



by Sascha Blum ([homepage](#))

Come installare un sistema LAMP



Abstract:

About the author:

Mi piace molto usare Linux perchè è estremamente potente e stabile, e perchè dà all'utente molte opzioni e risorse. Oltretutto, forse l'aspetto migliore, è aperto a tutti (OpenSource) così che tutti possono essere coinvolti nel suo sviluppo.

In questa guida, voglio mostrare come installare un server Linux con tutte le caratteristiche utili già incluse. In altre parole, vi mostrerò come installare in sistema LAMP.

Ma prima di tutto vi dirò per cosa sta l'abbreviazione LAMP.

LAMP significa Linux Apache MySQL PHP. Così, come potreste intuire dal nome, un sistema LAMP consiste in un sistema operativo Linux, un Web server Apache, un database MySQL, e il linguaggio di scripting PHP.

Translated to English by:

Orla Shanaghy

<o_shanaghy(at)yahoo.com>

Introduzione

Questa guida spiega come installare un sistema LAMP usando i Dynamic Shared Objects (DSO).

I DSO hanno un grosso vantaggio sulle installazioni statiche: è possibile sostituire ogni componente individuale con una nuova versione sempre e facilmente, senza dover ricompilare e reinstallare tutti gli altri moduli. Non importa se il modulo in questione è il modulo PDF-Lib, il modulo GD-Lib o qualche altro modulo. Con un'installazione statica, se si volesse aggiornare PHP 4.2.3 a PHP 4.2.4, ad esempio, si dovrebbe ricompilare e reinsallare tutto – e per tutto intendo il serve Apache, GD-Lib, il PHP, la PDF-Lib e tutti gli altri moduli che servono (e certamente anche lo stesso PHP). Con un'installazione DSO, l'aggiornamento riguarderebbe soltanto PHP, e tutto il resto rimarrebbe com'è.

Nota: in generale, bisogna leggere attentamente il file README di ciascun pacchetto prima di installare o compilare, perchè ogni instllazione può essere diversa. Spesso, un'installazione andata a buon fine può dipender da un parametro o da un'altro che bisognava o si poteva impostare usando ./configure. Detto questo, secondo le mie prove, questa installazione dovrebbe funzionare al primo colpo. Se ottenete degli errori, consultate il README. Assicuratevi di usare i permessi di root per l'installazione!

Basta con i preamboli. Cominciamo ad installare il nostro sistema LAMP.

Leggetevi questa guida attentamente fino alla fine prima di partire con l'installazione!

Materiali da scaricare

Avrete bisogno dei seguenti pacchetti, che dovete scaricare prima di cominciare con l'installazione:

- Apache 1.3.27
(<http://www.apache.org/>)
Direct 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/httpgd-1.8.4.tar.gz> (252 KB)
Note: per ragioni di licenza, la libreria GD non supporta più il formato GIF (e non lo fa da molto tempo)!
- 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)

Installazione

Dopo aver scaricato questi pacchetti, siete pronti. Per prima cosa, copiate i files nella seguente cartella (se non l'avete già fatto, create la directory lamp con il comando `mkdir/usr/local/src/lamp`):

`/usr/local/src/lamp/`

Gli unici files che non serve copiare sono gli RPM di MySQL. Si possono installare direttamente nel solito modo. È meglio fare questo prima. Per istruzioni, vedere la sezione "MySQL 3.23.52" sotto.

Ora tutti e sei i pacchetti dovrebbero essere in /usr/local/src/lamp/ come tar.gz. Adesso bisogna decomprimerli. Procedete come mostarto in basso.

Nota: i comandi che devono essere eseguiti sono in grassetto; l'output è in carattere normale. Tutto quello da inserire è preceduto da > .

Per rima cosa aprite una console (ad es. Bash da terminale) ed eseguite i comandi seguenti:

```
user:~ > su
[Inserite la vostra password di root]
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
```

Dopo aver decompresso tutti i pacchetti, eseguite il comando "ls -l" per mostrare tutte le cartelle.

D'ora in avanti, è fondamentale seguire i punti dell'installazione esattamente nell'ordine mostrato qui. Questo perchè alcuni pacchetti hanno bisogno di altri pacchetti per funzionare. Per esempio, la libreria GD ha bisogno di zlib e libpng, e libpng ha bisogno di zlib. Adesso il serve web Apache.

