

MANUAL DE ESTUDIO

LPI C-1

“Las obras de conocimiento deben ser libres, no hay excusas para que no sea así.”

Richard Matthew Stallman

“El único conocimiento verdadero es saber que no sabes nada”

Sócrates

- Dedicado a Marx -

(A toda la familia pero, en especial, a Groucho)

Prólogo

Este manual se ha desarrollado bajo los conceptos de un proyecto de software Libre, como obra colaborativa de la que pueden surgir trabajos derivados que favorezcan el conocimiento. El contenido se centra fundamentalmente en la certificación LPIC -1 , pero también cubre la nueva certificación Linux Essentials (aunque no es el objetivo básico), algunos aspectos de LPIC-2 y elementos que han parecido relevantes.

Por otra parte, entendemos que el conocimiento no es un compartimento estanco centrado en unos puntos que puedan perder su relación con una solución global, es por esto que, tras cubrir los conocimientos exigidos para obtener las certificaciones, se incluye material relevante relacionado con los temas que se estén tocando. Ponemos un ejemplo sencillo: la certificación exige configurar un cliente de DNS, para ello (aunque no sea una exigencia de la certificación) implementaremos un servidor de DNS básico para poder realizar las prácticas y afianzar los conocimientos.

INTRODUCCIÓN

Linux essentials es un curso reciente que ofrece una certificación básica de conocimiento sobre sistemas GNU/linux. Ya se ha comentado que no se va a seguir un guión específico para ésta pero vamos a incorporar el índice del curso para aquellos alumnos que deseen obtener esta certificación (cubierta por completo en LPIC-1) y puedan referenciarse a los capítulos de este mismo manual.

1. Ordenadores, software y sistemas operativos
2. Linux y el software libre
3. Primeros pasos con linux
4. Trabajando con el Shell?
5. Obtención de ayuda
6. Trabajar con ficheros
7. Expresiones regulares
8. Operaciones de entra/salida y comandos de filtrado
9. Más operaciones con el shell: scripts
10. El sistema de ficheros
11. Compresión y empaquetado de ficheros
12. Introducción a la administración de ficheros
13. Administración de usuarios
14. Control de acceso
15. Linux Networking

FUNDAMENTOS DE LINUX

Este capítulo versará sobre elementos esenciales del aprendizaje de Linux.

Concepto de ordenador.

Ya durante la II guerra mundial se desarrollaron máquinas que podríamos considerar ordenadores ya que cumplían estos preceptos básicos:

1. Procesamiento de datos en base a la ejecución de una secuencia de instrucciones automatizadas. Lo que conocemos como un programa.
2. Los programas deben permitir la ejecución condicional y los bucles.
3. Debe ser posible cambiar o reemplazar el programa que ejecuta el ordenador.

Formalmente, Howard Aiken fue un pionero de la computación y el diseñador de la primera computadora para IBM, el "Harvard Mark I". Varios fabricantes de computadoras funcionaron en los años 70 como por ejemplo Digital Equipment Corporation (DEC) pero no fue hasta finales de los años 70 y los años 80 cuando empezaron los ordenadores personales como el IBM PC o el Commodore 64. Sin embargo las premisas básicas son las mismas.

Elementos de un ordenador

Hardware

Recordamos algunos conceptos básicos que para la certificación Linux Essentials son necesarios y que se pueden completar con amplia información detallada consultando Internet.

- Placa Base: La placa base, también conocida como placa madre (motherboard o mainboard) es una tarjeta de circuito impreso a la que se conectan los componentes que constituyen el ordenador.
- Microprocesador: Es un circuito integrado conformado por millones de componentes electrónicos. Es el encargado de ejecutar los programas, desde el sistema operativo hasta las aplicaciones de usuario
- Memoria RAM: Random Access Memory, en la que se almacenan tanto los datos que se van a procesar como el código que se ejecuta. Es volátil, esto es, la información se pierde al perder la energía.
- Almacenamiento: Discos duros (HD) donde se almacenan los datos de forma permanente. Existen diversas tecnologías como Parallel ATA (PATA o IDE), Serial ATA (SATA), SCSI, o Serial SCSI. También existen conexiones de almacenamiento externas como USB o eSATA. En la actualidad se están introduciendo los discos SSD (Solid State Disk) que tienen un funcionamiento similar a la memoria y no a la mecánica de platos de las tecnologías anteriores.
- Fuente de alimentación: Que permite alimentar los componentes del ordenador convirtiendo la corriente alterna a continua con los voltajes adecuados.
- Periféricos: Como el monitor (antiguos CRT, LCD, ...), ratón, teclado, etc

