06 软件包管理

软件包

在Linux系统中，所有的软件都是以软件包的形式提供的。软件包主要有两种形式：

（1）二进制软件包（Binary Packages）：包含可执行文件、库文件、配置文件、文档资料和版本等信息。Ubuntu使用的软件包格式是Deb（以deb为文件的扩展名）。

（2）源代码软件包（Source Packages）：包含软件源代码、版本修改说明、构建指令以及编译工具等。

不同的Linux版本使用不同的软件包格式，同时使用不同的软件包管理工具。

软件源

软件源就是一个按照一定组织形式存储软件包的仓库，当我们需要某个软件时，可以直接从中搜索、下载安装即可。

Ubuntu使用文件/etc/apt/sources.list来保存软件包的安装和更新源的地址。

```
#deb cdrom:[Ubuntu 18.04.3 LTS _Bionic Beaver_ - Release amd64 (20190805)]/ bionic main restricted

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://cn.archive.ubuntu.com/ubuntu/ bionic main restricted
# deb-src http://cn.archive.ubuntu.com/ubuntu/ bionic main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://cn.archive.ubuntu.com/ubuntu/ bionic-updates main restricted
# deb-src http://cn.archive.ubuntu.com/ubuntu/ bionic-updates main restricted
```

sources.list文件格式

每一行包括如下几部分：

- 文件包类型。deb表示二进制软件；deb-src表示是源代码。
- URL。表示提供软件源的HTTP或FTP服务器的URL地址。
- 发行版本。一般为与版本代号相关的名称，如现在我们用Ubuntu18.04，版本名称为bionic，所以每行第三部分内容为 bionic或以bionic开头。
- 软件包分类。跟在发行版之后的就是软件包的分类，可以有一个或多个，用于帮助APT命令遍历软件库。分类是用空格隔开的字符串，每个字符串对应相应的目录结构。
  - **main**：Canonical支持的自由软件。
  - **restricted**：设备的专有驱动。
  - **universe**：社区维护的自由软件。
  - **multiverse**：受版权或法律保护的软件。

Index of /ubuntu/dists/bionic

Name	Last modified	Size
Parent Directory		-
Contents-amd64.gz	2018-04-26 05:59	38M
Contents-i386.gz	2018-04-26 07:25	37M
InRelease	2018-04-26 23:38	236K
Release	2018-04-26 23:38	236K
Release.gpg	2018-04-26 23:38	819
by-hash/	2017-10-25 09:04	-
main/	2018-04-24 01:33	-
multiverse/	2017-10-25 13:33	-
restricted/	2017-10-24 22:44	-
universe/	2017-10-25 13:33	-

Apache/2.4.18 (Ubuntu) Server at archive.ubuntu.com Port 80

更新软件源

可以编辑sources.list来更软件源，加入新的软件源或者某些专有源，也可以采用图像化工具“软件源”对sources.list进行设置。

软件之间的相互依赖

软件包的依赖关系有如下几种情况：

1. depends（依赖）：软件包A依赖软件包B，要运行A必须安装B。在有些情况下，A不仅依赖B，还要求B的特定版本。版本依赖通常有最低版本限制，A更依赖于B的最新版而非某个特定版本。
2. recommends（推荐）：软件包A推荐软件包B，软件包维护者认为所有用户都不会喜欢缺少某些功能的A，而这些功能需要B来提供。
3. suggests（建议）：软件包A建议软件包B，B中某些文件与A的功能相关，并能够增强A的功能。
4. conflicts（冲突）：软件包A与软件包B冲突，如果系统中安装了B那么A无法 运行。“conflicts”常和“replaces”同时出现。
5. replaces（替换）：软件包A替换软件包B，B安装的文件被A中的文件移除和覆盖了。
6. provides（提供）：软件包A提供软件包B，A中包含了B中的所有文件和功能。

DEB

Deb是Debian和Ubuntu系列发行版本上使用的软件包格式，使用dpkg命令进行管理。

dpkg命令用于安装、编译、卸载和查询deb软件包，使用dpkg命令安装或删除软件包时，需要管理员权限。

格式：

dpkg　[选项参数]　.deb文件名

选项参数：

-i	安装.deb软件包
-I	查看未安装软件包的信息
-r	删除.deb软件包，但不删除该软件包配置文件
-P	完全卸载软件包，包括所有相关配置文件
-c	列出deb软件包中所包含的文件
-l	查看软件包的信息
-L	查看当前系统中指定软件包安装的相关文件
-s	查询已安装的指定软件包的详细信息
-S	查询系统中某个文件所属的软件包

RPM

RPM是RedHat Package Manager的缩写，是由Red Hat公司提出的一种软件包管理标准，文件后缀名为.rpm。用于安装，查询，更新，卸载软件包，以及生成.rpm格式的软件包。

不建议在 Ubuntu 系统中安装RPM软件包。在绝大多数情况下，Ubuntu 自身的 .deb 软件包是可用的。然而，如果绝对必要，可以使用程序 alien 将 .rpm 文件转化为.deb 文件。

（1）默认情况下Ubuntu并没有安装alien实用程序，用户需要自己安装alien程序，安装命令如下：

```
sudo apt-get install alien
```

（2）使用管理员权限，执行下面的命令生成同名的.deb文件：

```
sudo alien package_file.rpm
```

（3）使用dpkg工具安装package_file.deb文件。

高级软件包管理工具

高级软件包管理工具通过Internet主动获取软件包，自动检查和修复软件包之间的依赖关系，实现软件的自动安装和更新升级。主要的高级软件包管理工具有两种：

Yum（Yellow dog Updater, modified）是一个基于RPM包的软件包管理工具。Red Hat Enterprise Linux、Centos、Fedora等Linux发行版采用Yum。

APT (Advanced Packaging Tools高级软件包工具) 是Debian及其派生发行版的软件包管理器。APT可以自动下载、配置、安装二进制或者源代码软件包。

APT工具

APT命令行工具有两个，apt-get用于执行与软件包安装有关的所有操作，apt-cache用于查询软件包的相关信息。

apt-get命令本身并不具有管理软件包功能，它只提供了一个软件包管理的命令行平台。

格式：apt-get 子命令 [选项参数] [软件包名称]

APT工具

apt-get的子命令

子 命 令	功 能 描 述
update	下载更新软件包列表
upgrade	将系统中所有软件包升级到最新的版本
install	下载所需软件包并进行安装配置
remove	删除软件包
purge	移除软件包和配置文件
autoremove	将不满足依赖关系的软件包自动删除
source	下载源码文件包
build-dep	为源码包构建所依赖的编译环境
dist-upgrade	会识别出当依赖关系改变的情形并作出升级处理
dselect-upgrade	根据dselect的选择来进行软件包升级
clean	删除缓存区中所有已下载的包文件
autoclean	删除缓存区中老版本的已下载的包文件
check	检查系统中依赖关系的完整性

APT工具

apt-get命令的选项参数

选项参数	参数含义
-d	仅下载软件包，而不安装或解压
-f	修复系统中存在的软件包依赖性问题
-m	当发现缺少关联软件包时，仍试图继续执行
-q	将输出作为日志保留，不获取命令执行进度
--purge	与remove子命令一起使用，完全卸载软件包
--reinstall	与install子命令一起使用，重新安装软件包
-b	在下载完源码包后，编译生成相应的软件包
-s	只是模拟命令执行结果，而不做实际操作
-y	apt-get不再进行任何提示，对所有询问都做肯定回答
-u	显示已升级软件包列表
-h	显示帮助信息
-v	显示apt-get版本号

