Hard-Disk-Management
im Novell-Netz des Rechenzentrums im TGM

Walter Riemer, N/NA, TGM

Eines der vielen Computer-Labors im Bereich der Abteilungen Nachrichtentechnik im TGM trägt noch heute den stolzen Namen "Rechenzentrum", weil dort früher eine Siemens-Rechenanlage mit dem Betriebssystem BS 2000 stand. Im dortigen Novell-Netz mit 25 Arbeitsplätzen findet insbesondere der Anfangsunterricht in EDV und Informatik statt. Aus pädagogischen Gründen ist dort nicht jedem Arbeitsplatz eine Benützerkennung zugeordnet (wie in allen anderen vernetzten PC-Sälen), sondern jeder Benützer (Schüler und Lehrer) hat eine individuelle Benützerkennung, die sich bei Schülern aus der Jahrgangsbezeichnung und der zweistelligen Katalognummer zusammensetzt, zum Beispiel 3AN07, welche natürlich durch ein Password abgesichert ist.

Im Zuge des LOGIN wird dem Benützer ein ihm zugeordnetes Unterverzeichnis auf der "Daten-Festplatte" des Servers U: (U für User) als Laufwerk I: (individuelles Laufwerk) ge"mapt", zum Beispiel `U:\3AN\3AN07`. Dort und nur dort hat er alle Rechte wie auf einem ihm gehörenden Laufwerk in einem stand-alone-Computer, allerdings begrenzt auf maximal 1,2 MB Speicherkapazität.

Diese Konfiguration bietet den pädagogisch wertvollen Vorteil, daß die Schüler mit dem ihnen zur Verfügung stehenden Massenspeicherplatz umgehen lernen, trotzdem aber auch das Sichern ihrer Daten und Programme auf einen zweiten Datenträger (Diskette) nicht vernachlässigen dürfen, da der Fortbestand der User-Daten nicht garantiert wird (etwa bei Festplatten-Crash).

Unter Berücksichtigung der Erfahrung, daß nicht alle Schüler die ihnen zur Verfügung stehende Kapazität von 1,2 MB wirklich ausnützen, ist das Produkt der Anzahl der Benützer (ungefähr 600) mit 1,2 MB wesentlich größer als die Kapazität der User-Festplatte. Alle Schüler dürften und könnten also nicht je 1,2 MB abspeichern.

Solange überwiegend PASCAL unterrichtet wurde, bewährte sich dieses System sehr gut, da selten mehr als Quellfiles (.PAS) und Backup-Files (.BAK) auf I: bestehen blieben. Seit Beginn des Schuljahrs 1993/94 wird jedoch auch im Anfängerunterricht ausschließlich C unterrichtet. Dies hat den unangenehmen Nebeneffekt, daß laufend auch .OBJ- und .EXE-Files entstehen, von denen insbesondere die letzteren oft 20 oder 50 oder noch mehr kilo-Bytes groß sind. Da die Gesamt-Plattenkapazität nicht erlaubt, daß jeder Schüler tatsächlich 1,2 MB ausnützt, kann man das dadurch gelegentlich erforderliche Aufräumen auch nicht den Schülern überlassen (wenn das auch pädagogisch von Wert wäre). Ohne entsprechende Maßnahmen wäre die User-Festplatte auf jeden Fall nach etlichen Wochen voll, wobei aber die meisten Dateien (.BAK-, .OBJ- und .EXE-Files) unnötig wären.

Es erwies sich daher als notwendig, das Aufräumen zu automatisieren. Das LOGOUT wird daher mittels einer Batch-Datei vorgenommen, welche zunächst das nachstehende Programm `KILLEOBX.EXE` mit dem Parameter I: aufruft und dann erst LOGOUT ausführt.

Das Programm enthält einige interessante Besonderheiten und wird deshalb hier veröffentlicht:

Im Hauptprogramm wird nach vorhergehendem Einstellen des als Kommandozeilenparameter angegebenen Wurzelverzeichnisses, von dem an das Löschen der .BAK-, .OBJ- und .EXE-Files erfolgt, das Unterprogramm SubDirs aufgerufen. Dieses hat nun die Aufgabe, sich durch alle eventuell noch angehängten Unterverzeichnisse hindurchzuarbeiten. Dies wird auf rekursivem Weg bewerkstelligt, das heißt, das Unterprogramm `SubDirs()` ruft sich selbst immer wieder auf, nachdem es im gerade aktuellen Unterverzeichnis alle passenden Dateien desselben Tages gelöscht hat (Aufruf der Funktion `KillFiles()`). Dann wird geprüft, ob es noch ein angehängtes Unterverzeichnis beliebigen Namens (*.*) gibt; wenn nicht, dann wird zum aufrufenden Programm zurückgekehrt. Andernfalls werden alle Unterverzeichnisse nacheinander (mittels `do-while`-Schleife) bearbeitet und dabei insbesondere `SubDirs()` selbst immer wieder (rekursiv) aufgerufen.