Apache 1.3.27

Nota: assicuratevi di leggere il file README! Ci sono alcune opzioni su ./configure che possono essere impostate qui.

Non compilate mai Apache usando l'opzione --enable-module=all! Se lo fate, non funzionerà niente. Il miglior modo di fare è specificare meno moduli possibile. Di solito, questo è più che sufficiente per il supporto DSO. Poi potrete aggiungere tutti i moduli che volete, che è dopotutto il pregio dell'installazione DSO.

Per installare e configurare Apache procedete come indicato sotto:

Per prima cosa aprite una console (ad esempio Bash), come prima.

Nota: non scrivete user:/usr/local/src/lamp > insieme ai vostri comandi. Questo è il prompt di Linux ed è il suo modo di dirvi che sta aspettando dell'input. Il vostro prompt potrebbe essere differente, siccome può essere configurato a piacere

```
user:/usr/local/src/lamp > cd apache_1.3.27
user:/usr/local/src/lamp/apache_1.3.27 > su
[Enter your root password]
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
```

Nota: inserite quest'ultima parte come una sola riga. Di solito c'è uno spazio prima di `--`. La directory dove saranno memorizzati i vostri siti web si trova dopo `--datadir`. Naturalmente potete scegliere la vostra directory per i documenti. Però, se mettete una cartella documenti diversa da `"/web/htdocs"`, assicuratevi di cambiare in accordo gli altri percorsi importanti in seguito.

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

Se avete inserito tutto correttamente, il vostro server Web Apache dovrebbe essere completamente compilato ed installato.

MySQL 3.23.52

Se avete seguito le istruzioni precedenti in questa guida, questo è già stato installato.

Nota sulla sicurezza: se il vostro server è collegato ad una rete pubblica, es. internet o una intranet, assicuratevi di mettere una root password per MySQL più complicata possibile!

```
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
```

Nota: sostituite `/home/user/download/mysql` con la directory dove ci sono i files RPM.

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
```

Commento: usiamo lo switch `--shared` per dire a zlib che vogliamo includerla come modulo dinamico in PHP.

libpng 1.2.3

L'installazione di libpng è un po' diversa dal solito. Per prima cosa, spostatevi nella directory `/libpng-1.2.3/scripts/` :

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

Poi inserite i seguenti comandi:

```
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 ..
```

Con questi comandi, avete appena copiato il makefile giusto nella cartella principale di libpng. Adesso dovete guardare il makefile e fare tutti i cambiamenti che richiede il sistema, ad esempio "include directories" speciali. Generalmentem tutti i dati presenti nel file sono corretti, ma è meglio sempre controllare, perchè questo permette di individuare più velocemente gli errori.

Per continuare, inserite i seguenti comandi:

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

Se non ricevete alcun errore, potete installare libpng con il seguente comando:

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

gd-1.8.4

Per prima cosa, spostatevi nella directory gd-1.8.4:

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

Dovete anche qui controllare il makefile. Se qualcosa nel vostro sistema è cambiato, dovete fare i cambiamenti corrispondenti nel makefile ora. Potete vedere e modificare il file con l'editor che preferite.

Comunque, generalmente, non c'è bisogno di fare cambiamenti al makefile.

Se siete soddisfatti del makefile, inserite i seguenti comandi:

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

Se ci sono errori a questo punto, eseguite questo comando:

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

Eseguite quest'ultimo comando solo se ci sono errori! Se eseguite make clean, dovrete controllare e riadattare

il makefile un'altra volta, e rifare make.

Nota: assicuratevi di controllare le impostazioni per INCLUDEDIRS e LIBDIRS!

PDF-Lib 4.0.3

Questo è più facile, per il fatto che il modulo è già compilato e dovete solo copiarlo nella directory /usr/local/lib.

per fare ciò, eseguite:

```
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

Per ultimo dovete installare PHP.

Spostatemi nella cartella di PHP:

```
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
```

Nota: scrivete l'ultima parte come un'unica riga! Di solito c'è un carattere di spazio prima di --. Non c'è un'errore nella seconda e nella terza riga ("sysvshm" and "sysvsem").