Software

- Firmware: Software que permite una serie de funciones básicas de reconocimiento e inicialización de dispositivos y que en el mundo PC se identifica con la BIOS y en otras

arquitecturas con sistemas OBP

- Sistema Operativo: Una serie de programas que permiten la gestión del hardware y que se ejecuten las aplicaciones de usuario. Existen una gran cantidad de sistemas operativos en todo tipo de dispositivos. Desde el IOS o JUNOS en infraestructura de red, Android o BlackBerry en dispositivos móviles, sistemas embebidos, hasta los diferentes Windows en el escritorio o sistemas de servidor. Es obligado hacer hincapié en sistemas windows (derivados todos del kernel NT) que están muy extendidos en el escritorio y servidores departamentales. Solaris es un unix con alta presencia en grandes empresas. En el ámbito doméstico, además de los omnipresentes Windows, nos podemos encontrar con un pequeño porcentaje de OS X de Apple. Y por supuesto GNU/linux sin el que no se entendería la actual internet pública.
- Programas de usuario: desde ofimática, navegadores, CAD, retoque fotográfico,

Más adelante ahondaremos un poco en estas divisiones y las comentaremos en el entorno del software libre o del código abierto cuando se expliquen estos conceptos.

En el principio fue UNIX (todo empezó el 1 de enero de 1970 a las 0:00)

Unix es un Sistema Operativo que se desarrolló en los laboratorios Bell de AT&T a principio de los años 70 por Ken Thompson, Dennis Ritchie y otros. La intención de estos ingenieros era crear un sistema de archivos exclusivo.

Debido a la importancia de poder transportar Unix a las diferentes máquinas que iban surgiendo, apareció la segunda versión de Unix en 1971, que fue la gran precursora del Unix moderno gracias a la incorporación del lenguaje de programación C y posteriormente al concepto de los pipes (tuberías), que contribuyeron en gran medida al tratamiento de los datos. Los pipes no son otra cosa que un mecanismo para entregar el resultado de una acción, por ejemplo la ejecución de una orden o comando, a otro comando, para que este último lo utilice y pueda llevar a cabo su trabajo.

Ritchie estimó que el primer sistema en C era de un 20 a un 40 por ciento más grande y lento porque no estaba escrito en lenguaje ensamblador, que fue el lenguaje con el que se creó la primera versión de Unix, pero las ventajas de usar un lenguaje de alto nivel superaban largamente a las desventajas, sobre todo en lo referente a la portabilidad ya que, escrito en C, Unix era relativamente fácil de trasladar a otras plataformas hardware. El lenguaje ensamblador es un lenguaje de bajo nivel y aunque esto podría dar a entender que se trata de un lenguaje poco potente, no es así. Lo que indica es que se trata de un lenguaje muy parecido al código máquina, que es el lenguaje de más bajo nivel basado en ceros y unos, y que es el que en realidad utilizan los ordenadores.

Las primeras interesadas fueron las universidades, debido a la capacidad de Unix de ejecutarse en diferentes computadoras. De esta forma, tanto las universidades como otras empresas interesadas, fueron añadiendo diferentes funcionalidades a Unix adaptándolo a sus necesidades. Así se fueron desarrollando multitud de variantes para las diferentes computadoras. Además, también surgieron gran variedad de aplicaciones debido a los trabajos conjuntos de universidades como Berkeley, compañías como Sun Microsystems y los laboratorios Bell. Como consecuencia, y debido a que Unix se podía ejecutar en diferentes tipos de hardware, los sistemas patentados de otros fabricantes fueron desapareciendo, ya que estos últimos sólo se podían ejecutar en un hardware específico.

De esta manera, mientras AT&T modificaba su versión de Unix, en la Universidad Californiana de Berkeley se estaba llevando a cabo la modificación de otra versión anterior, la cual recibió el nombre de BSD (Berkeley Software Distribution). Esto ha proporcionado la posibilidad de clasificar los diferentes sistemas Unix en dos grupos, los basados en System V de AT&T o en BSD de Berkeley. El que ambos tengan predecesores comunes ha dado como resultado que el funcionamiento de ambos sea muy similar pero con ligeras modificaciones, por ejemplo: el

comando para imprimir para versiones basadas en System V es lp, mientras que para las basadas en BSD es lpr.

Ya en 1993, Novell compró Unix a AT&T, integrando parte de su tecnología NetWare, pasando Unix a llamarse UnixWare y estando basado en System V R4 de AT&T. En 1995, Santa Cruz Operation (SCO) compró Unix a Novell y en 1998 lanzan al mercado la versión 7 de UnixWare, de 64 bits.

Actualmente existe una gran variedad de versiones de Unix. De la familia BSD surgió SunOS (Sun), SPARC-OS (Tatung) y SolOS (Solbourne Computers), y de la familia System V son ZEUS, XENIX (Microsoft), Irix,...

