

APT HOWTO (Obsolete Documentation)

Gustavo Noronha Silva <kov@debian.org>

Traduït per: Antoni Bella Perez <bella5ATtelelineDOTes>

Revisat per: Orestes Mas <orestesATtsc.upcDOTes>

1.8.5 - Juliol de 2003

Resum

Aquest document pretén proveir a l'usuari d'una bona comprensió sobre la funcionalitat de l'eina per a gestionar els paquets de Debian, APT. El seu objectiu és el de facilitar la vida als nous usuaris de Debian i el de servir d'ajuda als que volen tenir un coneixement més profund sobre l'administració d'aquest sistema. Fou creada per al projecte Debian per ajudar a millorar el suport disponible per als usuaris d'aquesta distribució.

Nota de Copyright

Copyright © 2001, 2002, 2003, 2004 Gustavo Noronha Silva

This manual is free software; you may redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

This is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

A copy of the GNU General Public License is available as `/usr/share/common-licenses/GPL` in the Debian GNU/Linux distribution or on the World Wide Web at the GNU General Public Licence. You can also obtain it by writing to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Índex

1	Introducció	1
2	Configuració bàsica	3
2.1	El fitxer <code>/etc/apt/sources.list</code>	3
2.2	Com s'usa APT localment	4
2.3	Decidir quina rèplica és la que s'inclourà al fitxer <code>sources.list</code> : <code>netselect</code> , <code>netselect-apt</code>	5
2.4	Afegir un CD-ROM al fitxer <code>sources.list</code>	6
3	Tracte amb els paquets	9
3.1	Actualització del llistat de paquets disponibles	9
3.2	Instal·lació de paquets	9
3.3	Eliminació de paquets	11
3.4	Actualització de paquets	12
3.5	Actualització a una nova versió	13
3.6	Eliminació dels fitxers de paquet no usats: <code>apt-get clean</code> i <code>autoclean</code>	15
3.7	L'ús d'APT amb <code>dselect</code>	16
3.8	Com mantenir un sistema híbrid	17
3.9	Com actualitzar paquets des de versions específiques de Debian	18
3.10	Com mantenir versions específiques de paquets instal·lats (complex)	18
4	Ajudants de molta utilitat	21
4.1	Com instal·lar paquets instal·lats localment: <code>equivs</code>	21
4.2	Eliminació de fitxers locale sense usar: <code>localepurge</code>	23
4.3	Com conèixer quins paquets poden ser actualitzats	23

5	Obtenir informació quant als paquets.	25
5.1	Descobrir els noms dels paquets	25
5.2	L'ús de dpkg per a cercar noms de paquet	27
5.3	Com instal·lar paquets "sota demanda"	28
5.4	Com esbrinar a quin paquet pertany un fitxer	29
5.5	Com mantindràs informat quant als canvis en els paquets.	30
6	Treballar amb els paquets font	31
6.1	Descàrrega de paquets font	31
6.2	Paquets necessaris per a compilar un paquet font	32
7	Com batre's amb els errors	33
7.1	Errorrs comuns	33
7.2	A on puc trobar ajuda?	34
8	Quines distribucions tenen suport per a APT?	35
9	Crèdits	37
10	Noves versions d'aquest tutorial	39

Capítol 1

Introducció

Al principi hi havien els `.tar.gz`. Els usuaris havien de compilar al seu sistema GNU/Linux cada programa que volien usar. Quan es va crear Debian, es jutjava com a necessari el fet que el sistema havia d'incloure un mètode de gestió per a l'instal·lació de paquets a la màquina. A aquest sistema se l'anomenà `dpkg`. Així fou com nasqué el primer "paquet" en el món GNU/Linux, fins i tot abans de que Red Hat decidís crear el seu propi sistema "rpm".

Ràpidament s'arribà a un nou dilema en les ments dels creadors de GNU/Linux. Ells necessitaven un mode ràpid, pràctic i eficient per a instal·lar paquets de manera que es gestionessin automàticament les dependències i que es fes càrrec dels fitxers de configuració mentre s'actualitzava. Altra vegada, Debian en fou el precursor i creà a APT, "Advanced Packaging Tool" (eina avançada d'empaquetament), la qual ha estat portada per Conectiva per a usar-la amb rpm i posteriorment ha estat adoptada per algunes altres distribucions.

Aquest manual no cobreix el `apt-rpm`, el nom amb el que és coneix al port APT de Conectiva, però "pedaços" a aquest document serien benvinguts.

Aquest manual està basat en la següent distribució de Debian, *Sarge*.

Capítol 2

Configuració bàsica

2.1 El fitxer `/etc/apt/sources.list`

Com a part del seu funcionament, APT usa un fitxer que llista les “fonts” des d'on podrà obtenir els paquets. Aquest fitxer és `/etc/apt/sources.list`.

Les entrades en aquest fitxer normalment segueixen aquest format:

```
deb http://lloc.http.org/debian distribució secció_1 secció2 secció_3
deb-src http://lloc.http.org/debian distribució secció_1 secció_2 secció_3
```

Per descomptat, les anteriors entrades són fictícies i no s'han d'usar. La primera paraula a cada línia, `deb` o `deb-src`, indica el tipus d'arxiu cap al que apunta: tan si conté paquets binaris (`deb`), aquests són, els paquets pre-compilats que s'usen normalment, o, els paquets font (`deb-src`), que són el codi font del programa original a més del fitxer de control de Debian (`.dsc`) junt amb el fitxer `diff.gz`, el qual conté els canvis necessaris per a “debianitzar” el programa.

Usualment trobarem el següent al `sources.list` per defecte de Debian:

```
# See sources.list(5) for more information, especialy
# Remember that you can only use http, ftp or file URIs
# CDROMs are managed through the apt-cdrom tool.
deb http://http.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://security.debian.org stable/updates main contrib non-free

# Uncomment if you want the apt-get source function to work
#deb-src http://http.us.debian.org/debian stable main contrib non-free
#deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib no
```

Aquestes són les línies necessàries per a una instal·lació bàsica de Debian. La primera línia `deb` apunta cap a l'arxiu oficial, la segona cap a l'arxiu non-US i la tercera cap a l'arxiu d'actualitzacions de seguretat de Debian.

Les dues últimes línies estan comentades (amb un caràcter “#” al davant), de manera que `apt-get` les ignorarà. Aquestes són línies `deb-src`, el qual ens indica que apunten cap al codi font de Debian dels programes. Si tot sovint descarregueu paquets font per a provar-los o recompilar-los, llavors descomenteu-les.

El fitxer `/etc/apt/sources.list` pot contenir diversos tipus de línies. Pel que un servidor coneix, APT sap com negociar amb arxius dels protocols `http`, `ftp`, `file` (fitxers locals, p.ex., un directori que continga un sistema de fitxers ISO9660 muntat) i `ssh`.

No us oblideu d'executar `apt-get update` després de modificar el fitxer `/etc/apt/sources.list`. Haureu de fer-ho per a obtenir els llistats de paquets des de les fonts que heu especificat.

2.2 Com s'usa APT localment

A vegades tindreu uns quants paquets `.deb` amb els quals us agradaria usar APT per a instal·lar-los de manera que les dependències siguin resoltes automàticament.

Per a fer-ho, creeu un directori i ubiqueu-hi els fitxers `.deb` que hi voleu indexar. Per exemple:

```
# mkdir /root/debs
```

Podreu modificar directament les especificacions establertes en el fitxer de control dels paquets per al vostre repositori usant un fitxer `override`. Dintre d'aquest fitxer es poden definir algunes opcions per a solapar a les que venen amb el paquet. De la manera que segueix:

```
paquet prioritat secció
```

«paquet» és el nom del paquet en si, la «prioritat» pot ser baixa, mitjana o alta i «secció» és la secció a la qual pertany el paquet. El nom del fitxer no importa, tan sols haureu de passar-lo com argument a `dpkg-scanpackages`. Si no desitgeu escriure un fitxer `override`, llavors, simplement useu `/dev/null` quan crideu a `dpkg-scanpackages`.

