



## Aufruf von C-Funktionen in Perl

**Autor:** Konstantin Tobros

**E-Mail:** konstantin AT ruhr.pm.org

**Datum:** 15. April 2008

<http://ruhr.pm.org/>

Dieses Dokument wurde veröffentlicht unter der Lizenz

## Creative Commons Attribution-Noncommercial-NoDerivs 2.0 Germany

Die Lizenz sowie entsprechende Übersetzungen sind einsehbar unter:  
<http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Zusammenfassend ergeben sich hieraus die folgenden Rechte:



Sie dürfen das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen.

Diese Rechte werden Ihnen unter den folgenden Bedingungen gewährt:



Namensnennung. Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen (wodurch aber nicht der Eindruck entstehen darf, Sie oder die Nutzung des Werkes durch Sie würden entlohnt).



Keine kommerzielle Nutzung. Dieses Werk darf nicht für kommerzielle Zwecke verwendet werden.



Keine Bearbeitung. Dieses Werk darf nicht bearbeitet oder in anderer Weise verändert werden.

Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter welche dieses Werk fällt, mitteilen.

Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die Einwilligung des Rechteinhabers dazu erhalten.

Diese Lizenz lässt die Urheberpersönlichkeitsrechte unberührt.



# Ruhr . pm

---

## Motivation

- hoehher
- schneller
- weiter



# Ruhr . pm

## Theorie

	<i>Glue Code</i>		
	C	<----->	Perl
(unsigned) int16/32/64	sv		
float/double	sv		\$sv
char *	sv		
char/int/float/double	av[]		@av
?! struct hv { int k; int v; }; ?!			%hv



# Ruhr . pm

---

## Moeglichkeiten

- xsubpp            bei Perl dabei!
- SWIG            <http://www.swig.org/>
- Glue Code selbst verfassen!?



# Ruhr . pm

---

## SWIG

- viele Sprachen
- sehr viele Sprachen
- einfacher als xsubpp



# Ruhr . pm

---

## xsubpp

- nur Perl
- viel Freiheit



# Ruhr.pm

## xsubpp in Aktion

- mit h2xs Modulskelett erstellen
- Makefile.PL editieren
- .xs Datei editieren
- Funktionen in .pm Datei exportieren und Modulbeschreibung erstellen
- (Tests schreiben)
- `$ perl Makefile.PL && make && make test/install`





# Ruhr . pm

## ein paar h2xs optionen

- `-n <name>` (Perl-)Name des Moduls
- `-X` .xs-Datei nicht erzeugen
- `-b x.y.z` kompatibel zu Perlversion x.y.z
- `-A` AutoLoader auslassen
- `-skip-ppport` keine Portierbarkeit zu alten Versionen von Perl
- `-x` .xs-Datei automatisch erzeugen



# Ruhr.pm

## XSUB Syntax

```
rueckgabetyp  
perlname( var1, var2 )  
    typ var1;  
    typ var2;
```

OUTPUT:

RETVAL

Funktionsdeklaration

folgende Daten werden an Perl  
weitergereicht

spezielle Variable fuer den  
Rueckgabewert



# Ruhr . pm

## XSUB Beispiel

```
double  
sin( a )  
    double a;  
    OUTPUT:  
    RETVAL
```

```
double  
sinus( a )  
    double a;  
    CODE:  
    RETVAL = sin(a);  
    OUTPUT:  
    RETVAL
```

Wenn

`$perlfunc->name eq $cfunc->name`

oder z.B.



# Ruhr.pm

## XSUB Syntax

CODE:

BOOT:

PREINIT:

Hier folgt "echter" Glue Code

extra Bootstrap-Anweisungen

erlaubt Deklaration von  
zusätzlichen Variablen



# Ruhr.pm

## Tests

- `<modulname>/t/<modulname>.t` editieren

```
ok( $bool, $testname );
```

```
is( $var1, $var2, $testname );
```

```
isnt( $var1, $var2, $testname );
```

```
cmp_ok( $var1 , 'op', $var, $testname );
```

ok wenn `$bool == true`

ok wenn `$var1 == $var2`

ok wenn `$var1 != $var2`

ok wenn `$var1 op $var2`



# Ruhr . pm

## wichtige weiterfuehrende Punkte

- IN/OUT/IN\_OUT/OUTLIST/IN\_OUTLIST
- Perl-API
- Pointer allgemein
- \$filehandle
- Beispiele mit @arrays und %hashes



# Ruhr . pm

---

## Fazit

- *Fortsetzung folgt !*

**Danke fuers Nicht-Davonlaufen!**