

Linux 系统概论

天津医科大学
生物医学工程与技术学院

2018-2019 学年下学期（春）
2017 级生信班

第六章 软件安装

伊现富 (Yi Xianfu)

天津医科大学 (TJMU)
生物医学工程与技术学院

2019 年 6 月



教学提纲

- 1 引言
- 2 二进制软件包管理
 - dpkg 与 APT 包管理
 - RPM 与 Yum 包管理
 - 二进制软件包管理比较
- 3 源代码安装
 - 源代码
 - 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装
- 4 脚本安装
- 5 使用 bioconda 管理生信软件
- 6 回顾与总结
 - 总结
 - 思考题

教学提纲

1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

● 选择软件

● 下载软件

● 编译和安装

脚本安装

● 使用 bioconda 管理生信软件

回顾与总结

● 总结

● 思考题



引言 | 软件

*.exe
*.msi



*.bin
*.dmg



*.sh
*.deb
*.rpm



Windows 平台下软件安装的易用性非常不错。多数时候就是一个可执行的 exe 或 msi 文件，双击然后一路点击下一步即可；又或者解压后就可以直接运行。基本上，会用电脑的人都知道怎样在 Windows 上安装软件。

软件包包含程序运行的所需文件。但同样是“包”，又可以分两种，一种是源文件包，一种是已经编译过的二进制包。二进制文件计算机可以直接执行，比如 Windows 下的 exe 文件，或者 Ubuntu 下的 deb 包；而源文件则不行，需要编译才可以。那为什么 Linux 下软件包不全部以二进制可直接执行的形式提供呢？这是因为 Linux 系统种类繁多，所使用的二进制有所不同，需要分别打包才行，比如，x86 架构跟 AMD64 的需要分别提供，32 位系统与 64 位的也需要分别提供。

也就是说，Windows 下提供的软件安装文件基本都是已编译过的，而 Linux 下就不一定。



引言 | 软件管理 | 金山卫士软件管理

金山卫士软件管理

软件宝库 游戏宝库 软件升级 软件卸载

搜索热词：QQ2013 Beta1

精品推荐

- 全部
- 最新
- 视频
- 游戏
- 浏览器
- 聊天
- 下载
- 音乐
- 系统
- 图像
- 输入法
- 杀毒
- 学习
- 文字
- 驱动
- 桌面

编辑推荐

- 最原汁原味的三国网游
- 轻松一点 让刷机成瘾
- 首款双核安全浏览器
- 上网更快速，更简单

上升最快 HOT

热门软件	装机必备	装机评测	游戏必备	手机必备	网购必备											
猎豹浏览器 安全、极速 下载	数据恢复 找回数据 下载	金山词霸 翻译准确 下载	PPTV 海量节目源 下载	酷我音乐 一点就播 下载	QQ2012 含推荐表情 下载	有道词典 全文翻译 下载	有道云笔记 1G容量 下载	JJ斗地主 即点即玩 下载	91手机助手 资源最丰富 下载	迅雷7 下载速度快 下载	2345好压 压缩利器 下载	暴风影音 5 在线看高清 下载	搜狗拼音 长词联想 下载	快播5 宅男最爱 下载	暴风看电影 上线最快 下载	金山快盘 免费15G 下载

引言 | 软件管理 | Deepin 深度商店

Deepin 深度商店界面，左侧为应用分类导航栏，右侧为主展示区。

左侧应用分类：

- 首页推荐
- 系统升级
- 卸载软件
- 安装管理
- ★ 首页推荐
- 网络应用
- 影音播放
- 游戏娱乐
- 图形图像
- 办公软件
- 行业软件
- 科学教育
- 编程开发
- 系统管理
- 实用工具

右侧主展示区：

maxthon 正式版

无缝浏览 极速体验

热门推荐 专题介绍 下载排行

下方展示了以下应用图标：

- BRIQUOLE
- CrossOver
- SimpleScreenRecorder
- A7Xpg
- Grub Customizer
- WinUSB
- FreeCAD
- Synapse
- MonoDevelop
- Kazam Screencaster
- gimp
- FatRat

底部有“加入我们”按钮和系统控制图标。



引言 | 软件管理 | Ubuntu Kylin 软件中心

The screenshot shows the Ubuntu Kylin Software Center interface. On the left is a vertical sidebar with icons for Home, Library, Upgrade, Uninstall, Win替换 (Win Replacement), and a download icon. The main area features a large banner for "WPS for Linux" with three preview images of its interface. Below the banner is a section titled "推荐软件" (Recommended Software) displaying nine packages in a grid:

图标	名称	大小	状态
WPS Office	WPS 办公软件	448.82 MB	已安装
优客助手	优客助手	9.45 MB	已安装
金山快盘 Linux 版	金山快盘 Linux 版	7.15 MB	已安装
搜狗输入法 Linux 版	搜狗输入法 Linux 版	41.40 MB	已安装
Gimp 图像处理器	音视频解码器	30 KB	已安装
Uget 下载管理器	P7zip 文档压缩	3.81 MB	已安装
Flash 插件	Flash 插件	137 KB	已安装

To the right of the software grid is a window for "WPS 表格" (WPS Spreadsheet) showing a table with data related to "Crude Oil". A chart below the table shows oil prices over time. At the bottom of the software center is a "评分排行" (Rating Ranking) list:

- 1 Gparted 分区编辑
- 2 GeoGebra
- 3 韦诺之战 (1.10)
- 4 音视频解码器
- 5 VLC 播放器
- 6 Stellarium
- 7 韦诺之战 (1.10)
- 8 新立得软件包管理
- 9 FileZilla
- 10 P7zip 文档压缩



引言 | 软件管理 | Ubuntu 软件中心

Ubuntu 软件中心

所有软件 已安装 历史 进度

搜索

更多

办公
附件
互联网
教育
开发工具
科学与工程
通用访问
图书与杂志
图形
系统
影音
游戏
主题与系统增强
字体

最新应用程序

RAR格式的文件 附件 免费	Merkaartor 地理 ★★★★★ (6) 免费	luckyBackup 附件 ★★★★★ (77) 免费	Overgod 游戏 ★★★★★ (6) 免费
Super Transball 2 游戏 ★★★★★ (1) 免费	Fraqtive 数学 ★★★★★ (15) 免费	照片编辑附加程序 系统 免费	Tickr Feed 阅读器 互联网 ★★★★★ (21) 免费
SuperCollider IDE 影音 免费			

向您推荐

由于要评选推荐软件，Ubuntu 软件中心会偶尔把当前已安装的软件列表发送给 Canonical。

最高评分

GParted 分区编辑器 主题与系统增强 ★★★★★ (537) 免费	Geany 集成开发环境 ★★★★★ (450) 免费	布诺之战 (1.10) 游戏 ★★★★★ (122) 免费	市诺之战地图编辑器 (1.10) 游戏 ★★★★★ (122) 免费
GeoGebra 数学	VLC 多媒体播放程序 影音	Stellarium 天文	新立得软件包管理器 主题与系统增强



引言 | 软件管理 | 新立得软件包管理器

新立得包管理器

文件(F) 编辑(E) 软件包(P) 设置(S) 帮助(H)

刷新 标记全部可升级软件包 应用 快速过滤 属性 搜索 synaptic

软件包	已安装的版本	最新版本	软件包描述
synaptic	0.81.1ubuntu1	0.81.1ubuntu1	图形界面的软件包管理器
muon		2.2.0-0ubuntu3..	package manager for KDE
packagekit		0.8.12-1ubuntu5	Provides a package management service
sessioninstaller	0.20+bzr141-0u1	0.20+bzr141-0u1	APT based installer using PackageKit
python-brian		1.3.1-1build1	simulator for spiking neural networks
apt-cacher		1.7.8	Caching proxy for Debian package archives
xserver-xorg-input-synap		1.7.4-0ubuntu1	Synaptics TouchPad driver for X.Org
xserver-xorg-input-synap		3:6	Transitional package for xserver-xorg-input-synap
xserver-xorg-input-synap		3:6	Transitional package for xserver-xorg-input-synap
xserver-xorg-input-synap		3:6	Transitional package for xserver-xorg-input-synap

