Fink: Software Libre en Mac OS X al estilo Debian

jgaliana@gmail.com

enero/2007 · Juan Galiana Lara

SOME RIGHTS RESERVED

Contenido

I. Introducción
II. Instalación
III. Uso de *apt-get*
IV. Uso del comando *fink*
V. Uso de *FinkComander*
VI. Referencias

I. Introducción

En primer lugar quiero dejar claro que esto no pretende ser ningún tipo de manual exhaustivo sobre fink ni nada por el estilo, más bien una pequeña guía para introducir al usuario en el uso de este programa y traer a su Mac el software libre ya que como me pasó a mi notaba que me faltaba algo en OS X hasta que descubrí *Fink*.
Lo de "Debian" en el título realmente es porque me centraré en el uso de *apt-get* y como veremos más adelante fink hace uso de las herramientas de Debian, aunque la filosofía de este sistema es muy similar a otros que se utilizan actualmente como por ejemplo los populares "*ports*"[1] de FreeBSD[2] (sería más correcto hablar de los ports de BSD en general).

El proyecto *Fink*, tal y como anuncian en su página web[3] trae el mundo del Software Libre[4] de Unix a Darwin[5] y Mac OS X[6].
Este proyecto modifica el código del software Unix de tal manera que pueda ser compilado y ejecutado en Mac OS X (lo porta). Utiliza las excelentes herramientas creadas por el proyecto Debian (*dpkg*, dselect y *apt-get*) para mantener el sistema de paquetes.
Estas herramientas permiten gestionar los programas que tenemos instalado en nuestro disco, podemos ver a *fink* como un motor de instalación de programas el cual permite realizar búsquedas, e instalar programas automáticamente a

través de Internet, ya que este sistema se encarga de conectarse a repositorios vía Internet, descargar el paquete, pasar por un proceso de configuración para por fin instalarse y estar disponible para la utilización, sin intervención alguna por parte del usuario.
Así podremos ampliar el surtido de herramientas en línea de comandos para nuestro OS X, además de una serie de aplicaciones gráficas desarrolladas para GNU[7]/Linux[8] y otras variantes de Unix, también podremos elegir si instalamos XFree86 u otra solución X11 (si es queremos alguna).
Nosotros elegimos que programas queremos instalar, sin miedo de estropear los programas de nuestro querido Mac OS X (ni viceversa) ya que fink mantiene todos los programas que instala en un sistema raíz diferente ubicado en /sw.
Podemos elegir entre bajar precompilados o bajar el código fuente directamente y compilarlo, todo al más puro estilo Debian[9] GNU/Linux.
Diré un par de cosas entre instalar desde binarios o desde código fuente:
Si elegimos instalar los paquetes precompilados, son paquetes que ya han sido compilados, con unas opciones generales para que puedan correr en cualquier equipo, como ventaja tenemos que la instalación es muy rápida ya que descargamos directamente el paquete binario e instalamos, sin embargo si decidimos instalar desde código fuente, se compilará el paquete para nuestro sistema por lo que estará más optimizado pero tardará más puesto que previamente pasará por el proceso de configuración de las opciones de compilación para seguidamente compilarse, lo cual llevará más o menos tiempo dependiendo de la máquina, este sistema evita al usuario de complicados procesos de configuración, pelearse con tedioso makefiles o resolver manualmente las dependencias de librerías que no tengamos en el sistema, también como pega si instalamos desde código nos harán falta más programas (compilador, librerías de desarrollo, etc).
Personalmente instalo prácticamente todo desde binarios, ya que es más rápido y pienso (solo es mi humilde opinión) que la optimización conseguida no es tanta por el tiempo que pierdes compilándolos, (aunque también hay sistemas como Gentoo[10] Linux en los cuales siguen la filosofía de compilar todo lo que se añada al sistema).

Tenemos varias maneras de instalar programas con esta herramienta, usando el comando *`fink`*, usando el comando *`apt-get`* o usando `FinkComander` (aplicación gráfica), después del proceso de instalación pasaremos a explicar la utilización de *`apt-get`* y más tarde añadiremos algunas notas sobre el comando *`fink`* y el programa gráfico *`FinkComander`*.

