

《同儿辈赋未开海棠》元好问

枝间新绿一重重, 小蕾深藏数点红。 爱惜芳心莫轻吐, 且教桃李闹春风。



在Ubuntu安装新软件,你使用过或者熟悉的两种方法是()?

- A Ubuntu软件中心
- □ 命令行工具apt
- c 新立得Synaptic
- D 二进制软件包安装
- E 源代码安装

第6讲 软件包管理

主讲:王洪泊



本章学习要点

软件包:应用程序、配置文件以及数据等支持文件。

- (1) Ubuntu软件安装方式(包格式为Deb)
- (2) APT工具
- (3) PPA安装
- (4) 软件包文件安装
- (5) 源代码安装



6.1 Linux软件安装基础

6.1.1 Linux软件包管理的发展过程

● 从源代码安装软件

- 早期Linux主要使用源代码包发布软件。
- 直接将源代码编译成二进制文件,并对系统进行相关配置。

● 使用软件包安装软件

- 软件包将应用程序的二进制文件、配置文档和帮助文档等合并打包在一个文件中。
- 主流的软件包格式有两种: RPM和Deb。

● 高级软件包管理工具

- Yum (Yellow dog Updater, Modified): 基于RPM包的软件包管理器。
- APT (Advanced Packaging Tools): Debian及其派生发行版(如Ubuntu)的软件包管理器。



6.1 Linux 软件安装基础

6.1.2 Ubuntu软件安装方式

在线安装

Ubuntu 软件中心

APT工具 (apt/Synaptic)

PPA安装(个人 软件包档案)

离线安装

Dpkg工具 (获得deb安装包)

对于RPM包,alien 工具转换为deb包, 用dpkg进行安装

命令行方式(二进制 包bin或run)

●源代码安装

- ❖ 可定制
- ❖ 编译安装
- ❖ 最复杂
- ❖ 难度大
- ❖ 有挑战性



6.2.1 理解APT

• APT三项基本功能

- a) 从Internet上的软件源下载最新的**软件包元数据**、二进制包或源 代码包。
- b) 利用下载到本地的**软件包元数据**,完成软件包的搜索和系统的 更新。
- c) 安装和卸载软件包时,自动寻找最新版本,并自动解决**软件的依 赖关系**。



6.2.1 理解APT

■ APT软件源

- Ubuntu安装时已提供Ubuntu官方的网络安装来源。
- 可以使用系统安装光盘作为安装源,或从非官方的软件源中下载 非官方的软件。
- /var/lib/apt/lists目录存放的是已经下载的各软件源的元数据。

