

Linux 软件安装与管理入门

杨博远

LUG@USTC

"life, love, linux"

软件如何安装？

- 用源代码编译安装？
- 自己下载“安装包”安装？
- 推荐利用软件源安装软件！
- 什么是软件源？

软件源

- 软件源是Linux系统的应用程序安装仓库，很多应用软件都会收录到这个仓库里面。
- 软件源可以是网络服务器，是光盘，甚至是可以硬盘上的一个目录。
- 中国科学技术大学开源软件镜像站：
- <https://mirrors.ustc.edu.cn/>

使用软件源安装软件的好处

- 方便且易用（只需一条命令）

例（Debian、Ubuntu）

```
sudo apt-get install <软件包名>
```

- 易进行软件更新（与系统更新同时进行）
- 安全有保证（数字签名，远离恶意软件）

- 如何使用已有的软件源？
- 通过发行版的包管理系统

软件管理的重要性

- 在远古的蛮荒时代.....
- 人们通过编译得到的源代码来安装软件，
人肉管理安装好的软件

问题：

人肉管理不可靠

需手动解决依赖问题

编译耗时费力

- 发行二进制包，不用再编译（想想Windows的exe）

问题：

- 来源无法保证，可能引入恶意软件、木马、病毒
- 依赖问题

需要一个软件包管理系统

- 软件包管理系统是在电脑中自动安装、配置、卸载和升级软件包的工具体组合。
- 包管理系统的类型和 Linux 发行版紧密相关:

包管理系统: 后端/前端	采用的发行版	包文件后缀名
dpkg/apt	Debian, Ubuntu及其衍生版, 如Mint等	.deb
rpm/yum	Fedora, RHEL, CentOS, RedFlag, Mandriva 等	.rpm
rpm/ZYpp/ YaST	OpenSUSE	.rpm
pacman	Arch	.pkg.*
Portage	Gentoo	基于源代码

DEB系软件包管理工具的使用

- `dpkg`: DEB系发行版中众多软件包管理工具的后端
- 最初于1993年发布
- 无法直接解决依赖关系, 不推荐直接使用
- 前端: `apt`, `aptitude`, `synaptic` (新立得), `ubuntu`软件中心等

dpkg 常见用法

```
dpkg -i <名字>.deb
```

手动安装deb包

```
dpkg -l
```

查询当前系统所有已安装deb包的信息，
并通过less和grep以执行更复杂的查询工作

```
例: dpkg -l | grep fcitx | less
```

APT

- Advanced Packaging Tools, 简称APT
- APT由以下的几个主要的命令构成:
 - apt-get
 - apt-cache
 - apt-file

在Debian系统中, APT建构于dpkg之上。

APT 的常见用法： 安装与卸载软件

- `sudo apt-get install <软件名>` 安装软件
- `sudo apt-get install <软件名> - - reinstall`
重新安装某软件
- `sudo apt-get remove <软件名>` 卸载软件
- `sudo apt-get purge <软件名>`
卸载某软件，并清理其遗留配置文件
- `apt-cache search <字符串>`
搜索名称含字符串的软件包

使用APT进行软件源管理

- 添加与管理软件源:
- 编辑配置文件:
 - `/etc/apt/sources.list`
 - `/etc/apt/sources.list.d/*.list`
- 配置文件的格式:
`deb [软件源地址] [发行版名字] [包类型]`

使用APT进行软件源管理

- 下面是 Ubuntu 的几行典型配置:
- `deb http://mirrors.ustc.edu.cn/ubuntu/
trusty main`
- `deb-src http://mirrors.ustc.edu.cn/ubuntu
trusty main restricted #这是注释`

