

Computer

92. október

92. október

286-ról 386-ra

Generációváltás

Piac: Optikai tárolók

Adatok a présben

Atari rovat

Madárház Düsseldorfbán

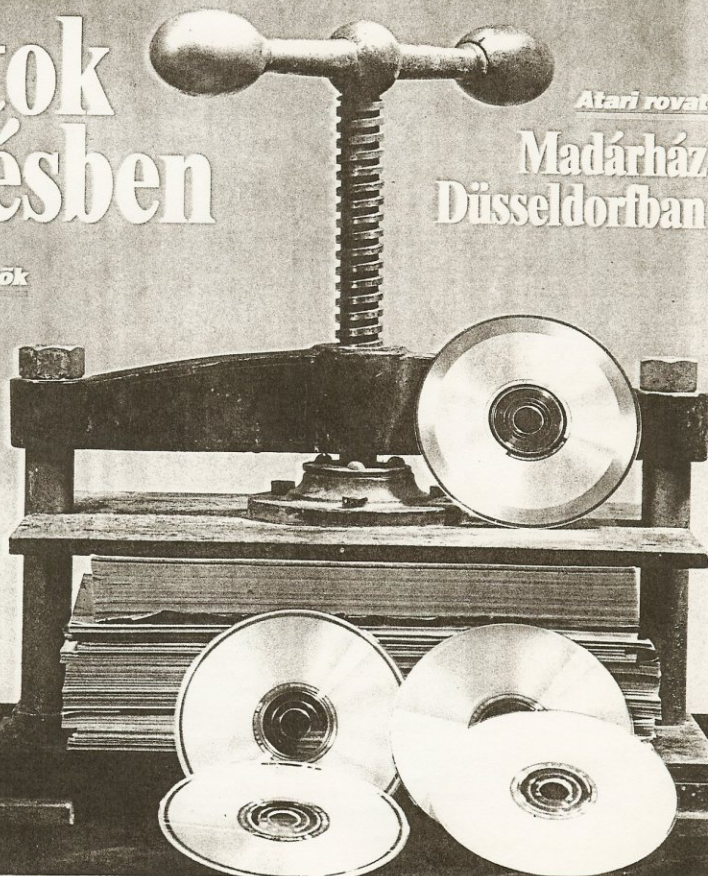
Windows trükkök

Pro memoria

AutoCAD

for Windows

Áthidaló megoldás



(Adjuk meg a gyűrű körülményeit)
 Diameter-around: of tube:
 (Adjuk meg a tengely körülményeit)
 Diameter-around: of tube:
 (Adjuk meg a cső körülményeit)
 Segments around tube circumference: of tube:
 (Adjuk meg a cső szegmenseinek számát)
 Segments around tube circumference: of tube:
 (Adjuk meg a gyűrű szegmenseinek számát)
 Az AutoCAD a körhöz körhöz közelebbi köröket használ.
 Ez az itt van a körhöz közelebbi.

WEDGE

E

Ez a program a szövegtárolással kezd, az egyik sorban a körhöz közelebbi köröket használ. A körhöz közelebbi köröket használ.

megy. Az AutoCAD a körhöz közelebbi köröket használ. Ez az itt van a körhöz közelebbi.

Comment WEDGE

Corner of wedge:

(Adjuk meg a körhöz közelebbi köröket)

Length:

(Adjuk meg az X irányú körhöz közelebbi köröket)

Width:

(Adjuk meg az Y irányú körhöz közelebbi köröket)

Height:

(Adjuk meg a körhöz közelebbi köröket)

Rotation angle about Z axis:

(Adjuk meg az Z irányú körhöz közelebbi köröket)

Nevezetesen a körhöz közelebbi köröket használ. Ez az itt van a körhöz közelebbi.

Dr. Szendrői Ágoston

(Fotó)

Vírusok

Naptár módszer

Vajon mindig új-e a friss dátumú VIRSCAN.DAT? Sajnos ugyanis a vírusok újabb verziói mindig megelőzik a víruskeresőket. Alábbi írásunkban erről a gondról lesz szó.

