

刘 军 李 东 等 编 著

Clipper 实用大全

科 学 出 版 社

Clipper 5.0 实用大全

刘 军 李 东 等 编著

科 学 出 版 社

1 9 9 4

(京)新登字 092 号

内 容 简 介

Clipper 是 dBASE 语言的最新编译版本,它使用 dBASE III Plus 的扩充作为其标准命令集。本书对 Clipper 的所有方面均做了详细阐述,主要包括 Clipper 的使用方法,Clipper 的语言环境、用户接口、网络应用等。本书不同于一般的 Clipper 使用手册,它兼顾了 Clipper 的基本操作与编程技巧,是一本很有价值的参考手册。

本书可供计算机软件开发人员、应用人员及相关专业的高校师生参考。

Clipper 5.0 实用大全

刘 军 李 东 等 编 著

责任编辑 那莉莉

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

北京朝阳大地印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1994 年 8 月 第 一 版 开本:787×1092 1/16

1994 年 8 月第一次印刷 印张: 22

印数:1—5 000 字数:504 000

ISBN 7-03-004070-8/TP·352

定价:22.00 元

前 言

众所周知,dBASE 数据库管理系统由于具有实用性强,易学、易用等特点,曾在我国得到广泛的推广与应用。但是,随着时间的推移,其缺点也越来越明显,如计算能力差、处理速度慢、没有图形功能等。这些缺点不但限制了 dBASE 在更多应用领域的发展,而且导致了以前使用 dBASE 开发的应用项目只能停留在较低的水平,难以满足高水平综合信息管理的需要。Clipper 是 dBASE 语言的最新编译型版本,自问世以来,对微机数据库管理系统开发产生了深远影响。

Clipper 是一个开发工具,它利用 dBASE III Plus 的扩充作为其标准命令集,其命令及函数是 dBASE III Plus 的超集,其扩展系统则允许用户存取用 C 与汇编语言编制的例程和函数。除此之外,Clipper 还提供了强有力的编译预处理程序、连接程序和调试程序。

本书不同于一般的使用手册,它全面介绍了运用 Clipper 进行程序设计的基本知识,特别注重编程与实例的介绍,是一本不可多得的参考书。

本书由刘军、李东、岳东、李晓燕、廖斌等编写,全书由任天电子信息技术研究所审校。由于时间仓促,加之水平有限,不当之处在所难免,敬请读者批评指正。

目 录

第一章 Clipper 简介	1
1.1 Clipper 的本质	1
1.2 Clipper 的特点	1
1.3 Clipper 的安装和软件工具	2
1.4 Clipper 软件工具	3
1.5 Clipper 技术规格	9
第二章 Clipper 程序设计	11
2.1 设计风格	11
2.2 程序设计基础	12
第三章 文件处理	21
3.1 文件的类型	21
3.2 文件结构	23
3.3 文件函数的用法	39
3.4 文件复制	42
3.5 文件规模	43
3.6 文件的行读入和读入暂存区	43
3.7 文件保护	49
第四章 Clipper 的内存管理技术	52
4.1 非常规内存管理	53
4.2 虚拟存储管理	53
4.3 与 DOS 的接口	54
4.4 DOS 环境	54
4.5 提供 DOS 服务	55
4.6 覆盖管理技术	56
第五章 程序编译、链接及调试	62
5.1 Clipper 编译程序	62
5.2 链接程序	63
5.3 Clipper 的调试程序	66
5.4 错误处理	69
第六章 Clipper 与 C 和汇编语言接口	78
6.1 概论	78
6.2 与 C 语言的接口	79
6.3 扩展系统的 C 语言函数	85
6.4 与汇编语言的接口	92