组别(S) 状态(T) 自源 自定义过滤器(C) 搜索结果(E) Architecture

获取截屏 获取变更日志 Visit Homepage

Synaptic 是一个基于 GTK+ 和 APT 的图形界面软件包管理工具，使用户可以方便地安装、升级和删除软件包。

Besides these basic functions the following features are provided:

- * Search and filter the list of available packages
- * Perform smart system upgrades
- * Fix broken package dependencies

共列出 22 个软件包，已安装 3479 个，已破损 0 个。将安装/升级 0 个软件包，将删除 0 个软件包



软件包管理系统

Linux 软件包管理系统是在电脑中自动安装、配制、卸载和升级软件包的工具组合，在各种系统软件和应用软件的安装管理中均有广泛应用。使用软件包管理系统将大大简化在 Linux 发行版中安装软件的过程。

软件包

- 二进制包（预编译的软件包）
 - deb 软件包：Debian, Ubuntu, Linux Mint
 - rpm 软件包：RHEL, CentOS, Fedora, SUSE, openSUSE, Mandriva Linux, Mageia, PCLinuxOS
- 源代码安装包
- 脚本安装包



软件包管理系统

Linux 软件包管理系统是在电脑中自动安装、配制、卸载和升级软件包的工具组合，在各种系统软件和应用软件的安装管理中均有广泛应用。使用软件包管理系统将大大简化在 Linux 发行版中安装软件的过程。

软件包

- 二进制包（预编译的软件包）
 - deb 软件包：Debian, Ubuntu, Linux Mint
 - rpm 软件包：RHEL, CentOS, Fedora, SUSE, openSUSE, Mandriva Linux, Mageia, PCLinuxOS
- 源代码安装包
- 脚本安装包



二进制包管理系统

`dpkg` 及其前端 `APT` Ubuntu, Debian, Linux Mint, Deepin

`RPM` 及其前端 `Yum` Red Hat Enterprise Linux, CentOS, Fedora

`ZYpp` 及其前端 `Zypper` SUSE, openSUSE

`urpmi` Mandriva Linux, Mageia Linux, ROSA Linux

`pacman` Manjora, Arch Linux

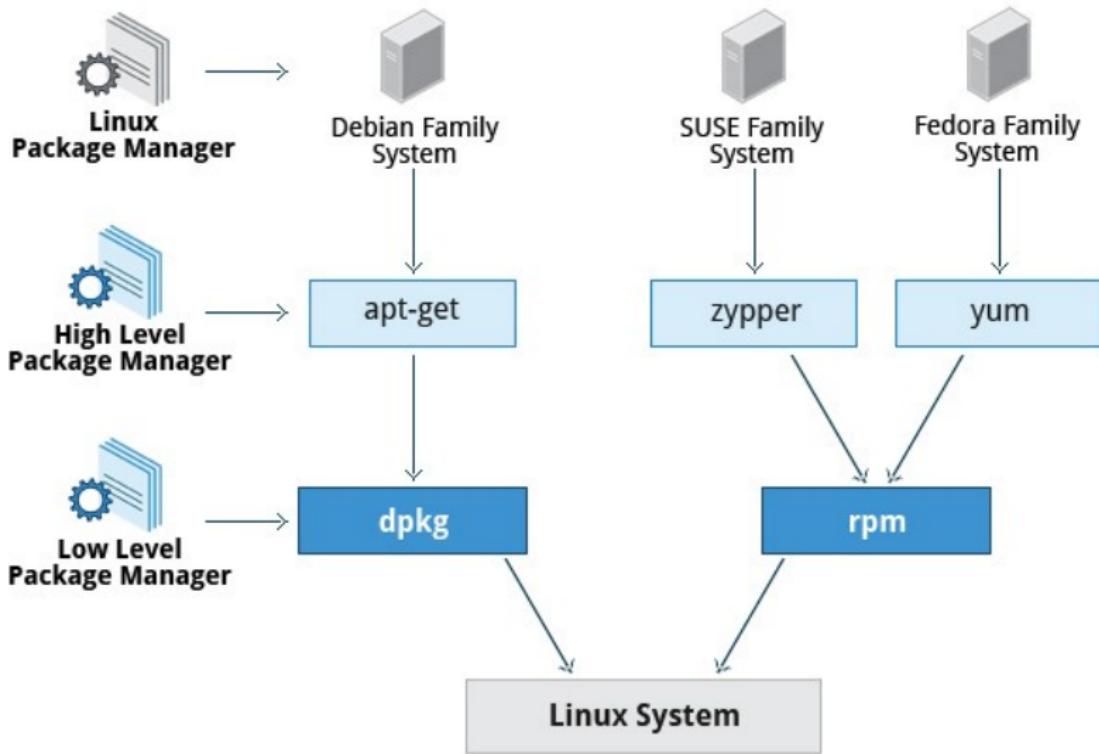
`Portage` Gentoo

`slapt-get` Slackware

... ...



引言 | 软件包管理系统



1

引言

2

二进制软件包管理

3

源代码安装

- 源代码

- 开放源代码许可证

- 选择软件

- 下载软件

- 编译和安装

脚本安装

使用 bioconda 管理生信软件

回顾与总结

- 总结

- 思考题



预编译软件包

预编译软件包是某个项目或程序立即可用的软件。

包中通常含有可执行文件、配置示例或基本的配置、文档和辅助数据文件。

包中还包含了元数据，如包的简要描述、谁或在何处编译了它、它提供了什么、它依赖于哪些包。



软件包管理工具

管理工具用于安装软件、自动安装必须预备的包、显示已安装的包、列出可用的包、显示文件列表、删除包、检索包、验证已安装的文件、提供包安全提示器等。

- 低级工具：能在后端实际安装、升级、卸载软件包文件
- 高级工具：负责确保能很好的执行依赖性解决和元数据检索的任务

发行版	低级工具	高级工具
Debian 版	dpkg	apt/apt-get/Aptitude
CentOS 版	RPM	Yum
openSUSE 版	RPM	Zypper



软件包管理工具

管理工具用于安装软件、自动安装必须预备的包、显示已安装的包、列出可用的包、显示文件列表、删除包、检索包、验证已安装的文件、提供包安全提示器等。

- 低级工具：能在后端实际安装、升级、卸载软件包文件
- 高级工具：负责确保能很好的执行依赖性解决和元数据检索的任务

发行版	低级工具	高级工具
Debian 版	dpkg	apt/apt-get/Aptitude
CentOS 版	RPM	Yum
openSUSE 版	RPM	Zypper



二进制软件包管理 | 管理工具 | 低级 vs. 高级

dpkg

基于 Debian 系统的一个低级包管理器。它可以安装，删除，提供有关资料，以及建立 *.deb 包，但它不能自动下载并安装它们相应的依赖包。

apt/apt-get

Debian 及其衍生版的高级包管理器，并提供命令行方式来从多个来源检索和安装软件包，其中包括解决依赖性。和 dpkg 不同的是，apt-get 不是直接基于.deb 文件工作，而是基于软件包的正确名称。

Aptitude

基于 Debian 系统的另一个高级包管理器，它可用于快速简便得执行管理任务（安装，升级和删除软件包，还可以自动处理解决依赖性）。它在 apt-get 的基础上提供了更多功能，例如提供对软件包的几个版本的访问。