Die Funktion `KillFiles()` eruiert zunächst für jede Datei einen Handle, um damit Zugriff zum Dateidatum zu erhalten. Das Dateidatum wird anschließend mit dem aktuellen Datum verglichen; im Fall der Übereinstimmung wird ein DOS-Befehl `DEL` aufgebaut und die Datei mittels der `system()`-Funktion gelöscht. Anschließend wird noch eine entsprechende Meldung auf den Bildschirm geschrieben (die allerdings im Netz durch den Operanden `>NUL` im Batch-File unterdrückt wird).

Die Funktion `RootEinstellen()` übernimmt den Kommandozeilenparameter vom Hauptprogramm und stellt das gewünschte Verzeichnis ein, gleichzeitig wird auch die Zeichenkette für den in der Funktion `KillFiles()` auszuführenden DEL-Befehl vorbereitet.

```
KillEOBX.C
löscht alle Dateien mit Erstellungsdatum = heutiger Tag aus
dem angegebenen Verzeichnis sowie allen untergeordneten
Unterverzeichnissen.
Aufruf: KILLEOBX wurzelverzeichnis
Beispiel KILLEOBX C:\TC2
```

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <dos.h>
#include <dir.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>

#define IsSubDir 0x10

struct date Heute;

void SubDirs (void);
void KillFiles(void);
void RootEinstellen(char *RootPfad);

void main (int KdoCount, char **KdoZeile)
{ int Zaehler;
  char RootPfad[MAXPATH];
  getdate(&Heute);
  if (KdoCount == 2)
  { strcpy (RootPfad, KdoZeile[1]);
    strupr(RootPfad);
    RootEinstellen (RootPfad);
    SubDirs ();
  }
  else
     printf ("Aufruf: KILLEOB rootDirectory\n");
  puts("\n");
}

void SubDirs (void)
{ struct ffblk FFStruktur;
  KillFiles();
  if (findfirst("*.*", &FFStruktur, IsSubDir))
     return;
  do
  { if (FFStruktur.ff_attrib != IsSubDir ||
        !strcmp (FFStruktur.ff_name, ".") ||
        !strcmp (FFStruktur.ff_name, "..") )
       continue;
    chdir (FFStruktur.ff_name);
    SubDirs ();
    chdir ( ".." ) ;
  }
  while (!findnext (&FFStruktur));
  return;
}
```

Schluß des Beitrags auf Seite 78