- 纵向分类：看软件的性质

- Ubuntu:

main: Canonical公司支持的免费和开源软件

universe: 社区维护的免费和开源软件

restricted: 设备的专有驱动

multiverse: 有版权和合法性问题的软件

- 对Debian来说，则是以下几类：

main: 系统必要和符合自由软件规范的软件包

contrib: 这些软件可以在Debian系统内运行，但多半依赖于某些非自由软件

non-free: 不属于自由软件范畴的软件

- 横向分类：从软件的稳定性/新旧进行分类

以Ubuntu举例：

- 重要安全更新： "security"
- 推荐的更新： "updates"
- 提前释放出的更新： "proposed"
- 不支持的更新： "backports"

第三方软件源的添加

- 1. 找到对应软件源地址，添加至配置文件中：
例：Virtualbox 提供的源：

- ```
deb http://download.virtualbox.org/virtualbox/debian
saucy contrib
```

- 2. 需要下载对应的公钥并安装：

- ```
$ wget -q  
http://download.virtualbox.org/virtualbox/debi  
an/oracle_vbox.asc -O- | sudo apt-key add -
```

- 软件源提供者通常会提供说明

PPA 的添加与使用

- PPA即Personal Package Archives，是一些用户/开发者在Launchpad上编译发布的软件包
- 这些软件源添加时，Ubuntu会自行下载公钥

PPA 的添加与使用

- 首先你要知道想用的PPA的名字和用途
例如 `gnome3-team/gnome3`

添加:

```
sudo add-apt-repository ppa:gnome3-team/gnome3
```

ppa的配置文件一般保存在 `/etc/apt/sources.list.d/` 目录下，一个ppa对应一个文件

第三方软件源的优劣

优点:

- 可以提供主源里没有的软件或更新的版本
- 软件可以随系统一起更新

缺点

珍爱生命，远离PPA！

- 包质量参差不齐，易产生 bug
- 可能会引起依赖问题

APT 的常见用法：软件管理与升级

- `sudo apt-get update` 更新软件包列表信息

- `sudo apt-get upgrade`

仅对已安装的包进行更新

- `sudo apt-get dist-upgrade`

对发行版进行更新，可能有新软件包的安装
与旧软件包的删除（Ubuntu）

发行版升级（Debian）

- `sudo do-release-upgrade`

手动发行版升级（Ubuntu）

APT 的常见用法：其它

- `sudo apt-get autoremove`
- 移除无用的软件包（非手动安装而不被依赖）
- `sudo apt-get autoclean`
- 自动清理无用的本地安装包缓存
- `sudo apt-get clean`
- 清理所有的本地安装包缓存
- 缓存位于 `/var/cache/apt/archives`

