

搭建 PHP 的开发环境

搭建 LAMP,需要在 Linux 操作系统上分别安装 Apache 网页服务器,PHP 应用服务器和 MySQL 数据库管理系统,以及一些相关的扩展。下面介绍三种环境搭建方式:

- 如果需要商业化运营网站,建议 Linux 下源代码包的安装方式;
- 如果选择 Windows 作为服务器的操作系统,可以选择在 Windows 系统上以获取组件安装 Web 工作环境的方式;
- 如果初学 PHP,可以选择集成软件安装,搭建供学习的 PHP 工作环境。

1 操作准备

除了安装 PHP 应用服务器外,还需要安装 Web 服务器 Apache、数据库管理系统 MySQL,以及安装一些相应的功能扩展。这几个服务器软件都能够运行在绝大多数主流的操作系统上,包括 Linux、UNIX、Windows 以及 Mac OS 等。

1.1 选择网站运营的操作系统

现在就有一个容易引起争论的话题,在哪一种操作系统环境下运行这些软件更好,不同的阵营会给出不同的答案。可以有把握地说,这几个相关软件在 UNIX/Linux 环境下的版本有着更高的质量,而且部署在 UNIX/Linux 环境下的软件程序往往有着更高的运行效率。因为 Apache、PHP 和 MySQL 这些软件都是先在 UNIX/Linux 下开发出来,然后才被移植到 Windows 操作系统环境上。另外,在开发时主要使用的 PHP 脚本编程语言,有一些功能模块都是针对 UNIX/Linux 系统开发的,而 Windows 环境则对这些功能模块没有提供所需要的标准化编程接口。所以同样的系统功能在 UNIX/Linux 环境下和 Windows 环境下的具体实现和部署机制往往会有所差异。开发者必须考虑到这类差异才能确保项目的成功。

目前使用 Windows 操作系统的人数还是远远多于使用 Linux 系统的人数。这都是因为 Linux 没有提供很好的图形操作界面,多数功能都要使用命令行工具来完成。所以用户会觉得使用 Linux 很困难,没有 Windows 这么容易上手,提供的程序开发工具软件也没有 Windows 系统中提供得多。不喜欢使用 Linux,所以选用 Windows 系统作为服务器使用。

1.2 选择网站开发的操作系统

一般来说,一个普通的网站软件,在哪个系统下开发并没有多大的差异,并不是一定要作为程序开发,非要先花大量的时间和精力去学习 Linux 操作系统。如果网站还处于开发阶段,用户使用的是一个测试环境,而这个测试环境通常只有开发者本人或者开发者所在的团队来访问。不会因为访问量很大、

访问者的成分很复杂而导致系统在安全或效率等方面出现问题，这个阶段软件在 Windows 系统和 Linux 系统上都有很好的兼容性。所以开发者在开发时应该选择自己最熟悉的操作系统。项目可以先在 Windows 系统下开发，开发完成后再把整个项目移植到 Linux 服务器上去。如果读者是处于 PHP 的学习阶段，这种做法就很值得考虑。读者了解和学习 Linux 可以在 LAMP 兄弟连网站下载 Linux 学习视频，也可以参考李明老师写作的《细说 Linux》。

2 Linux 系统

在 Linux 平台下安装 PHP 有几种方法：使用配置和编译过程，或是使用各种预编译的包。在 Linux 上安装软件，用户最好的选择是下载源代码包，并编译一个适合自己的版本。LAMP 组合中每个成员都是开源的软件，都可以从各自的官方网站上免费下载安装程序的源代码文件，并在自己的系统上编译，编译之前会检查系统的环境，并可以针对目标系统的环境进行优化。所以和自己系统的兼容性是最好的，不仅如此，允许你根据自己的需求进行定制安装。这是 LAMP 环境最理想的搭建方法，也是最复杂的安装方式。所以要搭建一个最完美的 LAMP 工作环境，多花费一些时间和精力在源代码包的安装上，还是值得的。

2.1 安装前准备

本书的源代码包安装方式，使用的 Linux 操作系统主要以 Red Hat Linux 操作系统系列为主。假设读者在电脑中已经安装好了 Red Hat 系列中的某一版本的操作系统，就可以按下面几种方式做安装前的准备工作了。

1. 获取软件包

安装之前，我们首先需要到相应的网站上，下载安装所需要的最新的软件源代码文件。包括最新的 Apache、MySQL、PHP 以及相关库文件的源码包。需要下载的软件和下载的地址如表 1 所示。

表 1 Linux 下 LAMP 环境安装所需要的源代码包列表及下载地址

软件名称	下载地址
httpd-2.2.9.tar.gz	http://www.apache.org/
mysql-5.0.41.tar.gz	http://dev.mysql.com/downloads/
php-5.2.6.tar.gz	http://www.php.net/downloads.php
phpMyAdmin-3.0.0-rc1-all-languages.tar.gz	http://www.phpmyadmin.com/
libxml2-2.6.30.tar.gz	ftp://ftp.gnome.org/pub/GNOME/sources/libxml2/2.6/libxml2-2.6.30.tar.gz
libmcrypt-2.5.8.tar.gz	http://prdownloads.sourceforge.net/mcrypt/libmcrypt-2.5.8.tar.gz?use_mirror=peterhost
zlib-1.2.3.tar.gz	http://www.zlib.net/zlib-1.2.3.tar.gz
gd-2.0.35.tar.gz	http://www.libgd.org/releases/gd-2.0.35.tar.gz
Autoconf-2.61.tar.gz	ftp://ftp.gnu.org/gnu/autoconf/autoconf-2.61.tar.gz
freetype-2.3.5.tar.gz	http://download.savannah.gnu.org/releases/freetype/freetype-2.3.5.tar.gz
libpng-1.2.31.tar.gz	ftp://ftp.simplesystems.org/pub/libpng/png/src/libpng-1.2.31.tar.gz
jpegsrc.v6b.tar.gz	http://www.ijg.org/files/jpegsrc.v6b.tar.gz

ZendOptimizer330a.tar.gz	http://wt1.mycodes.net/soft/20071108/ZendOptimizer330a.tar.gz
--------------------------	---

在表 1 中一共有 13 个 LAMP 环境安装需要的软件包，可以直接在浏览器的地址栏中输入下载地址进行下载或使用下载软件直接下载。读者也可以下载最新的软件来进行更新，并把下载的所有源代码包都放到 Linux 系统下的/usr/local/src/目录中。如下所示：

```
[root@localhost src]# ls /usr/local/src/ //显示/user/local/src/下的文件列表
autoconf-2.61.tar.gz      libxml2-2.6.30.tar.gz
freetype-2.3.5.tar.gz    mysql-5.0.41.tar.gz
gd-2.0.35.tar.gz         php-5.2.6.tar.gz
httpd-2.2.9.tar.gz       phpMyAdmin-3.0.0-rc1-all-languages.tar.gz
jpegsrc.v6b.tar.gz       ZendOptimizer330a.tar.gz
libmcrypt-2.5.8.tar.gz   zlib-1.2.3.tar.gz
libpng-1.2.31.tar.gz
[root@localhost src]# ls | wc -l //查看当前目录下的文件个数
13 //总计 13 个文件
[root@localhost src]#
```

2. 检查安装时使用的编译工具是否存在

我们下载的软件源代码文件都是使用 C 语言编写的，需要在本机编译后才能安装使用，所以在安装前首先要检查一下系统中编译工具 gcc 是否已经安装，可以在命令行中使用“gcc -v”来查看是否安装了 gcc。如下所示：

```
[root@localhost root]# gcc -v //查看是否安装过 gcc 编译程序
Reading specs from /usr/lib/gcc-lib/i386-redhat-linux/3.2.2/specs
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info
--enable-shared--enable-threads=posix --disable-checking --with-system-zlib
--enable-__cxa_atexit --host=i386-redhat-linux
Thread model: posix
gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5)
```

如果系统中已经安装了 gcc 编译工具，则显示上述信息，如果没有安装 gcc 工具，读者需要下载 gcc 工具软件并安装上。

3. 卸载默认的低版本环境

目前发行的 Linux 操作系统版本中，如果选择默认全部安装，就已经安装了 LAMP 环境，但是版本相对都比较低。我们可以再安装一个 LAMP 环境和原来的并存，但是这样做没有必要，因为同时只能开启一个 LAMP 环境。所以我们要在安装之前，先检查一下系统中是否已经安装了低版本的环境，如果已经安装过了，停止原来的服务运行，或者把原来的环境卸载掉。如下所示：

```
[root@localhost root]# rpm -qa|grep -i httpd //查询系统中已安装的 Apache 相关软件包
httpd-manual-2.0.40-21
httpd-2.0.40-21
redhat-config-httpd-1.0.1-18
[root@localhost root]# service httpd stop //如果 Apache 已开启，停止运行 Apache 服务器
[root@localhost root]# rpm -e httpd-manual-2.0.40-21 --nodeps //卸载 Apache 服务器
[root@localhost root]# rpm -e httpd-2.0.40-21 --nodeps //卸载 Apache 服务器
[root@localhost root]# rpm -e redhat-config-httpd-1.0.1-18 --nodeps //卸载 Apache 服务器

[root@localhost root]# rpm -qa|grep -i mysql //查询系统中已安装的 MySQL 相关软件包
mysql-devel-3.23.54a-11
mysql-3.23.54a-11
```

```
mysql-server-3.23.54a-11
[root@localhost root]# service mysqld stop //如果 MySQL 已开启, 停止运行 MySQL 服务器
[root@localhost root]# rpm -e mysql-devel-3.23.54a-11 --nodeps //卸载 MySQL 服务器
[root@localhost root]# rpm -e mysql-3.23.54a-11 --nodeps //卸载 MySQL 服务器
[root@localhost root]# rpm -e mysql-server-3.23.54a-11 --nodeps //卸载 MySQL 服务器

[root@localhost root]# rpm -qa|grep -i php //查询系统中已安装的 PHP 相关软件包
php-ldap-4.2.2-17
php-imap-4.2.2-17
php-4.2.2-17
[root@localhost root]# rpm -e php-ldap-4.2.2-17 --nodeps //卸载 PHP 应用服务器
[root@localhost root]# rpm -e php-imap-4.2.2-17 --nodeps //卸载 PHP 应用服务器
[root@localhost root]# rpm -e php-4.2.2-17 --nodeps //卸载 PHP 应用服务器
```

2.2 编译安装过程介绍

搭建 LAMP 环境时, 需要安装的所有软件都要按照一定的顺序安装, 我们按照 Apache->MySQL->PHP 顺序安装。但在安装 PHP 之前, 应先安装 PHP 5 需要的最新版本库文件, 例如 libxml2、libmcrypt, 以及 GD2 库等文件。安装 GD2 库是为了让 PHP 5 支持 GIF、PNG 和 JPEG 图片格式, 所以在安装 GD2 库之前还要先安装最新的 zlib、libpng、freetype 和 jpegsrc 等库文件。而且中间还会穿插安装一些软件。读者可以按照本节提供的顺序安装。

1. 解压 tar.gz 为后缀的压缩软件包

LAMP 环境搭建所需要的每个软件的源代码文件, 都是以 tar.gz 或 .tgz 提供给我们的打包压缩文件, 所以我们必须将其解压再解包。可以通过使用 Linux 操作系统的 SHELL 命令 tar, 再结合 “zxvf” 四个选项完成这个工作。使用 tar 的解压缩语法格式如下所示:

```
[root@localhost root]# tar zxvf tarfile.tar.gz //对压缩文件 tarfile.tar.gz 解压并打开包
```

2. 在 Linux 系统中源代码包安装过程

进行解压后的目录, LAMP 环境搭建所需要的软件都是使用 C 语言开发的, 所以安装源代码文件最少需要配置、编译和安装三个步骤:

➤ 配置 (configure)

每个软件的源代码目录中都会存在一个名为 configure 的脚本文件, 配置和安装过程被 configure 脚本中一系列命令行选项控制。每个软件包的配置方式是不同的, 所以可以在其源代码目录中, 通过 ./configure --help 命令了解所有可用的编译选项及简短解释。配置好后, 便可以开始编译模块。

➤ 编译 (make)

软件的配置过程成功完成后, 会在当前目录下生成一个 MAKEFILE 文件。可以通过 make 命令按 MAKEFILE 文件的配置进行编译, 编译成功后则可执行的二进制文件, 便可以开始进行软件安装了。

➤ 安装 (make install)

根据配置和编译过程, 在 Linux 命令行中通过执行 make install 命令, 将软件安装到指定的位置。

2.3 安装 libxml2 最新库文件

步骤一：进入下载的软件源码包所在目录/usr/local/src/下，解压软件包 libxml2-2.6.30.tar.gz 到当前目录 libxml2-2.6.30 下，并进入 libxml2-2.6.30 目录。命令行如下所示。

```
[root@localhost root]# cd /usr/local/src/           //进入软件源码包所在目录
[root@localhost src]# tar zxvf libxml2-2.6.30.tar.gz //解包解压到 libxml2-2.6.30 目录
[root@localhost src]# cd libxml2-2.6.30           //进入目录 libxml2-2.6.30 中
```

步骤二：使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件，命令行如下。

```
[root@localhost libxml2-2.6.30]# ./configure --prefix=/usr/local/libxml2
选项--prefix=/usr/local/libxml2 作用，是在安装时将软件安装到/usr/local/libxml2 目录下
```

步骤三：使用“make”命令编译源代码文件并生成安装文件。命令行如下。

```
[root@localhost libxml2-2.6.30]# make           //对软件源代码文件进行编译
```

步骤四：使用“make install”命令安装编译过的文件。命令行如下。

```
[root@localhost libxml2-2.6.30]# make install //开始安装 libxml2 库文件
```

如果安装成功以后，在/usr/local/libxml2/目录下将生成 bin、include、lib、man 和 share 五个目录。在后面安装 PHP5 源代码包的配置时，会通过 configure 命令的选项中加上“--with-libxml-dir=/usr/local/libxml2”选项，用于指定安装 libxml2 库文件的位置。

2.4 安装 libmcrypt 最新库文件

步骤一：再次进入软件源码包所在目录/usr/local/src/中，解压软件包 libmcrypt-2.5.8.tar.gz 到当前目录 libmcrypt-2.5.8 下，并进入 libmcrypt-2.5.8 目录。命令行如下所示。

```
[root@localhost root]# cd /usr/local/src/           //进入软件源码包所在目录
[root@localhost src]# tar zxvf libmcrypt-2.5.8.tar.gz //解压到 libmcrypt-2.5.8 目录
[root@localhost src]# cd libmcrypt-2.5.8           //进入目录 libmcrypt-2.5.8 中
```

