

第一章 dBASE III PLUS 运行须知

第一章主要介绍运行 dBASE III PLUS 所需的软硬件环境以及在安装过程中如何解决所遇到的问题。

dBASE III PLUS 运行环境

Ashton-Tate 公司曾尽全力使尽可能多的 PC 机及其兼容机的用户使用 dBASE III PLUS,其目的是用户不必在最新功能最强的计算机上使用 dBASE III PLUS。它对硬软件环境的最低要求是:有两台磁盘驱动器,带一台监视器,内存为 256K 的 PC 机或其兼容机,PC-DOS 2.0 或 MS-DOS 2.1 或以上版本的操作系统。用 DOS 3.0 或以上版本的操作系统运行 dBASE III PLUS 时,内存必须在 384K 以上。

要想更充分地利用 dBASE III PLUS,需有一个 80 列的打印机和附加的内存,尽管不需要 100 兆字节的内存或 200 兆字节的 CDROM,但采用一个硬盘会使运行速度加快,扩大了内存将使 dBASE III PLUS 本身的机能更加灵活自如。当然,多于 80 列的打印机也是可以用的。

dBASE 系统本身能检查是否有打印机,可以使用任何能与 PC 机连接上的监视器(彩显或单显),而且系统能使用多少内存和驱动器,dBASE 就能使用多少。

dBASE 系统几乎是能够自行安装的,需要安装者做的仅是回答一些简单的问题,主要是告诉 III PLUS 磁盘的系列号和

输入一个公司名(如果有的话)。

排除故障的方法

安装过程中会偶尔出现一些问题,dBASE 如果检测到了就会进行提示。如果在已有 dBASE III 或 III VAR(程序开发人员发行的版本)的硬盘目录或子目录上再安装 III PLUS,就会出现这个问题。这种情况的最好解决办法是:“卸出”dBASE III 的早期版本(这点不适于盘上已有 dBASE II 的情况)。在一个硬盘或软盘系统中安装 dBASE 的最好参考书是文献中提到的“Getting Started”一书。

如果在安装完之前(回到 DOS 提示符之前)出现了故障,那么只好热启动(同时按 CTRL,ALT,DEL 键)或关掉机器。版本 1 的用户注意:这不会导致计数错误,因为减少安装计数是在 SUPERLOK 安装过程中最后一步要做的事情。SUPERLOK 是 Ashton-Tate 用来在 dBASE III PLUS 系统中 1 号盘上进行拷贝保护的方法。不过,在 SUPERLOK 过程结束前停止安装过程会吃掉硬盘的一些可用空间。版本 1.1 的用户不必担心 SUPERLOK。

如果在使用一个硬盘系统时出现了故障,首先应该确信 dBASE 能在用户自己的机器上运行,在 dBASE III PLUS 软件包“Getting Started”一书上可以找到兼容的硬盘,所有 Kaypro 16 和 PC 机都能加到那张清单上(实际上写这本书时已使用过它们了)。对版本 1 的用户:在完成安装前 SUPERLOK 应停止,并执行 DOS CHKDSK 应用程序,这样才能清除磁盘并收回被“未完成文件”所占用的空间。做这一点时可先回到 DOS 提示符并键入以下内容来完成:

CHKDSK /F [RETURN]

这时假定 **DOS CHKDSK.COM** 实用程序在硬盘是可用的。如果 **CHKDSK.COM** 文件不在硬盘上,那么它应在 **PC/MS-DOS** 的引导盘上。

DOS 会提示出是否需要恢复丢失的簇,应回答:

Y[RETURN]

然后 **CHKDSK** 把丢失的链恢复到文件中,并把这些文件聚集起来,再以 **FILEnnnn.CHK** 文件名的形式写在盘上。这些文件能用 **ERASE** 或 **DELETE** 命令删除而使盘空间可用。可以用下述方式删除文件:

DEL FILE * .CHK[RETURN]

这样的结果是使硬盘有更多的自由空间,这个过程不损坏任何合法文件。

这些问题不会出现在软盘系统中,软盘系统具有一定的先进性。**CHKDSK** 清单列表如图 1.1 所示。

如果出现了故障并使安装停止了,只需再试装一次 **dBASE** 即可。有一些原因使安装停止,但并不是所有的都那么严重。如果不能在系统中安装 **III PLUS**,可在引导盘上试运行 **dBASE**。

