

MySQL und Perl: eine Zweckehe



by Georges Tarbouriech
<georges.tat@linuxfocus.org>

About the author:

Georges benutzt Unix schon seit langem. Er mag jene Produkte, die zur Verbreitung von Lösungen mit freier Software im professionellen Bereich geführt haben.

Abstract:

MySQL und Perl gibt es nun schon recht lange. Sie sind immer noch weit verbreitet, auch wenn sich die "Mode" ändert. In diesem Artikel geht es um diese zwei Produkte, die als Ganzes zusammen arbeiten, entweder im Internet oder in Ihrem lokalen Netzwerk. Das gegebene Beispiel betrifft Unixsysteme, freie und kommerzielle, auch wenn es auf andere weit verbreitete "Betriebssysteme" angewendet werden kann.

Was dieser Artikel ist: eine kleine Übersicht, was man mit diesem Paar machen kann, mit einer Vorstellung der Benutzerfreundlichkeit, Geschwindigkeit, Zuverlässigkeit, Sicherheit...

Was dieser Artikel nicht ist: weder ein MySQL-Tutorium, noch ein Perl-Tutorium; weder eine MySQL-Übersicht, noch eine Perl-Übersicht. Folglich werden wir MySQL in Verbindung mit Perl betrachten, ohne zu vergessen, daß gilt: "there is more than one way to do it (viele Wege führen nach Rom)".

Was gibt es über dieses nette Paar zu sagen?

MySQL ist ein Relational DataBase Management System (relationales Datenbankmanagementsystem)(RDBMS), erhältlich bei <http://www.mysql.com>. Es ist unter der GNU GPL veröffentlicht, frei, je nachdem, wofür es benutzt wird. Überprüfen Sie die Lizenzpolitik auf der MySQL-Webseite. Es funktioniert auf vielen Plattformen entweder als Server oder als Client. Es gibt noch ein paar andere RDBMS als freie Software, aber wir werden hier keine Vergleiche anstellen, da die Wahl von MySQL für diesen Artikel willkürlich ist. Auch werden wir es nicht mit den großen "kommerziellen" Programmen wie Informix, Oracle, Sybase, usw. vergleichen. Es reicht zu sagen, daß MySQL vermutlich eines der am weitesten verbreitetsten DBMS im Internet ist. Für diesen Artikel verwenden wir die Version 3.23.36 (wiederum willkürlich). Zum Zeitpunkt des Verfassens dieses Textes ist die aktuelle stabile Version 3.23.46 und die Alphaversion ist die lang erwartete Version 4.0. Diese können als Quelltext zum Kompilieren oder als Pakete heruntergeladen werden.

Um MySQL in Verbindung mit Perl zu benutzen, braucht man etwas mehr: die DBI-Module von Perl. Man kann zumindest DBI, Mysql-Mysql-modules, Data-Dumper and Data-ShowTable runterladen.

Wir werden nicht über ihre Installation sprechen, da sie einleuchtend ist und die Pakete alle Informationen

beinhalten, die Sie brauchen.

Perl bedeutet Practical Extraction and Report Language. Am Anfang war es für Dokumentmanipulation gedacht (Syntaxanalyse, Extrahierung...), aber bald wurde es viel mehr als das. Man kann mit Perl fast alles machen, von administrativen Aufgaben bis hin zu CGI-Scripts durch richtige Applikationen und, natürlich, Datenbankschnittstellen.

Perl ist bei vielen freien und kommerziellen Unix-Distributionen dabei, wenn nicht sogar bei allen. Zum Zeitpunkt der Erstellung dieses Artikels ist die aktuelle stabile Version 5.6.1 und die Betaversion 5.7.2. In diesem Artikel werden wir die gute, alte 5.005_03 verwenden. Sollten Sie kein Perl auf Ihrer Maschine haben (wie ist das nur möglich?), können Sie es von <http://www.perl.com/> herunterladen. Perl versorgt Sie tonnenweise mit Modulen für fast alles. Sie können sie sich aus dem CPAN der Webseite runterladen: eine Goldmine!

Nicht zuletzt, welch Überraschung, um mit beiden Werkzeugen zu arbeiten: Man braucht einen Webserver! Apache scheint die richtige Wahl zu sein, da er, wie gewöhnlich, Teil vieler freier und kommerzieller Unix-Distributionen ist. Sollten Sie ihn nicht haben (wo haben sie nur Ihre Distribution her?), ist er auf <http://www.apache.org/> erhältlich.

