

Linux Administrator

Tema 2: Instalación, configuración y gestión de Ubuntu server

Sonia Fernández Sapena

Determinación del hardware Linux

Linux es compatible con las siguientes arquitecturas:

- **x86** para los ordenadores cuyos procesadores son del tipo Intel (del 386 al Pentium 4) o
 - AMD (Athlon, Duron, Sempron) de 32.
- **x86_64** para los ordenadores cuyos procesadores son del tipo Intel (Pentium 4 a partir de las series 600, Xeon, Dual Core/Quad Core, i3, i5, i7) o AMD (Athlon 64, Sempron 64, Opteron, Phenom, FX) de 64 bits. Esta versión no funciona en los procesadores de 32 bits. El uso de una versión de 64 bits de Linux en los equipos que lo soportan es muy recomendable, siendo las ventajas numerosas.
- **arm**: es un caso más particular, ya que esta familia de procesadores se usa fundamentalmente en sistemas embebidos, especialmente en dispositivos multimedia, en «cajas» multifunción, en routers, en lectores DVD, DivX y Blu-ray de salón, en GPS o en smartphones y tabletas. Esta arquitectura ha descendido tanto en 32 bits y 64 bits, mono o multiprocesador.
- **ppc** para los ordenadores cuyos procesadores son de tipo PowerPC, o sea, los antiguos ordenadores de la marca Apple. Esta versión no se instalará en las últimas máquinas Apple, basadas en un procesador Intel. Aún existen algunas distribuciones para esta arquitectura. Los PowerPC no están muertos con Apple, ya que se emplean en varias consolas de juegos, como la Wii U

Determinación del hardware Linux

Los siguientes requisitos previos deben ser respetados:

- Un procesador (o más) de tipo Intel Pentium 4 o superior o un equivalente de la marca AMD. Para las distribuciones aparecidas a partir de 2010, los gigahercios no están de más. Los procesadores Atom, Celeron, o AMD E1, de gama baja, permiten que Linux funcione de forma correcta si van acompañados de una buena cantidad de memoria y de un soporte de almacenamiento rápido.
- 256 o 512 MB de memoria permiten arrancar Linux en modo texto en un servidor. Piense, no obstante, en disponer de 1 GB o más para un uso óptimo para un simple servidor web, y al menos 2 GB para un equipo de escritorio. La inversión en memoria no es un lujo. Partamos del principio de que, cuanto más memoria, mejor será.
- 500 MB de espacio en disco para una instalación mínima (sin interfaz gráfica y solo con las herramientas básicas), pero de 2,5 GB a 4 GB para una instalación estándar, a la cual hay que añadir el espacio para los datos del usuario y la partición de intercambio.
- Una tarjeta gráfica (incluso antigua) compatible con la norma Vesa, que acepte una resolución de 1024 x 768 en 65 536 colores para el entorno gráfico. La resolución no tiene ninguna importancia si trabaja en modo texto. Para juegos, una tarjeta Nvidia reciente es lo mejor.

Determinación del hardware Linux

Varios sitios disponen de bases de datos de hardware compatibles para informarle con precisión. Los motores de búsqueda siguen siendo su mejor fuente. Una búsqueda de los términos «Linux Hardware Compatibility List» proporcionará un elevado número de respuestas. A título orientativo, le presentamos una lista de los sitios que le ayudarán en sus búsquedas:

Lista de compatibilidad Novell y OpenSUSE: http://en.opensuse.org/Hardware?LANG=en_UK

Lista de compatibilidad Ubuntu: <https://wiki.ubuntu.com/HardwareSupport>

Lista de compatibilidad LinuxQuestions: <https://www.linuxquestions.org/hcl/>

Impresoras: <https://wiki.linuxfoundation.org/openprinting/database/databaseintro>

Escáneres: <http://sane-project.org/sane-supported-devices.html>

Todos los dispositivos: <http://linux-drivers.org/>

Tarjetas de sonido: <http://www.alsa-project.org/main/index.php/Matrix:Main>

Tarjetas Wi-Fi: <https://hewlettpackard.github.io/wireless-tools/>

Módems internos o externos de tipo Windomem: <http://linmodems.org/>

Determinación del hardware Linux

Varios sitios disponen de bases de datos de hardware compatibles para informarle con precisión. Los motores de búsqueda siguen siendo su mejor fuente. Una búsqueda de los términos «Linux Hardware Compatibility List» proporcionará un elevado número de respuestas. A título orientativo, le presentamos una lista de los sitios que le ayudarán en sus búsquedas:

Lista de compatibilidad Novell y OpenSUSE: http://en.opensuse.org/Hardware?LANG=en_UK

Lista de compatibilidad Ubuntu: <https://wiki.ubuntu.com/HardwareSupport>

Lista de compatibilidad LinuxQuestions: <https://www.linuxquestions.org/hcl/>

Impresoras: <https://wiki.linuxfoundation.org/openprinting/database/databaseintro>

Escáneres: <http://sane-project.org/sane-supported-devices.html>

Todos los dispositivos: <http://linux-drivers.org/>

Tarjetas de sonido: <http://www.alsa-project.org/main/index.php/Matrix:Main>

Tarjetas Wi-Fi: <https://hewlettpackard.github.io/wireless-tools/>

Módems internos o externos de tipo Windomem: <http://linmodems.org/>

Instalación de Ubuntu Server 18.04 LTS

Ahora vamos a centrarnos en la instalación de Ubuntu Server 18.04 LTS. Los requerimientos de hardware que necesitaremos son:

Install Type	Install Method	CPU	RAM	Hard Drive Space	
				Base System	All Tasks Installed
Server (Standard)	debian-installer	1 gigahertz	512 megabytes	1.5 gigabyte	2.5 gigabytes
	live server	1 gigahertz (amd64 only)	1 gigabyte	1.5 gigabyte	n/a
Server (Minimal)	debian-installer	300 megahertz	384 megabytes	1.5 gigabytes	2.5 gigabytes

Instalación de Ubuntu Server 18.04 Its

Canonical hace las siguientes recomendaciones mínimas:

- 5 GB de espacio libre en disco.
- 1 GB de RAM.
- Procesador a 1 GHz o superior.
- Un dispositivo DVD o un puerto USB para el soporte de instalación.

