
Unidad 5: Paquetes de software

**Curso de Administración de Servidores GNU/Linux
Centro de Formación Permanente
Universidad de Sevilla**

Autor: Jorge Juan <jjchico@dte.us.es> 2013-2014

Usted es libre de copiar, distribuir y comunicar públicamente la obra y de hacer obras derivadas bajo las condiciones de la licencia Attribution-Share alike de Creative Commons.

Puede consultar el texto completo de la licencia en <http://creativecommons.org/licenses/by-sa/3.0/>

Contenidos

- **Distribuciones**
- Repositorios de paquetes
- Gestión de paquetes
- Gestión a alto nivel: sistema APT
- Reconfiguración de paquetes
- Gestión a bajo nivel: dpkg
- Paquetes fuente
- Resumen

Métodos de instalación

- Programas binarios
 - problemas de compatibilidad
 - resolución manual de dependencias
 - actualizaciones manuales
 - propio de programas privativos
- Código fuente
 - necesario entorno de desarrollo y bibliotecas
 - resolución manual de dependencias
 - actualizaciones manuales
- Paquetes de la distribución
 - fácil, rápido, automático, centralizado ...
 - actualizaciones automáticas

3

Ciclo de desarrollo de una distribución



4

Ciclo de desarrollo

- Versión estable/oficial
 - válida para sistemas en producción
 - actualizaciones sólo por problemas de seguridad
 - soporte oficial: actualizaciones de seguridad
- Versión en pruebas (beta)
 - antesala de la próxima versión estable
 - actualizaciones de seguridad y ligeras mejoras
 - no tiene soporte oficial
- Versión de desarrollo (alpha)
 - campo de pruebas de próximas versiones
 - actualización constante de paquetes
 - no tiene soporte oficial

5

Dependencias

- Cada programa suele emplear funciones de varios programas y/o bibliotecas
- Esto genera un enorme número de dependencias entre paquetes
 - Programas/bibliotecas necesarios o recomendados
 - Versiones mínimas requeridas
 - Dependencias sucesivas
- La resolución automática de dependencias es una de las tareas más importantes en una distribución
- Ejemplo:
 - Ubuntu 9.10: ~26000 paquetes, ~190000 dependencias
 - Ubuntu 14.04: ~69000 paquetes, ~537000 dependencias

6

Compatibilidad entre distribuciones

- La compatibilidad entre distribuciones es pobre
 - Pocas veces un paquete de una distribución funcionará en otra distribución, incluso en una versión posterior de la misma
- Motivos
 - Gran número de dependencias: modularidad
 - Incompatibilidad de formatos de dependencias
 - Diversidad de criterios de empaquetado
 - Dependencia de versiones concretas de bibliotecas
 - Énfasis en compatibilidad del fuente en vez del binario

7

Contenidos

- Distribuciones
- **Repositorios de paquetes**
- Gestión de paquetes
- Gestión a alto nivel: sistema APT
- Reconfiguración de paquetes
- Gestión a bajo nivel: dpkg
- Paquetes fuente
- Resumen

8

Repositorios de paquetes

- Repositorio: lugar físico donde se encuentran paquetes software de la distribución (http, ftp, CD, DVD, ...)
- En un repositorio puede haber varias **versiones** de una distribución.
 - Ej. repositorio de Ubuntu
 - versiones soportadas anteriores
 - versión actual
 - versión de desarrollo
 - Ej. repositorio de Debian
 - versión actual (stable)
 - versión en pruebas (testing)
 - versión en desarrollo (unstable)

9

Repositorios de paquetes

- Cada versión de la distribución suele tener varios **componentes** por motivos diversos
- Ej. Ubuntu
 - **main**: sección principal, libre y con soporte oficial
 - **restricted**: software necesario no libre (distribuable)
 - **universe**: software libre adicional, no soportado oficialmente
 - **multiverse**: software no soportado oficialmente con posibles problemas de distribución
- Ej. Debian
 - main: sección principal y oficial. Sólo software libre
 - non-free: software no libre (distribuable)
 - contrib: software libre que depende de paquetes en non-free.