Estan encara al directori `/root` feu el següent:

```
# dpkg-scanpackages debs fitxer | gzip > debs/Packages.gz
```

A l'anterior línia, *fitxer* és el fitxer `override`, dit comandament genera un fitxer `Packages.gz` que conté informació diversa quant als paquets, la qual serà usada per APT. Finalment, per a usar aquests paquets afegiu:


```
deb file:/root debs/
```

Després d'això, useu els comandaments usuals d'APT. També podríeu generar un repositori de fonts. Per a fer-ho useu el mateix procediment, però recordeu que necessitareu tenir en el directori els fitxers `.orig.tar.gz`, `.dsc` i `.diff.gz`, i que usareu `Sources.gz` en comptes de `Packages.gz`. El programa emprat també és un altre. Aquest serà `dpkg-scansources`. La línia de comandaments es veuria així:

```
# dpkg-scansources debs | gzip > debs/Sources.gz
```

Observeu que `dpkg-scansources` no necessita un fitxer `override`. La línia per al `sources.list` seria:

```
deb-src file:/root debs/
```

2.3 Decidir quina rèplica és la que s'inclourà al fitxer `sources.list`: `netselect`, `netselect-apt`

Un dubte molt freqüent, principalment entre els usuaris més novells és: "Quina rèplica de Debian incloc al `sources.list`?". Hi ha moltes maneres de decidir-ho. Els experts probablement tindran un script que determinarà mitjançant ping quina rèplica és més ràpida. Però hi ha un programa que fa això per nosaltres: **netselect**.

Per a instal·lar `netselect`, com de costum:

```
# apt-get install netselect
```

Executar-lo sense paràmetres us mostrarà l'ajuda. Executar-lo amb una llista de màquines (rèpliques) separades per un espai, us retornarà un llistat amb la puntuació obtinguda per cadascuna de les màquines. Aquesta puntuació serà tinguda en compte segons el temps estimat de ping i el nombre de salts (el nombre de màquines per les que es travessa per arribar a destinació), el resultat és inversament proporcional al de la velocitat de descàrrega estimada (a més baix, millor). La màquina retornada és la que ha obtingut una puntuació menor (per a visualitzar la llista completa podeu afegir l'opció `-vv`). Mireu aquest exemple:

```
# netselect ftp.debian.org http.us.debian.org ftp.at.debian.org download.unes  
365 ftp.debian.org.br  
#
```

Això significa que de les rèpliques incloses com a paràmetres a `netselect`, `ftp.debian.org.br` ha estat la millor, amb una puntuació de 365. (Atenció!! Aquesta prova ha estat feta sobre el meu ordinador i la ubicació geogràfica de la xarxa és

extremadament diferent depenent del punt de connexió, aquest valor no és necessàriament la velocitat correcta en d'altres ordinadors).

Ara, només haureu d'afegir la rèplica més ràpida que hagi trobat netselect al fitxer `/etc/apt/sources.list` (consultar 'El fitxer `/etc/apt/sources.list`' a la pàgina 3) i seguir els consells en 'Tracte amb els paquets' a la pàgina 9.

Nota: La llista de les rèpliques es pot trobar en el fitxer http://www.debian.org/mirror/mirrors_full.

Començat amb la versió 0.3, el paquet netselect inclou l'script **netselect-apt**, el qual automatitza l'anterior procés. Sols cal introduir l'arbre de la distribució com a paràmetre (per defecte és `stable`) i el fitxer `sources.list` serà generat amb les millors rèpliques de main i non-US, i aquest serà desat al directori actual. El següent exemple genera un `sources.list` de la distribució `stable`:

```
# ls sources.list
ls: sources.list: File or directory not found
# netselect-apt stable
(...)
# ls -l sources.list
sources.list
#
```

Recordar: El fitxer `sources.list` és generat sota el directori actual, i l'haureu de `more` cap al directori `/etc/apt`.

Després, seguiu els consells en 'Tracte amb els paquets' a la pàgina 9.

2.4 Afegir un CD-ROM al fitxer `sources.list`

Si preferiu usar el vostre CD-ROM per actualitzar o instal·lar paquets al vostre sistema automàticament usant APT, podeu afegir-lo al vostre `sources.list`. Per a fer-ho, podeu usar el programa `apt-cdrom` d'aquesta manera:

```
# apt-cdrom add
```

amb el CD-ROM de Debian a la unitat lectora. Això muntarà el CD-ROM i si és un CD vàlid, Debian cercarà la informació de paquets al disc. Si la configuració del vostre CD-ROM és un xic inusual, també podeu usar les següents opcions:

```
-h           - Ajuda del programa
-d directori - Punt de muntatge del CD-ROM
-r           - Renomena a un CD-ROM reconegut
-m           - No muntar
-f           - Mode ràpid, no comprova els fitxers d'índex
-a           - Mode d'exploració acurada
```

Per exemple:

```
# apt-cdrom -d /home/kov/el_meu_cdrom add
```

També podeu identificar un CD-ROM sense afegir-lo a la vostra llista:

```
# apt-cdrom ident
```

Tingueu en compte que aquest programa tan sols funcionarà si el vostre CD-ROM està adequadament configurat al vostre sistema `/etc/fstab`.

Capítol 3

Tracte amb els paquets

3.1 Actualització del llistat de paquets disponibles

El sistema d'empaquetament usa una base de dades per a seguir quins paquets hi ha instal·lats, quins no i quins estan disponibles. El programa `apt-get` usa aquesta base de dades per a saber com instal·lar els paquets requerits pel usuari i obtenir la informació necessària dels paquets addicionals que seran necessaris i el seu ordre d'instal·lació per a que un paquet funcioni correctament.

Per actualitzar aquest llistat haureu d'usar el comandament `apt-get update`. Aquest comandament cercarà els llistats de paquets en els arxius indicats a `/etc/apt/sources.list`; per a més informació quant a aquest fitxer consulteu 'El fitxer `/etc/apt/sources.list`' a la pàgina [3](#).

És una bona idea executar aquest comandament de forma regular per a mantindre-us informats sobre de possibles actualitzacions, en particular les actualitzacions de seguretat.

3.2 Instal·lació de paquets

Finalment, el procés que estàveu esperant! Amb el vostre fitxer `sources.list` llest i el llistat de paquets disponibles al dia, tot el que haureu de fer és executar `apt-get` per a obtenir el paquet que desitgeu instal·lar. Per exemple, podeu executar:

```
# apt-get install xchat
```

APT cercarà a la seva base de dades per a la versió més recent de dit paquet i el descarregarà des de l'arxiu especificat al `sources.list`. En el cas que aquest paquet depengui de cap altre -com en aquest cas- APT en resoldrà les dependències i instal·larà els paquets necessaris. Vegeu aquest exemple:

```
# apt-get install nautilus
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 17.2MB will be used.
Do you want to continue? [Y/n]
```

El paquet `nautilus` depèn de les biblioteques compartides citades, així doncs, APT les obtindrà des de l'arxiu. Si prèviament especifiqueu el nom d'aquestes biblioteques a la línia de comandaments de `apt-get`, APT no us preguntarà si desitgeu continuar; donarà automàticament per fet que accepteu la instal·lació d'aquests paquets.

Això vol dir que APT tan sols demana confirmació si necessita instal·lar paquets que no li heu especificat a la línia de comandaments.