如果不能从引导盘上运行 **dBASE**,就把软件包送回销售商处,请求在销售商的一台机器上检测 **dBASE**(最好在会出现同样问题的机器上)。如果也不能装在那台机器上,试着把另一软件包装在这台机器上,若再失败,**dBASE** 就不能在所使用的样机上运行。若第二个软件包能工作,问题就在第一个软件包上,这时可带上新软件包回去,再尝试一次。

如果原来的软件包在销售商的机器上可以运行,则可能

```
C:\>chkdsk

10584064 bytes total disk space
 307200 bytes in 6 hidden files
 49152 bytes in 11 directories
4943872 bytes in 302 user files
 36864 bytes in bad sectors
5246976 bytes available on disk

524288 bytes total memory
298896 bytes free
```

```
C:\>
```

图 1.1 DOS CHKDSK 应用程序通过发现
没关闭的文件来清理硬盘和软盘

是用户自己的软盘驱动器需清洗或整修了。把它带到销售处并调试好。未校准的磁头会损坏数据和其它文件,而且对便携式计算机来说,磁头是要经常校准的。

如果还不成功,请直接与 Ashton-Tate 联系,或找代理商联系。

第二章 dBASE III PLUS 的文件类型

同 dBASE II 和 dBASE III 的早期版本一样, dBASE III PLUS 利用文件的扩展名来分辨文件的类型。扩展名在圆点(.)的后面,是文件名的一部分,计算机所用的文件名几乎都用以下形式

FILENAME. EXT

文件名可由用户自定,可以取名为 **ACCOUNTS**, **RECEIVAB** 或 **INCOME**。大多数程序允许用户给出文件名,以便今后再用时快速找到这个文件。

文件的扩展名往往由用户使用的程序确定,如确定数据库、字处理、画面显示文件的扩展名为“**FW2**”,确定“**TCM**”为电信文件。

dBASE 程序与多数其它程序一样,需要给出每个文件的扩展名(这将在第三章详细介绍)。

人们已经熟悉的 dBASE II 和 dBASE III 的不同版本都有如下 dBASE 文件类型:

DBF——数据库文件

DBT——数据库正文文件(即备注文件,为 dBASE III 独有)

FRM——报表格式文件

FMT——格式文件

LBL——标签格式文件

MEM——内存变量文件

NDX——索引文件

PRG——程序文件(也用作过程文件)

TXT——正文输出文件

除这些外,dBASE III PLUS 附加以下文件类型:

CAT——编目文件

QRY——查询文件

SCR——屏幕格式文件

VUE——视图文件

dBASE III PLUS 不强制使用缺省的文件扩展名。关于缺省,最容易的办法是给出特定文件的扩展名。不过在今后的工作中调用文件时必须用扩展名,如:

CREATE TEMP.CAT

令 **III PLUS** 生成一个新的数据库,通常是 **a.DBF** 文件,并称之为“**TEMP.CAT**”。这意味着用户正在以 **a.CAT** 扩展名的方式使用一个 **DBF** 文件,扩展名 **CAT** 通常是被编目文件使用的。

在定义 **SAY**、**GET**、**PRG**、**FMT** 之类的文件时,这种能力是非常有用的。所有这些文件都是包含各种 **III PLUS** 命令的标准 **ASCII** 码文件,只不过各自提供不同的服务而已。文件 **SAY** 描绘屏幕,**GET** 文件把用户信息送入系统或把系统信息送给用户,**PRG** 文件支持 **III PLUS** 命令与系统一起工作,**FMT** 文件在用命令 **<SET FORMAT TO filename>** 产生和被 **APPEND** 或 **INSERT** 使用时与 **GET** 和 **SAY** 有同样的功能,用户可以根据实际用途使用所有这些文件。

在生成一个图书管理系统时,可以使用 **LIBRARY.PRG**,

LIBRARY. FMT, LIBRARY. DBT, LIBRARY. N1, LIBRARY. N2, LIBRARY. GET, LIBRARY. SAY, LIBRARY. TMP 和 **LIBRARY. FRM** 等文件, 用户实际上可使用三个程序文件 (**PRG, SAY** 和 **GET**)、两个数据库文件 (**DBF** 和 **TMP**) 和两个索引文件 (**N1** 和 **N2**)。使用不缺省的扩展名可以更自由地根据各自最初的用途分组管理文件。在本例中是图书馆。

本章是根据不同类型文件分节的, 每一节说明一种类型的文件以及这些文件类型在 **dBASE III PLUS** 中如何工作、如何生成以及使用时的约束条件。

文件名. CAT——编目文件 (CATalog File)