II. Instalación

Ahora vamos a ver como instalarlo, lo podemos descargar de:

http://fink.sourceforge.net/download/index.php?phpLang=es

Ahí elegimos el fichero adecuado dependiendo si disponemos de un PowerPC o un Intel y la versión de Mac OS X si usamos una anterior a Tiger.
En mi caso he probado en Tiger tanto en plataforma Intel como en PowerPC.

```
Fink 0.8.1 Binary Installer (PowerPC) - 17930 KB
Fink 0.8.1 Binary Installer (Intel) - 17510 KB
```

Descargamos e instalamos Fink, el proceso de instalación es como ya estamos acostumbrados así que no tendremos ningún problema.

Al final de la instalación nos pregunta si debe modificar los ficheros de configuración del usuario para que funcione correctamente, le decimos que si...
Si hubiera algún problema durante este proceso podemos teclear la orden

/sw/bin/pathsetup.sh

para corregirlo.

III. Uso de *apt-get*

Para usar el comando apt-get e instalar programas necesitamos privilegios de root (euid=0) algo que los usuarios de Debian estamos acostumbrados, si intentamos instalar algo desde otro usuario obtendremos un error como el siguiente:

```
E: Could not open lock file /sw/var/lib/dpkg/lock - open (13 Permission denied)
E: Unable to lock the administration directory (/sw/var/lib/dpkg/), are you root?
```

Tenemos dos posibilidades para resolver esto, una es utilizando el comando "*sudo*" desde una cuenta con rol Administrador (esto es importante ya que en OS X las cuentas con este rol pertenecen al grupo "admin" (gid 80), el cual tiene permisos para usar este comando, sino tendríamos que cambiar la configuración del fichero "*/etc/sudoers*").

O bien hacernos root (con la orden "*su*" o "*sudo su*"), si optamos por esta opción para que todo funcione correctamente deberemos insertar en el home del usuario root (normalmente en Mac OS X /var/root), en el fichero .bash_profile una línea, contando con que nuestra shell sea bash (por defecto), esto lo haremos con el siguiente comando:

```
echo "test -r /sw/bin/init.sh && . /sw/bin/init.sh" >> ~/.bash_profile
```

Este fichero es el que modifica la instalación de *Fink* cuando nos pide confirmación para configurar, pero solo lo hace para el usuario actual.
Una vez insertado este comando en la Terminal, podremos "recargar" esta nueva configuración tecleando simplemente "*su -*".

Así ya estaremos en situación de utilizar apt-get, tanto si hemos optado por anteponer sudo a las ordenes como si cambiamos a usuario root.
Al igual que en debian disponemos de un fichero "*sources.list*", en este caso está ubicado en "*/sw/etc/apt/source.list*", éste es el fichero de configuración de apt donde se especifica los repositorios a los que se conecta para descargar los paquetes.

Una vez todo listo, tecleamos la orden "*apt-get update*", con esto sincronizamos la base de datos de software local con la información actualizada de los repositorios que vienen en el sources.list.
Recordar que quien no esté como usuario root, se denota porque el prompt del sistema cambia y acaba con el símbolo "#" deberá anteponer sudo al comando apt-get con lo que un update quedaría de esta manera "*sudo apt-get update*".

Para buscar un paquete usamos "*apt-cache search <patrón>*" donde patrón son palabras clave sobre lo que queremos buscar.
Si queremos una descripción detallada de un paquete en concreto, utilizaremos "*apt-cache show <paquete>*", *apt-cache* puede utilizarlo cualquier usuario, ya que no instala nada, solo busca.