Les següents opcions per a `apt-get` poden ser d'utilitat:

```
-h Aquest text d'ajuda.
-d Sols descàrrega - NO instal·la o desempaqueta els arxius
-f Intenta continuar si falla la comprovació d'integritat
-s No acta. Simula l'execució del procés ordenat
-y Assumeix Si a totes les preguntes
-u Mostra una llista dels paquets a actualitzar
```

Es poden seleccionar múltiples paquets per a instal·lar en una sola línia. Els fitxers descarregats des de la xarxa seran ubicats en el directori `/var/cache/apt/archives` per a la seva posterior instal·lació.

També podeu especificar paquets a eliminar a la mateixa línia de comandaments. Tan sols heu de posar un guionet `-` immediatament després del nom del paquet a eliminar, així:

```
# apt-get install nautilus gnome-panel-
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0
The following packages will be REMOVED:
  gnome-applets gnome-panel gnome-panel-data gnome-session
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 4 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 2594kB will be used.
Do you want to continue? [Y/n]
```

Per a més detalls sobre la eliminació d'un paquet consulteu la secció 'Eliminació de paquets' en aquesta pàgina.

Si d'alguna manera malmeteu cap dels paquets instal·lats o simplement desitgeu reinstal·lar els fitxers d'un paquet amb la versió més nova que hi hagi disponible, podeu usar l'opció `--reinstall` tal i com es mostra:

```
# apt-get --reinstall install gdm
Reading Package Lists... Done
Building Dependency Tree... Done
0 packages upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 1 not
Need to get 0B/182kB of archives. After unpacking 0B will be used.
Do you want to continue? [Y/n]
```

3.3 Eliminació de paquets

Si ja no requeriu més l'ús d'un paquet, podeu eliminar-lo del vostre sistema usant APT. Per a fer-ho simplement escriviu: `apt-get remove paquet`. Per exemple:

```
# apt-get remove gnome-panel
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  gnome-applets gnome-panel gnome-panel-data gnome-session
0 packages upgraded, 0 newly installed, 4 to remove and 1 not upgraded.
Need to get 0B of archives. After unpacking 14.6MB will be freed.
Do you want to continue? [Y/n]
```

Tal i com podreu apreciar a l'anterior exemple, APT també es fa càrrec d'eliminar paquets que en depenen preguntant-vos si els voleu eliminar. No hi ha cap manera d'eliminar un paquet usant APT sense també eliminar els paquets que en depenen.

Executant `apt-get` de la manera mostrada causarà que s'eliminin tots els paquets, però els seus fitxers de configuració, si n'hi ha cap, romandran intactes en el sistema. Per a una eliminació completa del paquet, executeu:

```
# apt-get --purge remove gnome-panel
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  gnome-applets* gnome-panel* gnome-panel-data* gnome-session*
0 packages upgraded, 0 newly installed, 4 to remove and 1 not upgraded.
Need to get 0B of archives. After unpacking 14.6MB will be freed.
Do you want to continue? [Y/n]
```

Tingueu en compte que l'asterisc "*" després dels noms indica que els fitxers de configuració per a cadascun d'aquests també seran eliminats.

A l'igual que en el cas del mètode `install`, podeu usar un símbol amb `remove` per a invertir la mesura per a un paquet en particular, si afegiu un caràcter "+" a la dreta després del nom del paquet, aquest serà instal·lat en comptes d'eliminar.

```
# apt-get --purge remove gnome-panel nautilus+
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
The following packages will be REMOVED:
  gnome-applets* gnome-panel* gnome-panel-data* gnome-session*
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 4 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 2594kB will be used.
Do you want to continue? [Y/n]
```

Observeu que `apt-get` llista els paquets addicionals que s'hauran d'instal·lar (és a dir, els paquets que necessita la instal·lació del paquet que heu sol·licitat), els que s'eliminaran i els que s'instal·laran (incloguen altra vegada als paquets extra).

3.4 Actualització de paquets

Les actualitzacions de paquets són una de les millors aportacions del sistema APT. Es poden realitzar des d'una única línia de comandaments: `apt-get upgrade`. Podeu usar aquest comandament per actualitzar els paquets de l'actual distribució, o bé, per actualitzar a una de nova, millor usar el comandament `apt-get dist-upgrade`; per a més detalls consulteu la secció 'Actualització a una nova versió' a la pàgina següent.

El seu ús és de molta utilitat junt amb l'opció `-u`. Aquesta opció causa que APT mostri la llista completa de paquets a actualitzar. Sense, estareu actualitzant a cegues. APT descarregarà les últimes versions de cada paquet i llavors els instal·larà en l'ordre adequat. Tot i que cal ressaltar que és molt important executar prèviament el comandament `apt-get update`. Consulteu la secció 'Actualització del llistat de paquets disponibles' a la pàgina 9. Observeu aquest exemple:

```
# apt-get -u upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages have been kept back
  cpp gcc lilo
The following packages will be upgraded
```



```

adduser ae apt autoconf debhelper dpkg-dev esound esound-common ftp indent
ipchains isapnptools libaudiofile-dev libaudiofile0 libesd0 libesd0-dev
libgtk1.2 libgtk1.2-dev liblockfile1 libnewt0 liborbit-dev liborbit0
libstdc++2.10-glibc2.2 libtiff3g libtiff3g-dev modconf orbit procps psmisc
29 packages upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 5055B/5055kB of archives. After unpacking 1161kB will be used.
Do you want to continue? [Y/n]

```

El procés és molt simple. Tingueu en compte que en les primeres línies `apt-get` indica alguns paquets que seran mantinguts. Això significa que hi ha una versió més nova però que no seran actualitzats per algun motiu. Possiblement siga per culpa a dependències trencades (un paquet del qual depenen no disposa d'una versió més nova per actualitzar) o dependències noves (el paquet ara depèn de nous paquets que no pas la versió anterior).

No hi ha una solució clara per al primer cas. Per al segon, n'hi ha prou amb executar `apt-get install` per al paquet en qüestió, de manera que és descarregaran les dependències. Una solució encara millor és la d'usar `dist-upgrade`. Consulteu la secció 'Actualització a una nova versió' en aquesta pàgina.

3.5 Actualització a una nova versió

Aquesta característica d'APT us permet actualitzar tot un sistema Debian d'una sola vegada a través de l'Internet o des d'un CD (comprat o descarregat com a imatge ISO).

també s'usa quan es fan canvis entre les relacions dels paquets instal·lats. Amb `apt-get upgrade`, aquests paquets serien omesos (mantinguts).

Per exemple, suposem que esteu usant la revisió 0 de la versió stable de Debian i que compreu un CD amb la revisió 3. En aquest punt podeu usar APT per actualitzar el vostre sistema des del nou CD. Per a fer-ho usareu `apt-cdrom` (consultar la secció 'Afegir un CD-ROM al fitxer `sources.list`' a la pàgina 6) per afegir el CD al vostre `/etc/apt/sources.list` i executareu un `apt-get dist-upgrade`.

És important fer menció que APT sempre cercarà la versió més recent dels paquets. Així doncs, si el vostre `/etc/apt/sources.list` tingués llistat un arxiu amb una versió més recent que la del CD, APT descarregaria el paquet des d'allí.

