Memory::Usage + Devel::Loading

Speicherverbrauch von Perl-Modulen bestimmen

Uwe Völker

XING AG

07.03.2012

Memory::Usage

- misst Speicherverbrauch vom Perl-Prozess
- verschiedene Messpunkte/-zeiträume möglich
- nur unter Linux (/proc)

Memory::Usage

```
> module-size CGI
time     vsz (  diff)    rss (  diff) shared (  diff)   code (  diff)   data (  diff)
   0    4932 (  4932)   1948 (  1948)   1544 (  1544)   1268 (  1268)    676 (   676) Base perl plus Memory
   0    5640 (   708)   2856 (   908)   1680 (   136)   1268 (     0)   1384 (   708) after 'use CGI'
```

Devel::Loading

- zeigt Lade-Reihenfolge von Perl-Modulen
- gelöst über Coderef in INC

Devel::Loading

```
perl -d:Loading -MCGI -e 1
Loading C:/strawberry/perl/site/lib/sitecustomize.pl at -e line 0.
Loading CGI.pm at -e line 0.
Loading Carp.pm at C:/strawberry/perl/lib/CGI.pm line 3.
Loading Exporter.pm at C:/strawberry/perl/lib/Carp.pm line 14.
Loading CGI/Util.pm at C:/strawberry/perl/lib/CGI.pm line 27.
Loading strict.pm at C:/strawberry/perl/lib/CGI/Util.pm line 3.
Loading vars.pm at C:/strawberry/perl/lib/CGI/Util.pm line 4.
Loading warnings/register.pm at C:/strawberry/perl/lib/vars.pm line 7.
Loading warnings.pm at C:/strawberry/perl/lib/warnings/register.pm line 24.
Loading constant.pm at C:/strawberry/perl/lib/CGI.pm line 33.
Loading overload.pm at C:/strawberry/perl/lib/CGI.pm line 3847.
```

Kombination der beiden Module

```perl
BEGIN {
    use Memory::Usage;
    $MU = Memory::Usage->new;
    $MU->record('base');
}
use Devel::Loading sub {$MU->record("before␣$_")};

foreach my $module (@ARGV) {
    eval "use␣$module";
    warn $@ if $@;
}

$MU->record('finished');
$MU->dump;
```