

SWIG Erweiterung für Forth Neuigkeiten

Gerald Wodni
gerald.wodni@gee.at

M. Anton Ertl
anton@mips.complang.tuwien.ac.at

TU Wien

16.4.2011

Motivation

- C-Libraries von Forth aus benutzen
- Funktionsdeklarationen direkt aus dem Header gewinnen
- Konstanten & *Structs* ebenso

```
extern float getFloatPointer( float a, float *b, float **c );
```

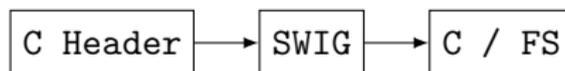


```
c-function getFloatPointer getFloatPointer r a a -- r
```

GForth

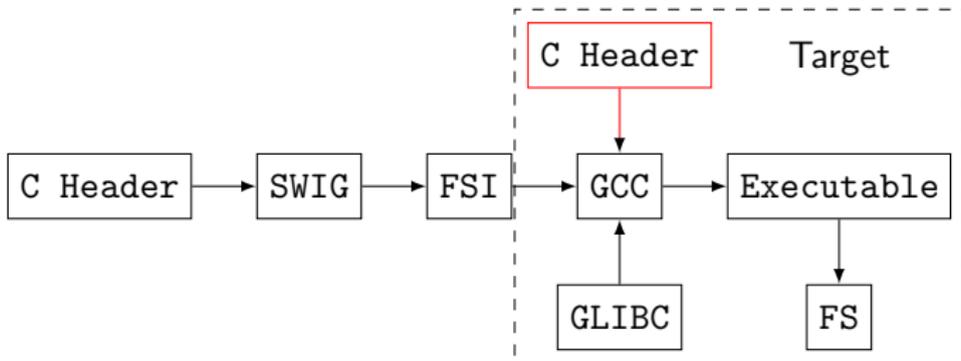
- Keine direkte Implementierung der ABI
- GCC wird zur Laufzeit eingebunden
- Plattformunabhängig
- Words: `c-function`, `add-lib`, `c-callback`

SWIG



- C/C++ Compiler mit individueller Ausgabe
- Wird für C Interfaces zu anderen Sprachen wie PHP oder Python benutzt, auch nicht-Scriptsprachen wie Java oder Lua
- Generiert C-Source, welcher in den Compiler/Interpreter kompiliert wird
- In unserem Fall soll allerdings Forth-Source erzeugt werden

FSI - Forth Script Independent



- Generiert C-Source, Konstanten werden auf der Zielmaschine aufgelöst
- Muss nur einmal auf einer beliebigen Plattform übersetzt werden
- C-Kompiler ist eher installiert als SWIG
- Plattformunabhängig

Konstanten

- Integers → "constant"
- Enums werden wie Integerkonstanten behandelt
- Floats → "fconstant"
- Strings → words (z.B. : TITLE s" SWIG-GForth-Interface" ;)

```
#define thaNAME "glforth_␣rocks"
#define Float 1.23
#define Int 23
#define Hex 0x33
#define Oct 077

enum DAYS{
  MONDAY=1,
  TUESDAY,
  WEDNESDAY=10,
  THURSDAY
};
```

```
\ _____ int _____
23 constant Int
$33 constant Hex
63 constant Oct
```

```
\ _____ float _____
1.23 fconstant Float
```

```
\ _____ string _____
: thaNAME s"␣glforth_␣rocks" ;
```

```
\ ==< enums >==
```

```
\ enum DAYS
1 constant MONDAY
2 constant TUESDAY
10 constant WEDNESDAY
11 constant THURSDAY
```

Typen

- C-Typen werden mit SWIG-typemaps in Forth-Typen übersetzt
- Alle pointer werden zu *addr* transformiert
- Structs werden gerade diskumplimentiert

Funktionen

- Stackeffekt mit originalen Parameternamen, (zumeist ist "n n n d - n" wenig hilfreich)
- "forthiefy" Funktionsnamen (get_nextItem wird zu get-next-item)

```
extern char *fgets(char *s, int size, FILE *stream);
extern float getFloatPointer( float a, float *b, float **c );
```

wird zu

```
c-function fgets fgets a n a — a ( s size stream — )
c-function get-float-pointer getFloatPointer r a a — r ( a b c — )
```

Demo

```
\ library laden
s" mysqlclient" add-lib

\ funktionen und konstanten laden
require mysql.fs

\ connection element anlegen
0 mysql_init constant connection

\ verbinden
connection s\" localhost\0" drop s\" forth\0" drop s\" h4x0r
\0"
drop s\" forth\0" drop 0 0 0 mysql_real_connect if ...

\ sql query absetzen
connection s\" SELECT * FROM 'systems' \0" drop mysql_query
if ...
```

Demo

```
: show-result ( -- )
  \ result holen
  connection mysql_use_result
  begin
    dup mysql_fetch_row ?dup
  while
    connection mysql_field_count 0 u+do
      dup @ tab .cstr
      cell+
    loop drop
  cr
  repeat
  mysql_free_result ;

show-result

connection mysql_close
```

Bibliotheken

- OpenGL
 - War die erste, welche problemlos übersetzt wurde
- GLIBC
 - Benötigt auf manchen Plattformen manuelles Eingreifen
- SDL
- MySQL
- *Gtk*
 - In Arbeit
- andere...
 - theForth.net: Sammlung von FSI-Dateien, welche einfach heruntergeladen und installiert werden können

Zusammenfassung

- Neben GForth werden jetzt auch SwiftForth und VFX unterstützt
- Keine FS-Erstellung mehr, dafür bessere FSI-Dateien
- Einfache Anbindung bestehender Bibliotheken
- Wachsende Sammlung auf theforth.net