



最新出版

Clipper 5.2

实用大全

金连甫 编著



电子工业出版社

Clipper 5.2 实用大全

金连甫 编著

電子工業出版社

内 容 提 要

本书以 Clipper 5.2X 版为背景,全面系统地介绍了编译型数据库系统 Clipper 的功能、函数、命令、驱动程序和编译连接等有关内容,并列举了大量的实例进行说明。书中还特别介绍了 Clipper 的程序设计方法和编程技巧,网络上的程序设计方法和经验,以及 Clipper 与 C、汇编语言的连接方法和应用实例。

本书深入浅出、文字流畅、结构严谨、注重实用、适于自学,可作为该课程的培训教科书及自学的材料,亦可作为广大计算机工作者或管理工作者的参考书。

Clipper 5.2 实用大全

金连甫 编著

责任编辑:李 健(特约) 贾 贺

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

北京科技印刷厂印刷

开本:787×1092 毫米 1/16 印张:38.5 字数:86 千字

1995 年 12 月第 1 版 1995 年 12 月第 1 次印刷

印数:5000 册 定价:58.00 元

ISBN7-5053-3039-X/TP·1056

前 言

Clipper 是一种基于 dBASE III Plus 基础上扩充的编译型关系数据库,它既保证与数据库 dBASE、FoxBase 向上兼容,又可通过 Clipper 的编译器、连接器将其编译成一个可直接执行的 .EXE 文件。众所周知,自从 ASHTON—TATE 公司推出的 dBASE II、dBASE III、dBASE III Plus、dBASE IV 和 FOX 软件公司推出的 FoxBase 等一系列兼容的关系数据库,以它的灵活性、方便性、高效率 and 兼容性赢得了广大用户的欢迎,并广泛地应用于管理信息系统(MIS)和办公自动化(OA)系统的开发,并成功地应用于企事业单位、银行、学校、商店等部门,促进了我国的管理科学化、规范化。但美中不足的是,上述数据库都是解释型的,所以运行速度慢且不利于应用程序的管理。Clipper 就是为了弥补上述数据库的不足而推出的一个理想的数据库系统。对熟悉数据库 dBASE、FoxBase 的用户来说,可以很方便地使用 Clipper 进行编程或将用户以前用 dBASE 或 FoxBase 编制的程序不加修改或经少量的修改在 Clipper 上进行编译连接成一个可直接执行的 .EXE 文件。但这并不意味着 Clipper 就是 dBASE、FoxBase。Clipper 系统在整体设计上、语法成份上都有了很大的改进,使其语法成份及规则向 C 语言的风格靠,但为了与 dBASE、FoxBase 兼容,所以也保留了 dBASE 的格式。

Clipper 是一种开发者的工具,它使用 dBASE III Plus 的扩充作为它的标准命令集,Clipper 的编译程序接受的命令和函数是 dBASE 语言的超集。Clipper 采用嵌入式的预处理程序,使得用户可以建立用户定义的命令、条件编译,支持动态覆盖连接技术,使得超过 640KB 的可执行文件也能运行。它还引入了 4 类对象,提供面向对象程序设计技术(OOP)的雏型。允许用户自定义函数,使程序设计更加自如灵活,同时还具备“代码块”命令格式,可以提高编程和运行效率。此外,Clipper 提供了一批新的函数,这些函数功能强大、种类丰富,能和操作系统、网络环境建立更多的联系。

Clipper 引进与数据库无关、与设备无关的思想,采用驱动技术,使 Clipper 能够访问操作 dBASE III、dBASE IV、FoxPro 和 Paradox 等格式的数据库文件,支持非标准的视频硬件和 ANSI 终端。

Clipper 本身是用 Microsoft C 编写的,所以它同 C 语言有着天然的联系。Clipper 支持与 C 语言、汇编语言编制的函数和过程的直接连接,使用户很方便地使用 C 语言、汇编语言编制用 Clipper 实现不了的或实现起来很困难的功能,从而使用 Clipper 编程更加方便。

Clipper 是由用户决定如何使用的,很多 Clipper 程序设计人员开始用它来开发系统,因为他们可以把经 Clipper 编译后的可执行程序提交给用户,而无需为每个用户提供一个 Clipper 的系统的副本。又由于 Clipper 的可执行程序支持在扩展内存上操作,所以对大于 1M 内存的计算机更显优越性。用户只有在使用过程中才会逐步感到 Clipper 的优越性,才能体会到 Clipper 编译型关系数据库是当今在微机上使用的一种最理想的数据库管理系统。

