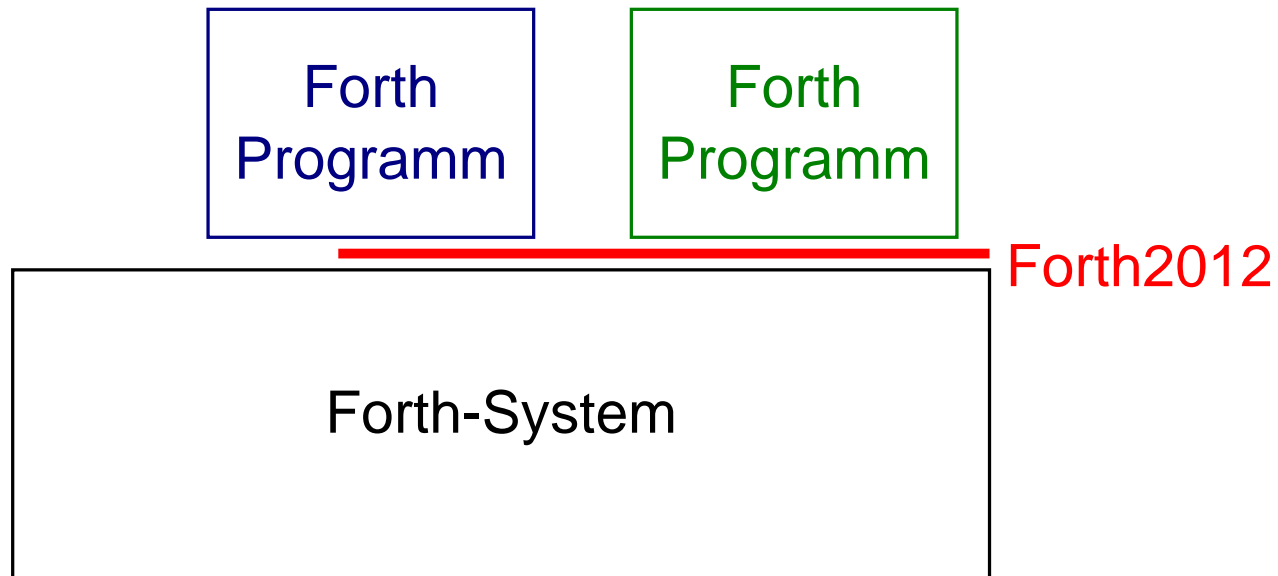


Forth-2012
Der neue Standard

M. Anton Ertl

Was ist ein Standard?



Was ist Forth-2012?

- Weiterentwicklung von Forth-94 (ANS Forth)
- Forth-94-programme sind auch Forth-2012-Programme
- Wenig Revolutionäres
- www.forth200x.org/documents/forth-2012.pdf
- <http://www.forth200x.org/documents/html/contents.html>
- Änderungen zu Forth-94 im Anhang C.7

Entfernte Wörter/Features

- Entfernung in Forth-94 angekündigt
- `tib #tib`, Ersatz: `source`
- `convert`, Ersatz: `>number`
- `expect span`, Ersatz: `accept`
- `query`, Ersatz: `accept ... evaluate`
- `word` lässt keinen Space hinter dem Wort
- `Wordset queries s" string" environment`, Ersatz: `[defined]`
- Nicht entfernt: `forget`

Jenseits von ASCII – XChars

- Z.B. für UTF-8-codierten Unicode
- String besteht aus u Bytes
- Wieviele Zeichen? Meistens egal
- Type funktioniert auch mit UTF-8
- UTF-8 nicht nur für Daten, sondern für alles
z.B. Wortnamen
: לאכול ." จักรกฤษณ์" ; ok
or לאכול
จักรกฤษณ์ ok
- Selten: Betrachtung einzelner Zeichen
Dafür Xchar wordset

