



Curso Operador Linux

Módulo 8

ADMINISTRACION DE SOFTWARE



Presentación

En esta unidad se introducirá a los participantes en administración, instalación, upgrade y desinstalación de software



Objetivos

Los participantes al finalizar la Unidad:

- Podrán obtener e instalar software
- Conocerán las diferencias y ventajas-desventajas de trabajar con binarios y fuentes
- Manejarán instalación de software por consola y en forma grafica
- Comprenderán el proceso de instalación a partir del código fuente de un programa



Temario

- 8.1 Instalación y upgrade de aplicaciones
- 8.2 Obtención e instalación de software
- 8.3 Uso de rpm, yast2 y apt
- 8.4 Instalación de aplicaciones desde el código fuente



Actividad de aplicación del conocimiento

Los participantes encontraran la actividad de aplicación en un archivo por separado en caso de ser requerida.



Examen

Los participantes deberán rendir el examen online o presentar el material solicitado según corresponda.



8.1 Instalación y upgrade de aplicaciones

En el principio los programas eran distribuidos por código fuente mediante archivos .tar.gz. Los usuarios tenían que compilar cada programa que quisieran usar en su sistema GNU/Linux. Cuando Debian fue creado, fue imperante que el sistema incluyera un programa que se encargara de manejar los paquetes o programas.

Este programa se llamó dpkg. Así fue como nació el primer “paquete” en el mundo GNU/Linux, aún antes de que Red Hat decidiera crear su propio programa denominado “rpm”.

Rápidamente llegó un nuevo dilema a las mentes de los creadores de GNU/Linux. Ellos necesitaban un modo fácil, rápido y eficiente de instalar programas, que manejara automáticamente las dependencias y se hiciera cargo de la configuración mientras se actualizan. Debian creó el APT, Herramienta Avanzada de Empaquetamiento (Advanced Packaging Tool) y ha sido adoptado por otras distribuciones.



**Existen diversos tipos de paquetes.
Por ejemplo tenemos los “.deb” para distribuciones
basadas en Debian y los “.rpm” para las basadas en
Red Hat.**

Con los paquetes deb tenemos la ventaja de poder instalar o actualizar un programa y que siga funcionando como lo hacía la versión anterior, sin problemas de dependencias y sin necesidad de reiniciar el equipo.

Para realizar la instalación de un paquete deb podemos utilizar dpkg. El comando dpkg instala el paquete deb que deseamos y resuelve los problemas de dependencias. En caso de existir problemas nos indicara que dependencias debemos también instalar. El problema que se puede mencionar es que no tiene acceso a Internet, por lo que nosotros deberemos proveerle los paquetes necesarios.

Para instalar un paquete deberemos entonces descargarlo de Internet y pasárselo a dpkg de la siguiente manera:

```
# dpkg -i kernel-image-2.6.27.7_KernelRebuild.01_i386.deb
```



Para instalar el paquete utilizamos lo siguiente:
dpkg -i

Para desempaquetar un paquete:
dpkg --unpack

Para reconfigurar un paquete previamente desempaquetado:
dpkg --configure

Para desinstalar o remover un paquete previamente instalado:
dpkg -r

Existen una serie de herramientas que completan las funcionalidades para trabajar con paquetes de las cuales podemos mencionar:

dpkg-source	Empaqueta y desempaqueta los archivos fuentes de un paquete Debian
dpkg-gencontrol	Lee la información de un árbol fuente Debian desempaquetado y genera un paquete binario de control, generando una entrada para éste en el fichero debian/files
dpkg-shlibdeps	calcula las dependencias de ejecutables respecto a librerías
dpkg-genchanges	lee la información de un árbol fuente Debian desempaquetado y ya construido, generando un fichero de control de los últimos cambios (un.changes)
dpkg-buildpackage	es un script de control que se puede utilizar para automatizar la construcción del paquete
dpkg-distaddfile	añade una entrada de un fichero a debian/files
dpkg-parsechangelog	lee el fichero de cambios changelog) de un árbol fuente Debian desempaquetado y genera una salida con la información de estos cambios, convenientemente preparada



