

Guía de referencia rápida Debian

Osamu Aoki <osamu@debian.org>
Coordinador de la traducción al español: Walter O. Echarri
<wecharri@infovia.com.ar>
'Autores' en la página [27](#)

CVS, jue jun 10 18:54:08 UTC 2004

Resumen

Esta Guía de referencia rápida Debian (<http://qref.sourceforge.net/>) intenta proporcionar una breve introducción al sistema Debian al igual que una **referencia rápida**. Es un extracto de Guía de referencia Debian (<http://qref.sourceforge.net/>).

Nota de Copyright

Copyright © 2001–2004 by Osamu Aoki <osamu@debian.org>

Este documento puede ser usado en los términos descritos en la Licencia Pública GNU versión 2 o posterior. (<http://www.gnu.org/copyleft/gpl.html>)

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions, except that this permission notice may be included in translations approved by the Free Software Foundation instead of in the original English.

Índice general

1. Prefacio	1
1.1. Convenciones en los documentos	1
1.2. Fundamentos sobre las distribuciones Debian	1
2. Actualizando una distribución	3
2.1. Transición a la versión Woody de APT	3
2.2. Preparación para la transición (<i>stable</i> a <i>testing</i>)	3
2.3. Actualizar el sistema Debian	4
2.3.1. Mejor método de actualización usando <i>dselect</i>	5
3. Administración de paquetes Debian	7
3.1. Introducción	7
3.1.1. Herramientas principales	7
3.1.2. Herramientas prácticas	8
3.2. Fundamentos de la administración de paquetes Debian Debian	8
3.2.1. Instalación de <i>tasks</i> con <i>tasksel</i>	8
3.2.2. Configurar el sistema APT	9
3.2.3. <i>dselect</i>	9
3.2.4. <i>aptitude</i>	10
3.2.5. Los comandos <i>apt-cache</i> y <i>apt-get</i>	10
3.2.6. Permanecer en una versión de la distribución Debian	12
3.2.7. Volver todos los paquetes a su versión estable	12
3.2.8. Repaso de <i>/etc/apt/preferences</i>	12
3.3. Comandos de supervivencia de Debian	13

3.3.1.	Búsqueda de fallos y ayuda en Debian	14
3.3.2.	Resolución de problemas al actualizar con APT	14
3.3.3.	Rescate usando dpkg	15
3.3.4.	Recuperar los datos de la selección de paquetes	16
3.3.5.	Recuperando un sistema luego de borrar /var	16
3.3.6.	Instalar un paquete en un sistema que no arranca	16
3.3.7.	Qué hacer si el comando dpkg se encuentra dañado	17
3.4.	Comandos 'nirvana' de Debian	17
3.4.1.	Información sobre un archivo	17
3.4.2.	Información sobre un paquete	18
3.4.3.	Instalación automática con APT	19
3.4.4.	Reconfigurar paquetes instalados	19
3.4.5.	Desinstalar y purgar paquetes	20
3.4.6.	Conservar paquetes antiguos	20
3.4.7.	Sistema mixto estable/de prueba/inestable	21
3.4.8.	Eliminar paquetes del caché	21
3.4.9.	Grabar/copiar la configuración del sistema	21
3.4.10.	Portar un paquete a la rama stable	21
3.4.11.	Archivo local de paquetes	22
3.4.12.	Convertir o instalar un paquete que no pertenece a Debian	23
3.4.13.	Verificar los paquetes instalados	23
3.5.	Otras particularidades de Debian	24
3.5.1.	El comando dpkg-divert	24
3.5.2.	El paquete equivs	24
3.5.3.	Comandos alternativos	24
3.5.4.	init del System-V y niveles de ejecución	25
3.5.5.	Servicios desactivados	26
A.	Apéndice	27
A.1.	Autores	27
A.2.	Garantías	29
A.3.	Comentarios	29

Capítulo 1

Prefacio

Resúmalo y simplifíquelos es mi principio rector.

1.1. Convenciones en los documentos

La “Guía de referencia rápida Debian” proporciona información mediante comandos del shell Bash.

Una referencia:

- en una página del **manual** se muestra así `bash(1)`.
- en una página TEXINFO GNU se muestra así `info libc`.

1.2. Fundamentos sobre las distribuciones Debian

Debian viene en 3 versiones o “sabores”:

- `estable`: conveniente para un servidor. Poco interesante para una estación de trabajo.
- `de prueba`: conveniente para una estación de trabajo.
- `inestable`: úsela con cierta prudencia.

Lea por lo menos la lista de correo `debian-devel-announce@lists.debian.org` para estar al tanto sobre las novedades de Debian.

En el mes de marzo del 2002, estas tres versiones correspondían a Potato (calidad de producción), Woody (versión beta, muy estable ya que su lanzamiento era inminente) y Sid (versión alfa). En agosto del 2002, justo después del lanzamiento de Woody, estas correspondían a Woody (calidad de producción), Sarge (versión beta, será un poco difícil durante cierto tiempo) y Sid (siempre es la versión alfa). Cuando los paquetes de la rama `inestable` no poseen

fallos críticos (RC) dados a conocer durante la primera semana de pruebas, automáticamente se transfieren a la rama de `prueba`.

En teoría, existen dos cosas que puede hacer para obtener las últimas versiones de los programas.

- ‘Fundamentos de la administración de paquetes Debian Debian’ en la página 8 (principalmente para estaciones de trabajo)
- ‘Portar un paquete a la rama `stable`’ en la página 21 (principalmente para servidores)

Capítulo 2

Actualizando una distribución

2.1. Transición a la versión Woody de APT

La actualización a través de la red del sistema APT y de algunos paquetes fundamentales a la versión se puede realizar de la siguiente manera luego de incluir repositorios de la rama estable en el `/etc/apt/sources.list` si aún desea usar Potato.

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

2.2. Preparación para la transición (stable a testing)

Utilizar la versión de prueba de Debian tiene como efecto colateral la lenta solución de los fallos de seguridad. Por lo tanto, tenga cuidado.