Los Sistemas Unix son muy apreciados debido a su gran fiabilidad y estabilidad, ya que a menudo funcionan durante meses o años sin problemas o caídas del Sistema. Además, ofrecen el mejor rendimiento, las mejores características y el soporte más fácil para la mayoría de los usuarios. Unix proporciona una gran versatilidad que le permite servir como estación de trabajo de un único usuario o como servidor, pudiendo soportar centenares o miles de usuarios que acceden a la vez a una base de datos común.

Unix ofrece también un gran número de utilidades, herramientas de tratamiento de redes y programación, así como aplicaciones personalizadas para casi todos los negocios e industrias.

Pero toda esta historia no está carente de posiciones filosóficas, ya que algunos de los pioneros en sistemas unix empezaron a encontrar problemas para poder trabajar. El problema de Richard Stallman con una impresora fue el detonante del proyecto GNU (GNU No es Unix, ante todo sentido del humor) en 1983 con el propósito de crear un sistema operativo similar y compatible con UNIX y los estándares POSIX. Dos años más tarde, 1985, creó la Fundación del Software Libre (FSF) y desarrolló la Licencia pública general de GNU (GNU GPL), para tener un marco legal que permitiera difundir libremente el software. De este modo el software de GNU fue desarrollado muy rápidamente, y por muchas personas. A corto plazo, se desarrolló gran cantidad de programas, de modo que a principios de los años 1990 había bastante software disponible como para crear un sistema operativo completo. Sin embargo, todavía le faltaba un núcleo.

Éste debía ser desarrollado en el proyecto GNU Hurd, pero Hurd demostró desarrollarse muy inactivamente, porque encontrar y reparar errores (eliminación de fallos, debugging en inglés) era muy difícil, debido a las características técnicas del diseño del micronúcleo. En la actualidad se puede probar Debian Hurd y todas las limitaciones de este kernel.

En 1991, en Helsinki, Linus Torvalds comenzó un proyecto que más tarde llegó a ser el núcleo Linux. El sistema operativo que él usó durante el desarrollo fue Minix, y el compilador inicial fue el GNU C compiler, que aún es la opción principal para compilar Linux hoy. Funcionaba con el procesador 80386 y fue licenciado bajo GPL. Desde ahí empezó todo este maravilloso mundo.

La Free Software Foundation reconoce otras licencias libres como la BSD, Apache, Mozilla o CDDL entre otras

Pero veremos todo esto con más detalle, a continuación

¿Qué es GNU/Linux?

Linux es un sistema operativo: un conjunto de programas que permiten interactuar con el ordenador y ejecutar otros programas.

Un sistema operativo consiste en varios programas fundamentales que necesita el ordenador para poder comunicar y recibir instrucciones de los usuarios; tales como leer y escribir datos en el disco duro, cintas, e impresoras; controlar el uso de la memoria; y ejecutar otros programas. La parte más importante de un sistema operativo es el núcleo. En un sistema GNU/Linux, Linux es el núcleo. El resto del sistema consiste en otros programas, muchos de los cuales fueron escritos por o para el proyecto GNU. Dado que el núcleo de Linux en sí mismo no forma un sistema operativo funcional, preferimos utilizar el término “GNU/Linux” para referirnos a los sistemas que la mayor parte de las personas llaman de manera informal “Linux”.

Linux está modelado como un sistema operativo tipo Unix. Desde sus comienzos, Linux se diseñó para que fuera un sistema multitarea y multiusuario. Estos hechos son suficientes para diferenciar a Linux de otros sistemas operativos más conocidos. Sin embargo, Linux es más diferente de lo que pueda imaginar. Nadie es dueño de Linux, a diferencia de otros sistemas operativos. Gran parte de su desarrollo lo realizan voluntarios de forma altruista.

En 1984 comenzó el desarrollo de lo que más tarde sería GNU/Linux cuando la Free Software Foundation (Fundación de software libre, N. del t.) comenzó a desarrollar un sistema operativo libre de tipo Unix, llamado GNU.

El proyecto GNU ha desarrollado un conjunto de herramientas de software libre para ser utilizados por Unix™ y sistemas operativos tipo Unix como Linux. Estas herramientas permiten a los usuarios desarrollar tareas que van desde las mundanas (como copiar o eliminar ficheros del sistema) a las arcanas (como escribir y compilar programas o hacer edición sofisticada en una gran variedad de formatos de documento).

Aunque hay muchos grupos e individuos que han contribuido a Linux, la Free Software Foundation ha sido quien más ha contribuido. No sólo creó la mayor parte de las herramientas que se utilizan en Linux sino también la filosofía y comunidad que hizo que Linux fuera posible.

El núcleo Linux apareció por primera vez en 1991, cuando un estudiante de informática finlandés llamado Linus Torvalds anunció en el grupo de noticias de USENET comp.os.minix, una primera versión de un núcleo de reemplazo para Minix. Para más referencias consulte la página de historia de Linux en Linux Internacional.