作者在长期的科研和教学实践的基础上,总结了使用 Clipper 进行编程的经验和编程技巧,特别是本书中的第七、八、十、十二、十三等章的内容都是作者在实际应用中的范例或经验。

全书共分十六章,第一章为 Clipper 的基本概况,第二章从 dBASE 的简单应用引出 Clipper 的初步应用,第三章介绍了 Clipper 的基础,第四、五、六章介绍了 Clipper 的函数、命令和控制语句,第七、八、十、十二、十三章讨论了程序设计的方法和技巧,给出了数据库操作程序设计、菜单程序设计、文件处理及操作、与 C 和汇编的接口的设计以及网络程序设计等,第九章介绍了 Clipper 的编译连接,第十一章给出了 Clipper 的错误处理,第十四章介绍数据库驱动程序和终端驱动程序,第十五章介绍了 Clipper 工具,第十六章介绍了 Clipper 的对象类。

本书在编写过程中得到了浙江大学计算机系软件研究所同事们的支持,浙江大学计算与信息中心办公自动化研究室的金龙、袁清、李小东等同志为本书编写作出了贡献,在此表示衷心的感谢。

由于编写时间仓促以及限于水平,书中如有错误和不妥之处敬请批评指正。

金连甫

1995 年 2 月于浙大求是园

目 录

第一章 数据库的基本概况	1
第一节 数据和数据处理	1
第二节 Clipper 数据库简介	3
第三节 dBASE III plus 与 Clipper	4
第四节 Clipper 版本及安装	11
一、Clipper 软件版本	11
二、Clipper 5.2 软件安装	12
三、对国家语言的支持	18
四、Clipper 5.2a 的更新安装	23
五、Clipper 5.2b 的更新安装	24
六、Clipper 5.2c 的更新安装	28
七、Clipper 5.2 汉字驱动程序安装	30
第五节 联机资料	31
第二章 Clipper 应用初步	35
第一节 Clipper 开发环境	35
一、Clipper 软件开发环境	35
二、Clipper 数据库操作环境	35
第二节 一个简单的应用实例	38
一、建立数据库结构	39
二、输入数据	43
三、数据库的备份复制	45
四、列数据清单	45
五、数据库编辑	47
六、删除记录	50
七、计算和替换	51
八、查询和检索	52
九、累加和统计	53
十、dBASE 程序设计	54
第三节 Clipper 程序设计举例	55
第三章 Clipper 的基础	58
第一节 Clipper 的系统规格	58
第二节 Clipper 文件的类型和用途	59
第三节 Clipper 规则	61
第四节 代码块	63

第五节 表达式	65
第四章 函数	70
第一节 数学运算函数	71
第二节 字符操作函数	75
第三节 日期时间函数	82
第四节 数组操作函数	87
第五节 转换函数	96
第六节 测试函数	102
第七节 设置函数	125
第八节 数据库操作函数	134
第九节 网络函数	151
第十节 二进制文件操作函数	154
第十一节 文本字符串操作函数	159
第十二节 输入函数	166
第十三节 输出函数	172
第十四节 代码块操作函数	188
第十五节 支持 RDD 函数	193
第五章 Clipper 命令	204
第一节 屏幕操作	204
一、清屏	204
二、@... BOX 命令	206
三、@... SAY... GET 命令	207
四、保存、恢复命令	211
五、菜单命令	212
六、控制台、打印机输出	215
第二节 输入操作	217
一、键盘输入	217
二、读命令	221
第三节 数据库操作	223
第四节 内存变量操作	249
第五节 参数设置	252
一、系统环境设置	252
二、日期格式设置	255
三、终端 I/O 设置	256
四、GET 设置	261
五、数据库设置	265
第六节 其它文件操作命令	267
一、DOS 文件命令	267
二、处理操作	269
三、报表、标签	272
第六章 Clipper 控制语言	274