La actualización a la versión de prueba (testing) se puede realizar de la siguiente manera (ejecute el guión go-woody (<http://www.debian.org/doc/manuals/debian-reference/examples/>) para hacerlo todo mediante un sólo comando):

Vacíe el archivo `sources.list` actual:

```
# cd /etc/apt
# cp -f sources.list sources.anterior
# :>sources.list
```

Obtenga una lista reciente de repositorios para la versión estable (stable)

```
# cd /
# apt-setup noprobe
... seleccione los repositorios, accesibles mediante HTTP o FTP
```

Añada la sección `testing` a la lista nueva. Las líneas `deb-src` están comentadas.

```
# cd /etc/apt
# grep -e "^deb " sources.list >sources.deb
# grep -e "^deb-" sources.list >sources.src
# sed -e "s/stable/testing/" sources.deb \
    >>sources.list
# sed -e "s/stable/testing/" sources.src | \
    sed -e "s/^deb-/#deb-/" >>sources.list

# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
Package: *
Pin: release a=testing
Pin-Priority: 600

Package: *
Pin: release a=unstable
Pin-Priority: 50

EOF
```

Opcionalmente, añade la sección “unstable” de los archivos.

```
# sed -e "s/stable/unstable/" sources.deb \
    >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
    sed -e "s/^deb-/#deb-/" >>sources.list
```

Véase ‘Fundamentos de la administración de paquetes Debian Debian’ en la página 8 configurar inteligentemente `/etc/apt/sources.list` y `/etc/apt/preferences`.

Ahora puede actualizar el sistema, usando uno de los métodos de la siguiente sección.

2.3. Actualizar el sistema Debian

Luego de configurar en forma adecuada los archivos `/etc/apt/sources.list` y `/etc/apt/preferences`, el sistema se puede actualizar a la versión de prueba (`testing`). Véase ‘Administración de paquetes Debian’ en la página 7 para los fundamentos y ‘Resolución de problemas al actualizar con APT’ en la página 14 si tropieza con problemas.

2.3.1. Mejor método de actualización usando `dselect`

Si un sistema tiene muchos paquetes que incluye paquetes `-dev`, etc., se recomienda el siguiente método usando `dselect` para un control minucioso de paquetes.

```
# dselect update # siempre hacer esto antes de actualizar
# dselect select # seleccione los paquetes adicionales
```

Todos los paquetes actuales serán seleccionados al arrancar `dselect`. `dselect` puede proponerle otros paquetes: dependientes, sugeridos y recomendados. Si no desea añadir ningún paquete, sólo pulse `Q` para abandonar `dselect`.

```
# dselect install
```

Deberá responder algunas preguntas para la configuración para esta parte del proceso, por lo tanto tome nota y dedique parte de su tiempo a esta parte. Véase ‘`dselect`’ en la página 9.

Use `dselect`. **Siempre funciona :)** Si necesita actualizar sin `dselect` a partir de Woody, considere el uso de `aptitude` u otras opciones.

Capítulo 3

Administración de paquetes Debian

Para reducir el tráfico en los repositorios Debian, asegúrese de configurar un proxy HTTP local usando `squid` para los paquetes descargados mediante APT y, de ser necesario, configurar la variable de entorno `http_proxy` o fijar el valor `http` en `/etc/apt/apt.conf`. Esto mejora notablemente el rendimiento de las actualizaciones en una red, en especial en una LAN con una gran cantidad de sistemas Debian.

Aunque el método de marcado (pinning) de `apt_preferences(5)` es poderoso, no resuelve la totalidad de temas sobre dependencias puesto que las definiciones de dependencias tienden a buscar las versiones más nuevas de otros programas fundamentales.

El uso de `chroot` es útil para asegurar, simultáneamente, tanto la estabilidad del sistema como el acceso a las últimas versiones de los programas.

Este capítulo se basa en sistemas Woody pero la mayoría de la información se aplica también a sistemas Potato (con excepción de `apt_preferences(5)` y lo concerniente al archivo `/etc/apt/preferences`).

3.1. Introducción

Si el leer toda la documentación del desarrollador es demasiado para Ud., lea primero este capítulo y empiece a disfrutar toda la potencia de Debian con `testing/unstable :-)`

3.1.1. Herramientas principales

```
dselect    - herramienta de administración de paquetes basada en menús
              (alto nivel)
dpkg       - instala paquetes (package-file centric)
apt-get    - instala paquetes (package-archive centric, CLI APT)
tasksel    - instala tareas (conjunto de paquetes)
aptitude  - instala paquetes (paquetes y tareas, interfaz de APT basada en
```

```
ncurses)
deity      - otra interfaz para APT basada en ncurses
synaptic, gsynaptic - otras interfaces gráficas para APT
```

Estas herramientas no están en un mismo nivel. `dselect` se ejecuta encima de APT (`apt-get`) y `dpkg`.

APT utiliza `/var/lib/apt/lists/*` para el seguimiento de paquetes disponibles, mientras que `dpkg` usa `/var/lib/dpkg/available`. Si instaló paquetes directamente usando `apt-get` o programas similares como, por ejemplo, `aptitude`, asegúrese de actualizar el archivo `/var/lib/dpkg/available` mediante la opción `[U]pdate` del menú de `dselect` o desde la línea de comandos “`dselect update`” antes de ejecutar `dselect select`, `tasksel` o `dpkg -l`.

En cuanto a la dependencia entre paquetes, `apt-get` obtiene automáticamente los paquetes que dependen entre sí (**depends**) pero ignora los paquetes recomendados (**recommends**) y sugeridos (**suggests**). `dselect`, en cambio, ofrece un control más avanzado de los paquetes basado en menús que ofrece la posibilidad de elegir paquetes basándose en **depends**, **recommends** y **suggests**. `aptitude` tiene la opción de descargar todos los paquetes automáticamente basándose en **depends**, **recommends** y **suggests**.

