

Installation eines LAMP-Systems



by Sascha Blum ([homepage](#))

About the author:

Ich nutze Linux sehr gerne, weil es sehr leistungsfähig und stabil ist und es dem Nutzer vor allem sehr viele Möglichkeiten bietet. Am allerbesten finde ich, dass es offen für alle (OpenSource) ist und dass sich jeder an der Entwicklung beteiligen kann.



Abstract:

Ich möchte Ihnen anhand dieses Tutorials einmal erklären, wie man einen Linux-Server mit so gut wie allen nützlichen Features installiert. Kurz gesagt, wir setzen ein LAMP-System auf.

Vorab möchte ich Ihnen aber erklären, für was die Abkürzung LAMP steht. LAMP heißt nichts anderes als Linux Apache MySQL PHP. Mit anderen Worten setzt sich das so genannte LAMP-System aus einem Linux-Betriebssystem, einem Apache Webserver, einer MySQL-Datenbank und der Scriptsprache PHP zusammen.

Vorwort

Dieses Tutorial erklärt, wie man ein LAMP (Linux Apache MySQL PHP) unter Verwendung von Dynamic Shared Objects (DSO) installiert.

DSOs haben einen ganz großen Vorteil gegenüber der statischen Installation:

Jedes einzelne Modul kann jederzeit und ohne großen Aufwand durch eine neuere Version ersetzt werden, ohne dass alle anderen Module neu kompiliert und installiert werden müssen. Dabei spielt es keine Rolle, ob es sich hierbei um das PDF-Lib-Modul, das GD-Lib-Modul oder ein anderes Modul handelt. Bei einer statischen Installation müsste man für den Fall eines Updates von PHP 4.2.3 auf z.B. PHP 4.2.4 alles neu kompilieren und installieren. Mit anderen Worten, man müsste den Apache-Server, die GD-Lib, die PDF-Lib, all die anderen Module, die man braucht, und selbstverständlich auch PHP neu kompilieren. Bei einer DSO-Installation hingegen muss man letztendlich NUR noch PHP neu kompilieren und installieren, alles andere bleibt unverändert und unberührt.

Hinweis: Es ist generell empfehlenswert die jeweiligen README-Dateien, welche bei den einzelnen Paketen beiliegen, vor einer neuen Installation bzw. schon vor jedem Kompilieren, gründlich durchzulesen, denn jede Installation kann anders verlaufen. Es gibt immer wieder Schalter, die beim Konfigurieren mittels `./configure` angegeben werden müssen/können, die über den erfolgreichen Verlauf der Installation letztendlich entscheiden. Es sollte, soweit ich es getestet habe, auf dem besagten

System auf Antriebe funktionieren. Bei Fehlern unbedingt die README zu Rate ziehen. Die Installation selbst muss mit Root-Zugriffsrechten erfolgen!

Nun aber genug der Hinweise. Beginnen wir jetzt mit der Installation unseres LAMP-Systems.

Vor der Installation sollte dieses Tutorial komplett und gründlich durchgelesen werden!

Download-Quellen bzw. was benötigt wird

Dazu gehören die folgenden Pakete, die vor der Installation besorgt werden müssen:

- Apache 1.3.27
(<http://www.apache.org/>)
Direkter Download:
http://www.apache.org/dist/httpd/apache_1.3.27.tar.gz (2,2 MB)
- MySQL
(<http://www.mysql.org/>)
RedHat packages (rpm):
MySQL 3.23.52 Server (i386) (7.4M)
MySQL 3.23.52 Client programs (i386) (2.2M)
MySQL 3.23.52 Libraries and Header files for development (i386) (743K)
MySQL 3.23.52 Client shared libraries (i386) (232K)
- zlib 1.1.4
(<http://www.gzip.org/zlib/>)
Download:
<ftp://ftp.info-zip.org/pub/infozip/zlib/zlib-1.1.4.tar.gz> (177 KB)
- GD-Library 1.8.4
(<http://www.boutell.com/gd/>)
Download:
<http://www.boutell.com/gd/http/gd-1.8.4.tar.gz> (252 KB)
Hinweis: Aus lizenzrechtlichen Gründen unterstützt die GD-Library seit geraumer Zeit kein GIF-Format mehr!
- PDF-Lib 4.0.3
(<http://www.pdflib.com/pdflib/index.html>)
Download:
<http://www.pdflib.com/pdflib/download/pdflib-4.0.3-Linux.tar.gz> (3,2 MB)
- PHP 4.2.3
(<http://www.php.net/>)
Download:
http://us3.php.net/do_download.php?download_file=php-4.2.3.tar.gz (3,3 MB)

