

# Instrukcja laboratoryjna do ćwiczenia:

## Instalacja pakietu LAMP w systemie KNOPPIX

### 1. Cel ćwiczenia:

Celem ćwiczenia jest zapoznanie się z zagadnieniami związanymi z instalacją i konfiguracją serwera WWW, bazy MySQL oraz parsera PHP w systemie KNOPPIX.

### 2. Wstęp teoretyczny

Pakiet LAMP jest nazwą oznaczającą w społeczności internetowej wspólną konfigurację systemu Linux, serwera WWW o nazwie Apache, serwera SQL o nazwie MySQL, oraz parsera PHP. Jest to najczęściej spotykana konfiguracja wykorzystywana do serwowania stron WWW w obecnych czasach. Na bazie jej popularności powstał m. in. projekt LAMPpix – specjalna ok. 130 MB wersja przenośnego serwera www z bazą MySQL i regułami PHP, powstała na bazie systemu operacyjnego Knoppix. Taki bootowalny system w praktyce często jest wykorzystywany przez administratorów w chwilach kryzysowych jako rozwiązanie awaryjne do szybkiego uruchomienia serwerów zastępczych, lub dodatkowych. Zwykły system Knoppix w swojej standardowej kompilacji również zawiera gotowe oprogramowanie służące do zamiany zwykłego komputera w serwer WWW.

Najnowszą wersję serwera Apache, parsera PHP i serwera MySQL wraz z niezbędną dokumentacją można pobrać z odpowiednich stron [www.apache.org](http://www.apache.org), [www.php.net](http://www.php.net) i [www.mysql.com](http://www.mysql.com). Na potrzeby niniejszego ćwiczenia będzie jednak wykorzystany Apache w wersji 1.3.41, PHP 5.2.6 i MySQL w wersji 5.0.77. Serwery we wspomnianych wersjach są udostępnione na stronie kursu.

### 3. Serwer MySQL

#### 3.1. Kompilacja oraz instalacja serwera MySQL

Po uruchomieniu systemu KNOPPIX, pobraniu serwera i zalogowaniu się jako użytkownik uprzywilejowany archiwum o nazwie *mysql\_5.0.77.tar.gz* należy pobrać ze strony i rozpakować. Możemy w tym celu wykorzystać program Midnight Commander (polecenie *mc*). Po jego uruchomieniu należy „wejść” do pliku *tar.gz* i skopiować katalog *mysql-5.0.77* np. do katalogu **/ramdisk**. Następnie należy wejść do utworzonego katalogu **/ramdisk/mysql-5.0.77** i dokonać konfiguracji serwera przed kompilacją wykonując poniższe polecenia:

```
mkdir /ramdisk/mysql  
./configure --prefix=/ramdisk/mysql
```

Parametr **--prefix** wskazuje katalog, do którego zostanie zainstalowany serwer.

Warto też zapoznać się z zawartością pliku **README** oraz **INSTALL**.

Kompilacja serwera odbywa się po wydaniu polecenia **make**. Instalacja serwera odbywa się po wydaniu polecenia **make install**.

### 3.2 Uruchomienie i testowanie serwera MySQL

Po zainstalowaniu serwera należy utworzyć bazy danych. W tym celu należy przejść do katalogu **/ramdisk/mysql/bin** i wykonać polecenie:

```
makedir /ramdisk/sqldata  
chown mysql:mysql /ramdisk/sqldata  
chmod go+w /ramdisk/sqldata  
./mysql_install_db --user=mysql --datadir=/ramdisk/sqldata
```

Aby uruchomić serwer MySQL należy przejść do katalogu **/ramdisk/mysql/libexec** i wykonać polecenie:

```
./mysqld --user=mysql --datadir=/ramdisk/sqldata
```

Do przetestowania poprawności pracy serwera można wykorzystać program o nazwie **mysql** znajdujący się w katalogu **/ramdisk/mysql/bin**:

```
./mysql --user=mysql
```

Jeśli program działa prawidłowo pojawi się znak zachęty **mysql>**. Wpisanie poniższych poleceń upewni nas o prawidłowej pracy serwera:

```
mysql> SHOW databases;  
mysql> USE mysql;  
mysql> SHOW tables;  
mysql> SELECT * FROM user;  
mysql> quit
```

Dwa pierwsze polecenia pozwalają na wyświetlenie i wybór bazy danych znajdującej się na serwerze. Trzecie polecenie wyświetla listę tabel znajdujących się w wybranej bazie danych. Czwarte polecenie wyświetla listę wszystkich rekordów znajdujących się w wybranej tabeli. Ostatnie polecenie kończy pracę s klientem **mysql**.

