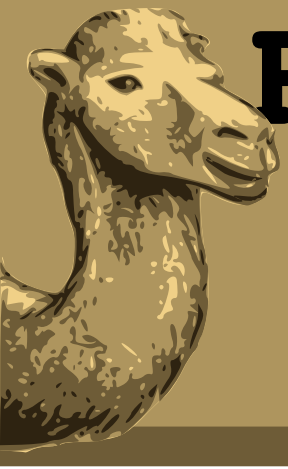
Perl Mongers im Ruhrgebiet



Ruhr.pm

Aufruf von C-Funktionen in Perl

Autor: Konstantin Tobros

EMail: konstantin AT ruhr.pm.org

Datum: 15. April 2008

http://ruhr.pm.org/

Ruhr.pm

Motivation

- hoeher
- schneller
- weiter

Ruhr.pm

Theorie

Glue Code

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

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.

XSUB Syntax

```
CODE:

BOOT:

PREINIT:
```

Hier folgt “echter” Glue Code

extra Bootstrap-Anweisungen

erlaubt Deklaration von zusaetzlichen Variablen

Ruhr.pm

Tests

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

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

ok wenn $bool == true

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

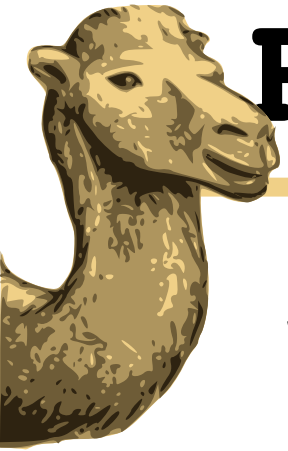
ok wenn $var1 == $var2

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

ok wenn $var1 != $var2

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

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!