

dBASE—III  
关系型数据库管理系统  
实用手册

(适用于微机0520、IBM-PC、IBM-PC/XT)

董西平 编 译  
戴之达



一九八四年十月

TP339.2  
1005

# dBASE—III

## 关系型数据库管理系统 实用手册

(适用于微机0520、IBM-PC、IBM-PC/XT)

董西平 编 译  
戴之达



30274928

一九八四年十月

551296

# 前 言

## 一、规定

### 1. 印刷约定

- 小写                    用户给出的信息；
- 大写                    dBASE II 命令的显示部分；
- [ ]                    dBASE II 命令的可选择部分；
- <>                    dBASE II 命令中由用户提供的部分；

### 2. 文件命名约定

文件名至多为 8 个字符，必须由字母开头，可由字母、数字和下划线组成。文件名内不允许嵌入空格。

字段、别名及存储器变量等命名的约定：

字段名、别名及存储器变量名至多为 10 个字符，必须由字母开头，可由字母、数字和下划线组成。字段名内不允许嵌入空格。

### 3. dBASE II 数据类型

- .DBF            数据库文件
- .DBT            备注字段文件
- .FMT            格式文件
- .FRM            报告型文件
- .LBL            标记型文件
- .MEM            存储器变量文件
- .NDX            索引文件
- .PRG            命令文件
- .TXT            文本输出文件

### 4. 全屏幕操作控制键

- | 键  | 等价键 | 功 能  |
|----|-----|--|
| ↑  | ^E  | 光标上移一行或移上一个字段；   |
| ↓  | ^X  | 光标下移一行或移下一个字段；   |
| ←  | ^S  | 光标左移一个空格。在菜单中移向左边一个选择项；                                      |
| →  | ^D  | 光标右移一个空格。在菜单中移向右边一个选择项；                                      |
| ^→ | ^B  | 执行 BROWSE 命令时，“镜头”向右摇一个字段；执行 MODIFY REPORT 命令时，将文件结构的显示上卷一次； |
| ↑  |     | 执行 MODIFY COMMAND 命令时，将光标移至本行末尾。                             |
| ^← | ^Z  | 执行 BROWSE 命令时，“镜头”向左摇一个字段；执行 MODIFY BEBORT 命令时，将文件结构的显示下卷一次； |
|    |     | 执行 MODIFY COMMAND 命令时，将光标移至本行开                               |

		头。
←		删除光标左边的一个字符；
Del	^G	删除光标下的一个字符；
End	^F	光标向右移一个字(WORD)；退出并存盘；
^End	^W	退出全屏幕操作并存盘；
Esc	^Q	退出、存盘但不改变原来盘中内容(即除执行 APPEND 和 BROWSE 命令时的当前记录外全部存盘)，并返回圆点提示符；
Home	^A	光标向左移一个字；
^Home		菜单选择项开关(适用于所有全屏幕操作命令)；
Ins	^V	INSERT 方式开关。打开时，在光标前面插入一个字符；关闭时，则“冲掉”光标下的字符；
^KW		执行 MODIFY COMMAND 命令时，将整个文件写到另一个文件；
^KB		执行 MODIFY COMMAND 命令时，将另一文件读入本文件；
^N		插入一个新行或一个新字段；
Pgup	^R	退至前一个记录或屏幕显示，执行 BROWSE 命令时，退至 1 一记录窗口；
^Pgup	^W	退出全屏幕操作并存盘；
pgDn	^C	移至下一个记录或屏幕显示，执行 BROWSE 命令时，移至下一个 1 一记录窗口；
^PgDn		进入并编辑备注字段；
<RETURN>		将光标移至下一字段或下一行。执行 APPEND 命令时，若光标处于一个空记录的第一个字符，打此命令存盘并退出；执行 MODIFY COMMAND 命令时，若 INSERT 开关已打开，打此命令插入一行；在菜单中打此命令作出一个选择；
^T		删除光标右面的一个字；
^U		执行 BROWSE 或 EDIT 命令时，给记录加删除标记；执行 MODIFY REPORT 或 MODIFY STRUCTURE 命令时，删去一个字段定义；
^Y		执行 MODIFY COMMAND 命令时，从当前位置开始至字段末全部删除或删除一个整行；