步骤二：同样使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件。命令行如下。

```
[root@localhost libmcrypt-2.5.8]# ./configure --prefix=/usr/local/libmcrypt
选项--prefix=/usr/local/libmcrypt，是在安装时将软件安装到/usr/local/libmcrypt 目录下
```

步骤三：使用“make”命令编译源代码文件并生成安装文件，命令行如下。

```
[root@localhost libmcrypt-2.5.8]# make           //对软件源代码文件进行编译
```

步骤四：使用“make install”命令进行安装，命令行如下。

```
[root@localhost libmcrypt-2.5.8]# make install //开始安装 libmcrypt 库文件
```

如果安装成功就会在/usr/local/libmcrypt/目录下生成 bin、include、lib、man 和 share 五个目录。然后在安装 PHP 5 源代码包的配置时，就可以通过 configure 命令加上“--with-mcrypt-dir=/usr/local/libmcrypt”选项，指定这个 libmcrypt 库文件的位置。

步骤五：安装完成 libmcrypt 库以后，不同的 Linux 系统版本有可能还要安装一下 libltdl 库。安装方法和前面的步骤相同，可以进入到解压缩的目录/usr/local/src/libmcrypt-2.5.8 下，找到 libltdl 库源代码文件所在的目录 libltdl，进入这个目录按照下面几个命令配置、编译、安装就可以了。

```
[root@localhost root]# cd /usr/local/src/libmcrypt-2.5.8/libltdl //进入软件源代码目录
[root@localhost libltdl]# ./configure --enable-ltdl-install //配置 ltdl 库的安装
[root@localhost libltdl]# make //编译
[root@localhost libltdl]# make install //安装
```

2.5 安装 zlib 最新库文件

步骤一：进入软件源码包所在目录/usr/local/src/中，解压软件包 zlib-1.2.3.tar.gz 到当前目录 zlib-1.2.3 下，并进入 zlib-1.2.3 目录。命令如下所示。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
[root@localhost src]# tar zxvf zlib-1.2.3.tar.gz //解包解压到 zlib-1.2.3 目录
[root@localhost src]# cd zlib-1.2.3 //进入目录 zlib-1.2.3 中
```

步骤二：同样使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件。命令行如下。

```
[root@localhost zlib-1.2.3]# ./configure --prefix=/usr/local/zlib
选项--prefix=/usr/local/zlib，是在安装时将软件安装到/usr/local/zlib 目录下
```

步骤三：使用“make”命令编译源代码文件并生成安装文件。命令行输入如下。

```
[root@localhost zlib-1.2.3]# make //对软件源代码文件进行编译
```

步骤四：使用“make install”命令进行安装，命令行如下。

```
[root@localhost zlib-1.2.3]# make install //开始安装 zlib 库文件
```

如果安装成功将会在/usr/local/zlib 目录下生成 include、lib 和 share 三个目录。在安装 PHP 5 配置时，在 configure 命令的选项中加入“--with-zlib-dir=/usr/local/libmcrypt”选项，用于指定 zlib 库文件的位置。

2.6 安装 libpng 最新库文件

步骤一：进入软件源码包所在目录/usr/local/src/中，解压软件包 libpng-1.2.31.tar.gz 到当前目录 libpng-1.2.31 下，并进入 libpng-1.2.31 目录。命令行如下所示。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
[root@localhost src]# tar zxvf libpng-1.2.31.tar.gz //解包解压到 libpng-1.2.31 目录
[root@localhost src]# cd libpng-1.2.31 //进入目录 libpng-1.2.31 中
```

步骤二：同样使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件。命令行如下。

```
[root@localhost libpng-1.2.31]# ./configure --prefix=/usr/local/libpng
选项--prefix=/usr/local/libpng，是在安装时将软件安装到/usr/local/libpng 目录下
```

步骤三：使用“make”命令编译源代码文件并生成安装文件。命令行如下。

```
[root@localhost libpng-1.2.31]#make //对软件源代码文件进行编译
```

步骤四：使用“make install”命令进行安装。命令行如下。

```
[root@localhost libpng-1.2.31]# make install //开始安装 libpng 库文件
```

如果安装成功将会在/usr/local/libpng 目录下生成 bin、include、lib 和 share 四个目录。在安装 GD2 库配置时，通过在 configure 命令的选项中加入“--with-png=/usr/local/libpng”选项，指定 libpng 库文件的位置。

2.7 安装 jpeg6 最新库文件

步骤一：安装 GD2 库前所需的 jpeg6 库文件，需要自己手动地创建安装需要的目录，它们在安装时不能自动创建。命令行如下。

```
[root@localhost root]# mkdir /usr/local/jpeg6 //建立 jpeg6 软件安装目录
[root@localhost root]# mkdir /usr/local/jpeg6/bin //建立存放命令的目录
[root@localhost root]# mkdir /usr/local/jpeg6/lib //创建 jpeg6 库文件所在目录
[root@localhost root]# mkdir /usr/local/jpeg6/include //建立存放头文件目录
[root@localhost root]# mkdir -p /usr/local/jpeg6/man/man1 //建立存放手册的目录
```

步骤二：进入软件源码包所在目录/usr/local/src/中，解压软件包 jpegsrc.v6b.tar.gz 到当前目录 jpeg-6b 下，并进入 jpeg-6b 目录。命令行如下所示。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
[root@localhost src]# tar zxvf jpegsrc.v6b.tar.gz //解包解压到 jpeg-6b 目录
[root@localhost src]# cd jpeg-6b //进入目录 jpeg-6b 中
```

步骤三：使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件，命令行如下，使用“\”将一个命令换成多行。

```
[root@localhost jpeg-6b]# ./configure \
> --prefix=/usr/local/jpeg6/ \ //在安装时将软件安装到/usr/local/jpeg6 目录下
> --enable-shared \ //建立共享库使用的 GNU 的 libtool
> --enable-static //建立静态库使用的 GNU 的 libtool
```

步骤四：使用“make”命令编译源代码文件并生成安装文件，命令行如下。

```
[root@localhost jpeg-6b]# make //对软件源代码文件进行编译
```

步骤五：使用“make install”命令进行安装，命令行如下。

```
[root@localhost jpeg-6b]# make install //开始安装 jpeg6 库文件
```

在安装 GD2 库配置时，可以在 configure 命令的选项中加入“--with-jpeg=/usr/local/jpeg6/”选项，指定 jpeg6 库文件的位置。安装 PHP 时也要指定该库文件的位置。

2.8 安装 freetype 最新库文件

步骤一：进入软件源码包所在目录/usr/local/src/中，解压软件包 freetype-2.3.5.tar.gz 到当前目录 freetype-2.3.5 下，并进入 freetype-2.3.5 目录。命令行如下所示。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
```

```
[root@localhost src]# tar zxvf freetype-2.3.5.tar.gz //解包解压到 freetype-2.3.5 目录
[root@localhost src]# cd freetype-2.3.5 //进入目录 freetype-2.3.5 中
```

步骤二：使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件。命令行如下。

```
[root@localhost freetype-2.3.5]# ./configure --prefix=/usr/local/freetype
选项--prefix=/usr/local/freetype，是在安装时将软件安装到/usr/local/freetype 目录下
```

步骤三：使用“make”命令编译源代码文件并生成安装文件，命令行如下。

```
[root@localhost freetype-2.3.5]# make //对软件源代码文件进行编译
```

步骤四：使用“make install”命令进行安装，命令行如下：

```
[root@localhost freetype-2.3.5]# make install //开始安装 freetype 库文件
```

如果安装成功将会在/usr/local/freetype 目录下存在 bin、include、lib 和 share 4 个目录。并在安装 GD2 库时，通过 configure 命令的选项中加上“--with-freetype=/usr/local/freetype/”选项，指定 freetype 库文件的位置。

2.9 安装 autoconf 最新的库文件

步骤一：进入软件源码包所在目录/usr/local/src/中，解压软件包 autoconf-2.61.tar.gz 到当前目录 autoconf-2.61 下，并进入 autoconf-2.61 目录。命令行如下所示。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
[root@localhost src]# tar zxvf autoconf-2.61.tar.gz //解包解压到 autoconf-2.61 目录
[root@localhost src]# cd autoconf-2.61 //进入目录 autoconf-2.61 中
```

步骤二：使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件。命令行如下。

```
[root@localhost autoconf-2.61]# ./configure //配置
```

步骤三：使用“make”命令编译源代码文件并生成安装文件。命令行如下。

```
[root@localhost autoconf-2.61]# make //对软件源代码文件进行编译
```

步骤四：使用“make install”命令进行安装，命令行如下。

```
[root@localhost autoconf-2.61]# make install //开始安装 autoconf 库文件
```

2.10 安装最新的 GD 库文件

步骤一：进入软件源码包所在目录/usr/local/src/中，解压软件包 gd-2.0.35.tar.gz 到当前目录 gd-2.0.35 下，并进入 gd-2.0.35 目录。命令行如下所示。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
[root@localhost src]# tar zxvf gd-2.0.35.tar.gz //解包解压到 gd-2.0.35 目录
[root@localhost src]# cd gd-2.0.35 //进入目录 gd-2.0.35 中
```

步骤二：使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件，命令行如下，使用“\”将一个命令换成多行。

```
[root@localhost gd-2.0.35]# ./configure \ //配置命令
> --prefix=/usr/local/gd2/ \ //指定软件安装的位置
```



```
> --with-zlib=/usr/local/zlib/ \           //指定到哪去找 zlib 库文件的位置
> --with-jpeg=/usr/local/jpeg6/ \         //指定到哪去找 jpeg 库文件的位置
> --with-png=/usr/local/libpng/ \         //指定到哪去找 png 库文件的位置
> --with-freetype=/usr/local/freetype/    //指定到哪去找 freetype 2.x 字体库的位置
```

步骤三：使用“make”命令编译源代码文件并生成安装文件，命令行如下。

```
[root@localhost gd-2.0.35]# make           //对软件源代码文件进行编译
```

步骤四：使用“make install”命令进行安装，命令行如下。

```
[root@localhost gd-2.0.35]# make install  //开始安装 GD 库文件
```

如果安装成功会在/usr/local/gd2 目录下存在 bin、include 和 lib 这三个目录。在安装 PHP 5 时，通过在 configure 命令的选项中加上“--with-gd=/usr/local/gd2/”选项，指定 GD 库文件的位置。

2.11 安装新版本的 Apache 服务器

步骤一：进入软件源码包所在目录/usr/local/src/中，找到软件包 httpd-2.2.9.tar.gz 解压到当前目录 httpd-2.2.9 下，并进入 httpd-2.2.9 目录。命令行如下。

```
[root@localhost root]# cd /usr/local/src/           //进入软件源码包所在目录
[root@localhost src]# tar zxvf httpd-2.2.9.tar.gz    //解包解压到 httpd-2.2.9 目录
[root@localhost src]# cd httpd-2.2.9                //进入目录 httpd-2.2.9 中
```

步骤二：使用“configure”命令检查并配置安装需要的系统环境，并生成安装配置文件，命令行如下，使用“\”将一个命令换成多行。

```
[root@localhost httpd-2.2.9]# ./configure \          //执行当前目录下软件自带的配置命令
> --prefix=/usr/local/apache2 \                    //指定 Apache 软件安装的位置
> --sysconfdir=/etc/httpd \                          //指定 Apache 服务器的配置文件存放位置
> --with-z=/usr/local/zlib/ \                        //指定 zlib 库文件的位置
> --with-included-apr \                              //使用捆绑 APR/APR-Util 的副本
> --disable-userdir \                                //请求的映像到用户特定目录
> --enable-so \                                       //以动态共享对象(DSO)编译
> --enable-deflate=shared \                          //缩小传输编码的支持
> --enable-expire=shared \                          //期满头控制
> --enable-rewrite=shared \                         //基于规则的 URL 操控
> --enable-static-support \                         //建立一个静态链接版本的支持
```

步骤三：使用“make”命令编译源代码文件并生成安装文件，命令行如下。

```
[root@localhost httpd-2.2.9]# make                 //对 Apache 源代码文件进行编译
```

步骤四：使用“make install”命令进行安装，命令行如下。

```
[root@localhost httpd-2.2.9]# make install          //开始安装 Apache 服务器软件
```

步骤五：检查安装的文件，进入到/usr/local/apache2/目录下，确认是否有以下目录。

```
[root@localhost httpd-2.2.9]# cd /usr/local/apache2/
[root@localhost apache2]# ls
bin      cgi-bin  htdocs  include  logs  manual
build    error   icons   lib       man   modules
```

检查配置文件是否指定正确，进入到/etc/httpd/目录下，查看配置文件 httpd.conf 和 extra 子目录是

否存在。

步骤六：启动 Apache 服务器，并查看端口是否开启，启动 Apache 服务器的命令行如下。

```
[root@localhost apache2]# /usr/local/apache2/bin/apachectl start //启动 Apache
```

步骤七：Apache 服务器启动之后，查看一下 80 端口是否打开，如果看到以下结果表明 Apache 服务器启动成功。命令行如下。

```
[root@localhost apache2]# netstat -tnl|grep 80 //查看 80 端口是否开启
tcp        0      0 0.0.0.0:80      0.0.0.0:*      LISTEN
```

步骤八：测试安装结果，打开浏览器，在地址栏内输入 URL 为 `http://localhost/` 去访问 Apache 服务器。如出现如图 1 所示的内容表示 Aapache 服务器可以使用。

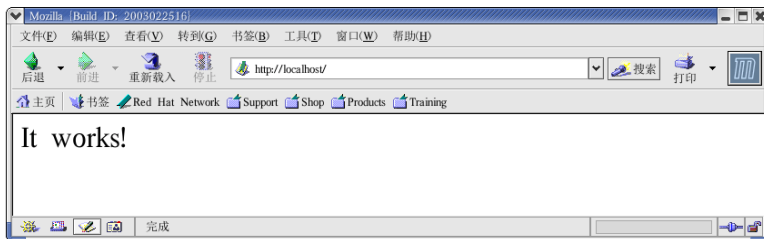


图 1 测试 Apache 是否安装并启动成功

步骤九：每种服务器软件都有必要制作成开机时自启动，Apache 服务器开机自启动，只要在“`/etc/rc.d/rc.local`”文件中，加上 Apache 服务器的启动命令即可。可以直接打开“`/etc/rc.d/rc.local`”文件，在最后一行写入 Aapache 启动命令，也可以使用“`echo`”命令追加进去，命令行如下。

```
[root@localhost root]# echo "/usr/local/apache2/bin/apachectl start" >> /etc/rc.d/rc.local
```