二进制软件包管理 | 管理工具 | 低级 vs. 高级

dpkg

基于 Debian 系统的一个低级包管理器。它可以安装，删除，提供有关资料，以及建立 *.deb 包，但它不能自动下载并安装它们相应的依赖包。

apt/apt-get

Debian 及其衍生版的高级包管理器，并提供命令行方式来从多个来源检索和安装软件包，其中包括解决依赖性。和 dpkg 不同的是，apt-get 不是直接基于.deb 文件工作，而是基于软件包的正确名称。

Aptitude

基于 Debian 系统的另一个高级包管理器，它可用于快速简便得执行管理任务（安装，升级和删除软件包，还可以自动处理解决依赖性）。它在 apt-get 的基础上提供了更多功能，例如提供对软件包的几个版本的访问。

二进制软件包管理 | 管理工具 | 低级 vs. 高级

dpkg

基于 Debian 系统的一个低级包管理器。它可以安装，删除，提供有关资料，以及建立 *.deb 包，但它不能自动下载并安装它们相应的依赖包。

apt/apt-get

Debian 及其衍生版的高级包管理器，并提供命令行方式来从多个来源检索和安装软件包，其中包括解决依赖性。和 dpkg 不同的是，apt-get 不是直接基于.deb 文件工作，而是基于软件包的正确名称。

Aptitude

基于 Debian 系统的另一个高级包管理器，它可用于快速简便得执行管理任务（安装，升级和删除软件包，还可以自动处理解决依赖性）。它在 apt-get 的基础上提供了更多功能，例如提供对软件包的几个版本的访问。

RPM

Linux 标准基础 (LSB) 兼容发行版所使用的一种软件包管理器，用来对软件包进行低级处理。就像 dpkg 一样， RPM 可以查询、安装、检验、升级和卸载软件包，它多数用于基于 Fedora 的系统，比如 RHEL 和 CentOS。

Yum

相对于基于 RPM 的系统， Yum 增加了系统自动更新的功能和带依赖性管理的软件包管理功能。作为一个高级工具，和 apt-get 或者 Aptitude 相似， Yum 需要配合仓库工作。



RPM

Linux 标准基础 (LSB) 兼容发行版所使用的一种软件包管理器，用来对软件包进行低级处理。就像 dpkg 一样，RPM 可以查询、安装、检验、升级和卸载软件包，它多数用于基于 Fedora 的系统，比如 RHEL 和 CentOS。

Yum

相对于基于 RPM 的系统，Yum 增加了系统自动更新的功能和带依赖性管理的软件包管理功能。作为一个高级工具，和 apt-get 或者 Aptitude 相似，Yum 需要配合仓库工作。



教学提纲

- 1 引言
- 2 二进制软件包管理
 - dpkg 与 APT 包管理
 - RPM 与 Yum 包管理
 - 二进制软件包管理比较
- 3 源代码安装
 - 源代码
 - 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装
- 4 脚本安装
- 5 使用 bioconda 管理生信软件
- 6 回顾与总结
 - 总结
 - 思考题



dpkg

dpkg (Debian Package) 是 Debian 软件包管理器的基础，它被伊恩·默多克创建于 1993 年。dpkg 与 RPM 十分相似，同样被用于安装、卸载和提供与.deb 软件包相关的信息。

dpkg 本身是一个底层的工具。上层的工具，像是 APT，被用于从远程获取软件包以及处理复杂的软件包关系。

APT

APT (Advanced Packaging Tools, 高级包装工具) 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格式的软件包，因此简化了 Linux 系统上管理软件的过程。

APT 最早被设计成 dpkg 的前端，用来处理 deb 格式的软件包。现在经过 APT-RPM 组织修改，APT 已经可以安装在支持 RPM 包管理的系统上了。

dpkg

dpkg (Debian Package) 是 Debian 软件包管理器的基础，它被伊恩·默多克创建于 1993 年。dpkg 与 RPM 十分相似，同样被用于安装、卸载和提供与.deb 软件包相关的信息。

dpkg 本身是一个底层的工具。上层的工具，像是 APT，被用于从远程获取软件包以及处理复杂的软件包关系。

APT

APT (Advanced Packaging Tools, 高级包装工具) 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格式的软件包，因此简化了 Linux 系统上管理软件的过程。

APT 最早被设计成 dpkg 的前端，用来处理 deb 格式的软件包。现在经过 APT-RPM 组织修改，APT 已经可以安装在支持 RPM 包管理的系统上了。

二进制软件包管理 | dpkg 与 APT | dpkg

命令	助记	作用
dpkg -c debfile	Contents	列出软件包的内容
dpkg -i debfile	Install	安装.deb 软件包
dpkg -l debfile	Info	从.deb 文件中提取软件包信息
dpkg -l pattern	List	列出所有与模式相匹配的软件包
dpkg -L package	Listfiles	列出已安装软件包的文件清单
dpkg -r package	Remove	卸载已安装的软件包
dpkg -P package	Purge	完全卸载包（卸载并删除配置文件）
dpkg -s package	Status	显示已安装包的信息
dpkg -S filename	Search	查询系统中某个文件属于那个软件包



二进制软件包管理 | dpkg 与 APT | dpkg

命令	作用
dpkg-reconfigure package	重新配置已安装的软件包
dpkg --get-selections	获取软件包状态
dpkg --set-selections	设置软件包状态
dpkg --force-all --purge package	强制卸载（有风险！）



二进制软件包管理 | dpkg 与 APT | dpkg

Syntax	Description	Example
dpkg -i { .deb package }	Install the package	dpkg -i zip_2.31-3_i386.deb
dpkg -i { .deb package }	Upgrade package if it is installed else install a fresh copy of package	dpkg -i zip_2.31-3_i386.deb
dpkg -R {Directory-name}	Install all packages recursively from directory	dpkg -R /tmp/downloads
dpkg -r {package}	Remove/Delete an installed package except configuration files	dpkg -r zip
dpkg -P {package}	Remove/Delete everything including configuration files	dpkg -P apache-perl
dpkg -l	List all installed packages, along with package version and short description	dpkg -l dokg -l less dpkg -l "apache" dpkg -l grep -i 'sudo'
dpkg -l {package}	List individual installed packages, along with package version and short description	dpkg -l apache-perl
dpkg -L {package}	Find out files are provided by the installed package i.e. list where files were installed	dpkg -L apache-perl dpkg -L perl
dpkg -c { .Deb package }	List files provided (or owned) by the package i.e. List all files inside debian .deb package file, very useful to find where files would be installed	dpkg -c dc_1.06-19_i386.deb
dpkg -S {/path/to/file}	Find what package owns the file i.e. find out what package does file belong	dpkg -S /bin/netstat dpkg -S /sbin/ippool
dpkg -p {package}	Display details about package package group, version, maintainer, Architecture, display depends packages, description etc	dpkg -p lsof
dpkg -s {package} grep Status	Find out if Debian package is installed or not (status)	dpkg -s lsof grep Status



二进制软件包管理 | dpkg 与 APT | APT | 工具

apt/apt-get 负责软件包的在线安装与升级，底层对 deb 包的处理还是用的 dpkg，解决依赖关系

apt-cache 用来查询软件包的状态和依赖关系

apt-file 负责查询软件包名称和软件包包含的文件（值得注意的是它要自己同步）

apt-cross 负责为交叉编译的软件包的安装与编译等

apt-offline 可以离线安装软件包

apt-build 可以简化源代码编译

... ...



