Time::y2038 - Keine Angst vor dem Schwarzen Dienstag 2038

Das *Jahr-2038-Problem* kann möglicherweise zu Softwareausfällen im Februar 2038 auf 32-Bit-Systemen führen.

Michael Schwern's `Time::y2038` schafft auf besonders einfache Art und Weise Abhilfe für Legacy Cody.

Für Windows steht `Time::y2038` leider noch nicht zur Verfügung.

Beispiel mit 2038-Bug:

```
#!/usr/bin/perl
use strict;
use warnings;
#
# Based on http://maul.deepsky.com/
#           ~merovech/2038.perl.txt
#

# Zeitzone auf GMT aka UTC setzen
$ENV{'TZ'} = "GMT";

# Epochsekunden kurz vor bzw. nach dem
# kritischen Event
# in einer Schleife durchlaufen und pruefen,
# ob Datum
# und Uhrzeit korrekt ausgegeben werden.
my $clock;
for ($clock = 2147483641;
     $clock < 2147483651; $clock++) {
      print scalar localtime($clock), "\n";
}
```

Ausgabe auf einem 32-Bit Linux:

```
Tue Jan 19 03:14:01 2038
Tue Jan 19 03:14:02 2038
Tue Jan 19 03:14:03 2038
Tue Jan 19 03:14:04 2038
Tue Jan 19 03:14:05 2038
Tue Jan 19 03:14:06 2038
Tue Jan 19 03:14:07 2038
Fri Dec 13 20:45:52 1901
Fri Dec 13 20:45:53 1901
Fri Dec 13 20:45:54 1901
```

Auf 32-Bit Unix/Linux-Systemen wird die Zeit auf den 13. Dezember 1901 gesetzt.

Ausgabe auf einem 32-Bit Windows:

```
Tue Jan 19 04:14:01 2038
Tue Jan 19 04:14:02 2038
Tue Jan 19 04:14:03 2038
Tue Jan 19 04:14:04 2038
Tue Jan 19 04:14:05 2038
Tue Jan 19 04:14:06 2038
Tue Jan 19 04:14:07 2038
```

Das Programm stoppt genau nach der kritischen Sekunde im Jahr 2038.

Beispiel mit Time::y2038

Dank `Time::y2038` muss nur *eine* Zeile Code hinzugefügt werden und das Programm ist Jahr-2038-fest.

```
#!/usr/bin/perl
use strict;
use warnings;

use Time::y2038;
#
# Based on http://maul.deepsky.com/
#           ~merovech/2038.perl.txt
#

# Zeitzone auf GMT aka UTC setzen
$ENV{'TZ'} = "GMT";

# Epochsekunden kurz vor bzw. nach dem
# kritischen Event
# in einer Schleife durchlaufen und pruefen,
# ob Datum und Uhrzeit korrekt ausgegeben
# werden.
my $clock;
for ($clock = 2147483641;
     $clock < 2147483651; $clock++) {
      print scalar localtime($clock), "\n";
}
```

Ausgabe des Programms (32-Bit Linux)

```
Tue Jan 19 03:14:01 2038
Tue Jan 19 03:14:02 2038
Tue Jan 19 03:14:03 2038
Tue Jan 19 03:14:04 2038
Tue Jan 19 03:14:05 2038
Tue Jan 19 03:14:06 2038
Tue Jan 19 03:14:07 2038
Tue Jan 19 03:14:08 2038
Tue Jan 19 03:14:09 2038
Tue Jan 19 03:14:10 2038
```

Datumsberechnungen (nicht nur über das Jahr 2038 hinaus)

Wer Datumsberechnungen (nicht nur über das Jahr 2038 hinaus) sicher durchführen möchte, sollte sich ohnehin nicht auf `time()` und *Epochem-Sekunden* verlassen, sondern besser ein CPAN-Modul wie `Date::Calc` oder `DateTime` verwenden.

Beispiel:

```
#!/usr/bin/perl
use strict;
use warnings;

use DateTime;

my $dt = DateTime->now();

print $dt->iso8601() , "\n";

$dt->add(years => 200);

print $dt->iso8601() , "\n";
```

Thomas Fahle