

Tematy: [Linux](#) | [Windows](#)

PHP jest oprogramowaniem niezależnym od platformy systemowej, jednak niezależność ta związana jest przede wszystkim z kodem źródłowym PHP — aby móc używać PHP w jednym z dwóch wymienionych w tytule rozdziału systemów operacyjnych, musimy uzyskać odpowiednią dla danego systemu wersję binarną PHP. O ile w przypadku systemów Windows najlepiej jest pobrać od razu gotową wersję binarną, o tyle w przypadku Linuksa optymalnym rozwiązaniem jest pobranie wersji źródłowej i samodzielna kompilacja.

W rozdziale przedstawiony został proces instalacji PHP oraz serwera WWW Apache w systemach Linux i Windows. Wybór systemów podyktowany został ich popularnością. **Linux**

Jeśli w systemie nie został wcześniej zainstalowany serwer WWW Apache, powinniśmy go zainstalować przed rozpoczęciem instalacji PHP. Ze strony <http://httpd.apache.org> musimy pobrać źródła najnowszej stabilnej wersji serwera (będzie to plik o nazwie zbliżonej do `httpd-2.X.X.tar.bz2`, np. `httpd-2.2.13.tar.bz2`, lub `httpd-2.X.X.tar.gz`, np. `httpd-2.2.13.tar.gz`). Mimo iż wersje rozwojowe mogą kusić, to musimy pamiętać, że czasem pracują one niestabilnie, co może się objawiać w postaci problemów z naszą stroną WWW. Wersje niestabilne przeznaczone są przede wszystkim dla testerów i programistów pracujących nad rozwojem oprogramowania. Dlatego też powinniśmy stosować wersje stabilne i przetestowane.

Pobrany z internetu plik umieszczamy w wybranym katalogu, najlepiej `/usr/src`. Uruchamiamy konsolę (wedle innego nazewnictwa terminal — przykładowy wygląd zobrazowany jest na rysunku 3.1) i przechodzimy do wybranego katalogu, wydając polecenie:

```
cd /usr/src/
```

Rysunek 3.1. Przykładowy wygląd konsoli systemowej

Archiwum musimy rozpakować — zakładamy, że plik ma nazwę `httpd-2.2.13.tar.bz2` lub `httpd-2.2.13.tar.gz` (jeśli dysponujemy inną wersją, należy odpowiednio dostosować występujące w opisach nazwy archiwum oraz katalogu). Ponieważ jest ono spakowane dwoma archiwizatorami — `gzip` i `tar` bądź `bzip2` i `tar` — wydajemy kolejno dwa polecenia:

```
bunzip2 httpd-2.2.13.tar.bz2
```

```
tar -xvf httpd-2.2.13.tar
```

albo:

```
gzip -d httpd-2.2.13.tar.gz
```

```
tar -xvf httpd-2.2.13.tar
```

Po rozpakowaniu archiwum w katalogu bieżącym utworzony zostanie podkatalog o nazwie `httpd-2.2.13`. Musimy do niego wejść:

```
cd ./httpd-2.2.13
```

i uruchomić wewnątrz tego katalogu skrypt konfigurujący pliki źródłowe (opcja `--enable-so` umożliwi później kompilację PHP jako modułu serwera):

```
./configure --enable-so
```

Listę wszystkich możliwych opcji skryptu konfigurującego można uzyskać, uruchamiając ten skrypt z parametrem `--help`. Użycie samego `./configure` powoduje zastosowanie domyślnych wartości dla wszystkich opcji, co w typowym zastosowaniu powinno być w zupełności wystarczające.

Następnym krokiem jest kompilacja oprogramowania — dokonujemy tego za pomocą komendy:

```
make
```

Jeśli kompilacja zakończy się powodzeniem (ten proces może zająć dłuższą chwilę), należy wydać komendę:

```
make install
```

Oprogramowanie serwera zostanie zainstalowane w systemie. Warto wiedzieć, że domyślnie Apache instalowany jest w katalogu `/usr/local/apache2` — żeby zmienić lokalizację programu, należy użyć opcji skryptu konfigurującego `configure` o nazwie `prefix`, np.:

```
./configure --prefix=/usr/local/apache2/
```

Po zainstalowaniu serwera WWW trzeba go uruchomić — pierwszy raz można to zrobić „ręcznie”, jednak warto uruchamianie serwera zlecić skryptom startowym systemu, żeby serwer uruchamiał się wraz z systemem operacyjnym.

