

# Linux平台PHP服务端开发——

## 第二讲 编译PHP

# 为何要选择自己编译

---

- 在Ubuntu上使用apt-get可以直接安装Nginx, PHP, MySQL。直接就构建了LNMP环境, 而且编译配置等很麻烦, 但是为何还要自己手动编译?
  1. 直接安装的方式是不能选择版本的, 只能使用软件源提供的版本, Ubuntu上使用编译好的deb软件包。如果系统依赖某些版本提供的功能, 只能使用编译安装的方式。
  2. 软件安装配置分散在多处, 管理不方便。
  3. 可移植性差, 更换系统需要重新安装, 如果安装过程中出现问题, 解决起来很麻烦。

# 目录

---

PHP源代码下载与目录结构

主要编译参数与开发库依赖

开始编译

1

---

## PHP源代码下载与目录结构

# 下载源代码

---

- 打开 <http://php.net> 网址下载PHP7稳定版源代码
- 目前PHP7主要有3个版本：PHP7.0， PHP7.1， PHP7.2
- 三个不同版本有一些功能区别，具体见PHP开发者手册
- PHP7.2变化较大，主要有：
  - 底层引擎加入一些宏指令
  - 移除mcrypt加密扩展
  - 加入sodium加密扩展

- 如何选择：

对于学习来说，三个版本都可以，对于实际应用来说，新版本是趋势，而一些已经成型的系统，不建议最新版本，因为会带来兼容性的问题。在实际使用上，一些流行的框架以及CMS等可以升级到PHP7.0，升级PHP7.1问题也不大。

我们这里选择PHP7.1进行编译，待熟悉之后，同学们自行编译PHP7.2

# 目录结构

- 解压PHP源代码，列出文件

```
wy@helloworld:~/source/php-7.1.12$ l
acinclude.m4      header      php.ini-development  README.WIN32-BUILD-SYSTEM
aclocal.m4        INSTALL    php.ini-production   run-tests.php*
appveyor/         install-sh  README.EXT_SKEL      sapi/
build/            LICENSE    README.GIT-RULES     scripts/
buildconf*        ltmain.sh  README.input_filter   server-tests-config.php*
buildconf.bat     main/      README.MAILINGLIST_RULES  server-tests.php*
CODING_STANDARDS  makedist*  README.md            snapshot*
config.guess      Makefile.frag  README.namespaces    stamp-h.in
config.sub        Makefile.gcov  README.NEW-OUTPUT-API  stub.c
configure*        Makefile.global  README.PARAMETER_PARSING_API  tests/
configure.in      makerpm      README.REDIST.BINS    travis/
CONTRIBUTING.md  missing      README.RELEASE_PROCESS  TSRM/
CREDITS           mkinstalldirs  README.SELF-CONTAINED-EXTENSIONS  UPGRADING
ext/              netware/      README.STREAMS        UPGRADING.INTERNALS
EXTENSIONS        NEWS         README.SUBMITTING_PATCH  vcsclean*
footer            pear/        README.TESTING         win32/
generated_lists  php7.spec.in  README.TESTING2       Zend/
genfiles*         php.gif      README.UNIX-BUILD-SYSTEM
wy@helloworld:~/source/php-7.1.12$
```

# 主要目录与文件说明

---

目录/文件	说明
build/	此目录下的脚本执行环境检测等一系列工作
main/	PHP语言主要实现
ext/	扩展目录
zend/	PHP引擎实现
sapi/	服务编程接口，程序由此开始执行，提供了cgi，cli，fpm等入口方式
configure	配置文件，用于编译初始化，并生成makefile文件

# 编译环境

---

- Ubuntu 16.04
- 编译软件：gcc
- 其他软件：make , autoconf , pkg-config
- 安装依赖软件：
  - Ubuntu : `sudo apt-get install gcc make autoconf pkg-config`