二进制软件包管理 | dpkg 与 APT | APT

命令	作用
apt-cache stats	显示软件包的统计信息
apt-cache search pattern	查找与模式相匹配的软件包
apt-cache show package	显示软件包的详细信息
apt-cache depends package	查找软件包的依赖关系
apt-cache showsrc	查看源码包的文件信息
apt-cache showpkg package	显示软件包的详细信息（包括和其他软件包的关系）



二进制软件包管理 | dpkg 与 APT | APT

命令	作用
apt-get install package	安装软件包
apt-get --reinstall install package	重新安装软件包
apt-get remove package	卸载软件包（保留配置文件）
apt-get --purge remove package	完全卸载软件包（删除配置文件）
apt-get -f install	修正依赖关系损坏的软件包
apt-get check	检查是否有损坏的依赖
apt-get clean	清除软件包缓存
apt-get autoclean	清除旧版本的软件包缓存
apt-get autoremove	删除不再使用的孤立软件



二进制软件包管理 | dpkg 与 APT | APT (续)

命令	作用
apt-get update	更新软件包列表
apt-get upgrade	更新已安装的软件包
apt-get dist-upgrade	升级系统到最新版本
apt-get source	下载源码包
apt-get build-dep package	构建源码包的编译环境



二进制软件包管理 | dpkg 与 APT | APT

Syntax	Description	Example(s)
apt-get install {package}	Install the new package. If package is installed then try to upgrade to latest version	apt-get install zip apt-get install lsof samba mysql-client
apt-get remove {package}	Remove/Delete an installed package except configuration files	apt-get remove zip
apt-get --purge remove {package}	Remove/Delete everything including configuration files	apt-get --purge remove mysql-server
apt-get update apt-get upgrade	Resynchronize the package index files and Upgrade the Debian Linux system including security update (Internet access required)	apt-get update apt-get upgrade
apt-get update apt-get dist-upgrade	Usually use to upgrade to Debian distribution. For example Woody to Sarge upgrade. 'dist-upgrade' in addition to performing the function of upgrade, also intelligently handles changing dependencies with new versions of packages; apt-get has a "smart" conflict resolution system, and it will attempt to upgrade the most important packages at the expense of less important ones if necessary.	apt-get update apt-get dist-upgrade

Aptitude

一个基于 Ncurses 的 APT 文字接口前端程序，是 Debian 及其派生系统中功能极其强大的软件包管理系统。

以交互形式运行时，它可以显示出所有可用的软件包，并允许用户选择包来安装或卸载。它还有一个强大的搜索系统，可通过多种搜索模式进行搜索。

最初，它是为 Debian 所开发，但现在已有可用于 RPM 包管理系统的派生。



二进制软件包管理 | dpkg 与 APT | 比较

- dpkg 是底层工具
- APT 是 dpkg 的前端
- Aptitude 是 APT 的前端（字符终端）
- Synaptic（新立得）是 APT 的前端（图形界面）



二进制软件包管理 | dpkg 与 APT | 比较

dpkg

主要是对本地的软件包（本地安装的软件包和已经下载但还没有安装的 deb 文件）进行管理，不解决依赖关系。

APT

高级的软件包管理工具，在安装软件时，会自动解决软件安装过程中的依赖关系，但不会自动删除不需要的软件包。

Aptitude

带有 UI 界面的更高级的软件包管理工具，自动解决软件包安装中的依赖关系；并且在删除的时候，会自动删除不需要的软件依赖关系安装包；更加的智能、高效。有两种基本的使用方法：文本界面和命令行。



二进制软件包管理 | dpkg 与 APT | 比较

dpkg

主要是对本地的软件包（本地安装的软件包和已经下载但还没有安装的 deb 文件）进行管理，不解决依赖关系。

APT

高级的软件包管理工具，在安装软件时，会自动解决软件安装过程中的依赖关系，但不会自动删除不需要的软件包。

Aptitude

带有 UI 界面的更高级的软件包管理工具，自动解决软件包安装中的依赖关系；并且在删除的时候，会自动删除不需要的软件依赖关系安装包；更加的智能、高效。有两种基本的使用方法：文本界面和命令行。



二进制软件包管理 | dpkg 与 APT | 比较

dpkg

主要是对本地的软件包（本地安装的软件包和已经下载但还没有安装的 deb 文件）进行管理，不解决依赖关系。

APT

高级的软件包管理工具，在安装软件时，会自动解决软件安装过程中的依赖关系，但不会自动删除不需要的软件包。

Aptitude

带有 UI 界面的更高级的软件包管理工具，自动解决软件包安装中的依赖关系；并且在删除的时候，会自动删除不需要的软件依赖关系安装包；更加的智能、高效。有两种基本的使用方法：文本界面和命令行。



PPA (Personal Package Archive, 个人软件包档案)

因为安全或不稳定或其他种种因素，Ubuntu 软件源中不可能收录所有软件，于是有了 PPA，它的作用类似于 Ubuntu 官方的源，只不过是由个人提供打包，当然也会有某些官方源中的软件通过 PPA 发布不稳定版本等。

Ubuntu Launchpad 网站提供的一项服务，当然不仅限于 Launchpad。它允许个人用户上传软件源代码，通过 Launchpad 进行编译并发布为二进制软件包，作为 APT/新立得源供其他用户下载和更新。在 Launchpad 网站上的每一个用户和团队都可以拥有一个或多个 PPA。

通常 PPA 源里的软件是官方源里没有的，或者是最新版本的软件。相对于通过 Deb 包安装来说，使用 PPA 的好处是，一旦软件有更新，通过 `sudo apt-get upgrade` 这样命令就可以直接升级到新版本。



三步走

- ① 添加 PPA 源 : sudo add-apt-repository ppa:USER/PPA-NAME
- ② 更新所有源 : sudo apt-get update
- ③ 安装软件 : sudo apt-get install PACKAGE_NAME

实例（安装 Java）

- ① sudo add-apt-repository ppa:webupd8team/java
- ② sudo apt-get update
- ③ sudo apt-get install oracle-java8-installer



三步走

- ① 添加 PPA 源 : sudo add-apt-repository ppa:USER/PPA-NAME
- ② 更新所有源 : sudo apt-get update
- ③ 安装软件 : sudo apt-get install PACKAGE_NAME

实例（安装 Java）

- ① sudo add-apt-repository ppa:webupd8team/java
- ② sudo apt-get update
- ③ sudo apt-get install oracle-java8-installer



教学提纲

1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装

4

脚本安装

5

使用 bioconda 管理生信软件

6

回顾与总结

- 总结
- 思考题



二进制软件包管理 | RPM 与 Yum | 简介

RPM

RPM 包管理器（简称 RPM，全称为 The RPM Package Manager）是在 Linux 下广泛使用的软件包管理器。最早由 Red Hat 研制，现在也由开源社区开发。RPM 仅适用于安装用 RPM 来打包的软件，目前是 GNU/Linux 下软件包资源最丰富的软件包类型之一。

RPM 软件包分为二进制包（Binary）、源代码包（Source）和 Delta 包三种。二进制包可以直接安装在计算机中，而源代码包将会由 RPM 自动编译、安装。源代码包经常以 `src.rpm` 作为后缀名。

Yum

Yum (Yellow dog Updater, Modified) 由 Duke University 团队，修改 Yellow Dog Linux 的 Yellow Dog Updater 开发而成，是一个基于 RPM 包管理的字符前端软件包管理器。能够从指定的服务器自动下载 RPM 包并且安装，可以处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。

二进制软件包管理 | RPM 与 Yum | 简介

RPM

RPM 包管理器（简称 RPM，全称为 The RPM Package Manager）是在 Linux 下广泛使用的软件包管理器。最早由 Red Hat 研制，现在也由开源社区开发。RPM 仅适用于安装用 RPM 来打包的软件，目前是 GNU/Linux 下软件包资源最丰富的软件包类型之一。

RPM 软件包分为二进制包（Binary）、源代码包（Source）和 Delta 包三种。二进制包可以直接安装在计算机中，而源代码包将会由 RPM 自动编译、安装。源代码包经常以 `src.rpm` 作为后缀名。

