

Clipper

dBASE III 语言编译工具

使dBASE III语言程序的执行加快2~20倍
为dBASE III语言程序提供保密手段

夏 东 涛

高档微机协会技术交流部
中国科学院科海培训中心

一九八六年五月

目 录

第一章 Clipper的一般介绍	(1)
1-1 系统环境	(1)
1-2 Clipper的加密机构	(2)
1-2-1 制作硬盘备份	(2)
1-2-2 制作软盘备份	(3)
1-2-3 在单软盘驱动器上制作软盘备份	(3)
1-3 Clipper原盘上的文件	(4)
1-4 DOS环境	(5)
第二章 Clipper与解释dBASE III语言的异同	(7)
2-1 解释执行与编译执行的区别	(7)
2-2 Clipper对于解释dBASE III增强的功能	(9)
2-2-1 技术参数的限制更少	(9)
2-2-2 可由用户定义函数	(10)
2-2-3 一个“双亲”可有多多个“子女”的关系	(13)
2-2-4 允许调用已经编译或汇编的目 的码文件	(14)
2-2-5 备注字段的使用更加灵活	(14)
2-2-6 可以使用科学表示法	(15)
2-2-7 增强了十多个新函数	(15)
2-2-8 宏替换的使用更方便	(17)
2-2-9 为最终用户提供了两种方式的 联机帮助	(17)
2-2-10 增加了SAVE SCREEN/RESTORE SCREEN语句	(20)

2-2-11	新的循环操作语句	(20)
2-2-12	可向程序内传送DOS命令行 上的信息	(21)
2-2-13	功能键设置范围扩大	(21)
2-2-14	SET COLOR TO语句可与宏替换 配合使用	(21)
2-3	与解释dBASEⅡ编程语言的差别	(22)
2-3-1	部分表达式结构不同	(22)
2-3-2	与宏替换有关的语句表示不同	(22)
2-3-3	尚不支持的dBASEⅡ命令	(23)
2-3-4	以不同方式支持的dBASEⅡ命令	(24)
2-3-5	dBASEⅡ与Clipper差别一览表	(25)
2-3-6	几点补充说明	(27)
2-3-6-1	数据类型输出精度	(27)
2-3-6-2	关于@显示语句	(27)
2-3-6-3	关于APPEND...FROM语句	(27)
2-3-6-4	关于CALL语句	(28)
2-3-6-5	关于ALT-C	(28)
2-3-6-6	关于计数语句	(28)
2-3-6-7	命令行的延续	(28)
2-3-6-8	关于过程名和文件名	(28)
2-3-6-9	关于光标移动键	(29)
第三章	Clipper编译、连接功能的使用	(30)
3-1	一般步骤	(30)
3-2	系统提供的默认步骤	(31)
3-2-1	使用默认步骤	(32)
3-2-2	修改批文件	(33)
3-3	使用编译程序	(34)
3-3-1	基本的编译命令	(34)

3-3-2	编译命令的选择项	(34)
3-3-3	对指定的程序进行编译	(35)
3-4	连接目的码程序	(35)
3-4-1	Plink86的基本命令	(36)
3-4-2	Plink86的复盖选择	(38)
3-4-3	复盖模块与系统的关系	(39)
3-4-4	生成·EXE文件内的复盖模块	(40)
3-4-5	如何制作·OVL复盖文件	(41)
3-5	使用DOS的连接程序	(42)
第四章	Clipper调试功能的使用	(44)
4-1	几点说明	(44)
4-2	进入调试功能	(44)
4-3	调试功能选择项	(44)
4-4	Clipper的出错信息	(47)
第五章	Clipper的文件生成实用程序	(49)
5-1	实用程序调用的命令格式	(49)
5-2	实用程序使用的一般步骤	(49)
5-3	实用程序使用中的功能键定义与默认值	(50)
5-4	索引实用程序	(51)
5-5	索引实用程序源文件清单与注释	(53)
5-6	摘要实用程序	(61)
5-7	摘要实用程序源文件清单与注释	(63)

第一章 Clipper的一般介绍

Clipper编译程序是美国Nantucket公司于1985年推出的软件产品。主要提供给对dBASEⅢ编程已有经验的程序员使用。因此本手册亦假定读者已熟悉dBASEⅢ程序语言。