A l'exemple mostrat a la secció 'Actualització de paquets' a la pàgina anterior, observem que alguns paquets són mantinguts. Resoldrem aquest problema amb el mètode `dist-upgrade`:

```

# apt-get -u dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Done
The following NEW packages will be installed:
  cpp-2.95 cron exim gcc-2.95 libident libopenldap-runtime libopenldap1

```

```

libpcre2 logrotate mailx
The following packages have been kept back
lilo
The following packages will be upgraded
adduser ae apt autoconf cpp debhelper dpkg-dev esound esound-common ftp gcc
indent ipchains isapnptools libaudiofile-dev libaudiofile0 libesd0
libesd0-dev libgtk1.2 libgtk1.2-dev liblockfile1 libnewt0 liborbit-dev
liborbit0 libstdc++2.10-glibc2.2 libtiff3g libtiff3g-dev modconf orbit
procps psmisc
31 packages upgraded, 10 newly installed, 0 to remove and 1 not upgraded.
Need to get 0B/7098kB of archives. After unpacking 3118kB will be used.
Do you want to continue? [Y/n]
```

Observeu que ara els paquets s'actualitzaran, i que hi ha nous paquets que també seran instal·lats (les noves dependències del paquets). Tingueu també en compte que lilo seguirà mantingut. Probablement tinga un problema més seriós que una simple nova dependència. Això ho podrem esbrinar si executem:

```

# apt-get -u install lilo
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
cron debconf exim libident libopenldap-runtime libopenldap1 libpcre2
logrotate mailx
The following packages will be REMOVED:
debconf-tiny
The following NEW packages will be installed:
cron debconf exim libident libopenldap-runtime libopenldap1 libpcre2
logrotate mailx
The following packages will be upgraded
lilo
1 packages upgraded, 9 newly installed, 1 to remove and 31 not upgraded.
Need to get 225kB/1179kB of archives. After unpacking 2659kB will be used.
Do you want to continue? [Y/n]
```

Tal i com es veu a sobre, lilo té un nou conflicte amb el paquet `debconf-tiny`, el qual vo dir que no podrà ser instal·lat (o actualitzat) sense eliminar `debconf-tiny`.

Per a saber què manté o elimina un paquet podeu usar:

```

# apt-get -o Debug::pkgProblemResolver=yes dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Starting
Starting 2
```

```

Investigating python1.5
Package python1.5 has broken dep on python1.5-base
  Considering python1.5-base 0 as a solution to python1.5 0
  Holding Back python1.5 rather than change python1.5-base
Investigating python1.5-dev
Package python1.5-dev has broken dep on python1.5
  Considering python1.5 0 as a solution to python1.5-dev 0
  Holding Back python1.5-dev rather than change python1.5
  Try to Re-Instate python1.5-dev
Done
Done
The following packages have been kept back
  gs python1.5-dev
0 packages upgraded, 0 newly installed, 0 to remove and 2 not upgraded.

```

D'aquesta manera és fàcil advertir que el paquet `python1.5-dev` no pot ser instal·lat degut a una dependència insatisfeta: `python1.5`.

3.6 Eliminació dels fitxers de paquet no usats: `apt-get clean` i `autoclean`

Quan instal·leu un paquet, APT descarregarà els fitxers necessaris des de les màquines llistades al `/etc/apt/sources.list`, llavors els desa en un repositori local (`/var/cache/apt/archives/`), i des d'allí procedeix a la seva instal·lació, consulteu 'Instal·lació de paquets' a la pàgina 9.

Amb el passar del temps, dit repositori local pot créixer considerablement i ocupar una preuada quantitat d'espai en disc. Afortunadament, APT proveeix d'eines per a gestionar el seu repositori local: els mètodes `apt-get clean` i `autoclean`.

`apt-get clean` ho elimina tot a excepció dels fitxers de blocat de `/var/cache/apt/archives/` i `/var/cache/apt/archives/partial/`. D'aquesta manera, si necessiteu reinstal·lar un paquet APT haurà de descarregar-lo una altra vegada.

`apt-get autoclean` tan sols elimina els fitxers de paquet que no poden ser descarregats de nou.

El següent exemple mostra com funciona la neteja automàtica d'`apt-get`:

```

# ls /var/cache/apt/archives/logrotate* /var/cache/apt/archives/gpm*
logrotate_3.5.9-7_i386.deb
logrotate_3.5.9-8_i386.deb
gpm_1.19.6-11_i386.deb

```

A `/var/cache/apt/archives` hi ha dos fitxers per al paquet `logrotate` i un per al paquet `gpm`.

```
# apt-show-versions -p logrotate
logrotate/stable uptodate 3.5.9-8
# apt-show-versions -p gpm
gpm/stable upgradeable from 1.19.6-11 to 1.19.6-12
```

`apt-show-versions` mostra que `logrotate_3.5.9-8_i386.deb` proveeix la versió d'actualització de `logrotate`, de manera que `logrotate_3.5.9-7_i386.deb` ja no és necessari. També ens indica que `gpm_1.19.6-11_i386.deb` tampoc no és necessari atès a que ja es pot descarregar una versió més recent del paquet.

```
# apt-get autoclean
Reading Package Lists... Done
Building Dependency Tree... Done
Del gpm 1.19.6-11 [145kB]
Del logrotate 3.5.9-7 [26.5kB]
```

Finalment, `apt-get autoclean` tan sols elimina els fitxers antics. Per a més informació quant a `apt-show` consulteu 'Com actualitzar paquets des de versions específiques de Debian' a la pàgina 18.

3.7 L'ús d'APT amb dselect

`dselect` és un programa d'ajuda a l'usuari per a la selecció de paquets Debian per a l'instal·lació. És considerat com un mica complicat i un xic avorrit, però amb la pràctica podreu usar la seva interfície de ncurses basada en consola.

Una de les característiques de `dselect` és que coneix com usar la capacitat dels paquets Debian per a "recomanar" i "suggerir" l'instal·lació d'altres paquets. Per a usar el programa, executeu "`dselect`" com a root. Escolliu "`apt`" com al vostre mètode d'accés. Això últim no és realment necessari, però si no esteu usant un CD-ROM i desitgeu descarregar els paquets des de l'Internet, la millor mode és usant `dselect`.

Per a entendre millor el seu ús, llegiu la seva documentació que es troba en la pàgina de Debian <http://www.debian.org/doc/ddp>.

Després de seleccionar els paquets amb `dselect`, useu:

```
# apt-get -u dselect-upgrade
```

com al següent exemple:

```
# apt-get -u dselect-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
```

```

The following packages will be REMOVED:
  lbxproxy
The following NEW packages will be installed:
  bonobo console-tools-libs cpp-3.0 enscript expat fingerd gcc-3.0
  gcc-3.0-base icepref klogd libdigest-md5-perl libfnlib0 libft-perl
  libgc5-dev libgcc300 libhtml-clean-perl libltdl0-dev libsasl-modules
  libstdc++3.0 metamail nethack proftpd-doc psfontmgr python-newt talk tidy
  util-linux-locales vacation xbill xplanet-images
The following packages will be upgraded:
  debian-policy
1 packages upgraded, 30 newly installed, 1 to remove and 0 not upgraded.
Need to get 7140kB of archives. After unpacking 16.3MB will be used.
Do you want to continue? [Y/n]

```

Compareu-ho amb el que es veu en executar «`apt-get dist-upgrade`» en el mateix sistema:

```

# apt-get -u dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Done
The following packages will be upgraded:
  debian-policy
1 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 421kB of archives. After unpacking 25.6kB will be freed.
Do you want to continue? [Y/n]

```

Observeu que la majoria de paquets seran instal·lats per causa d'un "suggeriment" o "recomanació" d'altres paquets. D'altres s'estan instal·lant o eliminant (com per exemple en el cas de `lbxproxy`) per les seleccions que hem realitzat mitjançant la navegació a través del llistat de paquets de `dselect`. `Dselect` pot ser una poderosa eina usada en conjunció amb APT.

3.8 Com mantenir un sistema híbrid

Algunes vegades la gent està interessada en usar una de les versions de Debian com a distribució del sistema principal i un o més paquets d'una altra branca.

Per a establir quina és la vostra versió de Debian principal haureu d'editar el `/etc/apt/apt.conf` amb una línia com la següent:

```
APT::Default-Release "versió";
```

A on *versió* serà la versió de Debian que voleu usar com a distribució principal. Les versions que podeu usar són `stable`, `testing` i `unstable`. Per a instal·lar paquets des d'una altra versió podeu usar APT de la següent manera:

```
# apt-get -t distribució install paquet
```

Per a que això funcioni, almenys necessitareu incloure una línia font per a APT en el vostre `/etc/apt/sources.list` per a la distribució de la qual voleu el paquet, i dit paquet haurà d'existir en la font.