Asegurarse de tener una copia de seguridad reciente de los datos importantes.

Además, se recomienda disponer de acceso a Internet durante el proceso de instalación.

Instalación de Ubuntu Server 18.04 Its

Para realizar el particionamiento tenemos que tener en cuenta qué tipo de servidor vamos a instalar, por ejemplo si vamos a configurarlo como servidor Web, sería interesante crear una partición y montarla en /var con el espacio suficiente para poder alojar las webs.

De forma automática y como mínimo nos creará las siguientes particiones:

1. Partición root / : dónde se instala el sistema operativo.
2. Una partición de 1 MB que se crea automáticamente y utilizará GRUB.
3. Partición /home: Para los datos

No hay partición swap (desde la versión 17.04) se crea swapfile : Memoria que se utiliza como memoria RAM virtual.

Instalación de Ubuntu Server 18.04 Its

Realización de práctica 1

Realización de práctica 2

El proceso de arranque Linux

El proceso de arranque de un sistema operativo Linux se inicializa de la siguiente manera:

- Cuando encendemos el servidor la BIOS/UEFI del equipo inicia las operaciones relacionadas con el arranque.
- La BIOS (Basic Input Output System) o **UEFI** (Unified Extensible Firmware Interface, interfaz extensible de firmware unificada), sucesor de EFI. Son pequeños programas escritos en lenguaje ensamblador cuya función es cargar el sistema operativo en la memoria RAM (Random Access Memory)
- una vez que la BIOS/UEFI carga el sistema operativo en RAM este inicia un proceso llamado POST (Power On Self Test) el cual es un proceso de diagnóstico y verificación de los componentes de entrada y salida de un servidor y se encarga de configurar y diagnosticar el estado del hardware
- Ya verificado el hardware, se inicia la fase de arranque del sistema (bootstrapping) el cual cede el control al GRUB (Grand Unified Bootloader), el GRUB es un gestor de arranque que hace uso de un menú gráfico que permite elegir el Sistema Operativo que se desea arrancar; Así mismo, el GRUB realiza las siguientes tareas:
 1. Cargar el kernel en memoria.
 2. Cargar el sistema de archivos virtual initd (o su sustituto systemd) el cual es usado típicamente para hacer los arreglos necesarios antes de que el sistema de archivos raíz pueda ser montado
 3. Pasarle los argumentos runlevel al kernel
 4. Comenzar la ejecución del kernel Al terminar de ejecutar todas las tareas anteriores el GRUB le cede el control total del arranque al kernel y este a su vez se encarga de realizar la llamada a la función starup la cual tiene como función detectar el tipo de CPU con el que el equipo cuenta así como de lo principal del sistema operativo, como el manejo de memoria, planificador de tareas, entradas y salidas, comunicación interprocesos, y demás sistemas de control, a partir de este momento se ejecuta el proceso INIT o systemd

Proceso INIT System V

Init System V en 2020

En 2020, ninguna distribución Linux digna de este nombre utiliza init System V.

Pero existen varias razones para estudiarlo:

1. Muchos editores solo ofrecen un script de arranque ubicado en `/etc/init.d/`.
2. En entornos empresariales hay muchas versiones antiguas de distribuciones Debian o Red Hat que funcionan con init.
3. Systemd y upstart mantienen una compatibilidad con la noción de runlevels y scripts init. Es evidente que esto es por la primera razón.
4. Los sucesores de init en Linux no son POSIX. Init es POSIX.

El primer proceso iniciado en el sistema

Si se miran los procesos en ejecución en el sistema, el proceso init se encuentra en primera posición. Tiene un PPID bastante particular (0) y es el padre de bastantes otros procesos.

Este proceso es el primero que se ejecuta en el arranque del kernel. Tiene evidentemente un rol privilegiado.

```
[root@beta ~]# ps -ef | head
UID      PID  PPID  C  STIME TTY          TIME CMD
root      1    0  0  09:07 ?        00:00:12 init [5]
root      2    1  0  09:07 ?        00:00:00 [migration/0]
root      3    1  0  09:07 ?        00:00:00 [ksoftirqd/0]
root      4    1  0  09:07 ?        00:00:00 [watchdog/0]
root      5    1  0  09:07 ?        00:00:09 [events/0]
root      6    1  0  09:07 ?        00:00:00 [khelper]
root      7    1  0  09:07 ?        00:00:00 [kthread]
root     10    7  0  09:07 ?        00:00:04 [kblockd/0]
root     11    7  0  09:07 ?        00:00:00 [kacpid]
[root@beta ~]#
```

Proceso INIT System V

Init

El programa init es el primer proceso que se inicia y el último que se para dentro del sistema, y su misión es ejecutar las demás tareas. La función de init consiste en iniciar y parar todos los servicios. Init ejecutará las diferentes tareas iniciales necesarias para el buen funcionamiento de Linux mediante la ejecución de varios comandos y scripts.

Una vez iniciado el sistema y ejecutados los servicios, init sigue activo para gestionar los cambios de estado de los procesos que controla y de los niveles de ejecución.