2.12 安装 MySQL 数据库管理系统

步骤一：首先要为 `mysqld` 增加一个登录用户和用户组，用户名和组名都为“`mysql`”，如果将这些用户和组命名为 `mysql` 之外的名称，在下面的步骤中也需要替换为相应的名称。命令行如下。

```
[root@localhost root]# groupadd mysql //添加一个 mysql 标准组
[root@localhost root]# useradd -g mysql mysql //添加 mysql 用户并加到 mysql 组中
```

步骤二：进入软件源码包所在目录 `/usr/local/src/` 中，解压软件包 `mysql-5.0.41.tar.gz` 到当前目录 `mysql-5.0.41` 下，并进入 `mysql-5.0.41` 目录。命令行如下。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
[root@localhost src]# tar zxvf mysql-5.0.41.tar.gz //解包解压到 mysql-5.0.41 目录
[root@localhost src]# cd mysql-5.0.41 //进入目录 mysql-5.0.41 中
```

步骤三：使用“`configure`”命令检查并配置安装需要的系统环境，并生成安装配置文件，命令行如下，使用“`\`”将一个命令换成多行。

```
[root@localhost mysql-5.0.41]# ./configure \
> --prefix=/usr/local/mysql \ //是在安装时将软件安装到/usr/local/mysql 目录下
> --with-extra-charsets=all //在安装 mysql 时安装所有字符集
```

步骤四：使用“`make`”命令编译源代码文件并生成安装文件，命令行如下。

```
[root@localhost mysql-5.0.41]# make //对 MySQL 源代码文件进行编译
```

步骤五：使用“make install”命令进行安装，命令行如下。

```
[root@localhost mysql-5.0.41]## make install //开始安装 MySQL 数据库系统
```

步骤六：创建 MySQL 数据库服务器的配置文件，可以使用源码包 support-files 目录中的 my-medium.cnf 文件作为模板，将其复制到/etc/目录下，命名为“my.cnf”文件即可。命令行如下。

```
[root@localhost mysql-5.0.41]# cp support-files/my-medium.cnf /etc/my.cnf
```

步骤七：如果还没有安装过 MySQL，必须创建 MySQL 授权表。进入到安装目录“/usr/local/mysql”下，执行 bin 目录下的 mysql_install_db 脚本，用来初始化 MySQL 数据库的授权表，其中存储了服务器访问允许。命令行如下。

```
[root@localhost mysql-5.0.41]# cd /usr/local/mysql //建入安装目录
```

```
[root@localhost mysql]# bin/mysql_install_db --user=mysql //创建授权表
```

如果使用 root 用户运行上面的命令，应当使用--user 选项，选项的值应与你在第一步为运行服务器所创建的登录账户（mysql 用户）相同。如果用 mysql 用户登录来运行上面命令，可以省略--user 选项。用 mysql_install_db 创建 MySQL 授权表后，需要手动重新启动服务器。

步骤八：将程序二进制的所有权改为 root 用户，数据目录的所有权改为运行 mysqld 程序的 mysql 用户。如果现在位于安装目录（/usr/local/mysql）下，命令行如下。

```
[root@localhost mysql]# chown -R root . //将文件的所有属性改为 root 用户
[root@localhost mysql]# chown -R mysql var //将数据目录的所有属性改为 mysql 用户
[root@localhost mysql]# chgrp -R mysql . //将组属性改为 mysql 组
[root@localhost mysql]# ls -l //长格式显示当前目录下的内容
```

```
总用量 40
drwxr-xr-x  2 root    mysql    4096  9 月 12 00:35 bin
drwxr-xr-x  3 root    mysql    4096  9 月 12 00:34 include
drwxr-xr-x  2 root    mysql    4096  9 月 12 00:34 info
drwxr-xr-x  3 root    mysql    4096  9 月 12 00:34 lib
drwxr-xr-x  2 root    mysql    4096  9 月 12 00:35 libexec
drwxr-xr-x  4 root    mysql    4096  9 月 12 00:35 man
drwxr-xr-x  8 root    mysql    4096  9 月 12 00:35 mysql-test
drwxr-xr-x  3 root    mysql    4096  9 月 12 00:34 share
drwxr-xr-x  5 root    mysql    4096  9 月 12 00:35 sql-bench
drwx----- 4 mysql   mysql    4096  9 月 12 01:16 var
```

步骤九：在所需要的东西被安装完成后，应当使用下面的命令启动 MySQL 服务了，命令行如下。

```
[root@localhost mysql]# /usr/local/mysql/bin/mysqld_safe --user=mysql &
```

步骤十：MySQL 数据库服务启动之后，查看一下它的端口 3306 是否打开，如果看到以下结果表明 MySQL 服务启动成功。命令行如下。

```
[root@localhost apache2]# netstat -tnl|grep 3306 //查看 3306 端口是否开启
tcp        0      0 0.0.0.0:3306      0.0.0.0:*        LISTEN
```

步骤十一：使用 mysqladmin 验证服务器在运行中。以下命令提供了简单的测试，可检查服务器是否已经启动并能响应连接。命令行如下。

```
[root@localhost mysql]# bin/mysqladmin version
bin/mysqladmin  Ver 8.41 Distrib 5.0.41, for pc-linux-gnu on i686
```

Copyright (C) 2000-2006 MySQL AB

This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license

```
Server version      5.0.41-log
Protocol version    10
Connection          Localhost via UNIX socket
UNIX socket         /tmp/mysql.sock
Uptime:             15 min 41 sec
```

```
Threads: 1  Questions: 25  Slow queries: 0  Opens: 12  Flush tables: 1  Open tables: 6  Queries per second avg: 0.027
[root@localhost mysql]# bin/mysqladmin variables //查看所有 mysql 参数
```

步骤十二：设置访问权限，在 MySQL 安装过程中，使用 `mysql_install_db` 程序安装了 MySQL 数据库授权表，表定义了初始 MySQL 用户账户和访问权限，所有初始账户均没有密码。这些账户为超用户账户，可以执行任何操作。初始 `root` 账户的密码为空，因此任何人可以用 `root` 账户不用任何密码来连接 MySQL 服务器，并具有所有权限，这意味着 MySQL 安装未受保护。如果你想要防止客户端不使用密码用匿名用户来连接，你应当为匿名账户指定密码或删除匿名账户，应当为 MySQL `root` 账户指定密码。使用 “`mysql -u root`” 启动 MySQL 客户端控制台，连接 MySQL 服务器。命令行如下。

```
[root@localhost mysql]# bin/mysql -u root //没有密码可以直接登录本机服务器
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.41-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

步骤十三：如果有匿名账户存在，它拥有全部的权限，因此删掉它可以提高安全，在 MySQL 客户端执行 SQL 语句如下。

```
mysql> DELETE FROM mysql.user WHERE Host='localhost' AND User='';
Query OK, 1 rows affected (0.08 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 1 rows affected (0.01 sec)
```

步骤十四：可以用几种方法为 `root` 账户指定密码，我们选用其中一种。在 MySQL 客户端命令行上使用 `SET PASSWORD` 指定密码，一定要使用 `PASSWORD()` 函数来加密密码。例如下面设置 `localhost` 域的密码为 “123456”。其他域可以使用同样的语句，使用的 SQL 语句如下。

```
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('123456');
Query OK, 0 rows affected (0.00 sec)
```

步骤十五：如果想退出 MySQL 客户端，可以在 MySQL 客户端提示符下输入命令 “`exit`” 或者 “`quit`”，还可以按键盘 “`Ctrl+C`” 组合键，都可以从 MySQL 客户端中退出。因为已经给 MySQL 服务器的 `root` 账号设置了密码，所以再次登录 MySQL 客户端就要提供密码才能进入。退出 MySQL 客户端和重新启动 MySQL 客户端的控制台命令如下。

```
mysql> exit //退出 MySQL 客户端
Bye
[root@localhost mysql]# bin/mysql -u root -h localhost -p //回车进入 MySQL 客户端
Enter password: //输入密码 “123456”
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.0.41-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

如果想关闭 MySQL 服务器，在命令行使用 MySQL 服务器的 `mysqladmin` 命令，通过 `-u` 参数给出 MySQL 数据库管理员用户名 `root` 和通过 `-p` 参数给出密码，即可关闭 MySQL 服务器。命令行如下。

```
[root@localhost mysql]# bin/mysqladmin -u root -p shutdown //关闭 MySQL 数据库
```

步骤十六：MySQL 服务器和 Apache 服务器一样也有必要设置为开机自动运行，设置方法是进入到 `mysql` 源代码目录 `/usr/local/src/mysql-5.0.41/` 中，将子目录 `support-files` 下的 `mysql.server` 文件复制到 `/etc/rc.d/init.d` 目录中，并重命名为“`mysqld`”，命令行如下。

```
[root@localhost mysql]# cd /usr/local/src/mysql-5.0.41
[root@localhost mysql-5.0.41]# cp support-files/mysql.server /etc/rc.d/init.d/mysqld
```

修改文件 `/etc/rc.d/init.d/mysqld` 的权限，命令行如下。

```
[root@localhost mysql-5.0.41]# chown root.root /etc/rc.d/init.d/mysqld
[root@localhost mysql-5.0.41]# chmod 755 /etc/rc.d/init.d/mysqld
```

使用 `chkconfig` 命令设置在不同系统运行级别下的自启动策略，首先使用“`chkconfig --add mysqld`”命令增加所指定的 `mysqld` 服务，让 `chkconfig` 指令得以管理它，并同时在系统启动的叙述文件内增加相关数据。使用命令如下。

```
[root@localhost mysql-5.0.41]# chkconfig --add mysqld
```

然后使用“`chkconfig --level 3 mysqld on`”命令和“`chkconfig --level 5 mysqld on`”命令，在第三等级和第五等级中开启 `mysqld` 服务，即在字符模式和图形模式启动时自动开启 `mysqld` 服务。命令如下。

```
[root@localhost mysql-5.0.41]# chkconfig --level 3 mysqld on
[root@localhost mysql-5.0.41]# chkconfig --level 5 mysqld on
```

再使用“`chkconfig--list`”命令检查设置。命令行如下。

```
[root@localhost mysql-5.0.41]# chkconfig --list mysqld
mysqld          0:关闭 1:关闭 2: 关闭 3:启用 4: 关闭 5:启用 6:关闭
```

2.13 安装最新版本的 PHP 模块

步骤一：进入软件源码包所在目录 `/usr/local/src/` 中，解压软件包 `php-5.2.6.tar.gz` 到当前目录 `php-5.2.6` 下，并进入 `php-5.2.6` 目录。命令行如下。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
[root@localhost src]# tar zxvf php-5.2.6.tar.gz //解包解压到 php-5.2.6 目录
[root@localhost src]# cd php-5.2.6 //进入目录 php-5.2.6 中
```

步骤二：使用“`configure`”命令检查并配置安装需要的系统环境，并生成安装配置文件，命令行如下，使用“`\`”将 `configure` 命令选项换成多行。

```
[root@localhost php-5.2.6]# ./configure \ //执行当前目录下软件自带的配置命令
> --prefix=/usr/local/php \ //设置 PHP5 的安装路径
```

```

> --with-config-file-path=/usr/local/php/etc \           //指定 PHP5 配置文件存入的路径
> --with-apxs2=/usr/local/apache2/bin/apxs \           //告诉 PHP 查找 Apache 2 的地方
> --with-mysql=/usr/local/mysql/ \                     //指定 MySQL 的安装目录
> --with-libxml-dir=/usr/local/libxml2/ \              //告诉 PHP 放置 libxml2 库的地方
> --with-png-dir=/usr/local/libpng/ \                  //告诉 PHP 放置 libpng 库的地方
> --with-jpeg-dir=/usr/local/jpeg6/ \                  //告诉 PHP 放置 jpeg 库的地方
> --with-freetype-dir=/usr/local/freetype/ \           //告诉 PHP 放置 freetype 库的地方
> --with-gd=/usr/local/gd2/ \                          //告诉 PHP 放置 gd 库的地方
> --with-zlib-dir=/usr/local/zlib/ \                   //告诉 PHP 放置 zlib 库的地方
> --with-mcrypt=/usr/local/libmcrypt/ \               //告诉 PHP 放置 libmcrypt 库的地方
> --with-mysqli=/usr/local/mysql/bin/mysqli_config \   //变量激活新增加的 MySQLi 功能
> --enable-soap \                                       //变量激活 SOAP 和 Web services 支持
> --enable-mbstring=all \                              //使多字节字符串支持
> --enable-sockets                                     //变量激活 socket 通信特性

```

步骤三：使用“make”命令编译源代码文件并生成安装文件。命令行如下。

```
[root@localhost php-5.2.6]# make //对 PHP 源代码文件进行编译
```

步骤四：使用“make install”命令进行安装。命令行如下。

```
[root@localhost php-5.2.6]# make install //开始安装 PHP5
```

步骤五：安装完成后，需要建立 PHP 配置文件。在使用 configure 命令安装配置时使用了“--with-config-file-path=/usr/local/php/etc/”选项，指定了配置文件的位置。将源码包目录下的“php.ini-dist”文件复制到指定的目录“/usr/local/php/etc/”中，并改名为“php.ini”即可。命令行如下。

```
[root@localhost php-5.2.6]# cp php.ini-dist /usr/local/php/etc/php.ini //创建配置文件
```

步骤六：整合 Apache 与 PHP，上面 PHP 编译之前，我们使用 configure 命令安装配置时，使用了“--with-apxs2=/usr/local/apache2/bin/apxs”选项以使 Apache 2 将 PHP 作为功能模块使用。但我们还需要修改 Apache 配置文件，添加 PHP 的支持，告诉 Apache 将哪些后缀作为 PHP 解析。例如，让 Apache 把.php 或.phtml 后缀的文件解析为 PHP。使用 vi 打开 Apache 的配置文件/etc/httpd/httpd.conf，找到“AddType application/x-gzip .gz .tgz”指令项，并在其下方添加一条指令“Addtype application/x-httpd-php.php.phtml”。也可以将任何后缀的文件解析为 PHP，只要在添加的语句中加入并用空格分开，这里以多添加一个.phtml 来示例。如下所示。

```

[root@localhost php-5.2.6]# vi /etc/httpd/httpd.conf //使用 vi 编辑 apache 配置文件
... ..
# If the AddEncoding directives above are commented-out, then you
# probably should define those extensions to indicate media types:
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz                                //在这行下面添加

Addtype application/x-httpd-php.php.phtml #添加这一条
#添加这一条是为了将 .phps 作为 PHP 的源文件进行语法高亮显示
Addtype application/x-httpd-php-source .phps
# AddHandler allows you to map certain file extensions to "handlers":
# actions unrelated to filetype. These can be either built into the server
# or added with the Action directive (see below)
... ..