安装软件包

使用apt-get命令安装软件包

sudo apt-get install 软件包名

安装mysql数据库

sudo apt-get install mysql-server mysql-common mysql-client

```
lei@lei-VirtualBox:~$ sudo apt-get install mysql-server mysql-common mysql-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libevent-core-2.1-6 libhtml-template-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libevent-core-2.1-6 libhtml-template-perl mysql-client
  mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server
  mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 10 newly installed, 0 to remove and 186 not upgraded.
Need to get 20.5 MB of archives.
After this operation, 161 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://cn.archive.ubuntu.com/ubuntu bionic/main amd64 mysql-common all 5.8+1.0
.4 [7,308 B]
```

修改ROOT认证模式

更改mysql root账户认证模式，切换到root用户，运行mysql命令

```
lei@lei-VirtualBox:~$ mysql
ERROR 1045 (28000): Access denied for user 'lei'@'localhost' (using password: NO)
lei@lei-VirtualBox:~$ sudo -i
root@lei-VirtualBox:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.27-0ubuntu0.18.04.1 (Ubuntu)
```

在mysql命令行运行一下命令，为root用户添加密码

```
select user, plugin from mysql.user;
update mysql.user set authentication_string=PASSWORD('123456'),
plugin='mysql_native_password' where user='root';
flush privileges;
exit;
```

MYSQL常用命令

- 连接数据库 mysql -u root -p
- 建立数据库 CREATE DATABASE mydb;
- 显示数据库 SHOW DATABASES;
- 使用数据库 USE mydb;
- 建表 CREATE TABLE mytable(name varchar(20),sex char(2));
- 查看表结构 DESCRIBE mytable;
- 插入数据 INSERT INTO mytable values('xiaoming','m');
- 查看数据 SELECT * FROM mytable;

创建新用户

- 创建数据库

```
CREATE DATABASE wordpress;
```

- 执行以下命令，创建一个新用户。例如 “user”，登录密码为 123456。

```
CREATE USER 'user'@'localhost' IDENTIFIED BY '123456';
```

- 执行以下命令，赋予用户对 “wordpress” 数据库的全部权限。

```
GRANT ALL PRIVILEGES ON wordpress.* TO 'user'@'localhost' IDENTIFIED BY '123456';
```

- 执行以下命令，使所有配置生效。

```
FLUSH PRIVILEGES;
```

删除软件包

使用apt-get删除软件包的命令格式主要有下面3种：

（1）删除已经安装的软件包，保留软件包的配置文件

apt-get remove 软件包名称

（2）彻底删除已经安装的软件包及相关配置文件

apt-get purge remove 软件包名称

（3）删除软件包及其所依赖的，不在保留不需要的软件包

apt-get autoremove 软件包名称

升级软件包

软件包升级更新一般分两步。

1. 首先执行命令

apt-get update

一般说来，这是安装软件要进行的第一步。apt-get update将扫描软件源中软件包列表文件(主要是文件名为package或者source列表文件)来更新本地数据库列表，它将使系统获得最新的软件包更新和安全更新等信息。

2. 更新已安装的软件包

apt-get upgrade

apt-get update只是让你知道软件有更新或者有新版本的软件，而apt-get upgrade才真正更新已经安装的软件。

搜索软件包命令apt-cache

apt-cache是一个底层的工具，用户可以利用它来查询软件包的相关信息，也可以操作APT的二进制缓存文件。

格式： apt-cache 子命令 [选项参数] [软件包名称]

apt-cache命令的子命令

子 命 令	命 令 含 义
show pkg	显示软件包“pkg”的相关信息，例如描述、版本、大小、适用的计算机架构名称、依赖以及冲突等信息
showpkg	显示二进制软件包的常规描述信息
snowsrc	显示源码包的详细描述信息
stats	显示软件源的统计信息
dump	显示软件源中所有软件包的简要信息
dumpavail	显示已安装的所有软件包的描述信息
unmet	显示所有未满足的依赖关系
search pkg	索搜与“pkg”表列包件软的配匹
depends	显示该软件包所依赖的软件包
rdepends	显示所有依赖于该软件包的软件包
pkgnames	列出所有已安装软件包的名称
policy	显示软件包当前安装状态

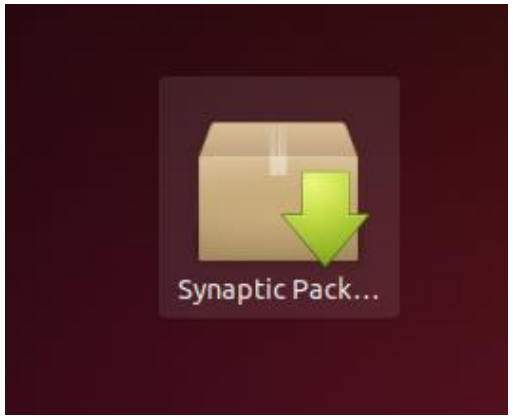
搜索软件包命令apt-cache

apt-cache命令的选项参数

选项参数	参数含义
-c	读取指定配置文件
-s	显示源代码包缓存文件
-q	关闭进度显示
-i	只显示重要依赖关系，与unmet和depends一起使用
-o	设置任意给定的配置选项
-h	显示帮助信息
-v	显示版本信息