Yum

Yum (Yellow dog Updater, Modified) 由 Duke University 团队，修改 Yellow Dog Linux 的 Yellow Dog Updater 开发而成，是一个基于 RPM 包管理的字符前端软件包管理器。能够从指定的服务器自动下载 RPM 包并且安装，可以处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。

二进制软件包管理 | RPM 与 Yum | RPM | 功能

选项	助记	说明
-q	Query	查询软件包
-V	Verify	校验软件包
-i	Install	安装软件包
-e	Erase	删除软件包
-U	Upgrade	升级软件包



二进制软件包管理 | RPM 与 Yum | RPM | 选项

选项	助记	说明	类别
-v	Verbose	详细信息	通用选项
-a	All	所有已安装的软件包	选择选项
-f	File	文件所属软件包	
-p	Package	指定软件包 (.rpm)	



二进制软件包管理 | RPM 与 Yum | RPM | 选项

选项	助记	说明	类别
-l	List	软件包中的文件列表	查询选项
-i	Info	软件包的信息	
-c	Configfiles	配置文件	
-d	Docfiles	文档文件	
-R	Requires	软件包依赖关系	
-s	State	软件包状态	
-h	Hash	进度条	安装选项
--nodeps	—	不检查依赖性	
--excludedocs	—	不安装文档	
--prefix	—	指定安装路径	
--test	—	测试而不安装	
--replacepkgs	—	覆盖安装	
--replacefiles	—	替换文件	
--force	—	强制安装	

二进制软件包管理 | RPM 与 Yum | RPM | 命令

命令	作用
rpm -qa	查询系统中已经安装的软件包
rpm -qf file	查询已安装的文件属于那个软件包
rpm -ql package	查询已安装软件包把文件安装到何处
rpm -qi package	查询已安装软件包的信息
rpm -qc package	查询已安装软件包的配置文件
rpm -qd package	查询已安装软件包的文档文件
rpm -qR package	查询已安装软件包的依赖关系
rpm -qpi rpmfile	查询软件包的信息
rpm -qpl rpmfile	查询软件包的文件列表
rpm -qpd rpmfile	查询软件包的文档文件
rpm -qpc rpmfile	查询软件包的配置文件
rpm -qpR rpmfile	查询软件包的依赖关系
rpm -ivh rpmfile	安装软件包
rpm -Uvh rpmfile	升级软件包
rpm -e package	删除软件包

二进制软件包管理 | RPM 与 Yum | RPM

Syntax	Description	Example(s)
rpm -ivh {rpm-file}	Install the package	rpm -ivh mozilla-mail-1.7.5-17.i586.rpm rpm -ivh --test mozilla-mail-1.7.5-17.i586.rpm
rpm -Uvh {rpm-file}	Upgrade package	rpm -Uvh mozilla-mail-1.7.6-12.i586.rpm rpm -Uvh --test mozilla-mail-1.7.6-12.i586.rpm
rpm -ev {package}	Erase/remove/ an installed package	rpm -ev mozilla-mail
rpm -ev --nodeps {package}	Erase/remove/ an installed package without checking for dependencies	rpm -ev --nodeps mozilla-mail
rpm -qa	Display list all installed packages	rpm -qa rpm -qa less
rpm -qi {package}	Display installed information along with package version and short description	rpm -qi mozilla-mail
rpm -qf {/path/to/file}	Find out what package a file belongs to i.e. find what package owns the file	rpm -qf /etc/passwd rpm -qf /bin/bash
rpm -qc {pacakge-name}	Display list of configuration file(s) for a package	rpm -qc httpd
rpm -qcf {/path/to/file}	Display list of configuration files for a command	rpm -qcf /usr/X11R6/bin/xeyes
rpm -qa --last	Display list of all recently installed RPMs	rpm -qa --last rpm -qa --last less
rpm -qpR {.rpm-file} rpm -qR {package}	Find out what dependencies a rpm file has	rpm -qpR mediawiki-1.4rc1-4.i586.rpm rpm -qR bash



二进制软件包管理 | RPM 与 Yum | Yum

命令	作用
yum install package	安装软件包
yum remove package	删除软件包
yum check-update	检查可以更新的软件包
yum update	更新所有软件包
yum update package	更新指定软件包
yum upgrade	升级系统
yum clean package	清除缓存中的 rpm 软件包
yum list	列出所有可以安装或更新的软件包
yum list package	列出指定的软件包
yum list updates	列出所有可以更新的软件包
yum list installed	列出所有已经安装的软件包
yum info	列出所有可以安装或更新的软件包的信息
yum search pattern	搜索匹配模式的软件包
yum provides file	搜索包含指定文件的软件包

二进制软件包管理 | RPM 与 Yum | Yum

command	action
yum install newpackage	Install newpackage
yum update [package]	Updates (upgrades) either all packages with updates available (if no argument given), or the specified package.
yum remove newpackage	Removes oldpackage.
yum info mypackage	Prints out information on the specified package (or use available or installed as the argument to get information on all available or installed packages).
yum search keyword	Searches name, summary, description, and URL fields for keyword
yum list [installed available]	Lists all installed or available packages from your repositories
yum list updates	Lists updates available for installed packages (use yum check-updates for shell scripts, as this returns an exit code of 100 if there are updates available)



- 1 引言
- 2 二进制软件包管理
 - dpkg 与 APT 包管理
 - RPM 与 Yum 包管理
 - 二进制软件包管理比较
- 3 源代码安装
 - 源代码
 - 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装
- 4 脚本安装
- 5 使用 bioconda 管理生信软件
- 6 回顾与总结
 - 总结
 - 思考题



二进制软件包管理 | 比较

Operation	RPM	Deb
Install a package	rpm -i foo.rpm	dpkg --install foo.deb
Install a package with dependencies from repository	yum install foo	apt-get install foo
Remove a package	rpm -e foo.rpm	dpkg --remove foo.deb
Remove a package and dependencies using repository	yum remove foo	apt-get remove foo
Update package to a newer version	rpm -U foo.rpm	dpkg --install foo.deb
Update package using repository and resolving dependencies	yum update foo	apt-get upgrade foo
Update entire system	yum update	apt-get dist-upgrade
Show all installed packages	rpm -qa yum list installed	dpkg --list
Get information about an installed package including files	rpm -qil foo	Dpkg --listfiles foo
Show available package with "foo" in name	yum list foo	apt-cache search foo
Show all available packages	yum list	apt-cache dumpavail
What packages does a file belong to?	rpm -qf file	dpkg --search file

二进制软件包管理 | 比较

Task	Red Hat/Fedora	Ubuntu
Adding, Removing and Upgrading Packages		
Refresh list of available packages	Yum refreshes each time it's used	apt-get update
Install a package from a repository	yum install package_name	apt-get install package_name
Install a package file	yum install package.rpm rpm -i package.rpm	dpkg --install package.deb
Remove a package	rpm -e package_name	apt-get remove package_name
Check for package upgrades	yum check-update	apt-get -s upgrade apt-get -s dist-upgrade
Upgrade packages	yum update rpm -Uvh [args]	apt-get upgrade
Upgrade the entire system	yum upgrade	apt-get dist-upgrade
Package Information		
Get information about an available package	yum search package_name	apt-cache search package_name
Show available packages	yum list available	apt-cache dumpavail
List all installed packages	yum list installed rpm -qa	dpkg --list
Get information about a package	yum info package_name	apt-cache show package_name