6.5	扩展系统的汇编语言函数	96
6.6	编译和连接	103
6.7	C 语言函数应用举例	107
第七章	查询技术及文件重整	121
7.1	打开数据库	121
7.2	搜寻一个值	125
7.3	查找记录	127
7.4	通过关联指令连接数据库	129
7.5	多重索引	132
7.6	建立数据库	132
7.7	文件重整	134
第八章	界面管理技术	138
8.1	简单的功能表	138
8.2	BOX 指令	142
8.3	按键的处理	146
8.4	光标的处理	149
8.5	填充键盘缓冲区	150
8.6	屏幕的存储与恢复	157
8.7	屏幕和 MEM 文件	161
8.8	GET 的处理	162
8.9	垂直滚动	175
第九章	Clipper 网络技术	188
9.1	多用户程序设计	188
9.2	网络程序设计难题	189
9.3	Clipper 网络命令	190
9.4	Clipper 遵循的原则	192
9.5	错误事件的处理	194
9.6	网络中的索引文件和其他文件	200
9.7	读取/修改/写入周期	204
9.8	使用整个文件数据的命令	207
9.9	在单用户系统下的测试	209
第十章	Clipper 对象类及原型开发	212
10.1	什么是对象	212
10.2	Error 对象	213
10.3	Get 对象	215
10.4	TBrowse 系统	219
10.5	TBColumn 对象和输出实例变量	224
第十一章	Clipper 的增强功能	226
11.1	表达式替代	226

11.2	用户自定义函数	227
11.3	扩展函数的用法	228
11.4	SET()函数	228
11.5	FOR...NEXT 循环	229
11.6	Clipper 数组	231
11.7	建立 KEY 过程	232
11.8	在@...SAY...GET 中的扩展功能	232
11.9	屏幕变量	235
11.10	增强菜单	235
11.11	低级文件处理	241
11.12	备注区和字符域处理	243
11.13	TBrowse——Clipper 面向对象的浏览	244
11.14	Clipper 的帮助工具	248
11.15	代码块	250
附录 A	ASCII 代码以及 INKEY()函数的返回值	253
附录 B	保留字	255
附录 C	Clipper 基本命令	261
附录 D	Clipper 基本函数	297
附录 E	Clipper 的错误信息	329

第一章 Clipper 简介

1.1 Clipper 的本质

Clipper 是一个由 Nantucket 公司推出的编译程序和连接程序,它是专门为 dBASE 语言应用程序而设计的。它建立的程序文件能运行在 IBM PC 及与其兼容的计算机上。Clipper 还提供了对 dBASE III Plus 语言的一系列扩充,可以执行一些对 dBASE III Plus 来说很困难或不可实现的任务。

用 Clipper 或 dBASE 语言写完一个程序后,就可以使用 Clipper 把它转换为一个能被计算机的磁盘操作系统(DOS)直接运行的独立程序。你不必为程序的每一个用户准备一个 dBASE III Plus 的备份。你也许想有一份 dBASE III Plus 或 dBASE IV 的备份以便为用户开发与维护数据文件,但却省下了为每一个用户机器装入一个 dBASE III Plus 备份的开销。人们不必购买一个软件包来运行所编制的软件。一个 Clipper 应用程序是完全自含的。

但是,Clipper 开发环境是自含的,Nantucket 公司提供了编写报表及创立数据文件的实用程序,用户需要为 Clipper 添加一个自己熟悉的文本编辑程序,通过这个文本编辑器来编写用户程序。

Clipper 有许多重要的开发工具,不需要 dBASE III Plus 或 dBASE IV 就可以开发并测试应用程序,Clipper 有全新的专用调试器。

软件包中的 DBU, PRG 实用程序,用于建立与修改数据和索引文件,它是用 Clipper 语言编写的,因此,用户可以认真阅读它,看它是如何实现其强大功能的,同时欣赏一下编写 Clipper 程序的一种良好风格。