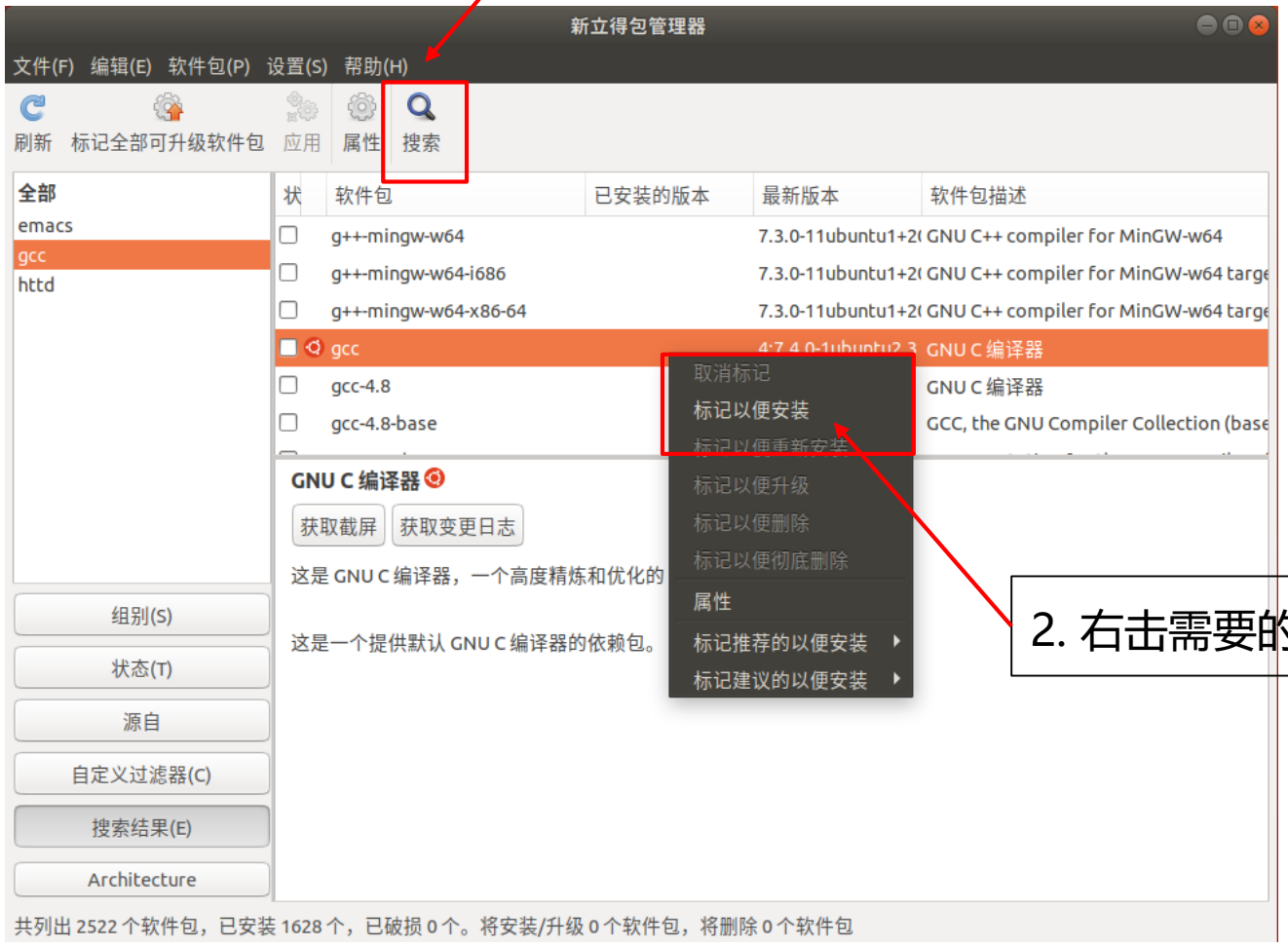
新立得（Synaptic Package Manager）

新立得软件包管理器是APT管理工具的图形化界面，通过新立得可以安装、删除、配置、升级软件包，对软件包列表进行浏览、排序、搜索以及管理软件源，甚至升级整个系统。



安装软件包

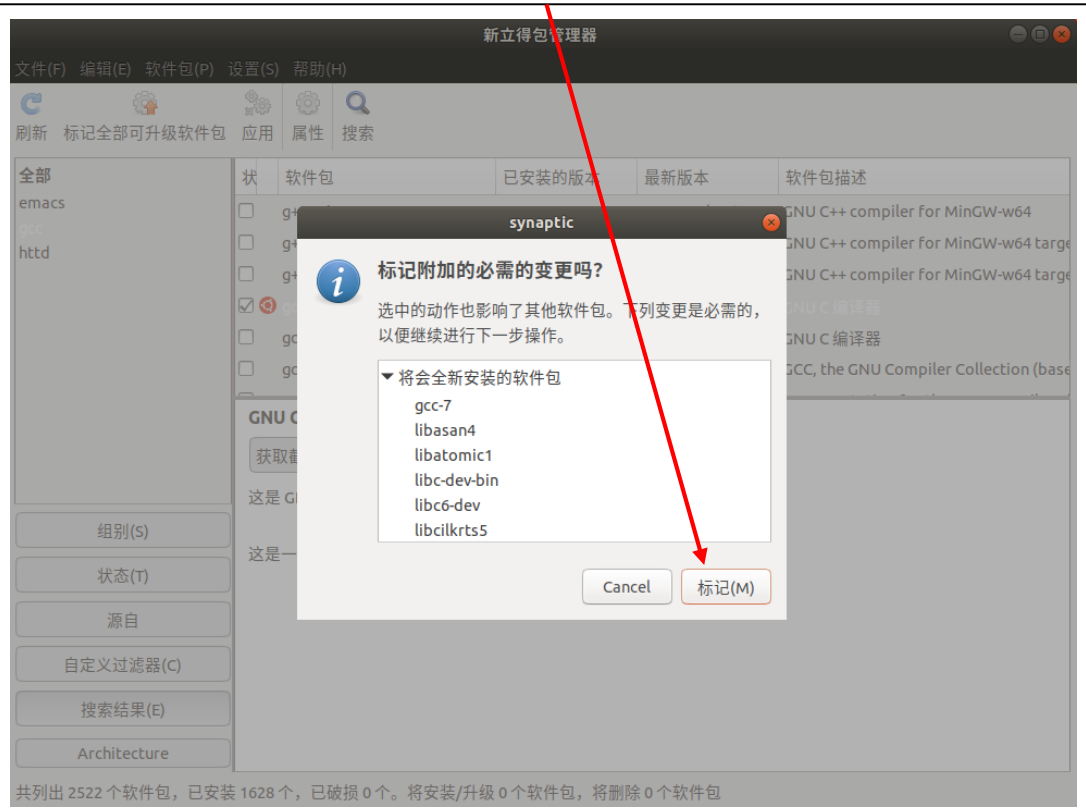
1.通过软件名称或描述信息搜索需要的软件包



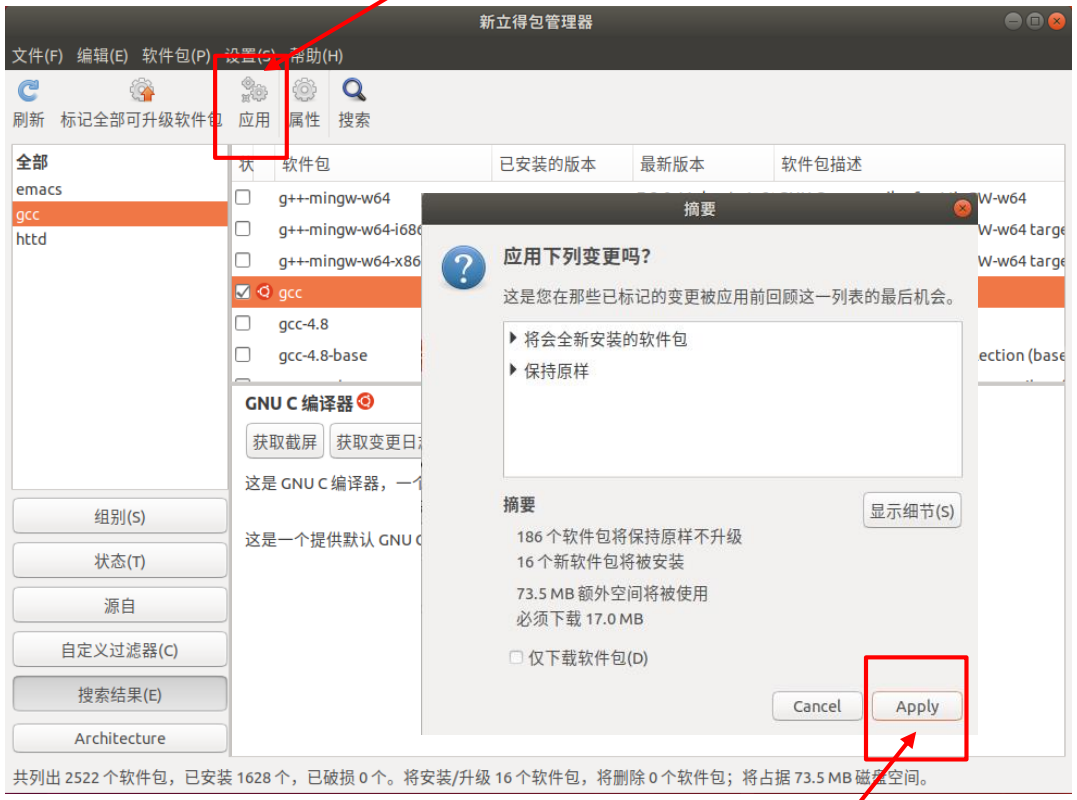
2. 右击需要的软件，选择标记以便安装

安装软件包

在弹出的对话框中，根据依赖关系指示要安装和升级的软件包，单击标记。



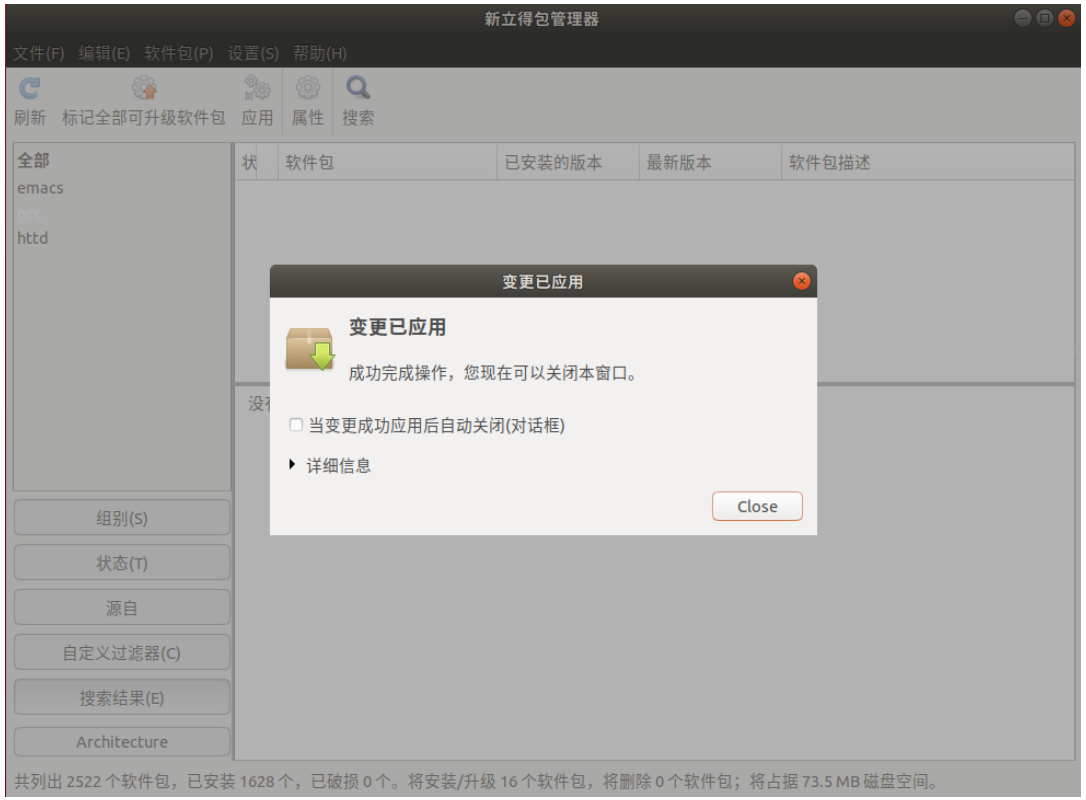
1. 点击工具栏中的应用按钮



2. 在弹出的摘要对话框单击应用

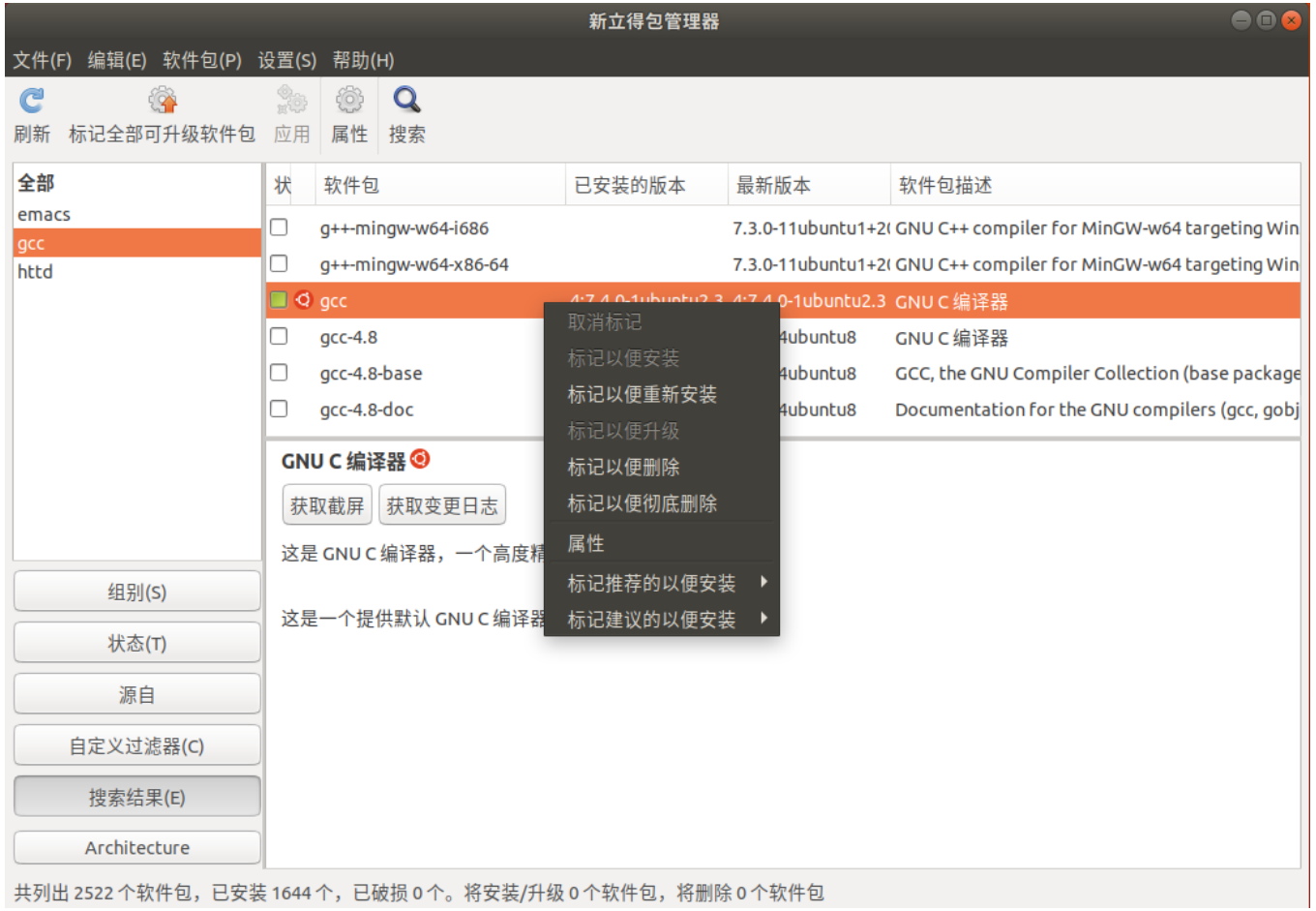
安装软件包

下载完毕将自动安装和配置软件包，并显示安装过程，完成后提示变更已应用，单击关机即可。



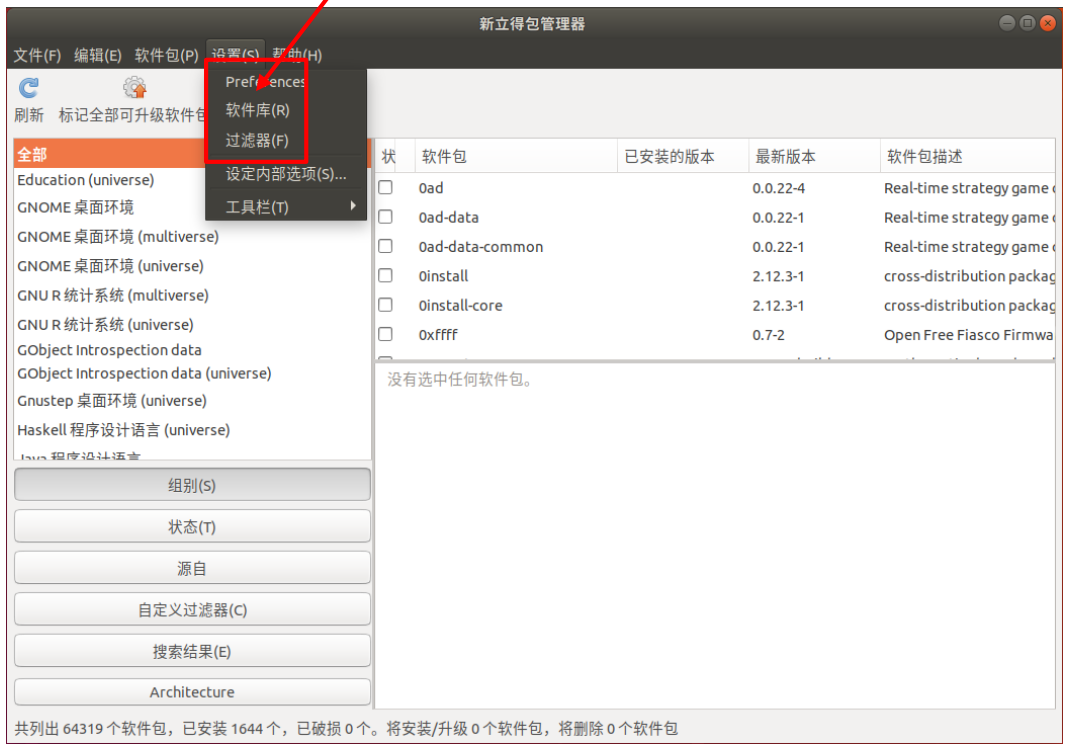
管理已安装软件包

在软件包列表，右击已安装的软件包，可以进行重新安装，升级，删除操作。

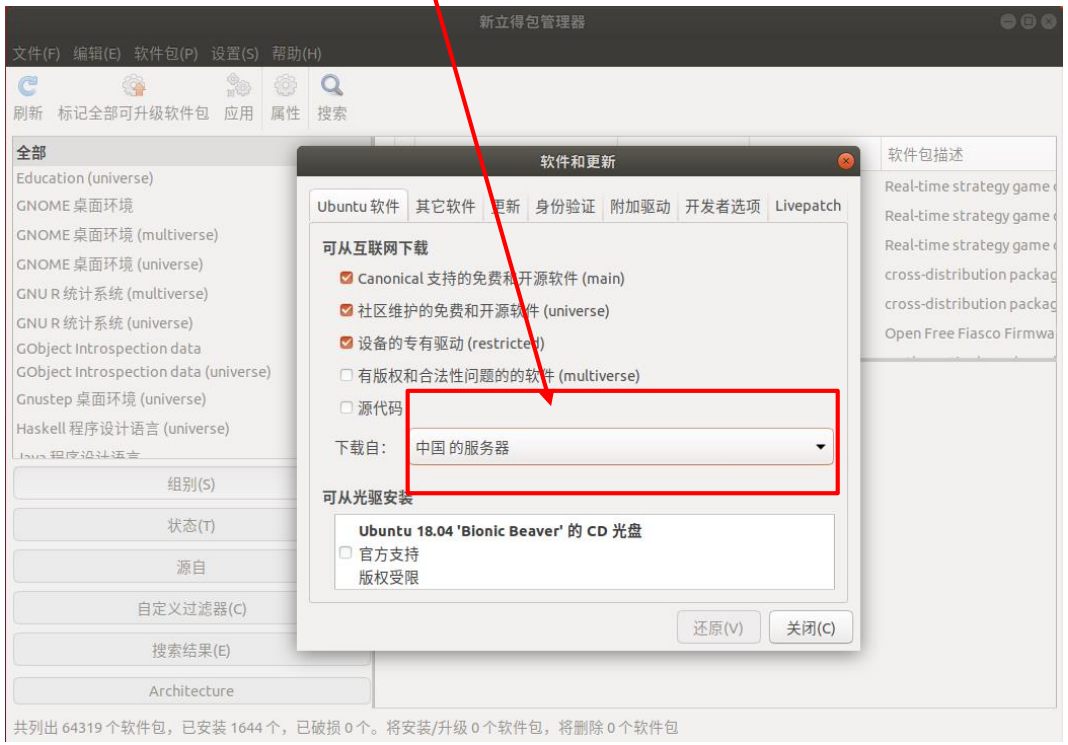


更新源

打开设置中的软件库



选择软件包下载服务器



APT-GET安装目录和安装路径

- apt-get 下载后，软件存放在/var/cache/apt/archives
- 软件安装目录一般在/usr/share
