

3. Instalación y gestión de paquetes

Elena García-Morato, Felipe Ortega, Enrique Soriano, Gorka Guardiola, Miguel Ortuño
GSyC, ETSIT. URJC.

Laboratorio de Sistemas (LSIS)

2 marzo, 2023





(cc) 2014-2023 Elena García-Morato, Felipe Ortega
Enrique Soriano, Gorka Guardiola, Miguel Ortuño.

Algunos derechos reservados. Este trabajo se entrega bajo la licencia
Creative Commons Reconocimiento - NoComercial - SinObraDerivada
(by-nc-nd). Para obtener la licencia completa, véase
<https://creativecommons.org/licenses/by-nc-nd/3.0/es/>.

Contenidos

- 3.1 Instalación de Linux
- 3.2 Gestión de paquetes software
- 3.3 *Advanced Package Tool* (APT)

3.1 Instalación de Linux

Distribuciones Linux

- Una distribución es una colección concreta de software de *área de usuario* y un núcleo (*kernel*) del sistema operativo.
- Hay muchas distribuciones de Linux (+600), agrupadas en familias.
- Las que usan el kernel de Linux y las herramientas de área de usuario de GNU se denominan GNU/Linux.
 - Ejemplo: Debian, Ubuntu, Red Hat, SUSE, ArchLinux, Manjaro, etc.
- Una distribución suele tener su sistema de gestión de paquetes para instalar el software.

Distribuciones Linux

- Ejemplos:
 - **apt** es el sistema de paquetes de las distribuciones basadas en Debian; formato de los paquetes **.deb**.
 - RPM es el sistema de gestión de paquetes de las distribuciones basadas en Red Hat; programas **yum** (alto nivel) o **rpm** (bajo nivel); formato de los paquetes **.rpm**.
 - **pacman** es el gestor de paquetes de ArchLinux; formato de los paquetes **tar**.

Distribuciones Linux

- Cada distribución tiene diferentes componentes: instalador, organización de directorios, ficheros de configuración, herramientas, si incluye o no software propietario, etc.
- Cada distribución tiene un esquema de versiones (número, alias, soporte a corto/largo plazo).
- Debemos tener cuidado de descargar la distribución compatible con la arquitectura de CPU de nuestro sistema.
 - Arquitectura amd64 para Intel/AMD 64 bits.
 - Arquitectura x86 para 32 bits.
 - Arquitectura arm para mini-pcs (ejemplo: Raspberry Pi) y otros dispositivos de bajo consumo.

Distribuciones Linux

- Algunas empresas dan soporte comercial a distribuciones profesionales: Canonical (Ubuntu), Red Hat, SUSE, etc.
- Ciertas distribuciones tienen usos específicos:
 - Seguridad informática (Kali).
 - Ocio (Kodi).
 - Plataformas específicas (Raspbian).
- Algunas tienen como objetivo ofrecer una interfaz y experiencia de usuario amigable para inexpertos, mientras que otras están más orientadas a personas que tienen más experiencia.

Estructura de directorios

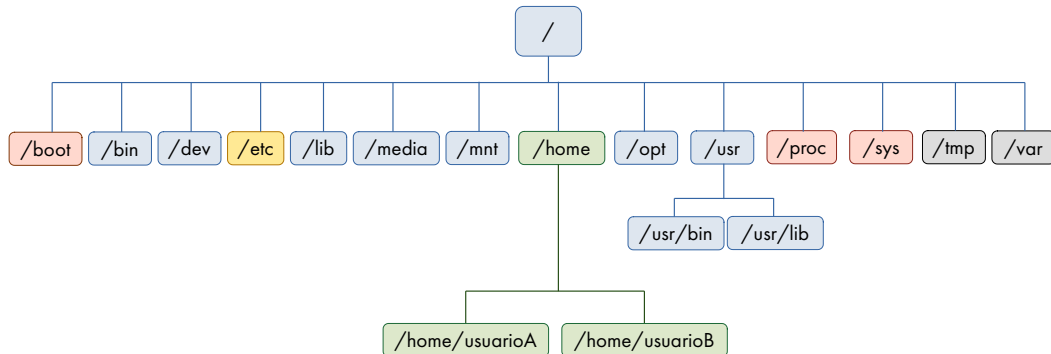


Figura 1: Algunos directorios importantes en sistemas Linux.

