

Package Management

Michael Tsai

2016/4/27

PXEBoot

- Diskless host operation
- Boot ROM on your network interface card uses DHCP to get the initial network setting
 - Including the location of the boot file at a TFTP server
- Then the boot file is used to initiate the operating system
- The root file system can be mounted from a server (for example, using NFS)

Setup your own PXE environment

- <http://goo.gl/Fgr0JT> ← Ubuntu 16.04 LTS 64 bit server ISO
- <http://goo.gl/duqdjS> ← Guide to setup a PXEBoot environment

Software Installation

- System administrators often perform the following:
 - **Automating** mass installations of OS's (not covered today. Additional:
 - Maintaining custom OS configurations for the **local** environment
 - Keeping systems and applications patched and up to date
 - Managing add-on software packages

Packages - WHY?

- The old ways - .tar.gz (“zipped tar ball”)
- Advantages of the use of packages:
 - Ability to **roll back** to the original state
 - Quick (**binary** packages v.s. compile from **source**)
 - Handles **dependency**
 - Ability to run scripts during installation

Automation is important

- Install software for once is easy
- Install the same software for 10 times is boring and painful
- Human errors
(typo, you forgot to install one of them)
- Human intervention takes the most time.
- Use a tool to **automate** the process!

The Old Ways

- tar - “store and extract files from a **tape** or **disk** archive”
- Example:
tar zcvf nasa117.tar.gz nasa117/
(create a tar archive with all files in nasa117/)
tar zxvf nasa117.tar.gz
(extract nasa117.tar)
- Commonly used options:
c: create; x: extract; f: specify the file name
v: verbose; z: gzipped; j: bziped; t: list the content

Common Linux Package Management System

- RPM: Red Hat Package Manager
 - rpm: package handler
 - yum: Yellowdog Updater, Modified (package finder, handle dependency)
- .deb: Debian distribution package
 - dpkg: package handler
 - APT: advanced package tool (package finder, handle dependency)



debian



dpkg

- Commonly used commands:
- List all installed packages: `dpkg -l`
Example: `dpkg -l | grep 217`
- Install a package: `dpkg -i 217-base-20150508.deb`

High-level Package Management System

- Simplify the task of locating & downloading packages
- Automate the process of updating/upgrading systems
- Facilitate the management of interpackage dependencies

Package Repository

- Place to store all the packages
- Default: points to HTTP or FTP servers where the distributor has control
- Concept:
 - Release**: a snapshot of the package universe
 - Component**: a subset of the software within a release.
 - Architecture**: a specific class of hardware
 - Individual Packages**: self-explained
- Binary packages —>
not always optimized for the specific machine you have.

Before we start ...

- Set up `/etc/apt/sources.list`
 - Make a copy of the original `sources.list`
`sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak`
 - Change all server names in `/etc/apt/sources.list` from `us.archive.ubuntu.com` to `free.nchc.org.tw` (or other servers in Taiwan)
Cheat sheet:
`sudo sed -i 's/us.archive.ubuntu.com/free.nchc.org.tw/g'`

Before we start ...

- Refresh apt-get's cache of package information
sudo apt-get update
- Let's take a look at /etc/apt/sources.list

deb | deb-src URL "distribution" list-of-components

universe: other Linux open source software

multiverse: include non-open-source content

Or, you can use apt-mirror

- apt-mirror will make a local version of the repository (把整個repository砍回家)
- First execution will download >100 GB of data
- Ref:
 - (1) <https://apt-mirror.github.io>
 - (2) How to setup a local mirror
https://www.howtoforge.com/local_debian_ubuntu_mirror

Install a package

- Search for the package:
`apt-cache search <regex>`
Example: `apt-cache search apache`
- Then install the package:
`apt-get install apache2`

Package Information

- Show information about a package
`apt-cache show <package name>`
- Name, Version, Maintainer, Architecture,
Description, Filename, ...
State: installed, not installed, removed
- Priority: required/important/standard/optional/extra

Package Information

- Dependencies: what is required by this package.
Depends / Recommends / Suggests
Conflicts / Replaces /Breaks

Removing a package

- Remove a package, but keep the configuration files
Example: `apt-get remove apache2`
- Remove a package, and also remove all the configuration files
Example: `apt-get purge apache2`

課堂小作業, part I

1. Update your `/etc/apt/sources.list` to use one of the Taiwan servers
2. Install the following packages:
`apache2`, `build-essential`, `dpkg-dev`, `debhelper`, `CDBS`, `dh-make`,
`lintian`
3. Use `apt-get` or `dpkg` to find one of the configuration files in
`apache2` package
4. Remove `apache2` package (but not purge)
5. Locate the configuration files
6. Show to one of the TAs that your `apache2` is already removed but
the configuration file is still there

Upgrade and Hold

- Upgrade all currently installed packages
(but do not install new dependencies)
`apt-get upgrade`
- Upgrade all currently installed packages and satisfy all
new dependencies
`apt-get dist-upgrade`
- Hold the version of a package
—>Newer version is not always good
`apt-mark hold/unhold <package>`

Additional Resources

- Slides from 3 years ago:
http://www.csie.ntu.edu.tw/~hsinmu/courses/_media/nasa_12fall/handout/package_managers.pdf
- Steps to Packaging:
<http://www.debian.org/doc/manuals/maint-guide/first.en.html>
- Debian 新維護人員手冊:
<http://www.debian.org/doc/manuals/maint-guide/>
- Get other package sources and see how other people do it.
apt-get source XXX

課堂小作業, part II

- Create a meta package <TA>