Über das Beispiel, welches wir benutzen werden

Sie haben wahrscheinlich schon gemerkt, daß LinuxFocus ein mehrsprachiges Magazin ist. Das bedeutet, wenn Sie ein Redakteur sind, müssen Sie den Status eines neuen Artikels und somit die Übersetzung überwachen. Anders gesagt: wer tut was, wann... Zur Zeit gibt es ungefähr 200 Artikel, oft in durchschnittlich 5 Sprachen verfügbar. Das ergibt ungefähr 1000 Artikel (na, wie schlau bin ich?) und es werden immer mehr. Diese Masse muß archiviert, formatiert und zusammengefaßt werden... Wie glauben Sie, wird das bewerkstelligt? Natürlich mit Perl.

Unser Chefredakteur, Guido Socher, hat schon viele Perl-Programme geschrieben, um unser Leben erheblich leichter zu machen. Er schrieb ebenfalls ein dreiteiliges Perl-Tutorial und eine Perl-Buchkritik. Schauen Sie sich doch mal den Abschnitt mit den Verweisen am Ende dieses Artikels an!

Javi, unser spanischer Redakteur, schrieb ein Program, um den Status der Übersetzung zu verwalten... in Perl. Atif, einer unserer Star-Autoren, kommt aus dem Königreich Perl, deshalb ist seine Muttersprache Perl. Letztendlich hat er auch zu MySQL beigetragen, indem er ein Web-Administrations-Tool verbessert hat. Wieder finden Sie etwas dazu im Abschnitt mit den Verweisen.

All dies zeigt deutlich, daß, wenn Sie nach einem Perl-Paradies suchen, Sie lediglich bei LinuxFocus mitmachen müssen.

Da ich einer der französischen LinuxFocus-Redakteure bin, und ich ziemlich faul bin, habe ich mir meine eigene Datenbank angelegt, die, raten Sie mal, auf MySQL und Perl basiert.

Die Datenbank anlegen

Dies setzt voraus, daß MySQL korrekt installiert wurde, Benutzer angelegt und sie mit Passwörtern geschützt wurden. Die Installation ist außerhalb des Rahmens dieses Artikels und die großartige Dokumentation, die bei MySQL zur Verfügung steht, stellt alle Informationen zur Verfügung.

Starten Sie den MySQL-Server, indem Sie das Skript *mysql.server* verwenden, da es den Daemon *safe_mysqld* aufruft und Sie diesem Daemon Optionen übergeben können.

Starten Sie eine Verbindung, indem Sie

```
mysql -h host -u user -p
```

eingeben. Wenn der Server auf Ihrer lokalen Maschine läuft, dann brauchen Sie `-h host` nicht. Nachdem Sie das Paßwort eingegeben haben, sind Sie mit dem Server verbunden (nun, das sollten Sie sein!). Und jetzt können Sie Ihre Datenbank anlegen. Am Prompt von `mysql` geben Sie nun

```
CREATE DATABASE lf;
```

ein. Dies ist unser Beispiel (lf steht für LinuxFocus) und es ist naheliegend, daß Sie Ihrer Datenbank den Namen geben, den Sie wollen. Als nächstes erteilen Sie den zugelassenen Benutzern einige Rechte, unter der Annahme, daß Sie das Recht haben, es zu tun (das soll heißen, daß Sie als User mit administrativen Rechten eingeloggt sind). Wenn Sie wollen, daß ein Benutzer in der Lage ist, die Datenbank zu verwalten, dann können Sie ihm die Privilegien geben, indem Sie

```
GRANT ALL ON lf.* TO username;
```

eingeben. Wählen Sie die Datenbank, die Sie gerade erzeugt haben, indem Sie `USE lf` eingeben. Erzeugen Sie eine Tabelle, gemäß Ihrem Bedarf. In unserem Beispiel legen wir eine Tabelle mit dem Namen `trissue` an:

```
CREATE TABLE trissue (num INTEGER UNSIGNED, category VARCHAR(25), title VARCHAR(40), author VARCHAR(20), en VARCHAR(20), es VARCHAR(20), fr VARCHAR(20), de VARCHAR(20), nl VARCHAR(20), ru VARCHAR(20), tk VARCHAR(20), issue VARCHAR(20));
```

Lassen Sie uns prüfen, ob es wie erwartet erzeugt wurde:

```
USE lf
SHOW TABLES;
DESCRIBE trissue;
```

Das ist alles.