编目文件实际上相当于一个具有特殊用途和特殊扩展名的数据库文件。用户可以自己生成一个与它具有相同功能的文件 (CAT 文件的结构见图 2.1)。实际上, 由于许多用户都需要建立这种文件, 所以 **Ashton-Tate** 公司决定把它作为系统的一部分。

什么是编目文件呢? 编目文件实际上是一种文件的过滤器, **III PLUS** 用它屏蔽掉所有与给定的应用无关的文件。

假定用户想开发一个用于随时跟踪作者所著的文章或书籍的系统, 磁盘上有几个不同的数据库文件和索引文件, 但用户感兴趣的只是 **MAGAZINE, BOOK, INCOME** 和 **EXPENSE** 数据库, 以及 **MAGAZINE** 和 **MAUKS, BOOK** 和 **BAUDS, RFROM** 和 **RFOR, PTO** 和 **PFOR** 等索引文件。

用户知道磁盘上已有许多不同的数据库和索引文件, 但要想办法把它们筛选出来, 因为工作于 **MAGAZINE** 文件和 **BOOK** 文件时, 用户并不想知道其它 **DBF** 和 **NDX** 文件存在与

```

-> DISP STRUC
Structure for database: D:\WRITE.cat
Number of data records: 0
Date of last update : 08/15/86
Field  Field Name  Type      Width  Dec
1  PATH      Character  70
2  FILE_NAME Character  12
3  ALIAS     Character  8
4  TYPE      Character  3
5  TITLE     Character  80
6  CODE      Numeric   3
7  TAG       Character  4
** Total **           181
->

```

图 2.1 目录文件给出与特殊数据库文件有关的信息

否。

这可以用命令 **SET CATALOG TO WRITE** 启动 **III PLUS**, 显示方式见图 2.2。

III PLUS 会生成一个名为 **WRITE.CAT** 的新 **CAT** 文件, 如图 2.2 所示。下一步给出 **III PLUS** 所需的文件。这样, 无论什么时候需用这组文件, 都只需敲入命令 **SET CATALOG TO WRITE** 即可。

编目文件是 **III PLUS** 系统的过滤器, 因此只显示与之有关的 **DBF**、**DBT**、**NDX** 等文件。以 **SET CATALOG TO WRITE** 为例, 要使用 **DBF** 文件, 但不能确定它在数据库系统的哪一部分, 就可用下述方式请求 **III PLUS** 列出 **WRITE** 使用的数据库清单:

USE ?

III PLUS 以列出输入到 **WRITE.CAT** 文件的数据库清单为响应, 如图 2.3 所示。

-> DIR *.CAT

4947968 bytes remaining on drive.

```
-> SET CATA TO WRITE
Enter title for file WRITE.cat: :ALL BOOKS FROM HPAL
-> USE MAGAZINE INDE A: MAGAZINE, A:MAIDS
Enter title for file MAGAZINE.dbf: :ALL MAGAZINE ENTRIES
-> USE BOOK INDE A:BOOK, A:BAUDS
Enter title for file BOOK.dbf: :ALL BOOK ENTRIES
-> USE INCOME INDE A:RFROM, A:RFOR
Enter title for file INCOME.dbf: :EARNING INCOME
-> USE EXPENSES INDE A:PTO, A:PTOR
Enter title for file EXPENSES.dbf: :EARNING EXPENSES
-> DIR *.CAT
WRITE.CAT          CATALOG.CAT
```

697 bytes in 2 files.
4939776 bytes remaining on drive.

->

图 2.2 在使用 SET CATALOG TO filename
命令时 III PLUS 创建一个新目录

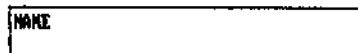
Select a DATABASE from the list.



ALL MAGAZINE ENTRIES

图 2.3 WRITE.CAT 文件列出
可用于 WRITE 的数据库

若想使两个 NDX 文件与 MAGAZINE 数据库文件一起工作, III PLUS 必须先记录下来, 等选择 MAGAZINE. DBF 文件之后, III PLUS 显示有关的 NDX 文件(见图 2.4)。



Position selection bar - ↑↓. Select - 4-J. Leave menu - ←, Select index file(s) to be used with the chosen file.