在 Clipper 的其他工具中,Make 实用程序用于追踪修改过的程序模块。

Clipper 语言的一个特征就是它的开放式结构,可以用 C 语言和汇编语言编写各种函数和例程,并将它们像 Clipper 语言写的程序一样挂入 Clipper 应用程序中。因此,在 Clipper 中提供了良好的 C 语言、汇编语言接口,用户应该充分利用这一点。

1.2 Clipper 的特点

Clipper 提供了许多独特和有力的工具给数据库设计人员使用,其主要优点有:

- (1) 运行速度极快。
- (2) 源程序绝对保密。
- (3) 可在 DOS 2.0 版以上(单用户)或 3.1 版以上(多用户)的系统中运行 Clipper 编译过的程序。
- (4) 改进的网络功能,对源程序有绝对的控制保护功能。
- (5) 可随意销售编译后的应用系统。

- (6) 可将程序及函数存放在同一个文件中,当作过程文件。
- (7) 许多函数和运算可处理 64KB 的字符串。
- (8) 可以选择更快速的 Clipper 索引文件(.NTX)或 dBASE III Plus 的索引文件(.NDX)。
- (9) 可存取低级的 DOS 文件及设备。
- (10) 可使用 2048 个活动的内存变量。
- (11) 每个数据库可使用 1024 个字段。
- (12) 利用 DOS 3.3 版可打开 255 个文件。
- (13) 可建立用户自定义函数。
- (14) 每个数据库可以与 8 个文件相关联。
- (15) 可调用无限多的外部 C 语言或汇编语言程序。
- (16) 能将备注型字段当作字符型串值处理。
- (17) 可递归或嵌套使用宏代换。
- (18) 在 @... Box 命令中可以用 9 个字符来指定矩形框的边界和中间填充字符。
- (19) 框的边界和中间填充字符。
- (20) 可以建立空结构的扩展文件,不需要先存在一个数据库文件。
- (21) 利用 PUBLIC CLIPPER,可在交互式系统中执行应用程序。
- (22) 可执行“FOR...NEXT”形式的循环命令。
- (23) 网络环境中对数据库操作失败时会自动返回一个逻辑值来指定。
- (24) 在共享文件中可使用“Append Blank”命令。
- (25) 当记录被其他用户锁定时,也能将记录指针指向该记录,并显示它。
- (26) 允许无限多的用户存取一个数据库。
- (27) 可提供更灵活方便的帮助功能。
- (28) 利用公用程序可在 DOS 系统下直接建立数据库索引文件、报表文件、标签文件等。
- (29) 具有完整的调试工具。

1.3 Clipper 的安装和软件工具

当拿到原版 Clipper 磁盘,绝不可直接用于编译或连接工作。首先,应该将 Clipper 拷贝到硬盘中,接着将它拷贝到一片软盘上作为备份。然后,在根目录下建立一个子目录以备存放 Clipper 文件,例如,下面这个指令将建立一个叫“Clipper”的子目录:

```
C>MD Clipper
```

一旦子目录建立好之后,请依照下列指令进入该子目录:

```
C>CD Clipper
```

现在请将 Clipper 系统磁盘插入 A 驱中,并请键入“A:”以转换到 A 驱上,接着请键入下列指令:

```
A>CLIPCOPY(<drive>)
```

上面指令中的(<drive>)是磁盘的识别字。若拿到的 Clipper 磁盘没有 CLIPCOPY.BAT 的文件,则将 Clipper 系统磁盘插入 A 驱中,接着请键入下列指令:

```
C>COPY A:*. *
```

上面的指令是将 A 盘中的所有文件复制到硬盘中。

1.4 Clipper 软件工具

当用户将 Clipper 系统安装好后,可以在 BIN 目录中找到一些 .EXE 和 .COM 文件,其中除了 Clipper 的编译器 CLIPPER. EXE 外,其他都是一些协助性软件工具。下面,我们将主要介绍四个软件工具:编译器 CLIPPER. EXE,链接器 RTLINK. EXE,程序管理软件 RMAKE. EXE 及资料库维护软件 DBU. EXE。

