Manage software and services

Ngoc Nguyen Tran

Contents

I. Principle of software management
II. Install software from source code
III. Install software by using management tools
IV. Manage software repositories

Principle of software management

Software components	Manage sofware	How to manage
• Execution files • Libraries • Configuration files • Temporary files	• Install • Remove • Reconfigure • Get information	• Independent • Script for each software • Software database • Tools

Install software from source codes

- Install source code
- Compile source code
  - Install dependent packages if needed
- Install the software
  - Installation scripting
- Configure software
  - Configuration scripting
- Remove software
  - Removal scripting
- Scripts for all above operations
- Makefile, Automake, make, other developing tools

Install software from managed program

- Program to install/remove/configure
- Check the conflict with other software
- Use tools/programs to manage software
  - Software are packed as packages
  - Software database
  - Detect software conflict (redundance, missing, different versions)

Tools to manage software

- Manage software packages + package database
- Manage list of software repository
- Install/remove software automatically
- Have interactive interface or GUI
- Simple
- Can install redundant software/packages

Tools to manage software

	Redhat	Debian
Manage packages	rpm	dpkg
Package Management System	yum, urpm*, dnf	apt-*
Interactive interface	dselect, taskshell	aptitude
GUI	krpm, yumex	synaptic

rpm

- Manage software packages for RedHat distributions
  - Create software package
  - Show information
  - Decompress
  - Install
  - Remove
- Manage software/package databases

Examples of using rpm

- Common commands
  - rpm --checksig rpm_file
    - Check the integrity of the package
  - rpm –ivh [--nodeps] rpm_file
    - Install a rpm package (--nodeps to ignore the dependent packages)
  - rpm -qpR rpm_file
    - Check the dependent packages before installation
  - rpm -q application
    - Check whether a package is installed or not?
  - rpm -qa --last
    - Show the last installed package (no –last option means listing all installed packages)
  - rpm -ev application
    - Remove an installed package
  - rpm -Uvh application
    - Update an installed package
- Note:
  - **rpm_file** is rpm file to install
  - **application** is installed application/software

dpkg

- Similar to rpm but for Debian distributions
  - Create software package
  - Show information
  - Decompress
  - Install
  - Remove
  - Reconfigure
- Manage software/package databases

Examples of using dpkg

- Common commands
  - dpkg -c/ -i/ -r deb_file
    - Show/ install/ remove a deb file
  - dpkg -l
    - List all installed packages
  - dpkg –l [application]
    - Check whether the package is installed or not
  - dpkg –-update-avail application
    - Update the application
- Note:
  - **deb_file** is deb file to install
  - **application** is installed application/software

Manage software repositories

- Manage multiple software
- Manage multiple software repositories
- Automatically install/remove software if needed
- E.g. yum, apt

yum

- Can use software repositories to download/ install/ remove software easier and simpler than using rpm command
- Some common yum commands :
  - yum install/remove/update application
  - yum list/search/info application
  - yum list | less
  - yum update/check-update
  - yum grouplist/ groupinstall/ groupupdate/ groupremove “group name”
  - yum repolist [all]
  - yum --enablerepo=epel install application

apt-*

- Same as yum but for Debian distributions
- Some operations
  - apt-get install/ download/ remove/ source application
  - apt-get clean
  - apt-get update/ upgrade
  - Note: New Debian OS: apt is a subset of apt-get or apt-cache