3.1.2. Herramientas prácticas

```
apt-cache      - búsqueda de paquetes en el caché local
dpkg-reconfigure - vuelve a configurar un paquete ya instalado (si utiliza
                 debconf)
dpkg-source    - se encarga de los archivos fuentes
dpkg-buildpackage - automatiza la creación de un paquete
...
```

3.2. Fundamentos de la administración de paquetes Debian Debian

Puede instalar un conjunto de paquetes denominado *tarea* o instalar paquetes individuales o actualizar el sistema usando las herramientas de administración de paquetes como se describe a continuación.

3.2.1. Instalación de *tareas* con `tasksel`

`tasksel` es el **Instalador de Tareas de Debian** que se ofrece durante la instalación como método de instalación “sencillo”.

Cuando uno necesita instalar una determinada función que requiere de múltiples paquetes, esta es la mejor manera de hacerlo. Asegúrese de ejecutar los comandos en el siguiente orden:

```
# dselect update
# tasksel
```

aptitude también ofrece acceso a las *tareas*. No sólo permite seleccionar *tareas* sino que permite eliminar paquetes de una *tarea* en forma selectiva mediante un menú.

3.2.2. Configurar el sistema APT

Para realizar actualizaciones selectivas desde la distribución de prueba el sistema APT (>Woody) debe configurarse como en ‘Transición a la versión Woody de APT’ en la página 3 para usar aprovechar las características de `apt_preferences(5)`.

Primero añade las fuentes para la versión estable (`stable`), de prueba (`testing`) e inestable (`unstable`) a su `/etc/apt/sources.list`. A continuación, edite `/etc/apt/preferences` para configurar correctamente el Pin-Priority:

```
Package: *
Pin: release a=stable
Pin-Priority: 500

Package: *
Pin: release a=testing
Pin-Priority: 600

Package: *
Pin: release a=unstable
Pin-Priority: 50
```

3.2.3. dselect

Al arrancar, `dselect` selecciona automáticamente todos los paquetes “Requeridos” (“Required”), “Importantes” (“Important”) y “Estándar” (“Standard”). En el sistema Potato, algunos programas muy grandes tales como `teTeX` y `Emacs` solían estar aquí y resultaba mejor omitirlos en una instalación inicial deseleccionándolos manualmente (mediante ‘_’). En Woody, han sido puestos en la categoría de paquetes “Opcionales” (“Optional”).

`dselect` tiene una interfaz de usuario algo extraña. Existen cuatro comandos un tanto ambiguos (Mayúsculas, significa ¡MAYÚSCULAS!):

Tecla	Acción
Q	Salir. Confirma la selección actual y sale de todas maneras. (no tiene en cuenta las dependencias)
R	¡Reestablecer! No quise decir esto.
D	¡Maldición! No me importa lo que piensa <code>dselect</code> . ¡Házlo!
U	Volver todo en el estado <code>sUgerido</code>

Con `D` y `Q`, puede elegir paquetes en conflicto bajo su entera responsabilidad. Utilice estos comandos con cuidado. Actualmente, `dselect` es una herramienta basada en menús madura que ofrece un control muy preciso sobre los paquetes sugeridos (**suggests**) y recomendados (**recommends**).

Añada una línea que contenga la opción “expert” en `/etc/dpkg/dselect.cfg` para reducir la cantidad de información mostrada. Para una máquina más lenta, puede ejecutar `dselect` en otra máquina más rápida para encontrar los paquetes y usar `apt-get install` para instalarlos.

`dselect` no ofrece acceso a los paquetes que no han sido favorecidos por la Pin-Priority.

3.2.4. aptitude

`aptitude` es un instalador de paquetes nuevo basado en menús similar a `dselect`. También se puede como alternativa de `apt-get` en la línea de comandos. Véase `aptitude(1)`.

`aptitude` acepta comandos de una tecla, que generalmente están en minúsculas.

Tecla	Acción
F10	Menú
?	Ayuda
u	Actualizar la información del archivo de paquetes
g	Descargar e instalar los paquetes seleccionados
q	Salir de la pantalla actual y guardar los cambios
x	Salir de la pantalla actual y descartar los cambios
Enter	Ver información sobre un paquete

`aptitude` ofrece la opción de descargar automáticamente todos los paquetes basándose en **depends**, **recommends** y **suggests**. Puede alterar este comportamiento eligiendo `F10 ->Opciones ->Manejo de dependencias` de su menú.

`aptitude` permite el acceso a todas las versiones de un paquete.

3.2.5. Los comandos apt-cache y apt-get

Con el ejemplo de arriba para trabajar en la versión de `prueba`, podemos controlar el sistema usando los siguientes comandos:

- `apt-get -u upgrade`
Permanece en la distribución de `prueba` y actualiza todos los paquetes del sistema instalando las dependencias desde ella misma.
- `apt-get -u dist-upgrade`
Permanece en la distribución de `prueba` y actualiza todos los paquetes del sistema resolviendo las dependencias desde ella misma.

- `apt-get -u dselect-upgrade`
Permanece en la distribución de prueba y actualiza todos los paquetes del sistema teniendo en cuenta la selección de `dselect`.
- `apt-get -u install paquete`
Instala el *paquete* y sus dependencias desde la distribución de prueba.
- `apt-get -u install package/unstable`
Instala el *paquete* desde la distribución inestable y sus dependencias desde la distribución de prueba.
- `apt-get -u install -t unstable paquete`
Instala el *paquete* y sus dependencias desde la distribución inestable asignando una Pin-Priority igual a 990 a `unstable`.
- `apt-cache policy loquesea1 loquesea2 ...`
Comprueba el estado de los paquetes *loquesea1 loquesea2 ...*.
- `apt-cache show loquesea1 loquesea2 | less`
Comprueba la información referente a los paquetes *loquesea1 loquesea2 ...*.
- `apt-get install loquesea=2.2.4-1`
Instala una versión particular (*2.2.4-1*) del paquete *loquesea*.
- `apt-get -u install loquesea1 loquesea2-`
Instala el paquete *loquesea1* y desinstala el paquete *loquesea2*.
- `apt-get remove loquesea1`
Desinstala el paquete *loquesea1* pero conserva intactos los archivos de configuración personalizados.
- `apt-get remove --purge loquesea1`
Desinstala el paquete *loquesea1* junto con todos sus archivos de configuración personalizados.