- 可执行的文件在/usr/bin，
- 配置文件可能安装到了/etc下等。
- 文档一般在 /usr/share
- 可执行文件 /usr/bin
- 配置文件 /etc
- lib文件 /usr/lib

使用源代码安装

查看安装说明文件INSTALL

```
APACHE INSTALLATION OVERVIEW

Quick Start - Unix
------------------

For complete installation documentation, see [ht]docs/manual/install.html or
http://httpd.apache.org/docs/2.4/install.html

   $ ./configure --prefix=PREFIX
   $ make
   $ make install
   $ PREFIX/bin/apachectl start
```

- 配置：./configure
- 编译：make
- 安装：make install
- 运行：PREFIX/bin/apachectl start

配置

- 源代码需要编译成二进制代码再进行安装，自动编译需要Makefile文件。
- 编译配置文件Makefile由configure生成。
- configure是一个Bash脚本，该脚本扫描系统，以确保程序需要的所有库文件已存在，并进行文件安装
- 路径和其它的一些设置工作。

--prefix=DIR：最常用的选项，指定安装路径

--exec-prefix=DIR：指定可执行文件安装路径

--datadir=DIR：指定通用数据文件的安装位置

--libdir=DIR：指定库文件的安装位置

--includedir=DIR：指定头文件的安装位置

编译和安装

- 编译：make

  make会依据Makefile中的设置对源代码进行编译生成可执行的二进制文件。

- 安装：make install

  make根据Makefile文件中关于install目标的设置，将上一步所编译完成的二进制文件、库和配置文件等安装到指定的目录中。

安装Apache服务器

下载地址： http://httpd.apache.org/download.cgi

不安全 | httpd.apache.org/download.cgi

- General Information
- Trunk
- 2.4

Documentation

- Version 2.4
- Trunk (dev)
- Wiki

Get Involved

- Mailing Lists
- Bug Reports
- Developer Info
- User Support

Subprojects

- Docs
- Test
- Flood
- libapreq
- Modules
- mod_fcgid
- mod_ftp

Mirror

The currently selected mirror is **http://mirrors.tuna.tsinghua.edu.cn/apache/**. If you encounter a prol

Other mirrors: http://mirrors.tuna.tsinghua.edu.cn/apache/ Change
You may also consult the complete list of mirrors.