8.2 Obtención e instalación de software

El archivo `/etc/apt/sources.list`

Como parte de su funcionamiento, APT utiliza un archivo que lista las “fuentes” en donde se encuentran los paquetes. Este archivo es: `/etc/apt/sources.list`

El contenido de este archivo, normalmente sigue este formato:

```
deb http://host/debian distrib sección1 sección2 sección3
deb-src http://host/debian distrib sección1 sección2 sección3
```

La primera palabra en cada línea, `deb` o `deb-src`, indican el tipo del archivo: ya sea que contenga paquetes binarios (`deb`), esto es, los paquetes pre-compilados que normalmente se usan, o los fuentes (`deb-src`), que son los códigos originales, más el archivo de control de Debian (`.dsc`) y el `diff.gz` que contienen los cambios necesarios para “debianizar” el programa

Generalmente se encuentra lo siguiente por defecto en el archivo `sources.list`:

```
# See sources.list(5) for more information, specially
# 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 non-free
```

Estas son las líneas necesarias para una instalación básica de Debian.

La primera línea `deb` apunta al archivo en el servidor oficial, la segunda línea apunta hacia el archivo del servidor non-US y la tercera línea apunta hacia el archivo del servidor de actualizaciones de seguridad

Las dos últimas líneas están deshabilitadas (con un “#” al inicio), así que `apt-get` las ignora.

Estas son las líneas de `deb-src`, esto es, apuntan hacia los paquetes fuente de Debian. Si frecuentemente descarga paquetes fuente para probar o recompilar, habilítelas (borrando el “#” al inicio de la línea).



El archivo `/etc/apt/sources.list` puede contener varios tipos de líneas. APT sabe como interpretar líneas del tipo `http`, `ftp`, `file` (archivos locales, por ejemplo, un directorio que contiene un CD-ROM) y `ssh`



No olvidar ejecutar el comando `apt-get update` después de modificar el archivo `/etc/apt/sources.list`.

Se debe hacer esto para permitir a APT obtener la lista de paquetes de las fuentes que especificó. Se trata de sincronizar el equipo con el site o mirror especificado en el `sources.list`.

Instalando paquetes

Finalmente, con su archivo `sources.list` listo y su lista de paquetes disponibles al día, todo lo que necesita es ejecutar `apt-get` para tener el paquete que quiera instalado.

Por ejemplo, puede ejecutar:

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

APT buscará en su base de datos para encontrar la versión más reciente del paquete y lo descargará del servidor correspondiente especificado en `sources.list`. Si este paquete necesitara otro para funcionar, como en este caso, APT resolverá las dependencias e instalará los paquetes necesarios. Por ejemplo:

```
# 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 paquete nautilus necesita las librerías compartidas mencionadas, así pues APT las descargará del servidor. Si se especifican antes los nombres de esas librerías con el comando apt-get APT no pregunta, si desea continuar o no; supone automáticamente que se desean instalar esos paquetes.

Esto significa que APT sólo pregunta por confirmación cuando se van a instalar paquetes que no fueron especificados en la línea de comando.

Las siguientes opciones de apt-get podrían ser útiles:

- h This help text.
- d Download only - do NOT install or unpack archives
- f Attempt to continue if the integrity check fails
- s No-act. Perform ordering simulation
- y Assume Yes to all queries and do not prompt
- u Show a list of upgraded packages as well



Pueden seleccionarse varios paquetes para instalar en una sola línea.

Los archivos descargados son almacenados en el directorio /var/cache/apt/archives para su instalación posterior.

Puede especificar también que paquetes serán eliminados en la misma línea. Sólo agregue un "-" inmediatamente después del nombre del paquete que quiere eliminar, por ejemplo:

```
# 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]
```



Eliminando paquetes

Si ya no necesita utilizar cierto paquete, puede eliminarlo de su sistema utilizando APT. Para realizar esta tarea sólo escriba: `apt-get remove paquete`.