10

Repositorios de paquetes

- Pueden combinarse cualquier número de repositorios, siempre que no existan conflictos entre los paquetes que los componen
- El sistema de gestión de paquetes elige la versión más moderna en caso de paquetes repetidos
- Ejemplo repositorios Ubuntu
 - trusty: repositorio principal oficial
 - trusty-security: actualizaciones de seguridad oficiales
 - trusty-updates: actualizaciones recomendadas
 - trusty-proposed: otras actualizaciones

11

Contenidos

- Distribuciones
- Repositorios de paquetes
- **Gestión de paquetes**
- Gestión a alto nivel: sistema APT
- Reconfiguración de paquetes
- Gestión a bajo nivel: dpkg
- Paquetes fuente
- Resumen

12

Synaptic

- Permite hacer todo tipo de operaciones
 - Actualizar
 - Actualizar la distribución
 - Búsquedas simples y avanzadas
 - Reinstalación de paquetes
 - Gestión de repositorios
 - ...

15

Utilidades apt-*

- Configuración del proxy para comandos apt-*
- /etc/apt/apt.conf

```
Acquire::http::Proxy "http://150.214.142.101:8080/";  
Acquire::ftp::Proxy "http://150.214.142.101:8080/";
```

- Configuración de repositorios
 - /etc/apt/sources.list, /etc/sources.list.d

```
...  
## Ubuntu  
deb http://security.ubuntu.com/ubuntu/ gutsy-security main restricted universe multiverse  
deb http://archive.ubuntu.com/ubuntu/ gutsy main restricted universe multiverse  
deb http://archive.ubuntu.com/ubuntu/ gutsy-updates main restricted universe multiverse  
deb-src http://archive.ubuntu.com/ubuntu/ gutsy main restricted universe multiverse  
  
## Ubuntu updates  
deb http://archive.ubuntu.com/ubuntu/ gutsy-backports main restricted universe multiverse  
...
```

Utilidades apt-*

- apt-get
 - apt-get update
 - apt-get install <paquetes>
 - apt-get --reinstall install <paquetes>
 - apt-get remove <paquetes>
 - apt-get purge <paquetes>
- apt-cache
 - apt-cache search <expresion>
 - apt-cache show <paquete>
 - apt-cache stats

17

aptitude

- Programa interactivo de terminal
- Potente, pero no demasiado intuitivo
- Puede usarse como comando (similar a apt-get)
- Ejemplos
 - aptitude install <paquetes>
 - aptitude reinstall <paquetes>
 - aptitude purge <paquetes>
 - aptitude search <expresion>
 - aptitude search ~dwww~dproxy
 - aptitude

18

Actualizar la distribución. Desde el escritorio

- Se actualizan todos los componentes de la distribución
 - kernel
 - configuración
 - todos los paquetes de software
- Proceso automático
 - Mediante gestor de actualizaciones: avisa cuando hay una nueva versión disponible
 - Actualizar a una versión de desarrollo:
 - `$ sudo update-manager -d`

19

Actualizar la distribución. General Debian/Ubuntu

- Proceso manual
 - Cambiar repositorios a nueva versión o distribución
 - synaptic: Configuración -> Repositorios
 - apt-get: `/etc/apt/sources.list`
 - Actualizar lista de paquetes
 - synaptic: Recargar
 - apt-get `update`
 - Actualizar todos los paquetes
 - synaptic: Marcar todas las actualizaciones (inteligente) + Aplicar
 - apt-get `dist-upgrade`
- Automático (Ubuntu)
 - `do-release-upgrade`

20

Reconfiguración de paquetes. Debconf

- Muchos paquetes Debian/Ubuntu disponen de asistentes para su configuración inicial
- El asistente se ejecuta la primera vez que se instala el paquete y en cualquier otro momento con el comando “dpkg-reconfigure”
- Si el paquete no posee asistente, dpkg-reconfigure simplemente vuelve a configurar el paquete
- Ejemplos

```
# dpkg-reconfigure synaptic
...
# dpkg-reconfigure debconf
...
# dpkg-reconfigure -p low postfix
...
# dpkg-reconfigure console-setup
...
```

21

Contenidos

- Distribuciones
- Repositorios de paquetes
- Gestión de paquetes
- Gestión a alto nivel: sistema APT
- Reconfiguración de paquetes
- **Gestión a bajo nivel: dpkg**
- Paquetes fuente
- Resumen

22

Gestión a bajo nivel: dpkg

- dpkg-*: familia de comandos que actúan directamente sobre archivos de paquetes (.deb)
- Casos de uso:
 - instalar paquetes sueltos que no pertenecen a ningún repositorio: paquetes compilados desde las fuentes, paquetes descargados directamente de Internet.
 - reparar el sistema de paquetes:
 - desinstalar paquetes temporalmente violando el sistema de dependencias
 - configurar paquetes que están a medio instalar
 - compilar paquetes fuente
 - manipulaciones diversas sobre archivos de paquetes.