Estructura de directorios

- Datos de sistema.
 - /boot contiene ficheros de arranque del sistema.
 - /proc y /sys interfaz para interaccionar con el núcleo.
- Configuración.
 - /etc contiene ficheros de configuración.
- Dispositivos.
 - /dev dispositivos físicos.
 - /media y /mnt puntos de montaje de sistemas de ficheros (almacenamiento).
- Bibliotecas.
 - /lib tiene las bibliotecas (código) que usan los programas ejecutables.
 - También hay librerías en /usr/lib.

Estructura de directorios

- Usuarios.
 - Todas las carpetas de usuario cuelgan de /home. Por ejemplo /home/usuarioA es la carpeta del usuario usuarioA.
- Programas ejecutables.
 - /bin, /usr/bin, /sbin (ejecutables del sistema).
 - /opt donde se instalan otras aplicaciones software *de terceros*.
- Datos temporales y en tiempo de ejecución.
 - /var contiene datos generados en ejecución (cache, logs, ficheros creados por programas, etc.).
 - /tmp datos temporales. Los archivos en este directorio **se borran tras cada reinicio del sistema**.

Instalación del sistema

- Los ficheros se deben instalar en un dispositivo de **almacenamiento persistente**, es decir, que mantenga los datos almacenados aunque se apague la alimentación.
- Almacenamiento secundario.
 - **Disco duro magnético** (con platos que giran, partes mecánicas). Gran capacidad pero muy lento.
 - **Discos SSD** (*Solid State Disk*): mucho más rápidos, aunque tienen menor capacidad (precio mayor). Pueden tener interfaz SATA o bien PCIe. Estos últimos se denominan NVMe y alcanzan velocidades de acceso muy altas.
 - **Pen drives** sirven para almacenar sistemas operativos de arranque sin instalación (*live mode*).
 - **Discos ópticos** (DVD, Bluray): copias persistentes de los ficheros para instalar por primera vez en el sistema.

Instalación del sistema

- Para instalar los ficheros del sistema operativo en una unidad de disco, lo primero es definir *particiones*.
- Una **partición** es una zona del disco independiente. A todos los efectos es como si tuviésemos varios discos, uno por cada partición.
 - Si el disco es magnético y el dispositivo se identifica como `/dev/sda`, las particiones se identifican como `/dev/sda1`, `/dev/sda2`, etc.
 - Si el disco es NVMe y el dispositivo se identifica como `/dev/nvme0` o `/dev/nvme0n1` entonces las particiones son `/dev/nvme0n1p1`, `/dev/nvme0n1p2`, etc.
- El comando `fdisk` permite manipular las particiones de un disco. El comando `parted` o su interfaz gráfica `gparted` permiten hacerlo de forma más cómoda.

Instalación del sistema

- En algún sitio del disco (normalmente al comienzo) hay información sobre qué particiones hay, cómo se llaman, etc.
- Se indica para cada partición el nombre, tipo (si es de arranque o no) y otros datos. Ese lugar se denomina *tabla de particiones*.
- El esquema más utilizado hoy día es UEFI (*Unified extensible Firmware Interface*), que tiene su propio esquema para representar las particiones: GPT (GUID Partition Table).
- El cargador está en una partición de tipo ESP (EFI System partition, que es de tipo FAT).

Instalación del sistema

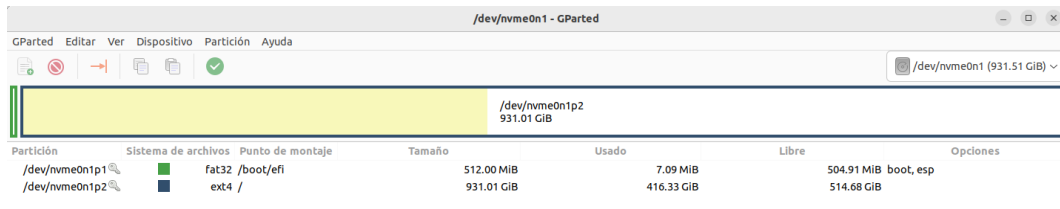


Figura 2: Ejemplo de particionado de un disco principal de Ubuntu, con el detalle de la partición UEFI (tipo FAT).