Und nun brauchen wir einige Daten. Um Daten in eine leere Tabelle zu laden, ist es das einfachste, eine Textdatei mit TAB-Trennern zu verwenden. Wenn Ihr Text fertig ist, schreiben Sie einfach:

```
LOAD DATA LOCAL INFILE "maindb.txt" INTO TABLE trissue;
```

Wenn Ihre Textdatei korrekt ist, dann ist die Tabelle nun gefüllt. Das können Sie mit dem Folgenden überprüfen:

```
SELECT * FROM trissue;
```

Dies sollte eine lange Liste anzeigen. Nun sind Sie in der Lage, alle Daten abzurufen, indem Sie Abfragen verwenden.

Soweit, so gut.

Bis jetzt haben wir nur MySQL verwendet und konnten alles damit machen. Was ist also der Sinn von Perl ?

Perl bei der Arbeit

Perl wird uns dabei helfen, diese Abfragen zu automatisieren und um die Ergebnisse in einem Web-Browser anzuzeigen usw. Auch hier ist vorausgesetzt, daß die Perl-Module korrekt installiert wurden, um MySQL in Verbindung mit Perl zu verwenden.

Wir werden jetzt Perl-Skripte schreiben, die wir als CGI-Skripte verwenden werden. Sie werden uns auch

ermöglichen Perl und HTML zu mischen, um die Datenbank abzufragen und die Ausgabe zu formatieren. Wir werden ein einfaches Beispiel-Skript verwenden, das uns erlaubt, alle vom gleichen Autor geschriebenen Artikel zu durchsuchen. Wir lassen die Artikelnummer, die Kategorie, den Titel, die Namen der Übersetzer für die verschiedenen Sprachen anzeigen (nur die vollständigen Projekte), die Ausgabe und wann die Artikel veröffentlicht wurden.

Sie können dieses Skript als Vorlage für Ihre eigene Anwendung benutzen, aber seien Sie sich bewußt, daß dieses Beispiel kein sehr sicheres Programm ist. Sie können eine ausführlicher kommentierte Version dort bekommen.

```
# First, we say this is a "Tainted" Perl script.

#!/usr/bin/perl -Tw
#
# This is a comment
# db consult
#
# We use the Perl DBI module
use DBI;

# As cgi :
use CGI qw(param());

print <<END_of_start;

Content-type: text/html

<html>
<title>LFAuthors main db</title>

<center><TABLE>
<TR VALIGN=TOP>
<TD><form action="/cgi-bin/lf.cgi" method="get">

# Here comes the button's title for the launching page
<input type="submit" value="          LFAuth          ">
</form>
</TD>
</TR>
</TABLE>
```

Jetzt weisen wir das Skript an, die Datenbank abzufragen:

```
<center><H2>Search by author</H2></center>

<form action="/cgi-bin/lf.cgi\" method="get\">Author name : <input
type="text\" size="30\" name="author\"><input type="submit\"
value="Search...\"></form></center>

END_of_start

if (param("author") ne '') {
    $author = param("author");

    $autsrch.=' ';
    $autsrch.=$author;
```

```

    $autsrch.='";

# We connect to the database named lf as user doe

$dbh = DBI->connect("DBI:mysql:lf","doe",'');

$sth = $dbh->prepare("
    select *
    from trissue
    where
    author = $autsrch
    ");

$sth->execute;

```

Nun lassen wir das Skript die Ergebnisse der Abfrage vorverarbeiten und anzeigen. Entweder haben wir keine Suche angegeben und es wird den Inhalt der ganzen Datenbank anzeigen oder wir haben nach einem Namen gefragt und es zeigt alle Artikel, die mit dem Namen dieses Autors verknüpft sind, an. Wenn Sie Tausende von Datensätzen in Ihrer Datenbank haben, empfehle ich, nicht den ganzen Inhalt ausgeben zu lassen!

```

print <<END_suite;

<center>
<TABLE BORDER=>
<tr bgcolor=#A1C4EE>
<th width=60 align=CENTER><font color=#000000> Num </font></th>
<th width=110 align=CENTER><font color=#000000> Category </font></th>
<th width=110 align=CENTER><font color=#000000> Title </font></th>
<th width=110 align=CENTER><font color=#000000> Author </font></th>
<th width=110 align=CENTER><font color=#000000> En </font></th>
<th width=110 align=CENTER><font color=#000000> Es </font></th>
<th width=110 align=CENTER><font color=#000000> Fr </font></th>
<th width=110 align=CENTER><font color=#000000> De </font></th>
<th width=110 align=CENTER><font color=#000000> Nl </font></th>
<th width=110 align=CENTER><font color=#000000> Ru </font></th>
<th width=110 align=CENTER><font color=#000000> Tk </font></th>
<th width=110 align=CENTER><font color=#000000> Issue </font></th>
</tr>

END_suite

while( ($num,$category,$title,$author,$en,$es,$fr,$de,$nl,$ru,$tk,$issue)
=$sth->fetchrow() ) {
print "<tr>";
print "<td width=60 bgcolor=#FFFFFFE8 align=center> $num</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $category</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $title</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $author</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $en</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $es</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $fr</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $de</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $nl</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $ru</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $tk</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $issue</td>";
print "</tr>";

}
print "</TABLE>";
print "<BR>";
print "<BR>";