第一节	分支语句	274
第二节	循环语句	276
第三节	变量作用域	280
第四节	函数和过程语句	286
第五节	过程和函数的使用	293
第六节	INIT 过程和 EXIT 过程	299
第七章	数据库操作程序设计	304
第一节	数据库的建立	304
一、	数据库文件的结构	304
二、	库结构的建立	305
第二节	数据库的基本操作设计	308
一、	数据库数据的录入	309
二、	数据库信息的修改	312
三、	数据库信息的查询	314
四、	打印报表	316
第三节	数据库全屏幕操作技巧	318
一、	DBEDIT 的使用	319
二、	BROWSE 的使用	321
三、	使用程序设计实现全屏幕操作	321
第四节	数据库操作实用程序	330
一、	建立索引文件实用程序	330
二、	重建索引文件实用程序	331
三、	显示索引关键字实用程序	332
第八章	菜单程序设计	334
第一节	菜单程序设计命令	334
一、	菜单程序设计主要命令	334
二、	菜单程序设计辅助命令	337
第二节	菜单程序设计技巧	338
一、	横式菜单设计	338
二、	利用@ ... PROMPT 命令实现竖式菜单设计	340
三、	利用 ACHOICE () 函数实现竖式菜单设计	342
四、	下拉式菜单程序设计	344
五、	按钮式菜单设计	348
第三节	联机系统帮助的设计	352
第九章	Clipper 的编译连接	354
第一节	Clipper 程序设计基础	354
第二节	Clipper 编译	362
第三节	Clipper 连接	367
第四节	RMAKE 工具	378
第五节	编译指令	385

第十章 文件处理及操作	393
第一节 二进制文件处理	393
一、二进制文件操作函数	393
二、二进制文件处理实例	395
第二节 文本处理	396
一、文本处理函数	396
二、文本操作实例	398
第十一章 Clipper 的错误处理	411
第一节 错误捕获	411
第二节 使用 BEGIN SEQUENCE 结构	412
第三节 Clipper 的错误对象	414
第四节 Clipper 的错误信息	421
一、编译错误信息	421
二、RTLINK 错误信息	432
三、RMAKE 错误信息	438
第十二章 Clipper 与 C、汇编的接口	441
第一节 Clipper 与 C、汇编间的参数传递	441
一、以值的形式传递参数	441
二、返回参数给 Clipper	448
三、以指针的方式传递参数	452
四、内存分配	456
五、Extend. h 头文件	458
第二节 Clipper 与 C 的接口	461
一、C 函数模板	461
二、C 函数程序例	462
三、编译连接	464
第三节 Clipper 与汇编的接口	465
一、样本程序框架	466
二、有关 MASM 函数的例子	467
三、编译、汇编及连接	473
第四节 建立用户自定义库	473
一、用 C 语言编制的公共程序	474
二、用汇编语言编制的公共程序	501
三、用 Clipper 编制的公共程序	512
四、使用 LIB 管理库	522
第十三章 网络程序设计	523
第一节 网络程序设计基础	523
一、多用户程序设计	523
二、Clipper 的网络命令	525
三、开发网络应用程序	526
四、防止死锁	527

第二节 网络程序设计技巧	528
一、打开数据库文件程序设计	528
二、重建索引程序设计	532
三、加锁的原则	532
四、追加记录程序设计	533
五、修改记录程序设计	539
六、非数据库文件锁定的程序设计	542
七、多个记录加锁程序设计	542
第十四章 驱动技术	545
第一节 可更换数据库驱动程序	545
一、DBFCDX 驱动程序	546
二、DBFMDX 驱动程序	548
三、DBFNDX 驱动程序	549
四、DBFNTX 驱动程序	552
五、DBPX 驱动程序	552
第二节 替代终端驱动程序	554
一、ANSITERM 驱动程序	554
二、NOVTERM 驱动程序	555
三、PCBIOS 驱动程序	556
第十五章 Clipper 工具	557
第一节 调试工具	557
一、调试工具的启动	558
二、调试工具的菜单	559
三、调试器命令	561
第二节 DBU 实用程序	565
第三节 RL 实用程序	567
一、生成一个报表	567
二、建立标签文件	568
第十六章 Clipper 对象的类	570
第一节 Error 对象	571
第二节 GET 对象	574
第三节 TBrowser 系统	579
第四节 TBColumn 对象	584
附录 A ASCII 代码	586
附录 B INKEY () 函数的返回值	590
附录 C Clipper 函数一览表	594
附录 D Clipper 命令一览表	601