Ora inserite i seguenti comandi, come prima:

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

Nota: compilare (make) il PHP può richiedere più tempo nei sistemi più vecchi. Perciò non spazientatevi se non sembra che stia succedendo niente per molto tempo. Potete cancellare la directory /usr/local/src/lamp (come utente root) usando "rm -r /usr/local/src/lamp". Assicuratevi di scriverlo correttamente, perchè se eseguite "rm -r /" come root, distriggerete l'intero sistema. Comunque, state attenti perchè se cancellate "/usr/local/src/lamp", sarà più laborioso disinstallare o aggiornare il sistema. Quindi, è meglio cancellare solo i pacchetti sorgente ".tar.gz" e tenere le cartelle con i sorgenti.

Configuration

httpd.conf

Quella era l'installazione. Ora la configurazione.

Prima di tutto, dovete dire ad Apache cosa deve fare con i files *.php- o *.php3.

Per fare ciò, spostatevi nella directory "conf" di Apache

```
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 >
```

Poi, aprite "httpd.conf" in un editor di testi in modo da modificarlo e salvarlo.

Nota: l'editor "Kate" è molto indicato per modificare il file config. Notare che KDE deve essere in esecuzione. Per aprirlo, premete Alt + F2 => kdesu kate. Premete Ctrl + G per andare alla riga desiderata.

Nel file, troverete questo attorno alla riga 190:

```
#
#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.
```

A questo punto, inserite il seguente testo, se non c'è già:

LoadModule php4_module libexec/libphp4.so

troverete questo attorno alla riga 770:

```
#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
```

A questo punto, aggiungete questo:

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

Nota: assicuratevi di inserire questo correttamente, altrimenti potrebbero verificarsi degli errori.

Se non volete che il parser PHP interpreti il files HTML, potete omettere le linee seguenti:

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

Ora httpd.conf è configurato.

Cos'è un parser?

Un parser è semplicemente un software che interpreta del testo. Il testo in questione può essere codice sorgente (come C++) o un linguaggio di marcatura documento (come HTML). Il parser controlla il testo per trovare errori di sintassi e di semantica, e passa il testo parsato, generalmente in un codice interno efficiente e compatto, all'applicazione che compie l'elaborazione.

Il testo che passa attraverso un parser ci impiega di più ad essere visualizzato. Questo significa che le pagine HTML sono mostrate più velocemente delle pagine PHP o degli scripts. Comunque, l'utente non nota molto ritardo. Un ritardo diviene notevole solo se alcuni utenti stanno accedendo alla stessa cosa, ad es. se alcuni utenti chiamano una pagina PHP o uno script allo stesso tempo, il server ci può impiegare di più a mostrare la pagina o lo script, in base all'hardware. Quindi, se volete rendere il vostro sistema LAMP accessibile al pubblico, ad es. connettendolo ad internet, ad una intranet o a una rete, dovete prendervi

un computer potente, altrimenti il sistema potrebbe diventare abbastanza lento. Se, s'altra parte, volete usare il vostro sistema LAMP per sviluppare pagine PHP o script insieme ad un database MySQL, potrete farlo in sicurezza usando un vecchio pc od un notebook. Lo stesso se siete il solo ad eseguire o a visualizzare pagine PHP o scripts nel vostro computer.

php.ini

Adesso dovete impostare e possibilmente configurare il file "php.ini".

Per prima cosa, dovete copiare il file "php.ini" al posto giusto. Per fare ciò, spostatevi nella directory di installazione di PHP:

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

Ora copiate il file "php.ini-dist" nella directory /usr/local/lib e rinominate il file "php.ini". fate come segue:

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

Poi scrivete "pdflib" nel file php.ini come "extension". Questo fa in modo che PHP sappia cosa fare con la corrispondente funzione PDF, se avete intenzione di usarle e di lavorare con loro. PHP trova gli altri moduli da solo (zlib, GD, etc.).

Ora aprite il file "/usr/local/lib/php.ini" in un editor. La sezione delle "extensions" si trova attorno alla riga 371.

Dovrebbe essere qualcosa del genere:

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

```
....
```

```
extension_dir = ./ <= rimuovete questo e sostituitelo con il seguente:
```

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

Ora salvate il file.

Avete finito – ora avete un sistema LAMP completo e funzionante!