图 2.4 数据库一旦被选择, III PLUS 就读出目录文件。

、查找出与之相关的文件, 如 NDX 文件

III PLUS 是如何在 CAT 文件中保存其所有信息的呢? 欲知道这些, 需了解 CAT 文件的结构, 见图 2.5。

CATALOG 文件通过 CODE 域显示文件的相关部分, 此域作为数据库文件的指针使用。在图 2.5 中, MAGAZINE. DBF 是 WRITE. CAT 列出的第一个 DBF 文件, 其码值(code)是 1, 两个 NDX 文件与 MAGAZINE 数据库有关, MAGAZINE. NDX 和 MAUDS. NDX 存于 WRITE. CAT 文件的不同记录中, 但它们的代码都是 1, 与 MAGAZINE. DBF 一样。

BOOK. DBF 文件的代码是 2, 意思是说它是 WRITE. CAT 文件列出的第二个 DBF 文件。BOOK. NDX 和 BAUDS. NDX

```

-> DISP ALL TRIM(PATH),TRIM(FILE_NAME),TRIM(ALIAS),TRIM(TYPE),CODE,TRIM(TAG)
Record# TRIM(PATH)
TRIM(FILE_NAME) TRIM(ALIAS) TRIM(TYPE) CODE TRIM(TAG)
1 D: MAGAZINE.dbf
MAGAZINE.dbf MAGAZINE dbf 1
2 A: MAGAZINE.ndx
MAGAZINE.ndx MAGAZINE ndx 1
3 A: MAUDS.ndx
MAUDS.ndx MAUDS ndx 1
4 D: BOOK.dbf
BOOK.dbf BOOK dbf 2
5 A: BOOK.ndx
BOOK.ndx BOOK ndx 2
6 A: BAUDS.ndx
BAUDS.ndx BAUDS ndx 2
7 D: INCOME.dbf
INCOME.dbf INCOME dbf 3
8 A: RFROM.ndx
RFROM.ndx RFROM ndx 3
Press any key to continue...

```

图 2.5 CAT 文件通过 CODE 域
显示文件的相关部分

文件与 **BOOK.DBF** 文件有关,代码都是 2。这也适用于 **INCOME.DBF** 和 **RFROM.NDX** 文件。如果用户使用一个数据库但没有把相关文件汇总在一起而留待以后再说,那也没有关系。**CAT** 文件的联系是通过码值方式进行的,这种文件关联方式也适用于 **FMT**、**FRM** 和其它文件。注意,**CAT** 文件首先知道 **DBF** 文件,然后将其它文件与那些 **DBF** 文件相关联(**VUE** 文件的情况除外,后面将会涉及到)。

最后一个注意事项是:改变磁盘或目录将会对 **CAT** 文件有灾难性的影响。如图 2.6 所示,改变磁盘会使 **PLUS** 清除 **CAT** 文件的信息,这个问题可通过关闭 **CAT** 文件的方式避免。关闭文件可用下述命令:

SET CATALOG OFF

也可用下述命令彻底关闭 **CAT** 文件(不用任何文件名):

```

-> SET CATA TO WRITE
A:MAGAZINE.ndx does not exist. Deleted from catalog
A:MAUDS.ndx does not exist. Deleted from catalog
A:BOOK.ndx does not exist. Deleted from catalog
A:BAUDS.ndx does not exist. Deleted from catalog
A:RFROM.ndx does not exist. Deleted from catalog
A:RFOR.ndx does not exist. Deleted from catalog
A:PTO.ndx does not exist. Deleted from catalog
A:PFOR.ndx does not exist. Deleted from catalog
->

```

图 2.6 在使用 CAT 文件时改变磁盘或目录
能改变打开文件所列出的信息

SET CATALOG TO

第一种方法能保持 CAT 文件,但不能在动态下修改。当通过命令 SET CATALOG TO 使用 CAT 文件时,占 10 号工作区,如图 2.7 所示,DISPLAY 命令显示包括 WRITE. CAT 文件的 10 号工作区。

文件名.DBF——数据库文件(DATABASE File)

DBF 文件是数据库中最易识别的文件,是最能反映 dBASE 特点的文件,而且正是这种文件保持了 dBASE 的能力。

Ⅲ PLUS DBF 文件和 dBASE Ⅲ DBF 文件相同,二者都使用 MEMO 域,但与 dBASE Ⅱ 的 DBF 文件不同,在 Ⅲ PLUS 中使用 dBASE Ⅲ 文件之前必须加以修改,反之亦然。

实际上,可以把 Ⅲ PLUS 的 DBF 文件比作一个显示画面,整个数据库文件是显示画面,每个数据库记录变成一个数据列,每个域变成一个数据行,如图 2.8 所示。

DBF 文件可用 CREATE 命令创建:

-> DISP STAT

Currently Selected Database:

Select area: 1, Database in Use: D:\EXPENSES.dbf Alias: EXPENSES

Select area: 10, Database in Use: D:\WRITE.cat Alias: CATALOG.