二进制软件包管理 | 比较

Task	apt (deb) Debian, Ubuntu	zypp (rpm) openSUSE	yum (rpm) Fedora, CentOS	urpmi (rpm) Mandriva, Mageia
Managing software				
Install new software from package repository	apt-get install <i>pkg</i>	zypper install <i>pkg</i>	yum install <i>pkg</i>	urpmi <i>pkg</i>
Install new software from package file	dpkg -i <i>pkg</i>	zypper install <i>pkg</i>	yum localinstall <i>pkg</i>	urpmi <i>pkg</i>
Update existing software	apt-get install <i>pkg</i>	zypper update -t package <i>pkg</i>	yum update <i>pkg</i>	urpmi <i>pkg</i>
Remove unwanted software	apt-get remove <i>pkg</i>	zypper remove <i>pkg</i>	yum erase <i>pkg</i>	urpmi <i>pkg</i>
Updating the system				
Update package list	apt-get update aptitude update	zypper refresh	yum check-update	urpmi.update -a
Update system	apt-get upgrade aptitude safe-upgrade	zypper update	yum update	urpmi --auto-select
Searching for packages				
Search by package name	apt-cache search <i>pkg</i>	zypper search <i>pkg</i>	yum list <i>pkg</i>	urpmq <i>pkg</i>
Search by pattern	apt-cache search <i>pattern</i>	zypper search -t pattern <i>pattern</i>	yum search <i>pattern</i>	urpmq --fuzzy <i>pkg</i>
Search by file name	apt-file search <i>path</i>	zypper wp <i>file</i>	yum provides <i>file</i>	urpmf <i>file</i>
List installed packages	dpkg -l	zypper search -s	rpm -qa	rpm -qa
Configuring access to software repositories				
List repositories	cat /etc/apt/sources.list	zypper repos	yum repolist	urpmq --list-media
Add repository	(edit /etc/apt/sources.list)	zypper addrepo <i>path name</i>	(add <i>repo</i> to /etc/yum.repos.d/)	urpmi.addmedia <i>name path</i>
Remove repository	(edit /etc/apt/sources.list)	zypper removerepo <i>name</i>	(remove <i>repo</i> from /etc/yum.repos.d/)	urpmi.removemedia <i>media</i>

二进制软件包管理 | 比较

Task	apt (deb) Ubuntu, Debian	yum (rpm) Fedora, CentOS	zypp (rpm) openSUSE	urpmi (rpm) Mandriva	slapt-get (tgz,tbz,tlz) Vector	pacman (pkg.tar.gz) Arch	smart http://smartipm.org
Managing software							
Install new software from package repository	apt-get install <i>pkg</i>	yum install <i>pkg</i>	zypper install <i>pkg</i>	urpmi <i>pkg</i>	slapt-get --install <i>pkg</i>	pacman -S <i>pkg</i>	smart install <i>pkg</i>
Install new software from package file	dpkg -i <i>pkg</i>	yum localinstall <i>pkg</i>	zypper install <i>pkg</i>	urpmi <i>pkg</i>	slapt-get --install <i>pkg</i>	pacman -U <i>pkg</i>	smart install <i>pkg</i>
Update existing software	apt-get install <i>pkg</i>	yum update <i>pkg</i>	zypper update -i package <i>pkg</i>	urpmi <i>pkg</i>	slapt-get --install <i>pkg</i>	pacman -S <i>pkg</i>	smart upgrade <i>pkg</i>
Remove unwanted software	apt-get remove <i>pkg</i>	yum erase <i>pkg</i>	zypper remove <i>pkg</i>	urpmi <i>pkg</i>	slapt-get --remove <i>pkg</i>	pacman -R <i>pkg</i>	smart remove <i>pkg</i>
Updating the system							
Update package list	apt-get update	yum check-update	zypper refresh	urpmi.update -a	slapt-get --update	pacman -Sy	smart update
Update system	apt-get upgrade	yum update	zypper update	urpmi --auto-select	slapt-get --upgrade	pacman -Su	smart upgrade
Check all dependencies	apt-get check						smart check
Fix dependency problems	apt-get -f install						smart fix
Searching for packages							
Search by package name	apt-cache search <i>pkg</i>	yum list <i>pkg</i>	zypper search <i>pkg</i>	urpmq <i>pkg</i>	slapt-get --search <i>pkg</i>	pacman -Ss <i>pkg</i>	smart query <i>pkg</i>
Search by pattern	apt-cache search <i>pat</i>	yum search <i>pat</i>	zypper search -t pattern <i>pat</i>	urpmq --fuzzy <i>pat</i>	slapt-get --search <i>pat</i>	pacman -Ss <i>pat</i>	smart search <i>pat</i>
Search by file name	apt-file search <i>file</i>	yum provides <i>file</i>	zypper wp <i>file</i>	urpmf <i>file</i>	--	pacman -Qo <i>file</i>	smart query --path= <i>file</i>
List installed packages	dpkg -l	yum list installed	zypper search -is	rpm -qa	slapt-get --installed	pacman -Q	smart query --installed
Configuring access to software repositories							
List repositories	cat /etc/apt/sources.list	yum repolist	zypper repos	urpmq --list-media	cat < /slapt-get/slapt-getc	cat /etc/pacman.conf	smart channel --show
Add repository	(edit /etc/apt/sources.list)	(add repo to /etc/yum.repos.d/)	zypper addrepo <i>path name</i>	urpmi.addmedia <i>name path</i>	(edit /etc/slapt-get/slapt-getrc)	(edit /etc/pacman.conf)	smart channel --add <i>name path</i>
Remove repository	(edit /etc/apt/sources.list)	(remove repo from /etc/yum.repos.d/)	zypper removerepo <i>name</i>	urpmi.removemedia <i>name</i>	(edit /etc/slapt-get/slapt-getrc)	(edit /etc/pacman.conf)	smart channel --remove <i>name</i>



教学提纲

1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装

4

脚本安装

5

使用 bioconda 管理生信软件

6

回顾与总结

- 总结
- 思考题



教学提纲

1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装

4

脚本安装

5

使用 bioconda 管理生信软件

6

回顾与总结

- 总结
- 思考题



源代码

源代码是创建软件的原始数据。

下载的源代码包通常提供由 C、C++、Perl、Python 或其他编程语言编写的源代码和文档，其中包括编译说明、帮助编译和安装软件的脚本和实用工具、数据文件、配置示例和各种其他文件。



1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

- 选择软件

- 下载软件

- 编译和安装

脚本安装

- 使用 bioconda 管理生信软件

回顾与总结

- 总结

- 思考题



开放源代码

以源代码形式提供的软件通常称为开放源代码（open source）。开放源代码软件是带有特定许可条款的软件，通常可以自由地查看、共享和使用它，（最重要的是）还可以自由地修改它。

开放源代码许可证

开放源代码软件许可证声明对源代码拥有版权，并确定代码的使用和发布、以及派生代码的使用和发布的自由范围和限制。



开放源代码

以源代码形式提供的软件通常称为开放源代码（open source）。开放源代码软件是带有特定许可条款的软件，通常可以自由地查看、共享和使用它，（最重要的是）还可以自由地修改它。

开放源代码许可证

开放源代码软件许可证声明对源代码拥有版权，并确定代码的使用和发布、以及派生代码的使用和发布的自由范围和限制。





GNU Operating System



mozilla
FOUNDATION



open source
initiative
Approved License



BSD 许可证

BSD 许可证 (Berkeley Software Distribution license) , 是自由软件中使用最广泛的许可证之一。BSD (Berkeley Software Distribution, 伯克利软件套件, Unix 变种) 就是遵照这个许可证来发布的, 该许可证也因此而得名。

跟其他许可证相比, 从 GNU 通用公共许可证 (GPL) 到限制重重的著作权 (Copyright) , BSD 许可证比较宽松, 甚至跟公有领域更为接近。事实上, BSD 许可证被认为是 **copycenter** (中间版权) , 界乎标准的 **copyright** 与 **GPL 的 copyleft** 之间。GPL 强迫后续版本必须一样是自由软件, BSD 的后续版本可以选择要继续是 BSD 或其他自由软件条款或封闭软件等等。

