

\$Linux#

Chương 7. Quản lý phần mềm

- Nguyên tắc quản lý phần mềm
- Cài đặt phần mềm từ mã nguồn
- Công cụ quản lý gói
- Quản lý các kho phần mềm.

(SGK tr 95)

7.1 Nguyên tắc quản lý phần mềm

- **Các thành phần của phần mềm hệ thống**
- **Các thao tác thực hiện với phần mềm**
- **Cách thức quản lý phần mềm.**

Các thành phần của phần mềm hệ thống

- **Mã thực hiện**
- **Thư viện phần mềm**
- **Các tệp tin yêu cầu cấu hình để chạy**
- **Dữ liệu cung cấp tạm thời**

Các thao tác thực hiện với phần mềm

- Cài đặt phần mềm
- Gỡ bỏ phần mềm đã cài đặt
- Cài lại, sửa lỗi khi cài đặt bị lỗi
- Cấu hình phần mềm sau khi cài đặt

Cách thức quản lý phần mềm.

- Phần mềm độc lập có thể chạy
- Phần mềm yêu cầu các đoạn code (Script) kèm theo thư viện chuẩn .lib
- Có thể chạy với nhiều file đã đóng gói (rpm/ deb)

Bản phân phối Linux phổ biến

- **RHEL (Redhat, Fedora, CentOS) -RPM**
 - **Ubuntu / Linux Mint (deb)**
 - **Mandriva Linux (Mageia)**
 - **Slackware Linux**
 - **OpenSUSE**
 - **Arch Linux**
 - **Puppy Linux**
 - **Gentoo Linux**
 - **MeeGo (Laptop/ Tablet, smartphone)**
-
- RPM (Redhat Package Manager) là công cụ dùng để Installing, Uninstalling và Upgrading software cho hệ thống Linux.
 - DEB (Debian) là phần mở rộng của định dạng đóng gói phần mềm Debian và cũng là tên thường gọi cho các gói nhị phân tương tự. Giống như phần “Deb” của thuật ngữ Debian, nó bắt nguồn từ tên gọi của Debra, là bạn gái và bây giờ vợ cũ của người sáng lập của Debian Ian Murdock.

7.2 Cài đặt phần mềm từ mã nguồn

- **Yêu cầu cài đặt**
 - Phần cứng đáp ứng
 - Sự tương thích 32/64 bit
 - Sự phù hợp (theo họ Linux). RPM/ DEB
 - Theo phiên bản phần mềm (version)

7.3 Công cụ quản lý gói

Cài đặt file nguồn mở (nén)

Các file nén (nguồn mở)tar.gz; bz2;tgz; tbz2

Trình biên dịch GCC (Linux)

Giải nén trước khi cài

```
#tar [option] filename
```

option = tùy chọn kiểu giải nén tương ứng kiểu file

Vd: #tar **xvzf** /**zxv** bash-2.05-8.tar.gz

#tar **jxf** bash-2.05-8.tar.bz2

#tar **jxf** bash-2.05-8.tar.tbz2

#tar **zxf** bash-2.05-8.tar.tgz

Cài đặt từ file mã nguồn mở (# -root/ \$sudo user)

```
#./configure
```

```
#make
```

```
# make install
```

Hoặc #rpm --bt name-version-release.tar.gz

Công cụ 7Zip (Windows/ Linux)

Công cụ cho phép giải nén các file (zip, rar; tar,gz, bz2...)

Thảo luận:

Khác nhua cơ bản khi cài đặt từ file **tarball** và **rpm/ deb**

Hủy bỏ cài đặt từ nguồn mở

Vào thư mục mã nguồn đã giải nén, thực hiện lệnh sau

- `#Make uninstall`
- `#make clean`
- `#make distclean`

Nếu mất file mã nguồn thì không loại bỏ được; mặc định khi cài đặt sẽ lưu trong `/usr`

Bạn có thể lưu sang thư mục khác

```
#mkdir /TM1/Test
```

```
# ./configure -prefix = /TM1/Test
```

Xem hướng dẫn: `# ./configure - help`

Quản lý gói RPM

(redhat packet manager)

