

## Assembler-Seminaraufgabe WS93/94

Entwickeln Sie - zuzüglich eines Hauptprogramms als Testumgebung - ein allgemein verwendbares Unterprogramm zur Eingabe von Zeichenketten. Dabei gelten folgende Randbedingungen :

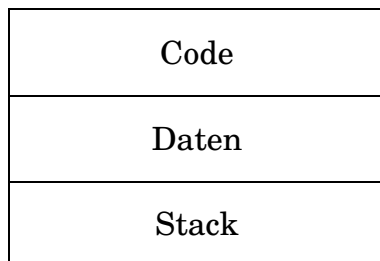
(1) Unterstützte Sondertasten der Eingaberoutine :

<Pfeil links> : Zeichen nach links  
<Pfeil rechts> : Zeichen nach rechts  
<Entf> : Zeichen an Cursorposition löschen  
<Backspace> : Zeichen links vom Cursor löschen  
<Einf> : Einfügemodus ein/aus

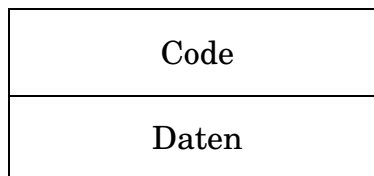
(2) Parameter des Unterprogramms :

Zeichenkette : INPUT / OUTPUT  
Länge der Zeichenkette : INPUT  
Linkes Ende der Zeichenkette (Zeile) : INPUT  
Linkes Ende der Zeichenkette (Spalte) : INPUT

(3) Minimale Segment- und Modulkonfiguration :



Hauptprogramm  
in  
Quellcodedatei 1



Unterprogramm  
in  
Quellcodedatei 2

(4) Entwicklungskonfiguration :

IBM PC (oder IBM-Kompatibler PC) und 80x86-Assembler (MASM oder TASM)

In dieser Seminaufgabe sind vordefinierte Module oder Makros nicht erwünscht; davon ausgenommen sind selbstverständlich alle während der Vorlesung besprochenen Beispielprogramme.

Spätester Abgabetermin für Programm und Dokumentation ist der 30.5.1994.

## Assembler-Seminaraufgabe SS94

Entwickeln Sie - zuzüglich eines Hauptprogramms als Testumgebung - eine Toolbox allgemein verwendbarer Unterprogramme zur Bearbeitung von Zeichenketten. Dabei gelten folgende Randbedingungen :

(1) Funktionalität der Toolbox :

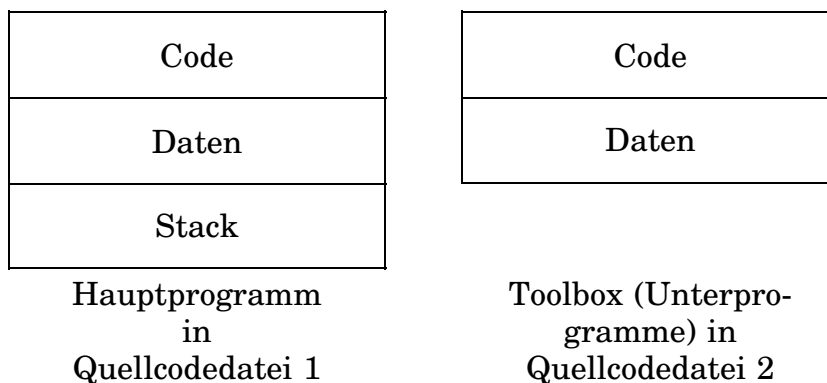
CONCAT : Zusammenfügen mehrerer Zeichenketten  
LENGTH : Länge einer Zeichenkette bestimmen  
COPY : Kopieren eines Teils einer Zeichenkette  
POS : Position einer Quellzeichenkette in einer Zielzeichenkette bestimmen  
INSERT : Einfügen einer Quellzeichenkette in einer Zielzeichenkette  
DELETE : Entfernen eines Teils einer Zeichenkette

Die Zeichenketten entsprechen entweder den TurboPascal- oder C-Konventionen. Alle Unterprogramme sind Prozeduren mit Datenübergabe via Referenzparameter.

(2) Funktionalität der Testumgebung :

Die Testumgebung gewährleistet die Erfüllung der Grundsätze des Maschinentests (vgl. Programmierung 1).

(3) Minimale Segment- und Modulkonfiguration :



Besser : Ablage der lokalen Variablen auf dem Stack statt in einem separaten Datensegment.

(4) Entwicklungskonfiguration :

IBM PC (oder IBM-Kompatibler PC) und 80x86-Assembler (MASM oder TASM)

In dieser Seminaufgabe sind vordefinierte Module oder Makros nicht erwünscht; davon ausgenommen sind selbstverständlich alle während der Bearbeitung dieser Seminaufgabe erstellten Module und Makros sowie alle während der Vorlesung besprochenen Beispielprogramme.

Spätester Abgabetermin für Programm und Dokumentation ist der 21.11.1994.

Ansprechpartner : Birger Wolter

## Assembler-Seminaraufgabe WS94/95

Entwickeln Sie unter Beachtung nachfolgender Randbedingungen die Dienstprogramme BAUM und MEHR :

(1) Funktionalität BAUM :

Graphische Ausgabe der Struktur eines angegebenen Verzeichnisses (z.B. D:\) über die Datei Standard-Ausgabe.

(2) Funktionalität MEHR :

Seitenweise Ausgabe der Datei Standard-Eingabe über die Datei Standard-Ausgabe.