1.4.1 编译器 CLIPPER. EXE

编译器 CLIPPER. EXE 用于将一个或多个源文件(. PRG)编译成一个目标文件(. OBJ),其语法如下:

CLIPPER[<源文件>|@<编译描述文件>[<编译选择项列表>]]

其中,

<源文件>:希望编译成目标文件的某个源程序文件名称,其文件扩展名默认为. PRG。

<编译描述文件>:当一个文书文件,文件内每一行记录一个程序文件名称(不需要附加文件扩展名. PRG)。当编译器进行编译时,它将根据描述文件的内容,对指定的源文件进行编译,然后将这些编译结果结合在一起,产生一个目标文件。编译描述文件的扩展名为. CLP。

<编译选择项列表>:Clipper 5. x 中提供了远比 Summer'87 版更多,但并非必须的编译选择项,因此用户可以根据需要来选用它们。选择项以字符 '/' 或 '_' 作为开始,每个选择项均可以用大写字母或小写字母表示,文件名与第一个选择项间必须用空格符分开,而选择项间不能留有空白字符。当选择多个选择项时,选择项排列的顺序对编译结果没有任何影响。

用户若在 DOS 环境下直接执行不带任何参数的编译指令,系统将在屏幕上列出所有可能使用的编译选择项。下面就是执行 Clipper 命令后列出的编译选择项列表:

Designed by Rich McConnell

Written by Denny Dias,Basil Hosmer and Rich McConnell

Options:

/a	automatic memvar declaration
/b	debug info
/credits	credits file search path
/d<id>[=<Val.]	#define <id>
/i<path>	#include file search path
/l	suppress line number information
/m	compile module only
/n	no implicit starting procedure
/o<path>	object file drive and/or path
/p	generate pre-processed output (ppo)file
/q	quiet
/r[<lib>]	request linker to search <lib> (or none)
/s	syntax check only

/u[<file>]	use command def set in <file> (or none)
/v	variables are assumed M->
/w	enable warnings
@<file>	compile list of modules in <file>

下面是这些选择项的功能说明及使用时机：

选择项 A 将使用 PRIVATE, PUBLIC 及 PARAMETERS 声明的变量内定为内存变量 (memvar ->)。

选择项 B 将在线检错程序所需的信息加入编译后的目标文件中。

选择项 Credits 显示 Clipper 5. x 系统设计人员名单。

选择项 D 借助预处理器, 将某一个可以设置内含值的标识符连同程序一起编译成目标文件。

选择项 I 指定文件搜索路径。可同时用多个选择项 I 指定多个搜索路径。

选择项 L 编译成不含程序行号的目标文件。

选择项 M 只编译目前指定的源程序文件, 而不会自动编译由 DO, SET FORMAT 或 SET PROCEDURE 所打开的外部程序文件。

选择项 N 抑制系统中自动将编译后目标模块名称内定为源程序文件名称的功能, 而以源程序内第一个程序或函数名称作为目标模块名称。

选择项 O 指定目标文件的存放路径或名称。

选择项 P 编译程序文件, 并产生一个与源文件相对应的预处理文件, 文件扩展名为 .ppo。

选择项 Q 编译时不在屏幕上显示程序行号。

选择项 R 提前指定本目标文件在其后连接时欲连接的库文件(. LIB)。可以用多个 R 选择项指定多个连接库, 当使用 R 指定连接库时, 内定的 CLIPPER. LIB, EXTEND. LIB 和 DBFNTX. LIB 不会自动搜索连接。