Installation

Sobald Sie alle oben genannten Pakete heruntergeladen haben, kann es losgehen. Als erstes müssen alle Dateien in das Verzeichnis (Verzeichnis lamp muss zuvor durch ein „mkdir

/usr/local/src/lamp" erstellt werden, soweit es noch nicht vorhanden ist!):

/usr/local/src/lamp/

kopiert werden. Die RPM-Dateien von MySQL müssen nicht dahin kopiert werden, diese können auch sofort wie gewohnt installiert werden. Machen Sie es am besten als erstes. Wie Sie dazu vorgehen müssen, erfahren Sie hier im Tutorial im Abschnitt "MySQL 3.23.52";.

In dem Verzeichnis /usr/local/src/lamp/ müssen anschließend alle 6 Pakete als tar.gz liegen. Diese müssen Sie nun entpacken. Das kann wie folgt gemacht werden:

Hinweis: Die einzugebenden Befehle sind fett geschrieben, die Ausgaben vom PC sind in Normalschrift. Einzugeben ist alles nach dem >-Zeichen!

Als erstes eine Text-Konsole (Shell-Terminal, z. B. Bash) öffnen, anschließend folgende Anweisungen ausführen:

```
benutzer:~ > su
[Geben Sie nun Ihr root-Passwort ein]
root: ~> cd /usr/local/src/lamp
root:/usr/local/src/lamp > tar -xvzf apache_1.3.27.tar.gz
root:/usr/local/src/lamp > tar -xvzf zlib-1.1.4.tar.gz
root:/usr/local/src/lamp > tar -xvzf libpng-1.2.2.tar.gz
root:/usr/local/src/lamp > tar -xvzf gd-1.8.4.tar.gz
root:/usr/local/src/lamp > tar -xvzf pdflib-4.0.3.tar.gz
root:/usr/local/src/lamp > tar -xvzf php-4.2.3.tar.gz
```

Nachdem Sie nun alle Pakete entpackt haben, können Sie den Befehl "ls -l" eintippen und sehen daraufhin deren Verzeichnisse.

Ab hier ist es sehr wichtig, dass die nachfolgende Installationsreihenfolge genau eingehalten wird. Der Grund hierfür ist, dass einige Pakete die anderen Pakete benötigen, um zu funktionieren. So braucht die GD-Library z.B. die zlib und die libpng, wobei die libpng auch wiederum die zlib benötigt. Zuerst machen wir nun den Apache Webserver fertig, und zwar gehen wir wie folgt vor:

Apache 1.3.27

Hinweis: Unbedingt die README-Dateien lesen! Es gibt bei ./configure eine ganze Menge Schalter, die angegeben werden können.

Der Apache Webserver darf niemals mit der Option --enable-module=all kompiliert werden! Falls es doch einmal passiert, hätte dies zur Folge, dass nachher gar nichts mehr funktioniert. Am sinnvollsten ist es, immer so wenig Module wie nötig anzugeben. In der Regel reicht das so für den DSO-Support auch vollkommen aus! Alle anderen Module kann man ja dann selbst nach Belieben einbauen, was wiederum ein Vorteil der DSO-Installation ist.

Zur Installation und Konfiguration vom Apachen bitte wie folgt vorgehen:

Zuerst wieder auf eine Text-Konsole (Shell-Terminal z. B. Bash) wechseln.

Hinweis: benutzer:/usr/local/src/lamp > nicht mit eintippen! Das ist der sogenannte Prompt von Linux;

damit zeigt Linux, dass es auf eine Eingabe wartet! Das Aussehen des Prompts kann bei Ihnen anders aussehen, da es individuell konfigurierbar ist.

```
benutzer:/usr/local/src/lamp > cd apache_1.3.27
benutzer:/usr/local/src/lamp/apache_1.3.27 > su
[Geben Sie Ihr root-Passwort ein]
root:/usr/local/src/lamp/apache_1.3.27 > ./configure --prefix=/usr/local/apache/1.3.27
--datadir=/web/htdocs --enable-rule=SHARED_CORE --enable-module=so
```

Hinweis: Das Vorangegangene als eine Zeile eingeben! Vor dem -- ist generell ein Leerzeichen dazwischen! Hinter --datadir steht das Dokumentenverzeichnis, in das Ihre späteren Webseiten abgelegt werden. Sie können natürlich ein eigenes Dokumentenverzeichnis wählen. Falls Sie als Dokumentenverzeichnis ein anderes Verzeichnis angeben als "/web/htdocs", so beachten Sie bitte, dass Sie weiter unten in diesem Tutorial die entsprechenden Stellen auf Ihre Angaben hin abändern müssen!

```
root:/usr/local/src/lamp/apache_1.3.27 > make
root:/usr/local/src/lamp/apache_1.3.27 > make install
```

Haben Sie nun alles korrekt eingetippt, dann sollte jetzt der Apache Webserver komplett fertig kompiliert und installiert sein.