(3) Beispiele (bezogen auf RZ3) :

```
C:\WORK>BAUM D:\ (vgl. Abbildung)
C:\WORK>BAUM D:\HANDOUT
C:\WORK>MEHR <D:\USERS\INFOTXT\INFO.TXT (vgl. Abbildung)
C:\WORK>BAUM D:\ | MEHR (vgl. Abbildung)
C:\WORK>BAUM D:\HANDOUT | MEHR
```

(4) Programmstruktur :

Die Möglichkeiten der Code- und Datenstrukturierung durch Module, Segmente und Prozeduren sowie Parameterübergaben an Unterprogramme und lokale Variablen in Unterprogrammen sollen ausgenutzt werden.

(5) Entwicklungskonfiguration :

80x86-Assembler (Microsoft MASM oder Borland TASM) auf beliebigem Rechnersystem.

(6) Zielkonfiguration :

Personalcomputer (PC) gemäß Intel-Architektur ( $\geq 8086$ ) unter Microsoft DOS ( $\geq$  Version 3.3)

In dieser Seminaraufgabe sind vordefinierte Module oder Makros nicht erwünscht; davon ausgenommen sind selbstverständlich alle während der Bearbeitung dieser Seminaraufgabe erstellten Module und Makros sowie alle während der Vorlesung besprochenen Beispielprogramme.

Spätester Abgabetermin für Programm und Dokumentation ist der 29.5.1995.

Ansprechpartner : Birger Wolter



## Assembler-Seminaraufgabe SS95 (IA)

Entwickeln Sie unter Beachtung nachfolgender Randbedingungen das Dienstprogramm SUCHE :

(1) Funktionalität SUCHE (Teil 1) :

Suche nach Dateinamen, die einem angegebenen Suchmuster (z.B. \*.BAT) entsprechen, in angegebenen Verzeichnissen (z.B. K:\ , N:\ und T:\) inklusive aller zugehörigen Unterverzeichnisse. Zur Ausgabe des Ergebnisses dient die Datei Standard-Ausgabe.

(2) Funktionalität SUCHE (Teil 2) :

Suche nach doppelten Dateinamen in angegebenen Verzeichnissen (z.B. K:\ , N:\ und T:\) inklusive aller zugehörigen Unterverzeichnisse. Zur Ausgabe des Ergebnisses dient die Datei Standard-Ausgabe.

(3) Beispiele (bezogen auf RZ3) :

C:\WORK>SUCHE \*.BAT K:\ N:\ T:\ (vgl. Abbildung)  
C:\WORK>SUCHE /DOPPELT K:\ N:\ T:\ (vgl. Abbildung)

(4) Programmstruktur :

Die Möglichkeiten der Code- und Datenstrukturierung durch Module, Segmente und Prozeduren sowie Parameterübergaben an Unterprogramme und lokale Variablen in Unterprogrammen sollen ausgenutzt werden. Für die Realisierung der "Doppeltsuche" ist ein vertretbarer (und begründeter) Kompromiß zwischen Programmieraufwand, Rechenzeit und Speicherplatzbedarf gesucht.

(5) Entwicklungskonfiguration :

80x86-Assembler (Microsoft MASM oder Borland TASM) auf beliebigem Rechnersystem.

(6) Zielkonfiguration :

Personalcomputer (PC) gemäß Intel-Architektur ( $\geq 8086$ ) unter Microsoft DOS ( $\geq$  Version 3.3)

In dieser Seminaufgabe sind vordefinierte Module oder Makros nicht erwünscht; davon ausgenommen sind selbstverständlich alle während der Bearbeitung dieser Seminaufgabe erstellten Module und Makros sowie alle während der Vorlesung besprochenen Beispielprogramme.

Spätester Abgabetermin für Programm und Dokumentation ist der 27.11.1995.

Ansprechpartner : Birger Wolter

K:\			
exita86.bat	38	29.08.94	13:34
exittasm.bat	39	17.05.94	12:45
exittp.bat	39	10.05.94	17:11
inita86.bat	258	29.08.94	13:34
inittasm.bat	256	17.05.94	12:45
inittp.bat	251	10.05.94	17:11
initmpz.bat	142	31.03.95	13:55
exitmpz.bat	103	31.03.95	13:55
K:\A86			
hlista86.bat	359	01.01.80	00:37
mexp.bat	34	01.01.80	00:37
mtcols.bat	26	01.01.80	00:37
erdemo.bat	38	01.01.80	00:37
flista86.bat	1011	01.01.80	00:37
K:\TP6\TVDEMOS			
genForms.bat	37	07.11.90	06:00
mkrdemo.bat	547	07.11.90	06:00
K:\PC65			
start.bat	24	07.04.92	12:24
pc65.bat	27	13.06.93	17:51
N:\B&P			
autprint.bat	27	21.02.92	11:45
N:\CORELDRV			
cdrom.bat	54	06.01.94	11:50
N:\RTTOOLS3\DEMO			
bigc.bat	1442	30.04.92	09:19
T:\4PRINT			
4print.bat	51	06.12.93	21:01
T:\CTSI			
ctpci.bat	25	06.12.93	12:00
T:\CTSI\GOODIES			
autoexec.bat	18	11.11.93	11:11
T:\DCC			
addcomnt.bat	1120	19.10.94	03:01
vircheck.bat	966	19.10.94	03:01
unpack.bat	789	19.10.94	03:01
beispiel.bat	364	19.10.94	03:01
delcomnt.bat	1126	19.10.94	03:01
T:\TBAV			
install.bat	11392	29.11.94	06:30
makeresc.bat	1855	29.11.94	06:30
T:\UTIL			
cdrom.bat	60	05.05.94	15:12
readme.bat	88	12.01.93	18:06
start.bat	320	12.01.93	18:06
keyboard.bat	11	12.10.93	16:01
T:\UTIL\SSTOR\SSTOR			
addevice.bat	383	03.04.89	12:00
install.bat	1718	03.04.89	12:00