选择项 S 仅检查程序语法是否正确, 不产生目标文件。

选择项 T 指定编译时存放临时文件的目录。

选择项 U 指定取代预处理时所使用的标准头文件(std. ch)的头文件名称。

选择项 V 编译时将所有未明确声明或无别名的变量, 全部内定为 PUBLIC 或 PRIVATE。

选择项 W 编译时对所有未明确声明或无别名的变量在屏幕上显示警告信息。

选择项 Z 抑制编译器对逻辑运算的短路(最优化)处理。

1.4.2 链接器 RTLINK. EXE

用于将一个或多个目标文件(. OBJ)与指定的库文件(. LIB)进行连接, 产生一个可执行文件(. EXE)或预连接库(. PLL)。其语法格式如下：

```
RTLINK [FILE<目标文件列表>
        [OUTPUT<执行文件名>|<预连接库文件名>]
        [LIBRARY[<参与连接的库文件列表>]]
        [<连接选择项列表>]]
```

或者

RTLINK [@<连接描述文件名>]

位置格式:

RTLINK [<目标文件列表>,
[<可执行文件名>|<预连接库文件名>],
[<连接后产生的对照文件名>],
[<参与连接的库文件列表>],
[<连接选择项列表>]]

或者

RTLINK[@<连接描述文件名>][;]

其中,

<目标文件列表>:希望连接的目标文件名列表,其文件的默认扩展名为.OBJ。

<可执行文件名>:连接后产生的可执行文件名称,其默认的扩展名为.EXE。

<预连接库文件名>:连接后产生的预连接文件名称,扩展名为.PLL。

如果在命令中未显式指出<可执行文件名>或<预连接库文件名>,则系统将把<目标文件列表>中的第一个目标文件名作为连接后的文件名。

<参与连接的库文件列表>:目标文件中所使用的各个库文件列表。

<连接选择项列表>:连接时根据需要指定的各个选择项列表,每个选择项都用字符“/”作为前导,选择项不分先后顺序和大小写,但选择项间不能有空白字符,选择项列表与前面的文件列表间以一个空格分开。下面是连接选择项的各项说明及含义,其中加上星号(*)字符的选择项为自由格式语法所专用。

选择项 BATCH

系统默认的状态为NOBATCH。当程序连接时发现有无定义的符号时,会在屏幕上显示提示信息,主用户输入该符号所在文件的名称。

选择项 *BEGINAREA...<连接参数列表>...ENDAREA

建立动态的覆盖结构。

选择项 DEBUG

在程序执行期间,当新的动态覆盖模块装载到内存时,显示一个提示信息到屏幕上。

选择项 EXCLUDE:<Symbol>

在预连接模式时,用于指定某些不连接的符号。

选择项 HELP

显示 RTLINK 所有的选择项及其使用时机。

选择项 INCREMENTAL[,<Wasted Space>]

系统默认为 NOINCREMENTAL 状态。它用于启动渐进连接的功能,只有修改过的目标文件才会被重新连接一次。

选择项 MAP[,<MapOption List>]

产生一个关于所有连接段的公用符号文件,可选参数有三个 S,N,B,A。

选择项 MODULE<模块名列表>

将所有指定模块的段指至当前的动态覆盖区内。

选择项 NODEFAULTLIBRARY SEARCH

系统默认为 DEFAULTLIBRARY SEARCH。它用于取消系统自动连接 Clipper. LIB, EXTECD. LIB, DBFNTX. LIB, TERMINAL. LIB 四个库文件的功能。

选择项 NOEXTDICTIONARY

系统默认为 EXTDICTIONARY。

系统默认为 EXTDICTIONARY。

外部附加字典文件是一个符号位置表,RTLINK 用它来加速库文件的搜索。使用 NOEXTDICTIONARY 时要小心,因为当有同名的符号存在时,链接器不会发出警告。