第一章 数据库的基本概况

第一节 数据和数据处理

人类活动的整个历史,离不开对信息和数据的收集、保存、利用和处理。在远古时代,人类仅能使用火光、语言和绘图等传递信息,通过结绳和刻划记录信息。文字、纸张和印刷术的发明,使人们能够超地域、超时间地传递和交换信息,促进了人类将信息大量积累并加以系统化和优化,逐渐形成知识。二十世纪以来,由于对电磁波的研究,找到了新的更先进的信息载体,发明了电报、电话以及广播和电视。人们不但可以超越空间在瞬间收听到千里之外的语言的音响信息,而且可以收看图像和文字。二次世界大战后诞生的计算机,性能价格比大幅度提高以及汉字信息处理已达到实用阶段,为高速高效准确地处理大量信息(包括中文信息、图像信息和声音)创造了良好的环境。

所谓数据处理,顾名思义,就是对数据进行加工处理。我们日常生活中接触到的如:实验数据、观测数据、统计数据、测量数据、库存数据等都是数据的例子。然而,数据处理中所说的数据还不止这些已经数值化的信息。其它诸如汉字形式、图像形式乃至声音形式的信息都可以看作为数据。数据处理任务就是对各种形式的数据进行收集、存储、传递、检索、分类、计算、加工及打印出各种报表或输出各种所需要的图形等。

数据处理并不是有了电子计算机以后才出现的。自 1890 年 Herman Hollerith 发明了第一台卡片制表机并用其编制人口普查以来,就已经开始了数据处理的历史。通常这种用机械设备来进行数据处理的系统称为机械数据处理系统(MDPS)。由于机械数据系统的存储数据的介质(如穿孔卡片)、计算速度以及处理能力的限制,效率很低,因而应用范围有限。由于电子计算机的出现,加上 60 年代以来存储数据的介质迅速发展,出现了磁带、磁鼓、磁盘乃至光盘等大容量的快速存储介质,使得数据处理的应用范围有了突飞猛进的发展。为了与机械数据处理系统相区别,人们把用电子计算机进行数据处理的系统称为电子数据处理系统(EDPS)。

从数据结构来看可分为五个层次,即字符、字段、记录、文件以及数据库。每一层次均为后一层次的组成部分。

- 字符(Character):一个数字(0,1,2,...,9)或一个字母(A,...,Z,a,...,z)或者一个其它的符号(. ; , \$...),这些字符在计算机中占据一个字节(byte)的存储单元。在汉字系统中字符也可以是 GB2312-80 字符集中的汉字或自定义的汉字,每个汉字一般占据两个字节存储单元。
- 字段(Field):它是最基本的不可分的有名字的数据单位。一个字段由一个或多个字符组成,组成一个字段的字符个数称为字段的长度(汉字时为汉字个数的两倍)。
- 记录(Record):关于一个个体的数据的总和。个体可以是人,也可以是物,甚至可以是一个概念。记录由若干个字段组成,组成记录的全部字段的长度的和称记录的长度。
- 文件(File):具有相同性质的记录的集合。一般来说,一个文件所含有的记录是同格式而

且等长的。

- 数据库(Data Base):数据库可以看成是有关某方面的所有文件的集合。譬如,在数据处理系统中,某个公司的全部文件的集合就可以构成一个数据库。但数据库的概念还不仅仅是指个别文件的和,它还蕴含着对文件的重新组织,以便最大限度地减少各文件中重复的数据库并增强数据文件之间以及文件的记录之间相互联系,实现对数据的合理组织与共享。

概括地说,数据库就是实现有组织地动态地存储大量关系数据并方便多用户访问的计算机硬、软件资源组成的系统。尽管数据库系统是在文件系统的基础上发展起来的,但与文件系统有着许多本质差别。在文件系统中,数据组织是面向应用的,文件之间无有机联系,而且不同文件之间往往有许多重复的数据。例如在人事档案中,像职工号、姓名、单位等数据大概是每个文件都必不可少的信息。这不仅造成存储空间的浪费,而且容易造成数据的不一致性。在数据库中,数据是从整体角度组织的,即数据的组织是面向数据的,数据文件之间存在联系,从而可反映出世界上事物之间的联系,数据重复少。所以节省存储空间、减少输入量并易于保证数据的一致性。数据库中的数据可为多个应用程序或用户所共享。在文件系统中,应用程序依赖于文件的结构,文件结构发生变化,程序必须随之改变,这就是所谓程序与数据之间不独立,会造成软件维护费用的增加。在数据库系统中,一般说来,程序与数据之间独立性较高,可降低维护费用。另一方面,数据库文件系统往往提供给用户简单易学的命令语言,所以用户使用数据库会感到比使用文件方便得多。但是与文件系统相比,一般说来数据库系统的效率要低一些。