N:\CDROM\			
CDMAP.EXE			
N:\CORELDRV\			
CDMAP.EXE			
T:\UTIL\			
CDMAP.EXE			
N:\CORELDRV\			
CDROM.BAT			
T:\UTIL\			
CDROM.BAT			
K:\TP6\BGI\			
CGA.BGI			
T:\CTSI\			
CGA.BGI			
T:\CTSI\			
COMTEST.EXE			
T:\UTIL\			
COMTEST.EXE			
N:\CORELDRV\			
CONFIG.SYS			
T:\CTSI\GOODIES\			
CONFIG.SYS			
N:\CDROM\CDINST\			
CORELCDX.COM			
N:\CORELDRV\			
CORELCDX.COM			
T:\UTIL\			
CORELCDX.COM			
T:\CTSI\			
CTCHIPZ.EXE			
T:\CTSI\CFG\			
CTCHIPZ.EXE			
N:\CDROM\CDINST\			
CUNI_NOV.SYS			
N:\CORELDRV\			
CUNI_NOV.SYS			
K:\TASM\			
DPMI16BI.OVL			
K:\TASM\TD\			
DPMI16BI.OVL			
T:\CTSI\			
DPMI16BI.OVL			
T:\DCC\			
DPMI16BI.OVL			
K:\TASM\			
DPMILOAD.EXE			
K:\TASM\TD\			
DPMILOAD.EXE			
K:\TASM\			
DPMIMEM.DLL			
K:\TASM\TD\			
DPMIMEM.DLL			
K:\TP6\BGI\			
EGAVGA.BGI			
K:\MPZ\			
EGAVGA.BGI			
N:\FLW\			
EGAVGA.BGI			
T:\CTSI\			
EGAVGA.BGI			
T:\F-PROT\			
FILE_ID.DIZ			
T:\SCAN\			
FILE_ID.DIZ			
T:\TBAV\			
FILE_ID.DIZ			
K:\TP6\UTILS\			
GREP.COM			
T:\UTIL\			
GREP.COM			
K:\TP6\BGI\			
HERC.BGI			
N:\FLW\			
HERC.BGI			
T:\CTSI\			
HERC.BGI			
T:\TBAV\			
INSTALL.BAT			
T:\UTIL\SSTOR\SSTOR\			
INSTALL.BAT			
N:\B&P\			
INSTALL.EXE			
N:\CDROM\CDINST\			
INSTALL.EXE			
K:\TP6\BGI\			
LITT.CHR			
N:\FLW\			
LITT.CHR			
K:\A86\			
PKUNZIP.EXE			
T:\PACKER\			
PKUNZIP.EXE			
T:\MISC\			
Q.EXE			
T:\UTIL\			
Q.EXE			

K:\TP6\			
README			
N:\RTK30\			
README			
T:\UTIL\			
README			
T:\UTIL\SSTOR\SSTOR\			
README			
N:\GPSSH\			
README.DOC			
T:\4PRINT\			
README.DOC			
T:\UTIL\			
README.DOC			
T:\DCC\			
REF.DOC			
T:\UTIL\			
REF.DOC			
N:\RTK30\KERNEL\TP60\STD28			
RTKERNEL.TPU			
N:\RTK30\KERNEL\TP60\STAND			
RTKERNEL.TPU			
N:\RTK30\KERNEL\TP60\DEBUG			
RTKERNEL.TPU			
N:\RTK30\KERNEL\TP55\STAND			
RTKERNEL.TPU			
N:\RTK30\KERNEL\TP55\DEBUG			
RTKERNEL.TPU			
T:\CTSI\			
RTM.EXE			
T:\DCC\			
RTM.EXE			
N:\B&P\			
SETTINGS.STM			
N:\OMP\			
SETTINGS.STM			
K:\PC65\			
START.BAT			
T:\UTIL\			
START.BAT			
N:\CDROM\			
SYS\$ERR.DTA			
T:\UTIL\			
SYS\$ERR.DTA			
N:\CDROM\			
SYS\$HELP.DTA			
T:\UTIL\			
SYS\$HELP.DTA			
N:\CDROM\			
SYS\$MSG.DTA			
T:\UTIL\			
SYS\$MSG.DTA			
K:\TASM\			
TD.EXE			
K:\TASM\TD\			
TD.EXE			
K:\TASM\			
TDHELP.TDH			
K:\TASM\TD\			
TDHELP.TDH			
K:\TP6\			
UNZIP.EXE			
T:\PACKER\			
UNZIP.EXE			
T:\DCC\			
WHAT\$NEW.TXT			
T:\UTIL\			
WHAT\$NEW.TXT			

## Assembler-Seminaraufgabe WS95/96 (IA)

Entwickeln Sie unter Beachtung nachfolgender Randbedingungen den Mini-Befehlsinterpreter ANALYSE :

(1) (Mindest-)Funktionalität :

*Alle Angaben laut MS-DOS Merkblatt I/II (Programmierung 1/2)*

Dateinamen :

```
Dateiname.Extension
├── nähere Beschreibung zu Dateiname (z.B. COM, EXE :
│   │   Maschinenprogramm;  SYS  :  Gerätesteuerungspro-
│   │   gramme;  PAS  :  Pascal-Programm;  BAT  :  Stapelver-
│   │   arbeitungsdatei (Kommandodatei))
└── Dateiname selbst (zugelassene Zeichen :  A-Z,  a-z,  0-9,  !,
    #,  $,  %,  &,  ',  (,  ),  -,  @,  ^,  _,  `,  {,  },  ~)
```

wild cards :

```
? beliebiges (d.h. kein oder ein) Zeichen an angegebener
  Position in Dateiname (ohne Extension) oder Extension
* beliebige (d.h. kein, ein oder mehrere) Zeichen ab
  angegebener Position in Dateiname (ohne Extension) oder
  Extension
```

