

Temperatuurbewaking met Linux (Deel 2)



door Stefan Blechschmidt
<sb(at)sbsbavaria.de>

Over de auteur:

Doordat ik opgeleid ben als electriciën, zat ik in 1990 achter een CAD werkstation om een schakel en controlepost te ontwikkelen. Kennelijk ben ik toen besmet met een toen nog onbekend 'virus' en dat bevalt me prima.

Vertaald naar het Nederlands door:
Hendrik-Jan Heins
<hjh/at/NOSPAM/dot/passys/dot/nl>



Kort:

In de November 2003 editie Temperatuurbewaking met Linux heb ik uitgelegd hoe een circuit werkte dat ons in staat stelde temperatuurinformatie via Linux te lezen. Om deze temperatuurinformatie te kunnen interpreteren, moeten we hem in een database bewaren.

Om echt gebruik te kunnen maken van dit artikel, zullen we de gegevens via een webinterface grafisch weergeven.

Voorzorgsmaatregelen

Er moeten al enkele toepassingen werkend op je systeem aanwezig zijn

- Perl
- Apache
- MySQL
- en enkele Perl modules die het makkelijker maken om programma's aan te maken, maar daarover later meer.

Let wel op, dit artikel is voor de wat verder gevorderde Linux gebruiker. Voor degenen die nog niet zover zijn, is dit een goede introductie in het lastiger werk ;-)

Het opzetten van de Database

In MySQL is het programma *mysql* de interface naar de database. Met het commando `mysql -u root -p mysql` leggen we verbinding met de MySQL bewaking.

Met de parameter `-u` kan de gebruiker worden opgegeven. De parameter `-u` zal een wachtwoord vragen en daarna kan de database die gebruikt gaat worden, geopend worden. In ons geval kiezen we de MySQL database Manager.

Je krijgt de `mysql >` prompt om de SQL commando's op te geven. Allereerst moeten we uitvinden welk soort tabellen er in de database staan. Het commando `show tables;` doet exact dat.

```
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| host            |
| tables_priv     |
| user            |
+-----+
6 rows in set (0.00 sec)
```

Nu moeten we de database voor onze temperatuurgegevens aanmaken. Met het commando `create database digidb` genereren we onze database genaamd *digidb* en met het commando `exit` kunnen we de bewaking sluiten, aangezien we de aanvullende commando's op een andere manier zullen opgeven.

Opmerking:

MySQL heeft een beheerder die meestal ook *root* heet. De standaard installatie heeft geen wachtwoord nodig. Met het commando `mysqladmin -u root -p password geheim` veranderen we het wachtwoord voor de gebruiker *root* in *geheim*.

Om deze verandering te activeren, moet de tabel van de beheerder opnieuw worden ingelezen, we doen dat met het commando `mysqladmin -u root -p flush-privileges`. Van nu af aan moet de gebruiker *root* iedere keer als hij toegang tot de database wil krijgen een wachtwoord opgeven.

Het opgeven van commando's via de bewaking is vrij gecompliceerd, maar MySQL biedt een andere mogelijkheid om commando's op te geven.

Om dit voor elkaar te krijgen, openen we een tekstbestand met SQL commando's en voegen een "<" toe aan het commando *mysql*.

Om dit te demonstreren, schrijven we een tekstbestand dat de eerste tabel voor sensor 0 genereert.

In dit bestand *sensor0.sql* schrijven we nu de commando's om de tabel op te bouwen. Het zal er ongeveer zo uitzien:

```
CREATE TABLE sensor0 (  
  id int(11) NOT NULL auto_increment,  
  monat char(3) NOT NULL default '',  
  tag char(2) NOT NULL default '',  
  dbtime timestamp(14) NOT NULL,  
  zeit time NOT NULL default '00:00:00',  
  messung decimal(4,2) NOT NULL default '0.00',  
  PRIMARY KEY (id)  
) TYPE=MyISAM;
```

Dit wordt met:

```
mysql -u digitemp -p digitemp < sensor0.sql ingegeven.
```

Aangezien we 2 sensoren gebruiken, hoeven we het bestand alleen maar te kopiëren en de regel `CREATE TABLE sensor0` te veranderen in `CREATE TABLE sensor1`.

Op dit punt ben je er waarschijnlijk wel van overtuigd dat het geven van SQL commando's via een bestand echte voordelen biedt.

Controle:

Om de nieuw gegenereerde tabellen te bekijken, gebruiken we het commando: `echo 'show tables' | mysql -u root -p digidb`. Natuurlijk werkt dit ook andersom.