```
# 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]
```

Como se puede apreciar en el ejemplo anterior, APT se hace cargo de eliminar los paquetes dependientes del paquete eliminado.

Ejecutando `apt-get` como en el ejemplo causará que los paquetes sean eliminados, pero sus archivos de configuración, si existían, permanecerán intactos en el sistema. Para una eliminación completa del paquete, ejecute:

```
# 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]
```



Observe el “*” después de los nombres. Indican que los archivos de configuración de cada paquete serán eliminados.



Al igual que en el caso de la instalación, puede utilizar un símbolo con la opción de “remove” para invertir el significado de un paquete en especial. En el caso de la eliminación, si agrega un “+” después del nombre del paquete, el paquete será instalado en vez de eliminarlo

```
# 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]
```

Observe que apt-get lista los paquetes que serán instalados aparte de los seleccionados (esto es, aquellos que su instalación es necesaria para el funcionamiento de aquellos que son seleccionados), aquellos que serán eliminados, y aquellos que serán instalados (incluyendo los paquetes adicionales otra vez).

Búsqueda de información de paquetes

Una vez ejecutado el comando apt-get update, la base de datos de apt se encuentra actualizada. Podremos luego realizar búsquedas de paquetes y pedir información de los mismos mediante las opciones que veremos a continuación.

```
# apt-cache search apt-get
apt - Advanced front-end for dpkg
apt-build - Frontend to apt to build, optimize and install packages
apt-move - Maintain Debian packages in a package pool
auto-apt - package search by file and on-demand package installation tool
cron-apt - Automatic update of packages using apt
#
```



Podemos obtener información extra de la siguiente manera:

```
# apt-cache show libgphoto2-2
Package: libgphoto2-2
Priority: optional
Section: libs
Installed-Size: 2976
Maintainer: Frederic Peters <email address hidden>
Architecture: i386
Source: libgphoto2
Version: 2.1.5-
...
Size: 823652
MD5Sum: b69e0a3667108525758d658c913863fc
Description: The gphoto2 digital camera library
#
```

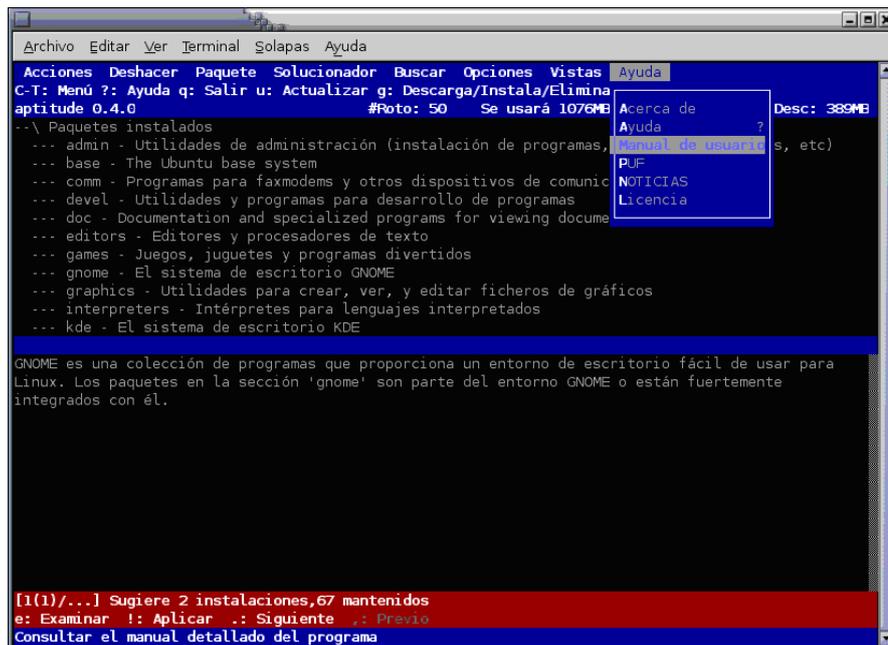
8.3 Uso de rpm, yast2 y apt

Ya vimos como trabajar con paquetes deb mediante dpkg y apt-get. Ahora veremos un par de herramientas graficas con las cuales podremos realizar las mismas funciones

aptitude

aptitude es una interfaz grafica para APT. Al iniciar el programa nos mostrara una lista de paquetes y nos permitirá en forma interactiva cuales instalar o remover.

Posee un sistema de búsqueda con patrones de búsqueda simple y flexible, esto facilita al usuario entender las complejas relaciones de dependencia entre paquetes. aptitude se basa en una biblioteca ncurses por lo cual tiene una interfaz con algunos elementos comunes a otras interfaces gráficas, como son los menús desplegables



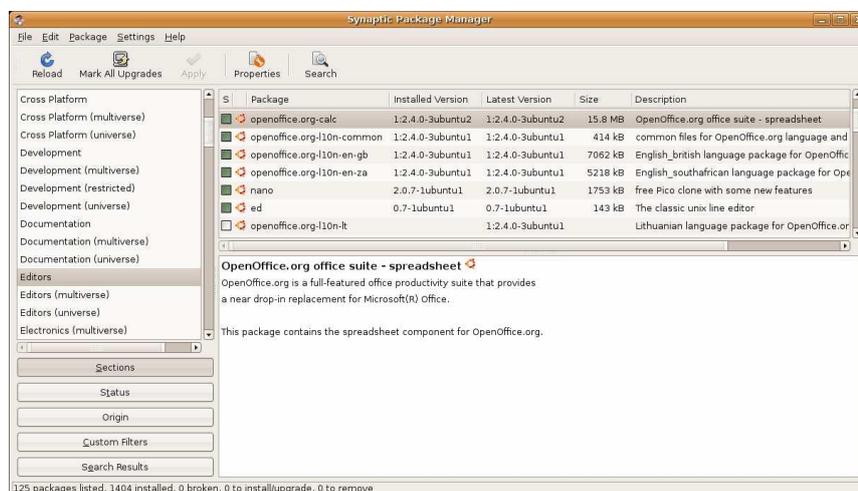
Synaptic

Synaptic es la segunda herramienta gráfica que veremos. Es un programa de interfaz gráfica GTK+ para APT. Generalmente se utiliza Synaptic para sistemas basados en paquetes .deb pero también puede ser usado en sistemas basados en paquetes RPM.

Synaptic permite su gestión mediante un menú interactivo. Permite en forma simple actualizar la lista de software disponible, seleccionar (marcar) los paquetes a instalar o bien desinstalar.

Posee un sistema avanzado de búsqueda y filtros para poder encontrar fácilmente el software que deseemos. También tiene la capacidad de de reparar dependencias rotas de paquetes y permite deshacer y rehacer las últimas selecciones de paquetes

Podremos también personalizar la interfaz gráfica mediante el menú de configuración





YaST

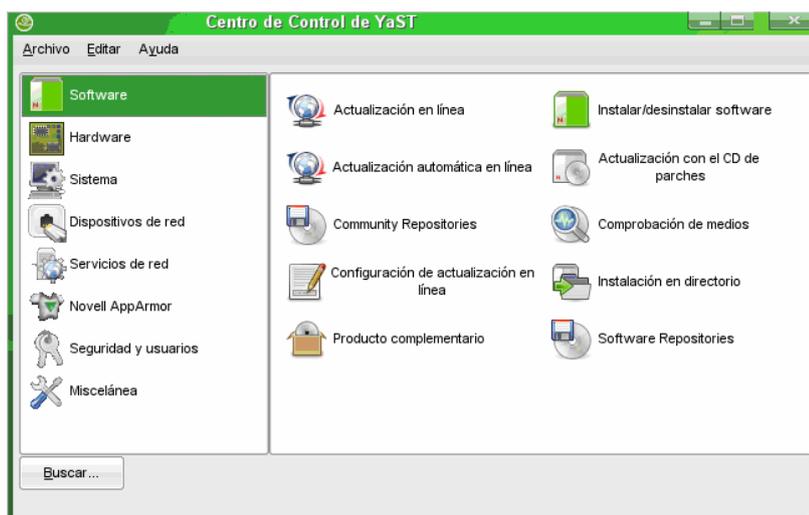
YaST (Yet another Setup Tool) es una aplicación incluida en la distribución de Linux SUSE. Esta herramienta facilita la administración del sistema y entre otras cosas la instalación de software. Se encuentra entre las herramientas más potentes y de uso más sencillo para la administración de sistemas Linux

YaST dispone de una interfaz gráfica hecha con Qt (llamada YaST2) y de una interfaz en ncurses que puede usarse en modo texto. Además se dispone de una interfaz en línea de comandos al sistema de gestión de software de YaST llamada zypper.

En sus inicios, YaST era de código cerrado pero luego fue liberado bajo la licencia GPL de GNU.

Entre sus funciones, se encuentran:

- Administración de servicios del superservidor (inetd e xinetd)
- Administración del servidor web apache
- Configuración del servidor de correo postfix
- Gestión de usuarios y grupos
- Políticas de seguridad
- Instalar/desinstalar software
- Configuración de hardware genérico (tarjetas de sonido, ratones, joysticks, tarjetas capturadoras de video, gestión de discos, impresoras, escáners, Bluetooth, tarjetas de red, WiFi...)
- Generar discos de arranque
- Cargar discos de controladores del fabricante (lee la mayoría de ficheros .inf de windows)
- Gestión avanzada del sistema (sysconfig)





Paquetes RPM

El nombre de paquetes RPM viene de Redhat Package Manager, es la herramienta de administración de paquetes para las distribuciones derivadas de Red Hat.

Dentro de un paquete “.rpm” se encuentra la información necesaria para que el programa de instalación conozca todo sobre él. La versión, los archivos y directorios que contiene y donde se instalarán, descripción y la lista de dependencias.

Si bien el archivo rpm contiene información de las dependencias, el programa RPM no las maneja (esto si vemos que lo hace APT). Por esto si falla la instalación por dependencias, deberemos instalar los paquetes necesarios y volver a instalar luego nuestra aplicación.

Para instalar un paquete haremos lo siguiente

