Clipper

A szerkezet szerkezete...

Az adatbázis-kezelők legfontosabb része az adatbázisfájl. Az ilyen állományok létrehozása azonban sok figyelmet és főképp időt igényel. Az alábbi rövid program nagyon jól használható a „mély vízben", vagyis munkahelyi körülmények között is.

A munkahelyen – mint tudjuk – minden tegnapra kell. Ha sok új – akár egymáshoz hasonló – dBASE struktúrát kell készítenünk, akkor idegesítően lassúnak tűnik a dBASE saját adatbázisszerkesztője. Valamennyi bevitel után Enter, majd egy rövid sípolás. Ha harminc mezőnél hosszabb fájlt szerkesztünk, akkor kész rabszolgának érezhetjük magunkat.

Megtehetjük persze, hogy a COPY STRU EXTE Clipper programmal készítjük az állományokat, de ilyenkor mindig új programra van szükség, amelyet újra kell fordítani, szerkeszteni. Arról már nem is beszélve, hogy mi történik akkor, ha esetleg hiba kerül a szerkezetbe, vagy valamit még javítani kell. Ilyenkor nincs más megoldás, újra kell kezdeni a fordítást, vagy marad a dBASE szerkesztő.

Ebben és a hasonló esetekben jó szolgálatot tehet a DBFDAT program, amely nemcsak a fenti gondokat orvosolja, hanem *mindkét irányban – text formátumra vagy fordítva – átalakítja a dBASE szerkezeti állományt.* Ezt követően már elegendő, ha kedvenc editorunkat (például a PE2-t) elindítva elkészítjük vagy papírról beírjuk a mezőinformációkat. És ha még az editor sor- és blokkmásolási lehetőségeit is kihasználjuk, akkor játszi könnyedséggel hozhatjuk létre dBASE állományainkat. A struktúrák szövegfájlba alakítása a dokumentáció elkészítésekor is jól jöhet.

A szöveges állományt az átmeneti fájl szerkezetéhez kell igazítanunk.

FieldNo:	Fieldname:	Type:	Width:	
1	FIELD_NAME	Char	10	
2	FIELD_TYPE	Char	1	
3	FIELD_LEN	Num	3	0
4	FIELD_DEC	Num	3	0
** Total **			18	

Egy szövegszerkesztett sornak tehát az alábbi módon kell festenie:

```
MEGJEGYZES  C     100 0
MENNYISEG   N       5 1
FIZDATUM    D       8 0
```

Az első oszlop itt a *mezőnév* (10 karakter), a második a *mező típusa* (1 karakter), a harmadik a *mező hosszúsága* (3 karakter), végül a negyedik a *tizedesek száma* (3 karakter).

A szerkesztett szövegállománynak *TXT* kiterjesztést kell kapnia. A program Clipper'87 és Nantucket Tools II segítségével készült. Munka közben mindvégig az egyszerűségre törekedtünk, a kód azonban így is mintegy 70 sornyi lett. A program csak a legszükségesebb hibaelhárításokat tartalmazza, de ez az adott feladathoz éppen elegendő. A fejlesztők így is jó hasznát vehetik.

Csizmazia István

A DBFDAT program forráslistája

```
*-----------------------------------------
*   Főprogram neve..........: DBFDAT.EXE
*   Szerző..................: Csizmazia D. István (C)
RamboSoft Works 1993.
*   Megrendelő..............:
*   A főprogram feladata....: dBASE struktúra oda-visszakon-
vertáló
*
*   Elkezdve................: 1991.08.18.
*   Paramétert kap..........:
*   Paramétert ad...........:
*   Hívója..................: A DOS
*   Az itt megírt függvények: - HELPIR
*
*   Megjegyzés..............: munkaközi segédprogramocska
*   Kell majd...............:
*
*
*-----------------------------------------
*

set curs    off                 && kurzor ki
set exact   on                  && pontos string egyezés
set scor    off                 && Clipper üzenetek kikapcsolása
setcancel( .f. )                && alt-C letiltása

*-- főprogram ------------------------------*
para pl

?'DBFDAT    dBASE & Text structure dualconverter
V:1.01'
?'If input file .DBF  -> convert to .TXT'
?'If input file .TXT  -> convert to .DBF'
?'Copyright (c)  Csizmazia István  1993.
1993.08.18.'

if pcount() < 1                 && ha kevés a paraméter
   ?'ERROR--> Not enough paraméter...'
   helpir()                     && usage & abort
endi

if ! file('&pl')                && ha nincs meg az input file
   ?'ERROR--> Missing input filename...'
   helpir()                     && usage & abort
end

*-- inputfile rövidalakjának készítése ------------*
pont = at( '.', pl )            && van-e a file-névnek kiter-
jesztése
if pont # 0                     && ha van, levágom
   rpl  = alltrim( left( pl, pont - 1 ) )
   kite = uppe( right( pl, 3 ) )
  else
   ?'ERROR--> Missing file extension...'
   helpir()                     && usage & abort
endi

*-- eset szétválogatás ----------------------*
do case
   case kite = 'DBF'            && dbf2txt irány
        use &rpl
        copy to #!@#@%@! stru exte
        use #!@#@%@!            && temporális állomány
        copy to &rpl sdf
        ?'O.K. The dBASE structure .TXT file is ready...'

   case kite = 'TXT'            && txt2dbf irány
        crea #!@#@%@!           && temporális állomány
        use #!@#@%@!
        appe from &rpl sdf      && paraméter txt beolvas dbf-be
        go top
        crea &rpl from #!@#@%@!
        ?'O.K. The dBASE data .DBF file is ready...'

   other
        ?'ERROR--> Incorrect parameter...'
        helpir()                && usage & abort
endc

filedelete( "#!@#@%@!.dbf" ) && meló után takarítás
?
close all                       && minden lezár
set curs    on                  && kurzor vissza
retu

*--- használati utasítás ----------------*
func helpir
   ?'USAGE: DBFDAT <filename.extension>'
   ?                            && soremelés
   ?chr( 7 )                    && pitty
   quit                         && a viszontlátásra
retu ( .t. )
```