软件安装

$ configure; make; make install

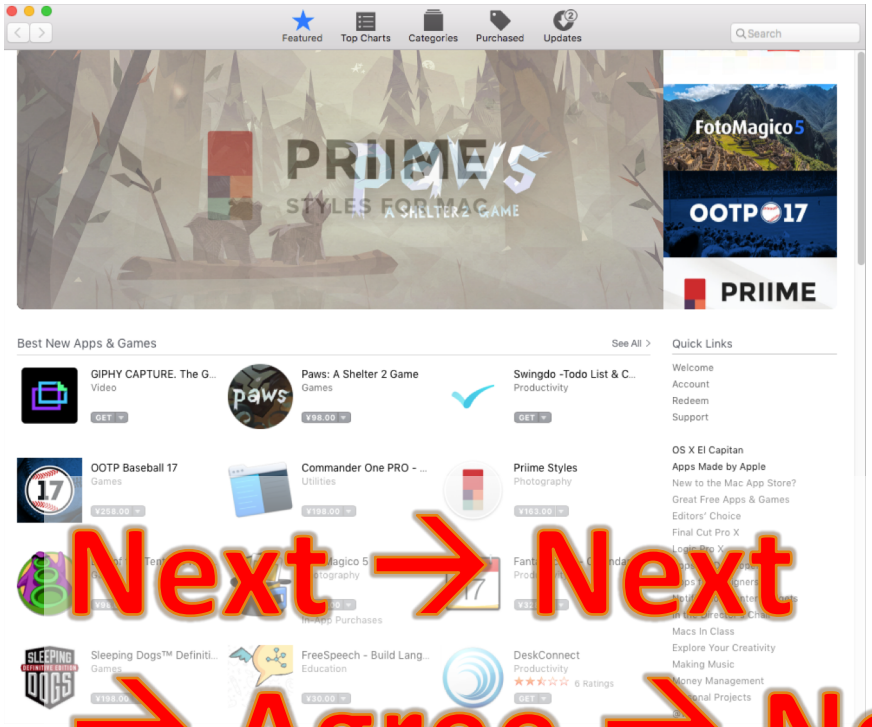
$ rpm

$ dpkg

$ yum

$ apt

- 编译源代码
- 安装软件包
- 在线安装



源代码安装

- 获取源代码，一般以tar.gz，或tar.bz2为后缀
  - 代码通过gzip，或者bz2进行了压缩，需要解压
  - $ tar –xvzf ***.tar.gz || tar –xvjf ***.tar.bz2
- $ configure： 生成Makefile，具有不同选项
- $ make： 利用Makefile进行编译和链接
- $ sudo make install：将生成的可执行文件、函数库等写入适当路径

make & Makefile

- $ make： 自动化软件构建工具
  - GNU make：是多数GNU/Linux安装的一部分
  - BSD make
  - Microsoft nmake
- Makefile中主要描述了源代码间的依赖关系与编译链接过程
- 通过读取Makefile文件，make调用shell命令，如gcc，g++，javac等来自动编译和链接target程序

Makefile example

```
demo: main.o function.o
        g++ -o demo main.o function.o
main.o: main.cpp     function.h
        g++ -c main.cpp
function.o: function.cpp function.h
        g++ -c function.cpp
install: demo
        sudo mv demo /usr/bin
clean:
        rm *.o
```

```
# 目标 demo依赖于main.o 和 function.o
# 若demo不存在，或所依赖的o文件有修改
# 则调用g++指令进行编译

#main.o 依赖于main.c & function.h 文件

#function.o 依赖于function.cpp & function.h

# 目标 install 依赖于demo
# 若demo被创建了或者修改了，
# 则调用mv 指令将其移送到指定目录

# 对中间文件进行清除
```


pwd: ~/demo/ch04/compile

RPM (RedHat Package Manager)

- 由Red Hat 公司开发，应用于Fedora, CentOS, SuSE ...
- 预先由软件商在特定平台（软硬件）上进行编译打包，形成**二进制安装**文件（*.rpm），直接安装即可
  - 不用重编译，且会对系统环境进行先验，也便于卸载
  - 要求安装环境与打包环境一致或相当，需要满足一系列的依赖条件（特别是动态链接库）