● 解决依赖关系

■ 多数情况下APT会安装最新的软件包,被安装的软件包所依赖的 其他软件包也会安装。

● 软件包更新

- APT可以<u>智能地从软件源下载最新版本的软件并且安装</u>。?
- APT还支持Ubuntu(或Debian)从一个发布版本升级到新的发布版本。



软件包自动更新那些事

❖/var/lib/apt/lists目录存放的是已经下载的各软件 源的元数据。

```
选择hong@LAPTOP-DTA5818O: /var/lib/apt/lists
security.ubuntu.com_ubuntu_dists_xenial-security_restricted_i18n_Translation-en
 ong@LAPTOP-DTA58180:/var/lib/apt/lists$ 1s -al
total 16488
                          512 Feb 29 16:16
drwxr-xr-x 1 root root
                          512 Feb 29 16:16
drwxr-xr-x 1 root root
         - 1 root root 246846 Feb 29 16:16 archive.ubuntu.com_ubuntu_dists_xenial_InRelease,
          1 root root 7228243 Apr 21 2016 archive.ubuntu.com ubuntu dists xenial main binary-amd64 Packages
         - 1 root root 3600400 Apr 21  2016 archive.ubuntu.com ubuntu dists xenial main i18n Translation-en
           root root 127112 Apr 15 2016 archive. ubuntu. com ubuntu dists xenial restricted binary-amd64 Packages
           root root 20517 Apr 15 2016 archive.ubuntu.com ubuntu dists xenial restricted i18n Translation-en
         - 1 root root 109196 Feb 29 16:16 archive.ubuntu.com ubuntu dists xenial-updates InRelease
         - 1 root root 7821340 Feb 29 14:30 archive.ubuntu.com_ubuntu_dists_xenial-updates_main_binary-amd64_Packages
           root root 7050671 Feb 24 23:00 archive. ubuntu. com ubuntu dists xenial-updates main i18n Translation-en
            root root 104448 Mar 17 2019 archive ubuntu com ubuntu dists xenial-updates restricted binary-amd64 Packages
                        15916 Mar 15 2019 archive.ubuntu.com ubuntu dists xenial-updates restricted i18n Translation-en
                            0 Feb 29 16:16 lock
                          512 Feb 29 16:16 partial
        - 1 root root 109198 Feb 29 16:16 security.ubuntu.com ubuntu dists xenial-security InRelease
         - 1 root root 5921859 Feb 26 20:36 security.ubuntu.com ubuntu dists xenial-security main binary-amd64 Packages
          1 root root 5954973 Feb 24 22:18 security. ubuntu. com ubuntu dists xenial-security main i18n Translation-en
      -r-- 1 root root 102468 May 30 2018 security.ubuntu.com ubuntu dists xenial-security restricted binary-amd64 Packages
rw-r--r- 1 root root 15564 May 30 2018 security.ubuntu.com_ubuntu_dists_xenial-security_restricted_i18n_Translation-en
 ong@LAPTOP-DTA58180:/var/lib/apt/lists$ _
```



软件包自动更新那些事

❖/var/lib/dpkg/status目录存放的是已经下载的各 软件版本信息。

```
hong@LAPTOP-DTA5818O: /var/lib/dpkg
ong@LAPTOP-DTA58180:/var/lib/dpkg$ ls -al
otal 720
                          512 Feb 29 16:45
drwxr-xr-x 1 root root
                          512 Feb 29 16:23
drwxr-xr-x 1 root root
                          512 Feb 29 16:45 alternatives
drwxr-xr-x 1 root root
     -r-- 1 root root 170080 Feb 29 16:12 available
                            8 Feb 29 16:12 cmethopt
                          434 Feb 29 16:23 diversions
486 Feb 29 16:23 diversions-old
      -r-- 1 root root
        - 1 root root
                          512 Feb 29 16:45 inf
                           0 Feb 29 16:45 lock
                            0 Feb 29 16:14 lock-frontend
                          512 Tan 13 2016
                          135 Feb 29 16:16 statoverride
      -r-- 1 root root 501020 Feb 29 16:45 status
        - 1 root root 501020 Feb 29 16:45 status-old
                          512 Feb 29 16:45 triggers
                          512 Feb 29 16:45 updates
drwxr-xr-x 1 root root
ong@LAPTOP-DTA58180:/var/lib/dpkg$ _
```



实现软件自动在线升级的原理

写两个程序,一个是主程序;一个是升级程序;所有升级任务都由升级程序完成。

- 1. 启动升级程序, 升级程序连接到网站, 下载新的主程序(也包括支持的库文件、XML配置文档等)到临时文件夹;
- 2. 升级程序获取服务器端XML配置文件中新版本程序更新日期、版本号、文件大小;
- 3. 升级程序获取原有客户端应用程序的最近一次更新日期或版本号或文件大小,两者进行比较;
- ❖ 如果发现升级程序的日期大于原有程序的最新日期,则提示用户是否升级;
- ❖ 或者是采用将现有版本与最新版本作比较,发现最新的则提示用户是否升级;
- ❖ 或者用其它属性如文件大小进行比较,发现升级程序的文件大小大于旧版本的程序的大小则提示用户升级。
- ❖ 也可以采用比较新旧版本更新日期号来提示用户升级。
- 4. 如果用户选择升级,则获取下载文件列表,开始进行批量下载文档;
- 5. 升级程序检测旧的主程序是否活动,若活动则关闭旧的主程序;
- 6. 删除旧的主程序, 拷贝临时文件夹中的文件到相应的位置;
- 7. 检查主程序的状态,若状态为活动的,则启动新的主程序;
- 8. 关闭升级程序,升级完成。