Suchwege :

```
Verzeichnisname [\Verzeichnisname] ... \ (relativ)
\Verzeichnisname [\Verzeichnisname] ... \ (absolut)
├-< Suchweg, falls kein Dateiname folgt ->|
├-< Suchweg, falls Dateiname folgt ->|
└── Stammverzeichnis
```

spezielle Verzeichnisnamen :

```
.  aktuelles Verzeichnis
.. direkt übergeordnetes Verzeichnis
```

Befehl ausführen : Befehlsname <RETURN>

Standardlaufwerk wechseln : Laufwerk: <RETURN>

CHDIR aktuelles Verzeichnis wechseln/anzeigen

```
CHDIR [Laufwerk:] [Suchweg] (alternativ : CD ...)
```

Hinweis : Für jedes Laufwerk ist ein aktuelles Verzeichnis eingestellt.

COPY Dateien kopieren

```
COPY [Laufwerk:] [Suchweg1] Dateiname1 [Laufwerk:] [Suchweg2]
      [Dateiname2]
```

DEL Dateien löschen

```
DEL [Laufwerk:] [Suchweg] Dateiname
```

**DIR** Inhalt des aktuellen oder angegebenen Verzeichnisses ausgeben

**DIR** [Laufwerk:] [Suchweg] [Dateiname] [/P]

/P seitenweise Ausgabe

**EXIT** Befehlsprozessor verlassen

**MKDIR** Verzeichnis anlegen

**MKDIR** [Laufwerk:] Suchweg (alternativ : MD)

**PROMPT** Eingabeaufforderung ändern

**PROMPT** [ [Text] [\$Zeichen] ...]

\$P Standardlaufwerk und aktuelles Verzeichnis

\$G Zeichen >

**REN** Dateien umbenennen

**REN** [Laufwerk:] [Suchweg] Dateiname1 Dateiname2

**RMDIR** Verzeichnis löschen

**RMDIR** [Laufwerk:] Suchweg (alternativ : RD ...)

**TYPE** Dateiinhalt anzeigen

**TYPE** [Laufwerk:] [Suchweg] Dateiname

(2) Informationsquellen :

- Assembler-Vorlesung
- Beispielprogramme auf RZ12/DOS:DATA\HANDOUT\ASS.IA
  - MSDOS.ASM
  - EXEC.ASM *NEU*
- Interrupt-Liste auf RZ12/DOS:DATA\HANDOUT\ASS.IA\INTX *NEU*

(3) Programmstruktur :

Die Möglichkeiten der Code- und Datenstrukturierung durch Module, Segmente und Prozeduren sowie Parameterübergaben an Unterprogramme und lokale Variablen in Unterprogrammen sollen ausgenutzt werden.

(4) Entwicklungskonfiguration :

80x86-Assembler (Microsoft MASM oder Borland TASM) auf beliebigem Rechnersystem.

(5) Zielkonfiguration :

Personalcomputer (PC) gemäß Intel-Architektur (≥ 8086) unter Microsoft DOS (≥ Version 3.3)



In dieser Seminaufgabe sind vordefinierte Module oder Makros nicht erwünscht; davon ausgenommen sind selbstverständlich alle während der Bearbeitung dieser Seminaufgabe erstellten Module und Makros sowie alle während der Vorlesung besprochenen Beispielprogramme.

Der Mini-Befehlsinterpreter ANALYSE muß auf den PC's im RZ1 lauffähig sein.

Spätester Abgabetermin für Programm und Dokumentation ist der 3.6.1996. Für verspätete Abgaben gilt folgende Regelung : Pro Tag Verspätung (incl. Sonnabende und Sonntage) werden pauschal 5% der erreichbaren Punktzahl abgezogen (z.B. erreichbare Punktzahl = 100, erreichte Punktzahl = 80, Abgabe am 5.6.1996 (d.h. zwei Tage zu spät) ⇒ bewertete Punktzahl = 70).

Ansprechpartner : Birger Wolter

## Assembler-Seminaraufgabe SS96 (IA)

Entwickeln Sie unter Beachtung nachfolgender Randbedingungen die Dienstprogramme (Filter) FINDE und SORTIERE :

(1) Funktionalität FINDE :

Zeichenfolge suchen

FINDE [/V] [/N] [/I] "Zeichenfolge" [Laufwerk:] [Suchweg] [Dateiname]

/V : Anzeige aller Zeilen, in denen Zeichenfolge nicht enthalten

/N : Anzeige incl. relativer Zeilennummern

/I : keine Berücksichtigung von Groß- und Kleinschreibung

Bedenke : Der *Dateiname* darf auch *wild cards* enthalten

(2) Funktionalität SORTIERE :

Sortierte Ausgabe der Standardeingabe

SORTIERE [/R] [/+N]

/R : umgekehrte Sortierreihenfolge

/+N : Sortierung nach den Zeichen ab Spalte N

(3) Beispiele (bezogen auf RZ1) :

```
C:\WORK>FINDE /N /I "ZUSTAND" M:\HANDOUT\PROG.IA\STRZAHL1.PAS
```

```
C:\WORK>FINDE /N /I "BEGIN" M:\HANDOUT\PROG.IA\*.PAS
```

```
C:\WORK>DIR M:\HANDOUT\PROG.IA | FINDE /N /I "PAS"
```

```
C:\WORK>DIR M:\HANDOUT\PROG.IA | FINDE /V /N /I "PAS"
```

```
C:\WORK>DIR M:\HANDOUT\PROG.IA | SORTIERE
```

```
C:\WORK>DIR M:\HANDOUT\PROG.IA | SORTIERE /R
```

```
C:\WORK>DIR M:\HANDOUT\PROG.IA | SORTIERE /+14
```