Als we alles goed hebben gedaan, krijgen we de volgende output:

```
Enter password:  
Tables_in_digidb  
sensor0  
sensor1
```

Gegevens invoeren in onze Database

Een klein Perl programma zal de gegevens overbrengen naar de database. Hiervoor wordt onze eerste Perl module (DBI) gebruikt, deze zal ons voorzien van een methode voor databasetoegang.

Opmerking:

Perl modules voor allerlei toepassingen zijn te vinden in de 'Comprehensive Perl Archive Network (CPAN, <http://www.cpan.org/>)' . Ik ga even voorbij aan de beschrijving van de installatie en verwijs je naar:

<http://www.pro-linux.de/news/2002/0070.html>

of

<http://www.linux-magazin.de/Artikel/ausgabe/1997/10/CPAN/cpan.html> .

```

#!/usr/bin/perl -w
#
# Digitemp preparing of log file and saving in database

# sbs 2003-08-09
#
use DBI;
use strict;

# Initialize database

my $datasource = "dbi:mysql:database=digidb";
my $user = "root";
my $pass = "geheim";

my $db = DBI->connect($datasource, $user, $pass)
    or "Verbindung zur Datenbank nicht möglich: " . $DBI::errstr;

# Filtering of Digitemp
while(<STDIN>) {
    chomp;
    # Skip output program name
    next if (m/Digi.*//);
    # Skip output blank line
    next if (m/^\$/);
    # Skip all to Fahrenheit
    m/(.*).F.*//;
    my $templine = $1;

    # Divide temp line and save in variables
    my ($monat, $tag, $zeit, $sensor_txt, $sensor_nr, $grad_txt, $grad_wert)
        = split(/ /,$templine);

    # Fill database
    $db->do( "insert into sensor$sensor_nr (monat, tag, zeit, messung)
        values ('$monat', '$tag', '$zeit', '$grad_wert')" )
        or die "do nicht möglich: " . $db->errstr();
}# END- Digitemp filter

# close database
$db->disconnect;

```

Een korte uitleg van het programma:

Het programma doet eigenlijk niet heel erg veel, het opent de database, leest de output die het ontvangt van *digitemp*, filtert alles uit wat we niet nodig hebben en schrijft de relevante gegevens in de juiste database tabel.

Continue gegevensvergaring kan worden geregeld via een goed geplaatste cron taak:

```
0-59/15 * * * * root /root/bin/digitemp -a | /root/bin/digipipe.pl
```

Dat is alles voor de gegevensvergaring, nu de web interface.

Perl en CGI

Perl geeft ons de juiste omgeving voor deze taak.

Allereerst moeten we weten in welke map Apache CGI programma's uitvoert. Dit kan worden gevonden in de configuratiebestanden van apache. Zoek naar een regel als deze: <Directory /usr/lib/cgi-bin>.

Voordat we beginnen met de grafische output, bouwen we eerst een programma dat ons voorziet van de laatst gemeten gegevens.

Het zou een groot voordeel zijn als je die gegevens kon opslaan in een submap; je moet je programma ook uitvoerbaar maken: `chmod 755 programmaam`.

We moeten de output beperken tot de laatste gegevens en ze in een Perl-CGI programma invoeren. Dit gebeurt via de SQL query.

```
#!/usr/bin/perl

use DBI;
use strict;

# Initialize database
my $datasource = "dbi:mysql:database=digidb";
my $user = "root";
my $pass = "geheim";

my $db = DBI->connect($datasource, $user, $pass)
    or "Verbindung zur Datenbank nicht möglich: " . $DBI::errstr;

# database work parameter
my $sql;
my $sth;

# Sensor work parameter
my $temp;
my $zeit;

#Prepare HTML output
print "Content-type: text/html\n\n";

# Output of individual sensors measurements
$sql = "select messung, zeit from sensor$i order by id desc limit 1;";

$sth = $db->prepare($sql)
    or die "prepare nicht möglich";
$sth->execute()
    or die "execute nicht möglich";
($temp, $zeit) = $sth->fetchrow_array();
$sth->finish();

print "<p>Temperatur Sensor$i: <b>[$temp]</b> $zeit</p>";

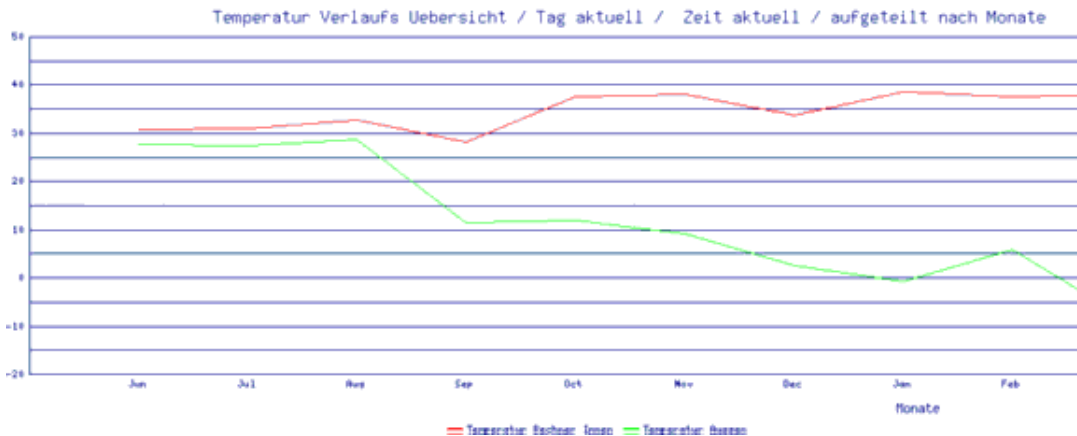
}

# Close database
```

```
$db->disconnect;
```