#### 5. 用于全屏幕操作时的控制字符

←	^H	光标回退时抹除内容；
<RETURN>	^M	与上述 <RETURN> 键功能相同；
^P		打印机开关；
^S		停止屏幕继续上卷或命令的执行；
^X		抹除命令行；

## 6. 命令分类

通常在使用dBASE III时,为完成一个特定的任务需结合使用各种命令。部分dBASE III的命令是仿照多数计算机高级语言所采用的结构化构造的。这些命令属于命令文件类。这些命令的使用应遵循专门的规则。

## 7. 文件生成

下列命令生成数据库文件及有关文件:

CREATE COPY                    COPY FILE                    INDEX  
JOIN            MODIFY LABEL    MODIFY REPORT    SAVE  
SORT            TOTAL

## 8. 添加数据

下列命令向数据库加入新的数据记录:

APPEND                    BROWSE                    INSERT

## 9. 编辑数据

下列命令对数据库中的命令进行编辑:

BROWSE            CHANGE            DELETE            EDIT  
PACK              READ              RECALL            REPLACE  
UPDATE            ZAP

## 10. 用户辅助

下列命令在联机条件下向用户提供信息:

ASSIST            DIR                    DISPLAY MEMORY  
DISPLAY STATUS    DISPLAY STRUCTURE    HELP

## 11. 显示数据

下列命令显示从数据库选择的数据,这些命令与SET TALK ON/OFF的状态无关:

@..SAY            AVERAGE            BROWSE  
COUNT            DISPLAY LIST            REPORT  
SUM                TEXT

## 12. 记录指针的定位

下列命令将当前记录指针定位于所指向的记录:

CONTINUE            FIND                    GOTO                    LOCATE  
SEEK                SKIP

## 13. 数据库操作

下列命令对整个数据库文件起作用:

APPEND FROM        CLOSE                    COPY  
REASE              MODIFY STRUCTURE    QUIT                    INDEX  
REINDEX            RENAME                    SELECT                    SORT  
USE

## 14. 操作其它类型的文件

下列命令对非数据库文件起作用:

MODIFY COMMAND    MODIFY LABEL    MODIFY REPORT    RUN

15. 使用存储器变量

下列命令对存储器变量进行操作:

ACCEPT      AVERAGE      CLEAR ALL      CLEAR GETS  
 CLEAR MEMORY COUNT      DISPLAY MEMORY      INPUT  
 PRIVATE      PUBLIC      READ      RELEASE  
 RESTORE      SAVE      STORE      SUM  
 WAIT

16. 编程

下列命令用于程序文件的控制和处理:

ACCEPT      CANCEL      DO      DO CASE  
 DO WHILE      IF      INPUT      MODIFY COMMAND  
 NOTE \*      PARAMETERS      QUIT      RETURN  
 RUN      TEXT      WAIT

17. 环境命令

下列命令控制dBASE II的环境和外部设备:

EJECT      CLEAR      SET

18. 参数控制

下列命令可被SET ON或SET OFF, 或SET TO某一数据。所有命令必须从SET开头:

ALTERNATE      BELL      CARRY      COLOR  
 CONFIRM      CONSOLE      DEBUG      DECIMALS  
 DEFAULT      DELETED      DELIMITER      DEVICE  
 ECHO      ESCAPE      EXACT      FILTER  
 FIXED      FORMAT      FUNCTION      HEADING  
 HELP      INDEX      INTENSITY      MARGIN  
 MENUS      PATH      PRINT      PROCEDURE  
 RELATION      SAFETY      STEP  
 UNIQUE

19. 运算符

(1) 算术运算符

按优先级次序排列:

( )      成组运算  
 ..      求幂  
 .      乘  
 /      除  
 +      加  
 -      减

(2) 关系运算符

<      小于  
 >      大于

- = 等于
- <> 或 # 不等于
- <= 小于等于
- >= 大于等于

### (3) 逻辑运算符

按优先级次序排列：

- .NOT. 逻辑非(一元算符)
- .AND. 逻辑与
- .OR. 逻辑或
- 字符串运算符
- + 字符串连接
- \$ 子字符串搜索

如果同一表达式中使用了一种以上的运算符，则优先级的次序为：数学的、关系的，最后是逻辑的。

同一优先级水平的运算按自左向右的次序执行。可用括号来改变所执行运算的次序。

### 20. 术语

- <alias> 替代文件名；
- <condition> 逻辑表达式；
- <cstring> 字符串；
- <dec> 小数点后位数；
- <delimiter> 单引号、双引号或方括号
- <exp> 字段、存储器变量、函数、字符串及它们的组合；
- <explist> 由逗号隔开的表达式清单；
- <expc> 字符表达式；
- <expD> 数据表达式；
- <expN> 数值表达式；
- <field> 字段名；
- <fieldlist> 由逗号隔开的字段名清单；
- <file> 文件名；
- <index file> 索引文件名；
- <key> 数据库文件中用于生成索引文件的部分；
- <memvar> 存储器变量名；
- <memvar list> 由逗号分隔开的存储器变量；
- <n> 一个数字；
- <procedure> 类似于子程序的一个命令文件；
- <scope> ALL、NEXTn和RECORDn；
- <skeleton> 用 \* 或 ? 来替代名字中的一部分；

### 21. 函数

- & <memvar> 宏替换；

ASC()	该字符的ASCII码;
AT(<expc1>、<expc2>)	子字符串搜索;
BOF()	指示文件头;
CDOW(<expD>)	求星期几;
CHR()	ASCII码变换为字符;
CMONTH(<expD>)	给出日期变量的月份;
COL()	指示光标的当前列位置;
CTOD(<expC>)	将字符串变换为日期变量;
DATE()	给出系统日期;
DAY(<expD>)	给出日期变量的日历值;
DELETED()	识别记录是否打上删除标志;
DOW(<expD>)	给出日历值所对应的数值;
DTOC(<expD>)	将日期变量变换为字符变量;
EOF()	指示文件尾;
EXP(<expN>)	指数函数
FILE("<filename>")	指示该文件存在否
INT(<expN>)	求整数函数
LEN(expC>)	求字符串长度;
LOG(<expN>)	求自然对数函数;
LOWER(<expC>)	大写变换为小写;
MONTH(<expD>)	给出日期变量的月历值;
PCOL()	确定打印机上的当前列位置;
PROW()	确定打印机上的当前行位置;
RECNO()	给出当前记录号;
ROUND(<expN>,<dec>)	四舍五入;
SPACE(<expN>)	确定光标的当前行位置;
SQRT(<expN>)	求平方根函数;
STR(<expN>[,<length>][,<dec>])	将数字表达式变换为字符串;
SUBSTR(<expC>,<start>[,<length>])	子字符串选择;
TIME()	给出系统时间;
TRIM(<expC>)	压缩字符串尾部空格;
TYPE(<expC>)	求表达式数据类型;
UPPER(expC>)	小写变换为大写;
VAL(expC>)	字符变换为数值;
YEAR(expD)	给出日期变量的年历值。

## 22. dBASE III命令

### ——命令的语法

? <exp>

在下一行显示该表达式;

? ? (exp)

在当前行显示该表达式；

@ <row,col> [SAY(exp)[PICTURE(clause)]]  
[GET(variable)[PICTURE(clause)]]  
[RANGE(exp,exp)][CLEAR]

SAY语句将用户的格式语句在显示器或打印机上显示。GET语句将用户的格式语句在显示器上显示用于编辑；

ACCEPT[<PromPt>]TO (memvar)

将一个字符串存入一个存储器变量；

APPEND[BLANK]

在数据库文件末端追加记录；

APPEND FROM <file name> [FOR <exp>] [SDF] [DELIMITED]