Linus Torvalds sigue coordinando el trabajo de varios cientos de desarrolladores con la ayuda de cierto número de responsables de subsistemas. Existe una página oficial del núcleo Linux. Se puede encontrar un excelente resumen semanal de las discusiones en la lista de correo linux-kernel en Kernel Traffic. Se puede encontrar más información sobre la lista de correo linux-kernel en el documento PUF de la lista de correo «linux-kernel».

Los usuarios de Linux tienen una gran libertad al elegir sus programas. Por ejemplo, un usuario de Linux puede elegir entre docenas de distintos intérpretes de línea de órdenes y entre distintos entornos de escritorio. Tantas opciones confunden a veces a los usuarios de otros sistemas operativos que no están acostumbrados a poder modificar el intérprete de línea de órdenes o el entorno de escritorio.

Es menos probable que un sistema Linux se colapse, además tiene mejor capacidad para ejecutar múltiples programas al mismo tiempo y es más seguro que muchos otros sistemas operativos. Debido a estas ventajas, Linux es el sistema operativo que ha experimentado mayor crecimiento en el mercado de los servidores.

<http://www.debian.org/releases/stable/s390/ch01s02.html.es>

Linux es uno de los principales ejemplos de software libre y de código abierto, pero quizás nos preguntemos que significa ésto. El software libre y de código abierto es el que está licenciado de tal manera que los usuarios pueden estudiar, modificar y mejorar su diseño mediante la disponibilidad de su código fuente.

Los conceptos de software libre y software de código abierto, que, si bien comparten modelos de desarrollo similares, tienen diferencias en sus aspectos filosóficos que destaca la Free Software Foundation.¹ El software libre se enfoca en las libertades filosóficas que les otorga a los usuarios mientras que el software de código abierto se enfoca en las ventajas de su modelo de desarrollo.

Un breve esquema :

Las 4 libertades del software libre

Ejecutar el programa con cualquier propósito (libertad 0)
(privado, educativo, público, comercial, militar, etc.)

Estudiar y modificar el programa (libertad 1)
(para lo cual es necesario poder acceder al [código fuente](#))

Distribuir el programa de manera que se pueda ayudar al próximo (libertad 2)

Distribuir las versiones modificadas propias (libertad 3)
(para lo cual es necesario poder acceder al [código fuente](#))

Las 10 premisas del software de código abierto

Libre redistribución: el software debe poder ser regalado o vendido libremente.

Código fuente: el código fuente debe estar incluido u obtenerse libremente.

Trabajos derivados: la redistribución de modificaciones debe estar permitida.

Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.

Sin discriminación de personas o grupos: nadie puede dejarse fuera.

Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.

Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa.

La licencia no debe ser específica de un producto: el programa no puede licenciarse solo como parte de una distribución mayor.

La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.

La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un

acceso por clic de ratón o de otra forma específica del medio de soporte del software.

Linux está licenciado bajo la GPL v2 y está desarrollado por colaboradores de todo el mundo. Sobre la licencia Pública General dice wikipedia:

“La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o simplemente sus siglas del inglés GNU GPL) es la licencia más ampliamente usada1 en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Esta licencia fue creada originalmente por Richard Stallman fundador de la Free Software Foundation (FSF) para el proyecto GNU (GNU project).”

La licencia GPL puede ser usada por cualquiera, su finalidad es proteger los derechos de los usuarios finales (usar, compartir, estudiar, modificar). Esta es la primera licencia copyleft para uso general. Copyleft significa que los trabajos derivados sólo pueden ser distribuidos bajo los términos de la misma licencia. Bajo esta filosofía, la licencia GPL garantiza a los destinatarios de un programa de ordenador los derechos-libertades reunidos en definición de software libre (free software definition) y usa copyleft para asegurar que el software está protegido cada vez que el trabajo es distribuido, modificado ó ampliado. En la forma de distribución (sólo pueden ser distribuidos bajo los términos de la misma licencia) se diferencian las licencias GPL de las licencias de software libre permisivas (permissive free software licenses), de las cuales los ejemplos más conocidos son las licencias BSD (BSD licenses).

El software bajo licencia GPL puede ser aplicado bajo todos los propósitos (incluidos los propósitos comerciales e incluso como herramienta de creación de software propietario (proprietary software)). En uso puramente privativo (ó interno) - sin ventas ni distribuciones implicadas - el software puede ser modificado sin liberar el código fuente (por contra, si hay implicadas ventas o distribuciones, el código fuente y cualquier cambio realizado en él debe estar disponible para los usuarios, ya que en este caso los derechos del usuario están protegidos por copyleft). De esta forma, las aplicaciones instaladas en sistemas operativos bajo licencia GPL como Linux, no es necesario que estén licenciadas bajo GPL ó que estén distribuidas con su código fuente disponible ya que las licencias no dependen de la plataforma. Por ejemplo, si un programa está formado completamente por código original, ó si está combinado con software que no se cumple los requisitos de copyleft no es necesario que se licencie bajo GPL ó que se distribuya con su código fuente disponible. Sólo si un programa utiliza fragmentos de código GPL (y el programa es distribuido) el código fuente en su totalidad debe estar disponible (bajo la misma licencia). La licencia LGPL (GNU Lesser General Public License) fue creada para tener derechos menos restrictivos que GPL, en este caso en un programa que utiliza fragmentos de código LGPL, no es necesario liberar el código original.