El proceso **init** es el padre de todos los procesos y tiene siempre el PID 1. Es arrancado por el kernel, que lleva el PID "virtual" 0. Su configuración está en el archivo `/etc/inittab`. Si este archivo está corrompido o es inutilizable, habrá que arrancar el sistema en modo single (S, s, 1 Single) y arreglarlo, o en el peor de los casos arrancar desde un soporte externo o un disco de emergencia. Es un archivo central del sistema operativo.

```
[root@beta ~]# ps -ef | head
UID      PID  PPID  C  STIME TTY          TIME CMD
root         1      0  0  09:07 ?        00:00:12 init [5]
root         2      1  0  09:07 ?        00:00:00 [migration/0]
root         3      1  0  09:07 ?        00:00:00 [ksoftirqd/0]
root         4      1  0  09:07 ?        00:00:00 [watchdog/0]
root         5      1  0  09:07 ?        00:00:09 [events/0]
root         6      1  0  09:07 ?        00:00:00 [khelper]
root         7      1  0  09:07 ?        00:00:00 [kthread]
root        10      7  0  09:07 ?        00:00:04 [kblockd/0]
root        11      7  0  09:07 ?        00:00:00 [kacpid]
[root@beta ~]#
```

Niveles de ejecución

Vamos a conocer los distintos tipos de niveles de ejecución que hay en Linux basados en System V y su compatibilidad con Systemd

Los niveles de ejecución

El funcionamiento de un sistema Linux se rige por los niveles de ejecución. Aunque este concepto parece un lastre hoy en día, más como una herencia del pasado que como una herramienta de administración real de una estación de trabajo Linux, su conocimiento es indispensable para una buena gestión del sistema.

Para empezar, hay que saber que un sistema Linux siempre está en algún nivel de ejecución, sea cual sea su actividad, ya se trate de un servidor apache que está respondiendo a una petición o un servidor nuevo todavía en su caja. La gestión de los niveles de ejecución consistirá en determinar cuál debe ser el comportamiento del sistema cuando este entra en un nivel concreto.

¿Qué es un nivel de ejecución?

Para simplificar, un nivel de ejecución es un nivel funcional en el que se ha determinado la lista de servicios que se arrancan o se detienen. Cuando un sistema entra en un nivel de ejecución, siempre consulta si hay que arrancar o parar servicios.

Niveles de ejecución

Los posibles niveles de ejecución

Nivel 0: El más simple: el sistema está detenido. Atención, esto no significa que no tenga que configurarse; todavía tiene que gestionar lo que sucede cuando el sistema entra en el nivel 0, es decir, cuáles son los servicios que se deben detener cuando se para físicamente una máquina.

El nivel 1 o monousuario: Un nivel un poco singular: está reservado para las operaciones de mantenimiento y solamente permite una conexión, la de la cuenta root. Además, la mayor parte de los servicios están detenidos en este nivel, lo que significa que el sistema tiene una actividad mínima. Esto es perfecto para el administrador que desea realizar operaciones de mantenimiento sin interferir con producción.

El nivel 2: En la mayoría de los sistemas, este nivel no se utiliza. Se deja a disposición del administrador, que podrá establecer a partir de este nivel un modo de funcionamiento particular con solamente algunos servicios iniciados. En los sistemas Debian y derivados (Ubuntu, por ejemplo) este nivel es, en cambio, el nivel funcional por defecto.

El nivel 3: En la mayor parte de los sistemas el nivel 3 es funcional, es decir, que todos los servicios están iniciados, pero no se dispone de interfaz gráfica.

El nivel 4: En la mayoría de los sistemas este nivel no se utiliza. Se deja a disposición del administrador, que podrá establecer a partir de este nivel un modo de funcionamiento particular con solamente algunos servicios iniciados.

El nivel 5: En la mayor parte de los sistemas el nivel 5 es funcional, es decir, que todos los servicios están iniciados y la interfaz gráfica está disponible. En los sistemas Debian y derivados (Ubuntu, por ejemplo) este nivel, en general, no se utiliza.

El nivel 6 :Temporal por definición, el nivel 6 es el de un sistema que está reiniciando. La configuración del nivel 6 consistirá en determinar qué servicios tienen que detenerse para el reinicio del sistema. Después del reinicio se aplicará un nuevo nivel de ejecución (en general el nivel por defecto) y se iniciarán los servicios asociados a este nivel.

Niveles de ejecución

Los 7 niveles de ejecución (runlevels) estándares

Nivel de ejecución	Nombre o denominación	Descripción
0	Alto	Alto o cierre del sistema (Apagado).
1	Modo de usuario único (Monousuario)	No configura la interfaz de red o los demonios de inicio, ni permite que ingresen otros usuarios que no sean el usuario root , sin contraseña. Este nivel de ejecución permite reparar problemas, o hacer pruebas en el sistema.
2	Multiusuario	Multiusuario sin soporte de red.
3	Multiusuario con soporte de red.	Inicia el sistema normalmente.
4	Multiusuario con soporte de red.	No es igual que el 3.
5	Multiusuario gráfico (X11)	Similar al <i>nivel de ejecución 3</i> + display manager .
6	Reinicio	Se reinicia el sistema.

Niveles de ejecución

¿Quién decide qué se encuentra en cada uno de los niveles?

En la inmensa mayoría de los casos, es la definición inicial de los niveles de ejecución la que se usa. Es decir, que el autor de la distribución (Ubuntu, Mandriva, Red Hat, etc.) elige lo que se debe activar en cada uno de los niveles de ejecución. El administrador de sistemas se conforma con esta decisión.

Sin embargo, puede darse el caso que el administrador de sistemas prefiera gestionar él mismo la configuración de los niveles de ejecución. Entonces puede elegir a qué nivel funcional corresponde cada uno de los niveles de ejecución y cuáles son los servicios asociados. A cada nivel de ejecución le corresponde entonces un conjunto de servicios.