Records / Structs

```
begin-structure flist
  field: flist-next
  ffield: flist-val
end-structure
```

```
0
```

```
  1 cells +field flist-next
  faligned 1 floats +field flist-val
constant flist
```

```
here falign flist allot constant flist1
0 flist1 flist-next !
1.23e flist1 flist-val f!
```

Zahlen mit expliziter Basis

`$ff`

`#99`

`%11`

`'a'`

`$-ff.`

Funktionstasten abfragbar

```
... ekey ekey>fkey if
  case
    k-up                               of ... endof
    k-f1                               of ... endof
    k-left k-shift-mask or k-ctrl-mask or of ... endof
    ...
  endcase
else
  ...
then
```


Neue Syntax für lokale Variablen

- Geänderte Reihenfolge

Alt: `: swap (a b -- b a) locals| b a | b a ;`

Neu: `: swap (a b -- b a) {: a b :} b a ;`

- Auch für Stack-Effekt-Kommentar:

`: swap {: a b -- b a :} b a ;`

- Uninitialisierte lokale Variablen

`: foo {: a b | c d -- e f :} ... ;`

- `locals|` wird veraltet

Einmaliges Laden von Dateien

- in `flip.fs`
`require foo.fs`
- in `flop.fs`
`require foo.fs`
- in `bar.fs`
`require flip.fs require flop.fs`
- `foo.fs` wird nur einmal geladen
- `flip.fs` und `flop.fs` alleinstehend nutzbar

Platzhalter

```
defer binop ( n1 n2 -- n3 )  
' - is binop  
4 3 binop . \ 1  
action-of binop ( xt )  
' * ' binop defer!  
4 3 binop . \ 12  
( xt ) is binop  
4 3 ' binop defer@ execute . \ 1
```

Reflexion

```
: words-count ( x nt -- x' f ) drop 1+ true ;
0 ' words-count forth-wordlist traverse-wordlist .

: all-words ( nt -- f ) name>string cr type true ;
' all-words get-current traverse-wordlist

\ name>interpret ( nt -- xt | 0 )
\ name>compile ( nt -- x xt )
( nt ) name>compile execute \ compilation semantics
```

Test auf Wort

```
[undefined] defer [if] ... [then]
```

```
[defined] k-up [if] ... [then]
```

Wort parsen

```
parse-name foo parse-name bar type type
```

```
parse-name ( -- addr u ) beläßt Wort im Puffer
```

Werfen von IORs

```
... open-file throw ...
```

String-Literale

```
s\" abc \"def\nghi\tjkl\"
```

Synonyme

```
synonym endif then
```

xVALUE

```
5. 2value d 6. to d  
5e fvalue f 6e to f
```

Uninitialisierte Puffer

```
500 cells buffer: b  
b 500 cells erase
```

Return Stack

```
get-order n>r  
nr> set-order
```

Textersetzung

```
: datum1 s" 2015-04-11" ; datum1 s" date" replaces
: zeit1 s" 21:00" ; zeit1 s" time" replaces
s" Um %time% am %date% ..." pad 100 substitute throw type
\ ergibt "Um 21:00 am 2015-04-11 ..."
s" On %date% at %time% ..." pad 100 substitute throw type
\ ergibt "On 2015-04-11 at 21:00 ..."
```

Text(teil)e unverändert lassen:

```
500 buffer: buf
s" zufälliges %date% im Text" buf unescape ( c-addr u )
pad 100 substitute throw type
\ ergibt "zufälliges %date% im Text"
```


Gleitkomma-Arithmetik

- Separater Gleitkommastack
- `s>f f>s`
- `ftrunc`
- Genauere Spezifikation von `fasinh fatan2`

Ausblick

- `locals| [compile]` werden veralten
- Recognizers
- 100. langfristig Gleitkomma
*123 für doubles
- 2er-Komplement
- `1 chars = 1`
- 16-bit Speicherzugriffe etc.
- `: count-words (wid -- n)`
`0 [: drop 1+ true ;] rot traverse-wordlist ;`
- multi-tasking multi-threading
- C Interface