```

步骤七：修改完成以后必须重新启动 Apache 服务器，才能重新加载配置文件使修改生效。命令如

下所示。

```
[root@localhost php-5.2.6]# /usr/local/apache2/bin/apachectl stop //停止 Apache 服务
[root@localhost php-5.2.6]# /usr/local/apache2/bin/apachectl start //开启 Apache 服务
```

步骤八：测试 PHP 环境是否可以正常运行，在/usr/local/apache2/htdocs 目录下建一个 test.php 或 test.phtml 的文件。内容如下所示。

```
[root@localhost htdocs]# vi test.php //编辑 test.php 文件
<?php
    phpinfo();
?>
```

打开浏览器，在地址栏中输入 URL 为 http://localhost/test.php 来运行该文件，如果出现如图 2 所示的内容表示 LAMP 环境安装成功。

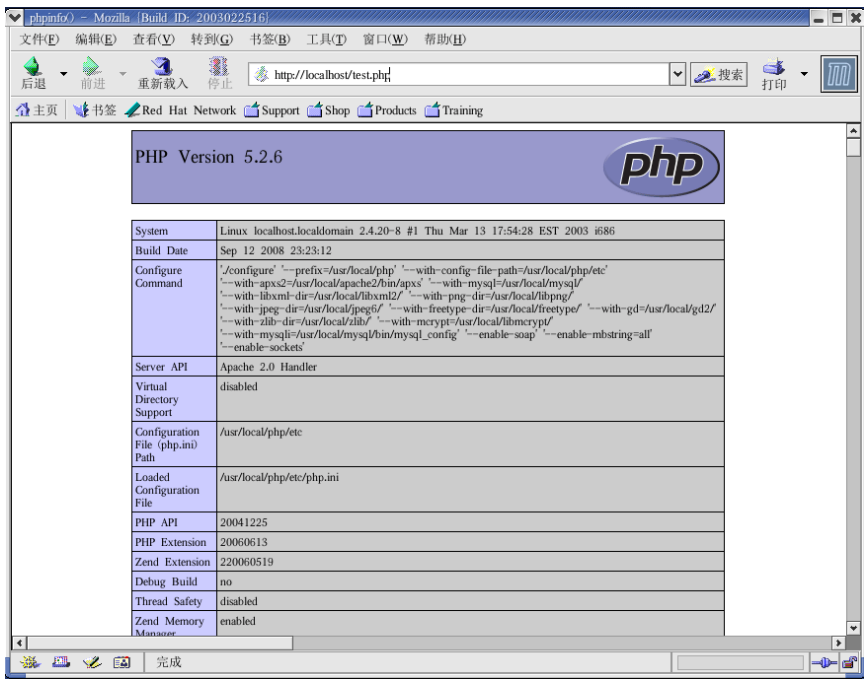


图 2 测试 PHP 是否安装并启动成功

上例中使用了 phpinfo()函数，作用是输出有关 PHP 当前状态的大部分信息内容，这包括关于 PHP 的编译和扩展信息、PHP 版本、服务器信息和环境、PHP 的环境、操作系统信息、路径、主要的和本地配置选项的值、HTTP 头信息和 PHP 的许可等。因为每个系统的安装不同，phpinfo()函数可以用于检查某一特定系统配置设置和可用的预定义变量等。它也是一个宝贵的调试工具，因为它包含了所有 EGPCS（Environment，GET，POST，Cookie，Server）数据。

2.14 安装 Zend 加速器

通过上面几节的操作已经搭建好 LAMP 环境了，为了提高 PHP 程序的运行速度，最好还是安装一个 Zend 加速器(Zend Optimizer)。Zend Optimizer 用优化代码的方法来提高 PHP 应用程序的执行速度。

实现的原理是对那些在被最终执行之前由运行编译器（Run-Time Compiler）产生的代码进行优化。一般情况下，执行使用 Zend Optimizer 的 PHP 程序比不使用的要快 40% 到 100%。这意味着网站的访问者可以更快地浏览你的网页，从而完成更多的事务，创造更好的客户满意度。更快的反应同时也意味着可以节省硬件投资，并增强网站所提供的服务。安装的步骤如下：

步骤一：进入软件源码包所在目录/usr/local/src/中，解压软件包 ZendOptimizer330a.tar.gz 到当前目录 ZendOptimizer-3.3.0a-linux-glibc21-i386 下，并进入该目录。命令行如下。

```
[root@localhost root]# cd /usr/local/src/ //进入软件源码包所在目录
[root@localhost src]# tar zxvf ZendOptimizer330a.tar.gz //解包解压
[root@localhost src]# cd ZendOptimizer-3.3.0a-linux-glibc21-i386 //进入解压目录
```

步骤二：直接执行目录下的 install.sh 文件安装。命令行如下。

```
[root@localhost ZendOptimizer-3.3.0a-linux-glibc21-i386]# ./install.sh //执行安装
```

执行上面的操作会出现一个图形安装界面，按提示安装即可，部分安装过程如图 3 至图 6 所示。

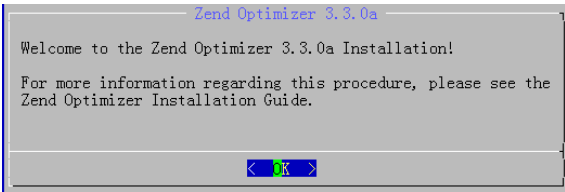


图 3 ZendOptimizer 安装的欢迎界面

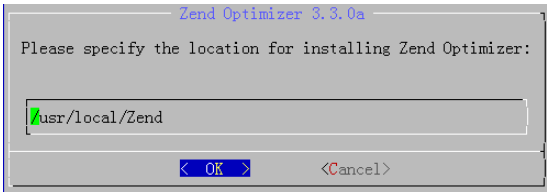


图 4 ZendOptimizer 选择软件安装位置

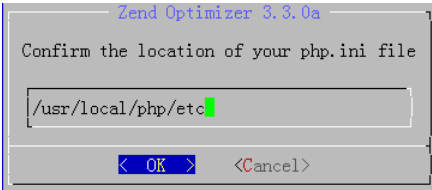


图 5 ZendOptimizer 选择 PHP 配置文件（php.ini）所在位置

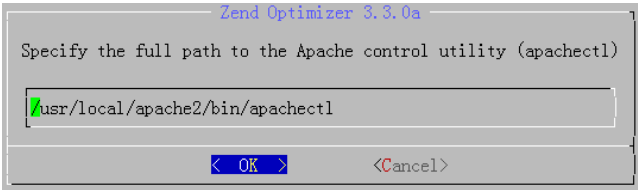


图 6 ZendOptimizer 选择 Apache 服务器启动命令所在位置

步骤三：安装完成以后同样使用 phpinfo()函数可以检查安装结果，如果你能在输出的 Zend 部分找到像下面的输出，即安装成功，如图 7 所示。



图 7 查看 ZendOptimizer 安装结果

2.15 phpMyAdmin 的安装

phpMyAdmin 是使用 PHP 脚本编写的一个 MySQL 系统管理软件，是最受欢迎的 MySQL 系统管理工具。安装该工具后，即可以通过 Web 形式直接管理 MySQL 数据，而不需要通过执行系统命令来管理，非常适合对数据库操作命令不熟悉的数据库管理者。它可以用来创建、修改、删除数据库和数据表；可以用来创建、修改、删除数据记录；可以用来导入和导出整个数据库；还可以完成许多其他的 MySQL 系统管理任务。

与其他的 PHP 程序一样，是一个 B/S 结构的软件，phpMyAdmin 软件需要在 Web 服务器上运行，因此它可以从互联网的 anywhere 访问操作。通常搭建的 MySQL 数据库服务器为了数据安全，只允许 localhost 域才能够操作，不允许远程连接访问，所以管理员在本机安装 phpMyAdmin 软件，就可以使用浏览器在远程登录管理 MySQL 数据库服务器了。安装的步骤如下：

步骤一：进入软件源码包所在目录 /usr/local/src/ 中，并解压软件包 phpMyAdmin-3.0.0-rc1-all-languages.tar.gz 到当前目录 phpMyAdmin-3.0.0-rc1-all-languages 下。命令行如下。

```
[root@localhost root]# cd /usr/local/src/           //进入软件源码包所在目录
[root@localhost src]# tar zxvf phpMyAdmin-3.0.0-rc1-all-languages.tar.gz
```

步骤二：把解压的目录 phpMyAdmin-3.0.0-rc1-all-languages 下的文件，全部复制到 Apache 的 DocumentRoot 目录下的某个子目录里。根据我们前面 Apache 的安装配置，复制到目录 /usr/local/apache2/htdocs 下，并新建一个名为“phpmyadmin”的目录下面，即安装完成。命令行如下。

```
[root@localhost src]# cp -a phpMyAdmin-3.0.0-rc1-all-languages \
/usr/local/apache2/htdocs/phpmyadmin           //拷贝目录到指定位置并改名为 phpmyadmin
```

步骤三：在使用 phpMyAdmin 之前，也需要先配置一下。配置的方法是通过 phpMyAdmin 顶层目录下的 config.inc.php 文件中的几个选项做一些设置即可。默认不存在 config.inc.php 文件，我们需要手工创建一个，也可以复制 config.sample.inc.php 模板得到最低限度的配置文件。下面的示例是对 config.inc.php 文件配置，只给出了必须要修改的部分，根据实际情况，也许还需要其他一些选项做出修改。创建 config.inc.php 配置文件命令行如下。

```
[root@localhost src]# cd /usr/local/apache2/htdocs/phpmyadmin/
[root@localhost phpmyadmin]# cp config.sample.inc.php config.inc.php
```

2.16 phpMyAdmin 的配置

我们通过身份验证模式的要求，可以有两种配置方案，一种是 HTTP 和 cookie 身份验证模式。在这两种模式下，用户必须先在一个登录窗口里输入 MySQL 数据库的有效用户名和密码，才能使用 phpMyAdmin 程序。这种做法有两个明显的好处：首先，因为 MySQL 数据库的密码没有出现在 config.inc.php 文件里，所以身份验证过程更加安全；其次，允许以不同的用户身份登录对自己的数据库进行管理。这两种身份验证模式尤其适合数据库中多个用户账号的情况。

第二种方案是，config 身份验证模式。这种模式下，密码以明文形式保存在 config.inc.php 文件里。只需要把 MySQL 用户名和密码直接写入到 config.inc.php 文件即可。这样，在登录 phpMyAdmin 时就不会提示输入用户名和密码了，而只直接用 config.inc.php 文件里写入的用户登录。如果只是在一个本地测试系统上使用 phpMyAdmin，可以使用这种模式。

1. HTTP 身份验证模式

如果想让 phpMyAdmin 使用 HTTP 身份验证模式，首先需要在 config.inc.php 文件里黑体部分做出如下所示的修改。具体内容如下：

```
[root@localhost phpmyadmin]# vi config.inc.php
....
$cfg['blowfish_secret'] = '';
$i = 0;
$i++;
$cfg['Servers'][$i]['auth_type'] = 'http'; //只将这一行修改成 HTTP 身份验证模式即可
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['extension'] = 'mysql';
....
```

当完成设置之后，我们启动 phpMyAdmin 时，屏幕上将弹出一个 Web 浏览器对话框，需要在这个对话框里输入 MySQL 用户名和密码，才能进入 phpMyAdmin 操作界面。如图 8 所示，在 Windows 客户端使用 IE 浏览器，访问 Web 服务器的 phpMyAdmin 目录下的 index.php 文件，即启动了 phpMyAdmin。

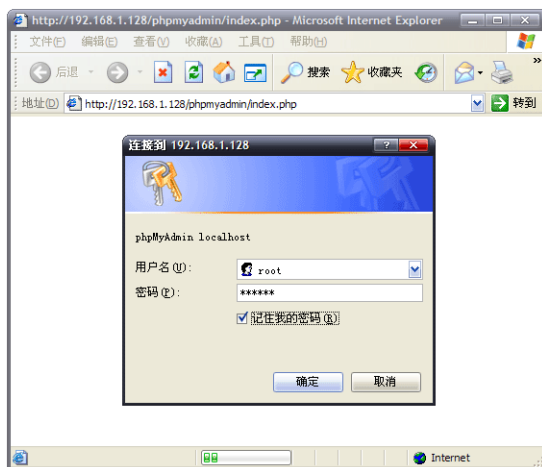


图 8 以 HTTP 身份验证模式登录 phpMyAdmin

2. cookie 身份验证模式

cookie 身份验证模式是 HTTP 身份验证模式的补充，不能使用 HTTP 身份验证模式的地方都可以使用它。cookie 身份验证模式要求用户必须允许来自 phpMyAdmin 的 cookie 进入自己的计算机。即用户需要在浏览器中开启客户端的 cookie 功能。

如果想让 phpMyAdmin 使用 cookie 身份验证模式，除了必须修改 config.inc.php 文件里的 auth_type 语句外，还必须向 blowfish_secret 参数提供一个字符串。这个字符串可以是任意的，目的是在把登录时使用的用户和密码存储在客户端电脑上的 cookie 之前，系统将会使用这个字符串对它们进行加密。在 config.inc.php 中修改的内容如下：

```
[root@localhost phpmyadmin]# vi config.inc.php
....
$cfg['blowfish_secret'] = "xxxxxxx"; //这里需要一个任意的字符串
$i = 0;
```

```

$i++;
$cfg['Servers'][$i]['auth_type'] = 'cookie'; //这条修改成 cookie 身份验证模式
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['extension'] = 'mysql';
... ..

```

和上面启动 phpMyAdmin 的方式一样，我们在 Windows 客户端使用 IE 浏览器，访问 Web 服务器上的 phpMyAdmin 目录下的 index.php 文件，需要提供 MySQL 的用户名和密码才能登录，如图 9 所示。



图 9 以 Cookie 身份验证模式登录 phpMyAdmin

3. config 身份验证模式

如果想让 phpMyAdmin 使用 config 身份验证模式，首先需要在 config.inc.php 文件里做出如下所示的修改。把 MySQL 数据库的用户名和密码以明文的方式写入，具体修改内容如下：

```

[root@localhost phpmyadmin]# vi config.inc.php
... ..
$i = 0;
$i++;
$cfg['Servers'][$i]['auth_type'] = 'config'; //这条修改成 config 身份验证模式
$cfg['Servers'][$i]['host'] = 'localhost';
//添加以下两个选项
$cfg['Servers'][$i]['user'] = 'root'; //使用你 MySQL 数据库的用户名
$cfg['Servers'][$i]['password'] = '123456'; //使用你 MySQL 数据库的密码
... ..