Apache HTTP Server 2.4.41 (httpd): 2.4.41 is the latest available version

The Apache HTTP Server Project is pleased to announce the release of version 2.4.41 of the Apache innovation by the project, and is recommended over all previous releases!

For details, see the Official Announcement and the CHANGES_2.4 and CHANGES_2.4.41 lists.

- Source: httpd-2.4.41.tar.bz2 [PGP] [SHA256]
- Source: httpd-2.4.41.tar.gz [PGP] [SHA256]
- Binaries
- Security and official patches
- Other files
- Files for Microsoft Windows

配置

```
NOTES: * Replace PREFIX with the filesystem path under which
         Apache should be installed.  A typical installation
         might use "/usr/local/apache2" for PREFIX (without the
         quotes).

       * Consider if you want to use a previously installed APR and
         APR-Util (such as those provided with many OSes) or if you
         need to use the APR and APR-Util from the apr.apache.org
         project. If the latter, download the latest versions and
         unpack them to ./srclib/apr and ./srclib/apr-util (no
         version numbers in the directory names) and use
         ./configure's --with-included-apr option. This is required
         if you don't have the compiler which the system APR was
         built with.  It can also be advantageous if you are a
         developer who will be linking your code with Apache or using
         a debugger to step through server code, as it removes the
         possibility of version or compile-option mismatches with APR
         and APR-Util code. As a convenience, prepackaged source-code
         bundles of APR and APR-Util are occasionally also provided
         as a httpd-2.X.X-deps.tar.gz download.
```

1. configure 使用选项--prefix=/usr/local/apache2 --with-include-apr。

2. 下载apr和apr-util并分别解压到./srclib/apr和./srclib/apr-util中。

下载apr和apr-util

下载地址： https://apr.apache.org/download.cgi

https://apr.apache.org/download.cgi

Download!

- from a mirror

APR Docs

- Version 1.7
- Trunk APR 2.0 (dev preview)

APR-util Docs

- Version 1.6
- Trunk APR 2.0 (dev preview)

APR-iconv Docs

- Version 1.2
- Trunk (dev preview)

Guidelines

- Project Guidelines
- Contributing
- Version Numbers

Miscellaneous

Mirror

The currently selected mirror is **http://mirrors.tuna.tsinghua.edu.cn/apache/**.
backup mirrors (at the end of the mirrors list) that should be available.

Other mirrors: http://mirrors.tuna.tsinghua.edu.cn/apache/ Change

You may also consult the complete list of mirrors.

APR 1.7.0 is the best available version

APR is the base portability library.

- Unix Source: apr-1.7.0.tar.gz [PGP] [SHA256]
- Unix Source: apr-1.7.0.tar.bz2 [PGP] [SHA256]
- Win32 Source: apr-1.7.0-win32-src.zip [PGP] [SHA256]
- Other files

APR-util 1.6.1 is the best available version

APR-util provides a number of helpful abstractions on top of APR.

- Unix Source: apr-util-1.6.1.tar.gz [PGP] [SHA256]
- Unix Source: apr-util-1.6.1.tar.bz2 [PGP] [SHA256]
- Windows Source: apr-util-1.6.1-win32-src.zip [PGP] [SHA256]
- Other files

解压复制

下载apr和apr-util并分别解压到./srclib/apr和./srclib/apr-util中。

```
tar zxvf apr-1.7.0.tar.gz
tar zxvf apr-util-1.6.1.tar.gz
cp -r  apr-util-1.6.1 ./httpd-2.4.41/srclib/apr-util
cp -r  apr-1.7.0 ./httpd-2.4.41/srclib/apr
```

```
lei@lei-VirtualBox:~$ pwd
/home/lei
lei@lei-VirtualBox:~$ ls
apr-1.7.0          apr-util-1.6.1.tar.gz  httpd-2.4.41.tar.gz  视频  下载
apr-1.7.0.tar.gz   examples.desktop       公共的               图片  音乐
apr-util-1.6.1     httpd-2.4.41           模板                 文档  桌面
lei@lei-VirtualBox:~$ cp -r  apr-util-1.6.1 ./httpd-2.4.41/srclib/apr-util
lei@lei-VirtualBox:~$ cp -r  apr-1.7.0 ./httpd-2.4.41/srclib/apr
```

```
lei@lei-VirtualBox:~$ cd httpd-2.4.41/srclib/
lei@lei-VirtualBox:~/httpd-2.4.41/srclib$ ls
apr  apr-util  Makefile.in
```

执行configure

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

```
lei@lei-VirtualBox:~/httpd-2.4.41$ ./configure --prefix=/usr/local/apache2
checking for chosen layout... Apache
checking for working mkdir -p... yes
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
configure:
configure: Configuring Apache Portable Runtime library...
```

```
configure: error: in `/home/lei/httpd-2.4.41/srclib/apr':
configure: error: no acceptable C compiler found in $PATH
See `config.log' for more details
configure failed for srclib/apr
```

安装gcc

sudo apt install gcc

```
configure: error: pcre-config for libpcre not found. PCRE is required and availa
ble from http://pcre.org/
```

安装PCRE

下载地址：ftp://ftp.pcre.org/pub/pcre/

不安全 | ftp://ftp.pcre.org/pub/pcre/

pcre-8.41.zip.sig	310 B	2017/7/5 上午8:00:00
pcre-8.42.tar.bz2	1.5 MB	2018/3/20 上午8:00:00
pcre-8.42.tar.bz2.sig	310 B	2018/3/20 上午8:00:00
pcre-8.42.tar.gz	2.0 MB	2018/3/20 上午8:00:00
pcre-8.42.tar.gz.sig	310 B	2018/3/20 上午8:00:00
pcre-8.42.zip	2.2 MB	2018/3/20 上午8:00:00
pcre-8.42.zip.sig	310 B	2018/3/20 上午8:00:00
pcre-8.43.tar.bz2	1.5 MB	2019/2/23 上午8:00:00
pcre-8.43.tar.bz2.sig	310 B	2019/2/23 上午8:00:00
pcre-8.43.tar.gz	2.0 MB	2019/2/23 上午8:00:00
pcre-8.43.tar.gz.sig	310 B	2019/2/23 上午8:00:00
pcre-8.43.zip	2.2 MB	2019/2/23 上午8:00:00
pcre-8.43.zip.sig	310 B	2019/2/23 上午8:00:00

安装PCRE

解压：tar zxvf pcre-8.43.tar.gz

./configure

```
checking windows.h usability... no
checking windows.h presence... no
checking for windows.h... no
configure: error: Invalid C++ compiler or C++ compiler flags
```

安装g++：sudo apt install g++

再次执行./configure

```
config.status: error: Something went wrong bootstrapping makefile fragments
    for automatic dependency tracking.  Try re-running configure with the
    '--disable-dependency-tracking' option to at least be able to build
    the package (albeit without support for automatic dependency tracking).
See `config.log' for more details
```

根据提示执行：./configure --disable-dependency-tracking

安装PCRE

安装make：sudo apt install make

执行make：make

```
lei@lei-VirtualBox:~/pcre-8.43$ make
rm -f pcre_chartables.c
ln -s ./pcre_chartables.c.dist pcre_chartables.c
make  all-am
make[1]: 进入目录“/home/lei/pcre-8.43”
  CC       pcretest-pcretest.o
  CC       pcretest-pcre_printint.o
  CC       libpcre_la-pcre_byte_order.lo
  CC       libpcre_la-pcre_compile.lo
  CC       libpcre_la-pcre_config.lo
```

```
  CXX      pcre_scanner_unittest-pcre_scanner_unittest.o
  CXXLD    pcre_scanner_unittest
  CXX      pcre_stringpiece_unittest-pcre_stringpiece_unittest.o
  CXXLD    pcre_stringpiece_unittest
make[1]: 离开目录“/home/lei/pcre-8.43”
```