Clipper本身是一个完整的系统。可用于开发应用软件。其主要优点如下：

- 执行速度较之解释执行的程序快2~20倍
- 提供了源程序的保密手段
- 与dBASEⅢ的功能和命令大致兼容
- 有些功能较dBASEⅢ更强

一般来说，解释dBASEⅢ程序可以不加修改地经Clipper编译运行，但遇如下情况时需对应用程序加以修改：

- 该应用程序使用了若干条目前Clipper还不支持的解释dBASEⅢ的命令
- 用户打算给该应用程序增加若干Clipper独具的功能

1-1 系统环境

Clipper运行于IBMPC及其兼容机上，所需的最小配置为：
256k RAM

- PC/MS-DOS 2.00以上版本
- 最好具有一软一硬磁盘驱动器，使用双软盘驱动器也可以建议使用8087(80287)浮点运算协处理器。

经编译、连接后得到的可执行应用程序可不仅仅在IBMPC及其兼容机上运行，只要具备以下条件的环境均可运行：

256k RAM。由于采用了复盖技术，因此在256k RAM系统中可运行非常大的程序

· PC/MS-DOS 2.00以上版本

· 采用 ANSI(美国国家标准信息交换码)标准的终端, 包括键盘、显示器和打印机

1-2 Clipper的加密机构

Clipper 原盘采用 Softguard System 公司的加密软件 Superlok 的最新版本进行了加密。原盘本身可以执行, 并且还可以生成四个可执行但不可再复制的备份, 这四个备份既可生成在软盘上, 也可生成在硬盘上, 分配任意。每个备份均可对类似 dBASE III 语言的源程序进行编译。为保险起见, 用户拿到原盘后, 应首先制作一个软盘备份, 并且使用这个软盘备份, 而将原盘存放起来。

经 Superlok 加密的软件有一个特点, 即所生成的备份可以被原盘“回收”。原盘上有一个计数器, 每制作一个备份, 计数器的数值加 1; 每回收一个备份, 计数器的数值减 1。

1-2-1 制作硬盘备份

(1) 将原盘插入驱动器 A 中

(2) 如果打算将 Clipper 系统程序装在子目录中, 则可由 MD 和 CD 命令生成并进入该子目录。

(3) 在系统提示符下, 打入:

```
A>CLIPCOPY C:H
```

这里 C 为硬盘驱动器标识符, H 表示驱动器类型为硬盘。

(4) 打回车键后, 系统开始制作备份。成功后, 显示出还可制作备份的次数。

(5) 如果不再需用此备份, 或打算将此备份移至另一硬盘上运行, 可打入如下命令进行回收:

```
A>CLIPBACK C:H
```

字母 C 和 H 的意义同前。

(6) 打回车键执行回收。成功后, 显示出还可制作备份的

次数。

装在硬盘上的Clipper备份是无法使用DOS的BACKUP和RESTORE命令进行存取的。(使用/P选择项也没有用)如果该硬盘打算重新格式化,应首先用Clipper进行回收,否则,该备份将被销毁。

1-2-2 制作软盘备份

此情况下,Clipper只能拷贝到软磁盘的根目录内。

(1) 原盘插入A驱动器中,B驱动器中插入一张格式化好的空盘。

(2) 打入命令

A>CLIPCOPY B:F

这里B是备份所在软盘驱动器标识符,F表示所用的是软盘驱动器。

(3) 如果不再需要此备份,则将原盘插入A驱动器,备份盘插入B驱动器,打如下命令进行回收。

A>CLIPBACK B:F

字母B和F的意义同上。

1-2-3 单软盘驱动器上制作软盘备份

如果所用的机器只有一个软盘驱动器,此时如需制作软盘备份,不可用Clippy命令,而应使用如下的命令:A>One-dr

事先准备好两张格式化好的空盘。第一张空盘用于存放加密的、可执行文件CLIPPER.COM。这文件是由命令One-dr自动装到备份盘上的。然后再用普通的COPY命令将原盘上的CREATE.EXE和LABEL.EXE、REPORT.EXE、*.OBJ拷贝到备份盘上。注意千万不可用COPY命令将原盘上的文件CLIPPER.COM也拷贝到备份盘上,因备份盘上的CLIPPER.COM文件是由系统自动作上去的。然后再将第二张原盘(实用程序盘)上的文件PLINK.EXE、*.LIB、*.PRG用COPY命令复制到第二张备份盘上,便得到了两张可用的备份