```

```

        print "<br>";

    } else {

# DB Connect

        $dbh = DBI->connect("DBI:mysql:lf","doe","");

# Search

        $sth = $dbh->prepare("
            select *
            from trissue
            ");

        $sth->execute;

# Display result

print <<SUITE;

<center>
<TABLE BORDER=>
<tr bgcolor=#A1C4EE>
<th width=60 align=CENTER><font color=#000000> Num </font></th>
<th width=110 align=CENTER><font color=#000000> Category </font></th>
<th width=110 align=CENTER><font color=#000000> Title </font></th>
<th width=110 align=CENTER><font color=#000000> Author </font></th>
<th width=110 align=CENTER><font color=#000000> En </font></th>
<th width=110 align=CENTER><font color=#000000> Es </font></th>
<th width=110 align=CENTER><font color=#000000> Fr </font></th>
<th width=110 align=CENTER><font color=#000000> De </font></th>
<th width=110 align=CENTER><font color=#000000> Nl </font></th>
<th width=110 align=CENTER><font color=#000000> Ru </font></th>
<th width=110 align=CENTER><font color=#000000> Tk </font></th>
<th width=110 align=CENTER><font color=#000000> Issue </font></th>
</tr>

SUITE

while( ($num,$category,$title,$author,$en,$es,$fr,$de,$nl,$ru,$tk,$issue)
=$sth->fetchrow() ) {
print "<tr>";
print "<td width=60 bgcolor=#FFFFFFE8 align=center> $num</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $category</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $title</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $author</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $en</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $es</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $fr</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $de</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $nl</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $ru</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $tk</td>";
print "<td width=110 bgcolor=#FFFFFFE8 align=left> $issue</td>";
print "</tr>";

    }
print "</TABLE>";
print "<BR>";

```

```

}
print end_html;
$sth->finish;

# Disconnect

$dbh->disconnect;

exit;

```

Hier ist die Ausgabe der Abfrage in einem Web-Browser:

Num	Category	Title	Author	En	Es	Fr	De	It	Ru	Th	Issn
53	UNIX Basics	Regular Expressions	G.Socher	G.Socher	F.Gleeno	J.D.Gilard	G.Socher	F.Lambrechts	Kiv	unk	July1996
64	UNIX Basics	Finding Files	G.Socher	G.Socher	R.Soffio	J.Per	G.Socher	unk	unk	Tuban	September1999
77	UNIX Basics	File Access Permissions	G.Socher	G.Socher	M.Mariano	J.Per	G.Socher	E.Milthagen	Aus	unk	January1999
114	Software Development	Part I	G.Socher	G.Socher	O.Abeiera	V.Hirou	K.Socher	F.Lambrechts	unk	unk	September1999
126	Software Development	Part II	G.Socher	G.Socher	H.Rodriguez	G.Tarbourieck	K.Socher	F.Lambrechts	nt	G.Aky	November1999
134	System Administration	Home Networking	G.Socher	G.Socher	V.Rossetti	J.Peyratout	H.Radke	T.Ujbert	nt	nt	January2000
138	Software Development	Part part II	G.Socher	G.Socher	J.R.Garcia	G.Tarbourieck	K.Socher	F.Lambrechts	nt	S.Dinc	January2000
140	Software Development	SNIFF+ for Linux	G.Socher	G.Socher	J.Pascucci	E.Cappau	nt	M.Reiners	nt	P.Hanulu	March2000
151	System Administration	Getting up IP-Masquerading	G.Socher	G.Socher	P.Vega	J.Peyratout	M.Bank	T.Ujbert	K.Poukhakov	K.Orluk	May2000
165	Hardware	Using Serial Line LCD	G.Socher	G.Socher	J.Nieto	J.Per	J.Link	nt	K.Poukhakov	Z.Gul	July2000
186	Hardware	A serial line shutdown button	G.Socher	G.Socher	nt	J.Peyratout	G.Socher	T.Ujbert	K.Poukhakov	H.Kaya	January2001
192	System Administration	Using different ISPs	G.Socher	G.Socher	R.Velasco	J.Peyratout	G.Socher	nt	K.Poukhakov	G.Bakcu	March2001
208	Software Development	Backup/review Professional Perl Program	G.Socher	G.Socher	A.Pardo	P.Lecaste	P.Thalman	A.van Eijckenburg	K.Poukhakov	E.Mutu	July2001
215	System Administration	E-mail over UUCP	G.Socher	G.Socher	R.Hemasse	G.Tarbourieck	G.Stein	T.Ujbert	E.Saenik	E.Mutu	September2001
220	Hardware	Altek Super Mini Optical Mouse	G.Socher	G.Socher	nt	P.Trbsch	G.Socher	nt	nt	E.Mutu	November2001
222	UNIX Basics	Running applications remote with X11	G.Socher	G.Socher	nt	nt	nt	nt	nt	nt	np