RPM (硬件适用平台)

平台名称	说明
i386	几乎适用于所有的x86平台，i 是指 Intel，即从386开始的所有芯片
i586	针对586等级的优化, Pentum MMX, AMD K5, 6 后
i686	针对Pentum II 后全系统，基本通用
x86_64	针对64位平台
arm	针对arm平台
noarch	没有硬件平台要求，往往是shell脚本

RPM 基本使用

- $ rpm –ivh *file.rpm*　　# 安装软件
  - -i，--install：安装
  - -v（verbose）：显示安装细节
  - -h，--hash：显示安装进度
- $ rpm –[F|U]vh *file.rpm*　　# 更新软件
  - -F，--freshen：若未安装过，则不安装，否则更新
  - -U，--upgrade：若未安装过，则直接安装，否则更新

RPM 基本使用

- $ rpm -qa *software_name*　　# 查询软件安装情况
  - -q，--query：查询
  - -a，--all：查询所有本地系统
- $ rpm -e *software_name*　　# 卸载安装的软件
  - -e，--erase：卸载

DPKG(Debian PacKaGer)

- 由Ian Murdock于1993年创建
- 和RPM非常相似，利用预编译的二进制安装文件(*.deb)被普遍应用在基于Debian的GNU/Linux发行版中，如Ubuntu中

DPKG基本使用

- $ dpkg –i *file.deb*　　# 安装 / 更新软件
  - -i，--install：安装
- $ dpkg -qa *name _pattern*　　# 查询软件安装情况
  - -l，--list：查询
  - -a，--all：查询所有本地系统
- $ dpkg -r *software_name*　　# 卸载安装的软件
  - -r，--remove：卸载

DPKG基本使用

- dpkg –I *file.deb*　　# 查询安装包细节
  - -I，--info：查询安装包信息
- $ file ： 查询文件类型
- $uname -a：查询当前系统信息，包括硬件架构

YUM (Yellowdog Updater Modified)

- 在线安装，以解决软件依赖问题
- 将软件保存于在线软件仓库（repository）中，并根据软件间的依赖关系相关链接、管理，形成软件清单
- 当需要某个软件时，将“本地已安装软件列表”与在线的关于目标软件的“依赖软件清单”比较，将所有依赖软件与目标软件一次性下载安装

YUM基本使用

- $ yum search *software_pattern*　　#搜索软件
- $ yum info *software_name*　　# 展示软件信息
- $ yum list　　#列出服务器上的软件
  - Installed Packages: ...
  - Available Packages: ...
- $ yum list updates　　#列出当前可升级的软件

YUM基本使用

- $ yum install *software_name*　# 安装软件
- $ yum update *software_name*　# 更新软件
- $ yum remove *software_name*　# 卸载软件

软件仓库清单

- 任何一台主机都可以设立为YUM仓库，以HTTP/FTP方式提供服务
  - 主机包含的软件类别、数量不一
  - 主机的软硬件、网络环境不同
- 需要在本地维护一些软件多、网速快的软件库列表
  - /etc/yum.conf：维护一个主YUM仓库
  - /etc/yum.repos.d/*.repo： 维护若干个YUM仓库

YUM软件仓库描述

[examplerepo]
name=Example Repository
baseurl=http://mirror.cisp.com/CentOS/6/os/i386/
enabled=1
gpgcheck=1
gpgkey=http://mirror.cisp.com/CentOS/6/os/i386/RPM-GPG-KEY-CentOS-6

[唯一的软件库ID]
name=软件库描述信息
baseurl=软件库连接信息
enabled=1：启动该软件库；0：关闭该软件库
gpgcheck=1：需要核验电子签名；0：不需要核验
gpgkey=电子签名地址

YUM基本使用

- 在修改软件库列表后，可以通过clean来消除本地信息与在线信息的不同步
- $ yum clean [packages|headers|all]
  - packages：清除已下载的软件包
  - headers：清除已下载的软件描述
  - all：清除所有的软件库资料