MySQL 3.23.52

Wurde, wenn die Anweisung weiter oben befolgt wurde, bereits installiert.

Sicherheits-Hinweis: Sollte Ihr Server an einem öffentlichen Netz, also Intranet oder am Internet angebunden sein, dann vergessen Sie auf keinen Fall, dem root-User des MySQL-Servers ein möglichst kompliziertes Passwort zu geben!

```
root:/home/user/download/mysql > rpm -Uvh MySQL-3.23.52-1.i386.rpm
root:/home/user/download/mysql > rpm -Uvh MySQL-client-3.23.52-1.i386.rpm
root:/home/user/download/mysql > rpm -Uvh MySQL-devel-3.23.52-1.i386.rpm
root:/home/user/download/mysql > rpm -Uvh MySQL-shared-3.23.52-1.i386.rpm
```

Hinweis: /home/user/download/mysql ist zu ersetzen durch das Verzeichnis, in welchem die entsprechenden RPM-Files liegen!

zlib 1.1.4

```
root:/usr/local/src/lamp/apache_1.3.27 > cd /zlib-1.1.4/
root:/usr/local/src/lamp/zlib-1.1.4 > ./configure --shared
root:/usr/local/src/lamp/zlib-1.1.4 > make
root:/usr/local/src/lamp/zlib-1.1.4 > make install
```

Anmerkung: Mit dem Schalter `--shared` zeigen wir `zlib` an, dass wir die Library als dynamisches Modul in PHP einbauen wollen.

libpng 1.2.3

Bei `libpng` läuft die Installation etwas anders ab als sonst. Zuerst bitte in das `/libpng-1.2.3/scripts/`-Verzeichnis wechseln:

```
root:/usr/local/src/lamp/zlib-1.1.4 > cd ../libpng-1.2.3/scripts/
```

Anschließend bitte folgende Anweisungen eingeben:

```
root:/usr/local/src/lamp/libpng-1.2.3/scripts > cp makefile.linux ../makefile  
root:/usr/local/src/lamp/libpng-1.2.3/scripts > cd ..
```

Mit diesen Anweisungen haben wir soeben das entsprechende Makefile in das `libpng`-Stammverzeichnis kopiert. Jetzt sollten wir uns aber einmal das Makefile anschauen, um eventuelle Änderungen, die das System benötigt, zu machen, wie z.B. spezielle `include`-Verzeichnisse, usw. ...

Normalerweise stimmen dort aber alle Angaben, jedoch sollte man sie sich kurz ansehen, um so schneller Fehler zu finden.

Zum Fortfahren bitte folgende Anweisung eingeben:

```
root:/usr/local/src/lamp/libpng-1.2.3 > make test
```

Falls hierbei keine Fehlermeldungen kommen, kann anschließend die `libpng` mit dem folgenden Befehl installiert werden:

```
root:/usr/local/src/lamp/libpng-1.2.3 > make install
```

gd-1.8.4

Zuerst bitte in das Verzeichnis `gd-1.8.4` wechseln:

```
root:/usr/local/src/lamp/libpng-1.2.3 > cd ../gd-1.8.4/
```

Auch hier bitte erst die Datei `Makefile` ansehen! Falls dort Änderungen nötig sind, weil sich etwas an Ihrem System geändert hat, so muss dieses jetzt geändert werden. Das Anschauen und

gegebenenfalls Editieren kann mit jedem beliebigen Text-Editor gemacht werden.

Im Normalfall kann die Makefile-Datei aber so belassen werden.