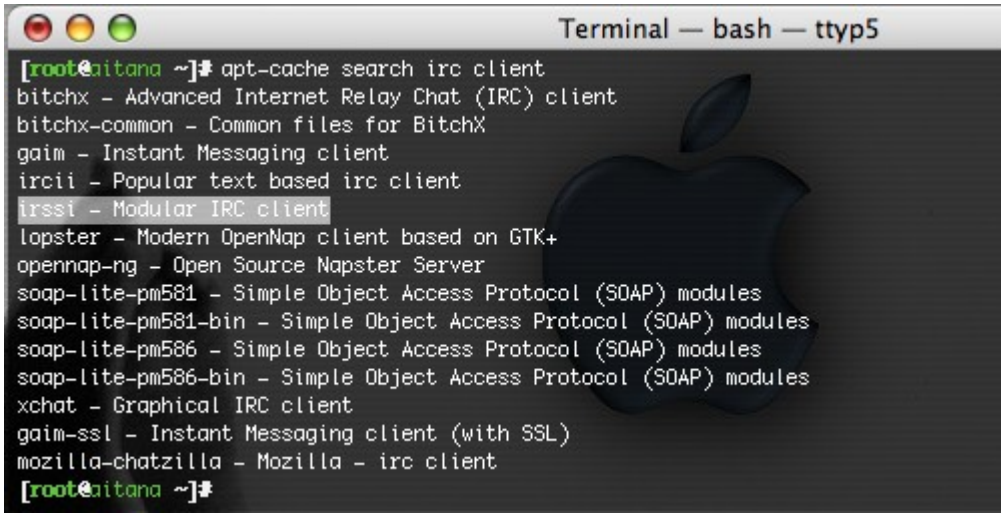
Para instalar un paquete en concreto, lo haremos mediante "*apt-get install <paquete>*"
Esta orden se encarga de resolver las dependencias que tenga este paquete, y bajar e instalar cada una de ellas.
Los "paquetes" de los que hablamos son ficheros en formato "deb", el formato de paquetes de Debian y distros basadas en ella (por ej Ubuntu), de la misma manera Red Hat y distros basadas en ella usan el formato de paquete "rpm".

Para borrar un paquete seria "*apt-get remove <paquete>*"
Si queremos actualizar todos los programas instalados vía apt-get basta con teclear "*apt-get upgrade*".

Los paquetes que vamos descargando se quedan grabados en una cache, ubicada en "*/sw/var/cache/apt/archives/*", podemos limpiarla con el comando "*apt-get clean*".
apt-get también permite instalar desde código, pero para eso tendríamos que añadir URI's de código (deb-src) en el sources.list y utilizar "*apt-get source...*", para instalar desde código usaremos el comando fink que veremos más adelante.

apt-get es un front-end para dpkg, lo que quiere decir, que apt-get se encarga de conectarse a Internet, descargar ficheros, resolver dependencias entre otras cosas pero "por debajo", internamente usa dpkg que es el encargado de añadir/quitar ficheros .deb, si bajamos un fichero deb suelto, por cualquier razón, deberemos instalarlo con "*dpkg -i <fichero.deb>*", al igual si queremos desinstalarlo sería "*dpkg -r <paquete>*.
También podemos usar dpkg para obtener información sobre qué software tenemos instalado en nuestro sistema y las versiones de cada uno de ellos, podremos ver una lista de todos los paquetes instalados con la orden "*dpkg -l*".
Otra opción que nos puede interesar es saber que ficheros ha instalado un paquete en nuestro sistema, podremos saberlo con "*dpkg -L paquete*".

Voy a poner un ejemplo de como instalar un cliente de irc en modo texto muy popular: *irssi*
Primero realizamos una búsqueda para encontrar el nombre del paquete, en nuestro caso si no sabemos el nombre exactamente, podemos probar por "*irc client*":