遵守 BSD 许可证的软件, 允许用作商业用途, 甚至可按照专属许可证进行再发布。比较著名的例子如微软产品中引入了 BSD 网络部分的代码, Mac OS X 中使用了不少 FreeBSD 的组件。也可以将一部分遵照 BSD 许可证发布, 另外一些采取其他许可证。

GPL

GNU 通用公共许可协议（GNU General Public License，缩写：GNU GPL、GPL），是一个广泛被使用的自由软件许可协议条款，最初由理查德·斯托曼（Richard Matthew Stallman）为 GNU 计划而撰写。

GPL 给予了计算机程序自由软件（free software）的定义，并且使用 Copyleft 来确保程序的自由被完善的保留。

GPL 与其他一些更“许可的”自由软件许可证（比如 BSD 许可证）相比，主要区别就在于 GPL 寻求确保“自由”能在复制件及演绎作品中得到保障。它通过一种由斯托曼发明的叫 **Copyleft** 的法律机制实现，即要求 GPL 程序的演绎作品也要在 GPL 之下。相反，BSD 式的许可证并不禁止演绎作品变成专有软件。



“自由”：free speech vs. free beer

GPL 授予程序接受人以下权利，或称“自由”：以任何目的运行此程序的自由；再发行复印件的自由；改进此程序，并公开发布改进的自由（前提是能得到源代码）。

“自由”

- 使用的自由：可以不受任何限制地使用软件
- 研究的自由：可以获得软件源代码、研究软件运作方式
- 散布的自由：可以自由复制软件及散布给他人
- 改良的自由：可以自行改良软件并散布改良后的版本

“病毒”

GPL 条款规定演绎作品也必须是 GPL 的：对已发布的软件所作的任何修改都必须以相同的许可证发布。

“自由”：free speech vs. free beer

GPL 授予程序接受人以下权利，或称“自由”：以任何目的运行此程序的自由；再发行复印件的自由；改进此程序，并公开发布改进的自由（前提是能得到源代码）。

“自由”

- 使用的自由：可以不受任何限制地使用软件
- 研究的自由：可以获得软件源代码、研究软件运作方式
- 散布的自由：可以自由复制软件及散布给他人
- 改良的自由：可以自行改良软件并散布改良后的版本

“病毒”

GPL 条款规定演绎作品也必须是 GPL 的：对已发布的软件所作的任何修改都必须以相同的许可证发布。

“自由”：free speech vs. free beer

GPL 授予程序接受人以下权利，或称“自由”：以任何目的运行此程序的自由；再发行复印件的自由；改进此程序，并公开发布改进的自由（前提是能得到源代码）。

“自由”

- 使用的自由：可以不受任何限制地使用软件
- 研究的自由：可以获得软件源代码、研究软件运作方式
- 散布的自由：可以自由复制软件及散布给他人
- 改良的自由：可以自行改良软件并散布改良后的版本

“病毒”

GPL 条款规定演绎作品也必须是 GPL 的：对已发布的软件所作的任何修改都必须以相同的许可证发布。

- ① The freedom to use the software for any purpose.
- ② The freedom to change the software to suit your needs.
- ③ The freedom to share the software with your friends and neighbors.
- ④ The freedom to share the changes you make.





Copyright

Protects work of the Author
from unauthorized copying
or selling the work



Copyleft

Provides a method for work
to be modified & distributed
back to the community



源代码安装 | 开源许可证 | 选择



教学提纲

1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

● 选择软件

- 下载软件
- 编译和安装

4

脚本安装

5

使用 bioconda 管理生信软件

6

回顾与总结

- 总结
- 思考题



- 最新的正式版本
一般以“LATEST”进行标记，通常情况下应该下载、安装正式版本。
- 在多个平台上稳定运行的以前的版本
- 用于测试的开发版本
测试版本通常称为 alpha、beta、候选版本（RC, release candidate）或每日快照（daily snapshot），它是潜在不稳定的，使用时必须小心。



1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

- 选择软件

- 下载软件

- 编译和安装

- 脚本安装

- 使用 bioconda 管理生信软件

- 回顾与总结

- 总结

- 思考题



文件格式

下载的源代码文件内容通常由 tar、gzip、bzip 命令打包压缩过，形成的文件称为 tarball，其典型的文件扩展名是.tgz、.tar.gz 或.tar.bz2。

下载技巧

- Web 浏览器：Firefox, Chrome
- FTP 客户端：FileZilla
- 命令行：wget, lftp, curl, aria2(aria2c), httpie(http)

文件校验

- md5：md5sum, md5deep
- SHA1：sha1sum, sha1deep
- SHA256：sha256sum, sha256deep
- CRC-32：crc32

文件格式

下载的源代码文件内容通常由 tar、gzip、bzip 命令打包压缩过，形成的文件称为 tarball，其典型的文件扩展名是.tgz、.tar.gz 或.tar.bz2。

下载技巧

- Web 浏览器：Firefox, Chrome
- FTP 客户端：FileZilla
- 命令行：wget, lftp, curl, aria2(aria2c), httpie(http)

文件校验

- md5 : md5sum, md5deep
- SHA1 : sha1sum, sha1deep
- SHA256 : sha256sum, sha256deep
- CRC-32 : crc32

文件格式

下载的源代码文件内容通常由 tar、gzip、bzip 命令打包压缩过，形成的文件称为 tarball，其典型的文件扩展名是.tgz、.tar.gz 或.tar.bz2。

下载技巧

- Web 浏览器：Firefox, Chrome
- FTP 客户端：FileZilla
- 命令行：wget, lftp, curl, aria2(aria2c), httpie(http)

文件校验

- md5：md5sum, md5deep
- SHA1：sha1sum, sha1deep
- SHA256：sha256sum, sha256deep
- CRC-32：crc32

教学提纲

1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

- 选择软件

- 下载软件

● 编译和安装

脚本安装

使用 bioconda 管理生信软件

回顾与总结

- 总结

- 思考题



准备工作

- ① 下载软件：wget -c https://url/software.tar.gz
- ② 提取文件：tar -xzvf software.tar.gz
- ③ 切换目录：cd software

安装软件

- ① 配置环境：./configure (分析程序构建环境)
- ② 编译软件：make (只构建需要构建的内容)
- ③ 安装软件：make install



准备工作

- ① 下载软件：wget -c https://url/software.tar.gz
- ② 提取文件：tar -xzvf software.tar.gz
- ③ 切换目录：cd software

安装软件

- ① 配置环境：./configure (分析程序构建环境)
- ② 编译软件：make (只构建需要构建的内容)
- ③ 安装软件：make install

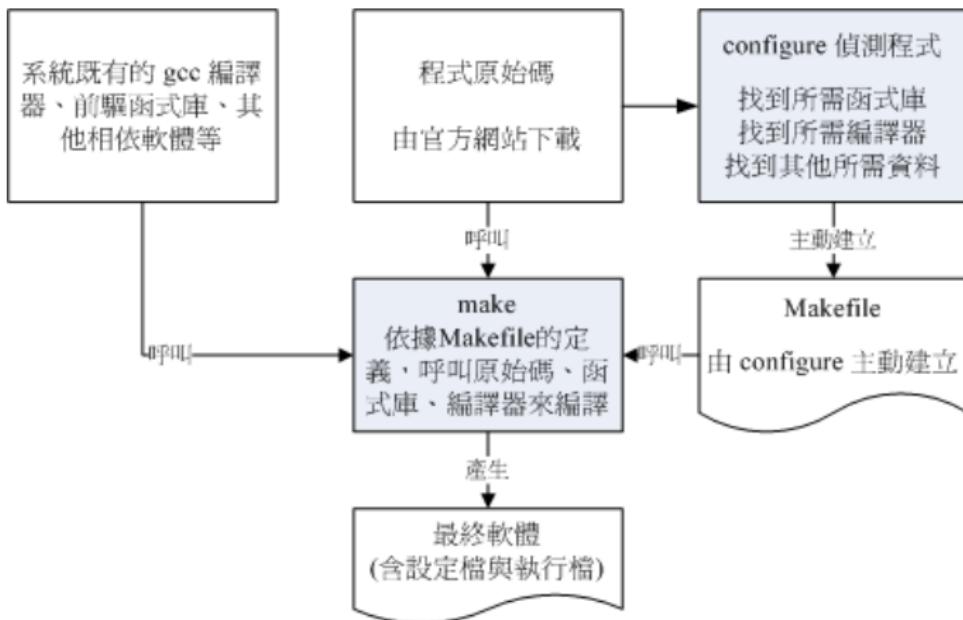


补充说明

- 阅读软件的指南或说明：vim INSTALL 或 vim README
- 指定软件的安装目录：./configure --prefix=PATH
- 安装软件前进行测试：make test 或 make check
- 以超级用户身份安装软件：sudo make install
- 删 除 编 译 产 生 的 临 时 文 件：make clean