盘。实际制作备份时应按照屏幕上的提示去做。

1-3 Clipper原盘上的文件

Clipper原盘共有两张，一张称为系统盘，其上的CLIPPER.COM编译程序经最新版本的加密软件Superlok加密，故系统盘是不可拷贝的，第二张盘称之为实用程序盘。两张盘上分布的文件如下：

系统盘

文 件 名	长度	作 用
README.IST	13007	提供了本版本Clipper软件对手册内容的补充和修正
INSTALL.COM	50192	Superlok 的安装回收程序，备份盘上的CLIPPER.COM便是由INSTALL程序安装回收的，但用户不必直接使用此程序，可借助批文件 CLIPCOPY.BAT、CLIPBACK.BAT和one-dr.BAT进行制作、回收备份操作。
CLIPCOPY.BAT (DB3COPY.BAT)	943	用于生成可执行的Clipper 编译程序 CLIPPER.COM 备份的批文件
CLIP COPY.HLP (DB3COPY.HLP)	677	用于 CLIPCOPY.BAT 的帮助文件
CLIPBACK.BAT (DB3BACK.BAT)	401	用于从软盘或硬盘中回收 Clipper 编译程序加密备份的批文件
CLIPBACK.HLP (DB3 BACK.HLP)	327	用于CLIPBACK.BAT的帮助文件
ONE-DR.BAT	1526	在单软盘驱动器的系统上生成 CLIPPER.COM 软盘备份的批文件
CLIPPER.COM (DBASE3.COM)	2016	Clipper 编译程序

文 件 名	长 度	作 用
CREATE.EXE	65578	生成.DBF文件的Clipper实用程序
DEBUG.OBJ	14421	实现Clipper 调试功能的目标模块
ANSI.OBJ	2347	终端I/O符合ANSI 标准的 MS-DOS机器所用的驱动程序

实用程序盘:

PLINK86.EXE	64320	专为Clipper 设计的连接程序
CLIPPER.LIB (DBASE3.LIB)	257024	连接程序需用的系统库文件
OVERLAY.LIB	23040	生成复盖模块所需用的 系统 库文件
TOTAL.PRG	6791	仿真dBASE III TOTAL命令的 Clipper 实用程序源程序
INDEX.PRG	5310	Clipper 索引实用程序的源程序
SCI.PRG	797	实现科学表示法实用程序的源程序

(注): 括号中的DB3*.*、DBASE3*.*是国内开发单位所更换后的文件名。

1-4 DOS环境

数据库应用程序 较之其他应用程序对磁盘可用空间和内存可用空间更关注。因此作好 DOS 环境 准备, 可使此类程序的运行更有效, 且不易发生操作系统错误。

启动盘根目录中必须有CONFIG.SYS文件, 内容至少有:

FILES=20

BUFFERS=8

此版本的Clipper只支持20个打开的文件。但所有应用程序文件, 包括用到的复盖文件, 合起来只算作一个打开的文件。

如果在应用程序中使用 RUN 命令调动了外部程序和命令, 操作系统要求首先找到系统文件COMMAND.COM。如果找不到, 运行时屏幕顶部给出错误代码<1,2>。欲查看DOS 会在何

处查找文件COMMAND.COM，可在操作系统提示符下打入“SET”，然后回车。操作系统将显示出当前COMSPEC=参数。此参数描述了DOS重新进入命令处理程序COMMAND.COM的路径。可使用如下语句来使操作系统以另一条路径来寻找文件COMMAND.COM。

```
SET COMSPEC=d : \<路径> \COMMAND.COM
```

这里d是所给的驱动器标识符。

第二章 Clipper与解释dBASEⅢ 语言的异同

Clipper语言的语法和逻辑结构尽可能地模仿解释 dBASE Ⅲ。但由于两者本质的不同，总存在着一些不可避免的差别。因此在将解释dBASE Ⅲ的程序用Clipper编译程序编译之前应对程序进行适当的修改、处理。另外，如果用户打算利用Clipper 独具的功能，也需要修改解释dBASE Ⅲ的源程序。

用dBASE Ⅲ语言编写的程序应首先转换得与 dBASE Ⅲ的逻辑结构和语法一致，否则无法被Clipper所接受。这可由 dBASE Ⅲ的转换应用程序dconvert来帮助实现。

整套Clipper 软件由两部分组成。其一是一个编译器，它将 dBASE Ⅲ源码程序转换为目的码程序；其二是一个连接程序。它将编译得到的目的码程序转换为机器可执行的，EXE 文件。