Procedemos a instalarlo:

```
Terminal — bash — ttyp5
[root@aitana ~]# apt-get clean
[root@aitana ~]# apt-get install irssi
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  glib2-shlibs
The following NEW packages will be installed:
  glib2-shlibs irssi
0 packages upgraded, 2 newly installed, 0 to remove and 21  not upgraded.
Need to get 2241kB of archives. After unpacking 8004kB will be used.
Do you want to continue? [Y/n]
Get:1 http://bindist.finkmirrors.net 10.4/release/main glib2-shlibs 2.6.6-1113 [399kB]
Get:2 http://bindist.finkmirrors.net 10.4/release/main irssi 0.8.9-1041 [1842kB]
Fetched 2241kB in 3m8s (11.9kB/s)
Selecting previously deselected package glib2-shlibs.
(Reading database ... 29259 files and directories currently installed.)
Unpacking glib2-shlibs (from .../glib2-shlibs_2.6.6-1113_darwin-powerpc.deb) ...
Selecting previously deselected package irssi.
Unpacking irssi (from .../irssi_0.8.9-1041_darwin-powerpc.deb) ...
Setting up glib2-shlibs (2.6.6-1113) ...
Setting up irssi (0.8.9-1041) ...

[root@aitana ~]#
```

Y ya tenemos disponible el comando "*irssi*" para ejectar y disfrutar.

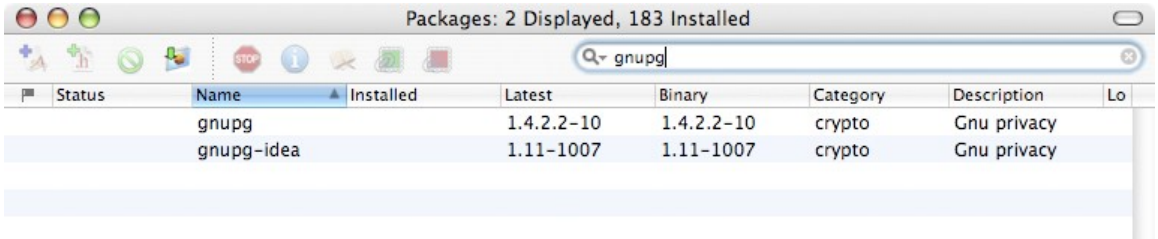
IV. Uso del comando *fink*

También podemos usar el comando "*fink*", el cual instala desde código fuente, para usar fink no necesitamos estar logeados como root o usar sudo, simplemente tener una cuenta con rol Administrador.
Para actualizar la cache local donde se realizan las búsquedas, usamos "*fink scanpackages*", y por ejemplo para instalar tecleamos "*fink install <paquete>*" o para borrarlo "fink remove paquete", si hacemos un "*man fink*" podremos consultar su página de manual donde viene especificadas detalladamente sus opciones de uso al completo.

V. Uso de *FinkComander*

También en la imagen que hemos bajado existe un directorio *FinkCommander*, en él se encuentra un programa del mismo nombre, este es un interfaz de usuario (GUI) para fink con el cual podemos usar toda su funcionalidad pero desde una aplicación gráfica.
Para instalarlo arrastramos FinkCommander a Aplicaciones, y una vez instalado (no funciona desde la imagen), lo ejecutamos.
Al igual que desde línea de comandos podemos instalar programas binarios (precompilados) o desde código fuente.
Para actualizar los binarios pinchamos sobre "*Binary -> Update Descriptions*", buscamos algún paquete, seleccionamos y pinchamos sobre "*Binary -> Install*", sin embargo para actualizar desde código pinchamos sobre "*Source -> Scanpackages*", y para instalarlo seria "*Source -> Install*", también podemos hacer lo mismo con los dos iconos de la parte superior izquierda, como se intuye el que pone "*+ A*" es para instalar el binario, y "*+ .h*" para instalar desde código fuente, internamente usa el comando "*apt-get install*" para instalar binarios y "*fink install*" para instalar desde código.