# configure脚本使用

---

- 可以使用Vim打开configure文件，此文件是具有可执行权限的shell脚本
- configure文件是编译PHP的初始化配置脚本
  - 配置编译参数
  - 编译环境检测
  - 生成makefile文件
- 使用：
  - 在命令终端切换PHP源代码目录，运行 `./configure --help` 可以查看配置参数以及解释
  - 使用 `./configure --prefix=/usr/local/php --enable-mysqlnd --enable-fpm .....`

# 2

---

## 主要编译参数与开发库依赖

## configure一些编译参数的说明

---

参数	说明
--prefix=	程序安装目录
--with-config-file-path	配置文件php.ini路径
--with-apxs2	编译共享的Apache2.0模块
--enable-sockets	支持sockets编程接口
--enable-fpm	建立fpm SAPI, 用于Nginx等CGI形式的接入
--enable-mysqlnd	编译mysqlnd扩展

## configure一些编译参数的说明

---

参数	说明
--enable-pcntl	进程控制支持
--enable-sysvmsg	进程间通信支持
--enable-sysvshm	Unix共享内存支持
--enable-sysvsem	此扩展对System V IPC进行封装，提供信号，共享内存，进程间通信支持
--enable-mbstring	多字节string支持

3

---

开始编译

## 编译之前

---

- 在以后的课件中使用PHP\_INSTALL\_DIR表示PHP的安装目录。
- 这里使用/webrun/php7115作为PHP的安装目录， php7115表示PHP的版本号为7.1.15，以后编译新的版本目录可以使用php[版本号] 的形式，互相不冲突。
- /webrun目录是自己创建的，在根目录下创建为的是可移植环境，不要在用户主目录下创建，这会带来权限，移植的麻烦。
- 以后编译Nginx也放在/webrun目录下。

# 编译PHP

---

- 运行脚本：  
`./configure --prefix=/webrun/php7115 --enable-bcmath --enable-calendar --enable-fpm --enable-ftp --enable-mbstring --enable-mysqlnd --enable-pcntl --enable-sockets --enable-sysvsem --enable-sysvshm --enable-sysvmsg --enable-zip`
- 运行此配置脚本，会自动检测编译环境，并把安装路径设置为 /webrun/php7115，同时开启相应的扩展支持。

# 编译PHP扩展的过程

---

- PHP源代码目录中的ext下有官方提供的扩展，如果是其他扩展可以去pecl网站上下载。
- PHP扩展编译的基本流程：

PHP\_INSTALL\_DIR/bin/phpize //生成configure文件

*/\*环境检测，并生成Makefile文件，后面的参数会根据php-config确定编译环境，安装目录等信息。运行configure根据不同扩展还需要其他参数，需要根据具体扩展决定。\*/*

`./configure --with-php-config=PHP_INSTALL_DIR/bin/php-config`

`make` //编译

`sudo make install` //安装扩展

*/\*安装扩展其实就是把modules/下的.so文件复制到PHP的扩展目录下，由于configure已经从php-config确定了扩展目录并写入到了Makefile文件，所以make install会自动复制过去。这一步直接使用cp命令复制过去也是可以的。PHP编译安装后的扩展目录在  
PHP\_INSTALL\_DIR/lib/php/extensions/no-debug-non-zts-20160303\*/*



## 编译PHP扩展示例：pdo\_mysql

---

- PHP源代码目录切换到ext/pdo\_mysql目录，依次运行：

*//生成configure文件*

PHP\_INSTALL\_DIR/bin/phpize

./configure --with-php-config=PHP\_INSTALL\_DIR/bin/php-config

*//根据Makefile文件进行编译*

make

*//安装扩展*

sudo make install

# 配置PHP启用扩展

---

- PHP配置文件默认在安装目录下的lib/php.ini
- 打开配置文件，找到extension相关的部分，加入extension=pdo.so即可启用pdo扩展。

*//配置示例*

extension=pdo\_mysql.so

# PHP-FPM控制脚本

---

- PHP-FPM的管理在Linux上使用bash脚本实现。
- PHP\_INSTALL\_DIR/var/run/php-fpm.pid记录了php-fpm的PID，通过获取PID可以控制进程的重启，退出，重新加载配置文件的操作。
- 管理脚本代码无法在ppt展示，参看实际代码文件。