Wydanie polecenia `httpd` z odpowiednimi argumentami pozwala kontrolować zachowanie serwera, np. w celu uruchomienia wydajemy polecenie:

```
/usr/local/apache2/bin/httpd -k start
```

a w celu zakończenia polecenie:

```
/usr/local/apache2/bin/httpd -k stop
```

Można również korzystać ze skryptu `apachectl`, którego wywołanie ma postać:

```
/usr/local/apache2/bin/apachectl komenda
```

gdzie argument `komenda` oznacza akcję, jaka ma być wykonana. Może to być:

-

`start` — dla uruchomienia serwera,

-

`stop` — dla zatrzymania serwera,

-

`status` — dla wyświetlenia statusu.

Pełne listy opcji dla `httpd` i `apachectl` są dosyć długie i osoby zainteresowane mogą je znaleźć w dokumentacji serwera.

Jeśli serwer ma być uruchamiany wraz ze startem systemu operacyjnego, należy umieścić wywołanie `apachectl` w jednym ze skryptów startowych, np. `/etc/rc.local`.

Uwaga

Jeśli podczas próby uruchomienia serwera `httpd` uzyskamy komunikat o błędzie opisany jako `cannot restore segment prot after reloc: Permission denied` przy próbie ładowania modułu PHP, najprawdopodobniej dysponujemy systemem z zainstalowanym rozszerzeniem SELinux (Security Enhanced Linux) i nie zostały określone właściwe uprawnienia dla tego modułu. Plik `libphp5.so` musi mieć zdefiniowane prawo (dokładniej rolę) `shlib_t`. Problem powinno zatem rozwiązać wykonanie polecenia:

```
chcon /usr/local/apache2/modules/libphp5.so -t shlib_t
```

Po uruchomieniu przeglądarki WWW i wpisaniu adresu `http://localhost` lub `http://127.0.0.1` powinniśmy otrzymać efekt jak na rysunku 3.2 (dla serwera Apache 2.2).

Rysunek 3.2. Strona startowa serwera Apache 2.2

Mając działający serwer WWW, możemy przystąpić do kompilacji PHP. Źródła najnowszej stabilnej wersji PHP możemy znaleźć na stronie <http://www.php.net>. Po pobraniu archiwum zawierającego źródła PHP (zakładamy, że mamy do czynienia z wersją 5.3.0 dystrybuowaną w archiwum o nazwie `php-5.3.0.tar.gz`) rozpakowujemy je do wybranego katalogu, najlepiej `/usr/src`, np. wydając polecenia:

```
gzip -d php-5.3.0.tar.gz
```

```
tar -xvf php-5.3.0.tar
```

Następnie przechodzimy do katalogu `php-5.3.0`, który został automatycznie utworzony podczas dekompresji archiwum i który zawiera źródła PHP gotowe do kompilacji:

```
cd ./php-5.3.0
```

Wewnątrz tego katalogu uruchamiamy skrypt konfigurujący źródła, ale musimy pamiętać o odpowiedniej opcji, która umożliwi nam kompilację i instalację PHP jako modułu serwera:

```
./configure --with-apxs2=/usr/local/apache2/bin/ --with-config-file-path=/etc
```

Pierwsza opcja (`--with-apxs2`) jest konieczna, jeżeli chcemy skompilować PHP jako moduł serwera Apache. Druga opcja jest wprowadzona w tym przykładzie dla naszej wygody — chodzi o to, aby plik konfiguracyjny PHP (`php.ini`) umieszczony był w katalogu `/etc`, co pozwoli nam na łatwy dostęp do niego.

Po udanej konfiguracji źródeł należy wydać komendy `make` i `make install` — PHP zostanie zainstalowany i skonfigurowany (częściowo) do współpracy z serwerem Apache. Musimy jeszcze wprowadzić zmiany w pliku konfiguracyjnym serwera `httpd.conf`, który znajduje się w katalogu `/usr/local/apache2/conf/`. Przede wszystkim musimy „nauczyć” serwer, co ma robić z plikami z rozszerzeniem `php`, dodajemy więc wiersz:

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

oraz umożliwić wczytywanie domyślnej strony w danym katalogu, jeżeli użytkownik nie określi w adresie strony nazwy pliku:

```
DirectoryIndex index.html index.php
```

Upewniamy się także, że w sekcji

Dynamic Shared Object (DSO) Support

znajduje się wpis:

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

Po zapisaniu zmian w pliku `httpd.conf` restartujemy serwer WWW.

Teraz sprawdzimy, czy udało się zainstalować poprawnie PHP. W tym celu w katalogu `/usr/local/apache2/htdocs` stworzymy plik o nazwie `test.php` i wprowadzamy do niego następujący wiersz:

```
<?php phpinfo(); ?>
```

Zapisujemy plik, uruchamiamy przeglądarkę WWW i wpisujemy adres `http://localhost/test.php` lub `http://127.0.0.1/test.php`. Efekt powinien być podobny do tego z rysunku 3.3, czyli powinniśmy zobaczyć stronę informacyjną PHP.