6.2 APT<u>工</u>具

6.2.2 APT命令行工具的使用

- 使用apt-cache命令查询软件包
 - 执行pkgnames子命令列出当前所有可用的软件包
 - 使用子命令search查找使用参数定义的软件包并列出该软件包的 相关信息,参数可以使用正则表达式。
 - 使用子命令show可以查看指定名称的软件包的详细信息。
 - 使用子命令depends可以查看软件包所依赖的软件包。
 - 使用子命令rdepends可以查看软件包被哪些软件包所依赖。
 - 使用子命令showpkg查看软件包的依赖关系信息。
 - 使用policy子命令显示软件包的安装状态和版本信息。



6.2.2 APT命令行工具的使用

● 使用apt-get命令操作软件包

子 命 令	说明			
apt-get update	获取最新的软件包列表,同步/etc/apt/sources.list和/etc/apt/sources.list.d中列出的源的索引,以确保用户能够获取最新的软件包			
apt-get upgrade	更新当前系统中所有已安装的软件包,并同时所更新的软件包相关的所依赖的软件包			
apt-get install	下载、安装软件包并自动解决依赖关系			
apt-get remove	卸载指定的软件包			
apt-get autoremove	自动卸载所有未使用的软件包			
apt-get purge	卸载指定的软件包及其配置文件			
apt-get source	下载软件包的源代码			
apt-get clean	清理已下载的软件包,实际上是清除/var/cache/apt/archives 目录中的软件包,不会影响软件的正常使用			
apt-get autoclean	删除已卸载的软件的软件包备份			



6.2.2 APT命令行工具的使用

● 安装软件包

- 在每次安装和更新软件包之前先执行apt-get update更新系统中 apt-get缓存中的软件包信息。
- 执行apt-get install命令安装指定名称的软件包。
- 安装过程中APT为用户提供了大量信息,自动分析并解决了软件 包依赖问题。®hong@laPTOP-DTAS8180:-



6.2.2 APT命令行工具的使用

● 卸载软件包

- 执行apt-get remove命令卸载一个已安装的软件包,但会保留该 软件包的配置文档。
- 要同时删除配置文件,则要执行apt-get purge命令。
- 更彻底的删除,可执行以下命令删除与该软件包及其所依赖的、 不再使用的软件包:

sudo apt-get autoremove 软件包名



6.2.3 配置APT源

● 升级软件包

- 执行apt-get upgrade命令会升级本地已安装的所有软件包。
- 执行命令apt-get dist-upgrade则可以识别出依赖关系改变的情形 并作出相应处理,会尝试升级最重要的包。
- 执行apt-get upgrade命令时加上-u选项可以让APT显示完整的可 更新软件包列表。
- 使用选项-s模拟升级软件包。
- 只想对某一具体的软件包进行升级,可以执行安装软件包命令时加上--reinstall选项。sudo apt-get --reinstall install 软件包名



6.2.3 配置APT源

- Ubuntu使用文本文件/etc/apt/sources.list来保存软件包的安装和 更新源的地址。
- 与该文件功能相同的是/etc/apt/sources.list.d/目录下的.list文件,为在单独文件中写入安装源的地址提供了一种方式,通常用来安装第三方软件。
- 可以通过修改/etc/apt/sources.list文件来更改APT源。
- 首先备份源列表,然后使用文本编辑器打开进行编辑,从列表中 选择合适的源,替换掉文件中所有的内容,保存编辑好的文件。