També podreu requerir una versió específica d'un paquet usant la següent sintaxi:

```
# apt-get install paquet=versió
```

Per exemple, a la línia de a sota s'instal·larà la versió 2.2.4-1 del paquet `nautilus`:

```
# apt-get install nautilus=2.2.4-1
```

IMPORTANT: La versió "unstable" de Debian és la versió en la que primer es pugen els paquets de Debian. En definitiva és la que veu tots els canvis que aniran als paquets, uns petits i d'altres més dràstics doncs afectaran a tot el sistema. Per aquest motiu, aquesta versió de la distribució *no* hauria de ser emprada per usuaris sense experiència o pels que necessiten proveir-se d'un sistema estable.

La distribució "testing" no és necessàriament millor que "unstable", això és deu a que no reb ràpidament les actualitzacions de seguretat. Per a servidors i d'altres sistemes en producció sempre s'hauria d'usar stable.

3.9 Com actualitzar paquets des de versions específiques de Debian

`apt-show-versions` proveeix d'una manera segura per actualitzar sistemes híbrids sense tenir que obtenir una distribució menys estable de la que tenien pensada. Per exemple, és possible actualitzar tan sols els vostres paquets de unstable després de tenir instal·lat el paquet `apt-show-versions`:

```
# apt-get install `apt-show-versions -u -b | grep unstable`
```

3.10 Com mantenir versions específiques de paquets instal·lats (complexe)

Possiblement tindreu l'ocasió de modificar quelcom en un programa i no disposareu de prou temps per a portar dits canvis cap a la nova versió del programa. O, podria ser que hagueu actualitzat la vostra distribució de Debian a 3.0, però alguns paquets tingan que continuar amb la versió que hi ha a la Debian 2.2. Podeu "marcar" la versió que ha de restar instal·lada de manera que aquesta no siga actualitzada.

L'ús d'aquest recurs és simple. Sols necessitareu editar el fitxer `/etc/apt/preferences`.

El format és simple:

```
Package: <paquet>
Pin: <marca de definició>
Pin-Priority: <marca de prioritat>
```

Per exemple, per a mantenir el paquet `sylpheed` que he modificat per a usar “reply-to-list” en la seva versió 0.4.99, afegeixo:

```
Package: sylpheed
Pin: version 0.4.99*
```

Observeu que he usat un `*` (asterisc). Aquest és un “jòquer”, el qual especifica que desitjo que aquesta “marca” siga vàlida per a totes les versions que comencin per 0.4.99. Això és fa així per culpa de les versions Debian d’aquest paquet amb una “revisió Debian” i quan no desitjo bloquejar l’instal·lació de les esmentades revisions. Així doncs, les revisions 0.4.99-1 i 0.4.99-10 seran instal·lades tan aviat com estigan disponibles. Tingueu en compte que si vàreu modificar el programa no voldreu fer això.

El camp `Pin-Priority` és opcional; si no s’especifica, està per omissió a 989.

Ara veurem com funcionen les prioritats de “marca”. Una prioritat menor que 0 indica que el paquet no hauria d’instal·lar-se mai. Les prioritats de 0 a 100 denoten paquets que no estan instal·lats i que no tenen versions disponibles. Aquests no estaran presents en el procés d’escollir versions. La prioritat 100 és assignada a un paquet instal·lat - per a la versió instal·lada que serà reemplaçada per una versió diferent, el reemplaçament haurà de tenir una prioritat major que 100.

Les prioritats majors que 100 indiquen que el paquet ha de ser instal·lat. Típicament, la versió instal·lada d’un paquet només canvia al ser actualitzada a una nova versió. Qualsevol prioritat entre 100 i 1000 (inclusiu) indica aquest típic comportament. Un paquet amb dita prioritat no serà degradat per una versió anterior disponible cap a un número de versió menor. Per exemple, si teniu instal·lat `sylpheed 0.5.3` i definiu la marca com a `sylpheed 0.4.99` amb prioritat 999, el paquet 0.4.99 *no* serà instal·lat per a satisfer dita marca. Per a fer “degradable” a un paquet, satisfent la marca, aquesta necessita posseir una prioritat major que 1000.

Una marca pot ser especificada sobre la versió, versió de Debian o a l’origen d’un paquet.

Marcant sobre una versió, tal i com hem vist, suporta els números de versió literalment així com l’ús de jòquers per a especificar diversos números de versió a la vegada.

L’opció `release` dependrà del fitxer `Release` d’una rèplica APT o del d’un CD. Aquesta opció podria no ser de molta utilitat si esteu usant repositoris de paquets que no proveeixin aquest fitxer. Podeu veure el contingut dels fitxers `Release` que teniu en `/var/lib/apt/lists/`. Els seus paràmetres són: `a` (arxiu), `c` (components), `v` (versió), `o` (origen) i `l` (nivell).

Un exemple:

```
Package: *
Pin: release v=2.2*,a=stable,c=main,o=Debian,l=Debian
Pin-Priority: 1001
```

En aquest exemple, escollim la versió 2.2* of Debian (la qual pot ser 2.2r2, 2.2r3 – això són “versions puntuals” que típicament inclouen adobs de seguretat i d’altres actualitzacions importants), el repositori `stable`, secció `main` (com a oposada a `contrib` o `non-free`) i origen i nivell de Debian. L’origen (`o=`) defineix qui va crear el fitxer Release, el nivell (`l=`) defineix el nom de la distribució: Debian per a Debian mateix i Progeny per a Progeny, per exemple. Exemple d’un fitxer Release:

```
$ cat /var/lib/apt/lists/ftp.debian.org.br_debian_dists_potato_main_binary-i386
Archive: stable
Version: 2.2r3
Component: main
Origin: Debian
Label: Debian
Architecture: i386
```


Capítol 4

Ajudants de molta utilitat

4.1 Com instal·lar paquets instal·lats localment: equivs

A vegades, la gent desitja usar una versió específica d'un programa disponible tan sols des del codi font, no des d'un paquet Debian. Fer això pot ser problemàtic per al sistema d'empaquetament. Supposeu que voleu compilar una nova versió del vostre servidor de correu. Tot està correcte, però hi ha diversos paquets que depenen d'un MTA (Agent per al transport de correu). Donat que l'heu instal·lat mitjançant una compilació pròpia, el sistema de paquets no en sap res al respecte.

Aquí és a on `equivs` entra en escena. Per a usar-lo, instal·leu el paquet amb aquest nom. El que fa és crear un paquet buit que possibilita el satisfer totes les dependències, fent que el sistema de paquets doni per satisfetes dites dependències.

Abans de començar és una bona cosa el recordar-vos que hi ha maneres més segures de compilar un programa que ja està empaquetat per a Debian amb diferents opcions i que no hauríeu d'usar `equivs` per a reemplaçar dependències sinó sabeu el que esteu fent. Per a més informació consulteu la secció 'Treballar amb els paquets font' a la pàgina [31](#).

Ara continuem amb l'exemple del MTA, tot just acabeu d'instal·lar el vostre nou `postfix` compilat i aneu a instal·lar `mutt`. De sobte descobriu que `mutt` vol instal·lar un altre MTA. Però ja teniu llest el vostre.