A fenti kérdésre egyetlen szóval válaszolhatunk: nem. A nehézségek feltárása előtt azonban beszéljünk kicsit a két fő vírusdetektorról, a SCAN-ról és a HTSCAN-ról.

A HTSCAN 1.15-ös verziója óta nem szükséges, sőt kifejezetten nem ajánlatos a tömörítők szignatúrájának hozzáadása a VIRSCAN.DAT állományhoz. Ezekből néhányat a HTSCAN.EXE is felismer, és tájékoztatást küld róluk:

Seems to be:

DISKTOOL.EXE seems to be compressed using PKLite.
 UNLZEXE.EXE seems to be compressed using LZexe.

A HTSCAN tehát – a SCAN-hez hasonlóan – két fő tömörítési módszert azonosít. Ha az állományban szerepelne a tömörítők szignatúrái, akkor a program ezeket – téves felismerésként – vírusként detektálna a memóriában, és például a memória krlite-lal fertőzött, kérem a gépet kikapcsolni! üzenettel reagálna. A tömörítők tehát nem véletlenül kerültek ki az állományból, visszatennünk ezeket semmiképpen nem érdemes.

Csak a teljesség kedvéért említjük, hogy a HTSCAN.EXE az alábbi észrevételeket teheti még:

BACKUP.COM has an EXE header.

ARC.BAT has an illegal date/time. 2056-10-23 00.00.00

Kis János: Új vírusleletek című könyvének 128. oldala szerint napjainkban mintegy 4000 vírusazonosító létezik. Sajnos nincs nagy különbség a különböző víruskeresők adatbázisainak tartalma között.

Ezt az alábbi rövid vizsgálattal is igazolni tudjuk. A DAT-CREA.C programmal létrehoztuk a szükséges darabszámú szignatúrát tartalmazó próba VIRSCAN.DAT állományokat a teszthez.

100 szignatúra AE 207"

1500 szignatúra AE 227"

Itt jegyezzük meg, hogy a HTSCAN egy heap memóriába helyezi a szignatúrákat, ezért a több elemből álló állomány nagyobb heap memóriát foglal le. A négyezer szignatúra inkább csak elméleti határ, amely talán az új DOS-változatokban, DOS=HIGH és DEVICE=HIGH esetén, „csupasz” gépen érhető el. 3200-as értéket produkáltunk egy 330-s MS-DOS operációs rendszerrel működő gépen, amelyben 589 Kb-át volt a szabad RAM. A vírusazonosítók hosszúsága 48 karakter volt.

Az új SCAN-változatokban – az /EXT opcióval – külső szignatúraállomány is használható. Ez azonban csak újdonságok tesztelésére vagy kísérletezésre alkalmas. A minták függvényében már komolyabb időtartam növekedés tapasztalható:

1 szignatúra → 106"

500 szignatúra → 228"

A beírtató szignatúrák száma fordítottan arányos a leírt azonosítók hosszával, az előbbieknél tehát egy fix memóriarészben kell elférniük.

Mivel a SCAN esetében – a VIRSCAN.DAT-tal ellentétben – nem látunk bele az azonosítók mintáiba, szélsőséges esetben például megeshet, hogy a SCAN-ben lévő amerikai MICHELAN-GELO változat alapszignatúráit véve a program ugyan felismeri a gépünkön lévő magyar átiratot, de a CLEAN rosszul fogja iratani a vírus, és ezzel minden vírussal katasztrofálisabb adatösszeomlást idéz elő.

Az mindenestre jó, hogy a HTSCAN folyamatosan karbantartható, a hozzá nem értők azonban ne barkácsoljanak bele, a szakmabeliek pedig az állomány kronológiai leírásában mindig dokumentálják a belenyúlás okát és dátumát. S még valami: a magyar kiegészítésekkel ellátott állományunkat nem tanácsos külföldre engedni.

A lehető legjobb azonban az, ha kompetens helyről szerezzük be az állományokat. Erre a VIRNET BBS (1154-402, 2400 Baud, 8N1, MNP5) a legalkalmasabb, a frissességet és a megbízhatóságot tekintve egyaránt.

