



Ein einfaches Quellcode–Bibliothekssystem

Ulrich Hoffmann

Fachhochschule Wedel
University of Applied Sciences

Jahrestagung 2008 der Forth Gesellschaft e.V.



Conventional wisdom reverses complexity.

Gliederung

1 Motivation

- Es gibt keine weit verbreiteten Bibliotheken für Forth.
- Wie arbeiten wir heute?
- Wie würden wir gerne arbeiten?

2 Quellcode–Bibliothekssystem

- Die Bibliotheks–Schnittstelle
- Realisierung der Benutzersicht



Es gibt keine weit verbreiteten Bibliotheken für Forth.

Motivation

"If you've seen one Forth, you've seen one Forth."

Wil Baden

- viele Implementierungen nützlicher Algorithmen in Forth
- Sammelsurium für verschiedene Forth-Systeme
- heutige Erfahrung: *nehmen, lädt nicht*
- Ziel: *nehmen, laden, geht*
- ANS-Forth schafft Voraussetzung



Es gibt keine weit verbreiteten Bibliotheken für Forth.

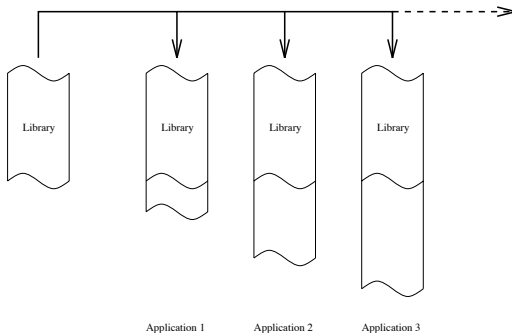
Forth–Bibliotheks–Projekte

- Forth Scientific Library
überwiegend mathematische Algorithmen
- Die Bibliothek von iForth (Marcel Hendrix)
- FFL — Forth Foundation Library (Dick van Oudheusden)
- euroForth 2007 Bibliotheks–Aktivität



Wie arbeiten wir heute?

Wie benutzen wir Bibliotheken?

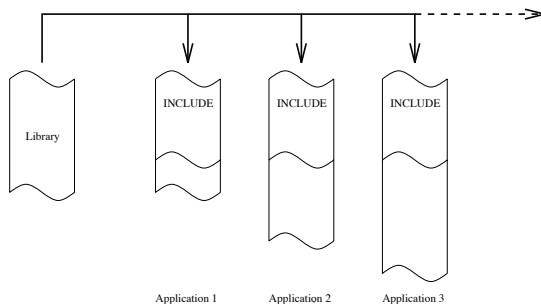


- Bibliothek wird für jede Applikation kopiert und angepasst.
- Software-Engineering-Albtraum



Wie arbeiten wir heute?

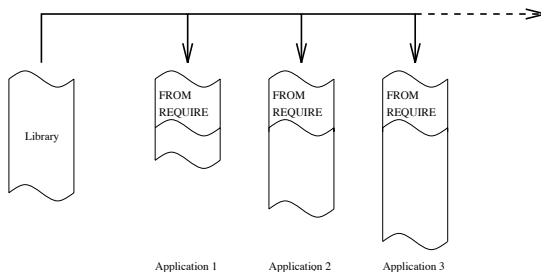
Die Bibliothek nicht immer kopieren



- Die Bibliothek wird unverändert und vollständig eingebunden.
- *Und wenn ich nur eine Funktion daraus brauche?*

Wie würden wir gerne arbeiten?

Nur Teile einer Bibliothek laden



- Die Applikation fordert die gewünschte Teilfunktionalität an.
- Die Bibliothek stellt Teile zur Verfügung.
- **Wie muss eine Bibliothek dafür strukturiert sein?**



Bibliotheks-Schnittstelle

Die Applikation fordert die gewünschte Teilfunktionalität an.

Wie sieht dafür die einfachste Schnittstelle zur Bibliothek aus?



Bibliotheks-Schnittstelle

Die Applikation fordert die gewünschte Teilfunktionalität an.

Wie sieht dafür die einfachste Schnittstelle zur Bibliothek aus?

```
( c-addr u -- )
```



Bibliotheks-Schnittstelle

Die Applikation fordert die gewünschte Teilfunktionalität an.

Wie sieht dafür die einfachste Schnittstelle zur Bibliothek aus?

```
( c-addr u -- )
```

Die Bibliothek macht mit diesem String, was sie für sinnvoll hält.