File search path:

Default disk drive: D:

Print destination: LPT1:

Margin = 0

Current work area = 1 Delimiters are ';' and ':'

Press any key to continue...

*** INTERRUPTED ***

->

图 2.7 命令 SET CATALOG TO filename

置 CAT 文件于 10 号工作区

CREATE filename

图 2.9 是一类 DBF 文件的屏幕格式生成显示情况(注意,这里 SET MENU 处于 ON 状态)。

在图 2.9 中, III PLUS 被告之要创建 TANK, III PLUS 就知道用户想要创建一个名为“TANK”的数据库。图 2.10 示出的是一些新的数据库域的填充情况。

DBF 文件创建时会在屏幕上显示许多关于这方面的信息,右上角窗口显示记录中剩余的字节数,每个 DBF 记录含有 4000 字节的信息。窗口从上到下显示两个相同的列集,每一个包括:

Field Name	Type	Width	Dec
------------	------	-------	-----

这些列由用户告诉 III PLUS 每个域的实际特性。特性定义如下:

	1	2	3	4	5	6	7
	EXPENSES						
PAYTO	Allstate	Allstate	Susan Ca	Susan Ca	mileage	MR Turnp	Computer
PAYFOR	Rental I	Auto Exp	Rent, Ut	Office S		Toll	Office S
CHECKNUM	tovisa	tovisa	tovisa	tovisa		cash	cash
PAYAMOUNT	57.00	246.00	332.89	298.00	0.00	5.00	14.50
PAYDATE	07/12/86	07/12/86	07/12/86	07/12/86	07/11/86	07/12/86	07/08/86
MILEAGE	0.00	0.00	0.00	0.00	146.00	0.00	0.00

图 2.8 图中,数据库比作显示画面,一个记录作为一行,一个域作为一行

域名(Field Name) 这是为 III PLUS 定义域名的地方,如 ADDRESS, CITY, FIRSTNAME, PO _ NUMBER, DATE _ IN 等等。注意,域名由任何的字母和数字组合而成,但长度不能超过 10 个字符,不过,使用正好 10 个字符是有好处的,这样可以使每个域描述得更清楚,例如,DO 和 DI 可用作 date-out 和 date-in 的域名,但“DATE _ OUT”和“DATE _ IN”更进一步地表达出每个域的特点。域名中不能有空格,首字符必须是字母。在域名中一般用下划线(“_”)表示空格(如 DATE _ IN 和 PO _ NUMBER)。

类型(Type) 这是为 III PLUS 定义可输入到此域的数据类型。最常见的是字符型(CCHARACTER),它允许输入任何字符。如果不能确定一个域用什么类型最好,就可用字符型。另

Bytes remaining: 4000

CURSOR (---)	INSERT	DELETE	Up a field: ↑
Char: + +	Char: Ins	Char: Del	Down a field: ↓
Word: Home End	Field: ^M	Word: ^M	Exit/Save: ^End
Pan: ^+ ^_	Help: F1	Field: ^H	Abort: Esc

Field Name	Type	Width	Dec
1	Character	20	

Enter the field name.
Field names begin with a letter and may contain letters, digits and underscores

图 2.9 一类 III PLUS DBF 文件的屏幕生成显示

Bytes remaining: 3830

Field Name	Type	Width	Dec
1 NAME	Character	30	
2 ADDRESS	Character	30	
3 CITY	Character	20	
4 STATE	Character	2	
5 ZIPCODE	Character	10	
6 BUSPHONE	Character	14	
7 HOMEPHONE	Character	14	
8 INFO	Memo	10	
9 LASTPURCHS	Date	8	
10 LASTPAYMNT	Numeric	10	2
11 BALANCE	Numeric	10	2
12 CREDIT	Logical	1	
13 CREDITGOOD	Logical	1	
14 CREDITLIMIT	Numeric	10	2
15	Character		

Enter the field name.
Field names begin with a letter and may contain letters, digits and underscores