6.2.4 使用新立得软件包管理器

● 简介

- 新立得软件包管理器(Synaptic Package Manager)是APT管理 工具的图形化前端。
- 可以在图形界面中通过鼠标操作就能安装、删除、配置、升级软件包,对软件包列表进行浏览、排序、搜索以及管理软件仓库, 甚至升级整个系统。
- 它相当于终端中运行的apt命令。



6.2.4 使用新立得软件包管理器

● 主界面





- 安装软件包
 - 标记要安装的软件包





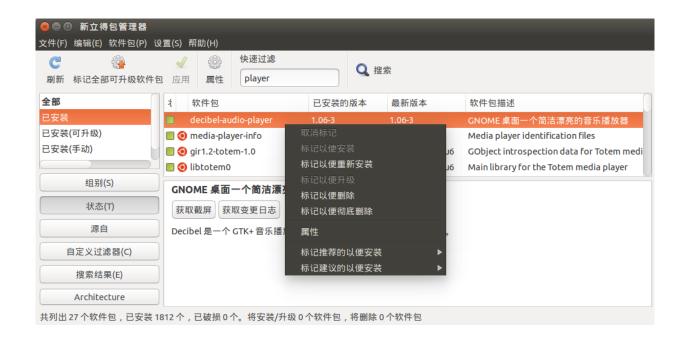
- 安装软件包
 - 安装过程







- 其他操作
 - 操作菜单





- 新立得配置
 - 设置首选项





- 新立得配置
 - 设置软件更新

❷ □ □ 软件和更新						
Ubuntu 软件 其它软件	更新	身份验证	附加驱动			
从下列地点安装更新:						
☑ 重要安全更新 (utopic-security)						
☑ 推荐更新 (utopic-updates)						
☐ 提前释放出的更新 (utopic-proposed)						
☑ 不支持的更新 (utopic-backports)						
自动检查更新:	每天			▼		
当有安全更新时:	立即显示			•		
当有其它更新时:	毎周显示一次			•		
有新版本时通知我:	适用任	何新版本		•		
		j	还原(V)	关闭(C)		



6.2.5 PPA安装

● 管理PPA源

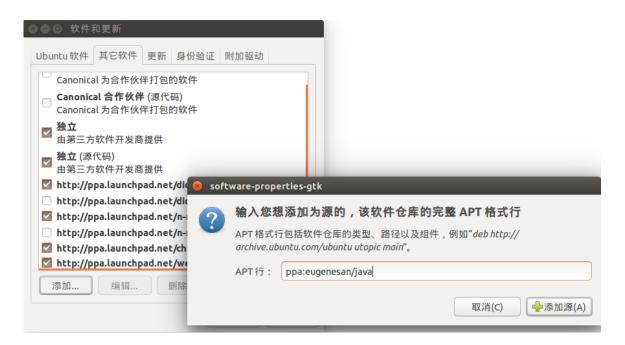
- PPA源的语法格式为:
 - ppa:user/ppa-name
- 具体软件包的表示可到launchpad.net网站查看。
- 添加PPA源的命令为:
 - sudo add-apt-repository ppa:user/ppa-name
- 删除PPA源的命令为:
 - sudo add-apt-repository -r ppa:user/ppa-name



6.2.5 PPA安装

● 管理PPA源

■ 通过图形界面的软件源设置来添加或删除PPA源





6.2.5 PPA安装

● 通过PPA源安装软件

- 很多PPA软件源提供多个版本,包括正式稳定版、每日创建版, 开发版、测试版等。
- 通过PPA源安装软件的步骤
 - (1)添加该安装源
 - (2) 更新系统软件源
 - (3) 安装软件



6.3.1 Deb软件包管理

● 查看Deb软件包

- 使用选项-I列出软件包的简要信息: dpkg -I 软件包名
- 使用选项-s来查看软件包状态的详细信息: dpkg -s 软件包名
- 使用选项-S获知已安装的软件包拥有哪些文件: dpkg -S 软件包名