## 4. Serwer Apache

### 4.1 Kompilacja oraz instalacja serwera Apache

Po pobraniu archiwum *apache\_1.3.41.tar.gz* należy rozpakować kody źródłowe serwera np. do katalogu **/ramdisk/apache\_1.3.41**. Następnie należy wejść do nowo utworzonego katalogu i dokonać konfiguracji serwera przed kompilacją wykonując poniższe polecenia:

```
mkdir /ramdisk/apache  
./configure --prefix=/ramdisk/apache --enable-module=so
```

Parametr **--prefix** wskazuje katalog, do którego zostanie zainstalowany serwer, zaś parametr **--enable-module=so** zapewni bezproblemową współpracę z parserem PHP.

Warto też zapoznać się z zawartością pliku **README** oraz **INSTALL**.

Kompilacja serwera odbywa się po wydaniu polecenia **make**. Instalacja serwera odbywa się po wydaniu polecenia **make install**.

Po zainstalowaniu serwera listę dostępnych modułów sprawdzamy poleceniem

**./httpd -l**

Na liście powinien znajdować się moduł **mod\_so**.

## 4.2 Uruchamianie i testowanie serwera Apache

Serwer można uruchomić poleceniem **httpd** lub poleceniem

**./apachectl start**

Zatrzymanie lub restart serwera następuje po wydaniu odpowiedniego polecenia

**./apachectl stop**

**./apachectl restart**

Jeżeli proces przebiegł bez problemów – nastąpi powrót do wiersza poleceń, a serwer zostanie uruchomiony. Jeśli serwer nie uruchomi się, informacje o błędach zostaną zapisane do pliku **error\_log**. (domyślnie w katalogu **logs**). Serwer jest zwykle uruchamiany jako demon, który wykonuje stałą kontrolę nad połączeniami. Wywoływanie Apache'a za pośrednictwem demona internetowego **inetd** nie jest zalecane.

W celu upewnienia się, że serwer pracuje poprawnie wystarczy uruchomić przeglądarkę stron www i w pasku adresowym wpisać lokalny adres IP komputera (127.0.0.1).

## 5 Instalacja biblioteki Libxml

Aby zapewnić poprawną pracę parsera PHP, przed jego instalacją należy zainstalować bibliotekę Libxml. Po pobraniu archiwum *libxml2-2.7.6.tar.gz* należy rozpakować kody źródłowe biblioteki np. do katalogu **/ramdisk/libxml2-2.7.6**. Następnie należy wejść do nowo utworzonego katalogu i dokonać konfiguracji serwera przed kompilacją wykonując poniższe polecenia:

**mkdir /ramdisk/libxml**

**./configure --prefix=/ramdisk/libxml**

Kompilacja oraz instalacja biblioteki następuje po wpisaniu poleceń **make** oraz **make install**.

## 6. Parser PHP

### 6.1 Konfiguracja i uruchomienie serwera PHP

Po pobraniu archiwum *php\_5.2.9.tar.gz* należy rozpakować kody źródłowe parsera do np. katalogu **/ramdisk/php\_5.2.7**. Następnie należy wejść do nowo utworzonego katalogu i dokonać konfiguracji parsera przed kompilacją wykonując poniższe polecenia:

```
mkdir /ramdisk/php
./configure
    --prefix=/ramdisk/php
    --with-apxs=/ramdisk/apache/bin/apxs
    --with-mysql=/ramdisk/mysql
    --with-libxml-dir=/ramdisk/libxml
```

parametr **--prefix** wskazuje katalog, do którego zostanie zainstalowany parser, parametry **--with-apxs**, **--with-mysql** oraz **--with-libxml-dir** zapewnią poprawną współpracę parsera z serwerem Apache, serwerem MySQL oraz biblioteką Libxml.

Kompilacja oraz instalacja parsera następuje po wpisaniu poleceń **make** oraz **make install**.

## 6.2 Konfiguracja serwera Apache do pracy z PHP

W kolejnym kroku należy otworzyć plik konfiguracyjny Apache (**/ramdisk/apache/httpd.conf**) i sprawdzić czy znajduje się w nim wiersz

```
LoadModule php5_module modules/libphp5.so
```

Jeżeli nie, należy go dopisać oraz dodać wiersz

```
AddType application/x-httpd-php .php .php5
```

Ostatnia dyrektywka zawiera listę plików, które będą parsowane przez PHP. Teraz należy zapisać zmiany i dokonać restartu serwera Apache - pakiet LAMP jest gotowy do pracy.