Wenn Sie nun der Meinung sind, dass in der Makefile so alles okay ist, dann geben Sie nun bitte folgende Befehle ein:

```
root:/usr/local/src/lamp/gd-1.8.4 > make  
root:/usr/local/src/lamp/gd-1.8.4 > make install
```

Sollten hierbei irgendwelche Fehler auftauchen, dann führen Sie bitte gleich Folgendes aus:

```
root:/usr/local/src/lamp/gd-1.8.4 > make clean
```

Letzteren Befehl aber nur, wenn Fehler auftauchen! Wenn das make clean ausgeführt wurde, muss die Makefile noch einmal kontrolliert und angepasst werden. Danach wieder von vorne mit make beginnen.

Hinweis: Insbesondere die Einstellungen für INCLUDEDIRS und LIBDIRS kontrollieren!

PDF-Lib 4.0.3

Hier ist es etwas einfacher, da hier das Modul bereits kompiliert vorliegt und nur noch in das Verzeichnis /usr/local/lib kopiert werden muß.

Hierbei ist wie folgt vorzugehen:

```
root:/usr/local/src/lamp/gd-1.8.4 > cd /  
root:/ > cp /usr/local/src/lamp/pdflib-4.0.3-Linux/bind/php/php-4.2.1/libpdf_php.so /usr/  
local/lib/libpdf_php.so
```

PHP 4.2.3

Als Letztes müssen wir uns noch PHP vornehmen und zwar wie folgt:

In das PHP-Verzeichnis wechseln:

```
root:/ > cd /usr/local/src/lamp/php-4.2.3/  
root:/usr/local/src/lamp/php-4.2.3 > ./configure --with-apxs=/usr/local/apache/1.3.27/bin/apxs  
--enable-track-vars --enable-ftp --with-zlib --with-gd --with-sockets --enable-sockets  
--with-sysvshm --with-sysvsem --disable-debug --with-pdfdir=/usr/local/lib  
--with-tiff-dir=/usr/local/lib --with-jpeg-dir=/usr/local/lib --with-png-dir=/usr/local/lib  
--with-zlib-dir=/usr/local/lib --with-mysql --with-xml
```

Hinweis: Das Obere als eine Zeile eingeben! Vor dem -- ist generell ein Leerzeichen dazwischen! In der zweiten bzw. dritten Zeile ist kein Fehler (wegen "sysvshm" und "sysvsem").

Als Nächstes folgt wie gehabt:

```
root:/usr/local/src/lamp/php-4.2.3 > make
root:/usr/local/src/lamp/php-4.2.3 > make install
```

Hinweis: Das Kompilieren (make) bei PHP kann bei langsamen PC-Systemen etwas länger dauern! Also nicht ungeduldig werden, auch wenn sich anscheinend über längere Zeit nichts tut! Das Verzeichnis /usr/local/src/lamp kann per "rm -r /usr/local/src/lamp" gelöscht werden (als root). Man sollte sich hierbei nicht vertippen, denn wer als root ein "rm -r /" ausgeführt hat, zerstört das gesamte System! Allerdings sollte man eines wissen, wer das Verzeichnis "/usr/local/src/lamp" komplett löscht, hat es im Falle einer Deinstallation bzw. eines Updates etwas komplizierter bzw. aufwendiger. Von daher sollten nur die gepackten Quellpakete ".tar.gz" gelöscht und die Verzeichnisse mit den Quellen erhalten werden!

Konfiguration

httpd.conf

So, die Installation wäre schon mal abgeschlossen. Jetzt kommt allerdings noch die Konfiguration.

Als erstes müssen wir nun dem Apache-Webserver mitteilen, was er mit den *.php- oder *.php3-Dateien anfangen soll.

Dieses machen wir nun wie folgt: Sie müssen nun in das "conf"-Verzeichnis von Apache wechseln:

```
root:/usr/local/src/lamp/php-4.2.3 > cd /usr/local/apache/1.3.27/conf
root:/usr/local/apache/1.3.27/conf >
```

Anschließend müssen Sie die Datei "httpd.conf" in einem Text-Editor zum Editieren öffnen und hinterher speichern.

Anmerkung: Zum Bearbeiten der Config-Datei eignet sich der Editor "Kate" ganz gut. Allerdings muß KDE im Hintergrund laufen! Gestartet wird dieser mittels Alt + F2 -Taste => kdesu kate. Mit der Tastenkombination Strg + G können Sie dann einfach in die gewünschte Zeile springen!