```
TSCRIPT4.ZIP    51964  890630  Version 4 Word Processor with Documentation.
TSPEECH.ARC     14930  881203  Turbo Pascal sources for PC voice routines
TTT5.ZIP       335438  890612  Turbo TechnoJock Toolkit Ver. 5.0 for TP V4 & 5
TTTDOC.ZIP      34929  890630  Documentation for TTTTPU - Turbo Pascal Unit.
TTTTPU.ZIP      37733  890630  TP Toolkit Unit - Menus, Popups, Mouse, More!
TUFPRINT.ZIP     2963  890630  Printer Error Control Routines.
TUR6_101.ZIP   102778  890630  Enhancements to TP that Use External Asm Lang
TUR6_102.ZIP   160171  890630  TP Utils: File Compression, TSR, Memory Managmnt
TURB_INC.ZIP    11229  890630  Inkey,Power,String /Lcase/Upcase/Trim/Pad/Justfy
TURBARG.ZIP      2631  890630  Routines to Parse & Get Cmd Line Arguments
TURBCOMM.ZIP    54115  890630  Communications Routines for the DEC Rainbow.
TURBDIR.ZIP      2242  890630  File Directory Program.
TURBHELP.ZIP    24524  890630  On-Line Turbo Pascal Version 3.0 Help Files.
TURBO.ZIP      161436  890630  Ninety-Three Turbo Pascal Programs of All Kinds.
TURBO30.ZIP      3670  890630  Comments on V3.0 & How to Use an Intel 8087
TURBO9.ZIP      96406  890630  70 Disk/Star/World/IO/Heap/Dump/Draw/Sort/Xref.
TURBO_SQ.ZIP    13364  890630  Squeeze/UnSQueeze Utility Written in TP.
TURBO_UT.ZIP    28649  890630  Miscell Graphics, I/O Routines with Docs.
TURBOBUG.ZIP     3530  890630  Subtraction Bug in Turbo Version 2.00 & Fix.
TURBOCLR.ZIP     3865  890630  Color Patch Correction to Turbo Version 2.00B.
TURBOFIX.ZIP     1062  890630  Changes TP COM Files for: Clr Scrn at Pgm Start
TURBOHLP.ZIP    24524  890630  Set of Online Help Files While Using TP.
TURBOINT.ZIP    11988  890630  Keyboard Read Routine for Keyboard Scan Codes.
TURBOIO.ZIP     33651  890630  Turbo Pascal Input/Output Routines.
TURBOP.ZIP      50091  890630  Toolbox ACCESS >64K Records, Cmd Line Parse,More
TURBOPIC.ZIP     1382  890630  V.2 Cnvrts BASICA Bload Screen to Memory Images.
TURBOPM.ZIP     50091  890630  Misc Procs - Xref,DB Access,Driver,Intr24,Etc.
TURBOPR2.ZIP    31467  890630  Pretty Printer Src Code Formatter & Xfer Prgrm.
TURBOPRT.ZIP    10576  890630  Print Pascal File, Expands Include Files
TURBOPW.ZIP      3749  890630  Signon Password Program.
TURBOSCR.ZIP    18198  890630  TP Code Screen Gen from Inbedded Special Chars.
TURBOTUT.ZIP   168100  890630  Turbo Pascal Version 3.0 Tutorial Package.
TURBOUT2.ZIP    59589  890630  Screen, I/O, Box, Menu Handling Routines w/Docs.
TURBOWHL.ZIP   137151  890630  Eighty File and I/O Routines with Documentation.
TURBSC.ZIP       8417  890630  Creates Source Code for Input Screen with Docs.
TURBSERL.ZIP     3645  890630  Serial Routines to Access COM1 and COM2 Ports.
TURBSTRG.ZIP    11172  890630  String Handling, Power, Inkey, with Docs.
TURBUTIL.ZIP    19543  890630  Window/Payment/Screen/Encrypt/Serial/Time
TURMEN.ZIP      15117  890630  Menu of Disk Input/Output Routines.
TURPRT16.ZIP    14750  890630  Pretty printer for Turbo Pascal source code.
TURUTI.ZIP       3237  890630  File Append/Check Kbd Locks/Other General Routns
TUTORPAS.ARC   202811  890919  Jack Crenshaw's compiler tutorial w/Pascal src
TVG102_S.ZIP    38416  911026  TVGRAPH 1.02: TURBO VISION In EGA/VGA: sources
TXREF1A.ZIP      6386  890630  Cross Reference and Pretty Printer List Program.
TXT2RND.ZIP      1321  890630  Conv Text File to 80 Char Rec File for Rndm Acc.
U.ZIP            3781  890630  Convert TP's Reserved Words to Upper Case Chars
UPCONV14.ZIP    16848  901205  Convert Pascal reserved word case w/TP5.0 src
USING.ZIP        2489  890630  Print Using Like BAS $###,###.## Routine in TP.
UTIL.ZIP        19331  890630  52 Scrn, Pwr,Ser,Encrypt, Menu, Window Routines.
V1DIR.ZIP        7869  890630  Directory & Char Matching Routines w/Docs
VALIDFIL.ZIP     6388  890630  Check for Valid Input Filenames w/ Doc, TP Src.
VDEL.ZIP         5736  890630  Verify Before Deleting Files.
VIDEO_IO.ZIP     5287  890630  Direct Vid On/Off/Cursor/Attrib/Read/Wrt Routns.
WAIT.ZIP          583  890630  Wait Until any Key is Pressed, Turbo Pascal.
WCTUNITS.ZIP    22875  910811  Collection of Turbo Pascal units with sources
WHATV2.ZIP       8893  890630  Lists to File Procedures/Functions in TPascal
WIH.ZIP         15068  890630  Stay Resident & Window Programs with Docs