Aneu a algun directori (`/tmp`, per exemple) i executeu:

```
# equivs-control nom
```

Substituiu *nom* pel nom del fitxer de control que voleu crear. El fitxer serà cerat de la següent manera:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1
```

```
Package: <Introduir el nom del paquet; per defecte a equivs-dummy>
Version: <Introduir la versió aquí; per defecte a 1.0>
Maintainer: <El vostre nom in adreça de correu electrònic; per defecte al nom>
Pre-Depends: <Paquets>
Depends: <Paquets>
Recommends: <Paquets>
Suggests: <Paquet>
Provides: <Paquet (virtual)>
Architecture: all
Copyright: <Fitxer de copyright; per defecte a GPL2>
Changelog: <Fitxer de canvis; per defecte al genèric changelog>
Readme: <Fitxer README.Debian; per defecte a un de genèric>
Extra-Files: <Fitxers addicionals per al directori doc, separats per comes>
Description: <Descripció curta; per defecte a algunes paraules sabies>
  Descripció llarga i informació rellevant.
.
Segon paràgraf.
```

Nosaltres tan sols necessitarem modificar això per a fer el que vulguem. Feu un cop d'ull al format del camp i a les seves descripcions, no hi ha necessitat d'explicar-les una a una, anem a fer el que requerim:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: mta-local
Provides: mail-transport-agent
```

Sí, això és tot. `mutt` depèn de `mail-transport-agent`, que és un paquet virtual proveït per tots els MTA, aquí podria simplement anomenar al paquet `mail-transport-agent`, però prefereixo usar l'esquema de paquets virtuals, usant `Provides`.

Ara només necessitareu construir el paquet:

```
# equivs-build nom
dh_testdir
touch build-stamp
dh_testdir
dh_testroot
dh_clean -k
# Add here commands to install the package into debian/tmp.
touch install-stamp
dh_testdir
```

```
dh_testroot
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol
dh_md5sums
dh_builddeb
dpkg-deb: building package `nom' en `../nom_1.0_all.deb'.
```

The package has been created.

Attention, the package has been created in the current directory,

I instal·lar el `.deb` resultant.

Tal com podeu veure, hi ha molts usos per a equivs. Es pot crear un paquet `els-meus-favorites`, el qual dependrà dels programes que instal·leu normalment, per exemple. Simplement deixeu volar la imaginació, però amb cura.

És important apuntar que hi ha fitxers de control en `/usr/share/doc/equivs/examples`. Comproveu-los.

4.2 Eliminació de fitxers locale sense usar: `localepurge`

La majoria d'usuaris Debian tan sols usen un locale (localització de l'idioma). Un usuari brasiler de Debian, per exemple, usualment usarà el locale `pt_BR` durant tot el temps i no li importarà el `es`.

`localepurge` és una eina de molta utilitat per aquests usuaris. Podreu alliberar una gran quantitat d'espai deixant instal·lats tan sols aquells locale que realment useu. Simplement executeu `apt-get install localepurge`.

És molt fàcil de configurar, les preguntes de `debconf` guiaran a l'usuari en una configuració passa a passa. Tingueu cura al respondre a la primera pregunta, una resposta errònia podria eliminar tots els fitxers locale, fins i tot aquells que necessiteu. L'única manera de recuperar-los és la de reinstal·lar els paquets que els proveïen.

4.3 Com conèixer quins paquets poden ser actualitzats

`apt-show-versions` és un programa que mostra quins paquets en el sistema poden ser actualitzats i diversa informació d'utilitat. L'opció `-u` mostra una llista dels paquets actualitzables:

```
$ apt-show-versions -u
```

```
libeel0/unstable upgradeable from 1.0.2-5 to 1.0.2-7  
libeel-data/unstable upgradeable from 1.0.2-5 to 1.0.2-7
```

Capítol 5

Obtenir informació quant als paquets.

Hi ha alguns programes d'interfície per al sistema APT que fan significativament fàcil l'aconseguir llistats dels paquets que estan disponibles per a la instal·lació o que ja ho estan, així com cercar la secció d'un paquet, prioritat, descripció, etc.

Però... el nostre objectiu aquí és el d'usar APT en estat pur. Així que: Com podreu conèixer el nom d'un paquet que voleu instal·lar?

Disposem d'un gran nombre de recursos per a realitzar aquesta tasca. Començarem exposant a `apt-cache`. Aquest programa és usat pel sistema APT per a manipular la seva base de dades. Nosaltres tan sols veurem algunes de les seves aplicacions més pràctiques.

5.1 Descobrir els noms dels paquets

Per exemple, suposem que voleu reviure l'època daurada de l'Atari 2600. Voleu usar APT per a instal·lar un emulador Atari, i després descarregar-vos alguns jocs. Podeu fer el següent:

```
# apt-cache search atari
atari-fdisk-cross - Partition editor for Atari (running on non-Atari)
circuslinux - The clowns are trying to pop balloons to score points!
madbomber - A Kaboom! clone
tcs - Character set translator.
atari800 - Atari emulator for svgalib/X/curses
stella - Atari 2600 Emulator for X windows
xmess-x - X binaries for Multi-Emulator Super System
```

Hem trobat diversos paquets relacionats amb el que estem cercant, junt amb les seves descripcions curtes. Per aconseguir més informació quant a un paquet específic, podem fer el següent:

```
# apt-cache show stella
Package: stella
```

```
Priority: extra
Section: non-free/otherosfs
Installed-Size: 830
Maintainer: Tom Lear <tom@trap.mtview.ca.us>
Architecture: i386
Version: 1.1-2
Depends: libc6 (>= 2.1), libstdc++2.10, xlib6g (>= 3.3.5-1)
Filename: dists/potato/non-free/binary-i386/otherosfs/stella_1.1-2.deb
Size: 483430
MD5sum: 11b3e86a41a60falc4b334dd96c1d4b5
Description: Atari 2600 Emulator for X windows
  Stella is a portable emulator of the old Atari 2600 video-game console
  written in C++. You can play most Atari 2600 games with it. The latest
  news, code and binaries for Stella can be found at:
  http://www4.ncsu.edu/~bwmott/2600
```

En aquesta eixida teniu molt detallada quant al paquet que voleu (o que no voldreu) instal·lar, junt amb una completa descripció del mateix. Si el paquet ja estigués instal·lat al vostre sistema i haguéssiu una versió més recent, veuríeu l'informació d'ambdues versions. Per exemple:

```
# apt-cache show lilo
Package: lilo
Priority: important
Section: base
Installed-Size: 271
Maintainer: Russell Coker <russell@coker.com.au>
Architecture: i386
Version: 1:21.7-3
Depends: libc6 (>= 2.2.1-2), debconf (>=0.2.26), logrotate
Suggests: lilo-doc
Conflicts: manpages (<<1.29-3)
Filename: pool/main/l/lilo/lilo_21.7-3_i386.deb
Size: 143052
MD5sum: 63fe29b5317fe34ed8ec3ae955f8270e
Description: LInux LOader - The Classic OS loader can load Linux and others
  This Package contains lilo (the installer) and boot-record-images to
  install Linux, OS/2, DOS and generic Boot Sectors of other OSes.
.
  You can use Lilo to manage your Master Boot Record (with a simple text screen)
  or call Lilo from other Boot-Loaders to jump-start the Linux kernel.

Package: lilo
Status: install ok installed
Priority: important
Section: base
```

```

Installed-Size: 190
Maintainer: Vincent Renardias <vincent@debian.org>
Version: 1:21.4.3-2
Depends: libc6 (>= 2.1.2)
Recommends: mbr
Suggests: lilo-doc
Description: LInux LOader - The Classic OS loader can load Linux and others
 This Package contains lilo (the installer) and boot-record-images to
 install Linux, OS/2, DOS and generic Boot Sectors of other OSes.
.
You can use Lilo to manage your Master Boot Record (with a simple text screen)
or call Lilo from other Boot-Loaders to jump-start the Linux kernel.