En estos dos capturas se puede apreciar como busca un programa (*gnupg*) y la salida que da cuando le damos a instalar del menú Binary.
Además de listar los paquetes, nos dice la versión que tenemos instalada en "Installed" (si lo tenemos instalado).



instalando...:

Packages: 2 Displayed, 188 Installed

gnupg

Status	Name	Installed	Latest	Binary	Category	Description
current	gnupg	1.4.2.2-10	1.4.2.2-10	1.4.2.2-10	crypto	Gnu privacy
	gnupg-idea		1.11-1007	1.11-1007	crypto	Gnu privacy

```
Reading Package Lists...
Building Dependency Tree...
The following extra packages will be installed:
  cyrus-sasl2-shlibs db44-aes-shlibs libusb-shlibs openldap-ssl-shlibs
The following NEW packages will be installed:
  cyrus-sasl2-shlibs db44-aes-shlibs gnupg libusb-shlibs openldap-ssl-shlibs
0 packages upgraded, 5 newly installed, 0 to remove and 16  not upgraded.
Need to get 4764kB of archives. After unpacking 13.2MB will be used.
Do you want to continue? [Y/n]
Get:1 http://bindist.finkmirrors.net 10.4/release/main libusb-shlibs 0.1.8-1014 [15.1kB]
Get:2 http://bindist.finkmirrors.net 10.4/release/crypto db44-aes-shlibs 4.4.16-1001 [786kB]
Get:3 http://bindist.finkmirrors.net 10.4/release/main cyrus-sasl2-shlibs 2.1.21-1014 [395kB]
Get:4 http://bindist.finkmirrors.net 10.4/release/crypto openldap-ssl-shlibs 2.1.22-1026 [482kB]
Get:5 http://bindist.finkmirrors.net 10.4/release/crypto gnupg 1.4.2.2-10 [3086kB]
Fetched 4764kB in 35s (133kB/s)
Selecting previously deselected package libusb-shlibs.
(Reading database ... 24747 files and directories currently installed.)
Unpacking libusb-shlibs (from .../libusb-shlibs_0.1.8-1014_darwin-powerpc.deb) ...
Selecting previously deselected package db44-aes-shlibs.
Unpacking db44-aes-shlibs (from .../db44-aes-shlibs_4.4.16-1001_darwin-powerpc.deb) ...
Selecting previously deselected package cyrus-sasl2-shlibs.
Unpacking cyrus-sasl2-shlibs (from .../cyrus-sasl2-shlibs_2.1.21-1014_darwin-powerpc.deb) ...
Selecting previously deselected package openldap-ssl-shlibs.
Unpacking openldap-ssl-shlibs (from .../openldap-ssl-shlibs_2.1.22-1026_darwin-powerpc.deb) ...
Selecting previously deselected package gnupg.
Unpacking gnupg (from .../gnupg_1.4.2.2-10_darwin-powerpc.deb) ...
Setting up libusb-shlibs (0.1.8-1014) ...
Setting up db44-aes-shlibs (4.4.16-1001) ...
Setting up cyrus-sasl2-shlibs (2.1.21-1014) ...
Setting up openldap-ssl-shlibs (2.1.22-1026) ...
Setting up gnupg (1.4.2.2-10) ...
* gpg: (gpg). GnuPG encryption and signing tool.
install-info(/sw/share/info/gpg.info): creating new section `GnuPG'
* gpgv: (gpgv). GnuPG signature verification tool.
```

Done

Lo que quiero dejar claro es que no importa que se instale desde *FinkComander* o desde línea de comandos, el software instalado se ubica en el mismo sitio, por ejemplo el programa que hemos instalado anteriormente en *FinkComander* podemos comprobar que esta instalado desde la consola con *dpkg*.

Terminal — bash — ttyp5

```
[root@titana ~]# dpkg -l|grep gnu
ii  gnupg          1.4.2.2-10     Gnu privacy guard - A Free PGP replacement
[root@titana ~]#
```

Think different.

VI. Referencias

Tenéis más información en la página del proyecto *Fink*, además tanto el programa como la página están en castellano.

[1] http://www.freebsd.org/ports/
[2] http://www.freebsd.org
[3] http://fink.sf.net
[4] http://www.opensource.org
[5] http://opensource.apple.com
[6] http://www.apple.com/macosx/
[7] http://www.gnu.org
[8] http://www.kernel.org
[9] http://www.debian.org
[10] http://www.gentoo.org