En los ejemplos anteriores, el uso de la opción `-u` asegura que `apt-get` muestre la lista de todos los paquetes que van a ser actualizados y pregunte al usuario antes de proceder. Lo siguiente elige la opción `-u` como acción predeterminada:

```
$ cat >> /etc/apt/apt.conf << .  
// Mostrar siempre los paquetes que van a ser actualizados (-u)  
APT::Get::Show-Upgraded "true";  
.
```

Utilice la opción `-s` para simular la actualización sin llevarla realmente a cabo.

3.2.6. Permanecer en una versión de la distribución Debian

Dependiendo de su preferencia por permanecer en una determinada versión de Debian, puede modificar el archivo de ejemplo `/etc/apt/preferences` en 'Configurar el sistema APT' en la página 9 para que se ajuste a sus necesidades:

```
permanecer en estable:           poner la Pin-Priority de testing en 50
permanecer en testing:          mantener la configuración dada anteriormente
permanecer en testing (unstable): poner la Pin-Priority de unstable en 500
permanecer en unstable(testing): poner la Pin-Priority de unstable en 800
```

Una idea para la elección del Pin-Priority consiste en moverse desde arriba hacia abajo en la tabla anterior conforme transcurre el tiempo entre la fecha inmediatamente posterior a la salida de una distribución y la fecha de congelación de la siguiente versión.

Advertencia: el permanecer en la versión de prueba de Debian tiene como efecto colateral la lenta solución de los fallos de seguridad.

Si mezcla las versiones de Debian, de prueba con estable o inestable con estable, es muy probable que descargue diversos paquetes importantes de la rama de prueba o inestable que pueden tener errores. Queda debidamente advertido.

Ejemplos de `/etc/apt/preferences` que bloquean algunos paquetes fundamentales en su versión más madura mientras actualizan a una versión menos madura paquetes que no son esenciales, están disponibles en el examples subdirectory (<http://www.debian.org/doc/manuals/debian-reference/examples/>) como `preferences.testing` y `preferences.unstable`. Por otra parte, `preferences.stable` fuerza a todos los paquetes a volver a su versión estable.

3.2.7. Volver todos los paquetes a su versión estable

Para volver todos los paquetes a la versión estable, edite el `/etc/apt/preferences` de la siguiente manera:

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

y ejecute `apt-get upgrade` que forzará el retorno a la rama estable debido a que `Pin-priority > 1000`. Tenga cuidado, puede haber algunos inconvenientes con las dependencias.

3.2.8. Repaso de `/etc/apt/preferences`

En el `/etc/apt/preferences` de ejemplo presentado en 'Configurar el sistema APT' en la página 9, la primera línea `Package: *` significa que esa sección se aplica a todos los paquetes. Puede usar el nombre de un paquete en vez de `*` para especificar la Pin-Priority del mismo.

En el segundo renglón, “Pin: release a=stable” significa que `apt-get` descargará el archivo `Packages.gz` del repositorio que contiene al archivo `Release` con la línea “Archive: stable”.

En el último renglón “Pin-Priority: 500” indica que la Pin-Priority es 500.

Si existen varios paquetes con el mismo nombre, generalmente el que tiene el valor de Pin-Priority más alto es el que se instala..

Los significados de los Pin-Priority son:

- 1001 en adelante: prioridades para volver a la versión anterior.
Se permite volver a la versión anterior del paquete.
- 100 a 1000: prioridades estándar.
No se permite volver a la versión anterior del paquete. Algunos valores de Pin-Priority claves:
 - 990: la prioridad dado por la opción `--target-release` or `-t` of `apt-get(8)` option.
 - 500: la prioridad predeterminada de todos los paquetes.
 - 100: la prioridad de los paquetes recientemente instalados.
- 0 a 99: prioridades que no son automáticas (se utilizan únicamente si el paquete no está instalado y no existe otra versión disponible).
- menor que 0: la versión nunca es seleccionada.

Se puede obtener el mismo efecto que `--target-release` configurando `/etc/apt/apt.conf` de la siguiente manera:

```
# echo 'APT::Default-Release "testing";' >> /etc/apt/apt.conf
```

La opción `--target-release` de la línea de comandos y la configuración de `/etc/apt/apt.conf` tienen prioridad sobre la configuración de `/etc/apt/preferences`. Tenga cuidado con estas interferencias cuando juegue con `/etc/apt/preferences`.

3.3. Comandos de supervivencia de Debian

Con estos conocimientos, se puede vivir **eternamente** de “actualizaciones” :-)

3.3.1. Búsqueda de fallos y ayuda en Debian

Si tiene problemas con un paquete determinado, asegúrese de consultar los siguientes sitios antes de pedir ayuda o enviar un informe de fallos. (`lynx`, `links` y `w3m` funcionan igualmente bien):

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/nombre-paquete # si conoce el nombre del paquete
$ lynx http://bugs.debian.org/número-fallo # si conoce el número de fallo
```

Busque en Google (www.google.com) incluyendo “`site:debian.org`” como palabra clave.

En caso de duda, recurra a la documentación. Inicialice `CDPATH` de la siguiente manera:

```
export CDPATH=./usr/local:/usr/share/doc
```

y escriba

```
$ cd nombre_paquete
$ mc
```

3.3.2. Resolución de problemas al actualizar con APT

Al actualizar desde `unstable/testing` pueden surgir problemas entre dependencias como se explicó en ‘Actualizar el sistema Debian’ en la página 4. La mayoría de las veces, es debido a que el paquete que se desea actualizar posee una nueva dependencia que no es satisfecha. Estos problemas se solucionan haciendo

```
# apt-get dist-upgrade
```

Si esto no funciona, entonces repita las siguientes operaciones hasta que el problema se resuelva por sí solo:

```
# apt-get upgrade -f # continue con upgrade a pesar de los errores
... o
# apt-get dist-upgrade -f # continue con dist-upgrade a pesar de los errores
```

Algunos scripts de actualización conflictivos pueden provocar inconvenientes en forma recurrente. Generalmente resulta conveniente resolver este tipo de situación inspeccionando los scripts `/var/lib/dpkg/info/packagename.{post,pre}{inst,rm}` del paquete problemático y ejecutando:

3.3.4. Recuperar los datos de la selección de paquetes

Si por alguna razón se inutiliza `/var/lib/dpkg/status`, el sistema Debian perderá los datos de la selección de paquetes y resultará enormemente afectado.. Busque el archivo `/var/lib/dpkg/status` anterior en `/var/lib/dpkg/status-old` o `/var/backups/dpkg.status.*`.

Puede resultar una muy buena idea mantener `/var/backups/` en una partición aparte ya que este directorio contiene una inmensa cantidad de datos importantes del sistema.

Si el archivo `/var/lib/dpkg/status` anterior no se encuentra disponible, aún puede recuperar información de los directorios situados `/usr/share/doc/`.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  grep -v '^texmf$' | \
  grep -v '^debian$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
# dselect --expert # reinstale el sistema, use dselect si es necesario
```

3.3.5. Recuperando un sistema luego de borrar /var

Puesto que el directorio `/var` contiene datos que se actualizan continuamente, como por ejemplo el correo, es susceptible de corromperse. Ubicándolo en una partición independiente disminuye los riesgos. Si ocurre un desastre, puede que deba reconstruir el directorio `/var` para recuperar el sistema Debian.

Obtenga la estructura del directorio `/var` de un sistema Debian funcional basada en una versión Debian igual o anterior, por ejemplo `var.tar.gz` (<http://people.debian.org/~osamu/pub/>), y ubíquela en el directorio raíz del sistema dañado. A continuación,

```
# cd /
# mv var var-old      # si quedan contenidos útiles
# tar xvzf var.tar.gz # utilice la estructura Woody
# aptitude           # o dselect
```

Esto proporcionará un sistema funcional. Puede recuperar la selección de paquetes, usando el método descrito en 'Recuperar los datos de la selección de paquetes' en esta página. ([FIXME]: Este procedimiento necesita de más pruebas.)

3.3.6. Instalar un paquete en un sistema que no arranca

Arranque Linux usando un diskete/CD de rescate o una partición alternativa en un sistema multi-arranque. Monte el sistema que no arranca en `/target` y utilice el modo de instalación `chroot` de `dpkg`.

```
# dpkg --root /target -i nombre_paquete.deb
```

Luego configure y solucione los problemas.

A propósito, si es un `lilo` dañado el que le impide arrancar el sistema, puede usar un disco de rescate estándar de Debian. En el indicador de arranque, suponiendo que la partición raíz de su instalación Linux está en `/dev/hda12` y que desea entrar al nivel de ejecución 3, escriba:

```
boot: rescue root=/dev/hda12 3
```

De esta manera arrancará un sistema prácticamente funcional cuyo kernel se encuentra en el diskete (pueden aparecer pequeños problemas debido a la falta de algunas de las características o módulos del kernel)

3.3.7. Qué hacer si el comando `dpkg` se encuentra dañado

Un `dpkg` dañado puede tornar imposible la instalación de cualquier archivo `.deb`. Un procedimiento similar al siguiente le ayudará a superar esta situación (en la primera línea, puede reemplazar “links” con su navegador de consola favorito)

```
$ links http://http.us.debian.org/debian/pool/main/d/dpkg/  
... descargue una versión de dpkg_version_arquit.deb  
$ ar x dpkg_version_arquit.deb  
$ su  
password: *****  
# mv data.tar.gz /data.tar.gz  
# cd /  
# tar xzfv data.tar.gz
```

Para una máquina `i386`, puede utilizar también la URL `http://packages.debian.org/dpkg`.

3.4. Comandos ‘nirvana’ de Debian

Bajo la luz de estos comandos una persona se salvará de la lucha kármica eterna contra el infierno de la actualización y le permitirá alcanzar el **nirvana** Debian. :-)

3.4.1. Información sobre un archivo

Para averiguar a qué paquete pertenece un archivo determinado:

```
$ dpkg {-S|--search} patrón # busca 'patrón' en los paquetes instalados
$ zgrep -e patrón /local/copy/of/debian/woody/Contents-i386.gz
    # busca 'patrón' en los archivos descargados de Debian
```

O utilice comandos de paquetes especializados:

```
# apt-get install dlocate
    # entra en conflicto con slocate (versión segura de locate)
$ dlocate nombre_archivo # alternativa rápida de dpkg -L y dpkg -S
...
# apt-get install auto-apt # herramienta de instalación automática de paquete
    # bajo demanda
# auto-apt update # crea el archivo db para auto-apt
$ auto-apt search patrón
    # busca de 'patrón' en todos los paquetes, instalados o no
```

3.4.2. Información sobre un paquete

Buscar y mostrar la información sobre un paquete. Asegúrese de apuntar a un/unos archivo(s) APT adecuado(s) editando `/etc/apt/sources.list`. Si desea ver las diferencias entre un paquete de la rama de prueba/inestable y uno instalado, utilice `apt-cache policy`.

```
# apt-get check # actualiza el caché y verifica las dependencias
$ apt-cache search patrón # busca un paquete a partir del 'patrón'
$ apt-cache policy paquete # información sobre la prioridad de un
    # paquete
$ apt-cache show -a paquete # muestra la descripción del paquete en
    # todas las distribuciones
$ apt-cache showsrc paquete # muestra la descripción de un paquete
    # fuente
$ apt-cache showpkg paquete # información de depuración de un paquete
# dpkg --audit|-C # busca paquetes parcialmente instalados
$ dpkg {-s|--status} paquete # descripción de un paquete instalado
$ dpkg -l paquete # estado de un paquete instalado (una
    # línea)
$ dpkg -L paquete # lista los archivos instalados por el
    # paquete
```

`apt-cache showsrc` no está documentado en el lanzamiento de Woody pero funciona :)

Asimismo puede encontrar información de paquetes en (utilizo mc para navegar):

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

La comparación de los siguientes archivos proporciona información sobre lo que ocurrió exactamente durante las últimas instalaciones.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

3.4.3. Instalación automática con APT

Para una instalación automática, añada la siguiente línea al `/etc/apt/apt.conf`:

```
Dpkg::Options {"--force-confold";}
```

Esto equivale a ejecutar `apt-get -q -y nombre_paquete`. Puesto que de esta manera se responden con "Sí" todas las preguntas formuladas, puede causar problemas. Por lo tanto use este truco con cuidado. Véase `apt.conf(5)` y `dpkg(1)`.