Runlevel

0 - Apagar



1 - Monousuario



2 - Multiusuario



3 - Multiusuario
con Network



5 - Modo Grafico
Multiusuario



6 - Reiniciar



El Runlevel 4 no es usado

Niveles de ejecución

Cambiar los niveles de ejecución:

Se puede cambiar los niveles de ejecución de varias maneras:

Comprobamos el nivel de ejecución que tenemos actualmente con el comando **runlevel** y posteriormente podemos modificarlo:

1. Con el comando `init`
2. Con el comando `telinit`
3. `Systemctl` en `systemd`

```
root@ubuntu18server:/home/sonia# init --help
init [OPTIONS...] {COMMAND}

Send control commands to the init daemon.

--help      Show this help
--no-wall   Don't send wall message before halt/power-off/reboot

Commands:
0           Power-off the machine
6           Reboot the machine
2, 3, 4, 5  Start runlevelX.target unit
1, s, S     Enter rescue mode
q, Q       Reload init daemon configuration
u, U       Reexecute init daemon
root@ubuntu18server:/home/sonia#
```

```
sonia@ubuntu18server:~$ runlevel
N 5
sonia@ubuntu18server:~$ sudo telinit 3
[sudo] password for sonia:
sonia@ubuntu18server:~$ runlevel
5 3
sonia@ubuntu18server:~$ _
```

Systemd

Fundamentos

Systemd "system Daemon" es, al igual que init System V o upstart, un sistema de arranque alternativo al init clásico. Está diseñado específicamente para el núcleo de Linux. Al igual que el init clásico, es el primer proceso arrancado por el núcleo una vez concluida la inicialización del mismo.

Debian decidió en febrero de 2014 de adoptar **systemd** lo que venció las últimas resistencias de Canonical, fundador de Ubuntu y autor de upstart. Todas las distribuciones mayores (Red Hat, SuSE, Ubuntu, Debian) y sus derivados utilizan systemd actualmente.

Unidades objetivo y servicios

En systemd, el concepto de nivel de ejecución ya sólo existe para la compatibilidad con System V. El estado deseado por el sistema después de haber ejecutado los servicios se llama **unidad objetivo** o **target unit**. El componente básico de systemd es la **unidad** o **unit**. Existen varias: servicios, sockets, periféricos, objetivos, etc.

Los distintos servicios que se cargan con systemd se encuentran en `/lib/systemd/system`

Systemd

Equivalencia con init System V

Runlevel	Target Unit	Descripción
0	runlevel0.target, poweroff.target	Extinción de la máquina
1	runlevel1.target, rescue.target	Boot en modo single
2	runlevel2.target, multi-user.target	Consola, multi-usuario
3	runlevel3.target, multi-user.target	Consola, multi-usuario
4	runlevel4.target, multi-user.target	Consola, multi-usuario
5	runlevel5.target, graphical.target	Gráfico, multi-usuario
6	runlevel6.target, reboot.target	Reboot del equipo

Systemd

Comandos systemd

- Conocer el objetivo por defecto: `systemctl get-default`
- Cambiar el objetivo por defecto: `systemctl set-default multi-user.target`
- Pasar de un objetivo a otro: `systemctl isolate graphical.target`
- Podemos listar los objetivos (estos pueden ser: servicios, discos, otros objetivos, etc) : `systemctl list-units --type target` (nos listaria solo los objetivos type service)

Modo seguro y emergencia

Dos objetivos particulares tienen un acceso directo desde el comando `systemctl`: `rescue` y `emergency`.

- # **`systemctl rescue`** : corresponde al modo mono usuario
- # **`systemctl emergency`**: se dedica a los casos críticos; proporciona el modo de funcionamiento más pequeño posible: el sistema de archivos raíz se monta en modo solo lectura, no se puede acceder a ningún otro sistema de archivos ni a la red

Systemd

Comandos systemd servicios

Los servicios terminan con el sufijo `.service`. Al ser unidades como las otras, se controlan con el mismo comando **systemctl**. Tienen el mismo rol que los servicios `init System V`, y pocas cosas más, se utilizan de manera idéntica. Sin embargo, donde los servicios `System V` son con frecuencia simples scripts shell (`bash` ou `sh`), las unidades de servicios son archivos de texto estandarizados que describen las reglas y comandos vinculados a ese servicio.

	systemctl	Descripción
service name start	systemctl start name	Arranca el servicio.
service name stop	systemctl stop name	Detiene el servicio.
service name restart	systemctl restart name	Reinicia el servicio.
service name condrestart	systemctl try-restart name	Reinicia el servicio solo si está arrancado.
service name reload	systemctl reload name	Vuelve a cargar la configuración.
service name status	systemctl status name systemctl is-active name	Indica si el servicio está arrancado.
service --status-all	systemctl list-units --type service -all	Muestra el estado de todos los servicios.

Configuración básica teclado

En Ubuntu Server, durante la instalación tenemos una utilidad que nos permite configurar tanto el teclado, como el idioma como la red.