Mivel már többször előfordult, hogy a dátum szerint friss VIRSCAN.DAT bővült ugyan, mégis hiányzott belőle néhány, az előző változatban szereplő azonosító, az időigényes kézi vázslás helyett közreadunk két apró utilityt.

Hogy a VIRSCAN.DAT állományt könnyen karbantarthatassuk, és a friss adatfájl összevethessük a régivel, ugyanakkor a VIRSCAN.DAT-ot ne felülírás, hanem mindig az előző állapotot kibővítve használhassuk, bemutatunk két rövid Clipper programot:

A DAT2DBF a VIRSCAN.DAT állományból dBase állományt, a kigyűjtött kommentekből pedig .TXT szövegállományt készít, végül SCAN.EXT néven külső szignatúraállományt hoz létre a SCAN külső állományának dzsokerezése kevésbé szabad. A konverztálások így csak azok a minták kerülnek át, amelyekben nem szerepel dzsokerkarakter ('%', '*', '?').

A létrejött dBase állományt abban segít, hogy az új VIRSCAN adatállományhoz jutunk, akkor könnyen áttekinthessük annak tar-

tálmát, vagy a fájlt egyszerűen összehasonlíthatjuk egy másik hasonló állománnyal.

A HT_EDIT programmal két DAT2DBF-fel készített állományt hasonlíthatunk össze. A két fájl eltérései (mindkét oldalról nézve) egy különbségállományba, a HT_DIFF.DBF-be kerülnek. Ha ezt ellenőrizzük és ez alapján bővítjük a VIRSCAN.DAT állományunkat, akkor biztos, hogy nem veszítünk szignatúra információkat, és a duplikátumok vagy az inkorrekt adatok is könnyen kiszűrhetők. Az azonos névvel, de más szignatúrával jelentkező, vagy az egyéb módon „összemosott” azonosítók „kivadásása”, korrigálása is lényegesen egyszerűbb lesz.

Figyelni kell arra is, hogy az adattáblákban kommentezett vírusleírások is lehetnek. Ezt a HT_TEXT.TXT állományból ellenőrizhetjük; ez a fájl az összes megjegyzést tartalmazza.

A bemutatott programok nem a legpontosabb módszerrel készülték, de működőképesek, és korrekt megoldást adnak a felvetett gondokra, ezenkívül ötleteket sugallnak. A hibakezelést szándékosan csak a szükséges és éppen elégséges mértékben alkalmaztuk.

Csizmáza István

A DAT_CREA alprogram listája

```

* _____
*
* Alprogram (függvény) neve: DAT_CREA
* Az alprogram feladata:..... Véletlen mintákat VIRSCAN.DAT file-
*      generál
* Elkezdve..... 1992.06.23.
* Paramétert kap..... hány darab szignatúra legyen
* Paramétert ad..... Errorlevel = 1, ha hiba miatt állt le
* Hívási rendje..... DAT_CREA <darabszám>
* Modell..... SMALL
* Fordítása..... BCC -ms -C -y -B -Z5 -w -G -lc:borlandc
*      include -Lc:borlandc\lib
* Hívója (DOS, CLIPPER)..... DOS
* Készítette..... Csizmáza D. István RamboSoft Works
* Az itt megírt függvények.....
*
*
* Megjegyzés..... Nincs ellenőrzés és hibakezelés, ez egy fej-
*      lesztői segédprogramcska.
* Kell majd.....
*
*
#include <conio.h> /* DOS console I/O függvények */
#include <ctype.h> /* Karakter konverziós makrók */
#include <clr.h> /* PATH nevel kezelő makrók és függvények */
#include <dos.h> /* DOS hívások deklarációi és definíciói */
#include <fcntl.h> /* OPEN() rutinhoz konstansok */
#include <io.h> /* Alacsony szintű I/O-hoz deklarációk és struktúrák */
#include <stdio.h> /* Kész adat beolvasási függvények */
#include <stdlib.h> /* Kétszámú használt rutink definíciói */
#include <string.h> /* string és memória kezelő rutink definíciói */
#include <time.h> /* Idő konverziós rutink és prototípusok */

static FILE *fp1;

void writefile(char outstring[])
{
    fprintf(fp1, outstring); /* kilírás a fileba */
    fputc(0x0D, fp1); /* sorvégjel chr(13) + chr(10) */
    fputc(0x0A, fp1);
}

/* főprogram kezdete */
void main(void)
{
    long i = 0; /* ciklusváltozó a randomszámszámhoz */
    long j = 0; /* ciklusváltozó a darabszámhoz */
    long darab = 0; /* a kívánt szignatúra darabszáma */