Nun sind wir fertig!

Der Sicherheitsaspekt

Es ist klar, daß Sie das Ganze sichern müssen, wenn Sie auf Ihrer Webseite eine Datenbank anbieten wollen. Natürlich werden wir keine schrittweise Anleitung geben, wie man eine Webseite oder einen Datenbank-Server sichert. Wie auch immer, es ist wichtig auf Grundlegendem zu bestehen.

Um es kurz zu machen: wenn Sie Dienste im WWW anbieten, ist das Erste was Sie sichern müssen, der Web-Server. Dies ist weit außerhalb des Rahmens dieses Artikels. Wenn Sie mehr über dieses Thema lernen wollen, gibt es dazu Unmengen an verfügbarer Dokumentation. Ein schöner Start ist das [Linux Documentation Project](#).

Der nächste Schritt betrifft den Datenbank-Server. Wenn Sie ein Werkzeug wie MySQL installieren, vergessen Sie nicht den Sicherheitsteil des Manuals zu lesen. Nochmals gilt die grundlegende Sorge den

Benutzerpaßwörtern: Lassen Sie niemals ein Benutzerkonto ohne Paßwort, insbesondere das root-Account der Datenbank (welches ein anderes als das des root-Account des Rechners sein sollte). Der andere wichtige Punkt betrifft die Rechte: geben Sie nicht jedem alle Rechte. Es scheint offensichtlich... und deshalb vergessen es viele Leute einfach!

Um noch weiter zu gehen: warum nicht die Datenbank in eine Chroot-Umgebung verschieben? Lesen Sie Mark's article "Chrooting all services" über dieses Thema. Er schreibt über eine andere Datenbank, aber was er schreibt, kann auf MySQL angewendet werden.

Eine andere Sicherheitsmaßnahme betrifft die Datenzirkulation. Es ist kein schlechter Gedanke, die Daten durch einen Tunnel zu senden und zu empfangen. Sie können den Artikel Durch den Tunnel für weitere Hinweise durchlesen.

Nicht zuletzt ist sicheres Programmieren einer der Grundgedanken. Perl ist eine großartige Sprache, aber es ist recht einfach, damit Programmierfehler zu machen. Ein weiterer LinuxFocus-Artikel kann Ihnen beibringen, was zu tun ist, besonders bei Perl. Schauen Sie sich mal hier um. Es ist der letzte Artikel der "Sicheres Programmieren"-Serie und betrifft besonders CGI-Skripte. Ein Muß!

Natürlich setzt dies voraus, daß Sie schon ein gesichertes System ohne bekannte Sicherheitslöcher haben, mit den letzten Patches und vielen nötigen Sicherheitswerkzeugen wie NIDS (Network Intrusion Detection System), wie Snort (auf <http://www.snort.org/>), Firewall, Port- und Sicherheitsscanner (nmap, nessus), etc. Wenn Sie es sich leisten können, dann können Sie sich auch für jeden angebotenen Dienst einen anderen Server einrichten: ein Web-Server, ein Datenbank-Server... und deren Mirror für Hochverfügbarkeit. Und so weiter! Sie werden niemals damit fertig, da Sicherheit niemals erreicht wird. Versuchen Sie einfach die Risiken zu reduzieren... und sie werden täglich immer schlimmer. Ich habe Sie gewarnt.