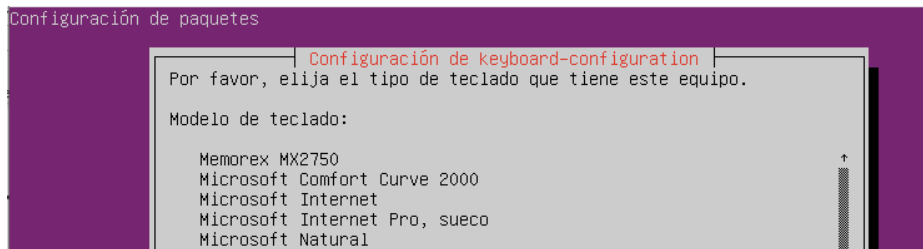
- No obstante si quisiéramos modificar esa configuración podríamos realizarlo mediante la edición del archivo `/etc/default/keyboard`,
- siempre y cuando no estemos trabajando con ninguna interface gráfica, si es así cambiará dependiendo de la interface.
- También podemos optar por el comando: `sudo dpkg-reconfigure keyboard-configuration` y nos volverá a solicitar la configuración del teclado.

```
# KEYBOARD CONFIGURATION FILE

# Consult the keyboard(5) manual page.

XKBMODEL="pc105"
XKBLayout="es"
XKBVARIANT=""
XKBOPTIONS=""

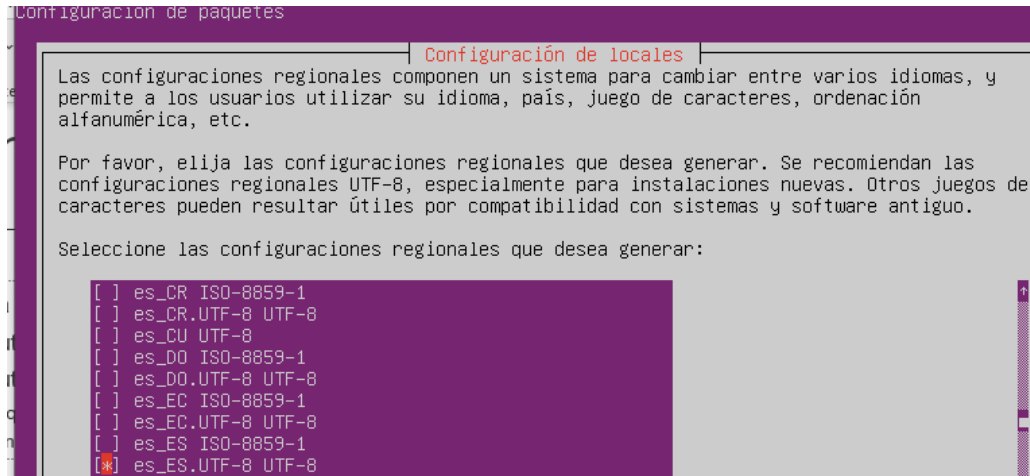
BACKSPACE="guess"
```



Configuración básica idioma

Vamos a ver como cambiar el idioma una vez instalado Ubuntu server

1. Ejecutamos locale (guarda toda la información sobre formato)
2. Ejecutamos dpkg-reconfigure locales
 1. Aquí podremos seleccionar de nuevo todas las opciones de configuración de idioma
 2. En el caso que no estuviera instalado el idioma, tendremos que instalarlo (lo veremos más tarde en el laboratorio 4)



Configuración básica zona horaria

Vamos a ver como cambiar la zona horaria una vez instalado Ubuntu server

1. Ejecutamos `timedatectl` para ver la información actual
2. Seleccionamos la zona horaria con:
 1. `sudo timedatectl set-timezone "Europe/Madrid"`
 2. Comprobamos que efectivamente la zona horaria ha cambiado y la hora también,

```
sonia@ubuntu18server:/etc$ sudo timedatectl
          Local time: mié 2020-02-26 18:53:40 UTC
          Universal time: mié 2020-02-26 18:53:40 UTC
          RTC time: mié 2020-02-26 18:53:41
          Time zone: Etc/UTC (UTC, +0000)
          System clock synchronized: no
          systemd-timesyncd.service active: yes
          RTC in local TZ: no
sonia@ubuntu18server:/etc$ sudo timedatectl set-timezone "Europe/Madrid"
sonia@ubuntu18server:/etc$ sudo timedatectl
          Local time: mié 2020-02-26 19:54:21 CET
          Universal time: mié 2020-02-26 18:54:21 UTC
          RTC time: mié 2020-02-26 18:54:22
          Time zone: Europe/Madrid (CET, +0100)
          System clock synchronized: no
          systemd-timesyncd.service active: yes
          RTC in local TZ: no
sonia@ubuntu18server:/etc$ _
```

Configuración básica tarjeta de red

Al igual que en el idioma y teclado, el asistente de la instalación de Ubuntu permite configurar la red desde el principio para poder tener acceso a las instalaciones y actualizaciones.

1. Puede ser que tengamos la necesidad de modificar la configuración de la tarjeta de red.
2. En Ubuntu Server desde la versión 17.04 esta configuración se hace a través de los archivos yaml gestionados por netplan.

YAML es un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python, Perl. YAML es un acrónimo recursivo que significa *YAML Ain't Markup Language* (en castellano, 'YAML no es un lenguaje de marcado'). Es un lenguaje sencillo que no acepta tabuladores.

La configuración de la red se encuentra dentro del directorio `/etc/netplan` y por defecto en el archivo `50-cloud-init.yaml`. Es un buen método hacer una copia de seguridad del archivo antes de su modificación.

```
sonia@ubuntu18server:/etc/netplan$ dir
50-cloud-init.yaml
sonia@ubuntu18server:/etc/netplan$ sudo cp 50-cloud-init.yaml 50-cloud-copia.yaml
sonia@ubuntu18server:/etc/netplan$ sudo nano 50-cloud-init.yaml
```

```
network:
  ethernets:
    ens33:
      addresses:
        - 192.168.1.75/24
      gateway4: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
  version: 2
```

Gestión de paquetes

Al contrario de otros sistemas operativos, con Linux y Unix no es habitual disponer de software proporcionado con un programa de instalación interactivo (y no install.exe).

Con Linux es muy habitual disponer de varios productos, herramientas, actualizaciones, etc. en forma de paquetes (packages).

Un **paquete** es un archivo que contiene el producto a instalar y unas reglas. Según su contenido, su tamaño puede ser muy imponente. El núcleo y todos sus módulos son, por ejemplo, proporcionados en esta forma.