Instalación del sistema

- Cada partición debe estar *formateada* con un **sistema de ficheros** antes de poder almacenar datos.
- En Linux hay muchos sistemas de ficheros alternativos: ext3, ext4, btrfs, xfs, swap.
- Cada sistema de ficheros tiene ventajas e inconvenientes. El más extendido es ext4 pero nuevos sistemas como btrfs ofrecen muchas ventajas con nuevas tecnologías de almacenamiento (SSD) si lo configuramos adecuadamente.
- El formato swap es especial. Define un área de intercambio en la que el sistema puede volcar datos de la memoria a disco si empieza a escasear espacio en la RAM. Este intercambio se denomina *swap*.

Montaje de sistemas de ficheros

- Normalmente, no todos los ficheros del árbol de directorios se encuentran en el mismo dispositivo.
- Punto de montaje es el lugar del árbol de directorios donde está montado un disco (o partición) junto con todo su contenido accesible.
- Los puntos de montaje, así como los dispositivos y/o particiones que están en ellos se pueden consultar con las instrucciones `mount` o `df`.

mount

- Ejemplo de ejecución de mount.

```
/dev/hda2 on / type ext3 (rw,noatime)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda5 on /scratch type ext3 (ro,noatime)
tmpfs on /tmp type tmpfs (rw)
```

df

- Ejemplo de ejecución de df.

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/hda2	28842780	6957692	20419960	26%	/
/dev/hda5	38448276	32838556	3656620	90%	/scratch
tmpfs	517960	1196	516764	1%	/tmp

Montaje de sistemas de ficheros

Montaje de un sistema de ficheros

- Primero creamos el directorio, si no existe:
`mkdir /var`
- Hacer visible el sistema de ficheros bajo ese directorio:
`mount -t ext4 -o rw /dev/sda3 /var.`
- Si queremos que el montaje sea automático y permanente (para todos los arranques) entonces ponemos la información en `/etc/fstab`.
- Si queremos desmontar o hacer invisible un sistema de ficheros que esté montado en un directorio, por ejemplo `/var` lo hacemos con `umount /var`.
- Si llamamos a `mount` sobre una ubicación configurada en `/etc/fstab` utilizará las opciones configuradas allí.

3.2 Gestión de paquetes software

Distribución Ubuntu

- Heredera de Debian, sigue sus formatos de paquetes e instaladores.
- Creada y mantenida por la empresa Canonical Inc.
- Dos distribuciones principales ([Desktop](#) y [Server](#)) y otras distribuciones secundarias con diferentes gestores de ventanas/escritorio: Kubuntu, Lubuntu. . .
 - La distribución Desktop está orientada a usuarios finales y trae un entorno de escritorio preinstalado.
 - La distribución Server está orientada a servidores que precisan alto rendimiento y no incluye entorno de escritorio preinstalado.
- Versiones de dos tipos: cada seis meses y LTS (*Long Time Support*), que se mantiene varios años.

Distribución Ubuntu

- Algunos comandos importantes:

Información de la versión de Ubuntu

\$ `lsb release -a`

Información de la versión del kernel de Linux

\$ `uname -a`

apt: Mantener la instalación al día:

\$ `apt update`

\$ `apt upgrade`

Actualiza la distribución a una nueva versión

¡CUIDADO!! Puede ser peligroso

\$ `do-release-upgrade -p`

\$ `ubuntu-drivers devices` *# Paquetes recomendados de drivers para instalar.*

Paquetes software

- El software que puede instalarse en el sistema se publica en forma de **paquetes**: ficheros `.deb`.
- Son ficheros tipo `ar` (como un `tar`), muchos ficheros empaquetados en uno solo.
- Se puede ver con `ar x f.deb` (descarga uno de prueba dentro de un directorio con `apt download xia`).
- Dentro hay ficheros de datos (lo que se instala `data.tar.xz`), de control (`control.tar.xz`, dependencias), y *scripts* para ejecutar antes de instalar y al terminar.
- Los ficheros se instalan y se borran directamente en `dpkg`.
- Normalmente los descargamos de la red. . . , para eso utilizamos varias herramientas: `apt-search`, `apt-get` ...