安装：sudo make install

安装httpd

./configure --prefix=/usr/local/apache2 --with-include-apr

```
config.status: executing default commands
configure: summary of build options:

    Server Version: 2.4.41
    Install prefix: /usr/local/apache2
    C compiler:     gcc
    CFLAGS:          -g -O2 -pthread
    CPPFLAGS:        -DLINUX -D_REENTRANT -D_GNU_SOURCE
    LDFLAGS:
    LIBS:
    C preprocessor: gcc -E
```

make

```
xml/apr_xml.c:35:10: fatal error: expat.h: 没有那个文件或目录
 #include <expat.h>
          ^~~~~~~~~
compilation terminated.
/home/lei/apr-util-1.6.1/build/rules.mk:206: recipe for target 'xml/apr_xml.lo'
failed
make[1]: *** [xml/apr_xml.lo] Error 1
make[1]: 离开目录“/home/lei/apr-util-1.6.1”
/home/lei/apr-util-1.6.1/build/rules.mk:118: recipe for target 'all-recursive' f
ailed
make: *** [all-recursive] Error 1
```

sudo apt-get install libexpat1-dev

安装httpd

再次执行make：make

```
ar: `u' modifier ignored since `D' is the default (see `U')
make[4]: 离开目录“/home/lei/httpd-2.4.41/modules/mappers”
make[3]: 离开目录“/home/lei/httpd-2.4.41/modules/mappers”
make[2]: 离开目录“/home/lei/httpd-2.4.41/modules”
make[2]: 进入目录“/home/lei/httpd-2.4.41/support”
make[2]: 离开目录“/home/lei/httpd-2.4.41/support”

make[1]: 离开目录“/home/lei/httpd-2.4.41”
```

安装：sudo make install

```
Installing header files
Installing build system files
Installing man pages and online manual
mkdir /usr/local/apache2/man
mkdir /usr/local/apache2/man/man1
mkdir /usr/local/apache2/man/man8
mkdir /usr/local/apache2/manual
make[1]: 离开目录“/home/lei/httpd-2.4.41”
```

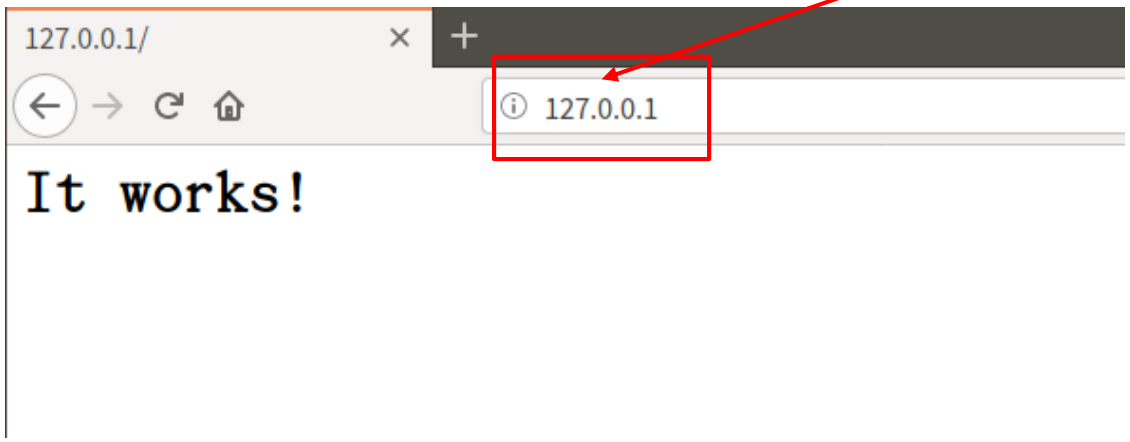
启动APACHE服务器

切换到apache安装目录的bin目录：cd /usr/local/apache2/bin/

执行：sudo ./apachectl start

```
lei@lei-VirtualBox:/usr/local/apache2/bin$ sudo ./apachectl start
AH00558: httpd: Could not reliably determine the server's fully qualified domain
 name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this
 message
```

在浏览器中输入127.0.0.1



去掉警告

```
lei@lei-VirtualBox:/usr/local/apache2/bin$ sudo ./apachectl start
AH00558: httpd: Could not reliably determine the server's fully qualified domain
 name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this
 message
```

修改配置文件：/usr/local/apache2/conf/httpd.conf

```
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName 127.0.0.1:80
```

重新启动服务

```
lei@lei-VirtualBox:/usr/local/apache2/bin$ sudo ./apachectl start
lei@lei-VirtualBox:/usr/local/apache2/bin$ ps aux | grep httpd
root       8340  0.0  0.0  73204  3740 ?        Ss   12:42   0:00 /usr/local/apache2/bin/httpd -k start
daemon     8341  0.0  0.1 825308  4300 ?        Sl   12:42   0:00 /usr/local/apache2/bin/httpd -k start
daemon     8342  0.0  0.1 825308  4300 ?        Sl   12:42   0:00 /usr/local/apache2/bin/httpd -k start
daemon     8343  0.0  0.1 825308  4300 ?        Sl   12:42   0:00 /usr/local/apache2/bin/httpd -k start
lei        8426  0.0  0.0  21532  1092 pts/1    S+   12:42   0:00 grep --color=auto httpd
```

编译httpd时报错

make

```
collect2: error: ld returned 1 exit status
Makefile:48: recipe for target 'htpasswd' failed
make[2]: *** [htpasswd] Error 1
make[2]: 离开目录“/home/lei/httpd-2.4.41/support”
/home/lei/httpd-2.4.41/build/rules.mk:75: recipe for target 'all-recursive' failed
make[1]: *** [all-recursive] Error 1
make[1]: 离开目录“/home/lei/httpd-2.4.41/support”
/home/lei/httpd-2.4.41/build/rules.mk:75: recipe for target 'all-recursive' failed
make: *** [all-recursive] Error 1
```

这个错误是因为执行configure 时没有加上选项--with-include-apr。

启动服务时找不到共享库

```
lei@lei-VirtualBox:/usr/local/apache2/bin$ ./apachectl start
/usr/local/apache2/bin/httpd: error while loading shared libraries: libpcre.so.1: cannot open
 shared object file: No such file or directory
lei@lei-VirtualBox:/usr/local/apache2/bin$ ldd  httpd
        linux-vdso.so.1 (0x00007ffd75fb9000)
        libpcre.so.1 => not found
        libaprutil-1.so.0 => /usr/local/apache2/lib/libaprutil-1.so.0 (0x00007f29460fc000)
        libapr-1.so.0 => /usr/local/apache2/lib/libapr-1.so.0 (0x00007f2945ec1000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f2945ca2000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f29458b1000)
        libexpat.so.1 => /lib/x86_64-linux-gnu/libexpat.so.1 (0x00007f294567f000)
        libcrypt.so.1 => /lib/x86_64-linux-gnu/libcrypt.so.1 (0x00007f2945447000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f2945243000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f29465d1000)
```

Linxu下的共享库采用了类似于高速缓存的机制，将库信息保存在/etc/ld.so.cache文件中，程序连接时首先从这个文件中查找，新装了库，需要使用ldconfig命令来更新ld的cache，使得新安装的的库可以使用。