```

Observeu que la primera en la llista és la del paquet disponible i que la segona correspon al paquet que ja està instal·lat. Per a informació més general quant a un paquet, podeu usar:

```

# apt-cache showpkg penguin-command
Package: penguin-command
Versions:
1.4.5-1 (/var/lib/apt/lists/download.sourceforge.net_debian_dists_unstable_main
Reverse Depends:
Dependencies:
1.4.5-1 - libc6 (2 2.2.1-2) libpng2 (0 (null)) libsdl-mixer1.1 (2 1.1.0) libsnd
Provides:
1.4.5-1 -
Reverse Provides:

```

I per a saber els paquets del que depèn:

```

# apt-cache depends penguin-command
penguin-command
  Depends: libc6
  Depends: libpng2
  Depends: libsdl-mixer1.1
  Depends: libsdl1.1
  Depends: zlib1g

```

En resum, tenim una gran varietat d'armes que podem usar per a esbrinar el nom d'un paquet que desitgem instal·lar.

5.2 L'ús de dpkg per a cercar noms de paquet

Una de les maneres per a localitzar el nom d'un paquet és coneixent el nom d'un fitxer important que es trobi dins d'aquest paquet. Per exemple, per a esbrinar el paquet que proveeix un

fitxer “.h” en particular que necessiteu per a una compilació podeu executar:

```
# dpkg -S stdio.h
libc6-dev: /usr/include/stdio.h
libc6-dev: /usr/include/bits/stdio.h
perl: /usr/lib/perl/5.6.0/CORE/nostdio.h
```

o:

```
# dpkg -S /usr/include/stdio.h
libc6-dev: /usr/include/stdio.h
```

Per a esbrinar els noms dels paquets instal·lats en el vostre sistema, el qual és d'utilitat, per exemple, si planegeu netejar el vostre disc dur, podeu executar:

```
# dpkg -l | grep mozilla
ii mozilla-browser 0.9.6-7 Mozilla Web Browser
```

El problema amb aquest comandament és que pot “trencar” el nom del paquet. A l'anterior exemple, el nom complet del paquet és mozilla-browser. Per adobar això, podeu usar la variable d'entorn COLUMNS de la següent manera:

```
[kov]@[couve] $ COLUMNS=132 dpkg -l | grep mozilla
ii mozilla-browser 0.9.6-7 Mozilla Web Brows
```

o la descripció o part d'aquesta de la següent manera:

```
# apt-cache search "Mozilla Web Browser"
mozilla-browser - Mozilla Web Browser
```

5.3 Com instal·lar paquets “sota demanda”

Esteu compilant un programa i de sobte, boom! Apareix un error perquè necessita una fitxer .h que no teniu. El programa auto-apt pot prevenir-vos d'aquestes situacions. Us demanarà d'instal·lar certs programes que són requerits, aturant els processos rellevants i continuant-los una vegada el paquet estiga instal·lat.

El que es fa, és bàsicament executar:

```
# auto-apt run comandament
```


A on “comandament” és el comandament que al ser executat podria necessitar d’alguns fitxers no disponibles. Per exemple:

```
# auto-apt run ./configure
```

Llavors us demanarà per a instal·lar els paquets necessaris i cridarà automàticament a `apt-get`. Si esteu executant `X`, una interfície gràfica substituirà a la interfície per defecte (mode text).

`Auto-apt` manté una sèrie de bases de dades que necessiten estar al dia per a que això siga efectiu. El qual es pot aconseguir cridant als comandaments `auto-apt update`, `auto-apt updatedb` i `auto-apt update-local`.

5.4 Com esbrinar a quin paquet pertany un fitxer

Si voleu instal·lar un paquet i no podeu esbrinar com s’anomena aquest cercant-lo amb `apt-cache`, però coneixeu el nom d’un fitxer del mateix programa o el de cap altre fitxer que pertanyi a aquest paquet, llavors podeu usar `apt-file` per a cercar el nom del paquet. Això es fa de la següent manera:

```
$ apt-file search nom_fitxer
```

Funciona de mode similar a `dpkg -S`, però també us mostra paquets no instal·lats que continguin el fitxer. També pot usar-se per a esbrinar quins paquets contenen els fitxers “include” necessaris per a la compilació de certs programes, tot i que per aquests casos `auto-apt` és una solució millor per aquests casos, consulteu ‘Com instal·lar paquets “sota demanda”’ a la pàgina anterior.

També podeu llistar el contingut d’un paquet executant:

```
$ apt-file list nom_paquet
```

`apt-file` manté una base de dades amb els fitxers que contenen tots els paquets, de la mateixa manera que ho fa `auto-apt` i per això necessita estar actualitzada. Això s’aconsegueix executant:

```
# apt-file update
```

Per defecte, `apt-file` usa la mateixa base de dades que `auto-apt`, consulteu ‘Com instal·lar paquets “sota demanda”’ a la pàgina anterior.

5.5 Com mantindràs informat quant als canvis en els paquets.

Cada paquet instal·la en el seu directori de documentació (`/usr/share/doc/nom_paquet`) un fitxer anomenat `changelog.Debian.gz` el qual conté una llista dels canvis realitzats al paquet des de l'última versió. Podeu llegir aquests fitxers amb l'ajuda de `zless`, per exemple, però no és gaire simple que després d'una actualització s'hagin d'estar cercant aquests registres de canvis per a cadascun dels paquets actualitzats.

Hi ha una manera per automatitzar aquesta tasca mitjançant una eina que s'anomena `apt-listchanges`. Per a començar primer es requereix que instal·leu el paquet `apt-listchanges`. Durant la seva instal·lació, `Debconf` la configurarà. Respongueu les preguntes com desitgeu.

L'opció "Voldreu que `apt` executi automàticament `apt-listchanges`?" és realment útil perquè mostra una llista dels canvis que s'han realitzat per a cadascun dels paquets que està instal·lant `apt` durant una actualització i us permet analitzar-los abans de continuar. L'opció "Tindrà `apt-listchanges` que demanar confirmació després de mostrar els canvis?" és d'utilitat perquè us demana si desitgeu continuar l'instal·lació després de llegir la llista de canvis. Si decidiu no continuar, `apt-listchanges` retornarà un error i avortarà l'instal·lació.

Una vegada instal·lat `apt-listchanges`, tan aviat com els paquets hagin estat descarregats (o llegits des d'un CD o dic muntat) per `APT`, es mostraran les llistes de canvis realitzats a dits paquets abans d'instal·lar-los.

Capítol 6

Treballar amb els paquets font

6.1 Descàrrega de paquets font

És molt comú en el món de programari lliure l'estudi del codi font o fer correccions a codi erroni. Per a fer això, necessitareu descarregar la font del programa. El sistema APT proveeix d'una senzilla manera per a obtenir el codi font de la majoria de programes continguts en la distribució, inclosos tots els fitxers necessaris per a crear un `.deb` per al programa.

Un altre ús comú de les fonts de Debian és el d'adaptar una versió més recent d'un programa, des de la distribució unstable, per exemple, per a usar-lo amb la distribució stable. La compilació d'un paquet dintre de stable generarà fitxers `.deb` amb les dependències ajustades per a coincidir amb els paquets disponibles en dita distribució.

Per aconseguir això, l'entrada `deb-src` en el vostre `/etc/apt/sources.list` haurà d'apuntar cap a unstable. Aquesta haurà d'estar habilitada (sense comentar). Consultar la secció 'El fitxer `/etc/apt/sources.list`' a la pàgina 3.

Per a descarregar un paquet font, haureu d'usar el següent comandament:

```
$ apt-get source nom_paquet
```

Això us descarregarà tres fitxers: Un `.orig.tar.gz`, un `.dsc` i un `.diff.gz`. En el cas de paquets fets específicament per Debian, l'últim d'aquests fitxers no serà descarregat i el primer usualment no tindrà "orig" en el nom.