Struktur einer einfachen Bibliothek

```

\ SAMPLE.LIB: Sample Library with sequential search  uho Mar95

( c-addr u -- )

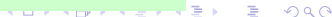
\ do not look up, if identification string is already
\ in dictionary

2DUP PAD CHAR+ SWAP MOVE DUP PAD C!
PAD FIND NIP [IF] 2DROP \ \ [THEN]

CR .( looking for ) 2DUP TYPE SPACE

desires case? [IF] \ -----
: case? ( flag -- ) OVER = DUP IF NIP THEN ;
\ \ [THEN]
...

```





Struktur einer einfachen Bibliothek

```

...

desires on [IF] \ -----

: on ( a-addr -- ) TRUE SWAP ! ;

\\ [THEN]

desires off [IF] \ -----

: off ( a-addr -- ) FALSE SWAP ! ;

\\ [THEN]

\ -----

CR TYPE .( NOT provided! ) QUIT

```



Struktur von Bibliotheken

- Einfache Bibliothek realisiert lineare Suche.
- Andere Möglichkeiten denkbar
 - binäre Suche
 - Hashen

 - Wildcards
- Je nach Anforderungen anpassbar,
dank einfacher Schnittstelle

Implementierung

Worte für die Definition der Bibliothek:

```
\ LIBRARY.ANS: Simple Source Code Library system          uho Mar 95
\ $Id: library.ans,v 1.3 2004/03/24 19:41:03 uho Exp $
: \ ( -- )
  SOURCE-ID 1+ 1 U> IF BEGIN REFILL 0= UNTIL THEN
  POSTPONE \ ; IMMEDIATE

: desired ( c-addr1 len1 c-addr2 len2 -- c-addr1 len1 false | true )
  2OVER COMPARE 0= DUP IF NIP NIP THEN ;

...

```



Aus Benutzersicht

Die Applikation fordert die gewünschte Teilfunktionalität an.

Aus Benutzersicht

Die Applikation fordert die gewünschte Teilfunktionalität an.

```
■ S" case?" INCLUDE library.fs
```

Aus Benutzersicht

Die Applikation fordert die gewünschte Teilfunktionalität an.

- `s" case?" INCLUDE library.fs`
- Problem: `s"` benutzt nur einen Puffer.
- Überschreiben bei geschachtelten Anforderungen

Aus Benutzersicht

Die Applikation fordert die gewünschte Teilfunktionalität an.

- `s" case?" INCLUDE library.fs`
- Problem: `s"` benutzt nur einen Puffer.
- Überschreiben bei geschachtelten Anforderungen
- String muss temporär gesichert werden.

Syntaktischer Zucker

Auf der Benutzungsebene:

- Wahl der Bibliothek mit **from**
- Wahl der Teilfunktionalität mit **require**

```
from sorting require quicksort
```

```
require bubblesort
```

Syntaktischer Zucker

Auf der Benutzungsebene:

- Wahl der Bibliothek mit **from**
- Wahl der Teilfunktionalität mit **require**

```
from sorting require quicksort  
  
                require bubblesort
```

```
{ from <library-name> { require <identification-string> }* }*
```

Implementierung

Worte für die Benutzung von Bibliotheken:

```

...

    BL WORD COUNT desired ;

CREATE $library 256 CHARS ALLOT

: from ( <spaces>ccc<spaces> -- )
    BL WORD COUNT $library CHAR+ SWAP DUP >R CMOVE R> $library C! ;

: required ( c-addr len -- )
    DUP ALLOCATE THROW DUP >R SWAP 2DUP 2>R CMOVE 2R>
    $library COUNT
    DUP ALLOCATE THROW SWAP 2DUP 2>R 2DUP 2>R CMOVE 2R>
    [' ] INCLUDED CATCH
    2R> 2DUP $library CHAR+ SWAP CMOVE $library C! FREE THROW
    R> FREE THROW THROW ;

: require ( <space>ccc<spaces> -- )
    BL WORD COUNT required ;

from extensions.f

CR . ( Simple Source Code Library System initialized! )

```



Zusammenfassung

- Ein Outer-Interpreter rockt.
- Implementierung ist extrem einfach und
- kann leicht in jedes System integriert werden.
- Bei mir in Benutzung seit 1995.

Ausblick

- Zusammenspiel mit der euroForth 2007-Beschreibung von Library-Metadaten
- Wie kommen wir also zu *nehmen, laden, geht*-Bibliotheken?