从其他文件向当前打开的数据文件追加记录；

ASSIST

帮助执行dBASE III命令，由菜单驱动；

AVERAGE <exp list> [<scope>] [FOR/WHILE <exp>]  
[TO <memvar list>]

计算数值表达式的算术平均值；

BROWSE[FIELDS <field list>]

使用全屏幕窗口显示进行编辑，每屏可达17个记录，由菜单提供帮助；

CANCEL

中止程序执行，返回园点提示符；

CHANGE [<scope>] [FIELDS <field list>] [FOR/WHILE <exp>]

在数据库中编辑指定的字段和记录；

CLEAR

清屏；

CLEAR ALL

关闭所有数据库文件、索引文件、格式文件和有关的文件。同时释放所有的存储器变量并选择工作区 1；

CLEAR GETS

释放由READ语句读数的当前GET变量；

CLEAR MEMORY

清当前存储器变量；

CLOSE[ALTERNATE/DATABASES/FORMAT/INDEX/PROCEDURE]

关闭指定类型的文件；

CONTINUE

将记录号定位于满足由LOCATE命令指定的条件的下一个记录上；

COPY FILE <file name> TO <file name>

复制任意类型的文件；

**COPY TO**<file name>[<scope>][**FIELDS**<field list>]

[**FOR/WHILE**<exp>][**SDF/DELIMITED**[**WITH**<deliter>]]/

将打开的数据库复制到另一个数据库或一个文本文件中（后一种情况需选择**SDF**或**DELIMITED**）；

**COPY STRUCTURE TO**<file>[**FIELDS**<field list>]

将打开的文件之结构复制到新的数据库；

**COUNT**[<scope>] [**FOR/WHILE**<exp>] [**TO**<memvar>]

对指定范围内的记录进行计数；

**CREATE**<.dbf file name>

定义新数据库文件的结构并将该文件加入目录；

**CREATE LABEL**<.lbl file name>

生成一个标号格式文件，由菜单驱动；

**CREATE REPORT**<.frm file name>

生成一个报告格式文件，由菜单驱动；

**DELFTE**[<scope>] [**FOR/WHILE**<exp>]

对指定的记录作删除标记；

**DIR**[<drive:>] [<Path>] [<skeleton>]

显示指定磁盘驱动器上的文件名；

**DISPLAY**[<scope>] [**FIELDS**<field list>] [**FOR/WHILE**

<exp>] [**OFF**]

显示现用数据库中的记录和字段；

**DISPLAY MEMORY**

显示当前存储器变量；

**DISPLAY STATUS**

显示有关现用数据库、索引文件、交替文件及系统参数方面的信息；

**DISPLAY STRUCTURE**

显示打开数据库文件的结构；

**DO**<.prg file name>/<procedure name> [**WITH**<parameter list>]

执行一个程序或过程，可以选择地向该程序传递参数；

**DO CASE..CASE..**[**OTHERWISE**]...**ENDCASE**

挑选几条可能路径中的一个执行，可以选择地带有一备择项。

**DO CASE**必须以**ENDCASE**结尾；

**DO WHILE..**<commands>..**ENDDO**

在程序中构成一个结构化的回路。使用**LOOP**后则跳过该语句与**ENDDO**之间的所有命令。**DO WHILE**必须以**ENDDO**结尾；

**EDIT**[**RECORD**]<n>]

改变数据库字段的内容；

**EJECT**

命打印机换页；

**ERASE**<file name>

从目录中删除指定的文件;

**EXIT**

退出 DO 回路, 但不终止程序的执行;

**FIND**<character string>

将记录指针定位于其索引关键字与所指定字符串相匹配的第一个记录;

**GO/GOTO BOTTOM/TOP/<expN>**

直接将记录指针定位于所指定的记录;

**HELP**[<keyword>]

解释dBASE III命令并提供其他信息, 由菜单驱动;

**IF..[ELSE]..ENDIF**

允许按条件执行程序中的命令, 可以选择地带有备择路径(ELSE)。每一个IF必须以一个ENDIF结尾;

**INDEX ON<exp>TO<.nds file name>**

使得相关联的数据库按照一个指定的关键字进行排序;