3.3 *Advanced Package Tool* (APT)

Gestión de software: apt

- La herramienta más sencilla de usar y más potente. Usa repositorios: almacenes centralizados donde se mantienen los paquetes.
- Las direcciones de los repositorios se indican en el fichero `/etc/apt/sources.list`, aunque muchas distribuciones *enterprise* incluyen herramientas gráficas de gestión.
- También se permite añadir repositorios alternativos, *Personal Package Archives* (PPAs), con versiones alternativas o utilidades adicionales. Hay que tener **mucho cuidado** y verificar la fiabilidad de estas fuentes antes de usarlas.
- Los repositorios de Ubuntu se dividen en 4 secciones:
 - 1 Main. Soportada oficialmente por ubuntu. Libre.
 - 2 Restricted. Soportada oficialmente. No libre.
 - 3 Universe. No soportada oficialmente. Libre.
 - 4 Multiverse. No soportada oficialmente. No libre.

Gestión de software: apt

- La herramienta más sencilla de usar y más potente. Usa repositorios: almacenes centralizados donde se mantienen los paquetes.
- Las direcciones de los repositorios se indican en el fichero `/etc/apt/sources.list`, aunque muchas distribuciones *enterprise* incluyen herramientas gráficas de gestión.
- También se permite añadir repositorios alternativos, *Personal Package Archives* (PPAs), con versiones alternativas o utilidades adicionales. Hay que tener **mucho cuidado** y verificar la fiabilidad de estas fuentes antes de usarlas.
- Los repositorios de Ubuntu se dividen en 4 secciones:
 - 1 Main. Soportada oficialmente por ubuntu. Libre.
 - 2 Restricted. Soportada oficialmente. No libre.
 - 3 Universe. No soportada oficialmente. Libre.
 - 4 Multiverse. No soportada oficialmente. No libre.

Gestión de software: apt

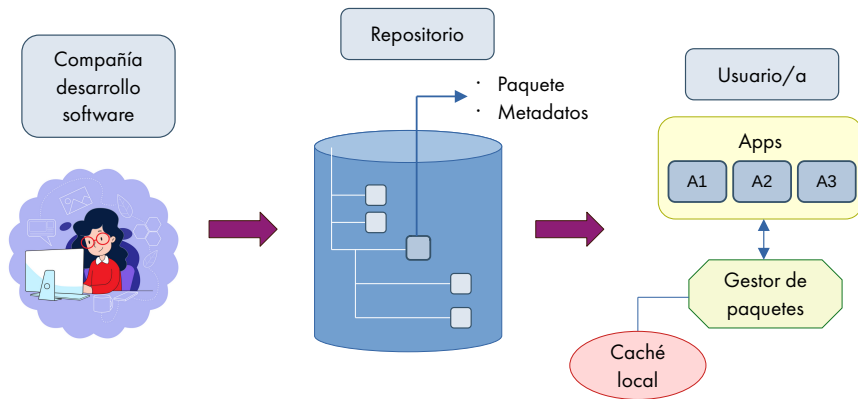


Figura 3: Proceso de desarrollo y publicación de aplicaciones software en sistemas Linux [1].

Gestión de software: apt

- Se pueden descargar paquetes de la red o instalarlos desde fuentes locales (como el disco o un disco óptico).
- Archivo `/etc/apt/sources.list`.
 - Los repositorios que empiezan por `deb` son binarios.
 - Los que empiezan por `deb-src` son de archivos de código fuente.
 - Los que tienen una dirección que comienza por `https` son remotos y se accede a ellos mediante un túnel seguro por la web.

