

FORMES D'INSTAL·LAR PROGRAMARI EN UBUNTU

En Ubuntu tenim moltes formes d'instal·lar programari. Evidentment, el més còmode és fer la instal·lació amb alguna eina gràfica com el **Centre de programari de l'Ubuntu** (software-center), el **Gestor de paquets Synaptic** (synaptic) o la eina **GDebi**, però en algunes ocasions aquestes eines no estan disponibles i hem de recórrer a altres que no disposen d'interfície gràfica.

Casuística:

Els paquets **.deb** (paquets Debian precompilats) que formen part dels dipòsits s'instal·len amb l'eina **apt**.

Els paquets **.deb** que no formen part dels dipòsits s'instal·len amb l'eina **dpkg**.

Els arxius **binaris** (.run o .bin) s'instal·len directament.

Els arxius **tarballs** (codi font comprimit amb extensions .tar, .tgz, .tar.gz o .tar.bz2) cal descomprimir-los i seguir les ordres del fabricant.

Els paquets **.rpm** (paquets Red Hat Package Manager precompilats) s'han de "debianitzar" amb eina alien abans d'instal·lar-los amb l'eina dpkg.

Eina apt

Quan un paquet forma part dels dipòsits podem fer servir l'eina apt. Aquest serà el cas més freqüent i còmode.

Sintaxi:

```
# apt-get install paquet1 paquet2 ...
```

Exemple:

```
# apt-get install chromium-browser
```

Apt té molts altres usos. Aquí teniu una llista dels més habituals.

Ordre	Ús
apt-get update	actualitza la llista de paquets
apt-get install paquet	instal·la el paquet i les seves dependències
apt-get upgrade	instal·la les noves versions dels paquets instal·lats

Ordre	Ús
apt-get remove paquet	desinstal·la el paquet sense esborrar els arxius de configuració
apt-get autoremove	desinstal·la paquets que no són necessaris
apt-get purge paquet	desinstal·la paquets i els seus arxius de configuració
apt-get clean	esborra tots els arxius del dipòsit local
apt-get autoclean	esborra els arxius obsolets del dipòsit local
apt-cache show paquet	mostra informació detallada del paquet
apt-cache search expressió	cerca paquets que coincideixin amb expressió

Podem editar l'arxiu de configuració dels dipòsits fent:

```
# nano /etc/apt/sources.list
```

Després d'haver fet canvis en l'arxiu de configuració és convenient actualitzar la llista de paquets:

```
# apt-get update
```

El dipòsit local es troba en el camí `/var/cache/apt/archives`

El sistema operatiu no permet fer servir dues eines d'instal·lació de paquets al mateix temps.

Eina dpkg

Quan descarreguem algun paquet `.deb` que no pertany als dipòsits hem d'instal·lar-lo amb l'eina `dpkg`. L'eina `dpkg` no instal·la les dependències del paquet com ho fan les eines `apt` o `aptitude`, per la qual cosa haurem d'assegurar-nos que quedaran satisfetes.

Exemple d'instal·lació de `webmin`:

Descarreguem el paquet:

```
$ wget http://downloads.sourceforge.net/project/webadmin/webmin/1.620/webmin_1.620_all.deb
```

Consultem la informació de les dependències:

```
# dpkg -I webmin_1.620_all.deb
new debian package, version 2.0.
size 21023860 bytes: control archive= 60835 bytes.
 211167 octets, 3784 línies changelog
   37 octets, 2 línies conffiles
  1624 octets, 16 línies control
  1810 octets, 38 línies copyright
  2292 octets, 89 línies * postinst          #!/bin/sh
```

```

    284 octets,      9 línies *  postrm          #!/bin/sh
    868 octets,     33 línies *  preinst         #!/bin/sh
    518 octets,     14 línies *  prerm          #!/bin/sh
Package: webmin
Version: 1.620
Section: admin
Priority: optional
Architecture: all
Depends: perl, libnet-ssleay-perl, openssl, libauthen-pam-perl, libpam-
runtime, libio-pty-perl, apt-show-versions, python
Pre-Depends: perl
Installed-Size: 130940
Maintainer: Jamie Cameron <jcameron@webmin.com>
Provides: webmin
Replaces: webmin-adsl, webmin-apache, webmin-bandwidth, webmin-bind, webmin-
burner, webmin-cfengine, webmin-cluster, webmin-core, webmin-cpan, webmin-
dhcpd, webmin-exim, webmin-exports, webmin-fetchmail, webmin-firewall,
webmin-freeswan, webmin-frox, webmin-fsdump, webmin-grub, webmin-heartbeat,
webmin-htaccess, webmin-inetd, webmin-jabber, webmin-ldap-netgroups, webmin-
ldap-user-simple, webmin-ldap-useradmin, webmin-lilo, webmin-logrotate,
webmin-lpadmin, webmin-lvm, webmin-mailboxes, webmin-mon, webmin-mysql,
webmin-nis, webmin-openslp, webmin-postfix, webmin-postgresql, webmin-ppp,
webmin-pptp-client, webmin-pptp-server, webmin-procmail, webmin-proftpd,
webmin-pserver, webmin-quota, webmin-samba, webmin-sarg, webmin-sendmail,
webmin-shorewall, webmin-slbackup, webmin-smart-status, webmin-snort, webmin-
software, webmin-spamassassin, webmin-squid, webmin-sshd, webmin-status,
webmin-stunnel, webmin-updown, webmin-usermin, webmin-vgetty, webmin-
webalizer, webmin-wuftpd, webmin-wvdial, webmin-xinetd
Description: web-based administration interface for Unix systems
  Using Webmin you can configure DNS, Samba, NFS, local/remote filesystems
  and more using your web browser. After installation, enter the URL
  https://localhost:10000/ into your browser and login as root with your
  root
  password.

