



OBSD.RU

[Home](#)

## Web server bozohttpd + PHP + Mysql

[printable page](#)

Калегин Сергей Николаевич

(май 2011 года)

В данной статье я хотел бы поделиться своим опытом и рассказать о том как использовать встроенный Web-сервер операционных систем NetBSD 5.x в связке типа HTTPD+PHP+MySQL вместо установки более мощного, но и более громоздкого и сложного Apache.

Итак. Для начала хотелось бы отметить тот факт, что в ОС NetBSD последних версий (5.0 и выше) уже есть простой, но довольно мощный, Web-сервер bozohttpd, который находится в папке /usr/libexec/. А следовательно, если не требуется особой гибкости и огромного количества поддерживаемых модулей (как в Apache), то этот сервер вполне подойдёт, например, для тестов начинающим Web-программистам или же для размещения домашнего (или корпоративного) сайта в локальной (корпоративной) или региональной (районной) сети. Преимущества такого решения очевидны:

- 1) Не требуется установки и конфигурации ещё одного сервера;
- 2) Не будет захламляться системный носитель и каталоги конфигурации всяким мусором;
- 3) Меньше будет установлено дополнительных пакетов;
- 4) У bozohttpd нет собственных файлов конфигурации, стартовых скриптов и дополнительно загружаемых модулей. А значит его настройка и обслуживание будет доступна даже новичкам.
- 5) При любых проблемах в системе и ошибках пользователя переустановка этого сервера не требуется! и т.д.

Данный список можно продолжить, но в этом нет необходимости. Я думаю, что каждый администратор найдёт своё применение этому замечательному серверу.

Запуск bozohttpd производится как обычно, из файла /etc/rc.conf следующей директивой:

**httpd=YES**

Однако, я рекомендую переписать сюда и другие директивы, относящиеся к httpd, чтобы потом было удобнее его настраивать. Найти их можно в файле /etc/defaults/rc.conf. А можно просто воспользоваться командой grep, например так:

**grep httpd /etc/defaults/rc.conf**

и затем выбрать нужные директивы, или же сразу дописать всё в /etc/rc.conf командой типа:

**grep httpd /etc/defaults/rc.conf >> /etc/rc.conf**

После этого нужно прописать домашний каталог нашего сайта в директиве httpd\_wwwdir, например так:

**httpd\_wwwdir="/WWW"**

Если ничего не менять, то по умолчанию будет использован каталог `/var/www/`.

И последняя директива, которую мы рассмотрим, `httpd_flags`. Именно здесь прописываются все параметры, которые передаются серверу при запуске. По умолчанию серверу никакие параметры не передаются, а следовательно кавычки будут пустые:

```
httpd_flags=""
```

Хотелось бы напомнить, что первая (стартовая) Web-страничка вашего сайта должна называться `index.html` и находиться непосредственно в папке, указанной в директиве `httpd_wwwdir`.

Всё, после перезапуска (или перезагрузки) `bozohttpd` будет работать и откликаться на 80-м порту, как и Apache.

Проверить это элементарно. Нужно запустить любой браузер на любом компьютере в сети и набрать в адресной строке IP-шник сервера, например:

```
http://192.168.1.50/ (если IP-адрес NetBSD-сервера 192.168.1.50)
```

или так:

```
http://192.168.1.50:80/ (с указанием номера порта)
```

После проверки работоспособности сервера можно приступать к установке и настройке дополнительных пакетов. Я приведу пример только для PHP и MySQL, однако другие ставятся и работают аналогично. Установить нужное ПО можно, как минимум, 2-мя способами: компиляцией непосредственно из исходников или же добавлением уже готового пакета с помощью `pkg_add`. Для краткости я возьму второй способ. Качаем с официального сайта пакеты типа `php-5.x.tgz`, `mysql-php.tgz` и `mysql-server-5.x.tgz` (версии могут быть любыми) и устанавливаем их с помощью `pkg_add`, например:

```
pkg_add -v php-5.3.6.tgz
```

После окончания установки всех пакетов перезагружаемся!

Затем подключаем интерпретатор языка PHP к `bozohttpd` с помощью флага (или параметра) `-C .php /usr/pkg/libexec/cgi-bin/php` и устанавливаем новый стартовый файл для сайта - `index.php`. Выглядит это так:

```
httpd_flags="-C .php /usr/pkg/libexec/cgi-bin/php -x index.php"
```

Теперь создаём тестовый PHP-скрипт (например `test1.php`) и запрашиваем его через браузер:

```
http://192.168.1.50/test1.php
```

Всё должно заработать!

Теперь проверяем работу СУБД MySQL. Запускаем `mysql client` (который устанавливается вместе с сервером) и даём несколько тестовых команд, например:

```
mysql -uroot -p (вход под root-ом с пустым паролем)
```

```
show databases; (выводит список баз данных)
```

```
use mysql; (подключаемся к базе mysql)
```

```
show tables; (выводит список таблиц этой базы)
```

```
quit; (выходим из mysql)
```

Если все эти команды были приняты и выполнены интерпретатором, значит MySQL уже работает. Так же можно проверить работу MySQL с помощью скриптов для `/bin/sh`, например таким:

-----  
**#!/bin/sh**

**echo "select Host, User from user" | mysql -uroot -p mysql;**

**exit 0;**

-----

Данный скрипт выведет список хостов и пользователей MySQL.

Теперь осталось только проверить связку PHP+MySQL и соединить их с Web-сервером.  
Пишем скрипт на PHP для соединения с СУБД и делаем простой запрос, например так:

-----

```
$dbcn=mysql_connect("localhost", "root", "");  
mysql_select_db("mysql");  
$r=mysql_query("select Host, User from user");  
for($i=0; $i < mysql_num_rows($r); $i++)  
{ $res=mysql_fetch_array($r); echo "$res[Host] $res[User]\n"; }  
mysql_close($dbcn);
```

-----

Результат выполнения этого скрипта будет аналогичен предыдущему на shell.

Ну вот и всё. Дерзайте!

---