

SWIG-GForth-Extension

Gerald Wodni

gerald.wodni@gee.at

M. Anton Ertl

anton@mips.complang.tuwien.ac.at

TU Wien

Euroforth 2009



Motivation

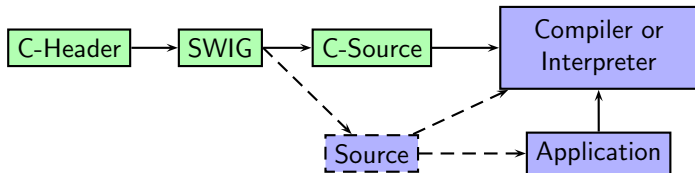
- Foreign-function-interface
- Getting the function-footprint directly from the C-Header
- Convert nested headers instead of searching definitions and constants
- GLForth: OpenGL & SDL where imported by hand

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



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

SWIG [1]



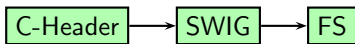
- C/C++ Compiler with custom output
- Typically used for C-interfaces to other languages like PHP or Python, but also non-scripting languages like Java or Lua
- Generates C-source, which is for example compiled into interpreter/compiler.
- In our case we also need to generate Forth-source



Output-Types

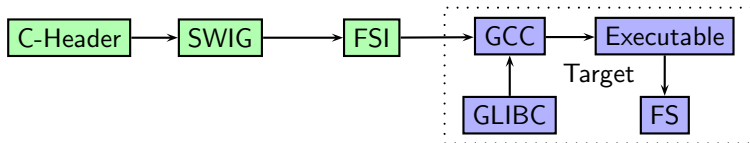
- FS
 - Header file is directly converted into a Forth-source
- FSI (Independent)
 - Generates C-code, compiled and executed on target machine

FS-Files



- Header file is directly converted into a Forth-source
- Easy to use but requires SWIG
- Only necessary when an interface to a custom library is needed
- Platform dependent

FSI-Files



- Generates C-code, constants are resolved using the compiler on target machine
- Only needs to be generated once on any platform
- C-compiler more likely to be installed than SWIG
- Platform independent

Constants

- Integers → "constant"
- Enums are treated as integer constants
- Floats → "fconstant"
- Strings → words (e.g. : TITLE s" SWIG-GForth-Interface" ;)

```
#define thaNAME "g|forth_└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
```

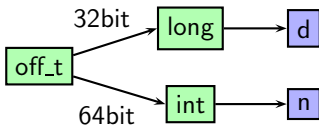
```
\ _____ float _____
1.23 fconstant Float
```

```
\ _____ string _____
: thaNAME s"└g|forth_└rocks" ;
```

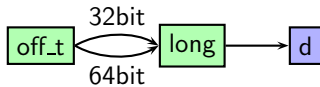
```
\ ==< enums >==
```

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

Types



different architectures



with option `-m32`

- `off_t` could either be `n` or `d`
- As `n` fits into `d`, `-m32` gcc-option is used to create 32/64bit independent code
- SWIG-typemaps used for mapping C-types into Forth
- All pointers are transformed into `addr`
- Structs are currently omitted (no direct namespace mapping)

Functions

- stackcomments with original parameter names, (sometimes "n n n d - n" is not helpfull)
- "forthieffy" function names (get_nextItem becomes get-next-item)

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

becomes

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

Example

```

...
#ifdef _SIZEOF_PTHREAD_BARRIERATTR_T
printf( "%ld\tconstant _SIZEOF_PTHREAD_BARRIERATTR_T\n", (
    long) _SIZEOF_PTHREAD_BARRIERATTR_T );
#endif
#ifdef _ALLOCA_H
printf( "%ld\tconstant _ALLOCA_H\n", (long) _ALLOCA_H );
#endif

/* —≡< functions >≡— */
printf( "\\ —≡< functions >≡—\n" );
printf( "\t\t( — )\n" );
printf( "c-function\t__ctype_get_mb_cur_max\
    \t__ctype_get_mb_cur_max\t— n\n" );
printf( "\t\t( __nptr — )\n" );
printf( "c-function\tatof\tatof\ta — r\n" );
printf( "\t\t( __nptr — )\n" );
printf( "c-function\tatoi\tatoi\ta — n\n" );
printf( "\t\t( __nptr — )\n" );
printf( "c-function\tatol\tatol\ta — n\n" );
...

```



Libraries

- OpenGL
 - First to compile without problems
- GLIBC
 - Already compiled to FSI-Files
 - Requires some human interaction
- others...
 - Creating a collection of FSI files, which could be downloaded and installed easily

Conclusion

- SWIG suitable for outputting other languages than C
- It's possible to adopt the configuration files to serve other Forth-C-Interfaces
- FSI collection

References

- 1 SWIG Manual <http://www.swig.org/Doc1.3/Extending.html>. 1995-2009.
- 2 Neal Crook, Anton Ertl, David Kuehling, Bernd Paysan, Jens Wilke. GForth-Manual. 1995-2008.