WINDMNG1.ZIP    91126  890630  V1 Window Manager Program with Docs & Src Code.
WINDMNG2.ZIP    56587  890630  V2 Window Manager Program with Docs & Src Code.
WINDO21.ZIP     29543  890630  V2.1, Routines for Making a Window.  Fast.
WINDOW.ZIP       2635  890630  Window Routines, WINDOW.DOC & WINDOW.PAS.
WINDOWS.ZIP      1436  890630  Window Include File in Turbo Pascal.
WRDS2.ZIP        1925  890630  Tallies Occurances of Each Word in a Text File.
WRITDEMO.ZIP     1069  890630  Demo of Writelin & Writexy Fast Routines.
WS2ASCII.ZIP     5548  890630  Converts Wordstar Formatted to ASCII Text Files.
WSX.ZIP          3931  890630  Converts Wordstar Files to ASCII Text Files.
XAFMT2.ZIP       2389  890630  Compresses CompuServe's Directory Format.
XREF.ZIP         2627  880812  (turbo) Pascal Cross Reference (.pas/.com)
XREF3B.ZIP       9337  890630  Xref Using TP Reserved Words, from Grogono's Bk
XREFPAS.ZIP      1922  890630  Xref from Wirth's Program 4.8, Algo Data=Pgms
XREFPAS4.ZIP    22275  890630  Cross-Reference Program for Turbo Pascal Files.
XREFTP.ZIP      10773  890630  Xref Using Faster Indexing. Also See XREF3B.ZIP.
YASYNC.ZIP      14797  890612  "Yet Another" Asynch I/O Unit.  Interrupt-Driven
YESNO302.ZIP     5509  890630  Bypasses the Request: "Include Error Messages?".
YESNO3B.ZIP      2943  890630  Latest Patches for Message File "Yes/No" Questn.
YMODEM.ARC       3670  880523  YMODEM protocol Turbo Pascal source
TXTUTL
ASC2EBC.ARC      2069  900203  Converts ASCII to EBCDIC, w/ASM source
ASC2WORD.ARC    10560  900203  Converts ASCII to MS Word docs, w/C source
BIGSORT.ZIP      7334  890613  Sort Large Files by Key.  OK for Pipes.
BREAKUP.C        5997  860426  Break up large text files into several smaller
BYACC.ZIP      103026  910228  Berkeley YACC for MS-DOS, w/C source & docs
CASER.ZIP       35042  901026  Text case changing pgms with TC++ v1.0 source
CAWF2.ZIP      134442  911004  Formats docs intended for nroff -man, w/C src
CSPELLA.ARC    109278  871231  Spelling checker (see CSPELSRC for source)
CSPELSRC.ARC   100733  871231  C language source for CSPELLA.ARC
DDJGREP.ARC     25365  871119  Find strings in files, with source
DOS2UNIX.ARC    28108  891005  MSDOS<->Unix newline conversion pgm, w/C src
DTSRCH11.ZIP   285800  910908  dtSearch: Indexed/unindexed text search w/edit
FLIP1SRC.ARC    26708  890713  Convert text files MSDOS<->UNIX format, 2of2
GESTALT.ARC     28278  881207  New string similarity function. ASM,C,TurboPas
INDEX.ARC        3375  880722  Create text file indexes in C
LEX_YACC.ARC   182821  900301  Lex/Yacc compiler compiler, w/Turbo Pascal src
LST60ASM.ARC    13833  880724  ASM for early V Buerg LIST file viewer
LUCIFER.ARC      7827  880723  Encrypt/De-crypt bytes, LUCIFER alogrithm
MORE13.ZIP      45278  901223  Scroll forward/backward thru text file, w/src
NGCLON11.ZIP    33170  901216  Norton Guides clone, w/Turbo Pascal 5.0 src
NOCZ12.ARC      10243  891013  Strip Ctrl Z's from files, 3 versions, ASM src
NOTE11.ARC      19456  870208  Adds command line text to notebook file
NROFF1.ZIP      34371  901210  Unix V7 nroff clone with source for MS C 5.1
PCPATCH.ARC     70019  880209  Patch old source to current version
PDSRT212.ZIP    23749  900713  PD disk-based external sort program, w/C src
PRINT.ZIP        1812  880808  Print TXT files w/page #, headers, etc...
PROFF.ZIP       95834  880420  Another version of ROFF with source code
QUOTE24.ZIP     66627  910615  Fortune cookie quotation generator, w/C source
SCRIPTO.ZIP      5298  910702  Builds cursive writing from text, w/TPas src
SED15.ZIP       67052  911015  Unix-compatible streaming editor v1.5 TC src
SSPELL01.ZIP    98649  911114  TurboC++ or Unix C source for Unix Spell clone
TABX10.ARC      14125  900305  Filter that expands TABs to SPACEs, w/TC src
TOADCR11.ARC     9598  891107  Unix <> DOS text file EOL converter w/ASM src
TOADTRIM.ARC     4325  900119  Strip trailing spaces from text file,w/ASM src
TRAVESTY.ARC    21854  881218  Text/music style simulator w/C source
TSORT.ZIP        3429  880728  Text file sort in C
UNDIGEST.C       5155  901127  Break down newsletter digests to individ. msgs
UNTAB4.ZIP      11499  880808  Remove tabs from C source listing
WISDOM2.ZIP    111085  910528  A fortune cookie program with Turbo C source
```

UEMACS