**INPUT**[<prompt>]TO<memvar>

将接收的输入表达式项送入存储器变量;

**INSERT**[BLANK] [BEFORE]

将一个记录放在数据库文件中指定的位置;

**JOIN WITH<alias>TO<new file>FOR<condition>**

[FIELDS<field list>]

组合来自两个数据库文件的指定的记录和字段;

**LABEL FORM<.lbl file name>**[SAMPLF] [<scope>]

[FOR/WHILE<condition>]

[TO PRINT] [TO FILE<file name>]

使用指定的标号格式文件打印标号;

**LIST**[<scope>] [FOR/WHILE<condition>]

[FIELDS<field list>]

[OFF]

将数据库记录和字段列表;

**LOCATE**[<scope>]FOR<condition>

将记录指针定位于满足指定条件的记录;

**LOOP**

跳过该语句与ENDDO之间的所有命令;

**MODIFY COMMAND**<file name>

文字处理程序, 可用来编辑包括程序文件(.prg)在内的ASCII文本文件;

**MODIFY LABEL**<.lbl file name>

生成并编辑标号格式文件, 由菜单驱动;

**MODIFY REPORT**<.frm file name>

生成并编辑报告格式文件，由菜单驱动；

**MODIFY STRUCTURE**

改变打开数据库的结构；

**NOTE/\* <undelimited character string>**

在程序文件中插入非执行的注解；

**PACK**

永久性地删除数据库中已打有删除标记的记录；

**PARAMETERS<Parameter list>**

规定存储器变量使用由DO...WITH命令传送的资料；

**PRIVATE**[ALL[LIKE/EXCEPT<skeleton>]]

[<memory variable list>]

将存储器变量隐入高级语言程序；

**PUBLIC**<memory variable list>

使存储器变量全局化；

**QUIT**

关闭所有文件并退出dBASE III

**READ**

接收输入数据，送至GET字段或变量；

**RECALL**[<scope>][FOR/WHILE<condition>]

恢复带有删除标记的记录；

**REINDEX**

对已存在的现用索引文件重新定义；

**RELEASE**[<memvar list>]

[ALL[LIKE/EXCEPT<skeleton>]]

抹除当前存储器变量；

**RENAME**<current file name>TO<new file name>

赋给文件新名字；

**REPLACE**[<scope>]<field>WITH<exp>

[,<field2>WITH<exp2>,...]

[FOR/WHILE<condition>]

以指定的值替换数字字段的内容；

**REPORT FORM**<.frm file name>[<scope>]

[FOR<exp>][PLAIN]

[HEADING<character string>]

[NOEJECT][TO PRINT]

[TO FILE<file name>]

显示数据的报告表格；

**RESTORE FROM**<.mem file name>[ADDITIVE]

恢复存入的存储器变量；

RETURN[TO MASTER]

程序结束。这条语句是程序的最后一个可执行语句；

SAVE TO<.mem file name>[ALL LIKE/EXCEPT  
<skeleton>]

将当前存储器变量复制到存储器文件；

RUN<command>

执行dBASE III外的程序；

SEEK<expression>

将记录指针定位于其索引关键字与指定的表达式相匹配的第一个记录；

SELECT<n/alias>

十个工作区的选择开关；

SET

置dBASE III的控制参数，由菜单驱动；

SET ALTERNATE TO[<file name>]

生成一个用于保存输出的文件；

SET ALTERNATE on/OFF

输出送(或不送)给一个文件；

SET BELL ON/off

数据输入时响铃(或不响铃)；

SET CARRY on/OFF

将最后一个记录的内容写(或不写)到追加的记录中；

SET COLOR TO<standard display>

[,<enhanced display>][,<border>]

设置屏幕显示的属性；

SET CONFIRM on/OFF

全屏幕方式下能够(或不能够)自动跳到下一字段；

SET CONSOLE ON/off

将输出送(或不送)至屏幕；

SET DEBUG on/OFF

将ECHO的输出送(或不送)至打印机；

SET DECIMALS TO<n>

对某些运算和函数的结果之显示设置小数点后位数的最小值；

SET DEFAULT TO<drive>

为文件搜寻指定当前驱动器；

SET DELETED ON/OFF

“复盖”(或处理)有删除标记的记录；

SET DELIMITER TO[<character string>]