Posteriormente puede configurar cualquier paquete siguiendo 'Reconfigurar paquetes instalados' en esta página.

3.4.4. Reconfigurar paquetes instalados

Utilice los siguientes comandos para reconfigurar un paquete ya instalado.

```
# dpkg-reconfigure --priority=medium paquete [...]
# dpkg-reconfigure --all # reconfigura todos los paquetes
# dpkg-reconfigure locales # genera los nuevos locales
# dpkg-reconfigure --p=low xserver-xfree86 # reconfigura el
# servidor X
```

Hago lo mismo con `debconf` si necesita cambiar en forma permanente el modo de diálogo de `debconf`.

Algunos programas vienen con scripts de configuración especiales.

```
apt-setup      - crea el /etc/apt/sources.list
install-mbr    - instala un administrador del sector de arranque maestro
tzconfig       - configura el huso horario local
gpmconfig      - configura el administrador del ratón gpm
smbaconfig     - configura Samba en Potato (Woody utiliza debconf)
eximconfig     - configura Exim (MTA)
texconfig      - configura teTeX
apacheconfig   - configura Apache (httpd)
cvsconfig      - configura CVS
```

```
sndconfig      - configura el sistema de sonido
...
update-alternatives - configuración del comando predeterminado, por ej. vim
                  por vi
update-rc.d     - administración de los scripts de arranque System-Vt
update-menus    - sistema de menús de Debian
...
```

3.4.5. Desinstalar y purgar paquetes

Desinstalar un paquete manteniendo su configuración:

```
# apt-get remove paquete ...
# dpkg --remove paquete ...
```

Desinstalar un paquete incluyendo su configuración:

```
# apt-get remove --purge paquete ...
# dpkg --purge paquete ...
```

3.4.6. Conservar paquetes antiguos

Por ejemplo, para conservar `libc6` y `libc6-dev` para `dselect` y `apt-get -u upgrade paquete` se puede hacer:

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`apt-get -u install paquete` no se verá afectado por esto. Para conservar un paquete forzando una actualización automática a una versión anterior mediante `apt-get -u upgrade paquete` o `apt-get -u dist-upgrade`, añade lo siguiente al `/etc/apt/preferences`:

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

En “Package:” no se pueden usar entradas del tipo “`libc6*`”. Si necesita conservar todos los paquetes binarios relacionados con el paquete fuente `glibc` en forma conjunta, debe listarlos en forma explícita.

El siguiente comando lista todos los paquetes que se desean conservar:

```
dpkg --get-selections "*" | grep -e "hold$"
```


3.4.7. Sistema mixto estable/de prueba/inestable

`apt-show-versions` puede mostrar las versiones de los paquetes disponibles en cada distribución.

```
$ apt-show-versions | fgrep /testing | wc
... cantidad de paquetes de la rama de prueba (testing)
$ apt-show-versions -u
... lista de paquetes que se pueden actualizar
$ apt-get install `apt-show-versions -u -b | fgrep /unstable`
... actualiza todos los paquetes de la rama inestable (unstable)
a su versión más reciente
```

3.4.8. Eliminar paquetes del caché

La instalación de paquetes con APT guarda los paquetes en `/var/cache/apt/archives`.

```
# apt-get autoclean # elimina únicamente los paquetes inútiles
# apt-get clean     # elimina la totalidad de los paquetes del caché
```

3.4.9. Grabar/copiar la configuración del sistema

Para hacer una copia local del estado de la selección de los paquetes:

```
$ dpkg --get-selections "*" >mis_selecciones # o usar \*
```

"*" incluye también los nombres de los paquetes a purgar en `mis_selecciones`.

Puede transferir este archivo a otra computadora e instalarlo haciendo:

```
# dselect update
# dpkg --set-selections <mis_selecciones
# apt-get -u dselect-upgrade # o dselect install
```

3.4.10. Portar un paquete a la rama stable

Para una actualización parcial del sistema estable, es conveniente reconstruir el paquete para el entorno dado a partir de su código fuente. Esto evita actualizaciones masivas de paquetes debido a las dependencias. Primero, añade las siguientes entradas al `/etc/apt/sources.list`:

```
deb-src http://http.us.debian.org/debian testing \
  main contrib non-free
deb-src http://non-us.debian.org/debian-non-US testing/non-US \
  main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
  main contrib non-free
deb-src http://non-us.debian.org/debian-non-US unstable/non-US \
  main contrib non-free
```

Cada entrada para `deb-src` está dividida en dos por cuestiones de impresión aunque en realidad las entradas en el `sources.list` deben escribirse en una única línea.

A continuación, obtenga el código fuente y cree una copia local del paquete:

```
$ apt-get update # actualiza la lista de paquetes del repositorio
$ apt-get source paquete
$ dpkg-source -x paquete.dsc
$ cd versión_paquete
... busque los paquetes requeridos (Build-depends en el archivo .dsc) e
  instálelos. Necesitará también el paquete "fakeroot".

$ dpkg-buildpackage -rfakeroot

...o (sin firma)
$ dpkg-buildpackage -rfakeroot -us -uc # utilice luego "debsign" si es necesaria

...luego para instalarlo
$ su -c "dpkg -i nombre_paquete.deb"
```

En general, se necesita instalar algunos paquetes con la opción “-dev” para satisfacer la dependencia entre paquetes. `debsign` pertenece al paquete `devscripts`. `auto-apt` puede fácilmente satisfacer estas dependencias. La utilización de `fakeroot` evita el uso innecesario de la cuenta de superusuario.