(4) Informationsquellen :

- Assembler-Vorlesung
- Beispielprogramme auf RZ12/DOS:DATA\HANDOUT\ASS.IA
- Interrupt-Liste auf RZ12/DOS:DATA\HANDOUT\ASS.IA\INTX

(5) Programmstruktur :

Die Möglichkeiten der Code- und Datenstrukturierung durch Module, Segmente und Prozeduren sowie Parameterübergaben an Unterprogramme und lokale Variablen in Unterprogrammen sollen ausgenutzt werden. Für die Realisierung von SORTIERE ist ein vertretbarer (und begründeter) Kompromiß zwischen Programmieraufwand, Rechenzeit und Speicherplatzbedarf gesucht.

(6) Entwicklungskonfiguration :

80x86-Assembler (Microsoft MASM oder Borland TASM) auf beliebigem Rechnersystem.

(7) Zielkonfiguration :

Personalcomputer (PC) gemäß Intel-Architektur ( $\geq 8086$ ) unter Microsoft DOS ( $\geq$  Version 3.3)

In dieser Seminaufgabe sind vordefinierte Module oder Makros nicht erwünscht; davon ausgenommen sind selbstverständlich alle während der Bearbeitung dieser Seminaufgabe erstellten Module und Makros sowie alle während der Vorlesung besprochenen Beispielprogramme.

Die Dienstprogramme (Filter) FINDE und SORTIERE müssen auf den PC's im RZ1 lauffähig sein.

Spätester Abgabetermin für Programm und Dokumentation ist der 25.11.1996. Für verspätete Abgaben gilt folgende Regelung : Pro Tag Verspätung (incl. Sonntage und Sonntage) werden pauschal 5% der erreichbaren Punktzahl abgezogen (z.B. erreichbare Punktzahl = 100, erreichte Punktzahl = 80, Abgabe am 27.11.1996 (d.h. zwei Tage zu spät)  $\Rightarrow$  bewertete Punktzahl = 70).

Ansprechpartner : Birger Wolter

```

----- M:\HANDOUT\PROG.IA\STRZAH1.PAS
[9]  zustand : zustaende;
[35] zustand := z0;
[36] while (index <= ord(zekette[0])) and (zustand <> zf) do begin
[37]   case zustand of
[40]     zustand := z1;
[43]     zustand := zf;
[47]   if zustand = z1

```

FINDE /N /I "ZUSTAND" M:\HANDOUT\PROG.IA\STRZAH1.PAS

```

----- M:\HANDOUT\PROG.IA\BSP01.PAS
[31]begin
[37]  begin
[51]begin
[56]  begin

----- M:\HANDOUT\PROG.IA\BSP02.PAS
[25]begin
[33]  begin
[40]  begin
[50]  begin

----- M:\HANDOUT\PROG.IA\BSP03.PAS
[22]begin

----- M:\HANDOUT\PROG.IA\BSP04.PAS
[26]begin
[35] then begin

----- M:\HANDOUT\PROG.IA\BSP05.PAS
[30]begin
[39] then begin
[60]  begin

----- M:\HANDOUT\PROG.IA\BSP06.PAS
[41]begin
[51]begin
[61]begin
[70] then begin
[93]  begin

----- M:\HANDOUT\PROG.IA\BSP07.PAS
[75]begin
[85]begin
[95]begin
[106]begin
[113] then begin
[129]  begin
[134]  begin
[152]  begin

----- M:\HANDOUT\PROG.IA\BSP08.PAS
[39]begin
[50] then begin

```

FINDE /N /I "BEGIN" M:\HANDOUT\PROG.IA\\*.PAS

```

----- M:\HANDOUT\PROG.IA\STRZAH1.PAS
[13]begin
[14] while (ord(zekette[0]) >= 1) and (zekette[1] = ' ') do begin
[23]begin
[29]begin
[36] while (index <= ord(zekette[0])) and (zustand <> zf) do begin
[39]   then begin

----- M:\HANDOUT\PROG.IA\STRZAH2.PAS
[14]begin
[15] while (ord(zekette[0]) >= 1) and (zekette[1] = ' ') do begin
[24]begin
[30]begin
[38] while (index <= ord(zekette[0])) and (zustand <> zf) do begin
[41]   then begin
[46]   then begin
[52]   then begin
[61] then begin

----- M:\HANDOUT\PROG.IA\STRZAH3.PAS
[15]begin
[16] while (ord(zekette[0]) >= 1) and (zekette[1] = ' ') do begin
[25]begin
[31]begin
[39] while (index <= ord(zekette[0])) and (zustand <> zf) do begin
[42]   then begin
[50]   then begin
[56]   then begin
[66] then begin

----- M:\HANDOUT\PROG.IA\DEBUG.PAS
[9]begin
[19]begin
[23]begin
[31]begin
[35]begin

----- M:\HANDOUT\PROG.IA\RECORDS.PAS
[20]begin
[25] With Heute Do Begin
[41] With Schueler Do Begin
[45] With Eintritt Do Begin
[50] With Austritt Do Begin

----- M:\HANDOUT\PROG.IA\GGT_U.PAS
[8]begin
[13] while (a <= 0) or (b <= 0) do begin
[14]   if a < 0 then begin
[18]   if b < 0 then begin

----- M:\HANDOUT\PROG.IA\GERADE_U.PAS
[9]begin
[17] else begin

----- M:\HANDOUT\PROG.IA\WURZEL_U.PAS
[8]begin
[13] while a <= 0 do begin

----- M:\HANDOUT\PROG.IA\GGWUNIT.PAS
[19]begin
[31]begin
[44]begin

