自动进行 DBT 与 TXT 文件转换的方法

华东电管局信息中心　陈导

一、问题的提出

目前，随着 d BASEⅢ用户的增多，用 Memo 字段存储有关信息的应用也趋于普遍。用过 d BASEⅢ或其相容软件的用户会发现，由于 Memo 字段与 DBT 文件的特殊结构，用户很难对它进行直观的操作。例如：

无法对整个 DBF 的 Memo 字段进行全屏幕编辑

无法对整个 Memo 字段的内容进行批量输入

这样，对于，某些需要打印或显示全部 Memo 内容，或需要全方位编辑 Memo 字段(如：块操作等)的用户带来不少困难。

二、问题的分析

d BASE 数据文件(＊·DBF)是一种自说明的数据文件，由自身结构说明框架和数据记录两部分组成。

现在以 d BASEⅢ为例：

1.结构框架

03H	年月日	记录个数	框架长度	记录长度	全 0
0　1	2 3 4	5 6 7 8	9　10	11　12	13　32

图为 *DBF* 文件说明部分存储结构

字段名	0	类型		宽度	小数位	全 0
0　10	11	12	13 14 15 16	17	18	19　32

图为 *DBF* 字段定义部分存储结构

值得一提的是，d BASEⅢ的框架长度不固定，它的计算公式是：

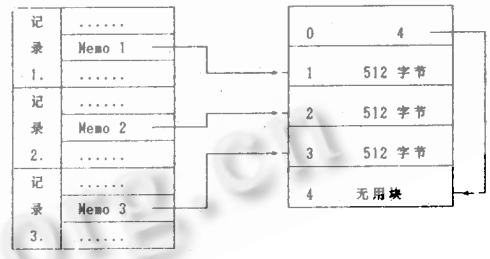
框架字节数＝文件说明部分字节数(32 个)×字段定义(32 个)+2 框架部分的最后两个字节是框架结束符 ODH，OOH。

2.数据记录

数据部分紧接框架部分存放，字符型数据项和数值型数据项的空位均以空格形式存放，各行记录以空格符相间隔，最后以 1 A 结尾。

在 d BASEⅢ中，用于存放 Memo 字段内容的是一个辅助文件(DBT)。在 DBF 中，每个记录的每个 Memo 字段提供一个指针，指向相应 DBT 中存放该信息的块(单位块容量为 512 字节)。每个 Memo 指针占 10 个字节，每个 Memo 字段最少占 512 字节，其内容也以 1 A 结尾。其中的第 0 块用于指向 dbt 文件末尾的下块位置，指针值占 0 块的前 4 个字符。

形如：



三、问题的解决

为解决 Memo 类型数据的直观操作与显示，本人用 Turbo Basic 编制了两个程序，目前运用在实际工作中效果很好。

dbt －txt：实现DBT文件转换为TXT文件(可用任何编辑软件进行编辑)

txt －dbt：实现TXT文件转换为DBT文件(可对Memo 字段进行全方位输送)

1 .dbt －txt：

设计思想：

(a)建立TXT文件，计算DBF中所有的Mwmo 指针

(b)逐记录，逐字段读 Memo 指针

© 中国科学院软件研究所 http://www.c-s-a.org.cn

(c)将Memo 字段结束标志1 A改为"#";若字段内容不满 512 字节,用空格补足。存入 TXT 文件中。

(d)重复(b),(c),直至 DBF 文件结束。

2.txt -dbt :

设计思想:

(a)建立TXT文件,计算DBF中所有的Memo 指针

(b)逐记录,逐字段读 Memo 指针

(c)以512 字节为单位读取TXT文件的内容,判别是否有标志"#",有:则转(e);没有:转(d)。

(d)将512 字节的内容存入Memo 指针指向的块中;Memo 指针+1 。转(f)。

(e)将标志"#"改为1 A;若内容不满512 字节,用空格补足。存入 Memo 指针指向的块中;并修改 DBF 中的 Memo 指针。转(f)。

(f)重复(b),(c),(d),(e),直至DBF文件结束。

在应用过程中,可以先对 DBF 运行 dbt -txt 程序,产生 TXT(由 DBT 转换得到),随后便可以利用现有的任一编辑软件,把 TXT 中的内容编辑成用户所希望的形式;最后,如果需要的话,运行 txt -dbt 把编辑过的 TXT 转换成 DBT。这样,当再次启动 d BASEⅢ,对其中记录发 edit 命令时,便可以看到所希望的 Memo 内容。同时,细心的用户还会发现,通过运行 dbt -txt 和 txt -dbt 后,DBT 文件中存放信息的块与块之间的是连续排列的,即起到压缩 dbt 空间的作用。