Los usuarios o compañías que distribuyen sus trabajos bajo licencias GPL, pueden cobrar ó distribuirlas gratuitamente. Esto distingue las licencias GPL de las licencias software que prohíben su distribución comercial. La FSF argumenta que no se debe restringir la distribución comercial del software (incluyendo la redistribución), la GPL establece explícitamente que las obras GPL se puede vender a cualquier precio”.

Software libre de uso habitual.

Conociendo el concepto de software libre ligado a la licencia (eso que nunca se suele leer) es interesante que hagamos mención de muchas aplicaciones que pueden funcionar bajo cualquier sistema operativo y cumplen con este requisito de licenciamiento.

Ofimática

En el ámbito de la ofimática podemos encontrar OpenOffice y un fork que le ganó el terreno, LibreOffice.

(extraído de la web oficial)

WRITER es el procesador de textos dentro de LibreOffice. Se usa para todo, desde garabatear una carta rápida a producir un libro entero con las tablas de contenido, ilustraciones embebidas, bibliografía y diagramas. El auto completado mientras escribe, el formato y la revisión ortográfica automáticos hacen fáciles tareas difíciles (pero son sencillos de desactivar si lo prefiere). Writer es suficientemente potente como para hacer frente a las tareas de autoedición como la creación de boletines de varias columnas y folletos. El único límite es su imaginación.

CALC doma sus números y ayuda a tomar decisiones difíciles cuando evalúa alternativas. Analice los datos con Calc y luego usarlos para presentar su resultado final. Gráficas y herramientas de análisis ayudan a dar transparencia a sus conclusiones. Un sistema de ayuda totalmente integrado hace más fácil el trabajo de introducir fórmulas complejas. Agregue datos de bases de datos externas como SQL u Oracle, luego ordenelos y filtrelos para producir análisis estadísticos. Utilice las funciones de graficación para mostrar gran cantidad de gráficos 2D y 3D de 13 categorías, incluyendo líneas, áreas, barras, circulares, X-Y, y red - con docenas de variaciones disponibles, usted puede estar seguro de encontrar una que se adapte a su proyecto.

IMPRESS es la manera más sencilla y rápida de crear presentaciones multimedia efectivas. Impresionantes efectos de animación y sensacionales efectos especiales le ayudarán a convencer a su público. Cree presentaciones que se vean aún más profesionales que aquellas presentaciones estándar que comúnmente ve en el trabajo. Obtenga atención de sus colegas y jefes mediante la creación de algo un poco diferente.

DRAW le permite crear diagramas y dibujos a partir de cero. Una imagen vale más que mil palabras, así que ¿por qué no intentar algo simple con diagramas de cajas y líneas? O bien ir más allá y construir fácilmente ilustraciones dinámicas 3D y efectos especiales. Es tan simple o tan potente como quiera que sea.

BASE es la interfaz de base de datos de la suite LibreOffice. Con Base, puede integrar perfectamente sus estructuras de base de datos existente. Partiendo de tablas importadas y vinculados y consultas de MySQL, PostgreSQL o Microsoft Access y muchas otras fuentes de datos, puede crear potentes bases de datos que contengan formularios, informes, vistas y consultas. La integración completa es posible con la base de datos HSQL incorporada.

MATH es un editor de ecuaciones simple que le permite diseñar y mostrar sus ecuaciones matemáticas, químicas, eléctricas o científicas rápidamente en notación escrita estándar. Incluso los cálculos más complejos pueden ser comprensibles cuando se muestran correctamente. $E=mc^2$.

LibreOffice también viene configurado con un creador de archivos PDF, lo que significa que puede distribuir documentos que puede estar seguro pueden ser abiertos y leídos por los usuarios de casi

cualquier dispositivo informático o sistema operativo.

Navegadores

El mundo de los navegadores estuvo dominado por Firefox, Iceweasel o Chromium. En las últimas estadísticas del año 2013, Firefox se sitúa como el segundo navegador más utilizado por detrás de Google Chrome (versión privativa de Chromium) y es el que incorpora en primer lugar los estándares abiertos.

Edición de imágenes

GIMP es una herramienta de manipulación fotográfica multiplataforma. GIMP es un acrónimo de GNU Image Manipulation Program. En GIMP se pueden realizar todo tipo de tareas de manipulación de imágenes, incluyendo retoque fotográfico, composición de imágenes y creación de imágenes.