执行ldconfig命令

sudo ldconfig

```
lei@lei-VirtualBox:/usr/local/apache2/bin$ sudo ldconfig
lei@lei-VirtualBox:/usr/local/apache2/bin$ ldd  httpd
        linux-vdso.so.1 (0x00007ffffe113000)
        libpcre.so.1 => /usr/local/lib/libpcre.so.1 (0x00007ff32d61f000)
        libaprutil-1.so.0 => /usr/local/apache2/lib/libaprutil-1.so.0 (0x00007ff32d3f5000)
        libapr-1.so.0 => /usr/local/apache2/lib/libapr-1.so.0 (0x00007ff32d1ba000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007ff32cf9b000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007ff32cbaa000)
        libexpat.so.1 => /lib/x86_64-linux-gnu/libexpat.so.1 (0x00007ff32c978000)
        libcrypt.so.1 => /lib/x86_64-linux-gnu/libcrypt.so.1 (0x00007ff32c740000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007ff32c53c000)
        /lib64/ld-linux-x86-64.so.2 (0x00007ff32dae8000)
```

命令安装apache2

安装Apache2

sudo apt-get install apache2

启动Apache2

sudo systemctl start apache2

停止

sudo systemctl stop apache2

重启

sudo systemctl restart apache2

安装PHP

更新依赖sudo apt-get update

安装php

sudo apt-get install libapache2-mod-php

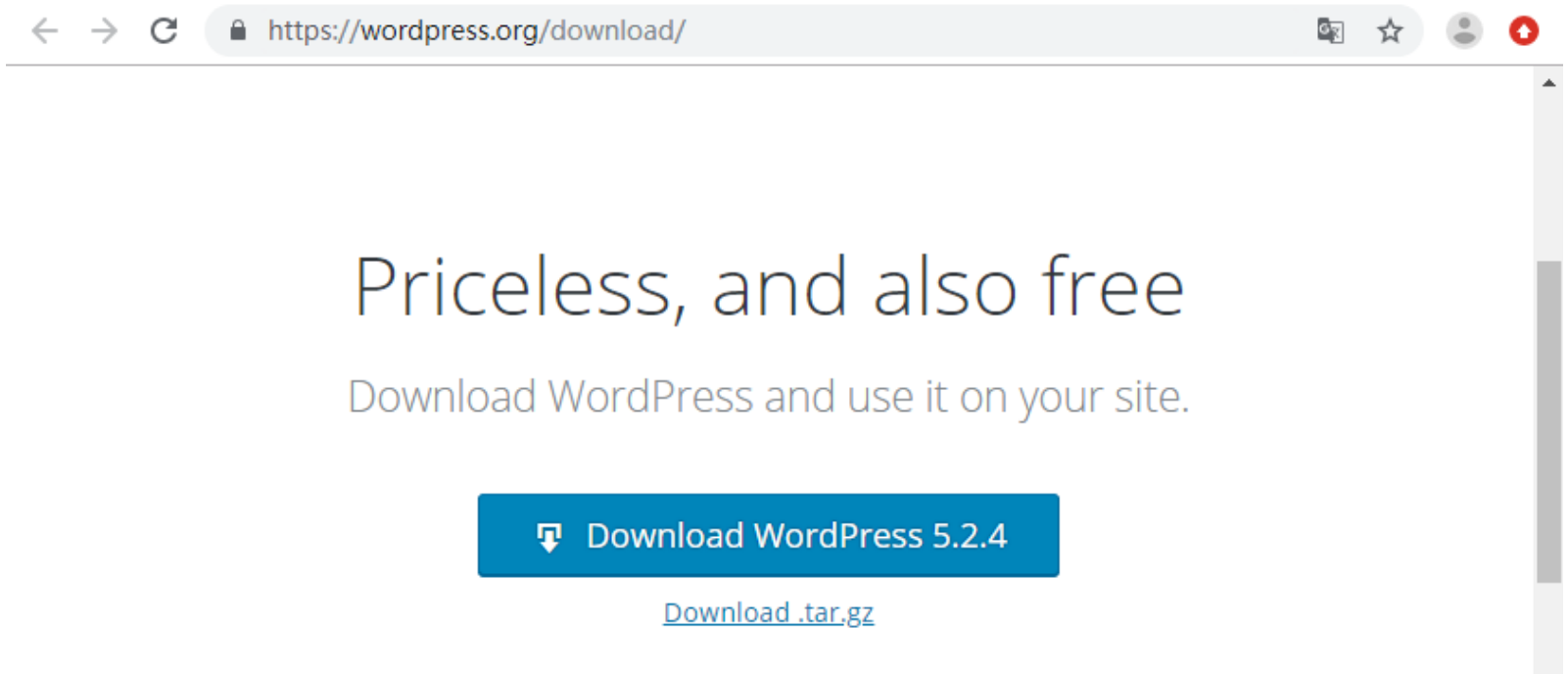
重启Apache2

sudo systemctl restart apache2

sudo apt-get install php-gd php-mysql

下载WordPress

https://wordpress.org/download/



安装WordPress

将下载的压缩包解压复制到/var/www/html

```
lei@lei-VirtualBox:~/www$ cd /var/www/html/
lei@lei-VirtualBox:/var/www/html$ ls
hello.php  index.html  wordpress
```

修改 WordPress 配置文件

cd /var/www/html/wordpress

cp w-config-sample.php wp-config.php

安装WordPress

cd /var/www/html/wordpress

cp w-config-sample.php wp-config.php

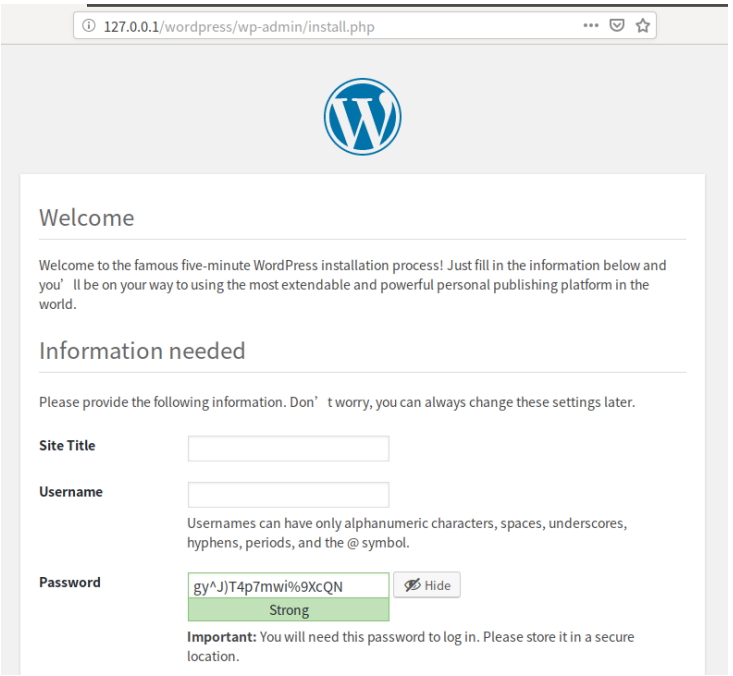
vim wp-config.php

修改 WordPress 配置文件，找到文件中 MySQL 的部分，并将相关配置信息修改为配置WordPress数据库中的内容。

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'user' );

/** MySQL database password */
define( 'DB_PASSWORD', '123456' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
```

访问WordPress

重启Apache2

sudo systemctl restart apache2

在浏览器里输入127.0.0.1/wordpress