```

和上面启动 phpMyAdmin 的方式一样，我们在 Windows 客户端使用 IE 浏览器，访问 Web 服务器上的 phpMyAdmin 目录下的 index.php 文件。但不用提供 MySQL 的用户名和密码就可登录，它是使用 config.inc.php 中以明文方式写入的用户名和密码登录的，如图 10 所示。

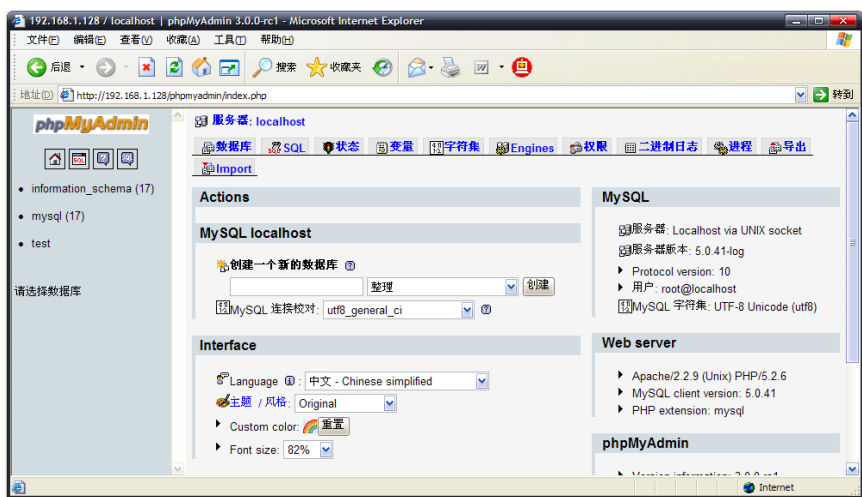


图 10 以 config 身份验证模式登录 phpMyAdmin

如图 10 所示，直接就可以登录 phpMyAdmin 操作 MySQL 数据库里的数据，这种模式不够安全，所以只适合在一个本地测试系统上使用。

3 Windows 系统

上一节详细介绍了 UNIX/Linux 系统上以源代码包的方式安装 Web 工作环境，虽然上一节安装的环境是最好的 Web 工作环境。但大多数读者对 Linux 系统并不熟悉，所以这一节主要介绍如何在 Windows 系统上分别独立安装 Apache2、PHP5、MySQL5 和 phpMyAdmin 等几个软件。独立安装的好处是可以自由选择这些组件的具体版本，清晰地掌握自己计算机里都安装了哪些程序，以及它们的具体配置情况，这将对以后的系统维护和软件升级工作带来很大的帮助。

3.1 安装前准备

在 Windows 系统下安装 Apache 服务器、PHP 和 MySQL 服务器，方法比较简单。以下安装方法同时适合于 Windows NT、Windows2000、Windows2003 及 Windows XP 等操作系统。本书主要以 Windows XP 系统为例，在安装之前，我们需要准备相应的最新版本的软件，需要下载的软件和下载的地址如表 2 所示。

表 2 Windows 下 Web 环境安装所需要的源代码包列表及下载地址

软件名称	下载地址
Apache_2.2.9-win32-x86-openssl-0.9.8h-r2.msi	http://httpd.apache.org/download.cgi
mysql-5.1.28-rc-win32.zip	http://dev.mysql.com/downloads/
php-5.2.6-Win32.zip	http://www.php.net/downloads.php
ZendOptimizer-3.3.0a-Windows-i386.exe	http://www.zend.com/en/downloads/
phpMyAdmin-3.0.0-rc1-all-languages.zip	http://www.phpmyadmin.net/home_page/downloads.php

在 Windows 系统的 C 盘上创建一个名为 lamp 的文件夹，作为 Web 环境的安装位置，将环境需要的所有软件都安装到这个文件夹下面。

3.2 安装 Apache

在 Windows 系统上安装 Apache 服务器非常容易，和其他的 Windows 软件安装方法相同，通过单击“下一步”的方法就可以安装成功。以下给出了 Apache 软件的详细安装步骤，具体安装步骤和安装配置如下：

步骤一：进入到软件的下载文件夹，找到“apache_2.2.9-win32-x86-openssl-0.9.8h-r2.msi”软件，或者是下载更高版本的 Apache 软件。双击这个文件就可以启动安装程序。屏幕上出现一个对话框，是软件安装向导的欢迎界面，直接单击“Next>”按钮即可到下一步安装，如图 11 所示。

步骤二：通过上一步的安装出现以下界面，确认同意软件安装使用许可条例，选择“I accept the terms in the license agreement”，单击“Next>”按钮继续，如图 12 所示。

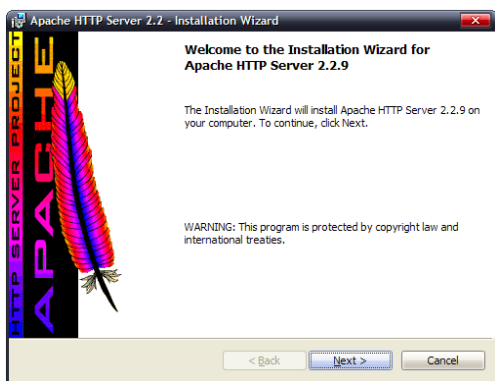


图 11 Apache 安装向导的欢迎界面

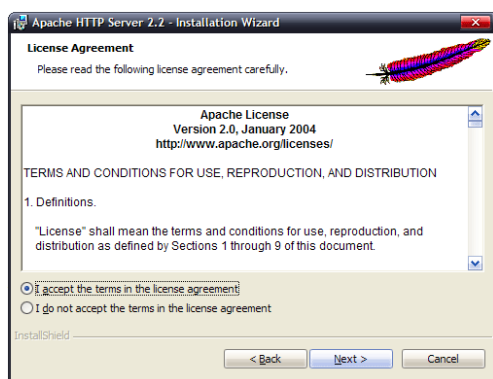


图 12 “确认同意软件安装使用许可条例”对话框

步骤三：弹出“Apache 安装到 Windows 上的使用须知”窗口，请阅读完毕后，单击“Next>”按钮继续安装，如图 13 所示。

步骤四：弹出一个“设置系统信息”对话框，要求输入几个基本的配置参数，是必要的服务器信息和安装选项。需要给出一个网络域名和一个服务器的主机名，还必须给出一个 Web 服务器管理员的电子邮件地址，以及是否要使用 80 端口的选项。这些信息是必须添写的，需要添写的这三条信息均可任意填写，无效的也行。例如 www.lampbrother.net 是主机名加上域名，在第一个输入框中输入“lampbrother.net”域名。在第二个输入框中添入 www.lampbrother.net 服务器名称。其中第三个要添入的是联系电子邮件地址，会在系统出现故障时提供给访问者，是这个 Web 服务器管理员的电子邮件地址，这里输入 lampteacher@gmail.com。最后要选择 Web 服务器的端口，使用默认的 80 端口，并作为系统服务自动启动。另外一个是为当前用户安装使用端口 8080，手动启动。全部输入完成之后，单击“Next>”按钮继续安装，如图 14 所示。

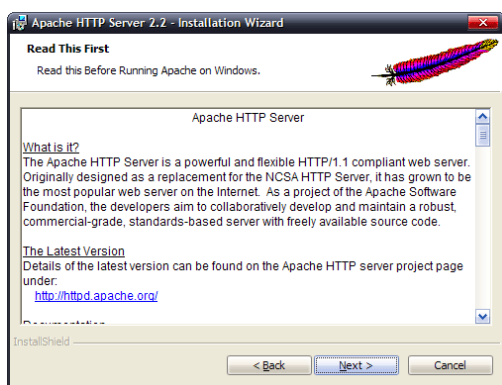


图 13 Apache 使用须知会话窗口

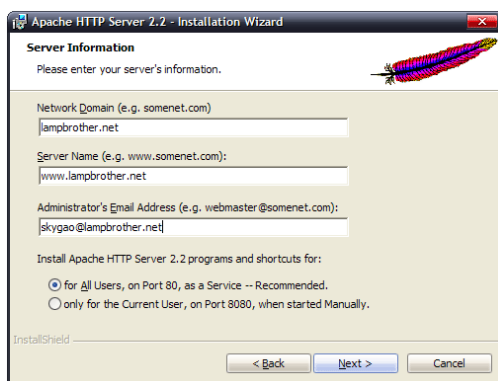


图 14 设置系统信息的对话框

步骤五：弹出选择安装类型对话框，“Typical”为默认安装，“Custom”为用户自定义安装，我们这里选择“Typical”类型。单击“Next>”按钮继续安装，如图 15 所示。

步骤六：弹出选择安装位置对话框，这里可以选择你喜欢的位置，我们这里将 Apache 安装的位置设置为“C:\lamp\apache2”下。单击“Next>”按钮继续安装，如图 16 所示。

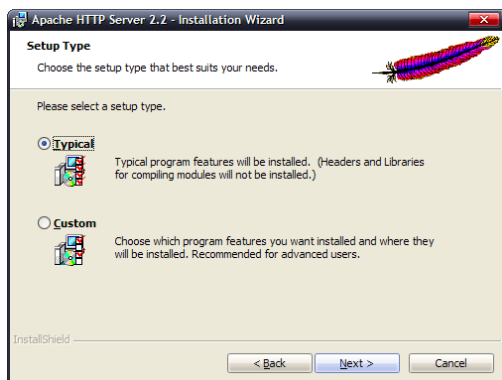


图 15 选择安装类型对话框

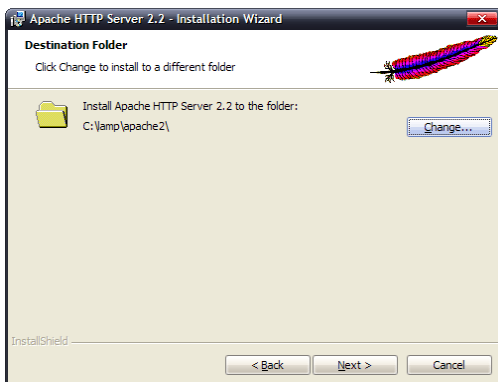


图 16 选择安装位置对话框

步骤七：弹出确认安装对话框。如果需要再检查一遍，可以单击“<Back”按钮一步步返回检查。确保选项无误，单击“Install”开始按前面设定的安装选项安装，如图 17 所示。

步骤八：弹出正在安装界面，请耐心等待，直到出现如图 18 所示的画面，安装向导成功完成，这时右下角状态栏应该出现 Apache 的绿色小图标管理工具，表示 Apache 服务已经开始运行，单击“Finish”按钮结束 Apache 的软件安装。

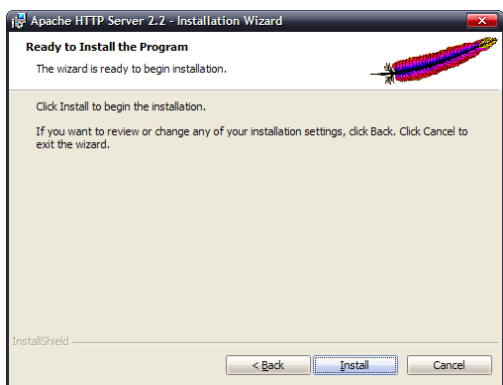


图 17 确认安装对话框

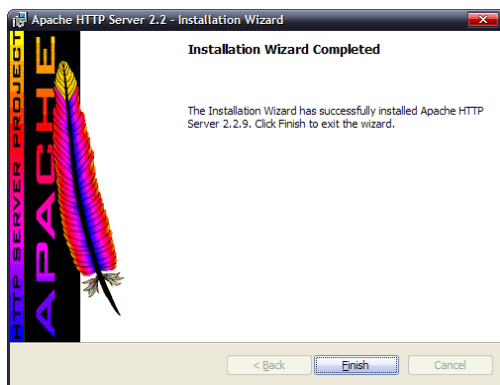


图 18 安装结束界面

Apache 按照以上步骤安装完成以后，它的安装目录被指定到“C:\lamp\apache2”下面。Apache 的配置文件是“C:\lamp\apache2\conf\httpd.conf”文件，Apache 存放网页的地方是“C:\lamp\apache2\htdocs”文件夹。

步骤九：在 Windows 系统里启动和停止 Apache，有多种方式，以下示例都可以做到。

- 单击开始菜单->所有程序->Apache HTTP Server 2.2->Control Apache Server 下面的 Stop 停止、Start 开启或者 Restart 重新启动。
- 右击我的电脑->管理->服务和应用程序->服务->Apache2.2 选项，单击停止、开启或者重新启动，如图 19 所示。
- 开启 Windows 命令行，通过如图 20 所示的示例命令，也可以停止和开启 Apache 服务器。

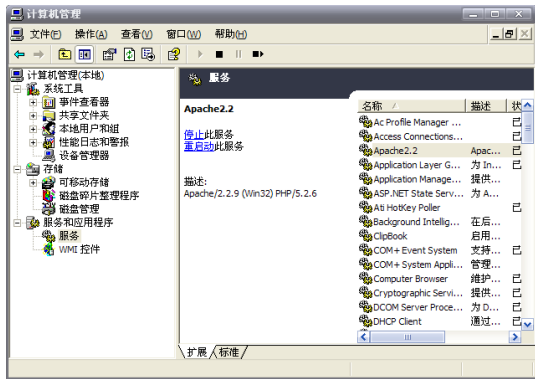


图 19 Windows 系统应用的应用程序管理界面

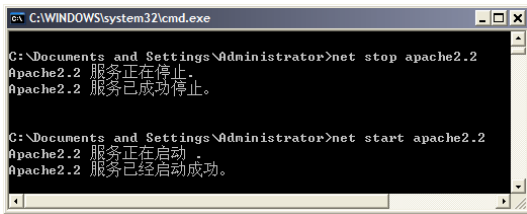


图 20 停止和开启 Apache 服务器的命令行窗口

- 通过 Windows 系统右下角状态栏上 Apache 的绿色小图标管理工具来开启、停止或者重新启动 Apache 服务器。

步骤十：Apache 如果启动成功，就可以通过浏览器测试。开启浏览器并在地址栏中输入 http://localhost/ 的网址，如果看到如图 21 所示的画面，这说明 Apache 安装成功，并可以使用了。