## 7. Zadania do wykonania

1. Pobrać ze strony kursu plik *mysql-5.0.77.tar.gz* i rozpakować do katalogu **/ramdisk/mysql-5.0.77**
2. Skompilować serwer MySQL w taki sposób, by jego późniejsza instalacja nastąpiła do katalogu **/ramdisk/mysql**.

**UWAGA!** Kompilacja serwera (polecenie **make**) jest czynnością czasochłonną, trwającą około 20 minut. W międzyczasie należy zainstalować serwer Apache oraz bibliotekę Libxml, tj. wykonać kroki 5-9 opisu. Kroki 3, 4 należy wykonać po zakończeniu się kompilacji serwera MySQL i rozpoczęciu kompilacji parsera PHP w kroku 10

3. Dokończyć instalację serwera MySQL (polecenie **make install**)
4. Uruchomić serwer MySQL i przetestować poprawność jego działania
5. Pobrać ze strony kursu pliki *apache\_1.3.41.tar.gz*, *libxml2-2.7.6.tar.gz*, oraz *php-5.2.6.tar.gz* i rozpakować je do odpowiednich katalogów **/ramdisk/apache\_1.3.41**, **/ramdisk/libxml2-2.7.6**, **/ramdisk/php-5.2.6**
6. Skompilować serwer Apache i zainstalować do katalogu **/ramdisk/apache**
7. Należy skonfigurować serwer Apache w ten sposób, by po jego uruchomieniu i wpisaniu adresu *http://127.0.0.1/* w oknie przeglądarki otworzyła się strona testowa. W tym celu należy w pliku konfiguracyjnym nadać zmiennej **ServerName** wartość

**localhost.localdomain** oraz zmiennej **DocumentRoot** wartość **/ramdisk/apache/htdocs**.

8. Uruchomić serwer Apache i przetestować poprawność jego działania
9. Zainstalować bibliotekę Libxml do katalogu **/ramdisk/libxml**
10. Skompilować i zainstalować parser PHP do katalogu **/ramdisk/php**

**UWAGA!** Kompilacja parsera (polecenie **make**) jest czynnością czasochłonną, trwającą około 10 minut. W międzyczasie należy dokończyć instalację i przetestować działanie serwera MySQL, tj. wykonać kroki 3, 4 opisu.

11. Skonfigurować serwer Apache do pracy z parserem PHP i zrestartować.
12. Ze strony kursu należy pobrać stronę (**index.tar.gz**) testującą poprawność działania serwera www, parsera PHP oraz serwera MySQL. Stronę należy rozpakować do katalogu **/ramdisk/apache/htdocs**, wyedytować oraz wpisać w odpowiednie miejsce nazwiska osób wykonujących ćwiczenie
13. Wpisać w pasku adresowym URL pliku *http://127.0.0.1/index.php*. Zrobić zrzut ekranu wyświetlonej strony i zamieścić w sprawozdaniu

## Załącznik 1 - Dyrektywy konfiguracyjne pliku **httpd.conf** i moduły serwera Apache

### Dyrektywy pliku konfiguracyjnego serwera Apache

**ServerAdmin** *adres\_pocztowy\_administratora* – umożliwia podanie adresu e-mail administratora serwera

**ServerName** *nazwa\_domenowa\_serwera* - nazwa hosta, która będzie zwracana klientom podczas pobierania danych z serwera,

**ServerRoot** "katalog" - określa katalog plików serwera, takich jak dzienniki, informacje o błędach, etc.

**ServerType** *typ* - możliwe wartości typu to **standalone** (ustawiany gdy serwer uruchamiany jest samodzielnie) lub **inetd** (gdy serwer uruchamiany jest za pośrednictwem demona inetd).

**Port** *nr* – określa numer portu używanego przez serwer (standardowo 80, w przypadku zmiany zalecane są numery powyżej 1024, np. 8000),

**BindAddress** *adres\_IP* - określa adres używany do komunikacji z serwerem, jeśli posiada on kilka adresów IP, znak \* nakazuje serwerowi odpowiadać na żądania kierowane pod dowolny adres

**Listen** *port* – nasłuch dodatkowych portów

**Listen** *adres:port* – nasłuch dodatkowych adresów IP i portów

**MinSpareServers** *nr* – określa minimalną liczbę bezczynnych procesów potomnych (gdy serwer pracuje w trybie standalone)