```

```

long a = 0; /* a randomszám változója */
char string1[80]; /* a file-ba kilírandó sor */
char seged[80]; /* a randomérték string konvertálásához */

darab = atoi(argv[1]); /* a kapott paraméter vizsgálata */
if (darab <= 0) /* ha hibás a paraméter, kilépes */
{
    printf("Hibaszint: DAT_CREA <darabszám>");
    exit(1); /* errorlevel 1 */
}

randomize(); /* random számok összerakása */
fp1 = fopen("vircan.dat", "wb"); /* file nyitás írásra */
writefile(string1); /* Próbá állomány a HTSCAN.EXE részére értéketlen mintákkal */
writefile(seg); /* Készült a DAT_CREA BC++ program segítségével */
writefile("Copyright (C) RamboSoft Works 1991-1992.");
writefile("");

printf("\n"); /* sorerekes a képernyőn */
for (j = 1; j <= darab; j++) /* ciklus a kívánt darabszámig, ha > 0 */
{
    printf("\nKészül: %d: %d", j, darab);
    strcpy(string1, ""); /* inicializálás */
    strcpy(seg, ""); /* inicializálás */
    strcpy(string1, "Yn Flew Ans A. virus #"); /* No comment */
    strcpy(seg, " "); /* fűző sorszám konvertálása stringgé */
    strcpy(string1, seg); /* konkatenálás */
    writefile(string1); /* fileba írás sorvégjellet */
    strcpy(seg, ""); /* inicializálás */
    strcpy(string1, "COM EXE SYS BOOT HIGH");
    writefile(string1);

    strcpy(seg, ""); /* inicializálás */

    for (i = 1; i <= 12; i++) /* szignatúrák készítése random módon */
    {
        a = 1000 + random(9000); /* trükk, hogy 4 jegyű legyen a szám */
        strcpy(seg, ""); /* inicializálás */
        strcpy(seg, a); /* konvertálás stringgé */
        strcpy(string1, seg); /* konkatenáció */
        /* for i = 12-ig vége */

        writefile(string1);
        writefile("");
    } /* for j = darab-ig vége */

    fclose(fp1); /* file lezárása */
    printf("\n\nCsao ragazzi!\n"); /* Elköszönünk... */
}

```

A DAT2DBF forráslistája

```

* _____
*
* Főprogram neve..... DAT2DBF
* Szerző..... Csizmáza D. István (C) RamboSoft Works 1991.
* Megrendelő.....
* Az alprogram feladata..... HTSCAN.DAT állományból dBASE adattáblát
*      készítése pl. két adattáblából összehasonlítással
*      hoz
* Elkezdve..... 1991.11.18.
* Paramétert kap.....
* Paramétert ad.....
* Hívója..... A DOS
* Környezet..... Clipper 37 + Nantucket Tools II.
* Az itt megírt függvények:
*
*
* Megjegyzés..... Nincs ellenőrzés és hibakezelés, ez egy fej-
*      lesztői munkaközi segédprogramcska
* Kell majd.....
*