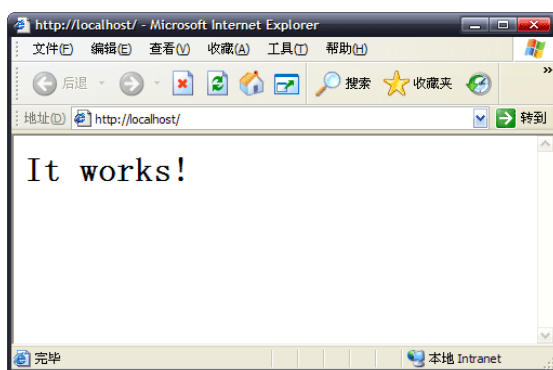


图 21 浏览器测试 Apache 安装成功窗口

3.3 安装 MySQL

安装 MySQL 数据库服务器可以分为两部分：一部分是程序的安装；另一部分是使用 MySQL 配置助手创建 MySQL 配置文件、一些详细配置选择，以及给 MySQL 服务器设置一个密码等内容。MySQL 配置助手是在第一部分安装结束后自动运行的，也可以在 MySQL 安装完成以后手工启动。以下给出了 MySQL 软件的详细安装步骤，具体安装步骤和安装配置如下：

步骤一：进入 MySQL 软件下载的文件夹，找到下载的软件名称为“mysql-5.1.28-rc-win32.zip”的文件，或者是下载更高版本的 MySQL 软件。这份压缩文档只包含一个名为 setup.exe 的文件，双击 setup.exe 文件就可以启动安装程序。屏幕上出现一个对话框，是软件安装向导的欢迎界面，直接单击“Next>”按钮即可到下一步安装，如图 22 所示。

步骤二：弹出安装类型选择对话框，如图 23 所示，有三个选项，分别是典型安装（Typical）、完全安装（Complete）和用户自定义（Custom）三种类型。我们这里选择“Custom”，得到更多的选项，方便我们指定安装位置。单击“Next>”按钮即可到下一步安装，如图 23 所示。



图 22 MySQL 软件安装向导的欢迎界面

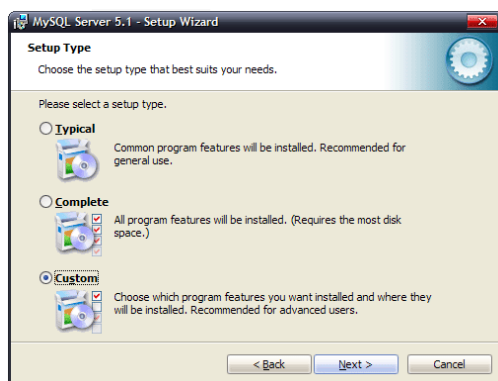


图 23 MySQL 服务器的安装类型选择对话框

步骤三：接着，弹出用户自定义设置页面，如图 24 所示。

如图 24 所示，我们还没有为具体的安装内容指定安装位置。在“Developer Components”选项上单击左键，再选择第二项“This feature, and all subfeatures, will be installed on local hard drive.”，即“此部

分，及下属子部分内容，全部安装在本地硬盘上”。操作过程如图 25 所示。

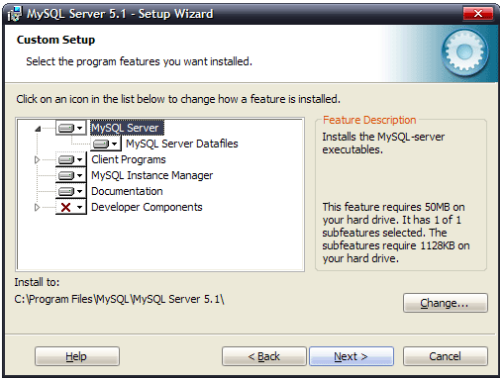


图 24 用户自定义设置页面

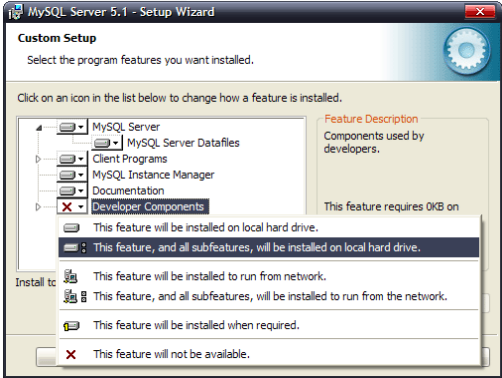


图 25 选择安装部分内容

接下来，就需要指定这个选项的安装位置。单击“Change...”按钮改变安装位置，如图 26 所示示例，我们将其指定到“C:\lamp\mysql5”文件夹下面。不仅这个选项要改变安装位置，前面几个选项包括 MySQL 服务器选项（MySQL Server）、MySQL 客户端程序选项（Client Programs）、文件选项（Documentation），都要改变安装位置，操作都是一样的。单击“Change...”按钮，手动改变安装位置，可以把这几个选项的安装位置都指向“C:\lamp\mysql5”文件夹下。

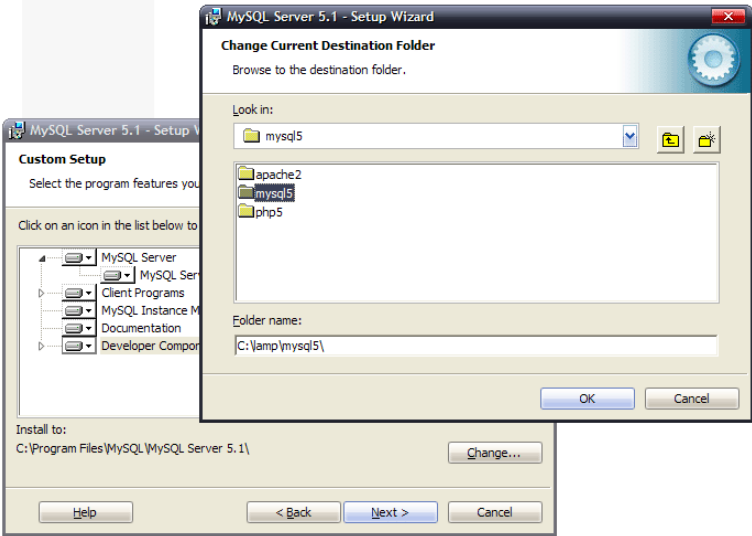


图 26 指定每个选项的安装位置

建议服务器数据文件“MySQL Server Datafiles”选项不要指定与操作系统在同一分区下。这样可以防止系统备份还原的时候，数据被清空。这里将其指定到 D:\datafiles\文件夹中。如图 27 所示，设置完成以后单击“Next>”按钮即可到下一步安装。

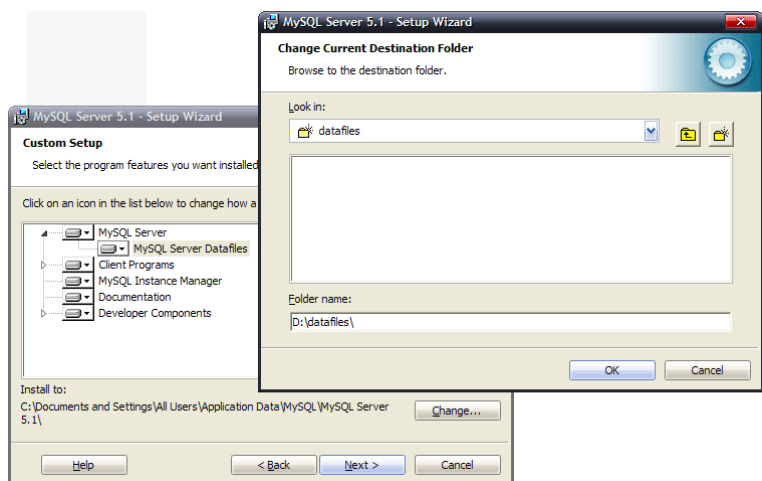


图 27 指定数据文件的安装位置

步骤四：弹出安装位置确认对话框，确认一下上一步设置的安装位置。如果有误，按“Back”返回重新设置，按“Install”则开始安装，如图 28 所示。

接着会弹出安装对话框，需要等待几秒进行安装，如图 29 所示。

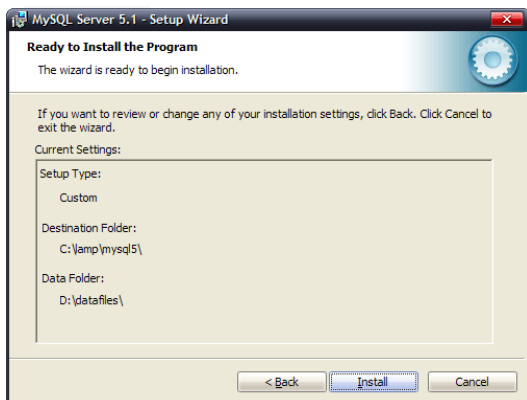


图 28 安装位置确认对话框

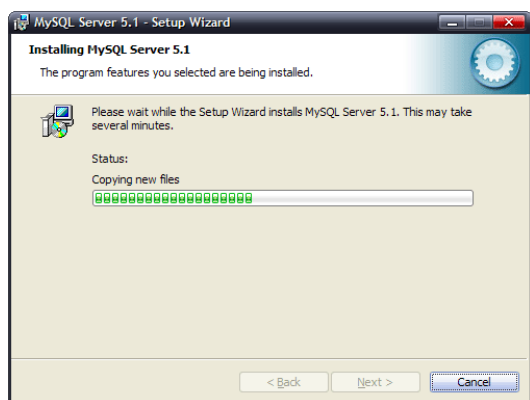


图 29 安装进行对话框

安装完以后弹出结束安装对话框，如图 30 所示。

到这里 MySQL 数据库软件就安装完成了，出现如图 30 所示的界面，勾选“Configure the MySQL Server now”，单击“Finish”按钮结束软件的安装并启动 MySQL 配置向导。

步骤五：经过前面几步的安装，MySQL 数据库软件成功完成安装。然后自动运行 MySQL 配置助手，到了安装的第二部分 MySQL 配置阶段。图 31 为启动 MySQL 配置向导的欢迎窗口，单击“Next>”按钮即可到下一步安装。

步骤六：弹出配置类型选择对话框，有两个选项“Detailed Configuration（详细配置）”和“Standard Configuration（标准配置）”。这里我们为了方便了解配置过程，选择第一项“Detailed Configuration”。以后几步配置都是在选择这一项时，MySQL 配置助手需要设置的一些内容。选好类型之后，单击“Next>”按钮继续下一步配置，如图 32 所示。



图 30 “结束安装”对话框

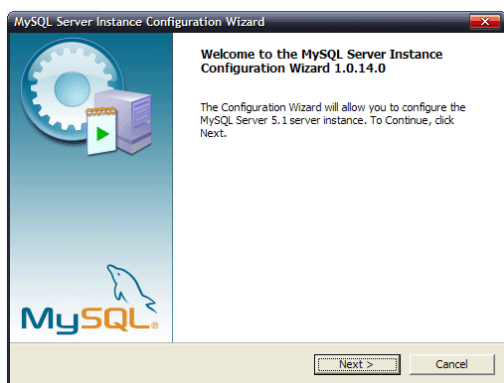


图 31 MySQL 配置向导的欢迎窗口

步骤七：弹出服务器类型（Server Type）选择对话框，它们决定着 MySQL 服务器将试图保留多少主内存。主内存越大，服务器的速度就越快，但对其他程序的影响也会越大。这里有三种选择：Developer Machine 选项是开发测试类，MySQL 占用很少资源；Server Machine 选项是服务器类型，MySQL 占用较多资源；Dedicated MySQL Server Machine 选项是专门的数据库服务器，MySQL 占用所有可用资源。读者可以根据自己的实际应用情况选择不同的类型，对于 Web 开发者来说，第一个 Developer 选项是最佳选择，其他两项最适合用于大数据库。选好类型之后，单击“Next>”按钮继续下一步配置，如图 33 所示。

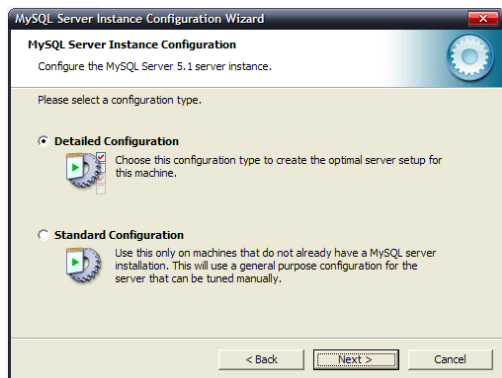


图 32 配置类型选择对话框

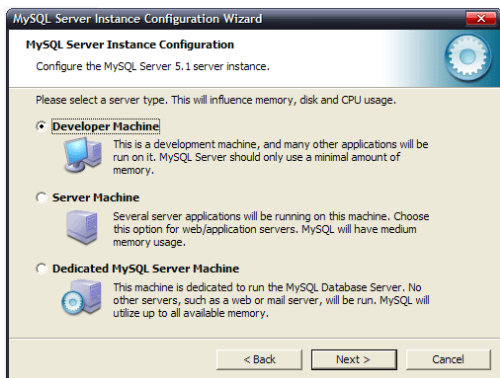


图 33 设置服务器类型选择对话框

步骤八：弹出数据库用途（Database Usage）对话框。它们决定 MySQL 服务器将支持哪几种数据表类型。这里也有三个选项：“Multifunctional Database”选项是通用多功能型，比较好的类型，是默认选择的类型；“Transactional Database Only”选项是服务器类型，专注于事务处理；“Non-Transactional Database Only”选项是非事务处理型，比较简单，主要做一些监控、记数用。用户可以根据实际数据库的用途进行改变，我们这里使用默认的选择，单击“Next>”按钮继续下一步配置，如图 34 所示。

步骤九：弹出 InnoDB 表空间设置（InnoDB Tablespace Setting）对话框，应该在这里为 InnoDB 数据库文件指定一个存放地点。默认存放地点是 MySQL 安装目录，这里使用默认的存放位置，单击“Next>”按钮继续下一步配置，如图 35 所示。