[DEFAULT]

为字段和变量的全屏幕显示指定分隔符；

**SET DELIMITER on/OFF**

全屏幕显示字段和变量时，以正常显示(反相显示)定界；

**SET DEVICE TO SCREEN/Print**

将@...SAY命令的结果送至屏幕(打印机)；

**SET ECHO on/OFF**

可在(或不可在)屏幕或打印机上得到命令行的回波；

**SET ESCAPE ON/off**

按ESC键后终止(或继续)命令文件的执行；

**SET EXACT ON/off**

字符比较时要求(或不要求)准确的匹配；

**SET FILTER TO<condition>**

使得数据库文件看起来就象它仅含有满足指定条件的那些记录；

**SET FIXED on/OFF**

固定(或不固定)显示值的小数点后位数；

**SET FUNCTION<key#>TO<character string>**

设置功能键的值；

**SET FORMAT TO<.fmt file name>**

为数据输入打开一个格式文件；

**SET HEADING on/OFF**

执行LIST或DISPLAY命令时，在字段内容上显示(或不显示)字段名；

**SET HELP ON/off**

当出错时用户要求(或不要求)帮助性提示；

**SET INDEX TO<.ndx file list>**

打开被指名的索引文件；

**SET INTENSITY ON/off**

全屏幕操作时采用(或不采用)反相显示；

**SET MARGIN TO<n>**

设置打印机的左部边界；

**SET MENUS ON/off**

全屏幕操作时显示(或不显示)菜单；

**SET PATH TO[<path list>]**

为文件搜寻指定一条路径；

**SET PRINT on/OFF**

将输出送(或不送)到打印机；

**SET PROCEDURE TO<procedure file name>**

打开被指名的过程文件；

**SET RELATION TO<key>INTO<alias>**

按照关键字表达式联接两个数据库；

**SET SAFETY ON/off**

设置防止文件被重写保护的级别；

SET STEP on/OFF

每条命令处理完后暂停(或不暂停)程序的执行；

SET TALK ON/off

将命令执行的结果送(或不送)到屏幕；

SET UNIQUE on/OFF

使用索引文件中第一个(或全部)带有相同关键字的记录；

SKIP<n>

使当前记录指针相对于它的当前位置前进或后退；

SORT TO<new file name>ON<field>

[/A][/D][,<field2>[/A][/D…]

[<scope>][FOR<condition>]

根据数据库生成一个排了序的拷贝，其中的数据按照某一数据字段排列；

STORE<exp>TO<memvar>[,<memvar list>]

将一个表达式存入一个或多个存储器变量；

SUM[<scope>][<expression list>TO<memvar list>

[FOR/WHILE<condition>]

计算并显示由指定范围内的数据库数据组成的表达式之总和；

TEXT..ENDTEXT

显示取自某一命令文件的文本数据块，TEXT必须以ENDTEXT结尾；

TOTAL TO<file name>ON<key>[<scope>]

[FIELDS<field list>]

[FOR/WHILE<condition>]

根据预先已排序的文件生成一个总计数据库，其中包括数值总和；

UPDATE[**RANDOM**]ON<key field>

FROM<alias>

REPLACE<field>WITH<exp>

[,<field2>WITH<exp2>…]

实现对数据库的成批修改；

USE[<.dbf file name>]

[INDEX<.ndx file list>][ALIAS<alias>]

在下一个USE命令发出前，本USE命令指定的数据库文件用于各种操作；

WAIT[<prompt>][TO<memvar>]

暂停程序处理，直到按下任意键；

ZAP

从现用数据库文件抹除全部记录；

## 二、说 明

### 系 统 要 求

dBASE II 需要如下的硬件和软件环境

- IBM-PC, IBM-PCXT, 或任何其它100%的IBM-PC兼容机;
- 最少要求256K字节的存储器;
- MS-DOS或PC-DOS(2.0以上的版本)操作系统;
- 二台360K软磁盘驱动器或一台360K软磁盘驱动器加一台硬磁盘驱动器;
- 至少80列的任意型号打印机。

