

## Monitoreo de Temperatura con Linux (Parte 2)



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

### *About the author:*

Electricista confirmado, me encontré a mi mismo en 1990 delante de una estación de trabajo CAD para desarrollar una estación de control y conmutación. Obviamente, me infecté por un 'virus' entonces desconocido.

*Translated to English by:*  
Jürgen Pohl  
<sept.sapins(Q)verizon.net>



### *Abstract:*

En la edición de Noviembre de 2003 [Monitoreo de Temperatura con Linux](#) mostré un circuito que nos permitía recoger datos de temperatura a través de Linux. Para evaluar los datos de temperatura, deberíamos guardarlos en una base de datos.

Para obtener la máxima ventaja de este artículo mostraremos los datos gráficamente a través de un interfaz web.

---

## Prerrequisitos

Algunas aplicaciones deberían estar instaladas en tu ordenador

- Perl
- Apache
- MySQL
- Y algunos módulos de Perl que harán más fácil la generación de programas, veremos más sobre esto más tarde.

Como estamos mostrando, este artículo es para usuarios de Linux algo avanzados. Para aquellos que todavía no lo son, será una buena introducción ;-)

# Preparando La Base De Datos

En MySQL el programa *mysql* nos provee de un interfaz hacia la base de datos. Con el comando `mysql -u root -p mysql` conectamos al monitor MySQL.

Con la opción `-u` introduciremos el usuario. La opción `-p` pedirá la contraseña, y finalmente habrá que seleccionar la base de datos que se va a usar. En nuestro caso estamos seleccionando la base de datos maestra de MySQL.

Estarás en el prompt de `mysql >` para introducir los comandos SQL. Al principio necesitaremos descubrir qué clase de tablas hay en la base de datos. El comando `show tables;` hace esto.

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

Ahora tenemos que crear nuestra base de datos para nuestros datos de temperatura. Con el comando `create database digidb` generamos nuestra base de datos llamada *digidb* y con el comando `exit` podemos salir del monitor, ya que los siguientes comandos los introduciremos de otra manera.

## Nota:

*MySQL* tiene un administrador que normalmente se llama también *root*. La instalación por defecto no requiere una contraseña. Con el comando `mysqladmin -u root -p password geheim` estamos cambiando la contraseña del usuario *root* a *geheim*.

Para hacer esta modificación efectiva, la tabla del administrador tiene que ser leída de nuevo, hacemos eso con el comando `mysqladmin -u root -p flush-privileges`. Desde ahora el usuario *root* tiene que proporcionar la contraseña cada vez que accede a la base de datos.

Escribir comandos a través del monitor es muy complicado, sin embargo *MySQL* ofrece otra posibilidad para escribir comandos.

Para hacer esto introducimos un archivo de texto con los comandos SQL añadiendo un `<` al comando *mysql*.

Para mostrar esto, escribiremos un archivo de texto para generar la primera tabla para sensor 0.

En este archivo *sensor0.sql* escribimos los comandos para crear la tabla, debería parecerse a esto.

```
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;
```

Lo introduciremos con:

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

Como estamos utilizando sólo dos sensores, necesitaremos sólo copiar el archivo y modificar la línea  
CREATE TABLE sensor0 por CREATE TABLE sensor1.

Llegados a este punto debería ser más convincente que es más ventajoso escribir comandos SQL por medio de un archivo.

### Probando :

Para mostrar las nuevas tablas generadas utilizamos el comando: `echo 'show tables' | mysql -u root -p digidb`, por supuesto también funciona en orden inverso.

Si lo hemos hecho todo correctamente tendremos la siguiente salida:

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

## Introduciendo Datos en nuestra Base de Datos

Un pequeño programa en Perl hará la transferencia de datos a la base de datos. Para esto será utilizado nuestro primer módulo de perl (DBI), nos proporcionará el método de acceso a la base de datos.

### Nota:

Módulos de Perl para toda clase de aplicaciones pueden ser encontrados en 'Comprehensive Perl Archive Network (CPAN, <http://www.cpan.org/>)' . Me salto la descripción de la instalación y te remito a:

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

or

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

.

```
#!/usr/bin/perl -w
#
# Digitemp preparación de el archivo de log y guardándolo
en la base de datos

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

# Inicialización de la base de datos

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;

# Filtrado de Digttemp
while(<STDIN>) {
  chomp;
  # Saltando en la salida el nombre del programa
  next if (m/Digi.*//);
  # Saltando en la salida las líneas en blanco
  next if (m/^\$/);
  # Saltando todo en Farenheit
  m/(.*).F.*//;
  my $templine = $1;

  # Dividiendo linea de temperatura y guardando en variables
  my ($monat, $tag, $zeit, $sensor_txt, $sensor_nr, $grad_txt,
    $grad_wert)
    = split(/ /,$tempzeile);

  # Llenando la base de datos
  $db->do( "insert into sensor$sensor_nr (monat, tag, zeit, messung)
    values ('$monat', '$tag', '$zeit','$grad_wert')"
    or die "do nicht möglich: " . $db->errstr();

}# Final -Filtro Digttemp

# cerrando la base de datos
$db->disconnect;

```

## Una breve explicación del programa:

El programa realmente no hace demasiado, abre la base de datos, lee la salida que recibe de *digitemp*, filtra todo lo que no necesitamos y escribe los datos relevantes dentro de la tabla de la base de datos correcta.

La recogida de datos es realizada con la siguiente entrada en el cron:

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

Eso es todo para la recolección de datos, ahora el interfaz web.