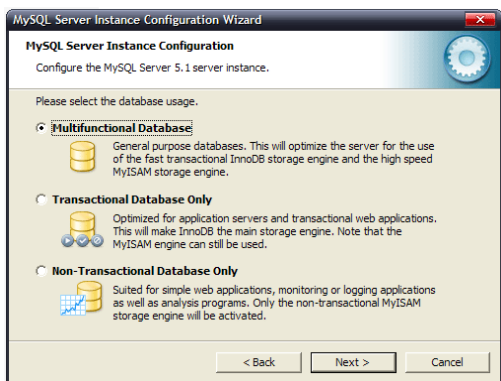


图 34 数据库用途选择对话框

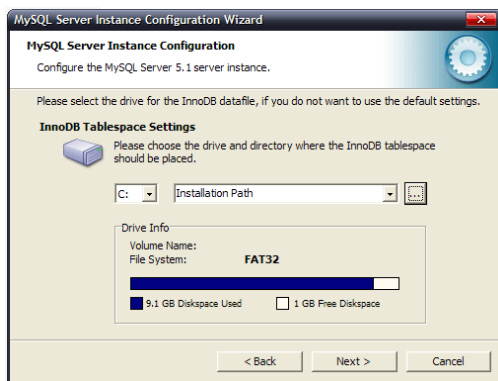


图 35 InnoDB 表空间设置对话框

步骤十：弹出并发连接（Concurrent Connections）选择对话框，这里的选项决定了同时打开的数据库连接的数量。共有三个选项：第一个“Decision Support (DSS) /OLAP”选项可以有 20 个连接；第二个“Online Transaction Processing (OLTP)”选项可以有 500 个连接；第三个“Manual Setting”选项手动设置一个连接数目。对于运行在一台用于软件开发的计算机上的 MySQL 服务器来说，不需要太多的连接，这个时候选择第一个“Decision Support (DSS) /OLAP”应该没什么问题。但是对一些热门的网站上全速工作的 MySQL 服务器来说，并发连接的个数可能非常大，这时应该选第二个“Online Transaction Processing (OLTP)”选项。这里使用默认的第三个选项，单击“Next>”按钮继续下一步配置，如图 36 所示。

步骤十一：弹出激活 TCP/IP 组网功能（Enable TCP/IP Networking）对话框。在 Windows 系统环境里，MySQL 服务器与应用程序的通信有两种方式：一是通过命名管道（named pipe）；二是通过网络协议 TCP/IP。MySQL 配置助手给出的推荐配置是激活 TCP/IP 并使用 3306 号端口。在这里如果没有特殊理由，使用默认推荐的配置。单击“Next>”按钮继续下一步配置，如图 37 所示。

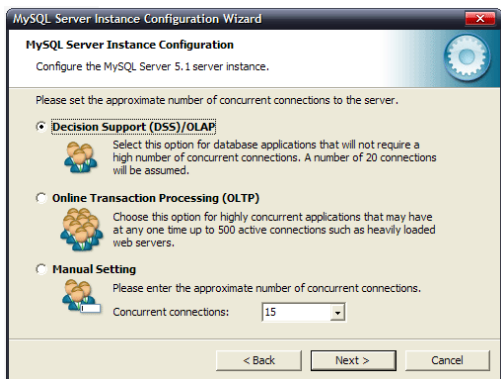


图 36 并发连接选择对话框

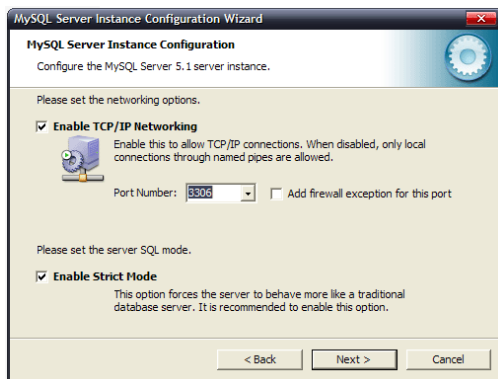


图 37 激活 TCP/IP 组网功能对话框

步骤十二：弹出默认字符集设置（Default Character Set）对话框，在把文本数据保存到数据库里时，MySQL 支持现有的各种字符集。这里我们选择第三项，为了更好地处理中文字符集，在 Character Set 那里选择或填入“utf8”，单击“Next>”按钮继续下一步配置，如图 38 所示。

步骤十三：弹出安装为 Windows 服务（Install as Windows Service）对话框。这里可以选择是否将

MySQL 服务安装为 Windows 系统服务，这样就可以通过 Windows 系统服务管理来管理 MySQL 服务器了。还可以指定 Service Name（服务标识名称），另一个选项是选择是否将 MySQL 的 bin 目录加入到 Windows PATH 里（加入后，就可以直接使用 bin 目录下的文件，而不用指出目录名了）。这里接受 MySQL 配置助手推荐的默认设置，单击“Next>”按钮继续下一步配置，如图 39 所示。

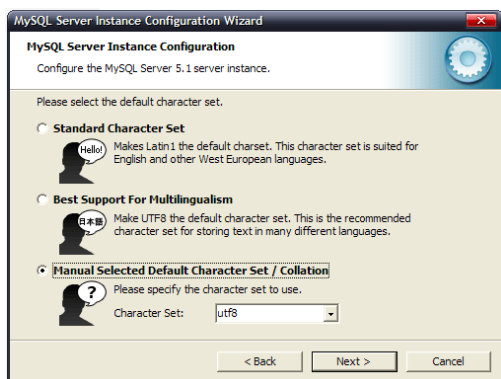


图 38 默认字符集设置对话框



图 39 安装为 Windows 服务对话框

步骤十四：弹出安全选项（Security Options）对话框。这是最重要的一个配置，需要决定允许哪些人来可以连接 MySQL 服务器，以及是否必须输入一个密码才能建立连接。图 40 的设置为最安全的设置，因为 root 是唯一的用户又是 MySQL 系统管理员用户，所以应该让 root 用户正确输入密码才能连接数据库，并且只允许他在本地登录。不要勾选第二个选项，这里不创建任何匿名账号，所谓的匿名账号指的是无须密码就可连接数据库的账号。单击“Next>”按钮继续下一步配置。

步骤十五：弹出确认执行对话框。确认所做的配置设置是否正确，如果有误，单击“<Back”按钮返回检查。如果确认无误，单击“Execute”使设置生效，就会出现如图 41 所示的对话框，单击“Finish”按钮结束 MySQL 的安装与配置。



图 40 安全选项设置对话框

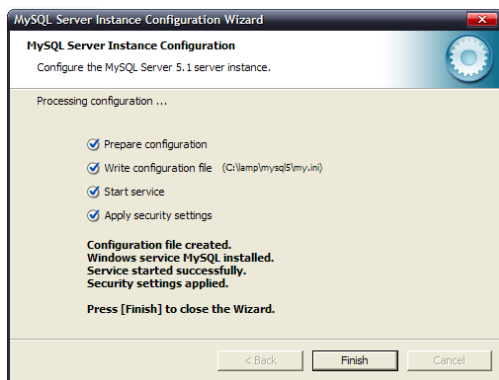


图 41 确认执行对话框

MySQL 安装后环境，按照以上步骤安装和配置完成以后，MySQL 的安装目录是“C:\lamp\mysql5\”下面；MySQL 的配置文件是“C:\lamp\mysql5\my.ini”文件；MySQL 数据文件存放的地方是“D:\datafiles\data”文件夹。

步骤十六：Windows 系统中启动和停止 MySQL，也有多种方式，以下示例都可以做到。

- 右击我的电脑->管理->服务和应用程序->服务->MySQL 选项，单击停止、开启或者重新启动，如图 42 所示。
- 开启 Windows 命令行，通过如图 43 所示示例命令，也可以停止和开启 MySQL 服务器。

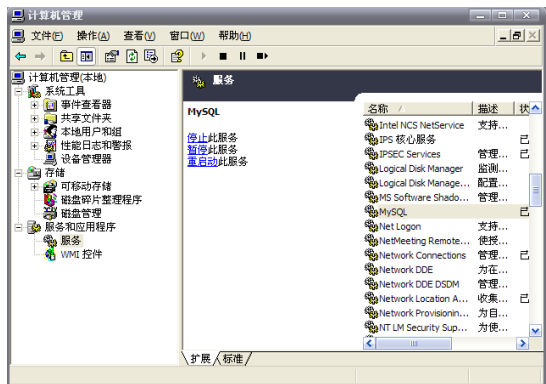


图 42 使用 Windows 管理 MySQL 启动与关闭

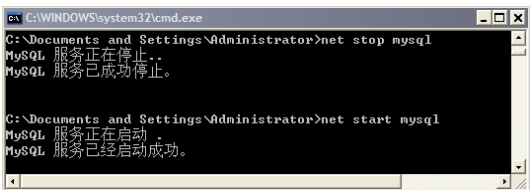


图 43 启动与关闭 MySQL 命令行

步骤十七：测试 MySQL 数据库安装是否成功，可以开启一个 MySQL 客户端去登录 MySQL 服务器进行测试。使用开始菜单->所有程序->MySQL->MySQL Server 5.1->MySQL Command Line Client 启动 MySQL 的命令行解释器程序 mysql.exe，然后输入在配置 MySQL 时输入的密码。如果一切顺利，应该在输入窗口里看到 MySQL 的输入提示符。接下来执行 status 命令，看到如图 44 所示的结果表示 MySQL 服务器成功安装。

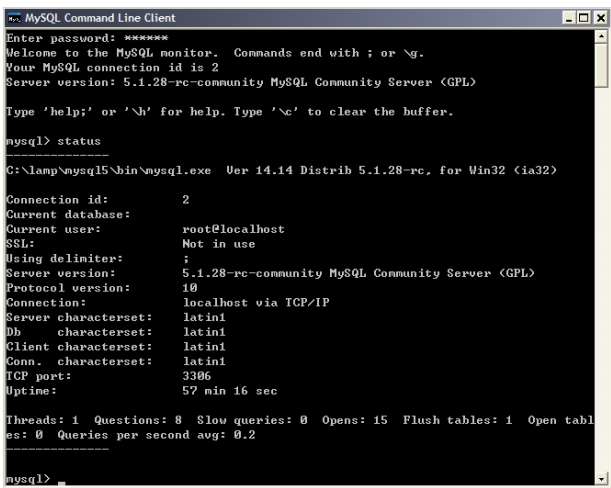


图 44 测试 MySQL 数据库安装成功演示

3.4 安装 PHP

安装 PHP 服务器时，有两个任务必须完成。PHP 属于 Apache 服务器的一个模块，要外挂到 Apache 上协助它来处理 PHP 文件，所以第一个任务就是在 Apache 里加载 PHP 模块。在开发时还要使用 PHP 程序连接数据库并对数据库操作，所以第二个任务就是配置 PHP 的 MySQL 扩展接口。以下给出了 PHP

软件的详细安装步骤，具体安装步骤和配置过程如下。

1. 将 PHP 安装为 Apache 服务器的一个扩展模块

步骤一：进入 PHP 软件下载的文件夹，找到下载的软件名称为“php-5.2.6-Win32.zip”的文件，或者是下载更高版本的 PHP 软件。把这个 ZIP 文档的内容解压缩到选定的某个子目录里即可，我们这里将它解压到“C:\lamp\php5”目录下。

步骤二：将 PHP 以模块的方式加载到 Apache 中去。加载的方式是在 Apache 配置文件 httpd.conf 里写入三行内容：第一行加载 PHP 模块；第二行告诉 Apache 服务器后缀名是.php 或指定其他后缀名的文件使用 PHP 解析；第三行指定 PHP 配置文件的位置。我们这里把下面三行内容插入到 Apache 配置文件 C:\lamp\apache2\conf\httpd.conf 中的某个位置即可。

LoadModule php5_module "C:/lamp/php5/php5apache2_2.dll"	#加载 PHP 模块
AddType application/x-httpd-php .php .phtml	#PHP 模块解析哪种后缀文件
PHPIniDir "C:/lamp/php5"	#指定 PHP 配置文件的位置

第一行，是指以 module 方式加载 PHP 5，将这一行插入到 Apache 配置文件 httpd.conf 中的“#LoadModule vhost_alias_module modules/mod_vhost_alias.so”行下面即可。其中 C:/lamp/php5/php5apache2_2.dll 是安装 PHP 的相应路径下文件。注意不要把 php5apache2_2.dll、php5apache2.dll 和 php5apache.dll 混淆，php5apache.dll 只适用于 Apache 2.0 以前的版本，PHP 5 压缩包里的 php5apache2.dll 只适用于 apache2.0.* 版本，如果是 2.2.* 以上版本，必须使用 php5apache2_2.dll。否则就可能会出现一些加载错误。

第二行，是添加 PHP 的支持。告诉 Apache 将哪些后缀作为 PHP 解析。例如，让 Apache 把.php 或.phtml 后缀的文件解析为 PHP。在 Apache 配置文件中找到“AddType application/x-gzip .gz .tgz”项，在其下方添加“Addtype application/x-httpd-php .php .phtml”行即可。可以将任何后缀的文件解析为 PHP，只要在添加的语句中加入并用空格分开，这里以多添加一个.phtml 来示例。

第三行，是指定 PHP 配置文件 php.ini 的位置，在第二行“Addtype application/x-httpd-php .php .phtml”下面添加就可以了，其中“C:/lamp/php5”指定的是先前选择的 PHP 解压缩的目录。

步骤三：建立 PHP 的配置文件。直接把 PHP 解压缩目录 C:/lamp/php5 下面的 php.ini-recommended 文件改名为 php.ini 即可。也可以把 C:/lamp/php5 目录下的文件 php.ini-dist 改名为 php.ini，但该配置文件自动给所有添加内容加上“//”，增加安全，同时也增加了麻烦；如果把 php.ini-recommended 改名为 php.ini，该配置文件不给所添加内容加上“//”，减少了安全，但更加便利。这里把 php.ini-recommended 文件改名为 php.ini 文件。

步骤四：重新启动 Apache 服务器。只要对 Apache 的配置文件 httpd.conf 或者 PHP 的配置文件 php.ini 进行了改动，就应该重新启动 Apache 服务器。Apache 服务器是在启动的时候加载配置文件 httpd.conf，使配置文件改动的内容生效。PHP 作为 Apache 服务器的一个工作模块，也是在 Apache 启动时加载配置文件 php.ini 使修改内容生效。

步骤五：测试 PHP 安装是否成功。以上步骤已经将 PHP 安装为 Apache 的一个扩展模块，并随 Apache 服务器一起启动。如果想检查一下 PHP 是否安装成功，可以在 Apache 服务器存放网页的文件夹 C:\lamp\apache2\htdocs 下，使用文本编辑器创建一个名为 test.php 的文件，并在文件中写入下面的内容：

```

<?php                                     //PHP 脚本开始标记
    phpinfo();                             //PHP 内部函数，用于打印 PHP 的状态信息
?>                                       //PHP 脚本结束标记