```

```

*
*
* Structure for VIRSCAN.DAT
* Number of records: 4
* Last update: 1992.06.05.
* FieldNo: Fieldname: Type: Width:
* 1 RLENEV Char 12
* 2 NEVEK Char 40
* 3 COMEX Char 25
* 4 SZIGN Char 80
* ** Total ** 158
*
*
* be- és kimenő file nevek bekérde
para p1 && a dat file neve a paraméter

set score off && Clipper üzenetek kikapcsolva
setcancel(.f.) && Alt-C letiltva

sorveg = chr(13) + chr(10) && EOL jel (800A)
output = spac(8) && a kimeneti file neve

if pcount() < 1 && paraméter nélkül kiépés hibaüzenettel
? HASZNÁLAT: DAT2DBF <filenév>
quit
endi

if ! file( &p1 ) && nem létezik a DAT file
? HIBA! -> A(z) ' + upper(p1) + ' file nem létezik!'
? HASZNÁLAT: DAT2DBF <filenév>
quit
endi

?
screenor = row()

input = p1
ponthely = at( ' ', input )
if ponthely = 0
output = left( 8 )
else
output = left( input, ponthely - 1 ) + spac( 8 - ponthely + 1 )
endi
input = alltrim( input )
output = alltrim( output )

* HTSCAN adatfile-ből .TXT másolat készítése
!copy &input *.txt &mul
if ponthely # 0
filenév = left( input, ponthely - 1 ) + '.TXT'
else
filenév = input + '.TXT'
endi

* ideiglenes segéd adatállományok létrehozása programból
crea tempora
appe blank
repl field_name with 'filenév' && melyik DAT állományból jön
repl field_type with 'C'
repl field_len with 12

appe blank
repl field_name with 'nevek' && vírus nevek
repl field_type with 'C'
repl field_len with 40

appe blank
repl field_name with 'comex' && miket fertőzhet
repl field_type with 'C'
repl field_len with 25

appe blank
repl field_name with 'szign' && szignatúrák

```

```

repl field_type with 'C'
repl field_len with 80
commit

crea &output from tempora && kimeneti .DBF létrehozása
close data
erase tempora.dbf && átmeneti állomány letakartása

crea tempora
appe blank
repl field_name with 'sor'
repl field_type with 'C'
repl field_len with 80
commit
create _munka_from tempora && segéd .DBF létrehozása programból
close data
erase tempora.dbf

* szöveg állomány beolvasása .DBF-be
use _munka
appe from &filenév sdf
copy to _commo_for left( sor, 1 ) = ':' && előkészület a SCAN.EXT készítéshez

* meggyűzősek és üres sorok törlése
delete for left( sor, 1 ) = ':'
delete for sor = spac( 80 )
pack
commit

* szignatúrák
go top
delete for recn() / 3 = int( recn() / 3 )
copy for deleted() to _szign_
pack

* feltalálási helyek
delete for recn() / 2 = int( recn() / 2 )
copy for deleted() to _comex_
pack

* megnevezések
copy all to _nevek_
close all

sele 1
use &output alias output
go top

sele 2
use _nevek_ alias _nevek_

sele 3
use _comex_ alias _comex_

sele 4
use _szign_ alias _szign_
j = recn()
go top

wnev = spac( 40 ) && munkaváltók és változatához
wcom = spac( 25 )
wszi = spac( 80 )

* a kívánt output .DBF összeállítása
for i = 1 to j && ciklus a rekordszámig
@ screenor, 00 say 'dBASE file készült: ' + str( i, 8 ) + '/' + str( j, 8 )
sele _nevek_
wnev = sor
skip

sele _comex_
wcom = sor
skip

sele _szign_
wszi = sor

```

```

skip

sele output
go bott
appe blank
repl nevek with wnev;
comex with wcom;
szign with wsgi

next
repl all filenev with ('&input') && melyik .dat-ból jöttek az infók

```

```

* lezárunk
close data

* Scan data file-t készítsünk
scanfile = output + '.ext'
use &output
go top
handle = fcreate('&scanfile', 0) && írató/olvasható file készítése
handle = fopen('&scanfile', 2) && nyitás írató/olvasásra

```

```

?
k = 0
screenrow = row()
do while ! eof() && ciklus a file végéig
  wnev = nevek
  wcom = comex
  wsgi = szign
  * — csak a konkrét szignatúrát tudja átvinni
  if (! '% $ wsgi' .and. (! '?' $ wsgi) .and.
    (! "" $ wsgi) .and. (len(wsgi, 2) # '00'))