```

FINDE /N /I "BEGIN" M:\HANDOUT\PROG.IA\\*.PAS (Fortsetzung)

```

----- M:\HANDOUT\PROG.IA\DATIOBJ1.PAS
[14]Begin
[15] With Instance Do Begin
[18] 1,3,5,7,8,10,12 : If Tag > 31 Then Begin
[22] 4,6,9,11 : If Tag > 30 Then Begin
[27] ((Tag > 29) And (Jahr Mod 4 = 0)) Then Begin
[32] If Monat > 12 Then Begin
[39]Begin

----- M:\HANDOUT\PROG.IA\DATIOBJ2.PAS
[15]Begin
[18] 1,3,5,7,8,10,12 : If Tag > 31 Then Begin
[22] 4,6,9,11 : If Tag > 30 Then Begin
[27] ((Tag > 29) And (Jahr Mod 4 = 0)) Then Begin
[32] If Monat > 12 Then Begin
[38]Begin

----- M:\HANDOUT\PROG.IA\DATIOBJ3.PAS
[17]Begin
[25]Begin
[28] 1,3,5,7,8,10,12 : If Tag > 31 Then Begin
[32] 4,6,9,11 : If Tag > 30 Then Begin
[37] ((Tag > 29) And (Jahr Mod 4 = 0)) Then Begin
[42] If Monat > 12 Then Begin
[54]Begin
[59]Begin
[63]Begin

----- M:\HANDOUT\PROG.IA\DATIREC1.PAS
[14]Begin
[17] 1,3,5,7,8,10,12 : If Instance.Tag > 31 Then Begin
[21] 4,6,9,11 : If Instance.Tag > 30 Then Begin
[27] Then Begin
[32] If Instance.Monat > 12 Then Begin
[38]Begin

----- M:\HANDOUT\PROG.IA\DATIREC2.PAS
[14]Begin
[15] With Instance Do Begin
[18] 1,3,5,7,8,10,12 : If Tag > 31 Then Begin
[22] 4,6,9,11 : If Tag > 30 Then Begin
[27] ((Tag > 29) And (Jahr Mod 4 = 0)) Then Begin
[32] If Monat > 12 Then Begin
[39]Begin

----- M:\HANDOUT\PROG.IA\GRAF.PAS
[21]Begin

----- M:\HANDOUT\PROG.IA\GRAFUNIT.PAS
[55]Begin
[62]Begin
[68]Begin
[74]Begin
[81]Begin
[88]Begin
[95]Begin

```

FINDE /N /I "BEGIN" M:\HANDOUT\PROG.IA\\*.PAS (Fortsetzung)

```

[7]PASCASSI DOC 1580220 29.06.93 22:11
[8]BSP01 PAS 1230 29.05.93 21:09
[9]BSP02 PAS 1045 29.05.93 21:09
[10]BSP03 PAS 1272 29.05.93 21:09
[11]BSP04 PAS 1313 03.06.93 15:32
[12]BSP05 PAS 1756 07.06.93 16:22
[13]BSP06 PAS 2182 07.06.93 16:23
[14]BSP07 PAS 3593 07.06.93 16:24
[15]BSP08 PAS 1903 29.06.93 21:23
[16]STRZAHL1 PAS 1421 08.06.94 17:04
[17]STRZAHL2 PAS 1949 08.06.94 17:12
[18]STRZAHL3 PAS 2233 08.06.94 17:13
[19]DEBUG PAS 672 23.04.96 12:56
[20]RECORDS PAS 1212 03.06.96 18:38
[21]GGT_U PAS 611 03.06.96 17:48
[22]GERADE_U PAS 547 03.06.96 17:49
[23]WURZEL_U PAS 419 03.06.96 18:00
[24]GWUNIT PAS 1201 03.06.96 17:58
[25]DATIOBJ1 PAS 1357 03.06.96 17:43
[26]DATIOBJ2 PAS 1293 03.06.96 17:44
[27]DATIOBJ3 PAS 1740 03.06.96 17:44
[28]DATIREC1 PAS 1491 03.06.96 17:41
[29]DATIREC2 PAS 1351 03.06.96 18:31
[30]GRAF PAS 1526 03.06.96 18:21
[31]GRAFUNIT PAS 2072 03.06.96 18:22

```

DIR M:\HANDOUT\PROG.IA | FINDE /N /I "PAS"

```

[1]
[2] Datenträger in Laufwerk M ist DOS
[3] Verzeichnis von M:\HANDOUT\PROG.IA
[4]
[5] . <DIR>
[6] .. <DIR>
[32] 27 Datei(en) 1615609 Byte
[33] 221970432 Byte frei

```

DIR M:\HANDOUT\PROG.IA | FINDE /V /N /I "PAS"

```

                221970432 Byte frei
    27 Datei(en) 1615609 Byte
Datenträger in Laufwerk M ist DOS
Verzeichnis von M:\HANDOUT\PROG.IA
.
<DIR>
..
<DIR>
BSP01 PAS      1230 29.05.93   21:09
BSP02 PAS      1045 29.05.93   21:09
BSP03 PAS      1272 29.05.93   21:09
BSP04 PAS      1313 03.06.93   15:32
BSP05 PAS      1756 07.06.93   16:22
BSP06 PAS      2182 07.06.93   16:23
BSP07 PAS      3593 07.06.93   16:24
BSP08 PAS      1903 29.06.93   21:23
DATIOBJ1 PAS   1357 03.06.96   17:43
DATIOBJ2 PAS   1293 03.06.96   17:44
DATIOBJ3 PAS   1740 03.06.96   17:44
DATIREC1 PAS   1491 03.06.96   17:41
DATIREC2 PAS   1351 03.06.96   18:31
DEBUG PAS      672 23.04.96   12:56
GERADE_U PAS   547 03.06.96   17:49
GGT_U PAS      611 03.06.96   17:48
GGWUNIT PAS   1201 03.06.96   17:58
GRAF PAS      1526 03.06.96   18:21
GRAFUNIT PAS   2072 03.06.96   18:22
PASCASSI DOC  1580220 29.06.93   22:11
RECORDS PAS   1212 03.06.96   18:38
STRZAH1 PAS   1421 08.06.94   17:04
STRZAH2 PAS   1949 08.06.94   17:12
STRZAH3 PAS   2233 08.06.94   17:13
WURZEL_U PAS   419 03.06.96   18:00

