

天津医科大学理论课教案首页

(共 5 页、第 1 页)

课程名称：Linux 系统概论	课程内容/章节：软件安装 / 第 19 章
教师姓名：伊现富	职称：讲师
授课对象：生物医学工程与技术学院 2013 级生信班（本）	教学日期：2015 年 6 月 16 日 13:30-15:20
授课方式：理论讲授	学时数：2
	教材版本：Unix 入门经典，第 1 版

教学目的与要求（分掌握、熟悉、了解、自学四个层次）：

- 掌握 APT 的使用方法，Yum 的使用方法，GPL 的思想，源代码安装软件的主要步骤。
- 熟悉 dpkg 的使用方法，RPM 的使用方法，命令行下载软件的方法，脚本安装软件的方法。
- 了解各种开源许可证，源代码编译的过程。
- 自学其他的二进制软件包管理方法。

授课内容及学时分配：

- (5') 引言与导入：介绍软件包管理的概念、常见的软件包，总结 Linux 中的二进制软件包管理系统。
- (45') 二进制软件包管理：详细讲解 dpkg 和 APT、RPM 和 Yum 进行软件包管理的命令，简要比较各种二进制软件包的管理。
- (40') 源代码安装：介绍常见的开源许可证，讲解 GPL 的思想，介绍选择和下载软件的方法，讲解从源代码编译和安装软件的步骤。
- (5') 脚本安装：介绍通过脚本安装软件的过程。
- (5') 总结与答疑：总结授课内容中的知识点与技能，解答学生疑问。

教学重点、难点及解决策略：

- 重点：APT 和 Yum 的使用，从源代码安装软件的步骤。
- 难点：dpkg 和 RPM 的使用。
- 解决策略：通过实例讲解与操作演示帮助学生理解、记忆。

专业外语词汇或术语：

开放源代码（open source） 自由软件（free software）

辅助教学情况：

- 多媒体：dpkg 和 APT、RPM 和 Yum 二进制软件包的管理命令，各种二进制软件包管理的比较，开源许可证。
- 板书：从源代码安装软件的步骤。
- 演示：dpkg 和 APT、RPM 和 Yum 的使用。

复习思考题：

- Ubuntu 和 CentOS 等常见 Linux 发行版使用的软件包管理系统是什么？
- 列举 dpkg 和 APT 软件包管理中的常用命令及其作用。
- 列举 RPM 和 Yum 软件包管理中的常用命令及其作用。
- 列举几个常见的开放源代码许可证。
- GPL 授予程序使用者哪些“自由”？
- 通过源代码安装软件的基本步骤是什么？

参考资料：

- (美) Harley Hahn 著，张杰良 译。Unix & Linux 大学教程，清华大学出版社，2010。
- 鸟哥 著，王世江 改编。鸟哥的 Linux 私房菜——基础学习篇（第三版），人民邮电出版社，2010。
- 维基百科等网络资源。

主任签字：

年 月 日

教务处制

一、引言与导入 (5 分钟) (通过和 Windows 中软件、软件管理的比较进行导入)

1. 软件包管理系统

- 自动安装卸载、配置和升级软件包的工​​具组合
- 大大简化在 Linux 发行版中安装软件的过程

2. 软件包

- 二进制包 (预编译的软件包)
 - deb 软件包
 - rpm 软件包
- 源代码安装包
- 脚本安装包

*.exe
*.msi



*.bin
*.dmg



*.sh
*.deb
*.rpm



3. 二进制包管理系统

- dpkg 及其前端 APT: Ubuntu, Debian, Deepin
- RPM 及其前端 Yum: Red Hat Enterprise Linux, CentOS, Fedora
- ZYpp 及其前端 Zypper: SUSE, openSUSE
- 其他: urpmi (Mandriva Linux, Mageia Linux, ROSA Linux) , pacman (Arch Linux) , slapt-get (Slackware) , Portage (Gentoo)

二、二进制软件包管理 (45 分钟)

1. dpkg 与 APT

(1) 简介

- dpkg: 底层工具, Debian 软件包管理器的基础
- APT: dpkg 的前端, Debian 及其派生发行版的软件包管理器
- Aptitude: APT 的前端 (文字终端)

(2) 【难点】dpkg (实例讲解、操作演示)

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 dkg -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

(3) 【重点】APT (实例讲解、操作演示)

- 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

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 isof 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

(4) PPA

- 添加 PPA 源: `sudo add-apt-repository ppa:user/ppa-name`
- 更新所有源: `sudo apt-get update`
- 安装软件: `sudo apt-get install <package_name>`

2. RPM 与 Yum

(1) 简介

- RPM: rpm 软件包管理器
- Yum: RPM 的前端

(2) 【难点】RPM (实例讲解、操作演示)

- 功能
 - 查询: -q
 - 校验: -V
 - 安装: -i
 - 删除: -e
 - 升级: -U
- 选项
 - 通用: -v
 - 选择: -a, -f, -p
 - 查询: -l, -i, -c, -d, -R, -s
 - 安装: -h, --nodeps, --prefix, --test, --replacepkgs, --force

(3) 【重点】Yum (实例讲解、操作演示)

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 {package-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}	Find out what dependencies a rpm file has	rpm -qpR mediawiki-1.4rc1-4.i586.rpm rpm -qR bash

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

3. 比较

三、源代码安装 (40 分钟)

1. 源代码

- 源代码: 创建软件的原始数据, 源代码 + 文档
- 开放源代码: 以源代码形式提供的软件, 带有特定许可条款

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>	urpme <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 <i>pattern</i> <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 -is	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 repo to /etc/yum.repos.d/)	urpmi.addmedia <i>name path</i>
Remove repository	(edit /etc/apt/sources.list)	zypper removerepo <i>name</i>	(remove repo from /etc/yum.repos.d/)	urpmi.removemedi <i>media</i>

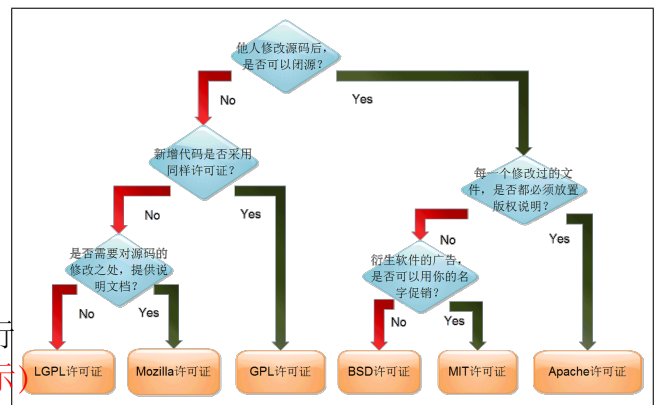
2. 开源许可证

- BSD 许可证: copycenter
- GPL 许可证: copyleft
 - 使用的自由: 使用不受任何限制
 - 研究的自由: 获得源代码并研究
 - 散布的自由: 复制散布软件
 - 改良的自由: 改良软件并散布

3. 选择软件: LATEST, alpha, beta, ……

4. 下载软件: 浏览器, FTP 客户端, 命令行

5. 【重点】安装软件 (实例讲解、操作演示)



• 准备工作

- (1) 下载软件: `wget -c 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`
- (2) 编译软件: `make`
- (3) 安装软件: `make install`

• 其他工作

- 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`

四、脚本安装 (5 分钟)

1. 下载软件: `wget -c X.tar.gz`
2. 提取文件: `tar -xzvf X.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`

五、 总结与答疑 (5 分钟)

1. 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

2. 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