```

```

    k = k + 1 && számoljuk a megfelelőeket
    @ screenrow, 00 say "External file készült:" + str(k, 0) + '/' +
      + str(row(), 0)

```

```

    sor = ' ' + altrim(wnev) + ' vírus a' + altrim(wcom) + ;
    'ben támad' + sorveg
    fwrite(handle, sor, len(sor))

```

```

    sor = chr(34) + altrim(wsgi) + chr(34) + ' ' + ;
    altrim(wnev) + sorveg
    fwrite(handle, sor, len(sor)) && sor kírása file-ba
    endi
    skip && lépés a következőre
  endd
  fclose(handle) && file becsuk

```

```

* csak a komment sorok kiigazítása szövegfájl-ba
use _comm_
copy to ht_text.txt

```

```

* takarítunk magunk után
!del '?????'.dot -nul
!del &filenev >nul
@ screenrow, 00 say "
?
?Csao ragazsi! && elköszönünk
retu

```

A HT_EDIT forráslistája

- * Főprogram neve.....: HT_EDIT.EXE
- * Szerző.....: Csizmazia D. István (C) RamboSoft Works 1992.
- * Megrendelő.....:
- * A főprogram feladata.....: Két HT_DBF által készített DBF állományból különböző file létrehozása
- * Elkészítve.....: 1992.06.05.
- * Paraméterek kap.....:
- * Paraméterek ad.....:
- * Hívója.....: A DOS
- * Környezet.....: Clipper 87 + Nantucket Tools II.
- * Használt naplójel.....:

* Az itt megírt függvények:

* Megjegyzés.....: Nincs ellenőrzés és hibakezelés, ez egy teljesítmény - munkaközi segédprogramcska

* Kell majd.....:

```

* Structure for VIRSCAN.DAT
* Number of records: 4
* Last update: 1992.06.05.
* FieldNo: FieldName: Type: Width:
* 1 FILENEV Char 12
* 2 NEVEK Char 40
* 3 COMEX Char 25
* 4 SZIGN Char 80
* " Total " 159

```

para file1, file2 && a két összehasonlítandó DBF állomány

set score off && Clipper üzenetek kikapcsolva
set cancel(.f.) && Alt-C letiltva

```

if pcount() < 2 && kevés a paraméter
  ? "HASZNÁLAT: HT_EDIT <filename> <filename>"
  quit && kilépünk
endi

```

```

if ! file('&file1')
  ? "HIBA! -> A(z) ' + upper(file1) + ' file nem létezik!"
  ? "HASZNÁLAT: HT_EDIT <filename> <filename>"
  quit && kilépünk
endi

```

```

if ! file('&file2')
  ? "HIBA! -> A(z) ' + upper(file2) + ' file nem létezik!"
  ? "HASZNÁLAT: HT_EDIT <filename> <filename>"
  quit && kilépünk
endi

```

sele a && egyik állomány nyitása és teljes indexelése
use &file1
index on (nevek + comex + szign) to \$temp1\$

sele b && másik állomány nyitása és teljes indexelése
use &file2
index on (nevek + comex + szign) to \$temp2\$

```

* — a két állomány összekapcsolása és hasonlítás a B terület felől —
sele b
set rela to (nevek + comex + szign) into a
copy to $temp1$ for (nevek + comex + szign) != (a>nevek + a>comex + a>szign)
set rela to

```

```

* — a két állomány összekapcsolása és hasonlítás az A terület felől —
sele a
set rela to (nevek + comex + szign) into b
copy to $temp2$ for (nevek + comex + szign) != (b>nevek + b>comex + b>szign)

```

```

* — Indexek és összekapcsolások feloldása —
set rela to
set index to
* — a két különbségi segédállomány összemérése —
use $temp2$
copy to ht_diff
use ht_diff
appe from $temp1$

```

```

* — lezárunk, takarítunk
close all
!del '?????'.dot -nul
?Csao ragazsi! && elköszönünk
retu

```