APT(Advanced Packaging Tools)

- 类似YUM，构建在dpkg上，在Debian系列系统下使用
- $ apt-cache search *software_pattern*　#检索软件
- $ apt-cache show *software_name*　#显示软件信息
- $ sudo apt-get install *software_name*　#软件安装
- $ sudo apt-get remove *software_name* #软件卸载
- $ sudo apt-get upgrade　#更新所有已安装的软件

APT 升级版

- 自 Ubuntu 16.04 起，可统一使用 apt 指令，以简化一般性软件管理操作，而不必细分apt-get, apt-cache：
  - $ apt search
  - $ apt show
  - $ apt install
  - $ apt remove
  - $ apt update
  - $ apt upgrade

并非简单封装旧有工具，而是彻底地重新设计与实现，但基本使用方式不变

APT软件仓库清单

- APT同样需要在本地维护软件库源清单，多是镜像
  - /etc/apt/sources.list

deb http://mirrors.ustc.edu.cn/ubuntu/ **utopic** main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ utopic-security main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ utopic-updates **main** restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ utopic-proposed main **restricted** universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ utopic-backports main restricted **universe** multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic main restricted universe **multiverse**
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic-security main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic-updates main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic-proposed main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ **utopic-backports** main restricted universe multiverse

APT软件仓库清单

- APT同样需要在本地维护软件库源清单，多是镜像
  - /etc/apt/sources.list

系统代号：14.10
Utopic Unicorn，乌托邦的独角兽

deb http://mirrors.ustc.edu.cn/ubuntu/ utopic main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ utopic-security main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ utopic-updates main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ utopic-proposed main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ utopic-backports main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic-security main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic-updates main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic-proposed main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ utopic-backports main restricted universe multiverse

二进制安装

源代码安装

库目标：后续更新软件

APT软件库源

- main 即“基本”组件，其中只包含匹配Ubuntu的许可证要求并可以从Ubuntu团队中获得支持的软件，致力于满足日常使用，位于这个组件中的软件可以确保得到技术支持和及时的安全更新。
- restricted 即“受限”组件，其中包含了非常重要的，但并不具有合适的自由许可证的软件，例如只能以二进制形式获得的显卡驱动程序。
- universe 即“社区维护”组件，其中包含的软件种类繁多，均为自由软件，但都不为Ubuntu团队所支持。
- multiverse 即“非自由”组件，其中包括了不匹配自由软件要求而且不被Ubuntu团队支持的软件包，通常为商业公司编写的软件。

Ubuntu - 维基百科

APT软件库源

	自由软件	非自由软件
官方支持	Main	Restricted
非官方支持	Universe	Multiverse

- **main** 即“
可以从Ubu
于这个组件
- **restricted**
合适的自由
动程序。
- **universe** 即“社区维护”组件，其中包含的软件种类繁多，均为自由软件，但都不为Ubuntu团队所支持。
- **multiverse** 即“非自由”组件，其中包括了不匹配自由软件要求而且不被Ubuntu团队支持的软件包，通常为商业公司编写的软件。

Ubuntu - 维基百科

APT软件库源

- **main** 即“基本”组
可以从Ubuntu团队
于这个组件中的软
- **restricted** 即“受限
合适的自由许可证
动程序。
- **universe** 即“社区维护”组件，其中包含的软件种类繁多，均为自由软件，但都不为Ubuntu团队所支持。
- **multiverse** 即“非自由”组件，其中包括了不匹配自由软件要求而且不被Ubuntu团队支持的软件包，通常为商业公司编写的软件。

修改软件库列表后，可以通过update来获得最新的软件列表

$ sudo apt udpate

Ubuntu - 维基百科

Linux下的软件安装

系统代表	管理机制	指令	在线安装指令
Red Hat / Fedora	RPM	rpm	yum
Debian / Ubuntu	DPKG	dpkg	apt


HAVE FUN