```

然后使用 Web 浏览器打开 <http://localhost/test.php> 页面。如果一切顺利，可以看到如图 45 所示的 PHP 状态信息。

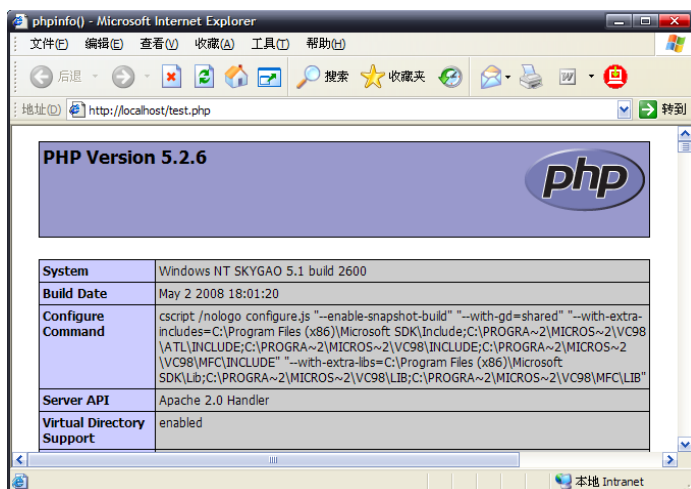


图 45 执行 phpinfo()函数测试结果

2. 配置 PHP 的 MySQL 扩展接口

步骤一：经过以上的配置，PHP5 已经安装成功。但 PHP 现在还不能够访问 MySQL 数据库服务器，所以 PHP 必须找到随 PHP 一起安装的两个和 MySQL 有关的 DLL 文件。在这之前首先要把 PHP 安装目录 C:\lamp\php5\下的 libmysql.dll 文件复制到 Windows 安装目录下。该目录通常是 C:\Windows，有的 Windows 服务器版本是 C:\Winnt 目录。也可以设置 Windows 系统的环境变量，将 PHP 的解压缩目录 C:\lamp\php5\加到 PATH 环境变量里。我们这里将 C:\lamp\php5\libmysql.dll 文件复制到 C:\Windows 下面。

步骤二：在 PHP 配置文件中开启 MySQL 功能。使用文本编辑器打开 PHP 的配置文件 C:\lamp\php5\php.ini。插入下面三项就可以开启 mysql 和 mysqli 扩展接口。也可以在 PHP 配置文件中找到下面三行，将前面用于注释的分号去掉。

```

extension_dir="c:/lamp/php5/ext"          #指定 PHP 的扩展接口目录，目录分隔符是"/"
extension=php_mysql.dll                   #开启 MySQL 扩展接口
extension=php_mysqli.dll                  #开启 MySQLi 扩展接口

```

步骤三：重新启动 Apache 服务器，使用 PHP 配置文件做出的修改生效。

步骤四：查看 PHP 测试页面里的内容，检查是否有对 mysql 和 mysqli 接口参数的描述信息。如果看到如图 46 显示的信息表示 PHP 的 mysql 和 mysqli 扩展接口加载成功，同样打开 Web 浏览器访问 <http://localhost/test.php> 页面进行测试。

经过以上的配置，我们完成了 PHP 的 MySQL 和 MySQLi 扩展接口的配置，如果需要配置其他的扩展接口也是同样的方法。

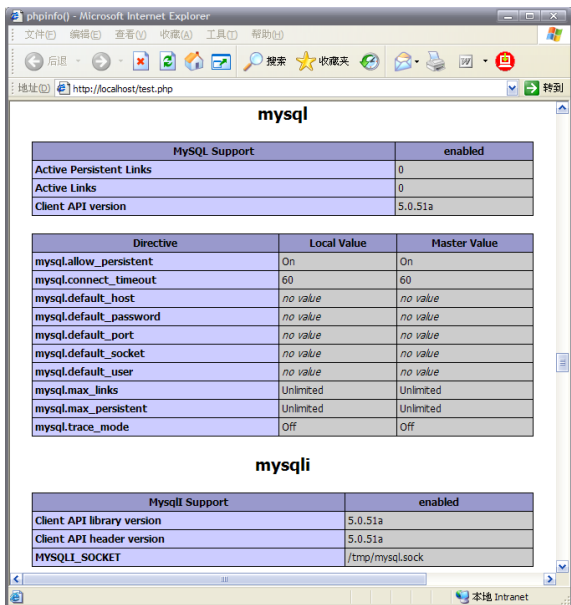


图 46 查看 mysql 和 mysqli 的扩展安装结果

3.5 安装 Zend 加速器

Zend Optimizer 是由 PHP 核心引擎，www.zend.com 创建者 Zend 技术公司所开发的免费 PHP 优化软件。Zend Optimizer 用优化代码的方法来提高 PHP 应用程序的执行速度。实现的原理是对那些在被最终执行之前由运行编译器（Run-Time Compiler）产生的代码进行优化。一般情况下，执行使用 Zend Optimizer 的 PHP 程序比不使用的快 40% 到 100%。以下给出了 Zend Optimizer 软件的详细安装步骤和具体配置过程。

步骤一：进入 Zend Optimizer 软件下载的文件夹，找到下载的软件名称为“ZendOptimizer-3.3.0 a-Windows-i386.exe”的文件。这个直接就是一个安装文件，双击就可以启动安装程序。屏幕上出现一个对话框，是软件安装向导的欢迎界面，直接单击“Next>”按钮即可到下一步安装，如图 47 所示。

步骤二：通过上一步的安装出现以下界面，确认同意软件安装使用许可条例，选择“I accept the terms in the license agreement”，单击“Next>”按钮继续，如图 48 所示。

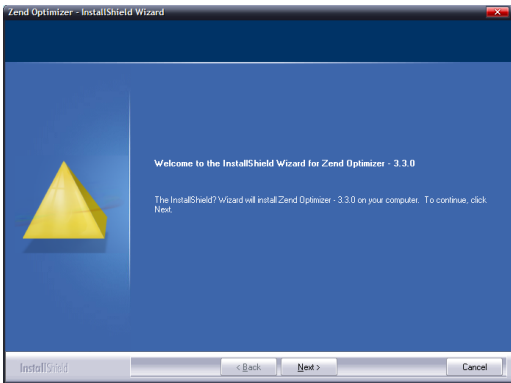


图 47 Zend Optimizer 软件安装向导的欢迎界面

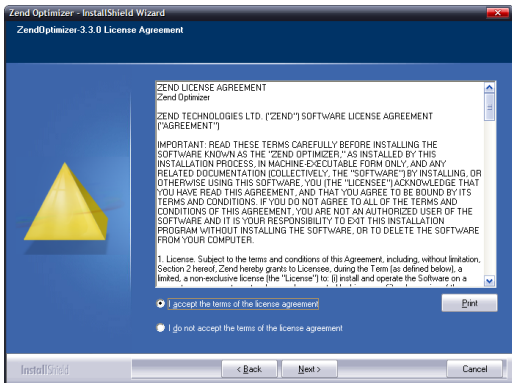


图 48 确认同意软件安装使用许可条例对话框

步骤三：弹出选择安装位置对话框，这里可以选择你喜欢的位置，我们这里将 Zend Optimizer 安装的位置设置为 “C:\lamp\ZendOptimizer\” 下，单击 “Next>” 按钮继续安装，如图 49 所示。

步骤四：弹出如下选择 Web 服务器种类的画面，选择 Apache2.x 选项。单击 “Next>” 按钮继续安装，如图 50 所示。

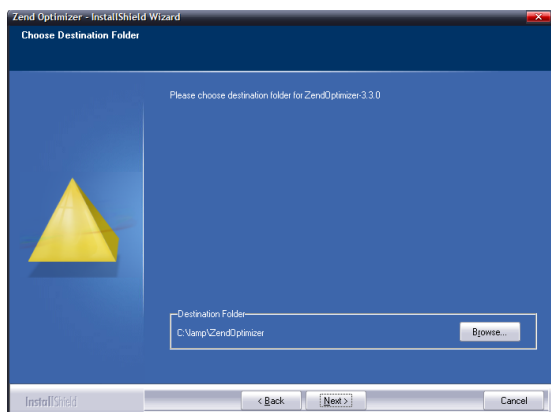


图 49 选择安装位置对话框

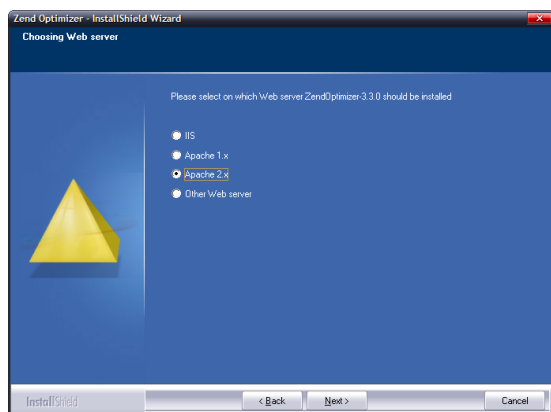


图 50 选择 Web 服务器种类的对话框

步骤五：弹出选择 PHP 配置文件 php.ini 位置对话框。在安装 PHP 时我们将 php.ini 指定到了 PHP 的安装目录 C:\lamp\php5\ 下面，所以这里选择的位置也是 C:\lamp\php5 文件夹。单击 “Next>” 按钮继续安装，如图 51 所示。

步骤六：弹出如下选择网站根目录的对话框，选择网站内容的根目录路径。在安装 Apache 时我们将网站根目录指定到了 Apache 的安装目录 C:\lamp\apache2\ 下面，所以这里选择的位置也是 C:\lamp\apache2\ 文件夹，单击 “Next>” 按钮继续安装，如图 52 所示。

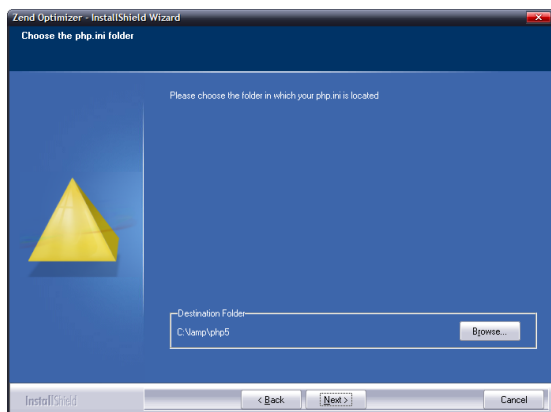


图 51 选择 PHP 配置文件 php.ini 位置对话框

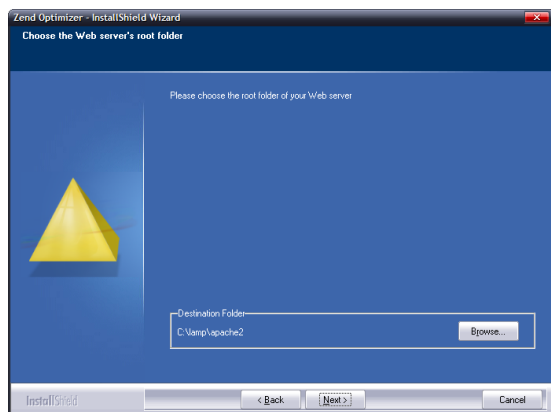


图 52 选择网站根目录的对话框

步骤七：弹出确认安装对话框。如果需要再检查一遍，可以单击 “<Back” 按钮一步步返回检查，确保选项无误，单击 “Install” 按钮开始按前面设定的安装选项安装，如图 53 所示。

步骤八：如果弹出下面的提示，要先关闭 Web 服务器，然后单击 “确定” 按钮继续安装，如图 54 所示。

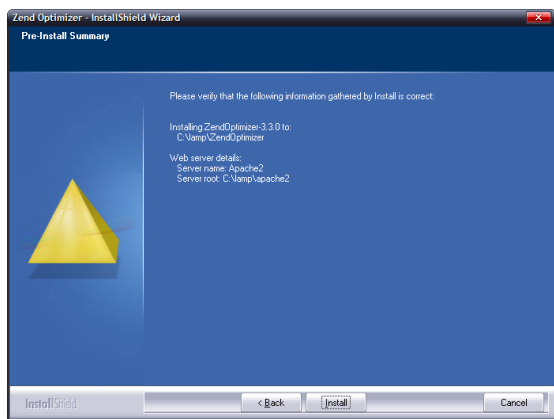


图 53 确认安装对话框

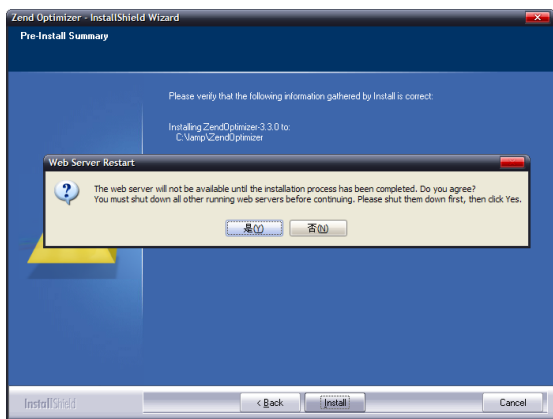


图 54 关闭 Web 服务器的提示对话框

步骤九：如果一切安装顺利将弹出下面的结束安装对话框，单击“Finish”按钮结束安装，这时安装完成，如图 55 所示。

步骤十：检查安装结果。启动 Apache 服务器，打开 Web 浏览器访问 <http://localhost/test.php> 页面，通过 `phpinfo()` 函数进行测试。如果出现以下内容说明安装成功，如图 56 所示。

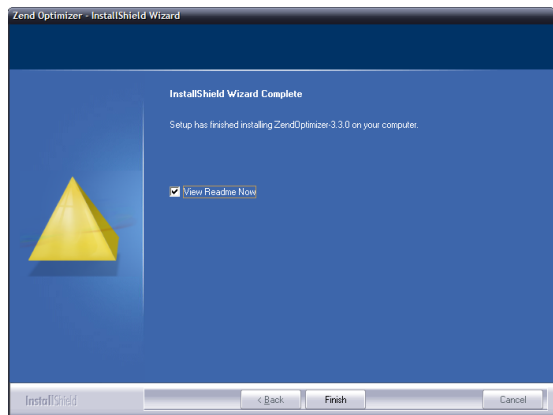


图 55 安装结束对话框



图 56 使用 `phpinfo()` 函数检查安装结果窗口

安装到这里 Windows 系统下 Web 工作环境已经搭建完成。如果也想安装一个数据库管理工具 phpMyAdmin，参考 2.2.15 节中 Linux 系统下 phpMyAdmin 的安装方法，安装与配置步骤是相同的。