

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ

УСТАНОВКА И НАСТРОЙКА ИНТЕРПРЕТАТОРА PHP

*Учебно-методическое пособие
по дисциплине
«ВЕБ-ПРОГРАММИРОВАНИЕ»*

**Набережные Челны
2018**

Галиуллин Л.А. Установка и настройка интерпретатора PHP: учебно-методическое пособие по дисциплине «Веб-программирование» [Электронный ресурс] / Казанский федеральный университет, Электронный архив, 2018.

Рассматривается установка интерпретатора языка PHP. Представлены установка PHP под Windows и установка PHP под Linux. Приведены контрольные вопросы. Для студентов направлений подготовки «Информатика и вычислительная техника», «Программная инженерия».

Введение

Язык *PHP* был разработан как инструмент для решения чисто практических задач. Его создатель, *Расмус Лердорф*, хотел знать, сколько человек читают его online-резюме, и написал для этого простенькую *CGI*-оболочку на языке Perl, т.е. это был набор Perl-скриптов, предназначенных исключительно для определенной цели – сбора статистики посещений.

CGI (*Common Gateway Interface* – общий интерфейс шлюзов) является стандартом, который предназначен для создания серверных приложений, работающих по протоколу HTTP. Такие приложения (их называют шлюзами или *CGI-программами*) запускаются *сервером* в режиме реального времени. *Сервер* передает запросы пользователя *CGI-программе*, которая их обрабатывает и возвращает результат своей работы на экран пользователя. Таким образом, посетитель получает динамическую информацию, которая может изменяться в результате влияния различных факторов. Сам шлюз (*скрипт CGI*) может быть написан на различных языках программирования – Си/C++, Fortran, Perl, *TCL*, UNIX Shell, Visual Basic, Python и др.

Вскоре выяснилось, что *оболочка* обладает небольшой производительностью, и пришлось переписать ее заново, но уже на языке *Cи*. После этого *исходники* были выложены на всеобщее обозрение для исправления ошибок и дополнения. Пользователи *сервера*, где располагался *сайт* с первой версией *PHP*, заинтересовались инструментом, появились желающие его использовать. Так что скоро *PHP* превратился в самостоятельный проект, и в начале 1995 года вышла первая известная версия продукта, называвшаяся *Personal Home Page Tools* (средства для персональной домашней страницы). Средства эти были более чем скромными: анализатор кода, понимающий всего лишь несколько специальных команд, и набор утилит, полезных для создания гостевой книги, счетчика посещений, чата и т.п.

К середине 1995 года после основательной переработки появилась вторая версия продукта, названная *PHP/FI* (*Personal Home Page / Forms Interpreter* – персональная домашняя

страница/интерпретатор форм). Она включала набор базовых возможностей сегодняшнего *PHP*, возможность автоматически обрабатывать *html-формы* и *встраиваться в html-коды*. Синтаксис *PHP/FI* сильно напоминал синтаксис Perl, но был более простым.

В 1997 вышла вторая версия Си-реализации *PHP – PHP/FI 2.0*. К тому моменту *PHP* использовали уже несколько тысяч человек *повсемю* миру, примерно с 50 тыс. доменов, что составляло около 1% всего числа доменов *Internet*. Число разработчиков *PHP* увеличилось до нескольких человек, но, несмотря на это, *PHP/FI 2.0* все еще оставался крупным проектом одного человека. Официально *PHP/FI 2.0* вышел только в ноябре 1997 года, просуществовав до этого в основном в бета-версиях. Вскоре после выхода его заменили альфа-версии *PHP 3.0*.

Установка PHP под Windows

Перейдем к *установке PHP*. Скачать его *дистрибутив* можно с официального сайта *PHP* – <http://www.php.net>. Для удобства лучше выбрать автоматический инсталлятор, как и в случае с *сервером*. Самое первое окошко при такой *установке PHP* содержит приветствие и предупреждение о существовании авторских прав на этот продукт (рис. 1).

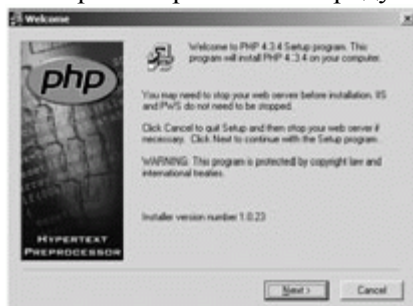


Рис. 1. Программа автоматической установки PHP

Далее нужно прочитать и согласиться (или не согласиться) с лицензией (рис. 2).

