Liebe Seminaristen und Seminaristinnen,

der 5.Newsletter beschaeftigt sich mit den Antworten auf zwei
individuelle (?) Fragen :

--- Anfang 1. Frage / Antwort

Hallo XYZ!

> Hallo Birger!
>
> Habe gerade deine Nachrichten gekriegt. Dann war ich wohl nicht der
> einzige

Die letzte Anfrage bezueglich INT21 fuer AH=47h liegt bereits ueber ein
Jahr zurueck. Deine E-Mail war daher der Ausloeser fuer den "Newsletter"
Nummer 2.

> der darueber gestolpert ist. Merkwuerdig nur, das auch in sogenannten

Da die Autoren haeufig mehr oder weniger voneinander abschreiben
wundert es mich nicht so sehr. Ausserdem passiert bei einem
umfangreichen Referenzteil leicht ein "Registerdreher".

> Fachbuechern das falsche steht. Eine Frage habe ich noch zwecks
> Dokumentation, reicht es die Quelltexte sehr ausfuehrlich
> auszudokumentieren, oder soll es ein richtiges Heft dazu sein? Meine
> Quelltexte sind im Moment so gestaltet, das hinter fast jeder Zeile
> steht was sie bedeutet, bei den Prozeduren habe ich zusaetzlich
> vermerkt, was sie machen, welche Eingaben, Ausgaben und AEnderungen
> der Register sie machen.
>

Alles was bereits Inline dokumentiert wurde, muss nicht noch einmal
Exline dokumentiert werden. Der Dokumentationsteil Problemanalyse und
-realisation (insbesondere Herleitung, Begruendung und Erlaeuterung der
Datenstrukturen und Kernalgorithmen) sollte jedoch grundsaetzlich
Exline fomuliert sein.

>
> Gruss XYZ!
>

Gruss Birger!

--- Ende 1. Frage / Antwort

--- Anfang 2. Frage

Hallo Birger,

mir ist beim nachprogrammieren des "Pascal-Findfirst" folgendes
aufgefallen, was mich ein "bisschen" verwirrt:

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
FindFirst Deklaration laut TP7-Hilfe:
    procedure FindFirst(Path: PChar; Attr: Word; var F: SearchRec);
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Laut TP.7-Hilfe ist der erste Parameter von FindFirst vom Typ "PChar",
was ja ein Zeiger auf einen nullterminierten String ist.

Allerdings kommt es zu einer Fehlermeldung, wenn man diesen auch Typ
tatsaechlich als "Path" benutzt. Vielmehr MUSS ein "Path" vom Typ
"String" verwendet werden, obwohl laut Deklaration (TP-Hilfe) "Path"
vom Typ "PChar" ist.

Das Problem, was dadurch aufkommt, ist folgendes:
Deklariere ich mein FindFirst (externel) im Pascal mit "Path" vom Typ
"PChar"

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   procedure FindFirst( Path:PChar ; Attr : Word ; var F : SearchRec)
   ;
external
   {$L c:\work\XYZ\toolbox}

   var path : PChar ;

   begin {main}
      ...
      ...
      findfirst(path,........)
      ...
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

(genauso, wie in der TP-Hilfe), dann MUSS auch der "Path" beim Aufruf vom Typ "PChar" sein. Bei z.B. "var Path : String" kommt es, anders als beim "orginal Findfirst" (!!), das auch einen "Path" vom Typ String verarbeitet, zur Compiler-Fehlermeldung!

Wird also "PChar" verwand, so legt der Compiler auch "brav" einen nullterminierten String im DS an.

Deklariere ich mein FindFirst wie folgt mit "Path : String",

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   procedure FindFirst( Path:String ; Attr : Word ; var F : SearchRec)
   ;
external
   {$L c:\work\XYZ\toolbox}

   var path : String ;

   begin {main}
      ...
      ...
      findfirst(path,........)
      ...
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

dann legt der Compiler einen PASCAL-String im DS an (mit String[0]=Stringlaenge), der natuerlich anders verarbeitet werden muss als ein Nullterminierter.

Soweit alles klar....? :-)

Meine Frage ist nun, welche ich Deklaration ich nehmen soll ?

Allerdings....
1) Nehme ich die mit "Path : PChar", dann muss, wie gesagt, auch beim Aufruf ein "Path" vom Typ "PChar" genommern werden. Das wuerde also auch nicht dem Orginal entsprechen, was ja den Typ "String" beim Aufruf verwendet. Zum Anderen arbeitet "PChar" auch nicht mit den ganzen Standard TP-DOS-Funktionen/Prozeduren (wie FSplit, FExpand,usw.) zusammenarbeitet, da diese "String", bzw. "PathStr" verwenden. Wenn ich das also richtig sehe, dann wuerden die Anwender meiner Toolbox unter Pascal Schwierigkeiten bekommen. Fuer die Anwender der Toolbox aus Assembler herraus, waere es perfekt, da auch dort der "Path" vom INT21h nullterminiert gelesen und geschrieben (im DTA) wird.

2) Wuerde ich mein FindFirst mit "Path : String" deklarieren, so wuerde es von der Pascalseite keine Probleme geben. Auch den Asseblercode NUR auf Pascal-Strings trimmen ist kein Problem.

Allerdings, waere es dann sehr aufwendig, wenn nicht gar unmoeglich - so wie ich meine, eine Toolbox zu programmieren, die den GLEICHEN Code fuer Pascal- UND Assembleraufrufe benutzt.

Wird naemlich ein "Path : String" oder ein "Path:PChar) an meine Asm-Prozedure uebergeben, so erhalte ich dann ja beiden Faellen einen Zeiger auf einen String auf dem Stack. Allerdings kann man nicht herrausfinden, ob es sich nun dabei um einen Pascal-String oder einen nullterminierten String handelt!!!

Damit kann ich also auch in meinem Asm-Code nicht ermitteln, ob ich mit "my_String[0]" im Asm nun das erste Zeichen oder die Stringlaenge addressiere. Ich hoffe, Du verstehst jetzt, welche Zwickmuehle ich hier gerade beschreibe. :-)

Ausserdem ist diese Findfirst-Deklaration mit "Path : String" dann ja nicht mehr ganz identisch mit der von TP , was mich natuerlich "stoert" !

Was soll' ich denn nun machen...? >:-|

Viele Gruesse
XYZ

--- Ende 2. Frage

--- Anfang 2. Antwort

Hallo XYZ,

am besten du folgst der TP6-Hilfe :

```
  FindFirst(Path: string; Attr: Word; var S: SearchRec);
```

Mit dieser Deklaration sollten deine Probleme behoben sein. Da ich von TP6 ausgegangen bin, sind mir derartige Probleme waehrend der Aufgabenstellung nicht aufgefallen.

Viele Gruesse
Birger

--- Ende 2. Antwort

Mit freundlichen Gruessen

Birger