字段。

Out of memory for memory variables

Out of memory variable memory

内存变量占用的内存空间超界。系统约定：在dBASE I中，所有内存变量至多占用1536字节；dBASE II中所有内存变量至多占用6000字节(当然可以在环境文件CONFIG.DB中用MVARSIK参数来修改这个约定值)。

Out of memory variable slots

内存变量的存储槽超界。如果定义的内存变量个数超过256个，则发生该提示信息。

...Out of range

超界。

Patch is invalid

路径无效。

Record length exceeds maximum size (of 100)

记录长度超过最大值 (100)。

Record not in index

Record is not in index

记录不在索引文件中。GOTO或任何带RECORD 短语的命令定位的记录指针指向的不是索引文件中的记录。这种情况通常是由于修改了数据库文件而没有修改相应的索引文件而造成的。

Record not inserted

记录未被插入。

Record out of range

Record is out of range

记录号码超界。有下列几种情况造成记录号码超界：

①在EDIT或GOTO命令中指定的记录号码大于数据库文件的最大记录号码：

②带有RECORD子句的任何命令指定的记录号码超出数据库文件实有的记录范围；

③数据库只有一个文件结构没有任何记录，索引文件映射一个空数据库；

④数据库文件的记录总数不同于索引文件的记录数而企图对不在索引文件中的记录进行存取。

Report file invalid

报表文件无效。

SAY/GET position is off the SCREEN

SAY/GET 定位的屏幕超界。

Sorter internal error, notify SCDP

分类器内部出错，通知SCDP。

Source and destination data types are different

源数据和目标数据类型不一致。

***** Stack overflow *****

堆栈溢出。

String too large

字符串太长。

Structure invalid

结构无效。

***** Syntax error *****

Syntax error

语法有错。

Syntax error in contents

存储信息有语法错误。

Syntax error in field expression

字段表达式中有语法错误。

Syntax error in group expression

小计(分组)表达式中有语法错误。

Syntax error in subgroup expression

子小计表达式中有语法错误。

Syntax error in format specification

格式描述中有语法错误。

Syntax error, Re-enter

Syntax error, try again:

语法错, 再试:

***** System error:**

系统出错。

"TO" phrase not found

"TO" 短语没有找到。

Too many characters

字符太多。当dBASE II不具备全屏幕操作功能或屏幕置到关(OFF)状态时, APPEND、CHANGE、CREATE、EDIT、INSERT命令中用到的实际输入数据比字段的宽度还长。索引文件的关键字段表达式超过100个字符(2.41, 2.42版本), 均可引起“字符太多”的错误信息。

Too many file are open

打开文件太多。

Too many indices

打开的索引文件太多。

Too many memory variables

内存变量太多。

Too many merge steps

组合分级太多。

Too many sort key fields

排序关键字段太多。

Total label width exceeds maximum allowable size

标签的总宽度超过了允许的最大值。

...Truncated

截断。如果文件名、字段名、内存变量名超过了规定的长度, 则产生截断。

Unassigned file number

Unassigned file

没有指定文件号码。内部dBASE出错。在dBASE II中, 由于不适当的使用RESET命令, 或磁盘被严重地格式化或磨损, 或在dBASE 2.4x版本中, 执行HELP命令而没

有Dbasemsg.txt文件，均可能产生这种内部出错信息。

Unbalanced parenthesis

圆括号不配对。

Undetermined string

不完整的字符串。如引号不配对。

***Unknown Command

***Unrecognized Command verb

不可识别的命令动词。

Unknown function

不可识别的函数。

Unrecognized phrase/keyword in Command

命令中出现了不可识别的短语或关键字。

Unterminated string

未终止的字符串。

Variable Cannot be found

Variable not found

变量没有找到。

Warning-Report Form empty

警告-报表格式为空。

“With” phrase not found

“With” 短语没有找到。

***Zero Divide

Zero divide (result=0)

零驱动器(结果=0)

企图驱动一个由零或数值表达式来表示的驱动器。