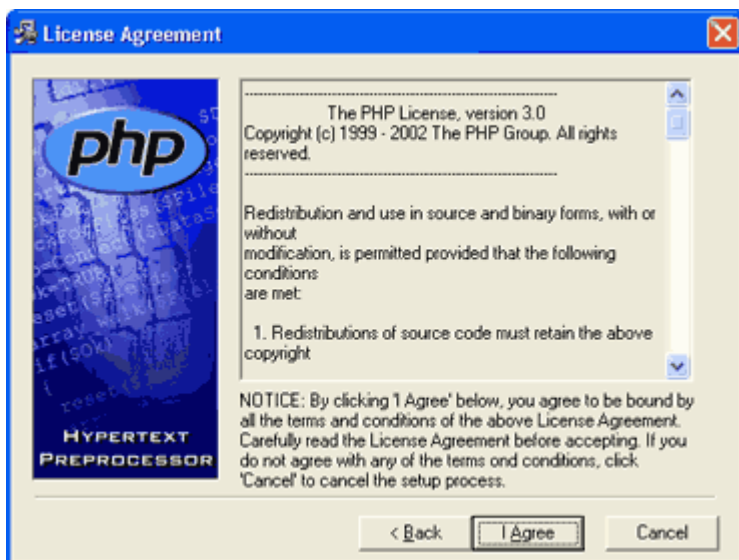


Рис. 2. Лицензионное соглашение

Затем выбираем тип *установки*: стандартный или для специалистов. Мы выберем стандартную *установку* (рис. 3).

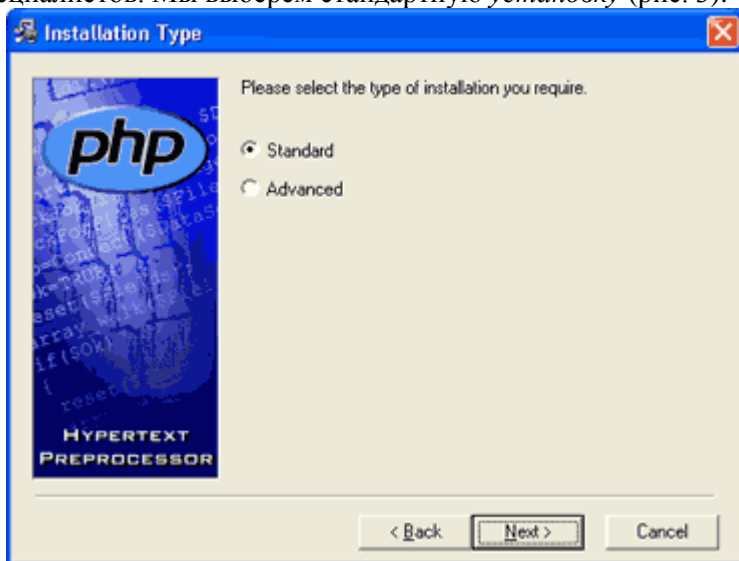


Рис. 3. Тип установки

В следующем окне программа предложит выбрать директорию для установки *PHP*. Туда будут скопированы файлы библиотек, расширений, *интерпретатор* командной строки, модули и т.п. (рис. 4).

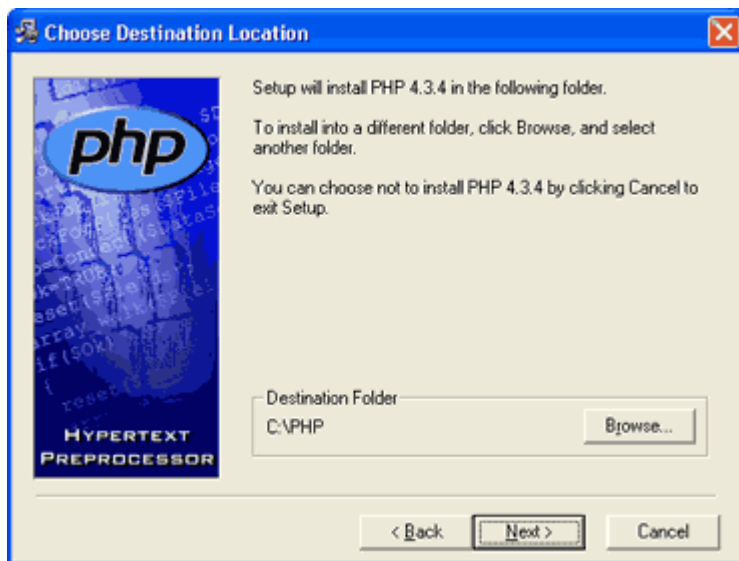


Рис. 4. Выбор папки, в которую будет установлен PHP

В следующем окошке предлагается ввести адрес SMTP-сервера, т.е. сервера отправки почтовых сообщений, и обратный адрес отправителя почты. Это окошко мы приводить не будем.