Las reglas pueden ser múltiples:

- Gestión de las dependencias: sólo se podrá instalar el producto si los productos que él mismo utiliza están ya presentes.
- Preinstalación: se deben prever acciones antes de poder instalar el producto (cambiar derechos, crear directorios, etc.).
- Postinstalación: se deben prever acciones después de la instalación del producto (parámetros de un archivo de configuración, compilación anexa, etc.).

En las distribuciones Red Hat, SuSE y derivadas (CentOS, Fedora, OpenSUSE...), el formato de paquete por defecto es el **RPM** (Red Hat Package Manager). En Debian y sus derivados (Ubuntu...) es el formato **DPKG** (Debian Package).

Gestión de paquetes

EL GESTOR RPM

RPM es un gestor de paquetes desarrollado por Red Hat que luego ha sido adoptado de manera masiva por muchas otras distribuciones. Simplifica mucho la distribución, la instalación, la actualización y la supresión de los programas. Se basa en comandos (p. ej.: rpm), una base de datos local y paquetes en formato rpm (extensión rpm).

La base de datos está ubicada en /var/lib/rpm. Contiene toda la información correspondiente a los programas instalados, sus versiones, sus archivos y derechos, y sus dependencias. A no ser que se produzca un grave problema, no se debe modificar NUNCA esta base de datos. Hay que utilizar las herramientas RPM.

Para instalar un paquete utilizaremos por ejemplo: ***rpm -Uvh php-5.4.16-42.el7.x86_64.rpm***

-U: después de instalar se actualizará

-v: verbosidad, para ver el paquete que se instala

-h: visualiza # para ver el progreso de la instalación

Gestión de paquetes

YUM

YUM es a los archivos **rpm** lo que APT es a los archivos **dpkg**: un programa de gestión de paquetes. Recupera los paquetes dentro de los repositorios y gestiona las dependencias en lugar de usted. YUM significa Yellow dog Updater Modified. Se utiliza principalmente en las distribuciones Red Hat (versiones Enterprise) y Fedora, pero se puede utilizar en cualquier distribución de tipo RPM si los repositorios asociados lo soportan.

Gestión de paquetes

Debian Package **dpkg**: el gestor de paquetes Debian

El comando **dpkg** es el equivalente de rpm para las distribuciones Debian y derivados, entre las cuales se encuentra Ubuntu. Hace lo mismo o casi que **rpm**. Los paquetes Debian llevan una extensión `.deb` para reconocerlos y disponen de la misma información y mismos medios que un paquete **rpm**. El comando **dpkg** es el encargado de la instalación, la creación, la supresión y la gestión de los paquetes Debian.

Se suele colocar la base de datos **dpkg** en `/var/lib/dpkg`. Los archivos de este directorio tienen formato de texto. Sin embargo, no edite los archivos manualmente. El archivo `/var/lib/dpkg/status` contiene la totalidad de los paquetes conocidos por **dpkg** con su estado.

Gestión de paquetes

INSTALACION

La opción -i, o --install, instala el paquete o los paquetes que se pasan como argumento: `# dpkg -i mipaquete.deb`

dpkg no gestiona las dependencias

Se actualiza de la misma manera que se instala, con -i. Si instala un paquete ya presente, dpkg efectúa su actualización. Así, una instalación o una actualización respeta la metodología siguiente:

- Extracción de los archivos de control del nuevo paquete.
- Cuando ya se instaló una antigua versión del mismo paquete, se ejecuta el script presupresión del antiguo paquete.
- Ejecución del script de preinstalación si se facilita con el paquete.
- Desempaquetado de nuevos archivos y copia de seguridad de los antiguos para poder restaurarlos en caso de problema.
- Cuando ya se ha instalado una antigua versión del paquete, se ejecuta el script postsupresión del antiguo paquete.
- Configuración del paquete.

Gestión de paquetes

La **supresión** de un paquete se lleva a cabo con el parámetro `-r` (en minúscula). Esta vez, también le corresponde a usted gestionar las dependencias. La supresión de un paquete sigue las etapas siguientes:

- Ejecución del script de presupresión.
- Supresión de los archivos instalados.
- Ejecución del script de postsupresión.

```
# dpkg -r zip
```

```
# dpkg -P apache
```

Con la opción `-P` para purgar todos los archivos al borrarlos

la utilización de las opciones `--force-all` y `--purge` permite forzar la desinstalación del paquete y suprimir los archivos de configuración asociados.

`dpkg --force-all --purge nombre_del_paquete`

Gestión de paquetes

Consultas dpkg

- `Dpkg -l` : listar paquetes
- `Dpkg -l "apt"`
- `dpkg --get-selections | grep ^linux`
- `dpkg -S /usr/bin/basename`

Gestión de paquetes

ADMINISTRADOR APT

Fundamentos

Ya sea con rpm o dpkg, el problema sigue siendo el mismo: estas dos herramientas detectan las dependencias de los paquetes para autorizar o no su instalación, pero no las resuelven. Dicho de otro modo, si una dependencia de un paquete no está, no se instalará salvo si se resuelve la dependencia:

APT permite resolver estos problemas gestionando las dependencias en lugar de usted. **APT** significa Advanced Packaging Tool. En vez de especificar un paquete (local o remoto), se encarga de los repositorios de paquetes

Un repositorio contiene un conjunto de paquetes que dependen o bien los unos de los otros, o bien de otros paquetes procedentes de otros repositorios. APT puede gestionar varios repositorios en varios sitios. Se las arregla solo: cuando usted instala un paquete, también instala sus dependencias (si las encuentra).