选择项 NOIGNORE CASE

系统默认为 IGNORECASE。将大写或小写的符号或段视为不同的两个符号或段。

选择项 PLL:<预连接项>

指定可执行文件或预连接库所依赖的预连接库,此关系只能有一层。

选择项 POSITIONAL

系统默认为 FREEFORMAT。设定链接器使用位置格式的语法。

选择项 PRELINK

启动预连接模式,在此模式下,输出文件将从可执行文件(.EXE)转变为预连接库文件(.PLL),同时也产生预连接转移文件(.PLT)。

选择项 PRELOAD

将动态覆盖区在主程序执行之前装入内存中。一般情况下,只有常驻部分才会在主程序执行之前装入内存,而动态覆盖区只有执行到该部分时,才会装入内存。

选择项 REFER:<Symbol>

在预连接模式下,搜索所有的默认库文件,将指定符号所在的程序代码,连接到输出文件中。

选择项 RESIDENT

系统默认为 DYNAMIC[:<覆盖文件>]。

将所有后续编译的模块加入到目前程序文件或预连接库的根节区内。

选择项 SECTION[=<节区名>][INTO<覆盖文件>]

建立一个动态覆盖区,也可以让后续 FILE 或 LIBRARY 选择项内非 Clipper 目标模块成为该覆盖区的部分。

选择项 SILENT

设定链接器在对描述文件做连接处理时,不在屏幕上显示提示信息。

选择项 STACK:<SizeBytes>

重新配置栈的内存单元数,取代先前在目标模块内所设定的数值。栈的最大容量为 65535 字节。

选择项 VERBOSE[:<level>]

在屏幕上显示链接器目前正在连接的模块名称及其先后顺序,可选择参数有 0,

1 及 2 三项。

1.4.3 程序管理软件 RMAKE.EXE

此软件使用在源程序的编译与连接中,它只针对后面有修改的程序段再作编译或连接操作,这样可以节省编译和连接程序的时间。其语法如下:

```
RMAKE [<.RMK 文件列表>]  
      [<宏指令列表>]  
      [<功能选择项列表>]
```

其中,

<.RMK 文件列表>:可同时执行多个 MAKE 程序。

<宏指令列表>:MAKE 内同时可以定义多个宏常数,格式为<宏名称>=<内含值>。

<功能选择项列表>:程序管理器的多个功能选择项。

如果用户在 DOS 环境下执行不带任何参数的 RMAKE 命令。系统将在屏幕上显示所有可用的选择项:

```
/B    Show debugging into  
/D    <name>[:<value>] Define macro  
/F    Force targets to be made  
/I    Ignore exit codes  
/N    Null make (don't execute actions)  
/Q    Quiet (suppress signon messages)  
/U    Use # +or comments  
/W    Show warnings  
/XS   Set symbol table size  
/XW   Set workspace size
```

下面是这些选择项的说明:

选择项 B 在屏幕上显示程序管理器目前正在处理的步骤。

选择项 D <宏名>[:<Value>]

定义一个宏及其内含值,未指定其值的,默认为空 NULL。

选择项 I 忽略所有的错误,继续向下执行。

选择项 N 虚拟执行,在屏幕上显示所有欲执行的命令,但实际上并未真正执行。

选择项 S 设定所有子目录为数据文件的搜索目录。

选择项 U 设定井号字符(#)为备注行符号<内定的备注行符号为惊叹号(!)>。

选择项 W 显示某些特定的警告信息。

选择项 XS <NumSymbols>

设定内部符号表的大小,其默认值为 500。

选择项 XW <NumBytes>

设定内部工作区的大小,默认值为 2048 字节。

1.4.4 数据库维护软件 DBU.EXE

此软件用于建立和维护数据库和索引文件、并可支持一些简单的数据库查询工作,其语法如下:

DBU[<单色、彩色显示器设置>][<文件名称>]

其中,

<单色、彩色显示器设置>:参数/C 表示彩色显示器,/M 或未设置则表示单色显示器。

<文件名称>:这里的文件名称是指浏览文件(.VEW)或数据库文件(.DBF)名称。

用户将文件 DBU.HLP 拷入工作目录后,可以使用 F1 键获取在线的帮助。DBU.EXE 是一个菜单驱动软件,用户可以根据菜单的选择项及其提示信息进行操作。表 1.1 是有关功能的说明。