El fitxer `.dsc` és usat per `dpkg-source` per a desempaquetar la font del paquet en el directori `nompaquet-versió`. Amb cada paquet font descarregat existeix un directori `debian/` que conté els fitxers necessaris per a la creació del paquet `.deb` corresponent.

Per a construir automàticament el paquet que s'està descarregant, tan sols heu d'afegir `-b` a la línia de comandaments, d'aquesta manera:

```
$ apt-get -b source nom_paquet
```

Si decidiu no crear el `.deb` en temps de descàrrega, podreu crear-lo més tard executant:

```
$ dpkg-buildpackage -rfakeroot -uc -b
```

des del directori que ha estat creat després de descarregar el paquet. Per a instal·lar el paquet que ha estat creat pels anteriors comandaments, s'haurà d'emprar el sistema de gestió de paquets directament, d'aquesta manera:

```
# dpkg -i fitxer.deb
```

Hi ha una diferència entre el mètode `apt-get source` i les seves altres opcions. El mètode `source` pot ser usat per usuaris normals, sense necessitar poders especials de root. Els fitxers són descarregats al directori des del que es crida al comandament `apt-get source package`.

6.2 Paquets necessaris per a compilar un paquet font

Normalment, és necessària la presència de capçaleres o biblioteques compartides per a que un paquet font siga compilat. Tots els paquets font tenen un camp en els seu fitxer de control anomenat "Build-Depends:" el qual indica quins paquets addicionals són necessaris per a que el paquet siga construït des de la font.

APT té un mode simple de descarregar aquests paquets. Simplement executeu `apt-get build-dep paquet`, a on "paquet" serà el nom del paquet que aneu a construir. Per exemple:

```
# apt-get build-dep gmc
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  comerr-dev e2fslibs-dev gdk-implib-dev implib-progs libgnome-dev libgnorba-de
  libgpmgl-dev
0 packages upgraded, 7 newly installed, 0 to remove and 1 not upgraded.
Need to get 1069kB of archives. After unpacking 3514kB will be used.
Do you want to continue? [Y/n]
```

Els paquets que seran instal·lats són els necessaris per a construir `gmc` correctament. És important fer notar que aquest comandament no cerca el paquet font del programa per a ser compilat. Per a obtenir-lo, llavors necessitareu executar `apt-get source` per separat.

Si tot el que voleu és comprovar quins paquets són necessaris per a construir un paquet en concret, hi ha una variant del comandament `apt-cache show` (consultar 'Obtenir informació quant als paquets.' a la pàgina 25, el qual mostra, entre d'altre informació, la línia `Build-Depends` que està llistada en aquests paquets.

```
# apt-cache showsrc paquet
```

Capítol 7

Com batre's amb els errors

7.1 Errors comuns

Els errors sempre existiran, la majoria dels quals són provocats per usuaris que no paren atenció. La següent és una llista d'alguns dels errors més comuns i com tractar amb aquests.

Si rebeu un missatge semblant al següent quan intenteu executar `apt-get install paquet...`

```
Reading Package Lists... Done
Building Dependency Tree... Done
W: Couldn't stat source package list 'http://people.debian.org unstable/ Pack
W: You may want to run apt-get update to correct these missing files
E: Couldn't find package penguineyes
```

us heu oblidat d'executar `apt-get update` després del vostre últim canvi al fitxer `/etc/apt/sources.list`.

Si l'error mostrat és com aquest:

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

quan escolliu qualsevol altre mètode per a `apt-get` diferent a `source`, és que no disposeu dels permisos d'usuari `root`, el qual ve a indicar, que l'esteu executant com a un usuari normal.

Hi ha un error similar a l'anterior que passa quan executeu dues còpies de `apt-get` alhora o quan intenteu executar `apt-get` mentre un procés `dpkg` està actiu. L'únic mètode que es pot emprar simultàniament amb els altres és `source`.

Si una instal·lació peta a la meitat del procés i us trobeu amb que ja no és possible instal·lar o eliminar paquets, intenteu executar aquests dos comandaments:

```
# apt-get -f install
# dpkg --configure -a
```

I després proveu-ho una altra vegada. Podria ser necessari executar el segon comandament més d'una vegada. Aquest és una important lliçó per aquells aventurers que usen "unstable".

Si rebeu l'error "E: Dynamic MMap ran out of room" a l'executar `apt-get update`, afegiu la següent línia a `/etc/apt/apt.conf`:

```
APT::Cache-Limit 10000000;
```

7.2 A on puc trobar ajuda?

Si esteu plagat de dubtes, consulteu la extensa documentació disponible per al sistema d'empaquetament de Debian. L'opció `--help` i les pàgines del manual poden ser-vos de gran ajuda, així com la documentació inclosa en els directoris `/usr/share/doc`, com per exemple `/usr/share/doc/apt`.

Si aquesta documentació no aconsegueix apaivagar les vostres pors, intenteu cercar la resposta per les llistes de correu de Debian. Podeu trobar més informació quant a les llistes específiques per a usuaris en el lloc web de Debian: <http://www.debian.org>.

Recordeu que aquestes llistes i recursos tan sols les han d'emprar els usuaris de Debian; els usuaris d'altres sistemes trobaran un millor suport des dels recursos comunitaris de les seves pròpies distribucions.

Capítol 8

Quines distribucions tenen suport per a APT?

Aquí hi ha els noms d'algunes de les distribucions que usen APT:

Debian GNU/Linux (<http://www.debian.org>) - aquesta fou la distribució en que es desenvolupà APT

Conectiva (<http://www.conectiva.com.br>) - aquesta fou la primera distribució en portar l'ús d'APT amb rpm

Libranet (<http://www.libranet.com>)

Mandrake (<http://www.mandrake.com>)

PLD (<http://www.pld.org.pl>)

Vine (<http://www.vinelinux.org>)

APT4RPM (<http://apt4rpm.sf.net>)

Alt Linux (<http://www.altlinux.ru/>)

Red Hat (<http://www.redhat.com/>)

Sun Solaris (<http://www.sun.com/>)

SuSE (<http://www.suse.de/>)

Yellow Dog Linux (<http://www.yellowdoglinux.com/>)

Capítol 9

Crèdits

Un gran abraç d'agraïment als meus amics en el projecte Debian-BR i al mateix Debian, els quals m'estan ajudant a continuar la tasca en benefici de l'humanitat, així com en ajudar-me amb el meu objectiu de salvar al món. :)

També agraeixo l'enorme ajuda que ha donat CIPSGA al nostre projecte i a tots els projectes lliures que ofereixen noves idees.

I un especial agraïment per a:

Yooseong Yang <yooseong@debian.org>

Michael Bramer <grisu@debian.org>

Bryan Stillwell <bryan@bokeoa.com>

Pawel Tecza <pawel.tecza@poczta.fm>

Hugo Mora <h.mora@melix.com.mx>

Luca Monducci <luca.mo@tiscali.it>

Tomohiro KUBOTA <kubota@debian.org>

Pablo Lorenzoni <spectra@debian.org>

Steve Langasek <vorlon@netexpress.net>

Arnaldo Carvalho de Melo <acme@conectiva.com.br>

Erik Rossen <rossen@freesurf.ch>

Ross Boylan <RossBoylan@stanfordalumni.org>

Matt Kraai <kraai@debian.org>

Aaron M. Ucko <ucko@debian.org>

Jon Åslund <d98-jas@nada.kth.se>

Capítol 10

Noves versions d'aquest tutorial

Aquest manual ha estat creat pel projecte Debian-BR (<http://www.debian-br.org>), amb l'objectiu d'ajuda a l'usuari de Debian en el seu fer diari.

Noves versions d'aquest document estaran disponibles a la pàgina de Projectes de documentació de Debian, a <http://www.debian.org/doc/ddp>.

Els comentaris i crítiques me les podeu enviar directament per correu a <kov@debian.org>.