```
# rpm -vhi paquete.rpm
```

Con esto instalaremos el paquete llamado paquete.rpm. En caso de encontrar el software ya instalado no hará nada. En caso de haber dependencias no resueltas tendremos un mensaje como el siguiente

```
# rpm -vhi paquete2.rpm  
failed dependencies:  
paquete1 is needed by paquete2
```

Para solucionar esto deberemos instalar los paquetes requeridos. Podemos forzar la instalación, eso no es recomendable dado que el software no funcionara correctamente, para esto utilizamos la opción –nodeps

Tenemos la posibilidad de realizar la actualización mediante:

```
# rpm -Uvh paquete.rpm
```

Mediante esta opción se actualiza o instala el paquete. Internamente lo que se realiza es la desinstalación de la versión anterior y la instalación de la nueva versión.

También actualiza los archivos de configuración realizando un backup de la configuración previa.



Para consultar la base de datos de paquetes instalados utilizaremos la opción siguiente:

```
# rpm -q paquete1  
paquete-7.1.2-4
```

Se visualizara el nombre y la versión del paquete instalado. Para visualizar la información de todos los paquetes instalados en el sistema podemos utilizar la combinación `-qa`.

Para realizar la desinstalación de un paquete:

```
# rpm -e paquete1
```

8.4 Instalación de aplicaciones desde el código fuente

Si deseamos no utilizar paquetes precompilados, o estos no están disponibles, podremos optar instalar desde el código fuente.

Primero debemos descargar de Internet, o de donde tengamos acceso, el código fuente que generalmente se nos proveerá en formato `.tar.gz`.

Para bajar el código fuente desde Internet podemos utilizar el comando `wget` con la dirección URL del archivo.

```
wget ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/openssh-5.1.tar.gz
```

A continuación debemos descompactar y descomprimir el archivo, para esto hacemos lo siguiente

```
tar xvzf openssh-5.1.tar.gz
```

Leemos el **README** o **INSTALL**, normalmente los encontraremos en el directorio que descomprimimos.

En líneas generales, los 3 pasos siguientes son comunes, aunque puede que alguno de ellos no sea necesario.

Corremos el script de configuración del programa

```
./configure
```



Este se encargara de buscar las librerías y demás dependencias para luego poder compilar el código. Si deseamos direccional el resultado a un cierto directorio lo haremos mediante el siguiente parámetro **--prefix=/directorio**

Compilamos el programa:

make

Con este comando realizaremos la compilación del programa. Dado que usualmente el código fuente esta programado en C, se vera la lista de archivos .c (programas en C) y .h (headers). Si tenemos problemas en esta instancia el programa no se compilara.

A continuación instalamos la aplicación:

make install

Con este comando realizaremos la instalación del programa que acabamos de compilar pudiendo a continuación ejecutarlo desde el sistema como lo hacemos con cualquier otro programa