```

La línia **Depends** ens informa dels paquets que són necessaris tenir instal·lats perquè quan el nostre paquet sigui instal·lat tot funcioni correctament, és a dir, les seves dependències.

Una manera còmoda de construir un arxiu que contingui aquesta línia és:

```
# dpkg -I webmin_1.620_all.deb | grep Depends > dependencies
```

El contingut de l'arxiu *dependencies* seria aquest:

```

Depends: perl, libnet-ssleay-perl, openssl, libauthen-pam-perl,
libpam-runtime, libio-pty-perl, apt-show-versions, python
Pre-Depends: perl

```

Ara haurem d'editar l'arxiu *dependencies* per tal de construir una ordre que instal·li aquests paquets:

```
# nano dependencies
```

I el deixarem d'aquesta forma:

```
apt-get install perl libnet-ssleay-perl openssl libauthen-pam-perl  
libpam-runtime libio-pty-perl apt-show-versions python
```

Donem permisos d'execució:

```
# chmod +x dependencies
```

Executem l'arxiu:

```
# ./dependencies
```

Ara estem en disposició de poder instal·lar el nostre paquet:

```
# dpkg -i webmin_1.620_all.deb
```

En cas de no haver resolt primer les dependències podríem resoldre aquesta situació amb l'eina apt fent:

```
# apt-get install -f
```

Arxius binaris

Quan ens trobem amb un paquet .run o .bin simplement l'hem d'executar. Si no té el permís d'execució l'hem d'afegir prèviament.

```
# chmod +x paquet.run  
# ./paquet.run
```

Arxius tarballs

Quan ens trobem amb un paquet .tar, .tgz, .tar.gz o .tar.bz2 haurem de descomprimir i seguir les ordres del fabricant.

```
# tar zxvf paquet.tar.gz  
# cd ./carpeta
```

Aleshores serà necessari seguir les instruccions que s'inclouen en el paquet. Aquestes es poden trobar habitualment en un fitxer anomenat README o INSTALL que trobareu un cop hàgiu descomprimit el fitxer tarball. Molt sovint s'inclou el codi font del programa en els fitxers tarball, el qual s'ha de compilar per a poder-lo utilitzar.

La compilació de programes requereix alguns paquets que no estan instal·lats

de manera predeterminada. La majoria d'aquests es poden instal·lar a través de la instal·lació del paquet build-essential.

```
# apt-get install build-essential
```

Paquets .rpm

Quan ens trobem que el paquet precompilat té l'extensió .rpm haurem de debianitzar-lo amb l'eina alien i posteriorment instal·lar-lo amb l'eina dpkg.

Exemple de debianització:

Descarreguem el paquet .rpm:

```
$ wget http://downloads.sourceforge.net/file-folder-ren/metamorphose2-0.8.0-9.fedora.noarch.rpm
```

Debianitzem amb l'eina alien:

```
# alien metamorphose2-0.8.0-9.fedora.noarch.rpm
```

Comprovem que s'ha generat l'arxiu .deb

```
# ls  
metamorphose2_0.8.0-10.1_all.deb metamorphose2-0.8.0-9.fedora.noarch.rpm
```

Ara estem en disposició de fer la instal·lació amb l'eina dpkg:

```
#dpkg -i metamorphose2_0.8.0-10.1_all.deb
```