Далее предлагается выбрать из списка сервер, с которым будет работать *PHP*. Поскольку мы установили *Apache*, то выберем из списка именно его (рис. 5).

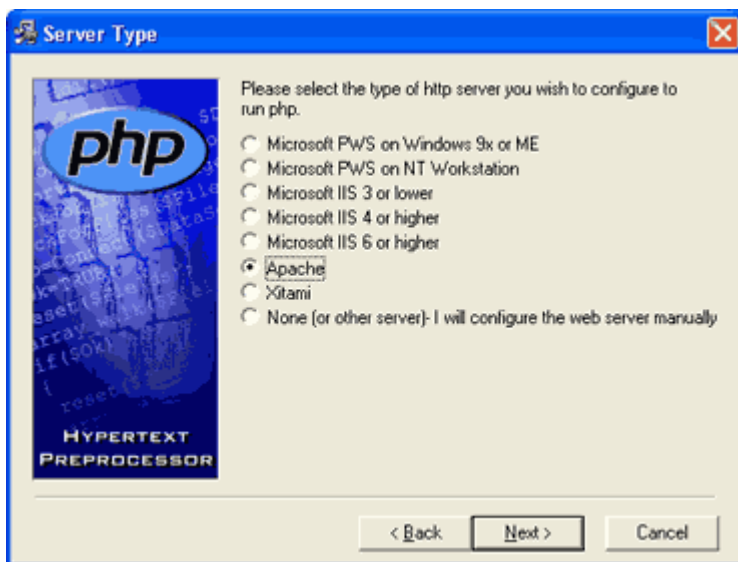


Рис. 5. Выбор сервера, на котором будет работать PHP

Следующее окошко начинает инсталляцию (рис. 6).

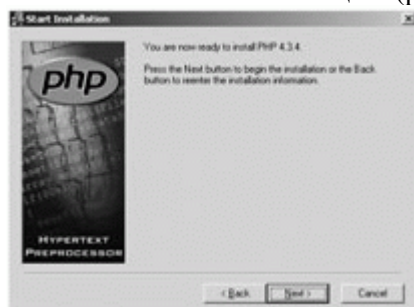


Рис. 6. Начало автоматической установки PHP

После копирования необходимых файлов мы получили следующее сообщение (см. рис. 7):

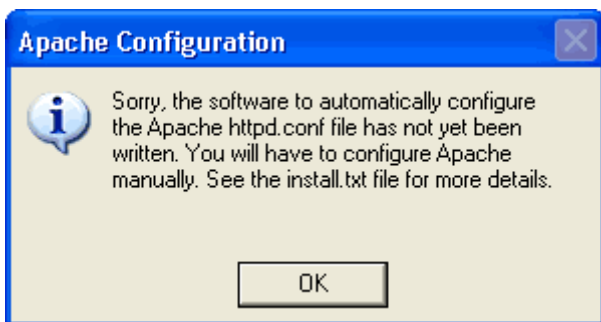


Рис. 7. Сообщение о невозможности автоматически настроить Apache для работы с PHP

Это значит, что нужно настраивать *сервер Apache* для работы с *PHP* вручную.

Сначала следует выбрать, как мы хотим *установить PHP*, поскольку он поставляется в двух видах: *CGI-скрипт* (*php.exe*) или набор *SAPI-модулей* (например, *php4isapi.dll*), используемых *сервером*. Последний вариант обладает новыми возможностями, однако из-за недостаточной проработанности может функционировать недостаточно надежно, особенно на платформах старше Windows 2000 (может появляться куча ошибок 500, могут возникать сбои в других *серверных* модулях, таких как ASP). Так что, если нужна абсолютная стабильность – надо выбирать *установку PHP* в виде *CGI* выполнимого приложения.