```

DIR M:\HANDOUT\PROG.IA | SORTIERE

```

WURZEL_U PAS   419 03.06.96   18:00
STRZAH3 PAS   2233 08.06.94   17:13
STRZAH2 PAS   1949 08.06.94   17:12
STRZAH1 PAS   1421 08.06.94   17:04
RECORDS PAS   1212 03.06.96   18:38
PASCASSI DOC  1580220 29.06.93   22:11
GRAFUNIT PAS   2072 03.06.96   18:22
GRAF PAS      1526 03.06.96   18:21
GGWUNIT PAS   1201 03.06.96   17:58
GGT_U PAS      611 03.06.96   17:48
GERADE_U PAS   547 03.06.96   17:49
DEBUG PAS      672 23.04.96   12:56
DATIREC2 PAS   1351 03.06.96   18:31
DATIREC1 PAS   1491 03.06.96   17:41
DATIOBJ3 PAS   1740 03.06.96   17:44
DATIOBJ2 PAS   1293 03.06.96   17:44
DATIOBJ1 PAS   1357 03.06.96   17:43
BSP08 PAS      1903 29.06.93   21:23
BSP07 PAS      3593 07.06.93   16:24
BSP06 PAS      2182 07.06.93   16:23
BSP05 PAS      1756 07.06.93   16:22
BSP04 PAS      1313 03.06.93   15:32
BSP03 PAS      1272 29.05.93   21:09
BSP02 PAS      1045 29.05.93   21:09
BSP01 PAS      1230 29.05.93   21:09
..
<DIR>
.
<DIR>
Verzeichnis von M:\HANDOUT\PROG.IA
Datenträger in Laufwerk M ist DOS
    27 Datei(en) 1615609 Byte
                221970432 Byte frei

```

DIR M:\HANDOUT\PROG.IA | SORTIERE /R

```

                221970432 Byte frei
WURZEL_U PAS   419 03.06.96   18:00
GERADE_U PAS   547 03.06.96   17:49
GGT_U PAS      611 03.06.96   17:48
DEBUG PAS      672 23.04.96   12:56
BSP02 PAS      1045 29.05.93   21:09
GGWUNIT PAS   1201 03.06.96   17:58
RECORDS PAS   1212 03.06.96   18:38
BSP01 PAS      1230 29.05.93   21:09
BSP03 PAS      1272 29.05.93   21:09
DATIOBJ2 PAS   1293 03.06.96   17:44
BSP04 PAS      1313 03.06.93   15:32
DATIREC2 PAS   1351 03.06.96   18:31
DATIOBJ1 PAS   1357 03.06.96   17:43
STRZAH1 PAS   1421 08.06.94   17:04
DATIREC1 PAS   1491 03.06.96   17:41
GRAF PAS      1526 03.06.96   18:21
DATIOBJ3 PAS   1740 03.06.96   17:44
BSP05 PAS      1756 07.06.93   16:22
BSP08 PAS      1903 29.06.93   21:23
STRZAH2 PAS   1949 08.06.94   17:12
GRAFUNIT PAS   2072 03.06.96   18:22
BSP06 PAS      2182 07.06.93   16:23
STRZAH3 PAS   2233 08.06.94   17:13
BSP07 PAS      3593 07.06.93   16:24
PASCASSI DOC  1580220 29.06.93   22:11
.
<DIR>
..
<DIR>
    27 Datei(en) 1615609 Byte
Datenträger in Laufwerk M ist DOS
Verzeichnis von M:\HANDOUT\PROG.IA

```

DIR M:\HANDOUT\PROG.IA | SORTIERE /+14

## Assembler-Seminaraufgabe WS96/97 (IA)

Entwickeln Sie - zuzüglich der Hauptprogramme PROBEASS und PROBEPAS als Testumgebung - eine Toolbox allgemein verwendbarer Unterprogramme. Dabei gelten folgende Randbedingungen :

(1) Toolbox :

- Vollständige Reimplementierung der TurboPascal-Prozeduren FINDFIRST und FINDNEXT
- Implementierung von Routinen zur Dateimanipulation im Sinne der TurboPascal-Prozeduren ASSIGN, RESET / REWRITE, BLOCKREAD / BLOCKWRITE, CLOSE für untypisierte Dateien
- Programmiersprache : 80x86-Assembler

(2) Hauptprogramme PROBEASS und PROBEPAS :

- Mindestfunktionalität (vgl. MS-DOS Merkblatt I/II (Programmierung 1/2)) :
  - Inhalt des aktuellen oder angegebenen Verzeichnis ausgeben (pro Eintrag: Name, Größe (nur für Dateien), Datum, Uhrzeit, Attribute)

```
PROBExxx /D [Laufwerk:][Suchweg][Dateiname]
Hinweis : Dateiname inklusive wildcards
```

- Datei kopieren

```
PROBExxx /C [Laufwerk:][Suchweg1]Dateiname1 [Laufwerk:][Suchweg2][Dateiname2]
Hinweis : Dateiname1 und Dateiname2 exklusive wildcards
```