Gestión de paquetes

ADMINISTRADOR APT	Comando apt-get	Comando equivalente apt	Función del comando
Comandos	<code>apt-get update</code>	<code>apt update</code>	Actualizar los repositorios de nuestra distribución. Los repositorios se acostumbran a configurar en el archivo <code>/etc/apt/sources.list</code>
APT permite ser ejecutado con dos utilidades que parecen iguales pero son distintas: APT y APT-GET. Son casi idénticas porque comparten el mismo núcleo de ejecución. <ul style="list-style-type: none">◦ APT nació en 2004 (más amigable)◦ Apt-get en 1998	<code>apt-get install</code>	<code>apt install</code>	Instalar uno o varios paquetes en nuestro sistema operativo.
	<code>apt-get upgrade</code>	<code>apt upgrade</code>	Actualizar los paquetes y programas que tenemos instalados. Con apt-get, si los paquetes requieren nuevas dependencias no se actualizarán. Con apt, si los paquetes actualizables requieren de nuevas dependencias, se instalarán las nuevas dependencias y se actualizarán los paquetes. Utilizando estos comando nunca desinstalaremos ningún paquete.
	<code>apt-get dist-upgrade</code>	<code>apt dist-upgrade</code>	Actualizamos la totalidad de paquetes que tenemos instalados. Durante el proceso de actualización se instalarán y actualizarán las dependencias necesarias. En caso que sea necesario se pueden desinstalar paquetes de nuestro sistema para resolver conflictos de dependencias.

Gestión de paquetes

ADMINISTRADOR APT

Comandos

Comando apt-get	Comando equivalente apt	Función del comando
<code>apt-cache depends</code>	<code>apt depends</code>	Mostrar la totalidad de dependencias de un paquete del sistema.
<code>apt-cache rdepends</code>	<code>apt rdepends</code>	Mostrar las dependencias inversas de un paquete.
<code>apt-cache policy</code>	<code>apt policy</code>	Muestra la prioridad de instalación de cada uno de los repositorios y sus ramas.
<code>nano /etc/apt/sources.list</code>	<code>apt edit-sources</code>	Editar el contenido de nuestros repositorios en el fichero <code>/etc/apt/sources.list</code> .
<code>apt-cache show</code>	<code>apt show</code>	Muestra la información y características sobre un determinado paquete presente en nuestros repositorios.
<code>apt-cache showsrc</code>	<code>apt showsrc</code>	Mostrar detalles de los paquetes fuente de nuestros repositorios.
<code>apt-show-versions</code>	<code>apt list</code>	Obtener listados de los paquetes instalados, actualizables, que se han actualizado manualmente, etc.

Gestión de paquetes

ADMINISTRADOR APT	Comando apt-get	Comando equivalente apt	Función del comando
Comandos	<code>apt-get remove</code>	<code>apt remove</code>	Desinstalar uno o varios paquetes manteniendo sus archivos de configuración.
	<code>apt-get purge</code>	<code>apt purge</code>	Desinstalar los paquetes y los archivos de configuración de las utilidades desinstaladas.
	<code>apt-get autoremove</code>	<code>apt autoremove</code>	Para desinstalar las dependencias de programas que en su día eran necesarias, pero ahora ya no son necesarias.
	<code>apt-get clean</code>	<code>apt clean</code>	Borrar los paquetes binarios almacenados en las ubicaciones <code>/var/cache/apt/archive</code> y <code>/var/cache/apt/archives/partial/</code> .
	<code>apt-get autoclean</code>	<code>apt autoclean</code>	Eliminar paquetes binarios que ya no pueden ser descargados de los repositorios de la ubicación <code>/var/cache/apt/archive</code> .
	<code>apt-get check</code>	–	Verificar que se cumplen todas las dependencias del sistema-
	<code>apt-get source</code>	<code>apt source</code>	Permite descargar el código fuente de un programa o paquete en el directorio.
	<code>apt-get download</code>	<code>apt download</code>	Descargar paquetes binarios de nuestros repositorios en el directorio actual.

Gestión de paquetes

LOS REPOSITORIOS

Los repositorios se definen en el archivo `/etc/apt/sources.list` y también en los archivos ubicados en `/etc/apt/sources.list.d`.

```
$ cat sources.list | grep -v "^(#|$)"
deb http://es.archive.ubuntu.com/ubuntu/ xenial main restricted
deb http://es.archive.ubuntu.com/ubuntu/ xenial-updates main restricted
deb http://es.archive.ubuntu.com/ubuntu/ xenial universe
deb http://es.archive.ubuntu.com/ubuntu/ xenial-updates universe
deb http://es.archive.ubuntu.com/ubuntu/ xenial multiverse
deb http://es.archive.ubuntu.com/ubuntu/ xenial-updates multiverse
deb http://es.archive.ubuntu.com/ubuntu/ xenial-backports main restricted
universe multiverse
deb http://security.ubuntu.com/ubuntu xenial-security main restricted
deb http://security.ubuntu.com/ubuntu xenial-security universe
deb http://security.ubuntu.com/ubuntu xenial-security multiverse
```

Una vez configurados sus repositorios, debe actualizar la base de datos local de APT con el comando `apt-get` y la opción `update`. Este comando actualiza la lista de paquetes y sus versiones, pero no instala nada.

```
root@ubuntu:~# apt-get update
o
root@ubuntu:~# apt update
```

Una vez actualizados los paquetes de los repositorios, si queremos actualizar las versiones de los paquetes que tenemos instalados ejecutaremos

```
root@ubuntu:~# apt-get upgrade
O
root@ubuntu:~# apt upgrade
```

Gestión de paquetes

Realización de Práctica 3

Librerías compartidas

Fundamentos

Una librería compartida es un archivo particular que contiene una lista de funciones, o API, accesible a cualquier programa que lo necesite sin tener que volver a escribirlas. Al contrario de lo que sucede con las librerías estáticas, el programa accede de manera dinámica a las funciones situadas en un archivo aparte. N programas diferentes pueden acceder a las funciones propuestas por la librería. Las librerías agrupan funciones propias de un dominio o un conjunto de dominios coherentes: tratamiento de imágenes, sonido, acceso a una base de datos, etc.