编译后的程序较之解释执行的程序有以下几个明显的优点：

- 执行速度快
- 源程序保密
- 增加了一些功能

前两条是由于编译程序与解释程序的执行方式不同得到的。

2-1 解释执行与编译执行的区别

欲使得dBASE Ⅲ高级语言编写的程序为计算机“读懂”，首先需使用编译/连接程序或解释程序将其翻译成低级语言的程序，要检查其是否有语法错误，给出出错提示，并将可接受的源码翻译成机器可执行的代码。

解释程序工作时，对源码的每一行，逐行地进行出错检查和翻译。如果发现了问题。解释程序显示出错信息并请求当场纠

错。显然，在程序测试过程中能给出即时反馈有很大好处。

但是即便程序是完好的，用户也无法禁止无用的出错检查过程，亦无法加快翻译过程。用户程序每一次运行，解释程序都要对每一行源码进行重新检错、重新翻译。如果是进行LOOP操作，解释程序便对相同的程序行重复地进行检错和翻译。

编译程序可避免多余的检错和翻译过程，因而可大大加快执行速度。与解释程序一样，Clipper 首先对源码的每一行进行检错。并将正确的源码翻译成可执行的二进制代码。但其处理方式不同：

(1) 发出编译命令。Clipper 对源码进行编译，实行必要的翻译和出错检查工作。

(2) 如果编译过程获得成功，Clipper 将目的码存入称之为OBJ文件的另一个程序模块中。否则便是得到了出错提示，调试后再返回(1)。

(3) 发出适当的连接命令，需给出应用程序涉及到的所有程序，子程序和库文件。

(4) 如果连接过程获得成功，便生成一个可执行的。EXE文件。否则，给出诸如“File not found”等出错信息，改正后返回(3)。

(5) 可在操作系统下以命令方式运行该。EXE文件。该文件中包含了运行应用程序所需要的全部信息。经编译/连接后的程序，不再借助于dBASEⅡ，也无需源码，甚至无需 Clipper 备份盘，只需在PC/MS-DOS操作系统下打入文件名即可执行。

统计表明，编译后得到的程序运行速度较之等价的解释执行的程序快 2 到 20 倍。

应用程序研制者为保护源码不被泄露，常需花费很多人力、物力进行加密。但加密往往只是暂时的。并且，常有人可作出解密程序在几分钟内重构源码程序。例如BASIC语言用“..P”方法进行的加密便已为人所破译。

但是经 Clipper 编译/连接得到的目标程序—EXE文件,是无法借助于任何解密程序而被重构成源码程序的。因此 Clipper 大大加强了程序的安全、保密性。

2-2 Clipper对于解释dBASE III 增强的功能

Clipper 较之解释执行的 dBASE III 提供了许多加强的功能

- (1) 可使用多达64,000个有效的存储器变量
- (2) 可同时打开 250 个文件
- (3) 每个数据库文件可使用多达1024个字段
- (4) 可生成用户定义的函数
- (5) 对一个“双亲”文件可建立多个“子女”关系
- (6) 可调用外部程序
- (7) 可将备注字段内容作为字符串存入备注变量中
- (8) 可使用科学表示法
- (9) 可在DO...WHILE语句中使用宏替换
- (10) 可使用嵌套的或递归的宏替换

Clipper还为最终用户提供了帮助手段:

①在应用程序正常运行中,如果程序员按照本章后面介绍的格式编写一个HELP·PRG文件,则最终用户可在应用中按 FJ 得到联机帮助。

②当执行过程中出了错时,如果程序员按照本章后面介绍的格式编写了ERROR·PRG程序,可保护数据库中数据的完整性,将调试时有用的信息存盘,并向最终用户显示出适当的出错信息。

这些增强的功能不能用于解释执行的 dBASE III 之下。

2-2-1 技术参数的限制更少

Clipper 实际上不再限制每个数据库可用的变量、文件和关系的数目。下面给出 Clipper 与解释性 dBASE III 的参数比较,

- 多达 250 个文件可被同时打开

Clipper 允许程序员在编写应用程序时同时打开 250 个文

数据库参数	Clipper	dBASE II
每个数据库可使用变量个数	64,000	256
每个数据库的字段数	1024	128
可同时打开的文件数(虚拟)	250	15
可同时操作的关系数 (SET RELATION TO)	一个双亲可以有八个子女	一个双亲, 一个子女