Dit voorbeeld is niet het meest elegante, maar het toont wel aan hoe eenvoudig deze taak kan worden uitgevoerd met Perl.

Grafische Output



Laten we nu eens kijken naar de grafische output. Het programma (download onderaan dit artikel) genereert krommen. Kijk voor meer grafische programma's naar de andere GD modules.

Het programma gebruikt tevens de CGI module die HTML output genereert met Perl. Ik verwijs hiervoor naar de vele beschrijvingen hierover op Internet.

Terug naar het programma. Het bestaat uit een hoofdbestanddeel en twee sub-programma's. Een sub-programma is verantwoordelijk voor de SQL query, het tweede voor het grafische werk.

Er worden slechts drie queries door het hoofdbestanddeel uitgevoerd en de gegevens worden daarna geparsed naar de sub-programma's.

1. Generatie van de schaal van de X-as
2. Gegevens van de eerste Sensor (sensor0)
3. Gegevens van de tweede Sensor (sensor1)

Alleen de queries hoeven te worden veranderd om een andere grafische output te genereren.

SQL Queries

Tenslotte wil ik je enkele SQL queries laten zien, aangezien ze het belangrijkste onderwerp van dit voorbeeld zijn.

De laatste vijf metingen

```
select tag, monat, zeit,
DATE_FORMAT(dbtime, '%Y-%c-%d %H:%i:%s') as dbtime, messung
from sensor0
order by id desc
limit 5;
```

De koudste dag van het jaar

```
select tag, monat, zeit,
DATE_FORMAT(dbtime, '%Y-%c-%d %H:%i:%s') as dbtime, messung
from sensor1
where YEAR(dbtime) = YEAR(NOW())
order by messung asc
limit 1
```

De warmste dag van het jaar

```
select tag, monat, zeit,
DATE_FORMAT(dbtime, '%Y-%c-%d %H:%i:%s') as dbtime, messung
from sensor1
where YEAR(dbtime) = YEAR(NOW())
order by messung desc
limit 1
```

Het berekenen van de aritmische mediaan (gemiddeld) van de dag

```
select day, month, YEAR(dbtime) as Jahr,
sum(messung)/count(*) as Durchschnitt
from sensor1
where YEAR(dbtime) = YEAR(NOW())
and DAYOFMONTH(dbtime)= DAYOFMONTH(NOW())
and MONTHNAME(dbtime) = MONTHNAME(NOW())
group by DAYOFMONTH(dbtime)
```

Conclusie

Ik ben altijd weer verrast hoe eenvoudig programma's te schrijven zijn in Perl. Eigenlijk schrijf je ook niet echt, maar je kopieert en combineert verschillende delen; op de een of andere manier bestaat ieder stukje al in de een of andere vorm.

Ik hoop dat ik je een kleine inkijk heb kunnen bieden in de onderwerpen Perl, CGI en MySQL.

Download

- CGI Programm "get current values"
(CGI Programma "Output Of Current Measurements")
- CGI Programm "graphical representation, relative to the current day/hour"
(CGI Programma " Graphic Presentation For The Current Day And The Requested Time")

Links / Referenties

- Thuisbasis CPAN
- Artikel Temperatuurbewaking met Linux
- Beschrijving installatie CPAN Module// or
- Referenz (de) mysql
- Thuisbasis Perl

Site onderhouden door het LinuxFocus editors team © Stefan Blechschmidt "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Vertaling info: de --> -- : Stefan Blechschmidt <sb(at)sbsbavaria.de> de --> en: Jürgen Pohl <sept.sapins(Q)verizon.net> en --> nl: Hendrik-Jan Heins <hjh/at/NOSPAM/dot/passys/dot/nl>
---	---