注意事项

- 每次编辑配置文件后都应手动运行

`sudo apt-get update`

使配置立即生效。

- 遇到依赖问题怎么办？

选用aptitude替代apt-get，会自动提出解决方案供你选择

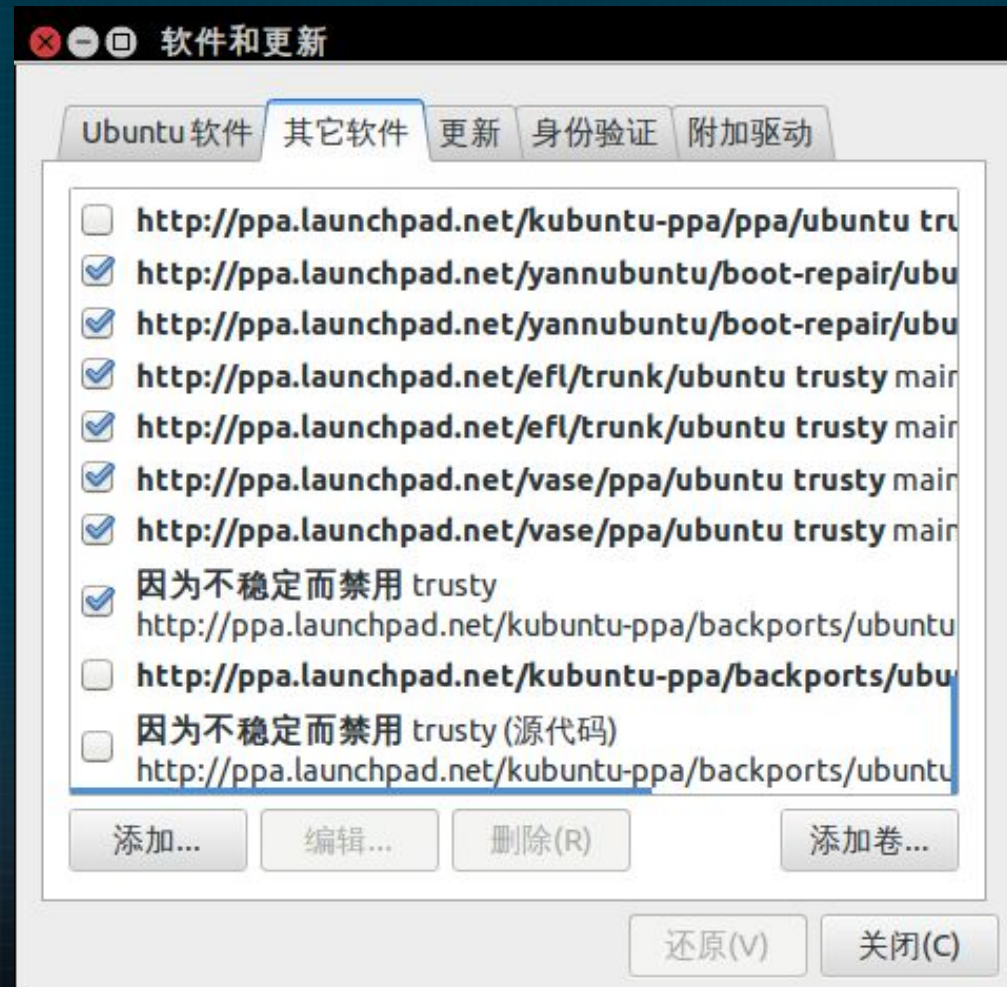
RPM 系发行版的软件管理

- 前端: yum 后端: rpm
- 相关配置放在 `/etc/yum.repos.d/` 下面:
`fedora.repo`、`fedora-updates.repo`、`fedora-updates-testing.repo`
- 由于本人不熟悉, 请大家自行查阅资料