Un conjunto de funciones propuestas por una o varias librerías compartidas forma una **API**, Application Programming Interface, y a veces se agrupan en un framework que ofrece una solución completa para un dominio dado.

El proceso de vinculación entre la librería y el programa se realiza durante la compilación.

En Linux (y Unix en general), las librerías compartidas se llaman **Shared Objects** (so) en el sentido en el que se trata de archivos de objetos sin bloque de instrucción **main**. Los nombres de los archivos asociados llevan el sufijo **.so**.

Librerías compartidas

Lugar de almacenamiento

Por convención se colocan las librerías compartidas en directorios llamados lib:

- /lib: librerías de sistema básicas, vitales.
- /lib64: librerías de sistema de 64 bits.
- /usr/lib: librerías de usuario básicas, no necesarias para el boot.
- /usr/local/lib: librerías locales para los programas para la máquina.
- /usr/X11R6/lib: librerías del entorno X Window

```
sonia@ubuntu18server:/usr/lib$ ls -l *.so*
lrwxrwxrwx 1 root root    21 dic  9 11:03 libDeployPkg.so.0 -> libDeployPkg.so.0.0.0
-rw-r--r-- 1 root root 31304 dic  9 11:03 libDeployPkg.so.0.0.0
lrwxrwxrwx 1 root root    20 dic  9 11:03 libguestlib.so.0 -> libguestlib.so.0.0.0
-rw-r--r-- 1 root root 26752 dic  9 11:03 libguestlib.so.0.0.0
lrwxrwxrwx 1 root root    16 dic  9 11:03 libhgfs.so.0 -> libhgfs.so.0.0.0
-rw-r--r-- 1 root root 167248 dic  9 11:03 libhgfs.so.0.0.0
lrwxrwxrwx 1 root root    18 dic  9 11:03 libvgauth.so.0 -> libvgauth.so.0.0.0
-rw-r--r-- 1 root root 85144 dic  9 11:03 libvgauth.so.0.0.0
lrwxrwxrwx 1 root root    19 dic  9 11:03 libvmttools.so.0 -> libvmttools.so.0.0.0
-rw-r--r-- 1 root root 627384 dic  9 11:03 libvmttools.so.0.0.0
```

Librerías compartidas

La edición de los vínculos con una librería compartida es dinámica y el sistema lo hace en el momento de la ejecución del programa con la ayuda de la librería **ld.so**. El binario proporciona el nombre de las librerías a vincular con la ejecución, pero no la ruta. Las funciones de **ld.so** determinan la librería a utilizar en función de su nombre, entre las rutas que conocen.

Cualquier programa se vincula a la librería **ld.so** o más bien **ld-linux.so** (ld-linux.so.2).

El cargador de vínculos **ld.so** busca las librerías en varios lugares, entre los cuales, se encuentran, por este orden:

1. las rutas precisadas en la variable de entorno **LD_LIBRARY_PATH**. Se separan las rutas, como para PATH, por «:».
2. el contenido del archivo /etc/ld.so.cache, que contiene una lista compilada (formato binario) de las librerías encontradas en las rutas predefinidas.
3. los directorios /lib y /usr/lib.

La búsqueda en /lib y /usr/lib está implícita. Por otro lado, el hecho de rellenar la variable **LD_LIBRARY_PATH** no impide la búsqueda de las librerías en otros sitios, si no está en una de las rutas de la lista.

Para evitar la instalación de una variable cuyo contenido se puede manejar con dificultad, ld.so propone una caché que puede modificar por sí mismo. Se construye la caché desde el contenido del archivo /etc/ld.so.conf y del comando **ldconfig**. Este archivo contiene la lista de los directorios que contienen las librerías compartidas.

Librerías compartidas

```
sonia@ubuntu18server:/etc$ cat ld.so.conf
include /etc/ld.so.conf.d/*.conf
```

Podemos decidir añadir un archivo en `/etc/ld.so.conf.d` que contenga la ruta o las rutas de sus nuevas librerías que vayamos a necesitar.

No basta con añadir la ruta: debe regenerar la caché con el comando **ldconfig**.

El comando **ldconfig**:

- actualiza la caché para las rutas definidas en `/etc/ld.so.conf` y asociadas, así como para `/usr/lib` y `/lib`;
- actualiza los vínculos simbólicos en las librerías;
- permite también listar las librerías conocidas en la caché.

Se aceptan las siguientes opciones:

Opción	Papel
-v	Modo verboso: indica lo que ldconfig efectúa
-N	No vuelve a construir la caché
-X	No actualiza los vínculos
-p	Lista el contenido de la caché

**bin**

Binarios de usuario

boot

Ejecutables y archivos requeridos para el arranque

dev

Archivos de información de todos los volúmenes

etc

Archivos de configuración del sistema y de aplicaciones

home

Directorio personal con las carpetas de usuario

lib

Bibliotecas necesarias para la ejecución de binarios

media

Directorio de montaje de volúmenes extraíbles

opt

Ficheros de aplicaciones externas que no se integran en /usr

proc

Ficheros de información de procesos

root

Directorio personal de superusuario

sbin

Binarios de sistema

srv

Archivos relativos a servidores web, FTP, etc.

sys

Archivos virtuales con información de eventos del sistema

tmp

Directorio de ficheros temporales

usr

Archivos de programas y aplicaciones instaladas

var

Archivos de variables, logs, emails de los usuarios del sistema, etc.