En Woody, estos temas sobre dependencias pueden simplificarse. Por ejemplo, para compilar el paquete fuente `pine`:

```
# apt-get build-dep pine
# apt-get source -b pine
```

3.4.11. Archivo local de paquetes

A fin de crear un archivo local de paquetes que sea compatible con los sistemas APT y `dselect`, es necesario crear el archivo `Packages` y archivar los paquetes en un árbol de directorios determinado.

Se puede crear un repositorio local deb similar al oficial de Debian haciendo:

```

# apt-get install dpkg-dev
# cd /usr/local
# install -d pool # los paquetes se ubicarán físicamente aquí
# install -d dists/unstable/main/binary-i386
# ls -l pool | sed 's/_.*$/ extra BOGUS/' | uniq > override
# editor override # ajuste BOGUS
# dpkg-scanpackages pool override /usr/local/ \
    > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
    >> /etc/apt/sources.list

```

O bien, rápidamente haciendo:

```

# apt-get install dpkg-dev
# mkdir /usr/local/debian
# mv /some/where/package.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
    gzip - > /usr/local/debian/Packages.gz
# echo "deb file:/usr/local/debian ." >> /etc/apt/sources.list

```

Se puede acceder en forma remota a estos archivos habilitando el acceso a estos directorios mediante HTTP o FTP y modificando convenientemente las entradas del archivo `/etc/apt/sources.list`.

3.4.12. Convertir o instalar un paquete que no pertenece a Debian

`alien` permite la conversión de paquetes `rpm` de Redhat, `slp` de Stampede, `tgz` de Slackware y `pkg` de Solaris al formato de paquetes `deb` de Debian. Si desea utilizar un paquete de una distribución Linux diferente a la que tiene instalada en su sistema, puede usar `alien` para convertirlo en su formato de paquetes preferido e instalarlo. `alien` soporta también paquetes LSB.

3.4.13. Verificar los paquetes instalados

`debsums` permite verificar la integridad de los paquetes instalados usando las sumas de control (checksums) MD5. Algunos paquetes no poseen sumas de control MD5. Veamos una solución provisoria para los administradores de sistema:

```
# cat »/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs { "xargs /usr/bin/debsums -sg" ; };
^D
```

por Joerg Wendland <joergland@debian.org> (sin verificar).

3.5. Otras particularidades de Debian

3.5.1. El comando `dpkg-divert`

`dpkg-divert` fuerza a `dpkg` a no instalar un archivo en su lugar habitual sino en otro diferente. Estos **desvíos** se pueden usar en los scripts de los paquetes Debian para quitar un archivo que provoca un conflicto. Los administradores de sistemas también pueden usarlo para no tener en cuenta algunos archivos de configuración o algunos archivos (que no han sido marcados como tales) necesitan preservarse de `dpkg` al instalar una versión nueva de un paquete que los incluya.

```
# dpkg-divert [--add] nombre_archivo # añade "diversion"
# dpkg-divert --remove nombre_archivo # elimina "diversion"
```

En general, resulta una buena idea no usar `dpkg-divert` salvo que sea absolutamente necesario hacerlo.

3.5.2. El paquete `equivs`

Si compila un programa a partir de su código fuente, resulta conveniente crear un paquete Debian (*.deb). Utilice `equivs` como último recurso.

```
Package: equivs
Priority: extra
Section: admin
Description: burlar las dependencias entre paquetes
Este es un paquete ficticio que se puede usar para
crear paquetes Debian que contienen únicamente información
sobre dependencias.
```

3.5.3. Comandos alternativos

Para utilizar el comando `vim` en lugar de `vi`, utilice `update-alternatives`:

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection      Command
-----
          1      /usr/bin/elvis-tiny
          2      /usr/bin/vim
*+       3      /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Los miembros del sistema de alternativos de Debian se encuentran en `/etc/alternatives` como enlaces simbólicos.

Para tener su entorno de ventanas X favorito, aplique `update-alternatives` a `/usr/bin/x-session-manager` y `/usr/bin/x-window-manager`.

`/bin/sh` es un enlace simbólico a `/bin/bash` o `/bin/dash`. Resulta más seguro utilizar `/bin/bash` para preservar la compatibilidad con scripts antiguos contaminados con comandos del `bash` aunque resulta más disciplinado usar `/bin/dash` para asegurar compatibilidad POSIX. La actualización al kernel 2.4 Linux kernel tends to set this to `/bin/dash`.

3.5.4. `init` del System-V y niveles de ejecución

El nivel de ejecución predeterminado se puede configurar en el `/etc/inittab`.

A diferencia de otras distribuciones, Debian deja la administración de los niveles de ejecución bajo la completa responsabilidad del administrador de sistemas. La administración de `init` en un sistema System-V es llevada a cabo mediante los scripts `update-rc.d`.

Arrancar `/etc/init.d/nombre` en los niveles de ejecución 1,2,3 y detenerlo en 4,5 con una prioridad igual a 20 (normal) se puede realizar haciendo:

```
# update-rc.d nombre start 20 1 2 3 . stop 20 4 5 .
```

Si el script existe en `init.d`, el enlace simbólico se puede eliminar haciendo:

```
# update-rc.d -f nombre remove
```

Para editar los niveles de ejecución, hago trampa. Edito manualmente las entradas usando el comando `mv` en el indicador del `mc` y copio los enlaces con `Alt-Enter`. Por ejemplo:

```
# mv S99xdm K99xdm # desactivo xdm (X display manager)
```

Incluso, como atajo rápido, desactivo un demonio escribiendo `exit 0` al comienzo de un script del `init.d`. Después de todo son archivos de configuración.

3.5.5. Servicios desactivados

La distribución Debian toma muy en serio la seguridad del sistema y espera que el administrador del sistema sea competente. Por lo tanto, la facilidad de uso a menudo se relega al segundo plano y determinados servicios brindados por los demonios vienen con el nivel de seguridad más alto y pocos servicios (o ninguno) están disponibles en su estado de instalación predeterminado.