Đóng gói : rpm filename

Name-version-release.architecture.RPM

Lệnh thực hiện

rpm -i/-v/-h/-q/-e filename.rpm

Kiểm tra gói tin rpm bị xóa:

rpm -Va

Kiểm tra 1 file rpm thuộc phần mềm nào

rpm -qpi file.rpm

Xây dựng lại gói rpm

#rpm --rebuild package.src.rpm

Thảo luận: Nhận biết cấu trúc file RPM và sự khác nhau

l386.rpm/ i686rpm/ src.rpm

Quản lý gói RPM

(Biên dịch lại gói tin RPM rebuilding)

Đóng gói : `scr.rpm filename`

B1: Tạo file.spec (xem hướng dẫn tài liệu)

Lưu file spec vào thư mục: `/usr/src/radhat/SPECS` và
`/usr/src/radhat/SOURCES`

B2 : `#rebuild -ba file.spec`

- Khi thành công; tạo ra 3 file đóng gói
- Hai file dưới lưu trong `/usr/src/radhat/i386 (X86_64)`
-> `file.i386.rpm` hoặc `X86_64.rpm`
(same file .exe)

-> `-debug-info-file-rpm`

File dưới lưu trong `/usr/src/radhat/SRPMS`

-> file `scr.rpm` (để khi cần rebuilding)

Thảo luận: khi nào cần rebuilding một file RPM

Quản lý gói .deb (Debian - Ubuntu)

Đóng gói : deb filename

Cài đặt từ dòng lệnh : #dpkg

#dpkg -i (install) file.deb

#dpkg -r (remove) file.deb

#dpkg -P (remove all config file)

#dpkg --reconfigure file.deb // cấu hình lại file.deb

Công cụ quản lý Tarball

(cài mã nguồn mở .gz,tgz,bz2,.. trên Ubuntu)

Quản lý gói .bin (Ubuntu)

- ***Cài đặt file .bin***
- Download và save file .bin tới thư mục /TM1.
Mở Terminal và gõ ***cd TM1***
- \$ sudo chmod **+x** filename.bin
- \$./filename.bin

(lưu ý: file.binlaf định dạng ubuntu khác với file trong cấu hình Lab ảo Packet Tracer)

Quản lý gói .deb- Mandriva Linux

Đóng gói

Cài đặt từ dòng lệnh : **#urpm**

urpmi –file Cài đặt

urpme –file Xóa Cài đặt

urpmi –auto-orpans/ -auto-update/ -clean

Giao diện đồ họa (GUI)

Gurpmi / rpmdrake / smart

Chuyển đổi Debain (apt) sang RPM

\$sudo apt-get install alien / #apt-get install alien

Cài đặt gói tin alien

alien -k filename.rpm // để convert từ file .rpm -> .deb.

Công cụ quản lý phần mềm (gói tin)

APT (Debian linux – Ubuntu)

#apt **[install -f / clean/ update/ update --fix-missing]**

#YUM (Yellow Dog Manager) chạy trên hệ thống RHEL
(Redhat Linux Enterprise, trước đây là YUP (YU Updater))

#DNF (Dandified Packaging Tool) – nâng cấp và mạnh hơn YUM , DNF chỉ chạy trên Fedora22 trở lên.

Others:

up2date, urpm(Mandriva Linux), **Zypper, ZYpp** (OpenSUSE)

#Pacman (Arch Linux)

#ABS (Arch Build System)

#Potage (Gentoo Linux)

Slackware (Slackware Linux)

Cài đặt từ internet

Ubuntu: **#apt-get** install filename.apt

RHEL (Fedora, CentoOS)

\$sudo và # (quyền root); \$sudo chiếm quyền root

#YUM (Yellow Dog Manager) chạy trên hệ thống RHEL (redhat Linux Enterprise)

yum install file.rpm

yum search file.rpm

#DNF / \$sudo dnf (Dandified Packaging Tool) – nâng cấp và mạnh hơn YUM , DNF chỉ chạy trên Fedora22 trở lên

\$sudo dnf check-update

sudo dnf upgrade

sudo dnf search <tên gói>/ sudo dnf search all <tên gói>

sudo dnf info <tên gói>

sudo dnf repoquery --requires package // file phụ thuộc

sudo dnf install <tên gói 1> <tên gói 2> ... // xác nhận Y

sudo dnf install -y <tên gói>

sudo dnf install <tên gói>.rpm / cài đặt từ Local giống #rpm -ivh file.rpm

sudo dnf erase <tên gói>

sudo dnf update //nâng cấp Fedora

man dnf // xem hướng dẫn dnf

Thêm kho phần mềm

sudo dnf install --nogpgcheck <http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-24.noarch.rpm>