**MaxSpareServers** *nr* – określa maksymalną liczbę procesów potomnych (gdy serwer pracuje w trybie standalone)

**StartServers** *nr* - liczba procesów potomnych serwera uruchamianych przy starcie systemu (gdy serwer pracuje w trybie standalone)

**MaxClients** *nr* – określa max. liczbę równocześnie obsługiwanych klientów (gdy serwer pracuje w trybie standalone)

**DocumentRoot** "katalog" - katalog z główną stroną serwera

**UserDir** *katalog* - określenie katalogu przechowującego strony domowe każdego użytkownika systemu

**DirectoryIndex** *nazwa\_pliku* – domyślna nazwa pliku przesyłanego klientom gdy żądanie nie zawiera nazwy pliku (np. **index.html**).

**<Directory "katalog"></Directory>** - określenie dyrektyw odnoszących się do danego katalogu

**<VirtualHost "nazwa\_domenowa"></VirtualHost>** - określenie domeny wirtualnego hosta, na którą zareaguje serwer (adres DNS), oraz dyrektyw odnoszących się do tego serwera.

**LoadModule nazwa\_modułu adres\_pliku\_modułu** – dyrektywa umożliwia załadowanie dodatkowego modułu rozszerzeń serwera. Opisy wybranych modułów podano poniżej.

## **Moduły serwera Apache**

**core** - podstawowe funkcje zawsze dostępne w dystrybucji (kontrola innych modułów)

**mod\_access** - kontrola dostępu do plików w zależności od adresu IP i/lub nazwy komputera klienta. Użycie tego modułu pozwala na dokładną kontrolę użytkowników, np. administrator może zezwolić na wykonywanie skryptów CGI tylko pracownikom firmy.

**mod\_actions** - odpowiada za wykonywanie skryptów CGI w zależności od typu danych lub sposobu pobrania.

**mod\_alias** - udostępnia część systemu plików w katalogu głównym Apache'a, przekierowuje adresy URL.

**mod\_asis** - deklaracja plików wysyłanych bez nagłówków HTTP (pliki \*.asis).

**mod\_auth** - moduł odpowiedzialny za uwierzytelnianie użytkowników na podstawie zdefiniowanych plików tekstowych

**mod\_auth\_anon** – moduł pozwala anonimowym użytkownikom na dostęp do danych podlegających weryfikacji dostępu

**mod\_auth\_db** - uwierzytelnianie za pomocą plików DB (Berkeley)

**mod\_auth\_dbm** - uwierzytelnianie za pomocą plików DBM

**mod\_auth\_digest** - uwierzytelnianie za pomocą MD5

**mod\_autoindex** - automatyczne tworzenie indeksów (wyświetlenie zawartości) dla katalogów, które nie mają standardowych plików index.\*htm\*

**mod\_cern\_meta** - emulacja plików CERN HTTPD, pozwala dodawać dodatkowe nagłówki do wszystkich plików.

**mod\_cgi** - umożliwia wykonanie skryptów CGI po stronie serwera i zwrócenie wyników klientowi

**mod\_digest** - uwierzytelnianie za pomocą algorytmu MD5

**mod\_dir** - operacje na katalogach - do uzupełniania adresu, np. *http://serwer.pl/plik* zostanie zastąpiony poprawnym wywołaniem *http://serwer.pl/plik/*

**mod\_env** - Moduł odpowiedzialny za przekazywanie zmiennych środowiskowych do skryptów CGI/SSI

**mod\_example** - demonstracja możliwości interfejsu programowego, Apache API

**mod\_expires** - dodaje znacznik Expires (strona wygasa, traci ważność) do stron WWW przesyłanych klientowi, ważne dla często zmienianych serwisów, które powinny być zawsze aktualne

**mod\_headers** - pozwala na dowolną modyfikację nagłówków HTTP

**mod\_imap** - wsparcie dla map plików graficznych (\*.map), używane po stronie serwera. Zastępuje program CGI imagemap

**mod\_include** - pozwala włączać zawartości plików lub wyniki działania skryptu do zwykłych plików HTML i zwracać ich zawartość klientowi

**mod\_info** - odpowiedzialny za informację o ustawieniach serwera Apache

**mod\_isapi** - pozwala używać rozszerzeń serwerowych ISAPI. Tylko dla Apache'a w wersji dla Windows

**mod\_log\_agent** – pozwala na zapisywanie w logach nazw i wersji przeglądarek internetowych klientów

**mod\_log\_config** – pozwala na konfigurowalne logowanie zdarzeń, pliki log zapisywane są w formacie Common Logfile Format