GIMP tiene muchas capacidades. Se puede usar como un sencillo programa de pintura, un programa de retoque fotográfico profesional, un sistema en línea de proceso por lotes, un generador de imágenes para producción en masa, un conversor de formatos de imágenes, etc.

GIMP es ampliable y extensible. Está diseñado para ampliarse con complementos y extensiones. La interfaz avanzada de guionado (scripting) permite automatizar desde las tareas más simples hasta los procedimientos más complejos de manipulación de imágenes.

Uno de los fuertes de GIMP es su libre disponibilidad desde varias fuentes para muchos sistemas operativos. Casi todas las distribuciones de GNU/Linux incluyen al GIMP como una aplicación estándar. El GIMP también está disponible para otros sistemas operativos como Microsoft Windows™ o Mac OS X™(Darwin) de Apple. GIMP es una aplicación de Software Libre cubierta por la Licencia Pública General .

Edición de audio

Audacity es un editor de audio libre, fácil de usar , multipista para Windows, Mac OS X, GNU/Linux y otros sistemas operativos. El interface está traducido a varios idiomas. Puede usar Audacity para:

- Grabar audio en vivo.
- Grabar el sonido que se esté escuchando en el equipo si utiliza Windows Vista o superior.
- Convertir cintas y grabaciones a sonido digital o CD.
- Editar archivos WAV, AIFF, FLAC, MP2, MP3 y Ogg Vorbis.
- Cortar, copiar, unir y mezclar sonidos.
- Cambiar la velocidad o el tono de una grabación.
- Y mucho más. Vea la lista completa de [funciones](#).

Animación

Blender es un programa informático multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales. El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre. Actualmente es compatible con todas las versiones de Windows, Mac OS X, GNU/Linux, Solaris, FreeBSD e IRIX.

Tiene una muy peculiar interfaz gráfica de usuario, que se critica como poco intuitiva, pues no se basa en el sistema clásico de ventanas; pero tiene a su vez ventajas importantes sobre éstas, como la configuración personalizada de la distribución de los menús y vistas de cámara.

Servidores web

El **servidor HTTP Apache** es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, *a patchy server* (un servidor "parcheado") suena igual que *Apache Server*.

Otros productos con un futuro prometedor pueden ser NGINX, Lighttpd o Cherokee

Bases de datos

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

MariaDB es un *fork* directo de MySQL que asegura que permanecerá una versión de este producto con licencia GPL.

PostgreSQL es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

Servicio de mail

Postfix es un servidor de correo de software libre / código abierto, un programa informático para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura al ampliamente utilizado Sendmail. Anteriormente conocido como **VMailer** e **IBM Secure Mailer**, fue originalmente escrito por Wietse Venema durante su estancia en el Thomas J. Watson Research Center de IBM, y continúa siendo desarrollado activamente.

Postfix es el agente de transporte por omisión en diversas distribuciones de Linux y en las últimas versiones del Mac OS X.

Servicio DNS

BIND (*Berkeley Internet Name Domain*, anteriormente: *Berkeley Internet Name Daemon*) es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix, en los cuales es un Estándar de facto. Es patrocinado por la Internet Systems Consortium. BIND fue creado originalmente por cuatro estudiantes de grado en la University of California, Berkeley y liberado por primera vez en el 4.3BSD. Paul Vixie comenzó a mantenerlo en 1988 mientras trabajaba para la DEC.

Una nueva versión de BIND (BIND 9) fue escrita desde cero en parte para superar las dificultades arquitectónicas presentes anteriormente para auditar el código en las primeras versiones de BIND,

Servicio de directorio

OpenLDAP es una implementación libre y de código abierto del protocolo Lightweight Directory Access Protocol (LDAP) desarrollada por el proyecto OpenLDAP.

Está liberada bajo su propia licencia OpenLDAP Public License. LDAP es un protocolo de comunicación independiente de la plataforma.

Muchas distribuciones GNU/Linux incluyen el software OpenLDAP para el soporte LDAP. Este software también corre en plataformas BSD, AIX, HP-UX, Mac OS X, Solaris, Microsoft Windows (NT y derivados, incluyendo 2000, XP, Vista), y z/OS.

Lenguajes de programación

C es el lenguaje de desarrollo usado para elaborar unix usa licencia GPL al igual que Perl u otros más populares como PHP o Python (con sus propias licencias libres).

El número de aplicaciones de código abierto y software libre es impresionante y todos las usamos e forma habitual. Como muestra, buena parte de la información anterior proviene de una aplicación libre conocida por todos: wikipedia.

¿Qué son las distribuciones de linux?

Linux es un sistema de libre distribución por lo que podéis encontrar todos los ficheros y programas necesarios para su funcionamiento en multitud de servidores conectados a Internet. La tarea de reunir todos los ficheros y programas necesarios, así como instalarlos en tu sistema y configurarlo, puede ser una tarea bastante complicada y no apta para muchos. Por esto mismo, nacieron las llamadas distribuciones (distros de forma coloquial) de Linux, empresas y organizaciones que se dedican a hacer el trabajo "sucio" para nuestro beneficio y comodidad.