## Perl y CGI

Perl nos ofrece el entorno adecuado para este trabajo.

Lo primero que necesitamos es conocer dónde está el directorio donde Apache está procesando sus programas CGI. Se puede encontrar en el archivo de configuración de Apache. Busca una entrada como ésta `<Directory /usr/lib/cgi-bin>`.

Antes de comenzar con la salida gráfica debemos crear un programa que nos suministre el último dato medido.

Estaría bien si guardaras ese dato en un subdirectorio separado; también tienes que hacer tu programa ejecutable: `chmod 755 nombredelprograma`.

Necesitamos limitar la salida al último dato e introducirlo en un programaPerl-CGI. Esto será hecho con la consulta SQL.

```
#!/usr/bin/perl

use DBI;
use strict;

# Inicializar la base de datos
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;

# parámetros de trabajo de la base de datos
my $sql;
my $sth;

# Parámetros de trabajo de los sensores
my $temp;
my $zeit;

# Preparar la salida HTML
print "Content-type: text/html\n\n";

# Salida de las medidas de los sensores independientemente
$sql = "select messung, zeit from sensor$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();

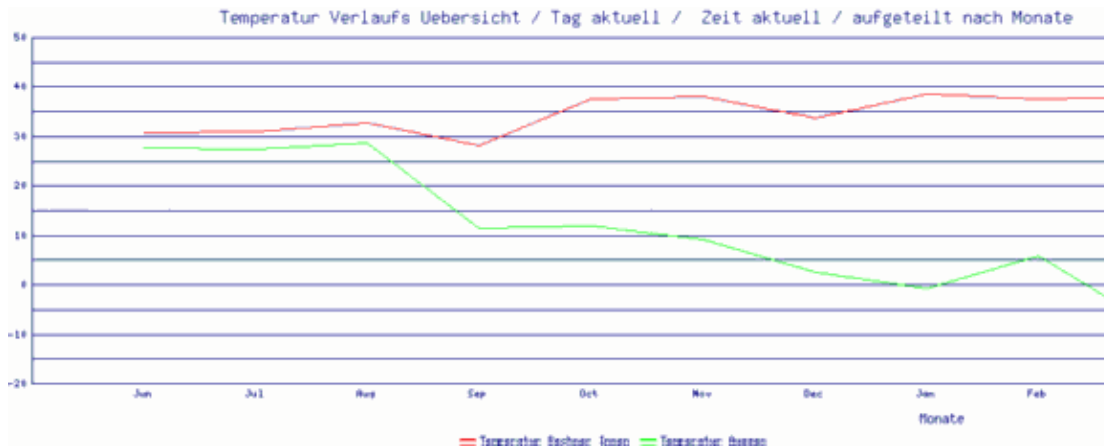
print "<p>Temperatur Sensor$i: <b>[$temp]</b>
$zeit</p>";

}

# Cerramos base de datos
$db->disconnect;
```

Este no es el ejemplo más elegante, debería sólo mostrar qué fácil es realizar esta tarea con Perl.

## Salida Gráfica



Ahora trasteemos con la salida gráfica. El programa (descárgalo al final del artículo) genera curvas gráficas, para otros gráficos mira los otros módulos de GD.

El programa está utilizando el módulo CGI que activa la salida HTML con Perl. Aquí doy referencias a las numerosas descripciones que hay de esto en Internet.

Volvamos al programa. Abarca una parte principal y dos subprogramas. Una parte es responsable de la consulta SQL, la segunda es responsable de los gráficos.

La parte principal sólo realiza tres consultas y los datos se pasan a los subprogramas.

1. Generando la escala del eje de las X
2. Datos del primer sensor (sensor0)
3. Datos del segundo sensor (sensor1)

Solamente las consultas necesitan variar para generar distinta salida gráfica.

## Consultas SQL

Finalmente quiero mostraros algunas consultas SQL debido a que son el principal asunto de este ejemplo.

### Las Últimas Cinco Medidas

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

### El día más frío del año

```
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
```

## El día más caluroso del año

```
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
```

## Calculando la media (promedio) del día

```
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)
```

## Conclusión

Siempre me sorprende de qué fácil es escribir programas en Perl. Realmente no han sido escritos, pero han sido copiados y las secciones combinadas; de alguna manera todo existe en alguna parte.

Espero que haya podido proporcionaros un pequeño vistazo a los tópicos de Perl, CGI y MySQL.

## Descarga

- [Programa CGI "toma los valores actuales"](#)  
( Programa CGI "Salida De las Medidas Actuales" )
- [Programa CGI "representación gráfica, relativa l día/hora actual"](#)  
( Programa CGI " Presentación Gráfica Para EL Día Actual y la Hora Seleccionada" )

## Enlaces / Referencias

- [Homepage CPAN](#)
- [Artículo Monitoreo de Temperatura con Linux](#)
- [Descripción de la instalación del módulo CPAN// or](#)

- [Referenz \(de\) mysql](#)
- [Homepage Perl](#)

---

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © <u>Stefan Blechschmidt</u> "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Translation information: de --&gt; -- : Stefan Blechschmidt &lt;sb(at)sbsbavaria.de&gt; de --&gt; en: Jürgen Pohl &lt;sept.sapins(Q)verizon.net&gt; en --&gt; es: José Manuel González Calvar &lt;jmcalvar(at)speedymail.org&gt;</p>
--	---

2005-01-10, generated by lfparsr\_pdf version 2.51