

Devel::TrackObjects

Dieser Vortrag beschäftigt sich mit dem Modul TrackObjects, welches zum Auffinden von perl-Objekten genutzt werden kann, die unerwarteter Weise von Perl nicht zerstört werden.

Autor

Steffen Ullrich <Steffen_Ullrich@genua.de>

- Autor von `Net::SIP`, `Devel::TrackObjects`
- Maintainer von `IO::Socket::SSL`
- CPAN: SULLR
- seit 2001 bei GeNUA mbH, 100% Perl Job

Abstract

Devel::TrackObjects ist ein kleines Perl-Modul, welches zum Auffinden von Speicherlecks durch nichtzerstörte Objekte dienen kann.

Es wurde u.a. benutzt um Speicherlecks innerhalb von Net::SIP aufzudecken.

Was kann Devel::TrackObjects

- Das Modul kann die Erzeugung aller durch Perl-Code erzeugten Objekte, die nach dem Laden des Modules erstellt wurden, kontrollieren.
- Es kann ausgeben,
 - wenn Objekte erstellt werden
 - welche Objekte zu einer bestimmten Zeit (z.B. am Programmende) existieren.
- Das Tracking kann beschränkt werden auf Objekte von bestimmten Klassen.
- Anwendungszweck ist, ohne Änderung am Code, die Erstellung von bestimmten Objekten zu kontrollieren und zu verifizieren, dass diese zu einer bestimmten Zeitpunkt nicht mehr existieren.

Wie benutzt man es - cmdline

- Von der Kommandozeile ohne Änderungen am Programm:

```
perl -MDevel::TrackObjects=/^IO::/ server.pl
```

Das startet das Programm und zeigt beim Beenden an, wie viele Objekte von den verschiedenen IO::* Klassen noch existieren.

Wie benutzt man es - im Code

- Im Programmcode zur Untersuchung bestimmter Teile des Programmes:

```
use Devel::TrackObjects qr/^Net::SIP/, '-verbose', '-debug';
use Net::SIP;
...
Devel::TrackObjects->show_tracked_detailed( "-HIER" );
...
```

- das ergibt zB folgende Ausgabe

```
TrackObjects: register Net::SIP::SDP=HASH(0x195383c) .../Net/SIP/SI
LEAK-HIER >>
-- Net::SIP::SDP=HASH(0x195383c) | .../Net/SIP/SDP.pm:189
LEAK-HIER --
LEAK >> empty --
```

Optionen

Genauere Informationen enthält man über die Switches

- `-debug` (Anzeigen, wenn Objekte erstellt werden) und
- `-verbose` (bei `track_objects` anzeigen, an welchen Stellen die Objekte erstellt wurden):
- `perl -MDevel::TrackObjects=/^IO::/, -debug, -verbose server.pl`

Wie funktioniert es

- Das Modul redefiniert die Funktion `CORE::GLOBAL::bless` und ist damit in der Lage, in die Erstellung der Objekte einzugreifen.
- Beim Aufruf von `bless` wird eine weak Referenz auf das neue Objekt in einer Liste abgelegt.
- Beim Aufruf von `Devel::TrackObjects->show_tracked` wird diese Liste durchlaufen, alle nicht mehr existenten Objekte entfernt und der Rest reported.
- Für mehr Komfort wird `show_tracked` automatisch beim Beenden des Programmes aufgerufen.

Overhead

Der Overhead durch das Überdefinieren von `CORE::GLOBAL::bless` kommt erst, wenn Klassen zum Tracken angegeben werden.

Solange man keine Objekte tracked ist kein Overhead, selbst wenn man ein `use Devel::TrackObjects` ohne Argumente in das Programm einbindet.

Dankeschön

Fragen?