Una distribución no es otra cosa, que una recopilación de programas y ficheros, organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet, o comprando los CDs de las mismas, los cuales contendrán todo lo necesario para instalar un sistema Linux bastante completo y en la mayoría de los casos un programa de instalación que nos ayudara en la tarea de una primera instalación.

Se puede consultar en Internet la existencia de centenares de distribuciones que han derivado de unas pocas. Fundamentalmente existen dos familias: las derivadas de Red Hat y las derivadas de Debian, estas son las que trataremos en este manual.

DEBIAN

Ian Murdock fundó oficialmente el proyecto Debian el 16 de agosto de 1993. Hasta ese momento, el concepto de una «distribución» de Linux era nuevo. Ian pretendió que Debian fuera una distribución realizada de forma abierta, siguiendo el espíritu de Linux y GNU (lea el manifiesto provisto como un apéndice a este documento para más detalles). La creación de Debian fue patrocinada por el proyecto GNU de la FSF durante un año (noviembre de 1994 a noviembre de 1995).

Debian estaba pensada para ser desarrollada cuidadosa y conscientemente y ser mantenida y soportada con un cuidado similar. Lo que comenzó con un pequeño y grupo muy unido de hackers de software libre, fue creciendo gradualmente hasta convertirse en una gran comunidad de desarrolladores y usuarios bien organizada.

Debian es la única distribución que está abierta a las contribuciones de cada desarrollador y usuario que deseen participar con su trabajo. Y es la única distribución relevante de Linux que no es una entidad comercial. Es el único gran proyecto con una constitución, contrato social, y documento de directrices que organizan el proyecto. Debian es también la única distribución que se «microempaquetá» y que utiliza una detallada información de las dependencias de cada paquete con respecto a otros para asegurar la consistencia del sistema cuando tiene lugar una actualización.

Debian ha adoptado un gran conjunto de directrices y procedimientos para el empaquetamiento y la distribución de software para poder alcanzar y mantener altos estándares de calidad. Se producen herramientas, sistemas automáticos y documentación de cada uno de los aspectos claves de Debian de una forma abierta y visible para poder sostener estos estándares.

Debian marca unas interesantes directrices con su contrato social:

Debian permanecerá 100% libre

“ "Las directrices de software libre de Debian" (DFSG) son el criterio que nosotros utilizamos para determinar si el software es "libre" o no. Prometemos mantener el sistema así como todos sus componentes completamente libres de acuerdo con este criterio. No obstante, daremos soporte también a aquellos usuarios que desarrollen y ejecuten software no libre en Debian pero nunca haremos que el sistema tenga que utilizar obligatoriamente un componente que no sea libre.

Contribuiremos a la comunidad de software libre

Cuando escribamos nuevos componentes del sistema Debian, los licenciaremos de forma consistente con nuestra definición de software libre. Haremos el mejor sistema que podamos, de forma que el software libre tenga amplia difusión y uso. Enviaremos parches, mejoras, peticiones de los usuarios, etc. a los autores originales (esto se conoce en inglés como "upstream", N. del T.) del software incluido en nuestro sistema.

No ocultaremos los problemas

Mantendremos nuestra base de datos de informes de error accesible al público en todo momento. Los informes de error que los usuarios envíen serán visibles por el resto de usuarios de forma inmediata.

Nuestra prioridad son nuestros usuarios y el software libre

Nos guiarímos por las necesidades de nuestros usuarios y de la comunidad del software libre. Sus intereses serán una prioridad para nosotros. Daremos soporte a las necesidades de nuestros usuarios para que puedan trabajar en muchos tipos distintos de entornos de trabajo. No pondremos objeciones al software no libre que vaya a ejecutarse sobre Debian ni cobraremos a las personas que quieran desarrollar o usar ese tipo de software (no libre). Permitiremos a otros crear distribuciones de valor añadido basadas en Debian sin cobrarles nada por ello. Es más, entregaremos un sistema integrado de alta calidad sin restricciones legales que pudieran prevenir este tipo de uso.

Trabajos que no siguen nuestros estándares de software libre

Reconocemos que algunos de nuestros usuarios necesitan usar trabajos que no sigan las directrices de software libre de Debian (DFSG). Por ello, hemos creado las secciones "contrib" y «non-free» en nuestro archivo para estos trabajos. Los paquetes en estas secciones no son parte del sistema Debian, aunque han sido configurados para usarse con Debian. Animamos a los distribuidores de CDs a que lean las licencias de los paquetes en estas secciones para poder determinar si pueden distribuir este software en sus CDs. Así pues, aunque los trabajos que no sean libres no son parte de Debian, damos soporte para su uso, y proporcionamos infraestructura (como nuestro sistema de informe de errores y listas de distribución) para paquetes no libres.”