**mod\_log\_referer** – pozwala na logowanie odwołań do plików umieszczonych na serwerze

**mod\_mime** – pozwala na określenie typu pliku na podstawie rozszerzenia

**mod\_mime\_magic** – pozwala na określenie typu pliku na podstawie kilku bajtów jego zawartości

**mod\_mmap\_static** – pozwala określić pewne niezmiennie pliki, które zostaną umieszczone w pamięci serwera Apache w celu szybszego dostępu

**mod\_negotiation** - odpowiedzialny za uzgadnianie najlepszej reprezentacji danych w przeglądarce klienta. Wprowadzony ze względu na zgodność z HTTP/1.1

**mod\_proxy** - Apache staje się serwerem proxy dla stron WWW, przyspiesza dostęp do często używanych danych, gdy serwer WWW (komputer) jest wykorzystywany do zapamiętywania danych

**mod\_rewrite** - pozwala modyfikować adresy URL w locie

**mod\_setenvif** – modyfikuje zmienne środowiskowe na podstawie danych klienta

**mod\_so** - ładowanie dodatkowych modułów podczas działania serwera

**mod\_speling** - poprawianie pomniejszych błędów w adresach URL

**mod\_status** - wyświetla bieżący stan serwera Apache

**mod\_userdir** - ustawienia dotyczące katalogów domowych użytkowników

**mod\_unique\_id** - generuje unikalny identyfikator dla każdego żądania

**mod\_usertrack** - śledzenie zachowania użytkowników za pomocą Cookies (ciasteczek), szczególnie przydatne dla np. stałych klientów w sklepach internetowych lub do określania preferencji użytkownika

**mod\_vhost\_alias** - dynamiczna konfiguracja wirtualnymi hostami

## Załącznik 2 - Parametry skryptu configure parsera PHP

Opis i przykładowe wartości parametrów skryptu configure:

**--with-apxs=/ramdisk/apache/bin/apxs** - informujemy, że chcemy skompilować PHP jako moduł serwera Apache i podajemy ścieżkę do programu apxs dostarczanego wraz z Apache.

**--with-config-file-path=/ramdisk/apache/conf** – wymagamy, by plik konfiguracyjny PHP znajdował się w katalogu /ramdisk/apache/conf

**--with-zlib** - dodajemy bibliotekę **zlib**, potrzebną niektórym modułom

**--with-mysql=/ramdisk/mysql** – aktywujemy „stare” rozszerzenie do obsługi baz danych MySQL, podając ścieżkę do katalogu, w którym znajduje się folder z nagłówkami C serwera.

**--with-mysqli=/ramdisk/mysql/bin/mysql\_config** - aktywujemy „nowe” rozszerzenie do obsługi baz danych MySQL, podając ścieżkę do programu **mysql\_config**, generującego odpowiednie pliki.

**--with-pdo-mysql=/ramdisk/mysql/bin/mysql\_config** - aktywujemy sterownik serwera MySQL dla biblioteki PHP Data Objects. Ścieżka dostępu także prowadzi do programu mysql\_config.

**--with-gd** - aktywujemy bibliotekę GD do generowania obrazków

**--enable-gd-native-ttf** - aktywujemy wbudowaną obsługę czcionek TTF w bibliotece GD.

**--with-libxml-dir=/ramdisk/libxml** - podajemy ścieżkę dostępu do katalogu biblioteki libxml, dzięki czemu aktywne będą moduły do obsługi XML’a w PHP.

## Załącznik 3 - Konfiguracja serwera MySQL

Poniżej znajduje się lista wybranych poleceń interpretowanych przez serwer MySQL, nie będących natywnymi elementami języka SQL.

**flush-logs** - odświeża pliki dziennika

**flush-tables** - odświeża tabele

**password *nowe\_hasło*** - zmiana hasła na nowe

**ping** - sprawdza dostępność bazy danych

**processlist** - lista aktywnych procesów

**reload** - przeładowanie tabel

**refresh** - powoduje odświeżenie baz i logów

**show databases** – wyświetla listę baz danych

**show tables** – wyświetla listę tabel wybranej bazy danych

**shutdown** - wyłączenie serwera

**status** - informacja o statusie serwera

**use baza\_danych** – wybór bazy danych

**variables** - informacja o zmiennych

**version** - podaje informacje o oprogramowaniu

Plik konfiguracyjny serwera jest standardowo instalowany w katalogu **/etc/mysql**. Konfiguracji bazy danych MySQL dokonujemy poprzez zmianę pliku **my.cnf**.