Was noch?

Da gilt: "There Is More Than One Way to Do It (viele Wege führen nach Rom)", können Sie es sich selbst aussuchen, wie Sie die Sache verwirklichen. Es gibt viele RDBMS und viele Sprachen, um mit ihnen zu kommunizieren. Der Gedanke hinter diesem Artikel war, zu zeigen, wie gut MySQL und Perl arbeiten, wenn man sie zusammen nutzt.

Natürlich war die Wahl sehr subjektiv: Ich liebe MySQL, weil es eher klein ist, unter vielen Betriebssystemen läuft, schnell und zuverlässig ist... Ausserdem schätze ich die Arbeit des MySQL- Teams sehr, ohne die zahlreichen Helfer zu vergessen. Und was mir am meisten gefällt: diese Leute haben nicht versucht, das Rad neu zu erfinden. Sie liessen den Dingen ihre Einfachheit.

Bezüglich Perl wurde schon alles gesagt: Was könnte ich noch hinzufügen? Ich glaube, Sie können nicht ohne arbeiten, ob Sie nun ein Netzwerkadministrator sind oder ein Entwickler oder was auch immer. Die Perl-Community ist eine der größten Quellen von gemeinschaftlich geteiltem Wissen. Es ist ein Magazin erhältlich, welches sich Perl Journal nennt, welches inzwischen jeder zweiten Ausgabe des SysAdmin Magazins beiliegt. Wenn Sie es abonnieren möchten, dann gehen Sie auf <http://www.samag.com/>.

Da wir gerade über großartige Arbeit sprechen: hier kommt der übliche Abschnitt mit dem Off-Topic-Thema. Sie, die LinuxFocus-Leser, haben wahrscheinlich die kleine Anzahl von beteiligten Leuten dieses Magazins noch nicht wahrgenommen. Trotzdem können Sie es in vielen verschiedenen Sprachen lesen. Haben Sie jemals bemerkt, daß einige der Teams fast immer aus einer oder zwei Personen besteht, die die ganze Arbeit machen? Sie sind Webmaster, Übersetzer usw. Schauen Sie beim russischen oder türkischen Team nach: Sie werden sehen, daß die meisten der Artikel von Kyril oder Erdal übersetzt werden. Schauen Sie bei den Projekten, die am Entwickeln sind, wie Arabisch oder Portugiesisch: gleiches Ergebnis! Ich möchte ihnen allen zu dieser großartigen Arbeit gratulieren. Danke an Euch alle: Die Free Software Community verdankt Euch viel.

Entschuldigung für diese Abschweifung, aber ich glaube, das mußte gesagt werden.

Zurück zum Thema: Lassen Sie mich mit ein paar Worten über Freie Software abschliessen. Die Leute bei MySQL oder Perl verdienen ein herzliches Dankeschön. Sie versorgen Sie mit großartigen Werkzeugen, meist kostenlos. Dennoch sind diese Werkzeuge oft so gut wie echte kommerzielle Produkte (wenn nicht sogar

besser), sie werden regelmäßig verbessert, sehr gut dokumentiert und Sie können sie auf fast jedem Unix verwenden. Können Sie woanders etwas Vergleichbares finden? Ich fürchte nicht!
Dieser Artikel wird Ihnen wahrscheinlich nicht viel beibringen, wenn es Sie jedoch dazu anregt, diese Produkte auszuprobieren, ist er nicht nutzlos.
Leben wir nicht in einer großartigen Zeit?

Referenzen

[Perl mongers](#)

Guidos Perl-Tutorium:

[Perl I](#)

[Perl II](#)

[Perl III](#)

Buchkritik von "Professional Perl Programming":

[Perl Programming](#)

[Atifs](#) Beitrag zu MySQL.

Eine Übersicht von MySQL in LinuxFocus; ein alter, aber immer noch aktueller Artikel:

[MySQL](#)

Ein altes LinuxFocus SQL-Tutorium in zwei Teilen:

[SQL Teil I](#)

[SQL Teil II](#)

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Georges Tarbouriech "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Georges Tarbouriech <georges.tat/linuxfocus.org> en --> de: Hubert Kaißer et al <hubert.kaisser/at/botb.de></p>
---	---