6.3.1 Deb软件包管理

● 安装Deb软件包

- 所有的软件包安装之前必须保证所依赖的库和软件已经安装到系统上。
- 首先要获取Deb软件包文件,然后使用选项-i安装Deb软件包: sudo dpkg -i 软件包文件名

● 卸载Deb软件包

- 使用选项-r卸载软件包同时会保留该软件的配置信息: sudo dpkg -r 软件包名
- 要将配置信息一并删除,应使用选项-P: sudo dpkg -P 软件包名
- 使用dpkg工具卸载软件包不会自动解决依赖性问题。



6.3.2 RPM软件包管理

- RPM软件包的名称具有特定的格式: 软件名称-版本号(包括主版本和次版本号).软件运行的硬件平台 .rpm
- Ubuntu的软件包格式是.deb,不应当直接安装RPM包。
- 安装RPM包,要先用alien工具将.rpm转换成.deb格式,然后再使用dpkg工具安装.deb文件。



6.3.3 . run与. bin二进制包软件包安装

• .run软件包安装

- run安装包实质上是由一个安装脚本加上要安装的程序合成的一个 文件,文件名后缀为.run。
- 首先要为该文件增加可执行属性: sudo chmod +x 文件名.run
- 然后在终端中执行该文件: sudo ./文件名.run



6.3.3 . run与. bin二进制包软件包安装

- 卸载使用run安装包安装的软件
 - 到安装目录中查看帮助文档,通常该目录中会提供反安装脚本 uninstall,可以切换到该安装目录执行此脚本。

sudo ./uninstall

- 如果不提供该脚本,则会提供其他维护或卸载工具
- .bin软件包安装
 - 在执行.bin文件之前,也需要赋予执行权限,安装与卸载过程可以参照run安装包。



6.4 使用源代码安装

6.4.1 源代码安装的基本步骤

- 下载和解压软件包
 - Linux、Unix最新的软件通常以源代码打包形式发布,最常见的是.tar.bz2和.tar.gz这种两种压缩包格式。
- 執行configure生成编译配置文件Makefile
 - 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,在源代码包中使用configure命令生成。
- 执行make命令编译源代码
 - make会依据 Makefile文件中的设置对源代码进行编译并生成可执行的二进制文件。
- 执行make install安装软件
 - make根据Makefile文件中关于install目标的设置,将上一步骤所编译完成的二进制文件、库和配置文件等安装到预定的目录中。



6.4 使用源代码安装

6.4.2 源代码安装示例——Apache服务器

● 安装Apache

- (1) 将源代码包文件复制到用户主目录中,对其解压缩。
- (2) 阅读其中的INSTALL和README文件,了解安装事项。
- (3)下载APR源代码包(例中为apr-1.5.2.tar.gz),并进行安装。
- (4)到官网www.apache.org上下载APR-util的源代码包(例中为apr-util-1.5.4.tar.gz),并进行安装。
 - (5) 到官网pcre.org上下载PCRE的源代码包,进行安装。
 - (6) 切换到http-2.4.12目录执行configure脚本,成功生成Makefile。
 - (7)继续运行make命令,完成源代码编译。这一步花费的时间略长。
 - (8) 继续运行sudo make install命令,完成安装。

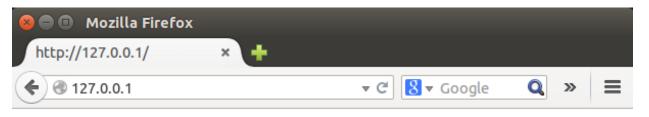


6.4 使用源代码安装

6.4.2 源代码安装示例——Apache服务器

● 测试Apache

- 先执行Idconfig命令,再使用Id检查,发现库文件已经正常识别。
- 接着执行以命令启动Apache。



It works!