#zypper (openSUSE)

7.4 Quản lý các kho phần mềm

- **Công cụ quản lý phần mềm bậc cao**

rpm format (redhat, Fedora, CentOS)

urpmi

(#rpm -u filename.rpm)

- **Debian format (Ubuntu)**

apt (Debian Package Tool)

dpkg (Debian Package)

- **Các kho phần mềm cài đặt**

- Lưu trữ trên Server

- Tải về và cài đặt

- Hoặc cài trực tiếp qua mạng: #sudo ...

Tham khảo:

<https://quantrimang.com/trinh-quan-ly-goi-linux-pho-bien-nhat-163830>

APT advanced packet Tools

APT để quản lý gói tin ban đầu thay thế cho dpkg quản lý file.dep
Hiện nay APT quản lý cả RPM (redhat, Fedora, CentOS..); và
PCLinuxOSS hiện nay là file.apt

APT Command:

```
#apt-get [option] file
```

Option: install/ remove/ autoremove/ install -f

APT (GUI)

- **Gdebi**: Cài các gói tin lẻ
- **Synaptic**: GTK+ (cài đặt trên GNOME) file .deb/ .rpm
- **KPackage** : Cài đặt trên KDE
- **Cydia** : Cài trên Iphone OS
- **Wajig**:

Quản lý các kho phần mềm trên mạng

- # APT-DEB Repository
- # APT-RPM Repository
- # DPKG Installed Packages
- # Mirror Information
- # Red Carpet Channel
- # RPM Directory
- # RPM Header List
- # RPM MetaData (YUM)
- # RPM Installed Packages
- # Slackware Repository
- # Slackware Installed Packages
- # URPMI Repository

Tham khảo 1

- **Công cụ nâng cao gói (APT)**

Bạn có thể đã quen với apt-get, một lệnh trong đó sử dụng các công cụ đóng gói tiên tiến để tương tác với hệ thống gói của hệ điều hành. Các lệnh có liên quan và hữu ích nhất là (được chạy với quyền root):

apt-get install package-name(s) - Cài đặt gói phần mềm quy định, cùng với bất kỳ phụ thuộc

- **apt-get remove package-name(s)** - Loại bỏ gói phần mềm quy định, nhưng không loại bỏ sự phụ thuộc
- **apt-get autoremove** - Loại bỏ bất kỳ phụ thuộc.
- **apt-get clean** - Loại bỏ các tập tin gói đã tải về (.deb) cho các phần mềm đã được cài đặt
- **apt-get purge package-name(s)** - Kết hợp các chức năng của loại bỏ và làm sạch cho một gói cụ thể, cũng như các file cấu hình
- **apt-get update** - Đọc tập tin /etc/apt/sources.list và cập nhật dữ liệu của hệ thống về gói sẵn để cài đặt. Chạy lệnh này sau khi thay đổi tập sources.list.
- **apt-get upgrade** - nâng cấp tất cả các gói có bản cập nhật có sẵn. Chạy lệnh này sau khi chạy apt-get update.
- Trong khi **apt-get** cung cấp các chức năng thường dùng, lệnh **apt-cache** cung cấp thêm thông tin:
 - **apt-cache search package-name(s)** - Nếu bạn biết tên của một phần mềm nhưng apt-get install không thành công hoặc điểm trở đến phần mềm sai, điều này có vẻ tên đã bị thay đổi.
 - **apt-cache show package-name(s)** - Hiển thị thông tin phụ thuộc, số phiên bản và mô tả cơ bản của gói.
 - **apt-cache depends package-name(s)** - Liệt kê những gói cụ thể mà phụ thuộc vào gói chính. Đây là những gói sẽ được cài đặt với apt-get install.
 - **apt-cache rdepends package-name(s)** - kết quả đầu ra là một danh sách các gói mà phụ thuộc vào một gói cụ thể. Danh sách này thường xuyên có thể khá dài, vì vậy tốt nhất nên kết hợp thêm lệnh **less**.
 - **apt-cache pkgnames** - Tạo ra một danh sách các gói cài trên hệ thống của bạn. Danh sách này thường là khá dài, vì vậy tốt nhất nên kết hợp thêm lệnh **less**, hoặc đưa output vào một tập tin văn bản.