图 2.10 本图类似图 2.9, 不过这里的数据库域已经定义

一方面,如果要确保输入数据为某种确定的类型,则必须把这个域的数据类型确定下来。注意,地址(如“32 **Foxmoor Circle**”)是字符型数据。类型可用输入一个字母的方式确定(**C**表示字符,**M**表示 **Memo**,**D**表示日期,**L**表示逻辑,**N**表示数字)。

宽度(Width) **III PLUS** 用它定义域的宽度,如用户可能把一个人的全名存在 **NAME** 域,则应定义此域的宽度为 40 字符。用户可能想把一个人的姓存在“**FIRSTNAME**”域中,而名字存在另一个域“**LASTNAME**”中。在这种情况下,就可为 **III PLUS** 定义 **FIRSTNAME** 和 **LASTNAME** 域,每个域的大小均为 20 字符。注意,这里有些特殊情况,如果定义类型为 **LOGICAL** 的域,**III PLUS** 会自动生成一个单字符域。如果定义的是 **MEMO** 域,**III PLUS** 会自动把它的宽度定义为 10 个字符。

小数(Dec) 这一列是已描述的 **Width** 列的辅助列,**III PLUS** 只有在说明了 **NUMERIC** 型的域之后才允许向本列输入数据,在这一列可为 **III PLUS** 定义数据的小数。例如,说明 **NUMERIC** 域 10 字符宽,且有 2 位小数,则以 99.00 的方式显示 99.3 字符宽且没有小数位时显示 99.4 字符宽有一位小数显示 99.0。**NUMERIC** 型数据的宽度的上限是 19 字符。

屏幕底部的状态行(**Status**)显示 **CREATE** 方式,现正在 **C** 盘上创建一个名为 **TANK** 的文件,在第一个域,且总共有一个域,下面一行提示域名,最后一行显示允许信息的类型。比较图 2.9 和图 2.10,在图 2.10 中已输入了若干域,且提示下一域的类型(注意,这些显示都是在 **SET STATUS ON** 之下的)。

图 2.11 显示了 DBF 用 MODIFY 修改或用 CREATE 创建时出现的另一菜单。该菜单允许用户在 DBF 的域中快速移动,同时也具有 SAVE 或 ABANDON 文件的能力。

Bottom Top **FIELD** Save Abandon **W2-11-83 PM**

Enter field number: **01** tes remaining: 3399

Field Name	Type	Width	Dec	Field Name	Type	Width	Dec
1 WARDEN ID	Character	10		16 RESULT	Memo	10	
2 CLNAME	Character	30		17 FOLLOWUP	Logical	1	
3 CLADDRESS1	Character	30		18 NEXTCONTACT	Date	8	
4 CLADDRESS2	Character	30		19 BILLING	Character	1	
5 CLCITY	Character	20		20 LASTPURCHS	Date	8	
6 CLSTATE	Character	2		21 PURCHSAMNT	Numeric	10	2
7 CLZIPCODE	Character	10		22 BALANCE	Numeric	10	2
8 BUSIPHONE	Character	14		23 CREDITCARD	Character	4	
9 HOMEPHONE	Character	14		24 CCNUMBER	Character	25	
10 INCOME	Numeric	12	2	25 EXPIRATE	Date	8	
11 LEAD	Character	20		26 STORECREDT	Logical	1	
12 REFERENCE	Character	254		27 CREDITLIMIT	Numeric	10	2
13 BANK REF	Memo	10		28 CREDITGOOD	Logical	1	
14 SALES	Character	20		29 CLHISTORY	Memo	10	
15 CONTACT	Date	8		30 PURCHSHIST	Memo	10	

MODIFY STRUCTURE OF FIELD **FIELD** **W2-11-83** **COPY**

Position selection bar with ←. Select with →.

图 2.11 在 CREATE 或 MODIFY 一个文件时

按 CTRL-HOME 即可显示此可用菜单

还需注意,如果 SET SAFETY 处于 ON 状态,III PLUS 就会警告用户可能还有另一个同名的 DBF 文件存在,这样就再也不能使用这个名字或停止操作,并且选择另一个文件名。

每个数据库文件可能是 CHARACTER、NUMBER、LOGICAL、MEMO 和 DATE 数据类型的任意组合。对 DBF 文件只有大小的限制,记录的个数和域的个数都有上限。任何一个数据库允许记录的最大数只能是 1G 个,每个数据库文件最多可达 2G 字节(1G=10⁹)。

不过,尽管可获得所有空间,但也不要生成每个记录有 128 个域的数据库文件,也别指望生成一个超过 1M 数据的数