件。这 250 个文件的限制包括了程序文件、数据库文件、索引文件、报告文件以及 DOS 操作系统需使用的文件。实际上, 在 DOS 环境下只允许同时打开 20 个文件, 但由于整个经 Clipper 编译、连接后的应用程序连同若干个复盖程序一起只算作一个文件, 因此对程序员而言具有了同时打开 250 个虚拟文件的能力。

在选择数据库文件时:

(1) 可由号码选择前十个数据库文件, 默认的别名分别为 A 到 J。

(2) 其他的数据库文件必须赋给一个别名。别名不可由数字打头, 但可以是字母后跟数字。

当内存空间不够用时, Clipper 将不需用的文件关闭, 在需用的时候再自动将其打开, 此过程对于程序员和用户均是透明的。

• 多达 1024 个字段。

借助于 Clipper 的 CREAT 实用程序, 可为每个数据库定义 1024 个字段。

2-2-2 可由用户定义函数

Clipper 支持 dBASE II 编程语言已经定义的全部函数。此外还允许用户定义自己的函数。函数一经定义, 便可用于程序中任何一处。用户甚至可以建立一个函数库, 从而可方便地用于各种程序中。

(1) 格式:

给函数一个名字
FUNCTION<名字>

定义参数

Parama,b

· <语句>

·

· RETURN(<函数值>)

(2) 例1:

下面的程序首先定义了取最大值函数;然后用其选出两个值中较大者

* func.prg

do main

return

* 函数定义

function max

param n1,n2

if(n1>n2)

return(n1)

else

return(n2)

endif

* 函数调用

procedure main

input"enter number 1:"to n1

input"enter number 2:"to n2

?max(n1,n2)

return

(3) 例2: 定义使数值型参数(变量或常数)以科学表示法将

储的函数，

函数名：SCIENCE()

调用方式：SCIENCE(<数值表达式1>，<数值表达式2>)<数值表达式1>给出被操作的数值参数，<数值表达式2>给出保留的小数点后位数。函数运算的结果是给出<数值表达式1>的科学表示：

```
FUNCTION SCIENCE
PARAM NUM,DPL
E=0
SET DECIMALS TO DPL
IF NUM>=0
    SGN=" "
ELSE
    SGN="-"
    NUM=NUM*(-1)
ENDIF
IF NUM<1
    DO WHILE NUM<1
        NUM=NUM*10
        E=E-1
    ENDDO
ELSE
    DO WHILE NUM>=10
        NUM=NUM/10
        E=E+1
    ENDDO
ENDIF
SN=STR(NUM)
SE=STR(E)
```

```

CTRA=1
CTRB=1
DO WHILE SUBSTR(SNC,TRA,1)=" "
    CTRA=CTRA+1
ENDDO
DO WHILE SUBSTR(SE,CTRB,1)=" "
    CTRB=CTRB+1
ENDDO
S=SGN+SUBSTR(SN,CTRA)+"E"+SUBSTR(SE,
CTRB)
RETURN(S)

```

2-2-3 一个“双亲”可有多多个“子女”的关系

Clipper 支持文件之间可以建立多个关系。第一个数据库文件称为“双亲”，与其有关联的数据库文件称为“子女”，它们之间的关系根据双亲文件的关键字表达式建立。实现的语句是：

```

SET RELATION(TO<关键字表达式>/REC-NO( )
/<数值型表达式>INTO<别名>)[,TO<关键字表达式>/
RECNO( )/<数值型表达式>INTO<别名>...]

```

(1) 在Clipper中，一个双亲可以有八个子女，即双亲与子女的关系可有八个之多。例如，双亲文件名为 DAD-FILE，它有两个子女文件BABY和TODDLER。

(2) 双亲中的关键字表达式与相关联子女的关键字表达式可以相同，也可以不同。例如，可将 DAD-FILE 文件的字段 STOCK-NO 与子女文件 TODDLER 的 TOY-NO 字段建立关系。Clipper 根据建立索引后的子女文件的内容(若子女文件未建索引，则根据其记录号)寻找匹配的记录。

(3) 双亲文件关键字表达式中的内容可以是子女文件关键字表达式内容的一个子集。例如，BABY 的字段 TOY-NO 内容前半部分与 DAD-FILE 文件之 STOCK-NO 字段的内容一样。