Если мы хотим *установить PHP* как серверный модуль, то в файле конфигурации *сервера* (*httpd.conf*) нужно написать:

```
LoadModule php4_module  
    c:/php/sapi/php4apache.dll  
AddType application/x-httpd-php .php .phtml  
AddModule mod_php4.c
```

Если мы устанавливаем *PHP* как *cgi - программу*, то в *httpd.conf* нужно написать:

```
ScriptAlias /php/ "c:/php/"  
AddType application/x-httpd-php .php .phtml  
Action application/x-httpd-php "/php/php.exe"
```

В этом случае могут возникнуть проблемы с безопасностью.

Рекомендуется исправить директорию, где лежит исполняемый файли*интерпретатора* (`c:\php\`), на что-нибудь менее очевидное (например, на `c:\abc_php\`). Мы советуем устанавливать *PHP* как серверный модуль.

Еще нужно отредактировать файл *php.ini* (в папке `c:\Windows`), чтобы *PHP* "знал", где находится корневая директория*сервера*, где пользовательские директории, а где его собственные библиотеки расширений. За это в файле *php.ini* отвечают соответственно переменные *doc_root* , *user_dir* и *extension_dir* . Зададим их таким образом:

```
doc_root = "c:\Program Files\Apache
           Group\Apache\htdocs"
user_dir = "c:\users"
extension_dir = "c:\php\extensions"
```

Кроме того, можно выбрать расширения, которые будут загружаться при запуске *PHP*. В реализацию *PHP* под Windows изначально входит очень мало расширений. Чтобы загрузить расширение, нужно раскомментировать в *php.ini* соответствующую ему строчку'`extension=php_*.dll`'. Например, чтобы загрузить расширение для работы с *MSSQL*, нужно раскомментировать строку'`extension=php_mssql.dll`'. Некоторые расширения требуют дополнительных библиотек. Поэтому рекомендуется скопировать дополнительные библиотеки в системную директорию (из папки `c:\php\dlls`). При первой *установке* следует настроить и протестировать *PHP* без расширений.

Для того чтобы *настройки*, выполненные в конфигурационных файлах *сервера* и *PHP* вступили в силу, нужно перезапустить *сервер*.

Проверим, работает ли *PHP*. Для этого создадим тестовый файл (`1.php`) в директории пользователя (`c:\users\nina`) со следующим содержанием:

```
<?php
echo"<h1>Привет всем!</h1>";
?>
```

Запустим этот файл через браузер, набрав `http://localhost/~nina/1.php`. Если что-то не так, то на экран

будет выведен текст этого файла. Если все хорошо, то наш *скрипт* должен обработаться *сервером* и вывести большими буквами строку "Привет всем!".

Установка PHP под Linux

Как и в случае с Windows, для Linux существует два способа *установки PHP*: используя *пакеты* и непосредственно компилируя *исходный код PHP*, который можно скачать с <http://www.php.net>. Мы остановимся на первом варианте и рассмотрим *установку PHP + Apache* на примере дистрибутива *AltMaster2.2* (<http://altlinux.ru>).

Для того чтобы *установить apache*, необходимо выбрать одноименный *пакет*. Это можно сделать, например, с помощью менеджера *пакетов Synaptic*. Воспользовавшись меню "**Пакет -> Установить**" (рис. 8) нужно выбрать интересующие *пакеты*. После этого необходимо перейти к *установке* выбранных *пакетов* с помощью меню "**Действия -> Выполнить**". После *установки Apache* имеет смысл перезагрузиться или выполнить следующую команду:

```
/etc/init.d/httpd start
# /etc/init.d/httpd start
Starting httpd:
[ OK ]
```

Это приведет к запуску *сервера* и можно будет увидеть по адресу <http://localhost> почти такую же страницу, как и при *установке Apache* под Windows. Возможным отличием будет присутствие логотипа *AltLinux*.



Рис. 8. Выбор пакета для установки Apache под Linux

После установки Apache можно приступить к установке непосредственно PHP. Опять же можно использовать PHP с помощью CGI и через модуль Apache. В первом случае достаточно пакета *php* и необходимых для его установки пакетов. Во втором случае необходимо дополнительно установить пакет *mod_php*. (рис. 9).