```
SPELLSRC.ARC    30720  880215  MicroSPELL Spelling checker - source code
UE311C.ZIP     344108  911108  MicroEMACS 3.11 C source code
```

UUCP

```
SMAILSRC.ZIP   124737  900526  smail/PC - smart uucp mailer for MS-DOS, 2of2
UUPC11NS.ZIP   302280  911104  Mail w/UUCP source for BC++ 2.0 or MS C 6.0
```

VGA

```
VBALLS11.ZIP   159627  910805  VGA Palette animation (BGI) w/src,exe v1.1
VGA.C            6774  890415  TC:Make use of 256 color modes of Paradise VGA
```

WAFFLE

```
CWEXP104.ZIP    24147  911104  Free expire program for Waffle; includes C src
```

WINDOWS3

```
SPY30.ZIP       34092  911015  WIN3: Displays other windows' info (w src)
WINSRC20.ZIP   273408  901024  Source code for Fractint v2.0 for Windows3
WLOCK.ZIP       10448  910422  Combination Lock pgm for Windows3, w/src code
```

XLISP

```
XLSP21TC.ZIP   152175  910414  XLISP v2.1, TC++ 2.0 source & some net chatter
```

ZIP

```
A2Z16D.ARC      48704  891016  Convert ARC/PAK to ZIP 1.01 (TP 5 src) v1.6D
DEZIP20.ARC     47247  890908  ZIP extractor with Turbo Pascal 5.0 source
PKV100.ZIP      13993  901203  ZIP file viewer with TASM source
SHRINK12.ARC    40954  890414  Partial PKZIP clone with Pascal source
UNZIP41.ZIP     96731  910514  C source code for free Unix/VMS/MS-DOS UNZIP
ZIPV107.ARC     14749  890315  V.Buerg's ZIP dir lister ver. 1.07, w/ASM src
```

ZMODEM

```
RZSZ.ZIP        63123  880107  C source code for Zmodem module DSZ.EXE
RZSZ2.ZIP       64963  880724  Reformatted ZMODEM source for UNIX
```

ZOO

```
BOOZ20.ZIP      31098  910713  ZOO file extractor, handles new v2.10, w/C src
```

□

Schluß des Beitrags Harddiskmanagement

```
void KillFiles()
{ char FileType[3][6]={"*.EXE","*.OBJ","*.BAK"};
  char Pfad[MAXPATH];
  int Handle, Zaehler, RueckWert;
  struct ffblk FFStruktur;
  struct ftime DateiZeit;
  char Delete[20];
  for (Zaehler=0; Zaehler<=2; Zaehler++)
  { RueckWert=findfirst (FileType[Zaehler],&FFStruktur,0);
    while (! RueckWert)
    { Handle=open (FFStruktur.ff_name,O_RDWR);
      getftime(Handle,&DateiZeit);
      close (Handle);
      if ((Heute.da_year == DateiZeit.ft_year+1980)
         && (Heute.da_mon == DateiZeit.ft_month)
         && (Heute.da_day == DateiZeit.ft_day))
      { strcpy(Delete,"DEL ");
        strcat(Delete,FFStruktur.ff_name);
        system(Delete);
        getcwd(Pfad,MAXPATH);
        if (strlen(Pfad) > 3)
           strcat(Pfad,"\\");
        printf("\n%s gelöscht.",
               strcat(Pfad,FFStruktur.ff_name));
      }
      RueckWert=findnext (&FFStruktur);
    }
  }
  return;
}
```

```
void RootEinstellen(char *RootPfad)
{ char Pfad[MAXPATH]="";
  if (RootPfad[1] == ':')
  { setdisk(RootPfad[0]-'A');
    strcpy(Pfad,RootPfad);
  }
  else
  { Pfad[0]=getdisk()+'A';
    Pfad[1]=':';
    Pfad[2]=0;
    if (RootPfad[0] == '\\')
       strcat(Pfad,"\\");
    strcat(Pfad,RootPfad);
  }
  chdir (Pfad);
  strcat (Pfad,"\\");
  return;
}
```

□