Tham khảo 2

- **Quản lý Package [Fedora](#) và [CentOS](#)**

Fedora và CentOS là các bản phân phối liên quan chặt chẽ, là upstream và downstream (tương ứng) của Red Hat Enterprise Linux. Sự khác biệt của chúng xuất phát từ cách gói được chọn để đưa vào kho.

Cả hai hệ thống sử dụng yum để tương tác với kho hệ thống và cài đặt phụ thuộc, và cũng bao gồm một công cụ cấp thấp hơn được gọi là rpm, cho phép bạn tương tác với từng gói.

Yellow Dog Updater, Modified (YUM)

Công cụ YUM được phát triển cho hệ thống Yellow Dog Linux như là một thay thế cho Yellow Dog Updater (yup). RedHat tìm thấy công cụ YUM là một bổ sung có giá trị cho hệ thống của họ. Ngày nay, YUM là gói mặc định và kho lưu trữ công cụ quản lý cho một số hệ điều hành.

Bạn có thể sử dụng các lệnh sau để tương tác với YUM:

yum install package-name(s) - Cài đặt các gói cùng với bất kỳ phụ thuộc yêu cầu

- **yum remove package-name(s)** - Loại bỏ các gói cụ thể từ hệ thống của bạn
- **yum search search-pattern** - Tìm kiếm danh sách các tên gói và mô tả cho các gói phù hợp với mô hình tìm kiếm và cung cấp một danh sách các tên gói, với kiến trúc và mô tả ngắn gọn về các nội dung gói.
- **yum deplist package-name(s)** - Danh sách tất cả các thư viện và các module mà gói có tên phụ thuộc vào, cùng với tên của các gói (bao gồm cả các phiên bản) cung cấp phụ thuộc.
- **yum check-update** - làm mới bộ nhớ cache cục bộ của cơ sở dữ liệu yum vì vậy thông tin phụ thuộc và các gói mới nhất luôn được cập nhật.
- **yum info package-name(s)** - Kết quả của lệnh info cung cấp tên, mô tả của gói, cũng như liên kết tới trang chủ cho phần mềm, phiên bản phát hành và kích thước cài đặt của phần mềm.
- **yum reinstall package-name(s)** - Xóa và sau đó tải về một bản sao mới của tập tin gói và cài lại đặt các phần mềm trên hệ thống của bạn
- **yum localinstall local-rpm-file** - Kiểm tra sự phụ thuộc của một file .rpm địa phương và sau đó cài đặt nó
- **yum update optional-package-name(s)** - Tải xuống và cài đặt tất cả các bản cập nhật bao gồm các bản vá lỗi, phiên bản bảo mật và nâng cấp, được cung cấp bởi các nhà phân phối của hệ thống điều hành của bạn.
- **yum upgrade** - Nâng cấp tất cả các gói được cài đặt trong hệ thống của bạn lên phiên bản mới nhất.

Tham khảo 3

#DNF / \$sudo dnf (Dandified Packaging Tool) – nâng cấp và mạnh hơn YUM , DNF chỉ chạy trên Fedora22 trở lên

\$sudo dnf check-update

sudo dnf upgrade

sudo dnf search <tên gói>/ sudo dnf search all <tên gói>

sudo dnf info <tên gói>

sudo dnf repoquery --requires package // file phụ thuộc

sudo dnf install <tên gói 1> <tên gói 2> ... // xác nhận Y

sudo dnf install -y <tên gói>

sudo dnf install <tên gói>.rpm / cài đặt từ Local giống #rpm -ivh file.rpm

sudo dnf erase <tên gói>

sudo dnf update //nâng cấp Fedora

man dnf // xem hướng dẫn dnf

Thêm kho phần mềm

sudo dnf install --nogpgcheck <http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-24.noarch.rpm>

Tham khảo 4

- **RPM Package Manager (RPM)**

Các lệnh sau đây nên được chạy dưới quyền user root. Một số lệnh phổ biến:

rpm --install --verbose --hash local-rpm-file-name.rpm or rpm -ivh filename.rpm

- Cài đặt gói rpm từ tập tin. rpm cũng có khả năng cài đặt các file RPM từ nguồn http và ftp.