源代码安装 | 编译和安装



1

引言

2

二进制软件包管理

3

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

4

源代码安装

5

- 源代码
- 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装

脚本安装

6

回顾与总结

- 总结
- 思考题



基本步骤

- ① 下载软件 : wget -c https://url/software.tar.gz
- ② 提取文件 : tar -xzvf software.tar.gz
- ③ 切换目录 : cd software
- ④ 查阅说明 : vim README
- ⑤ 安装软件 : ./setup.sh 或 ./install.sh



- 1 引言
- 2 二进制软件包管理
 - dpkg 与 APT 包管理
 - RPM 与 Yum 包管理
 - 二进制软件包管理比较
- 3 源代码安装
 - 源代码
 - 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装
- 4 脚本安装
- 5 使用 bioconda 管理生信软件
- 6 回顾与总结
 - 总结
 - 思考题



conda

Conda 是开源的包管理系统和环境管理系统，可以安装软件包的多个版本和依赖，而且各环境之间可以很方便得切换。Conda 支持 Linux, Mac OS X 和 Windows 系统。

Conda 主要为 Python 程序所创建，但是可以打包和分布任意软件。
Package, dependency and environment management for any language
—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN.

Conda 有多个版本，包括 Anaconda, Anaconda Server 和 Miniconda。



安装与配置

① 下载

```
wget -c https://url/  
Miniconda3-latest-Linux-x86_64.sh
```

② 安装

```
bash Miniconda3-latest-Linux-x86_64.sh
```

③ 配置（修改.bashrc）

```
export PATH="PathToMiniconda/bin:$PATH"
```

④ 配置.bashrc（修改.bash_profile 或者.profile）

```
source $HOME/.bashrc
```

补充说明

- 安装 Miniconda 的过程中可以让其自动配置.bashrc，从而避免手动配置的繁琐。
- 一般.bash_profile 里面已经有加载.bashrc 的命令了；而.profile 可能需要手动创建。

安装与配置

① 下载

```
wget -c https://url/  
Miniconda3-latest-Linux-x86_64.sh
```

② 安装

```
bash Miniconda3-latest-Linux-x86_64.sh
```

③ 配置（修改.bashrc）

```
export PATH="PathToMiniconda/bin:$PATH"
```

④ 配置.bashrc（修改.bash_profile 或者.profile）

```
source $HOME/.bashrc
```

补充说明

- 安装 Miniconda 的过程中可以让其自动配置.bashrc，从而避免手动配置的繁琐。
- 一般.bash_profile 里面已经有加载.bashrc 的命令了；而.profile 可能需要手动创建。

bioconda

Bioconda is a channel for the conda package manager specializing in bioinformatics software. Bioconda consists of:

- a repository of recipes hosted on GitHub
- a build system that turns these recipes into conda packages
- a repository of >1500 bioinformatics packages ready to use with conda install
- Over 130 contributors that add, modify, update and maintain the recipes



Using bioconda

Bioconda supports only 64-bit Linux and Mac OSX.

- ① Install conda
- ② Set up channels (It is important to add them in this order)

```
1 conda config --add channels defaults
2 conda config --add channels bioconda
3 conda config --add channels conda-forge
```

- ③ Install packages

```
1 # install into the current conda
  environment:
2 conda install bwa
3 # a new environment can be created
4 conda create -n aligners bwa bowtie
```

镜像

- 清华 (TUNA) 镜像

<https://mirrors.tuna.tsinghua.edu.cn>

- 中科大 (USTC) 镜像

<https://mirrors.ustc.edu.cn>

添加镜像 (1/2)

- Anaconda

```
1 conda config --add channels TUNA/anaconda/
   pkgs/free/
2 conda config --add channels TUNA/anaconda/
   pkgs/main/
3 conda config --set show_channel_urls yes
```

镜像

- 清华 (TUNA) 镜像

<https://mirrors.tuna.tsinghua.edu.cn>

- 中科大 (USTC) 镜像

<https://mirrors.ustc.edu.cn>

添加镜像 (1/2)

- Anaconda

```
1 conda config --add channels TUNA/anaconda/
   pkgs/free/
2 conda config --add channels TUNA/anaconda/
   pkgs/main/
3 conda config --set show_channel_urls yes
```

添加镜像 (2/2)

- bioconda

```
1 conda config --add channels TUNA/anaconda/  
cloud/bioconda/
```

- Conda Forge

```
1 conda config --add channels TUNA/anaconda/  
cloud/conda-forge/
```



- conda 使用手册
- bioconda
- Anaconda 镜像使用帮助
- 生物信息学软件的安装与维护



- 1 引言
- 2 二进制软件包管理
 - dpkg 与 APT 包管理
 - RPM 与 Yum 包管理
 - 二进制软件包管理比较
- 3 源代码安装
 - 源代码
 - 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装
- 4 脚本安装
- 5 使用 bioconda 管理生信软件
- 6 回顾与总结
 - 总结
 - 思考题



1

引言

2

二进制软件包管理

- dpkg 与 APT 包管理
- RPM 与 Yum 包管理
- 二进制软件包管理比较

3

源代码安装

- 源代码
- 开放源代码许可证

● 选择软件

● 下载软件

● 编译和安装

脚本安装

使用 bioconda 管理生信软件

6

回顾与总结

● 总结

● 思考题



知识点

- 软件包管理：软件包的类型，管理系统
- 二进制软件包管理：dpkg 与 APT, RPM 与 Yum
- 源代码安装：开源许可证，版本选择，安装步骤
- 脚本安装：基本步骤

技能

- Ubuntu 中的软件管理
- CentOS 中的软件管理
- 通过源代码安装软件
- 使用 conda 管理生信软件



- 1 引言
- 2 二进制软件包管理
 - dpkg 与 APT 包管理
 - RPM 与 Yum 包管理
 - 二进制软件包管理比较
- 3 源代码安装
 - 源代码
 - 开放源代码许可证

- 选择软件
- 下载软件
- 编译和安装
- 4 脚本安装
- 5 使用 bioconda 管理生信软件
- 6 回顾与总结
 - 总结
 - 思考题



- ① Ubuntu 和 CentOS 使用的软件包管理系统分别是什么？
- ② 列举 dpkg 和 APT 软件包管理中的常用命令及其作用。
- ③ 列举 RPM 和 Yum 软件包管理中的常用命令及其作用。
- ④ 列举几个常见的开放源代码许可证。
- ⑤ GPL 授予程序使用者哪些 “自由” ？
- ⑥ 通过源代码安装软件的基本步骤是什么？
- ⑦ 如何使用 conda 管理生信软件？



下节预告

你知道哪些编辑器？用过哪些编辑器？有何感受？
总结编辑文件时的常见操作。你是如何完成这些操作的？



Powered by



TeX L^AT_EX X_ET_EX Beamer