Sie finden dort, um die Zeile 190 rum, folgenden Text:

```
#
#Dynamic Shared Object (DSO) Support
#
```

```
#To be able to use the functionality of a module which was built as a DSO you
#have to place corresponding `LoadModule' lines at this location so the
#directives contained in it are actually available _before_ they are used.
#Please read the file README.DSO in the Apache 1.3 distribution for more
#details about the DSO mechanism and run `httpd -l' for the list of already
# built-in (statically linked and thus always available) modules in your httpd
#binary.
```

An dieser Stelle müssen Sie nun Folgendes einfügen, soweit es da noch nicht stehen sollte:

```
LoadModule php4_module libexec/libphp4.so
```

Ca. ab Zeile 770 finden Sie dann das Folgende:

```
#AddType allows you to tweak mime.types without actually editing it, or to
#make certain files to be certain types.
#AddType application/x-tar .tgz
```

An dieser Stelle bitte Nachfolgendes einfügen und/oder ergänzen:

```
AddType application/x-httpd-php .htm
AddType application/x-httpd-php .html
AddType application/x-httpd-php .phtm
AddType application/x-httpd-php .phtml
AddType application/x-httpd-php .php
AddType application/x-httpd-php .php3
AddType application/x-httpd-php .php4
AddType application/x-httpd-php-source .phps
```

Hinweis: Bitte auf genaue Schreibweise achten, ansonsten könnte es zu Fehlern kommen!

Wenn man nicht möchte, dass HTML-Dateien durch den PHP-Parser laufen, kann man auch die Zeilen:

```
AddType application/x-httpd-php .htm
AddType application/x-httpd-php .html
```

einfach weg lassen....

Damit wäre nun die httpd.conf konfiguriert.

Erklärung: Was ist ein Parser?

Ein Parser ist nichts anderes als eine Software, die Text interpretiert. Hierbei kann der Text ein Quelltext (C++) sein oder eine Dokumentenbeschreibungssprache (wie HTML). Er kontrolliert den Text auf syntaktische und semantische

Fehlerfreiheit und übergibt den „geparsten“ Text, meist in einem effizienten und kompakten internen Code, an die verarbeitende Einheit.

Text, der durch einen Parser läuft, braucht ein wenig länger, bis dieser dann angezeigt wird. Was nun bedeutet, reine HTML-Seiten werden schneller geladen und angezeigt als PHP-Seiten/-Scripts. Aber als Anwender bemerkt man kaum einen großen zeitlichen Unterschied. Es macht sich erst bemerkbar, wenn sehr viele Anwender auf ein und dasselbe zugreifen, d.h. wenn sehr viele Anwender eine PHP-Seite/-Script gleichzeitig aufrufen, kann es je nach Server-Hardware hier und da etwas länger dauern, bis diese angezeigt werden. Sprich, wer vor hat, ein LAMP-System sehr vielen Nutzern öffentlich zugänglich zu machen, z.B. Anbindung an Internet, Intranet oder Netzwerk, der sollte sich von vornherein einen leistungsfähigen und schnellen Computer zulegen, ansonsten wird es unter Umständen ziemlich lahm. Wer allerdings sein LAMP-System zum Entwickeln von PHP-Seiten/Scripten in Verbindung mit einer MySQL-Datenbank nutzen möchte, der kann dieses auch durchaus auf einem älteren PC/Notebook tun. Das Gleiche gilt auch, wer PHP-Seiten/-Scripte alleine auf einem Computersystem ausführen/anzeigen möchte.

php.ini

Jetzt müssen wir allerdings die php.ini einrichten und eventuell anpassen:

Die Datei „php.ini“ muss nun an die richtige Stelle hinkopiert werden, wo diese ordnungsgemäß auch hingehört. Hierzu muss man nun wieder in das PHP-Install-Verzeichnis wechseln:

```
root:/usr/local/apache/1.3.27/conf > cd /usr/local/src/lamp/php-4.2.3/
```

Jetzt müssen wir die Datei `php.ini-dist` in das Verzeichnis `/usr/local/lib` kopieren und zugleich diese Datei umbenennen in `php.ini`. Das machen Sie nun wie folgt:

```
root:/usr/local/src/lamp/php-4.2.3 > cp php.ini-dist /usr/local/lib/php.ini
```

Anschließend müssen Sie noch in die `php.ini` `pdflib` hineinschreiben, als `extension` (Erweiterung), damit PHP weiss, was es mit den entsprechenden PDF-Funktionen anfangen soll, wenn Sie diese einmal benötigen und damit arbeiten wollen. Die anderen Module dagegen, wie z.B. `zlib`, `GD`, usw. ... , findet PHP selbst.

Öffnen Sie nun die Datei `"/usr/local/lib/php.ini"` in einem Text-Editor. Um die Zeile 371 dieser Datei finden Sie den Abschnitt über die `extensions`.

