Zeichenketten (Strings) in volksFORTH#

Hier befinden sich grundlegende Routinen zur Stringverarbeitung. Vor allem wurden auch Worte aufgenommen, die den Umgang mit den von manchen Betriebssystemen geforderten 0-terminated Strings ermöglichen. FORTH hat hier gegenüber C den Nachteil, daß FORTH-Strings standardmäßig mit einem Count-Byte beginnen, das die Länge des Strings enthält. Ein abschließendes Zeichen (z.B. ein Null-Byte) ist daher unnötig. Wenn das Betriebssystem aber in C geschrieben wurde (Atari TOS, MS-DOS), müssen Strings entsprechend umgewandelt werden.

Standardmäßig arbeitet FORTH mit counted Strings, die lediglich durch eine Adresse gekennzeichnet werden. Das Byte an dieser Adresse enthält die Angabe, wie lang die Zeichenkette ist. Auf dieses "count byte" folgt dann die Zeichenkette selbst. Dadurch ist die Läunge eines Standard-Strings in FORTH auf 255 Zeichen begrenzt. Die kürzeste Zeichenkette ist ein String der Länge NULL, für dessen Überprüfung der Befehl **NULLSTRING?** zur Verfügung steht.

So sieht der String FORTH an der Adresse addr im Speicher unter FORTH aus.

- ."
- "
- ,"
- nullstring?
- "lit
- .(
- (
-) - dies ist kein Forth Wort, sondern ein Stoppzeichen

String-Manipulationen#

Hier im Glossar bezeichnet der Stackkommentar (string --) die Adresse eines counted Strings, dagegen (addr len --) die Charakterisierung durch die Anfangsadresse der Zeichenkette und ihre Länge.

Keine Stringvariable? - Benutze:

```
: String:  Create dup , 0 c, allot DOES> 1+ count ;
```

- caps
- capital
- upper
- capitalitze
- /string
- -trailing
- scan
- skip
- ?"
- bounds
- type
- >type
- place
- attach
- append
- detract
- match

- search

Im Dictionary#

- (find
- find

0-terminated Strings#

Es gibt noch eine andere Darstellungsform für Strings, die beispielsweise für MS-DOS geeigneter ist. Diese Strings werden zwar ebenfalls durch eine Adresse gekennzeichnet; diese Adresse enthält aber kein count byte. Statt dessen werden diese Zeichenketten mit einem Nullbyte abgeschlossen.

- asciz
- >asciz
- counted

Konvertierungen: Strings -- Zahlen#

String in Zahlen wandeln#

- digit?
- accumulate
- convert
- number?
- number
- dpl

Ein Beispiel der Umwandlung von Zeichen in Zahlen:

In FORTH wird die Eingabe von Zahlen oft mit der allgemeinen Texteingabe und über die Befehle zur Umwandlung von Strings in Zahlen realisiert. In der Literatur wird dazu oft diese Lösung mit **QUERY** angeboten:

```
: in#  ( string -- d tf  n tf  addr ff )
   query bl word  number? ;
```

Diese Lösung ist ungünstig, da **QUERY** den **TIB** löscht. Zugleich stellt die Definition von **NUMBER?** eine unglückliche Stelle im volksFORTH dar. Es gibt im Laxen&Perry-F83 ein Wort gleichen Namens, das ganz anders (besser!) mit den Parametern umgeht. Hier folgt die Definition des F83-NUMBER?, das auf dem volksFORTH **NUMBER?** aufbaut:

```
: F83-NUMBER?  ( string -- d f )
  number?  ?dup IF 0< IF extend THEN true exit THEN
  drop 0 0 false ;
```

Damit stellt das Wort **INPUT#** eine wenig aufwendige Zahleneingabemöglichkeit für 16/32Bit-Zahlen dar:

```
\ input#
: input#  ( string -- d f )
  pad c/l  1- >expect     \ get 63 char maximal
  pad F83-number? ;       \ convert string->number
```

So kann der Anwender das übergebene Flag auswerten und die doppelt-genaue Zahl entsprechend seinen Vorstellungen einsetzen, im einfachsten Fall mit DROP zu einer einfachgenauen Zahl machen.

Zahlen in Strings wandeln#

- #
- #s
- hold
- sign
- #>