Adesso testiamo il server. Questo prova se potete avviare con successo il server. Il primo passo è chiudere eventuali vecchi server che potrebbero essere in esecuzione (se un Web server era già installato quando avete installato la distribuzione, ad esempio). Per fare ciò, eseguite il seguenti comandi:

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

Ora tentate di avviare il server come segue:

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

Se vedete il messaggio seguente...

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

... tutto è a posto il il vostro server è Up and Running!

Ora spostatevi nella vostra cartella "web/htdocs" (DocumentRoot – se l'avete chiamata in modo diverso, ricordatevi di modificare in accordo sotto) e lì create un nuovo file. Chiamate il nuovo file info.php. Per fare ciò, procedete come segue:

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

Apriete "info.php" con un editor e scrivetevi:

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

Nota: assicuratevi di scrivere questo esattamente come appare qui, incluse le parentesi!

salvate il file e chiudetelo. Adesso la parte più emozionante! Apriete la seguente URL nel vostro browser:

```
http://127.0.0.1/info.php  
or  
http://localhost/info.php  
or  
http://rechnername/info.php  
or  
http://local_IP/info.php
```

A questo punto, se vedete l'output di phpinfo(), tutto è andato secondo i piani, e potete programmare in php

già da subito. Congratulazioni! Adesso non avete solo un sistema LAMP completamente funzionante ma anche un Web server.

Nota: ora potete creare quante subdirectories volete nella subdirectory /web/htdocs (o qualsiasi altra directory avete specificato durante l'installazione). Ad esempio, se avete più progetti web, potete creare una directory per ogni progetto.

Notare che /web/htdocs (o l'altra directory che avete specificato) è la cartella root per quanto riguarda il webserver. Questo è il motivo perchè l'URL è `http://127.0.0.1/info.php`. Se avete altre subdirectories, ad es. `project1`, `project2`, etc., dovreste modificare le URL in accordo: `http://127.0.0.1/project1/` o `http://127.0.0.1/project2/`, etc.

Note: i files PHP sono "parsati" (mostrati/eseguiti) solo se si trovano in questi percorsi, ad es. o /web/htdocs o le sue subdirectories!

Potete aggiungere componenti al webserver al vostro piacere.

Raccomandazioni

In questa sezione raccomando alcuni strumenti di amministrazione che renderanno il vostro lavoro con il sistema e con il webserver molto più facile:

Webmin 1.000

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

Webmin permette di amministrare il sistema con assoluta facilità. Potete anche usarlo come un modo facile per configurare il vostro webserver, es. per fare ripartire il server ad ogni avvio del sistema. Lo stesso vale per MySQL. Webmin è semplice da usare e per questo è ideale per i principianti di webserver.

Download:

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

or

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

phpMyAdmin 2.3.1

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

phpMyAdmin è un potente strumento per MySQL. Permette di creare, cancellare e modificare tabelle e molto di più. Anche questo è caldamente raccomandato.

Download:

[phpMyAdmin-2.3.1-php.tar.gz](http://www.phpmyadmin.net/index.php?dl=2)

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

Configurazione usando Webmin

Impostazioni per il server web Apache:

webmin => Server => Apache Webserver

Module config:

Apache server root directory:
/usr/local/apache/1.3.27/bin/

Path to httpd executable:
/usr/local/apache/1.3.27/bin/httpd

Apache version:
selezionate un campo vuoto e scrivete: => **1.3.27**

Path to apachectl command:
in un campo vuoto => **/usr/local/apache/1.3.27/bin/**

Command to start Apache:
in un campo vuoto => **/etc/init.d/apachectl start**

Nota: se apachectl non è nella directory, copiatelo lì:
root > cp /usr/local/apache/1.3.27/bin/apachectl /etc/init.d/

Command to stop Apache:
in un campo vuoto => **/etc/init.d/apachectl stop**

Display virtual servers as:
=> **Icons**

Order virtual servers by :
=> **order in config file(s)**

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

Path to httpd.conf
in un campo vuoto => **/usr/local/apache/1.3.27/conf/httpd.conf**

Path to srm.conf
in un campo vuoto => **/usr/local/apache/1.3.27/conf/srm.conf**

Path to access.conf
in un campo vuoto => **/usr/local/apache/1.3.27/conf/access.conf**

Path to mime.types
in un campo vuoto => **/usr/local/apache/1.3.27/conf/mime.types**

File to add virtual servers to:
=> **httpd.conf**

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

Nota: non inserite => !