Müsste in etwa so aussehen:

```
;Directory in which the loadable extensions (modules) reside.
```

```
....
```

```
extension_dir = ./ <= ist zu entfernen und durch Folgendes zu ersetzen:
```

```
extension_dir = /usr/local/lib  
extension=libpdf_php.so
```

Nun die Datei abspeichern.

So, wir haben es nun endlich geschafft. Sie haben nun ein komplett lauffähiges LAMP-System.

Jetzt folgt der Test, ob sich der Server auch richtig starten lässt! Hierzu muss man erstmal einen alten Server, der eventuell schon läuft (z.B. wenn bei Installation der Distribution ein Webserver bereits installiert wurde) beenden. Das erreicht man wie folgt:

```
root:/usr/local/src/lamp/php-4.2.3 > killall httpd
```

Nun müssen Sie versuchen, den neuen Server zu starten, und zwar so:

```
root:/ > /usr/local/apache/1.3.27/bin/apachectl start
```

Wenn Sie nun dort eine Nachricht sehen, die in etwa so aussieht:

```
/usr/local/apache/1.3.27/bin/apachectl start: httpd started
```

Dann ist alles in Ordnung und der Server ist nun lauffähig!

Jetzt wechseln Sie bitte in Ihr "web/htdocs"-Verzeichnis (DocumentRoot / Sollten Sie diesem einen anderen Namen gegeben haben, so ändern Sie das Nachfolgende entsprechend!) und erstellen dann dort einfach mal eine neue Datei, mit dem Namen "info.php", dies machen Sie wie folgt:

```
benutzer:/ > cd /web/htdocs/  
benutzer:/web/htdocs > touch info.php
```

Diese neue Datei "info.php" mit einem Editor öffnen und folgenden Inhalt hineinschreiben:

```
<?PHP  
echo phpinfo();  
?>
```

Hinweis: Mit allen Klammern usw., genau wie es da steht!

Nun noch schnell speichern und geöffnete Datei schließen. Jetzt kommt der spannende Augenblick! Surfen Sie nun mit Ihrem gewohnten Internet-Browser zu folgender Adresse:

```
http://127.0.0.1/info.php  
oder  
http://localhost/info.php  
oder  
http://rechnername/info.php  
oder  
http://lokale_IP/info.php
```

Wenn Sie nun die Ausgabe von phpinfo() sehen, ist alles bestens verlaufen und Sie können nun sofort mit der PHP-Programmierung loslegen! Herzlichen Glückwunsch – Sie haben nun ein vollfunktionsfähiges LAMP-System und zugleich einen Webserver.

Anmerkung: Sie können nun im Verzeichnis /web/htdocs (oder einem anderen, wenn Sie ein anderes bei der Installation angegeben haben) beliebig viele Unterverzeichnisse anlegen, z.B. Sie haben mehrere Web-Projekte, so könnten Sie z.B. pro Projekt ein Verzeichnis anlegen.

Hierbei müssen Sie noch wissen, dass /web/htdocs (oder das andere, von Ihnen gewählte) sozusagen Ihr Root-Verzeichnis ist, was Ihren Webserver angeht, darum heißt die URL auch http://127.0.0.1/info.php! Hätten Sie nun noch weitere Unterverzeichnisse, wie z.B. projekt1, projekt2 usw., dann müssten Sie diese URL entsprechend anpassen: http://127.0.0.1/projekt1/ oder http://127.0.0.1/projekt2/ usw. ...

Hinweis: PHP-Dateien werden nur geparst (angezeigt/ausgeführt), wenn diese in diesen Pfaden liegen, also entweder /web/htdocs oder deren Unterverzeichnisse!

Ihren Webserver können Sie nach Belieben ausbauen und erweitern.

Empfehlungen

Als ein allerletztes möchte ich an dieser Stelle noch ein paar Admin-Tools empfehlen, die die Arbeit mit Ihrem System/Server um einiges erleichtern können:

Webmin 1.000

[\(http://www.webmin.com/\)](http://www.webmin.com/)

Mit Webmin kann man auf einfachste Weise nicht nur das bestehende System wunderbar einrichten, sondern auch bequem Ihren Web-Server konfigurieren, z.B. können Sie dort veranlassen, dass Ihr Server bei jedem Systemstart automatisch gestartet wird; das Gleiche gilt auch für MySQL. Webmin ist sehr einfach in der Bedienung und daher für Webserver-Anfänger hervorragend geeignet!