- Programmiersprachen : 80x86-Assembler (PROBEASS) und TurboPascal (PROBEPAS)
- Die Mindestfunktionalität gilt sowohl für PROBEASS als auch PROBEPAS

(3) Beispiele (bezogen auf RZ1) :

```
C:\WORK>PROBExxx /D C:\
AUTOEXEC BAT      376 28.11.95 15:03 [A ]
CONFIG  BSP      1816 16.10.95 10:57 [A ]
CONFIG  TEL      1816 16.10.95 14:06 [A ]
MOUSE   INI      1269 15.05.93 22:21 [A ]
COMMAND COM     50031 11.06.91 12:00 [A ]
WORK    <DIR>    18.12.96  7:38 [  ]
```

```
C:\WORK>PROBExxx /D M:\HANDOUT\SOFTWARE\ASS\PTL\P*.ASM
P      ASM      475 13.12.94 11:35 [A ]
P1     ASM      419 11.01.95 17:28 [A ]
P2     ASM      228 24.11.93 11:21 [A ]
P3     ASM      228 24.11.93 11:22 [A ]
PS     ASM      619 13.12.94 12:21 [A ]
PS1    ASM      517 20.12.94 15:09 [A ]
PS2    ASM      217 24.11.93 11:32 [A ]
```

```

C:\WORK>COPY M:\HANDOUT\SOFTWARE\ASS\PTL\P*.ASM
M:\HANDOUT\SOFTWARE\ASS\PTL\P.ASM
M:\HANDOUT\SOFTWARE\ASS\PTL\P1.ASM
M:\HANDOUT\SOFTWARE\ASS\PTL\P2.ASM
M:\HANDOUT\SOFTWARE\ASS\PTL\P3.ASM
M:\HANDOUT\SOFTWARE\ASS\PTL\PS.ASM
M:\HANDOUT\SOFTWARE\ASS\PTL\PS1.ASM
M:\HANDOUT\SOFTWARE\ASS\PTL\PS2.ASM
7 Datei(en) kopiert

```

```

C:\WORK>PROBExxx /D *.ASM
P      ASM      475 13.12.94 11:35 [A ]
P1     ASM      419 11.01.95 17:28 [A ]
P2     ASM      228 24.11.93 11:21 [A ]
P3     ASM      228 24.11.93 11:22 [A ]
PS     ASM      619 13.12.94 12:21 [A ]
PS1    ASM      517 20.12.94 15:09 [A ]
PS2    ASM      217 24.11.93 11:32 [A ]

```

```

C:\WORK>PROBExxx /C M:\HANDOUT\SOFTWARE\ASS\PTL\PASASM.PAS

```

```

C:\WORK>PROBExxx /C PASASM.PAS PASASM1.PAS

```

```

C:\WORK>PROBExxx /C PASASM.PAS PASASM2.PAS

```

```

C:\WORK>PROBExxx /D *.PAS
PASASM PAS      787 10.01.97 13:58 [A ]
PASASM1 PAS     787 10.01.97 13:59 [A ]
PASASM2 PAS     787 10.01.97 13:59 [A ]

```

```

C:\WORK>PROBExxx /D
P      ASM      475 13.12.94 11:35 [A ]
P1     ASM      419 11.01.95 17:28 [A ]
P2     ASM      228 24.11.93 11:21 [A ]
P3     ASM      228 24.11.93 11:22 [A ]
PS     ASM      619 13.12.94 12:21 [A ]
PS1    ASM      517 20.12.94 15:09 [A ]
PS2    ASM      217 24.11.93 11:32 [A ]
PASASM PAS      787 10.01.97 13:58 [A ]
PASASM1 PAS     787 10.01.97 13:59 [A ]
PASASM2 PAS     787 10.01.97 13:59 [A ]

```

#### (4) Informationsquellen :

- Assembler-Vorlesung
- Beispielprogramme auf RZ12/DATA:HANDOUT\SOFTWARE\ASS\PTL insbesondere : MSDOS.ASM, PASASM.PAS, SWAPSM.ASM
- Interrupt-Liste auf RZ12/DATA:HANDOUT\SOFTWARE\ASS\PTL\INTX insbesondere : INT 21h für AH=3Ch, 3Dh, 3Eh, 3Fh, 40h, 4Eh, 4Fh

#### (5) Entwicklungskonfiguration :

80x86-Assembler (Microsoft MASM oder Borland TASM) und TurboPascal (Version 6.x/7.x) auf beliebigem Rechnersystem

#### (6) Zielkonfiguration :

Personalcomputer (PC) gemäß Intel-Architektur (≥ 8086) unter Microsoft DOS (≥ Version 5.0)



In dieser Seminaufgabe sind vordefinierte Module oder Makros nicht erwünscht; davon ausgenommen sind selbstverständlich alle während der Bearbeitung dieser Seminaufgabe erstellten Module und Makros sowie alle während der Vorlesung besprochenen Beispielprogramme.

Die Testumgebung (Hauptprogramme PROBEASS und PROBEPAS) muß auf den PC's im RZ1 lauffähig sein.

Spätester Abgabetermin für Programm und Dokumentation ist der 9.6.1997. Für verspätete Abgaben gilt folgende Regelung : Pro Tag Verspätung (incl. Sonnabende und Sonntage) werden pauschal 5% der erreichbaren Punktzahl abgezogen (z.B. erreichbare Punktzahl = 100, erreichte Punktzahl = 80, Abgabe am 11.6.1997 (d.h. zwei Tage zu spät) ⇒ bewertete Punktzahl = 70).

Ansprechpartner : Birger Wolter (wol@fh-wedel.de)