Ejecute `ps aux` o consulte el contenido de `/etc/init.d/*` y `/etc/inetd.conf`, si posee alguna duda (sobre Exim, DHCP, ...). También consulte el `/etc/hosts.deny`. Asimismo resulta útil el comando `pidof` (véase `pidof(8)`).

En versiones recientes de Debian X11 no permite, por defecto, conexiones TCP/IP (remotas). El redireccionamiento de X también se encuentra deshabilitado en SSH.

Apéndice A

Apéndice

A.1. Autores

La Guía de referencia rápida Debian fue iniciada por Osamu Aoki <osamu@debian.org> a partir del resumen de su instalación personal que inicialmente denominó “Referencia Rápida...” (“Quick Reference ...”). Muchos de los contenidos provienen de los archivos de la lista de correo “debian-user”. También fueron referenciados “Manual de Instalación Debian ” y “Debian Release Notes”.

Siguiendo las sugerencias de Josip Rodin, un miembro muy participativo del Proyecto de Documentación Debian (<http://www.debian.org/doc/ddp>) (DDP) y el encargado actual de las “FAQ de Debian”, el documento fue renombrado como “Referencia Debian” y se nutrió con diversos capítulos de las “Debian FAQ”. Luego se creó el resumen “Guía de referencia rápida Debian”.

Este documento fue editado, traducido y ampliado por los siguientes miembros del equipo QREF:

- Inglés original de la “Quick Reference...”
 - Osamu Aoki <osamu@debian.org> (coordinador: de todos los contenidos)
- Relectura del original en inglés y contribuciones adicionales
 - David Sewell <dsewell@virginia.edu> (coordinador: en estilo)
 - Thomas Hood <jdthood@yahoo.co.uk> (temas relacionados con redes)
 - Brian Nelson <nelson@bignachos.com> (en especial, temas relacionados con X)
 - Jan Michael C Alonzo <jmalonzo@spaceants.net>
 - Daniel Webb <webb@robust.colorado.edu>
 - Intercambio de opiniones con todos los traductores
- Traducción al francés
 - Guillaume Erbs <gerbs@free.fr> (coordinador: fr)
 - Régnald Casagraude <rcasagraude@interfaces.fr>
 - Jean-Pierre Delange <delange@imagnet.fr>
 - Daniel Desages <daniel@desages.com>
- Traducción al italiano
 - Davide Di Lazzaro <mc0315@mcLink.it> (coordinador: it)

- Traducción al portugués brasileño
 - Paulo Rogério Ormenese <pormenese@uol.com.br> (coordinador: pt-br)
 - Andre Luis Lopes <andreloup@ig.com.br>
 - Marcio Roberto Teixeira <marciotex@pop.com.br>
 - Rildo Taveira de Oliveira <to_rei@yahoo.com>
 - Raphael Bittencourt Simoes Costa <raphael-bsc@bol.com.br>
 - Gustavo Noronha Silva <kov@debian.org> (coordinador)
- Traducción al español
 - Walter Echarri <wecharri@infovia.com.ar> (coordinador: es)
 - José Carreiro <ffx@urbanet.ch>
- Traducción al alemán
 - Jens Seidel <tux-master@web.de> (coordinador: de)
 - Willi Dyck <wdyck@gmx.net>
 - Stefan Schröder <stefan@fkip.uni-hannover.de>
 - Agon S. Buchholz <asb@kefk.net>
- Traducción al polaco—los siguientes miembros de PDDP (<http://debian.linux.org.pl>):
 - Marcin Andruszkiewicz
 - Mariusz Centka <mariusz.centka@debian.linux.org.pl>
 - Bartosz Feński <fenio@debian.linux.org.pl> (coordinador: pl)
 - Radosław Grzanka <radekg@debian.linux.org.pl>
 - Bartosz 'Xebord' Janowski
 - Jacek Lachowicz
 - Rafał Michaluk
 - Leonard Milcin, Jr.
 - Tomasz Z. Napierała <zen@debian.linux.org.pl>
 - Oskar Ostafin <cx@debian.linux.org.pl>
 - Tomasz Piękoś
 - Jacek Politowski
 - Mateusz Prichacz <mateusz@debian.linux.org.pl>
 - Marcin Rogowski
 - Paweł Różański
 - Mariusz Strzelecki
 - Krzysztof Ścierański
 - Przemysław Adam Śmiejek <tristan@debian.linux.org.pl>
 - Mateusz Tryka <uszek@debian.linux.org.pl>
 - Cezary Uchto
 - Krzysztof Witkowski <tjup@debian.linux.org.pl>
 - Bartosz Zapalowski <zapal@debian.linux.org.pl>
- Traducción al chino (simplificado)
 - Hao "Lyoo" LIU <iamlyoo@163.net> (coordinador: zh-cn)
 - Ming Hua <minghua@rice.edu>
- Traducción al chino (tradicional)
 - Tang Wei Ching <wctang@csie.nctu.edu.tw> (coordinador: zh-tw)
- Traducción al japonés
 - Shinichi Tsunoda <tsuno@ngy.1st.ne.jp> (coordinador: ja)

A.2. Garantías

Puesto que no soy un experto, no pretendo ser un entendido en Debian o Linux en general. Las consideraciones sobre seguridad quizás sean aplicables únicamente al ámbito doméstico.

El presente documento no reemplaza a ninguna de las guías autorizadas.

No se ofrecen garantías de ningún tipo. Todas las marcas son propiedad de sus respectivos dueños.

A.3. Comentarios

Son bienvenidos todos los comentarios y sugerencias. Por favor, envíen un mensaje a Debian BTS system (<http://bugs.debian.org/>) para el paquete `debian-reference` o los respectivos paquetes traducidos. El uso de `reportbug` facilita el envío de un reporte de fallos. Incluso pueden enviar un mail para Osamu Aoki (<http://people.debian.org/~osamu/>) a `<osamu@debian.org>` en inglés o a cada traductor en su respectivo idioma)