表 1.1 DBU 的功能说明

功能键	功能类型	功能说明
F1	Help	Help: 进入在线帮助屏幕
F2	Open	Database: 打开数据库文件(.DBF) Index: 打开索引文件(.NTX) View: 打开浏览文件(.VEW)
F3	Create	Datavase: 建立、修改数据库结构 Index: 建立、修改索引文件
F4	Save	View: 将目前文件打开状态存入浏览文件 Structur: 存储当前打开的数据库结构
F5	Browse	Database: 浏览数据库文件 View: 查阅浏览文件
F6	Utility	Copy: 拷贝当前数据库内容到另一个数据库或文本文件 Append: 从另一个数据库文件中取数据添加到当前数据库 Replace: 置换当前数据库中某些字段的值 Delete: 给符合条件的记录加上删除标志 Recall: 消除指定记录的删除标志 Pack: 消去数据库中有删除标志的记录 Zap: 消除数据库中所有记录 Run: 执行外部 DOS 命令或可执行文件
F7	Move	Seek: 搜索满足索引关键字值的记录 Goto: 移动记录指针到指定记录号的记录处 Locate: 搜索满足指定条件的记录 Skip: 将记录指针前移或后移指定的记录数
F8	Set	Relation: 设定数据库间的关联条件 Filter: 设定数据库内记录的过滤条件 Fields: 设定数据库欲显示的字段名称

1.4.5 预处理指令

Clipper 5.0 版新提供了一项预处理功能。在用户下达编译命令而程序真正执行编译操作之前,预处理器会先检查程序中是否含有预编译指令,若有则先将该预编译指令转换成相对应的程序码,待所有的,预编译指令皆已转换完毕后,才将整个程序编译成目标代码。

使用预编译指令可使用户在设计程序时多了一项实用工具,用户可以指定自己的命令名称、命令格式及虚拟函数,在程序中包含头文件等。

用户如果希望了解预处理器的实际功能及源程序与经过预编译处理后的程序有何不同时,可以在编译时使用/P 选择项,使编译器能输出一个扩展名为.PPO 的预处理输出文件。将源文件、预处理输出文件与存放在\Clipper5\Include 下的 Std.ch 文件相对照,可以观察三者的关系。

下面是有关的预处理指令:

#Command ——用于转换用户的自定义命令。

#Translate ——用于转换虚拟函数。

#Define ——定义一个常数或假函数。

#Error ——将指定的编译错误信息显示在屏幕上。

#Ifdef/#Else/#Endif ——利用某个标识符被定义与否,选择性地编译程序。

#Include ——读入指定的头文件,并编译它。

#Undef ——取消某个用#Define 定义的宏。

#Xcommand ——用于转换用户自定义命令。

#Xtranslate ——用于转换虚拟函数。

1.5 Clipper 技术规格

(1)数据库文件:记录条数为 10 亿条。

(2)记录:

结构:顺序定长

大小:视可用的 RAM 而定

字段个数:视可用的 RAM 而定,最大为 1024

(3)字段:

字符型:最长可达 64K 字符

数值型:最多为 19 字节

日期型:8 个字节

逻辑型:1 个字节

备注型:最多 64K 字符

(4)索引采用 B-tree 检索类型。

(5)数组:

个数:最多有 2048 个数组

元素:数组最多有 2048 个元素

元素大小:与内存变量相同,字符型为 64KB,数值型 19 字节,日期型 8 字节,逻辑型 1 字节

(6)内存变量:

个数:最多 2048 个

大小:字符型最多 64KB,数值型 19 字节。日期型 8 字节,逻辑型 1 字节

(7)文件:

打开的文件数:最多 255 个(DOS 3.3)

打开的索引文件数:每个工作区 15 个

(8)索引键长度最多为 250 个字符。

(9)每个文件可打开无限多个程序文件。