Rysunek 3.3. Strona konfiguracyjna PHP

Pozostaje nam tylko skonfigurować PHP, ale szczegóły na ten temat znajdują się w dalszej części kursu. Na dobry początek warto do katalogu `/etc` skopiować plik `php.ini-recommended` znajdujący się w katalogu ze źródłami PHP i zmienić jego nazwę na `php.ini`. Pamiętajmy, że przy każdej zmianie zawartości pliku `php.ini` powinniśmy zrestartować serwer WWW.

Jeśli dotarliśmy do tego miejsca i nie stało się po drodze nic, czego nie przewidzieliśmy, to oznacza, że możemy już zacząć pracę w PHP działającym w systemie Linux. **Windows**

Instalacja PHP w systemach Windows jest prostsza niż w systemach Linux, ponieważ nie musimy kompilować oprogramowania — możemy użyć dobrze przygotowanych wersji binarnych zarówno serwera WWW, jak i samego PHP.

Zacznijmy od instalacji serwera WWW Apache. Na początek ze strony <http://httpd.apache.org> musimy pobrać najnowszą stabilną i współpracującą z PHP wersję serwera (w trakcie pisania tego kursu była to wersja 2.2.13) — plik instalacyjny będzie miał nazwę podobną do `apache_2.2.13-win32-x86-no_ssl.msi`. W trakcie instalacji będziemy musieli podać parę niezbędnych do poprawnej pracy serwera danych, takich jak:

-
domena sieciowa serwera,

-
nazwa serwera,

-
adres e-mail administratora,

oraz podjąć decyzję, czy serwer ma pracować jako usługa (domyślnie), czy jako program uruchamiany ręcznie.

Po zakończeniu instalacji serwer zostaje uruchomiony — możemy sprawdzić jego stronę startową, wpisując w polu adresu przeglądarki `http://localhost`. Zobaczymy witrynę z napisem `It Works!`, czyli taką samą jak w przypadku instalacji w systemie Linux.

Jeśli serwer WWW działa, możemy przystąpić do instalacji PHP. Wersję binarną PHP dla Windows znajdziemy na stronie `http://www.php.net` — będzie to zarówno archiwum zawierające samo PHP z niezbędnymi dodatkami, jak i wersja zawierająca typowy instalator (plik będzie miał nazwę podobną do `php-5.3.0-Win32-VC9-x86.msi` lub `php-5.3.0-Win32-VC9-x86.zip`). Jeżeli zdecydujemy się na wersję z instalatorem, poprowadzi on nas przez wszystkie etapy instalacji. W trakcie tego procesu będziemy mogli wybrać serwer WWW — oczywiście wskażemy Apache. Wtedy cała konfiguracja powinna odbyć się automatycznie, bez naszego udziału.

Jeśli pobierzemy tylko archiwum zip, rozpakowujemy jego zawartość do wybranego folderu, np. `c:\Program Files\php` (lub nawet `c:\php`). Następnie w pliku konfiguracyjnym serwera Apache wpisujemy następujące wiersze:

```
LoadModule php5_module "c:/Program Files/php/php5apache2.dll"
```

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

Pierwszy z nich powoduje załadowanie interpretera PHP jako modułu serwera, drugi definiuje, co serwer ma robić z plikami z rozszerzeniem PHP. Oczywiście możemy jeszcze dodać wiersz:

```
DirectoryIndex index.html index.php
```

który umożliwi nam pobieranie domyślnej strony WWW zapisanej jako skrypt PHP. Po dodaniu tych wierszy do pliku konfiguracyjnego serwera WWW i zmianie nazwy pliku `php.ini-recommended` na `php.ini` (w folderze `c:\Program Files\php`) możemy zrestartować serwer WWW.

Aby przetestować instalację PHP, tworzymy plik `index.php` o treści jak w przypadku pliku `test.php` przedstawianego w opisie instalacji w systemie Linux i umieszczamy go w folderze `htdocs` znajdującym się w folderze, w którym zainstalowany został serwer Apache. Po wprowadzeniu adresu `http://localhost` powinniśmy zobaczyć efekt jak na rysunku 3.4.

Rysunek 3.4. Test instalacji PHP w systemie Windows

Jeżeli okno naszej przeglądarki przypomina okienko z rysunku 3.4, jesteśmy gotowi do rozpoczęcia przygody z PHP w systemie Windows.