数据库系统包含数据库和数据库管理系统两部分,两者之间的关系如图 1.1 所示。

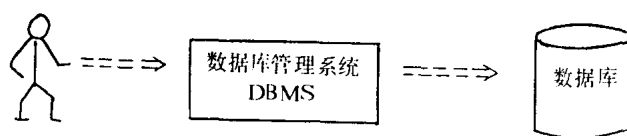


图 1.1

数据库管理系统的功能是建立数据库,对数据库中的数据进行分类、查找、检索、合并、增加、删除等多种操作。操作人员通过使用数据库管理系统来管理数据库。因此,一个数据库管理系统的好坏是衡量数据库系统优劣的重要因素。

各种不同的数据库系统所使用的数据结构的不同可分为三类:层次模型(Hierarchical model)、网状模型(Network model)和关系模型(Relational model)。

目前国内使用较多的是关系数据库,特别在微机上的 dBASE II、dBASE III、dBASE IV、dBASE III plus、FoxBase、Clipper、Unify、Oracle... 等等均为关系型数据库,并且这些数据库管理系统广泛应用于 MIS 和 OA 等的开发中,起到主导的作用。

关系数据库的结构,是把数据关系看成一个二维表,整个表叫做一个关系,赋予一个名字。关系模型在数学上有严格定义。我们这里不打算涉及关系模型的数学定义,从用户观点来看,关系就是二维表,表中的行称为元组,列称为属性。实际上,一个关系往往和一个文件相对应,一个元组就对应一个记录,一个属性对应一个字段。表 1.1 给出二维表格称为一个关系数据库文件,它有八个字段(职工号、姓名、性别、出生时间、工作时间、职称、婚否、工资)三个记录。

从表 1.1 可以看出,作为一个关系应满足如下条件:

- 表格中的每一项必须是基本数据项(不允许是重复组);
- 表格中的每一项必须具有相同的数据类型;
- 每列都必须冠以字段名,且在同一表格中字段名不允许重复;

- 表格中不应有相同的行(即相同的记录);
- 行的顺序与列的顺序均不影响表格中的信息内容。

表 1.1

职工号	姓名	性别	出生时间	工作时间	职称	婚否	工资
1001	李大名	男	27.05.21	49.10.01	高工	.T.	882.50
1002	张丽丽	女	38.10.05	60.08.15	工程师	.T.	725.00
1008	王小荣	男	60.10.12	83.08.15	助工	.F.	598.50
...	...	...	...	...	...	...	...

第二节 Clipper 数据库简介

Clipper 是一个编译型的关系数据库。最早由 Nantucket 公司针对 dBASE 各版推出的关系数据库管理系统。它经历了 85 年冬季版、86 年秋季版、87 年夏季版、88 年修定版,这些版本的功能对大多数用户而言大致作为 dBASE III 的编译器使用,只不过比 dBASE III 多增加一些功能而已。到 1990 年 Clipper 5.0 版上市后,这一情况才有所改观。1991 年推出了 Clipper 5.01,在功能上更强,所以使该系统在社会上得到广泛应用。1993 年由美国 CA 公司(Computer Associates International)推出 CA—Clipper 5.2 版,它与 Clipper5.01 版向上兼容,并且扩展和增强了很多功能,特别是系统提供了数据库驱动技术,允许 Clipper 访问和操作 dBASE III、dBASE IV、Foxpro、Paradox 等的数据库结构文件,紧接着又推出 5.2a、5.2b、5.2c 升级版。