Download:

<http://prdownloads.sourceforge.net/webadmin/webmin-1.000.tar.gz?download>

oder

<http://www.webmin.com/>

phpMyAdmin 2.3.1

[\(http://www.phpwizard.net/projects/phpMyAdmin/\)](http://www.phpwizard.net/projects/phpMyAdmin/)

phpMyAdmin ist ein wunderbares Tool für MySQL – Mit ihm kann man problemlos Tabellen anlegen, löschen, bearbeiten und vieles mehr ... – Ebenfalls sehr empfehlenswert!

Download:

(phpMyAdmin-2.3.1-php.tar.gz)

<http://www.phpmyadmin.net/index.php?dl=2>

Konfiguration per Webmin

Einstellungen für den Apache-Server:

webmin => Server => Apache Webserver

Modulkonfiguration:

Apache Server Root Verzeichnis:

/usr/local/apache/1.3.27/bin/

Pfad zur ausführbaren Datei httpd:

/usr/local/apache/1.3.27/bin/httpd

Apache version:

leeres Feld anwählen und eintragen: => **1.3.27**

Pfad zum apachectl Kommando:

ins leere Feld => **/usr/local/apache/1.3.27/bin/**

Kommando zum Starten von Apache:

ins leere Feld => **/etc/init.d/apachectl start**

Hinweis: Falls apachectl dort nicht im Verzeichnis ist, einfach noch zusätzlich hineinkopieren!

root > cp /usr/local/apache/1.3.27/bin/apachectl /etc/init.d/

Kommando zum Stoppen von Apache:
ins leere Feld => **/etc/init.d/apachectl stop**

Zeige virtuelle Server an als:
=> **Icons**

Sortiere virtuelle Server nach:
=> **Reihenfolge in Konfigurationsdatei(en)**

Maximum number of servers to display
=> **100**

Pfad zu httpd.conf
ins leere Feld => **/usr/local/apache/1.3.27/conf/httpd.conf**

Pfad zu srm.conf
ins leere Feld => **/usr/local/apache/1.3.27/conf/srm.conf**

Pfad zu access.conf
ins leere Feld => **/usr/local/apache/1.3.27/conf/access.conf**

Pfad zu mime.types
ins leere Feld => **/usr/local/apache/1.3.27/conf/mime.types**

Datei, der virtuelle Server hinzugefügt werden sollen:
=> **httpd.conf**

Test config file before applying changes?
=> **Yes**

Hinweis: Das => ist nicht einzutragen!

Möchte man, dass der Apache-Server mit System-Boot automatisch gestartet wird, so kann man das wie folgt mit Webmin einstellen:
webmin => System => Systemstart- und Herunterfahren-Dienste

Falls dort Apache noch nicht aufgeführt ist, einfach einen neuen Dienst erstellen ...

Name => **apachectl**
Script => **wird normalerweise automatisch hineingeladen ...**
Start beim Systemstart? => **Ja**
Systemstart-Kommandos => **/etc/init.d/apachectl start**
Herunterfahren-Kommandos => **/etc/init.d/apachectl stop**

Update

Update von PHP

Als erstes kommt die gepackte Datei von der neuen PHP Version in das

```
/usr/local/src/lamp/
```

Verzeichnis. Nehmen wir mal an, die nächste PHP Version trägt die Bezeichnung PHP 4.2.4, ob es diese Version einmal geben wird, kann ich allerdings nicht sagen. In diesem Kapitel soll nur verdeutlicht werden, wie man im Falle eines Updates vorzugehen hat. Folglich braucht man eigentlich nur die neue Version gegen die alte auszutauschen.

Die soeben kopierte Datei müssen Sie nun entpacken. Das kann wie folgt gemacht werden:

Als erstes eine Text-Konsole (Shell-Terminal z.B. Bash) öffnen, anschließend folgende Anweisungen ausführen:

```
benutzer:~ > su  
[Geben Sie nun Ihr root-Passwort ein]  
root:~ > cd /usr/local/src/lamp/  
root:/usr/local/src/lamp > tar -xvzf php-4.2.4.tar.gz
```

Falls Sie noch Ihr altes Source-Verzeichnis unter `„/usr/local/src/lamp“` besitzen, können Sie nun komfortabel wie folgt vorgehen:

Unser altes Source-Verzeichnis heißt: `„/usr/local/lamp/php-4.2.3“`;