### 关 于 主 盘

dBASE II 系统盘采用名为PROLOK 的软件保护系统实现了反拷贝。PROLOK并不阻止磁盘被拷贝,但拷贝得到的盘是无用的,除非原始主盘在某一个软盘驱动器中。

带 PROLOK 盘上的文件应复制到另外的磁盘上作为后备,但在运行这些程序之前,必须从后备盘上将这些文件重新拷贝到原始的带PROLOK盘上。对于硬盘系统,带 PROLOK 盘必须使用,但用法是不同的。

### 重 要 事 项

1. 绝不要使用DISKCOPY命令将文件拷贝到dBASE II 主盘上,只能使用 COPY 命令;
2. 绝不要 FORMAT dBASE II 系统主盘。  
违反上述两项将造成系统主盘的永久性损坏。主盘损坏后,只能再购买一块新主盘。另外,原始盘使用时不可以是写保护状态。

### 制 作 备 份

要制作全盘的备份:

1. 使用DOS系统盘上的FORMAT 命令格式化一块磁盘;
2. 将dBASE II 主盘插入A驱动器,格式化后的盘插入驱动器B,打入:

**COPYA: \*.\* B:**

如果主盘上的文件已受损或由于疏忽而抹除了,可使用DOSCOPY命令将此文件从备份盘拷回主盘。

如果试图以复制盘来运行dBASE II,会得到如下的出错信息“Unanchored Duplicate (非法复制)”,然后控制返回操作系统。

### 硬 盘 系 统

在硬盘系统上使用dBASE II:

1. 使用DOS COPY命令将整个主盘拷贝到硬盘。设dBASE II主盘在A驱动器中，硬盘命名为C，则打入：

COPYA: \*.\* C:

2. 此时只要程序执行过程中原始主盘在某个软盘驱动器中，便可从硬盘中运行dBASE II。

如果试图在软盘驱动器无主盘的情况下从硬盘中运行dBASE II，亦会得到出错信息“Unauthorized Duplicate”，然后控制返回操作系统。

### dBASE II 主要指标

关系型数据库管理系统；

可用于交互式运行或应用程序开发，具有完整的关系运算及可编程特性；

使用方便，具有“自动运行”方式，明了的help提示，类英语命令；

每个记录可有128个字段，可容纳4000个字符；

新型的备注(MEMO)数据类型使记录可达500,000字符长；

每个文件可容纳10亿个以上的记录；

同一时刻可使用10个数据文件；

一个数据库中的文件个数没有限制；

极快的多重字段排序；

对索引过的文件可进行极快的查找；

完整的应用程序开发语言，具有过程调用、参数传送及更多的特性；

运行于IBMPC及其兼容机上。

### 建立系统备份

对任何计算机而言，制作备份盘片都是一种很好的数据处理方法。应当经常地制作备份盘片，以防电源掉电或偶然事故一下子毁了几个月来积累的数据。

### 在IBM-PC上运行dBASE II

IBM-PC机上已完成了dBASE II的系统建立，并保留了其功能键的多数优点。

每当敲入了一条dBASE II指令或输入了数据，须按ENTER键以通知计算机，本文以C表示。

你的微机有一个“超级字型转换”(Supershift)键，其功能类似于打字机上的字型转换(SHIFT)键，所不同的是，它不是用来产生大写字母或某些符号的，而是使你能实现计算机的另外一些不同的功能。

这种“超级字型转换键”即是Ctrl(控制)键，其使用方法与SHIFT键一样，即：按住该键的同时，按键盘上的另一个键。

每当你应使用Ctrl键时，本文以符号“^”来表示，如^W表示(control-W)。首先压下Ctrl键，同时压下字母W。

本书中使用符号<>表示用户应给出的信息。例如，书中给出<file name>，用户便