众所周知自从 ASHTON—TATE 公司推出 dBASE II、dBASE III、dBASE III plus、dBASE IV 和 FOX 软件公司推出的 Foxbase 和 FoxPro 的一系列兼容的关系数据库,都以及 Borland 公司的 Paradox 系统产品,都以它们的巨大的灵活性、方便性、高效率 and 兼容性赢得了广大用户的欢迎,并广泛地应用于管理信息系统(MIS)和办公自动化(OA)的开发,为促进我国计算机的广泛应用起到了不可估量的作用,成功地广泛应用于企事业单位、银行、学校、商店等部门,促进了我国的管理科学化、规范化。但感到还有美中不足的就是上述数据库是解释型的,所以运行速度慢且不利于应用程序的管理。为了弥补上述数据库的不足,推出的基于 dBASE III plus 基础上扩充的编译型关系数据库 Clipper,它既保证与上述数据库 dBASE、Foxbase 等向上兼容,以及能直接对 dBASE III、dBASE IV、Foxpro 和 Paradox 数据库的访问和操作,又可以通过 Clipper 的编译器、连接器将其编译连接成一个可直接执行的 .EXE 文件的一个编译型的关系数据库。

用 Clipper 或 dBASE 语言写完一个程序后,就可以使用 Clipper 把它转变为一个能被计算机的磁盘操作系统(DOS)直接运行的独立的程序,不必为程序的每一个用户准备一个 dBASE III plus 的备份。如果需要有一份 dBASE III plus 或 dBASE IV 的备份以便为用户开发和维护数据库文件,却省下了为每一个用户机器装一个 dBASE III plus 备份的开销。一个 Clipper 应用程序是完全自含的。

Clipper 开发环境也是自含的。Clipper 系统提供了编写报表及创立数据文件的实用程序。你所需添加的只是一个好的文本编辑程序。dBASE 的备份只是一种便利设施。

编译 dBASE 程序的另一个原因是为了加强代码保密性,存放源代码的 ASCII 文件很容

易被打开和读出。这种存取的便利对开发应用程序很有益,但当敏感的方案、价格安排或工资信息需要作为程序的一部分时,就成了不利因素了。而一个编译过的程序几乎不可能被打开和读出。

Clipper 是 dBASE 语言的一种版本。Clipper 的编译程序接受的命令和函数是 dBASE 语言的超集。用户定义的函数、数组以及许多 dBASE IV 的新功能才刚刚具备 Clipper 程序设计人员运用已久的功能。最新的 Clipper 版本提供了嵌入式的预处理程序,使得用户可以建立用户定义的命令、条件编译,以完全控制 Clipper 环境;它还提供了一个新的连接程序,它动态地分配内存,这样即使多个非常大的程序也能在有限的内存上运行,这个新的连接程序的功能之一是它能够建立预先连接好的库(Pre-Linked Library)。PLL 是已连接好的程序代码文件,如果需要可以被装入内存,PLL 可以大到写满它们所在的磁盘,但在任何时候只有部份需要的代码被装入内存。这意味着你可以在一台只有 512KB RAM 的机器上运行一个 900K 的程序。

Clipper 提供的扩充系统(Clipper Extend System)已被加强以允许更容易地连接用 C 和汇编语言编制的例程和函数。简而言之,如果你发现 Clipper 语言中少了什么东西,就可以很方便地使用 C 语言或汇编语言把它们建立起来。

Clipper 还具有使用 dBASE III、dBASE IV、Foxpro、Paradox 数据文件的能力。除了这些功能外,Clipper 还包括一个强有力的调试程序。

所有这些功能都是为了一个目的:增强功能。Clipper 是由用户决定如何使用的。已经有很多的 Clipper 程序设计人员广泛使用该系统,因为他们可以把 dBASE 程序提交给用户而无需为每个用户购买一份 dBASE 副本。还有另外一些 Clipper 程序设计人员使用 Clipper 把那些不需要为使用多个工作站花费巨大开销的程序提交给连网的多用户。许多 Clipper 应用程序比标准 dBASE III plus 程序的编译后版本相差无几。但当你看完并明白 Clipper 的增强功能后,你就能够开始感到使用 Clipper 的优越性,就能体会到 Clipper 编译型关系数据库是当今微机上所使用的一种最理想的数据库管理系统,使用 Clipper 来编制新的应用程序甚至也可以将以前用 dBASE 或 Foxbase 开发的系统使用 Clipper 进行编译和连接成可执行的 EXE 程序。

本书以 Clipper 5.2X 为背景,介绍编译型数据库系统 Clipper 及其程序设计技巧。

第三节 dBASE III plus 与 Clipper

Clipper 是开发者的工具,与 dBASE 相比,它是以不同的计算机用户作为有服务对象的。dBASE 的目标是吸引广大的用户,从新手到有经验的程序员,而 Clipper 则不是面向所有人的。