Se volete che il server Apache parta automaticamente al boot di sistema, potete impostarlo in Webmin come segue:
webmin => System => Bootup and Shutdown

Se Apache non è visualizzato lì, basta aggiungerlo come un nuovo servizio.

Name => **apachectl**

Script => **is usually loaded automatically**

Start at boot time? => **Yes**

Bootstrap commands => **/etc/init.d/apachectl start**

Shutdown commands => **/etc/init.d/apachectl stop**

Aggiornamento

Aggiornare PHP

Prima di tutto, copiate il file compresso della nuova versione di PHP nella seguente directory:

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

Mettiamo che la nuova versione di PHP è chiamata PHP 4.2.4 (Non so se ci sarà mai una versione di PHP con questo nome). Questa sezione descrive le operazioni che dovete fare per aggiornare PHP. In sostanza, tutto quello che dovete fare è sostituire la vecchia versione di PHP con quella nuova.

Decomprimete il file che avete appena copiato, come segue.

Aperte una console (shell terminal, e.g. Bash), ed eseguite i comandi seguenti:

```
user:~ > su  
[inserite la vostra password di root]  
root:~ > cd /usr/local/src/lamp/  
root:/usr/local/src/lamp > tar -xvzf php-4.2.4.tar.gz
```

Se esiste ancora la vostra vecchia directory dei sorgenti sotto "/usr/local/src/lamp", procedete come segue.

La nostra vecchia directory dei sorgenti è chiamata "/usr/local/lamp/php-4.2.3".

We first need to create a copy of libphp4.so. Give the copy the name "libphp4-4.2.3.so". Do this as follows:

```
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
```

Adesso creiamo un backup del vecchio php.ini:

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

Ora cancellate il vecchio php.ini, siccome è più sensato usare il nuovo:

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

É stata una buona idea quella di tenere la vecchia directory dei sorgenti di PHP, perchè vi siete risparmiati un bel po' di tasi da battere!

Questo perchè la vecchia directory, "/usr/local/lamp/php-4.2.3", contiene un corto Shell script. Prima dell'ultima installazione, questo script ha memorizzato tutti i parametri da ./configure. Quindi, se non avete enuto la vecchia directory, adesso dovrete inserire a mano tutti i parametri!

Ora eseguite ./configure, come segue:

```
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
```

Nota: se non avete tenuto la vecchia directory "/usr/local/lamp/php-4.2.3", dovrete inserire a mano tutti i parametri di ./configure, come descritto nella sezione sopra "Installazione => PHP 4.2.3".

Ora copiate il nuovo php.ini nella directory corretta:

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

Ora basta adattare il nuovo php.ini come serve, come descritto nella sezione sopra "Configurazione => php.ini".

Infine, restartate Apache, e così è finito l'aggiornamento:

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

Note di chiusura

Nota finale

Come tutti sappiamo, nessuno è perfetto in questo tutorial. Se un esperto della materia trattata ha letto questa guida e pensa che qualcosa vada corretto, che manchi qualcosa, o che necessiti di ulteriori spiegazioni, me lo faccia sapere, in modo che io possa migliorare questa guida. Ho messo molta cura e attenzione nella creazione di questa guida ed essa è stata provata con successo in alcuni sistemi con SuSE Linux 8.0 (ma comunque dovrete riuscire ad usarla anche con altre distribuzioni di Linux).

Ci sono certamente molti altri modi di configurare un sistema LAMP oltre a quello qui descritto, ma questa guida è intesa come un aiuto per i principianti a mettere su un Web server. Cercherei di tenere questa guida aggiornata. Suggesto di visitare spesso la mia homepage per vedere se è disponibile una versione aggiornata (vedere il commento a fianco del link per il download).

Riferimenti

- Terrò questa guida aggiornata qui <http://linux.computerbraxas.de/> [in German]
- <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/>

Webpages maintained by the LinuxFocus Editor team

© Sascha Blum

"some rights reserved" see linuxfocus.org/license/

<http://www.LinuxFocus.org>

Translation information:

de --> -- : Sascha Blum ([homepage](#))

de --> en: Orla Shanaghy <o_shanaghy(at)yahoo.com>

en --> it: Massimo Grava <g-massimo(at)libero.it>