注意:运行 tzt -dbt 程序时,编辑过的 TXT 中,每个还未连到 DBT 中的 Memo 内容必须用"#"符隔开。

附:源 程 序 DBT- TXT · BAS和 TXT- DBT · BAS。

```
rem  ————————————————
rem  program 1 : dbt -txt · bas
rem  ————————————————
dim offset% (128 )
cls
input "please input · dbf :";f $
f1 $ =f $+"· dbf"
f2 $ =f $+"· dbt"
f3 $ =f $+"· txt"
open f1 $ for binary as #1
get $ #1 ,1 ,a $
if asc (a $ ) <>3 and asc (a $ ) <>131 then
   print chr $ (7 )+"not a · dbf file
end if
open f2 $ for binary as #2
open f3 $ for binary as #3
rem ————————
seek #1 ,4
get $ #1 ,1 ,n4 $
get $ #1 ,1 ,n3 $
get $ #1 ,1 ,n2 $
get $ #1 ,1 ,n1 $
Creco% =asc (n4 $ )+asc (n3 $ ) * 10+asc (n2 $ )
 * 100 +asc (n1 $ ) * 1000
get $ #1 ,1 ,n2 $
get $ #1 ,1 ,n1 $
Lstat% =asc (n1 $ ) * 10+asc (n2 $ )
get $ #1 ,1 ,n2 $
get $ #1 ,1 ,n1 $
Lreco% =asc (n1 $ ) * 10+asc (n2 $ )
seek #1 ,32
get $ #1 ,1 ,Sig $
n % =0
Cmemo%
Acc% =1
do until sig $ =chr $ (13)
n % =n %+1
seek #1 ,n % * 32+11
get $ #1 ,1 ,Typ $
seek #1 ,n % * 32+16
get $ #1 ,1 ,Num $
IF Typ $ = "M" or Typ $ = "m " THEN
    Cmemo% =Cmemo%+1
    offset% (Cmemo%) =Acc%
end if
Acc% =Acc%+asc (Num $ )
seek #1 , (n %+1 ) * 32
get #1 ,1 Sig
```


© 中国科学院软件研究所 http://www.c-s-a.org.cn

```
loop
rem ————————
Nreco% = 1
do until Nreco% > Creco%
   for i % = 1 to Cmemo%
    seek #1 , Lstat% + ( Nreco% - 1 )
            * Lreco%+offset% (i %)
    get #1 , 10, tps
    Ltp& = val (tps )
    if Ltp& > 0 them
     seek #2 , Ltp& * 512
     get #2 , 512 , rl
     wz% = instr (1 , r1, chr (26)
     do while wz% = 0
      put #3 , r1
      get #2 , 512 , r1
      wz% = instr (1 , r1, chr (26))
     loop
      r1 = left (r1, wz%-1 )+chr (13)+chr (10)
      +"#"+chr (13)+chr (10)
      put #3 , r1
     END IF
   next i %
   Nreco% = Nreco%+1
   loop
close
end

rem ————————————————
rem program 2 : txt -dbt · bas
rem ————————————————
dim offset% (128 )
cls
input "please input the name of · dbf : "; f
f1 = f +"· dbf "
f2 = f +"· dbt "
f3 = f +"· "
f4 = f +"· txt "
open f1 for binary as #1
get #1 , 1 , a
if asc (a ) < > 3 and asc (a ) < > 131 them
 print chr (7 )+"not a · dbf ! ! ! "
end if
open f2 for binary as #2
open f3 for binary as #3
open f4 for binary as #4
put #3 , space (512 )
rem ————————————
seek #1 , 4
get #1 , 1 , n4
get #1 , 1 , n3
get #1 , 1 , n2
get #1 , 1 , n1
Creco% = asc (n4)+asc (n3) * 10+asc (n2)
 * 100 +asc (n1) * 1000
get #1 , 1 , n2
get #1 , 1 , n1
Lstat% = asc (n1) * 10+asc (n2)
get #1 , 1 , n2
get #1 , 1 , n1
Lreco% = asc (n1) * 10+asc (n2)
seek #1 , 32
get #1 , 1 , sig
n % = 0
Cmemo% = 0
Acc% = 1
do until sig = chr (13)
n % = n %+1
seek #1 , n % * 32+11
get #1 , 1 , Typ
seek #1 , n % * 32+16get #1 , 1 , Num
IF Typ $ = "M" or Typ $ = "m "THEN
    CMemo % = Cmemo%+1
    offset% (Cmemo%) = Acc%
end if
Acc% = Acc%+asc (Num)
seek #1 , (n %+1 ) * 32
get #1 , 1 , sig
```


© 中国科学院软件研究所 http://www.c-s-a.org.cn

```
loop
rem ————————
Nreco% = 1
nd& = 1
md& = 1
do until Nreco% > Creco%
   for i % = 1 to Cmemo%
    get #4 , 512 , r1
    dw% = instr (1 , r1, "#")
    if dw% = 0 then
    put #3 , r1
    md& = md&+1
    nd& = 512 +nd&
    seek #4 , nd&
  else
    r1 = left (r1, dw%-1 )+chr (26)
    +space (512 -dw%)
    put #3 , r1
    seek #1 , Lstat% (Nreco%-1 ) * Lreco%
    +offset% (i %)
    a = str (md&)
    La % = len (a )
    a = space (10-La %)+a
    put #1 , a
    nd& = nd&+dw%
    md& = md&+1
    seek #4 , nd&
  end if
  next  i %
  Nreco% = Nreco%+1
  loop
a $ = mkl (md&)
a $ = len (a )
a $ = a $ +space $ (512 -a %)
seek #3 , 0
put $    #3 , a
close
kill    f2 $
KILL    f4 $
name f3 $       as f2 $
end
```

© 中国科学院软件研究所 http://www.c-s-a.org.cn