通才只能在每一方面做一些工作,而专家则把时间花在一些更具体的问题上,这样他们能解决一些非常棘手的问题。dBASE 是通才;而 Clipper 是专家。Clipper 存在的目的只有一个——建立强有力的数据库管理应用软件。Clipper 没有与 dBASE IV 的控制中心或点提示符相对应的设施,它不是交互的环境,它是一种既包含简洁优雅又具有强大功能的程序设计语言。

由于 Clipper 没有交互方式和点提示符,许多 dBASE 命令动词不再存在,在大多数情况下,你不会想起它们。

大多数的全屏幕编辑命令不被支持——特别象 APPEND 和 EDIT。文本编辑程序和

BROWSE 命令是用 MEMOEDIT() 函数和 TBROWSE() 系统来实现。实用程序被用来提供纯粹的交互活动,如创建数据文件、报表和标签。

本节讨论的是 dBASE 和 Clipper 的区别。Clipper 是从 dBASE III plus 衍生而来,并且其能力在许多方面超过了 dBASE III plus。Clipper 引入了许多丰富的功能,其中的一些只是在 dBASE IV 中,而且其它的只有在 Clipper 环境下才能使用。

花些时间熟悉这些差别,特别对熟悉 dBASE、Foxbase 的程序员是特别有益的。这样当开始编写代码时,就会知道如何处理这些丢失的命令。下面部份给出以字母顺序排列的 Clipper 中没有的 dBASE III plus 的命令。

1. APPEND

APPEND 命令用于 dBASE III plus 中,允许用户在全屏幕编辑方式下把新记录加到文件尾部。在使用 Clipper 把记录加到数据库文件中通常用 @... SAY... GET 命令完成,@... SAY... GET 命令提供了对键入的数据的更准确的控制。

APPEND BLANK 命令和象在 dBASE III plus 中一样被使用。这一命令在文件的末尾加一个空记录以便填入数据。

2. ASSIST

ASSIST 命令在 Clipper 中不支持。

3. BROWSE

dBASE III plus 中的 BROWSE 命令,使用户可以存取数据库中的每一个记录和字段。Clipper 的面向对象的 TBrowse 系统或现在已过时的 DBEDIT() 函数可以用来提供一个功能更强大的 dBASE III plus 的 BROWSE 命令的功能。

4. CHANGE

CHANGE 是一个与 EDIT 命令相当的 dBASE III plus 命令。

5. CLEAR FIELDS

当使用任一全屏幕编译命令(BROWSE、EDIT 等)时,可以确定一张待操作的字段表。CLEAR FIELDS 命令取消由 SET FIELDS TO 命令建立的表。在 dBASE 程序列出一些字段的地方,每个命令行会包括一个字段表,或将待使用的字段装入一个数组。

6. CREATE/MODIFY LABEL

CREATE/MODIFY LABEL 命令可用 Clipper 实用程序 RL.EXE 取代,来建立一个在功能上与在 dBASE III plus 中建立的.LBL 文件。

7. CREATE/MODIFY QUERY

Clipper 中不使用查询文件或类似的功能。

8. CREATE/MODIFY REPORT

CREATE/MODIFY REPORT 命令可用 Clipper 实用程序 RL.EXE 取代,来建立一个在

功能上与在 dBASE III plus 中建立的 .FRM 文件。

9. CREATE/MODIFY SCREEN

屏幕格式在使用 EDIT 和 APPEND 语句的 dBASE III plus 开发环境中是非常有用的。dBASE III plus 程序员长期以来一直使用由 CREATE/MODIFY SCREEN 命令建立的 .FMT 文件把屏幕格式插入他们的程序中,这样使得屏幕格式化更为简单。对于 Clipper 通过把 @... SAY... GET 命令装入一个过程就可以做同样的事情。然后使用 SET FORMAT TO procname,不必考虑关闭一个格式文件。