图形界面工具简介

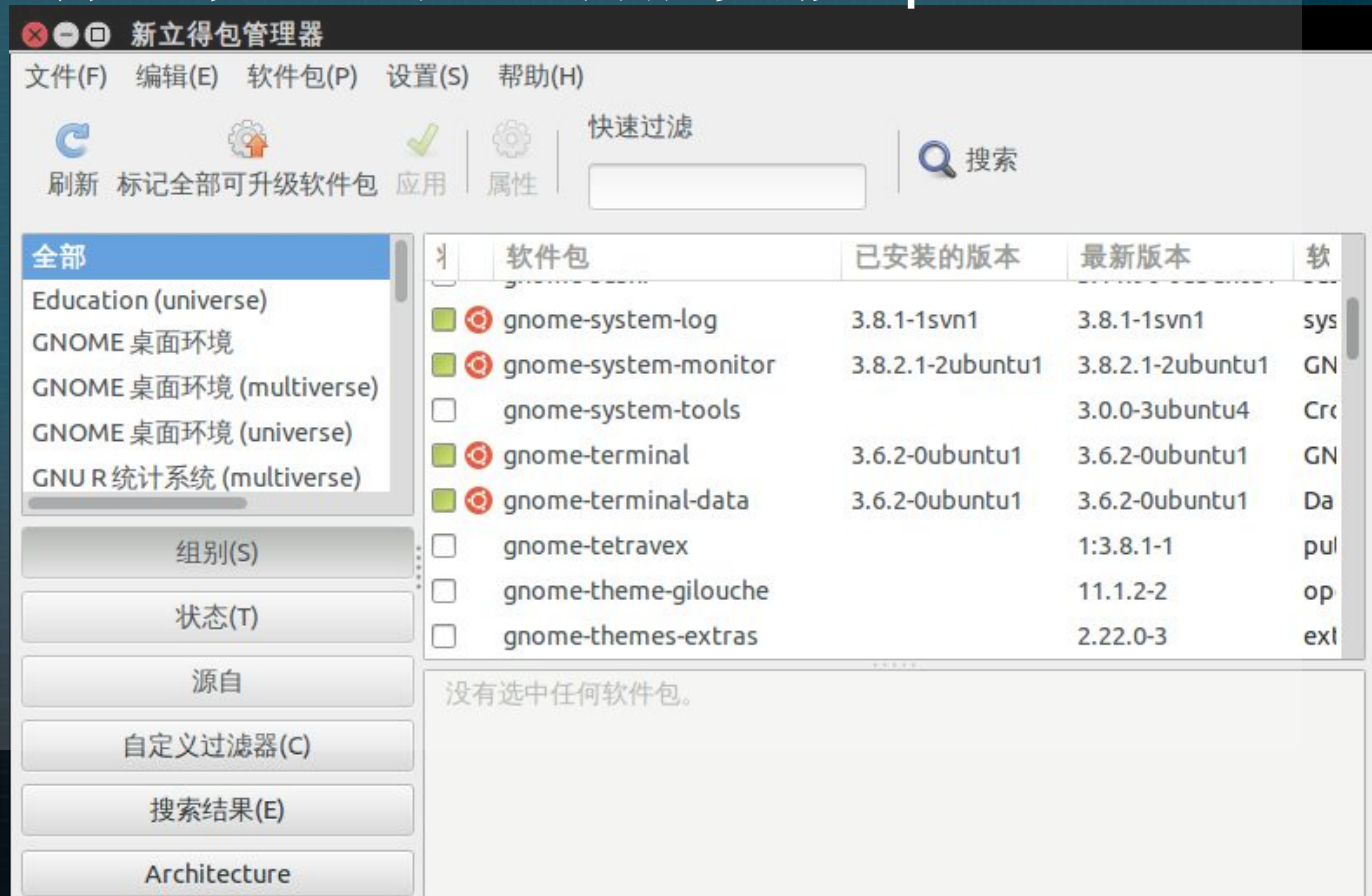
系统设置-软件与更新:

- 图形化管理
/etc/apt/sources.list
的工具



图形界面工具简介

- 新立得软件包管理器(synaptic):
- 较高级的软件包管理工具，功能类似apt和aptitude



图形界面工具简介

- gdebi/qapt:图形化安装deb包
- 替代ubuntu下的ubuntu软件中心



其它：手动安装软件包

- 手动安装软件都需要解决依赖。
- 注：详细请见
<https://archive.lug.ustc.edu.cn/2014-May/016454.html>
- 得到*.deb或*.rpm的软件包：

```
sudo dpkg -i XXX.deb
```

```
rpm -i XXX.rpm
```

注：deb系发行版可借助alien工具将rpm包
转为deb包安装

其它：手动安装软件包

- 得到类似xxx.run的文件：
是二进制可执行文件

```
chmod +x ./XXX.run    #添加可执行权限  
(sudo) ./XXX.run      #运行之
```

其它：从源代码编译安装

- 不建议新手编译安装软件
- 需要事先装上编译所需软件包，如 `gcc, g++, automake, build-essential` 等
- 得到的源代码包文件名常为 `XXX.tar.gz` `XXX.tar.xz` `XXX.tar.bz2` 等等，实际上是一个压缩文件

其它： 从源代码编译安装

- 解压缩： `tar -xvaf <文件名>`
- 在编译之前阅读解压所得目录下的README和INSTALL文档确认安装方式与所需依赖
- 例：对于使用autoconf的源码包的安装
- `./configure`
- `make`
- `sudo make install`



Q & A