23

Gestión a bajo nivel: dpkg

- dpkg --install <paquetes.deb>
 - instala archivos de paquetes
 - los paquetes que causen conflictos de dependencias no se instalarán (esto se puede forzar con opciones --force-*)
 - los paquetes que dependan de otros se desempaquetan pero no se configuran
- dpkg --configure <paquetes.deb> | --pending
 - configura paquetes previamente desempaquetados
- dpkg --remove | --purge <paquetes.deb>
 - desinstala paquetes (también la configuración con --purge)
 - no desinstala paquetes que romperían las dependencias
- dpkg --force-depends --remove <paquetes.deb>
 - desinstala paquetes, aunque se rompan las dependencias

24

Gestión a bajo nivel: dpkg

- `dpkg --list [<paquete o patrón>]`
 - lista paquetes disponibles y su estado (instalado, etc.)
- `dpkg --listfiles <paquete>`
 - lista archivos que contiene un paquete
- `dpkg --search <patrón>`
 - busca paquetes (instalados) que contienen archivos que corresponden al patrón
- `dpkg --print-avail <paquete>`
 - muestra información del paquete (`/var/lib/dpkg/available`)
- `dpkg-repack paquete`
 - re-empaqueta un paquete instalado, con su configuración actual.

25

Gestión a bajo nivel: dpkg. Ejemplo práctico

- Se han instalado paquetes “externos” a la distribución. Synaptic o apt-get no pueden eliminarlos porque se rompería el sistema de dependencias o porque son versiones más nuevas que las de los repositorios oficiales
- Solución:
 - buscar los paquetes externos con synaptic, apt-cache o `dpkg --list`
 - desinstalar los paquetes con “`dpkg --remove`”. Si es necesario, usar “`dpkg --force-depends --remove`”.
 - reparar las dependencias instalando paquetes oficiales necesarios con “`apt-get -f install`”
 - repetir con otros paquetes si es necesario

26

Contenidos

- Distribuciones
- Repositorios de paquetes
- Gestión de paquetes
- Gestión a alto nivel: sistema APT
- Reconfiguración de paquetes
- Gestión a bajo nivel: dpkg
- **Paquetes fuente**
- Resumen

27

Paquetes fuente

- Los paquetes Debian compilados (binarios) se generan a partir de paquetes fuente
- Un paquete fuente contiene:
 - código original del programa sin modificar en formato comprimido (.tar.gz)
 - modificaciones del desarrollador del paquete (.diff.gz)
 - archivo de descripción del paquete firmado digitalmente por el desarrollador (.dsc)
- Un mismo paquete fuente puede generar varios paquetes binarios

28

Paquetes fuente. Utilidad

- *Backporting*: compilar un paquete proveniente de una versión superior de la distribución o de otra distribución
 - la compatibilidad de paquetes fuente es mucho mayor que la de paquetes binarios
- Optimización:
 - los paquetes binarios se compilan con opciones genéricas para una arquitectura (Ej: i386).
 - la compilación en en la propia máquina optimiza el código para una arquitectura concreta (Ej: AMD, 64bits)
- Adaptación:
 - modificación del código fuente
 - activación/desactivación de opciones en tiempo de compilación
 - adaptación de opciones de compilación

29

Contenidos

- Distribuciones
- Repositorios de paquetes
- Gestión de paquetes
- Gestión a alto nivel: sistema APT
- Reconfiguración de paquetes
- Gestión a bajo nivel: dpkg
- Paquetes fuente
- **Resumen**

30

Resumen

- Buscar e instalar o desinstalar un paquete
 - Aplicaciones -> Centro de software de Ubuntu
 - Synaptic
 - apt-cache search|show + apt-get install/remove ...
- Añadir, quitar, editar repositorios
 - Sistema -> Administración -> Actualizar la distribución → Opciones
 - Synaptic
 - Editar /etc/apt/sources.list

31

Resumen

- Actualizar la distribución
 - Sistema -> Administración -> Gestor de actualizaciones
 - Synaptic
 - apt-get update + apt-get dist-upgrade
- Reconfigurar un paquete instalado
 - dpkg-reconfigure ...
- Instalar paquetes “suelos”
 - Centro de software (pinchar sobre archivo de paquete)
 - dpkg --install ...
- Reparar paquetes rotos
 - Synaptic
 - apt-get -f install
 - dpkg --configure --pending

32