10. SCREEN/MODIFY VIEW

在 dBASE III plus 中,一个窗口文件是用来把两个不同文件中的信息合并得象一个文件一样。实际上是在两个文件之间建立一种联系,这是通过以一个公共的字段把它们连接起来,并建立一个可以在其上显示两个文件的信息的屏幕来实现的。同样地,这对在点提示符状态下工作是尤其有用的。这一技术受到一定限制,因为不能建立新的记录。在编写一个程序文件时通常用 @... SAY... GET 命令来创建一个屏幕。可以通过在 GET 命令中指定别名来混合和匹配希望存储数的地方。例如:

```
@ 10, 5 SAY "Enter Vehicle mileage:" GET VEH-->miles
```

```
@ 12, 5 SAY "Enter tire condition:" GET MAINT-->tires
```

```
READ
```

使用这些代码,可以在正确放置文件指针后在同一时间内往两个文件中添加信息。使用 CLEAR GETS 代替 READ 命令将显示信息(其中的 GET 处于高亮度状态),但不接受输入信息。

11. DISPLAY/LIST FILES

DISPLAY/LIST FILES 命令实质上与 Clipper 支持的 DIR 命令一样。DIRECTORY() 函数提供了执行几乎所有目录操作的工具。可以处理隐含的和系统的文件。

12. DISPLAY/LIST MEMORY

DISPLAY/LIST MEMORY 命令通常用来调试程序。如需要显示一批内存变量和它们的值,使用 @... SAY 命令并一个一个地指示这些变量。Clipper 包括功能更强的调试程序。可以查看所有变量的状态,数据文件的状态以及程序的线索——这都是在应用程序运行时。

13. DISPLAY/LIST STATUS

在 dBASE III plus 交互式,DISLAY/LIST STATUS 命令用来确认哪些文件是打开的,哪些索引是激活的以及 DOS SET 命令对操作环境做了些什么。在 Clipper 中没有与之相当的命令,但大多数的信息可以使用 Clipper 命令收集并报告或是用调试程序来查看。

14. DISPLAY/LIST STRUCTURE

在 dBASE III plus 中,DISPLAY/LIST STRUCTURE 命令用来列出数据文件中所有字段和它们的类型。但是当编写程序时,很少需要为程序的操作者列出一个文件结构的表。在为

数不多的需要显示结构的情况下,可以使用 DBSTRUCT()函数做这一工作。

15. DISPLAY/LIST USERS

DISPLY/LIST USERS 命令被 dBASE III plus 系统管理程序用来控制网络环境下的程序执行。Clipper 不需要这一层额外的软件来执行网络上的用户函数。如果你的应用程序需要知道是谁在网络上,那么可以建立一个为每一个应用程序注册或退出的数据文件。那么任何应用程序都可以读这一数据文件并知道目前注册的是谁。

16. EDIT

dBASE III plus 的 EDIT 命令与 CHANGE 命令相当。这一命令不被支持,这是因为它象 BROWSE 一样,用户可以不受限制地存取所有数据,如需要让用户以表格形式存取数据,请使用 TBROWSE 系统。

17. ERROR()

在 dBASE III plus 中,可以使用 ERROR()函数来捕捉一些预料不到的事件,以便能产生一个反应。Clipper 允许用户修改运行时刻错误例程以满足应用程序的需要。

18. EXPORT TO

dBASE III plus 中的 EXPORT TO 命令把数据转移成一种能被 PFS:FILE 读出的格式。Clipper 提供了许多在 DOS 这一层次创建和处理文件的函数。可以使用这些函数为几乎任一种文件格式建立文件解释例程。

19. HELP

Clipper 没有提供一个标准的帮助实用程序,可以建立用户自己的上下有关的帮助屏幕,它们通过 F1 键被访问。

20. IMPORT FROM

IMPORT FROM 命令是与 EXPORT TO 相反的命令。

21. INSERT

在 dBASE III plus 中,INSERT 命令被用来把一个记录放在另外两个记录之间。这个命令有时候对维护作为查询表使用的小的未被标记的文件有用。在 Clipper 中,使用 APPEND BLANK 命令往记录中填入数据,然后再用 SORT 命令。

22. LIST HISTORY

dBASE III plus 有一个非常有用的功能叫做 HISTORY,它能记住在点提示下键入的最后 20 个命令。这样就可以用上、下箭头前后遍历这此命令。由于 Clipper 中没有点提示状态,这个命令也就没有必要保留了。

23. LOAD

在 dBASE III plus 中,可以使用 LOAD 命令运行外部例程,然后通过 CALL 调用。在