Gestión de software: apt

- Además del archivo `/etc/apt/sources.list` el directorio `/etc/apt/sources.list.d` puede contener archivos en los que cada uno contiene más coordenadas de repositorios de paquetes adicionales que podemos descargar y gestionar.
- Si es un PPA se puede usar el comando `add-apt-repository` para añadirlos automáticamente.

```
add-apt-repository ppa:libreoffice/ppa
```

Gestión de software: apt

- Aunque solicitemos la instalación de un paquete, puede que no sea posible y el paquete quede retenido (*hold*).
- Posibles causas:
 - El paquete depende de otro que no está disponible en la distribución actual.
 - El administrador lo ha retenido a mano (no le gusta, puede ser problemático, etc.).
 - El paquete depende de otro pero de debe instalar la dependencia explícitamente. Se puede resolver ejecutando `apt-get install nombre-paquete`
- Cuidado: un paquete retenido para apt puede que no lo esté para aptitude, o viceversa.

Gestión de software: apt

- `aptitude remove <paquete>`: Desinstala un paquete resolviendo conflictos.
- `aptitude -purge remove <paquete>`: Además, borra todos los archivos de configuración relacionados con el paquete (resolviendo conflictos).
- `aptitude dist-upgrade`: Actualiza todos los paquetes, pasando si es necesario a una versión más reciente de la distribución.
 - En la actualidad, casi todas las distribuciones derivadas de Debian ofrecen sus propias herramientas para actualizar a una versión más reciente de la distribución.
- `aptitude clean`: Borra la caché local (usualmente `/var/apt/cache`), eliminando todos los archivos `.deb` descargados.

Gestión de software: apt

- ¿Cómo encontrar el paquete que necesito? (además de usar Google, claro...).
- `aptitude search <regex>`: Busca una regexp en los campos nombre o descripción de los paquetes.
- `dpkg -l <cadena>`: Igual que el anterior, muestra el estado de instalación del paquete.
- `aptitude show <paquete>`: Muestra una descripción completa del paquete.
- `dpkg-reconfigure <paquete>`: Dispara de nuevo el proceso de configuración del paquete.

Gestión de software: apt

- `apt autoremove` elimina los paquetes que el usuario no ha instalado directamente y que ya no son necesarios, por no estar en las dependencias de paquetes que están instalados.
- `apt clean` libera espacio de la caché de APT. Por ejemplo, borra `/var/cache/apt/archives/*`.
- Para ver a qué paquete ya instalado pertenece un fichero ejecutamos:
`dpkg -S fichero`.
- Para buscar qué paquete del repositorio contiene un fichero:
`apt-file search fichero`.

Gestión de software: apt

- Los archivos normalmente se descargan en `/var/cache/apt/archives`
- Se pueden consultar con `apt-cache` o bien `apt-get`.
- Ejemplo:

```
$ apt-cache search software-properties-common
software-properties-common - manage the repositories that you install software from (common)
```

Gestión de software: dpkg

- A un nivel más bajo que apt el comando **dpkg** permite manipular ficheros .deb individuales.
- Todas las dependencias previas deben estar satisfechas previamente antes de intentar la instalación con dpkg o incluirse en el propio comando. De lo contrario da error, no sabe resolverlas por sí solo.
- Típicamente casi nunca lo usaremos directamente, pero debemos entender su función.

Gestión de software: dpkg

```
root@host10# dpkg -i xia 2.2-3 all.deb
Selecting previously unselected package xia.
(Reading database ... 402253 files and directories currently installed.)
Preparing to unpack xia 2.2-3 all.deb ...
Unpacking xia (2.2-3) ...
Setting up xia (2.2-3) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

root@host10# dpkg -r xia
(Reading database ... 402696 files and directories currently installed.)
Removing xia (2.2-3) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
Processing triggers for mime-support (3.60ubuntu1) ...
```

Para saber más

- El capítulo 6 del libro de referencia de M. Hausenblas [1] explica con más detalle el proceso de desarrollo de aplicaciones software para Linux, así como otros gestores de paquetes en diferentes distribuciones, por ejemplo, RPM para Red Hat, Fedora y otras distribuciones.
- La sección 12.7 de [2] también incluye información adicional sobre los sistemas de gestión de paquetes software a alto nivel en Linux.

Referencias I

- [1] M. Hausenblas. *Learning Modern Linux*. O'Reilly Media, abr. de 2022.
- [2] E. Nemeth y col. *Unix and Linux System Administration Handbook*. 4ª ed. Pearson, 2010.