Als erstes müssen wir eine Kopie von libphp4.so anlegen. Die Kopie erhält dann den Namen `„libphp4-4.2.3.so“`. Hierzu gehen wir wie folgt vor:

```
root:/ > cd /usr/local/apache/1.3.27/libexec/  
root:/usr/local/apache/1.3.27/libexec > cp libphp4.so libphp4-4.2.3.so
```

Als nächstes erstellen wir ein Backup von der alten php.ini, und zwar wie folgt:

```
root:/ > cd /  
root:/ > cp /usr/local/lib/php.ini /usr/local/lib/php-4.2.3.ini
```

Anschließend löschen wir unsere alte php.ini, da es sinnvoller ist, die neue zu verwenden.

```
root:/ > rm /usr/local/lib/php.ini
```

Da Sie ja Ihr altes PHP-Source Verzeichnis noch besitzen, haben Sie richtig Glück gehabt, denn Sie haben sich dadurch eine Menge Tipparbeit erspart.

In dem alten Verzeichnis `„/usr/local/lamp/php-4.2.3“` befindet sich nämlich ein kleines

Shell-Script. Dieses hat sich von der letzten Installation alle Parameter von `./configure` gemerkt, d.h. Sie müssen nun nicht beim `./configure` mühselig wieder alle Parameter von Hand eingeben!

Das `./configure` wird nun wie folgt ausgeführt:

```
root:/ > cp /usr/local/lamp/php-4.2.3/config.nice /usr/local/lamp/php-4.2.4/config.nice
root:/ > cd /usr/local/lamp/php-4.2.4
root:/usr/local/lamp/php-4.2.4 > ./config.nice
root:/usr/local/lamp/php-4.2.4 > make
root:/usr/local/lamp/php-4.2.4 > make install
```

Hinweis: Und alle, die das Verzeichnis `"/usr/local/lamp/php-4.2.3"` nicht mehr haben, müssen beim `./configure` von Hand die ganzen Parameter, wie im Kapitel `"Installation => PHP 4.2.3"` beschrieben, eingeben.

Nun noch die neue `php.ini` ins richtige Verzeichnis kopieren:

```
root:/ > cd /
root:/ > cp /usr/local/lib/php.ini-dist /usr/local/lib/php.ini
```

Jetzt brauchen Sie nur noch die neue `php.ini`, wie im Abschnitt `"Konfiguration => php.ini"` beschrieben ist, zu ändern und anschließend zu speichern.

Einfach den Apache neu starten und schon ist das Update abgeschlossen:

```
root:/ > /usr/local/apache/1.3.27/bin/apachectl restart
```

Schlusswort

Nachtrag

Natürlich ist kein Mensch perfekt, deshalb können in diesem Tutorial auch Fehler enthalten sein. Falls ein Experte dieses Tutorial mal lesen sollte und der Meinung ist, dass etwas falsch ist oder etwas fehlt bzw. ergänzt werden müsste, so bitte ich, mir dies mitzuteilen, damit ich dieses Tutorial verbessern kann. Das Tutorial wurde mit Sorgfalt erstellt und auf mehreren Systemen mit SuSE Linux 8.0 erfolgreich getestet. Aber dieses Tutorial sollte auch bei anderen Linux-Distributionen auf diese Art und Weise anwendbar sein.

Sicherlich gibt es noch wesentlich mehr Konfigurationsmöglichkeiten eines LAMP-Systems, aber dieses Tutorial soll im Grunde nur eine Hilfe zur Einrichtung eines Webservers für Einsteiger sein. Ich versuche, nach Möglichkeit, dieses Tutorial immer auf aktuellem Stand zu halten. Einfach immer mal wieder auf meiner Homepage vorbeischauchen, ob eine aktualisierte Version dieses Tutorials zum Download bereit steht (Anmerkung neben dem Downloadlink beachten).

Referenzen

- Unter: <http://linux.computerbraxas.de/> [Deutsch] werde ich dieses Dokument weiter pflegen
- <http://www.apache.org/>
- http://www.apache.org/dist/httpd/apache_1.3.27.tar.gz
- <http://www.mysql.org/>
- <http://www.gzip.org/zlib/>
- GD: <http://www.boutell.com/gd/>
- <http://www.pdflib.com/pdflib/index.html>
- <http://www.php.net/>
- <http://www.webmin.com/>

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Sascha Blum "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: de --> -- : Sascha Blum (homepage)</p>
---	---

2005-01-11, generated by lfparsr_pdf version 2.51