Рис. 9. Выбор пакетов для установки PHP под Linux

Менеджер пакетов *Synaptic* – это оболочка к программе *apt-get*, более подробно о ней можно узнать с помощью команды `man apt-get`. Воспользуемся утилитой *apt-get* для установки пакета *mod_php*. Команда *apt-get build-dep mod_php* установит пакеты, которые нам необходимы для инсталляции *mod_php*. Команда *apt-get install mod_php* завершит установку:

```

[root@greydragon apt]# apt-get build-dep mod_php
Reading Package Lists... Done
Collecting File Provides... Done
Building Dependency Tree... Done
Note, selecting libgdbm-devel instead of gdbm-devel
The following NEW packages will be installed:
  apache-devel bison byacc flex libexpat-devel
libgdbm-devel libltdl libmm-devel libpam-devel libtool
0 packages upgraded, 10 newly installed,
0 removed and 73 not upgraded.
Need to get 937kB of archives.
After unpacking 2357kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 ftp://ftp.altlinux.ru ALTLinux/main byacc
  1.9-ipl9mdk [31.8kB]
.....
Get:10 ftp://ftp.altlinux.ru ALTLinux/main
libtool 2:1.4.2-alt0.2 [302kB]
Fetched 937kB in 9m19s (1675B/s)
Executing RPM (/bin/rpm -Uvh --fancypercent
--oldpackage)...
Preparing...
#####
[100%]
....
[root@greydragon apt]# apt-get install mod_php
Reading Package Lists... Done
Collecting File Provides... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  mod_php
0 packages upgraded, 1 newly installed,
0 removed and 73 not upgraded.
Need to get 46.6kB of archives.
After unpacking 69.2kB of additional disk space will be used.
Get:1 ftp://ftp.altlinux.ru ALTLinux/main
mod_php 1:4.3.0-alt0.cvs20030207 [46.6kB]
Fetched 46.6kB in 24s (1918B/s)

```

```
Executing RPM (/bin/rpm -Uvh --fancypercent --oldpackage)...
Preparing...
```

```
#####
```

```
[100%]
```

```
1: mod_php
```

```
#####
```

```
[100%]
```

```
Shutting down httpd: [ OK ]
```

```
Checking configuration sanity for httpd: [ OK ]
```

```
Starting httpd: [ OK ]
```

```
Листинг 1.0. Результат работы менеджера пакетов Synaptic
```

```
Как видно из приведенного
```

```
листинга, скрипт установки самостоятельно
```

```
перезагрузил сервер Apache. Также он самостоятельно внес изменения в конфигурационный файл Apache, полный путь до которого: /etc/httpd/conf/httpd.conf
```

```
В httpd.conf добавлена строка:
```

```
Include conf/addon-modules/mod_php4.conf
```

```
Содержимое файла mod_php4.conf
```

```
LoadModule php4_module
```

```
    usr/lib/apache/libphp4.so
```

```
AddModule mod_php4.c
```

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

```
    .php .php4 .php3 .phtml
```

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

```
AddHandler application/x-httpd-php
```

```
    .php .php4 .php3 .phtml
```

```
Как видно, изменения в httpd.conf подобны изменениям, которые мы вносили при установке под Windows.
```

```
Теперь можно воспользоваться тестовым файлом для проверки работоспособности PHP + Apache.
```

```
Создадим test.php со следующим содержанием:
```

```
<?php
```

```
echo"<h1>Привет всем!</h1>";
```

```
?>
```

```
Корень сервера находится /var/www/html/. Можно создать каталог test и разместить там файл test.php. Вызвав этот файл из браузера по его адресу (http://localhost/test/test.php), получим
```

страничку с текстом "Привет всем".

Контрольные вопросы

1. Что Вы знаете о HTTP?
2. Что Вы знаете о HTML?
3. Что Вы знаете о PHP?
4. Что Вы знаете о Windows?
5. Что Вы знаете о Linux?
6. Что Вы знаете об установке PHP под Windows?
7. Что Вы знаете об установке PHP под Linux?
8. Что Вы знаете о хостинге?
9. Что Вы знаете о веб-сервере Apache?
10. Что Вы знаете о доменных именах?

Рекомендуемые источники

1. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2017. - 416 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=336649>.
2. Гагарина Л.Г. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2015. - 400 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=389963>.
3. Голицына О. Л. Программирование на языках высокого уровня: Учебное пособие / О.Л. Голицына, И.И. Попов. - М.: Форум, 2016. - 496 с. [Электронный ресурс]. <http://znanium.com/bookread.php?book=139428>.