- **rpm --erase package-name(s) hoặc rpm -e** - Loại bỏ các gói. Thông thường sẽ không hoàn thành nếu package-name phù hợp với nhiều hơn một gói.
- **rpm --query --all hoặc rpm -qa** - liệt kê tên của tất cả các gói cài đặt hiện tại.
- **rpm --query package-name(s) hoặc rpm -q** - cho phép bạn để xác nhận liệu một gói đã được cài đặt trong hệ thống của bạn chưa.
rpm --query --info package-name(s) hoặc rpm -qi - hiển thị các thông tin về một gói cài đặt
- **rpm --query --list package-name(s) hoặc rpm -ql** - tạo ra một danh sách các tập tin cài đặt của một gói phần mềm chỉ định. Điều này được bổ sung bởi:
- **rpm --query --file hoặc rpm -q qf filename** - kiểm tra để xem gói được cài đặt sở hữu các tập tin nào.
- Lưu ý RPM không tự động kiểm tra phụ thuộc, vì vậy bạn phải cài đặt chúng bằng tay. Để biết thêm thông tin xin vui lòng tham khảo những nguồn bên ngoài:

Tham khảo 5

- **Quản lý gói trong [Arch](#) Linux với Pacman**

Arch Linux sử dụng các gói nhị phân trong một định dạng .tar.xz, và cũng cung cấp một "cổng" xây dựng hệ thống tạo điều kiện cho xây dựng các gói.

Arch Linux chạy trên một lịch trình phát hành, có nghĩa là các gói được bổ sung vào kho lưu trữ chính khi các phụ thuộc của chúng đã sẵn sàng để cung cấp.

Do đó, quản trị viên của Arch Linux phải xem xét đầu ra của pacman cẩn thận trước khi quyết định nâng cấp hoặc cập nhật bất kỳ gói nào

Pacman: Công cụ pacman là rất mạnh mẽ, nhưng nó cũng rất đơn giản. Có ba lệnh lõi để quản lý gói cơ bản:

pacman --query package-name(s) hoặc pacman -Q - Tìm kiếm cơ sở dữ liệu gói cho một package name và số phiên bản.

- **pacman --sync package-name(s) hoặc pacman -S** - Cài đặt gói mới, download nội dung mới cho các cơ sở dữ liệu và/hoặc nâng cấp hệ thống, tùy thuộc vào các tùy chọn và tên gói hoặc các gói.

- **pacman --remove package-name(s) hoặc pacman -R** - Loại bỏ các gói hoặc các gói.

- Dưới đây là một số ví dụ với mô tả ngắn gọn:

pacman -Qi package-name(s) - Hiển thị thông tin về một gói nhất định, bao gồm cả thông tin phụ thuộc, ngày cài đặt của gói, một liên kết đến các nguồn upstream và các thông tin hữu ích khác.

pacman -Qo filename – Xuất ra số phiên bản và tên của gói đang sở hữu một tập tin chỉ định.

pacman -Qs package-name(s) - Tìm kiếm trong số các gói cài đặt phù hợp với package-name.

pacman -Qu – Danh sách các gói phiên bản cũ được cài đặt mà đang cần phải cập nhật

pacman -Sy - Triggers một database refresh, và đồng bộ hóa cơ sở dữ liệu địa phương với các cơ sở dữ liệu từ xa.

- **pacman -Su** - Kích hoạt một hệ thống cập nhật đầy đủ và download gói mới để nâng cấp hệ thống.

pacman -Sc - Xóa bỏ cài đặt các gói từ các bộ nhớ cache và cố gắng để làm sạch bản cũ của cơ sở dữ liệu lưu trữ.

pacman -S --ignore package-name(s) - Bỏ qua nâng cấp một gói chỉ định hoặc danh sách các gói

- **pacman -Rs** Loại bỏ một gói và phụ thuộc của nó, miễn là các phụ thuộc không cần thiết và không được cài đặt một cách rõ ràng bởi người sử dụng. Lệnh này là nghịch đảo của pacman -S.

pacman -Ru Loại bỏ các gói không cần thiết.