RED HAT

Red Hat Inc. es la compañía responsable de la creación y mantenimiento de una distribución del sistema operativo GNU/Linux que lleva el mismo nombre: Red Hat Enterprise Linux, y de otra más, Fedora. Así mismo, en el mundo del middleware patrocina jboss.org, y distribuye la versión profesional bajo la marca JBoss Enterprise.

Red Hat es famoso en todo el mundo por los diferentes esfuerzos orientados a apoyar el movimiento del software libre. No sólo trabajan en el desarrollo de una de las distribuciones más populares de Linux, sino también en la comercialización de diferentes productos y servicios basados en software de código abierto. Asimismo, poseen una amplia infraestructura en la que se cuentan más de 2.000 empleados en 28 lugares del mundo.

Programadores empleados de Red Hat han desarrollado múltiples paquetes de software libre, los cuales han beneficiado a toda la comunidad. Algunas de las contribuciones más notables han sido la creación de un sistema de empaquetación de software (RPM), y varias utilidades para la administración y configuración de equipos, como sndconfig o mouseconfig.

Ahora bien, Red Hat distribuye el código fuente de la distribución excepto de aquellas herramientas en las que la licencia no lo exige y tiene un alto coste. Por esta causa varias comunidades han optado por elaborar sus propias distribuciones basadas en este código fuente y se las denomina clones. Es por ello que mantienen compatibilidad binaria con RHEL y tienen por ello un alto índice de uso a nivel empresarial. En este curso usaremos una de las más extendidas llamada CentOS.

CENTOS

CentOS (Community ENTerprise Operating System) es, como ya hemos comentado, una distro compilada por voluntarios a partir del código fuente liberado por Red Hat. Su orientación es ,

fundamentalmente, hacia la empresa. La numeración de las versiones de esta distro es coincidente con las sacadas por RH.

CentOS tiene numerosas ventajas sobre algunos de los proyectos de otros clones que incluyen:

1. La principal ventaja es que se obtiene un conjunto estable de la mayoría de paquetes que por lo general solo incluyen correcciones de errores.
2. Una comunidad de usuarios activa y creciente, reconstruido rápidamente, probado.
3. Una extensa red de servidores espejos (mirrors), los desarrolladores que están localizables y sensible, múltiples vías de apoyo gratuitos, como el IRC Chat en vivo, las listas de correo, Foros , una dinámica de preguntas frecuentes.
4. Esta dirigido a personas que buscan la estabilidad de clase empresarial del sistema operativo sin el costo de la certificación y apoyo.

OTRAS DISTRIBUCIONES

Existen muchas distribuciones y entre las más conocidas podemos nombrar:

Ubuntu: quizá la más conocida. Derivada de Debian pero con “vida propia” desarrollada por Canonical y con características muy particulares tratando de difundir los sistemas GNU/linux entre los usuarios finales.

Linux Mint: una derivación de ubuntu y Debian que se ha popularizado en los entornos de escritorio al no adaptar los nuevos entornos gráficos (GNOME Shell y Unity de Ubuntu) que fueron, en un principio, rechazados por la comunidad.

OpenSuse: La versión libre de Suse que tiene su versión empresarial y herramientas propias.

Gentoo: una distribución de nivel avanzado que puede usar tanto el núcleo Linux como un BSD. Los binarios se compilan de forma concreta en vez de descargarse e instalarse como en la mayoría de distros.

Y muchas más que se pueden consultar en http://es.wikipedia.org/wiki/Anexo:Distribuciones_Linux

Tras toda esta vorágine de distribuciones, cabe preguntarse que similitudes tienen y cuales son las diferencias entre ellas. LA FHS (File Hierarchy Standard) o LSB (Linux Standard Base) han sido proyectos de unificación de todas las distribuciones aunque no se ha logrado la homogeneidad absoluta. En los entornos gráficos existen muchas distros con entornos comunes pero en los últimos tiempos los dos grandes proyectos (GNOME, KDE) han perdido adeptos por algunos más ligeros como XFCE o LXDE o particulares como Unity (de Canonical para Ubuntu). Incluso en los servidores de X-Windows comienzan nuevas soluciones como Wayland o Mir frente al viejo X-org. La instalación de paquetes también difiere en función de la derivación debian (.deb) o RHEL (rpm) entre otras.

Por todo ello, el mundo GNU/linux es complejo y debemos conocer las particularidades de la plataforma en la que estemos trabajando.

Por último, señalar que, quizás, el derivado de linux más utilizado sea **Android**, un sistema operativo con licencia Apache y GPL diseñado principalmente para dispositivos móviles con

pantalla táctil, como teléfonos inteligentes o tabletas. Una prometedora unión entre robots y pingüinos ;-)

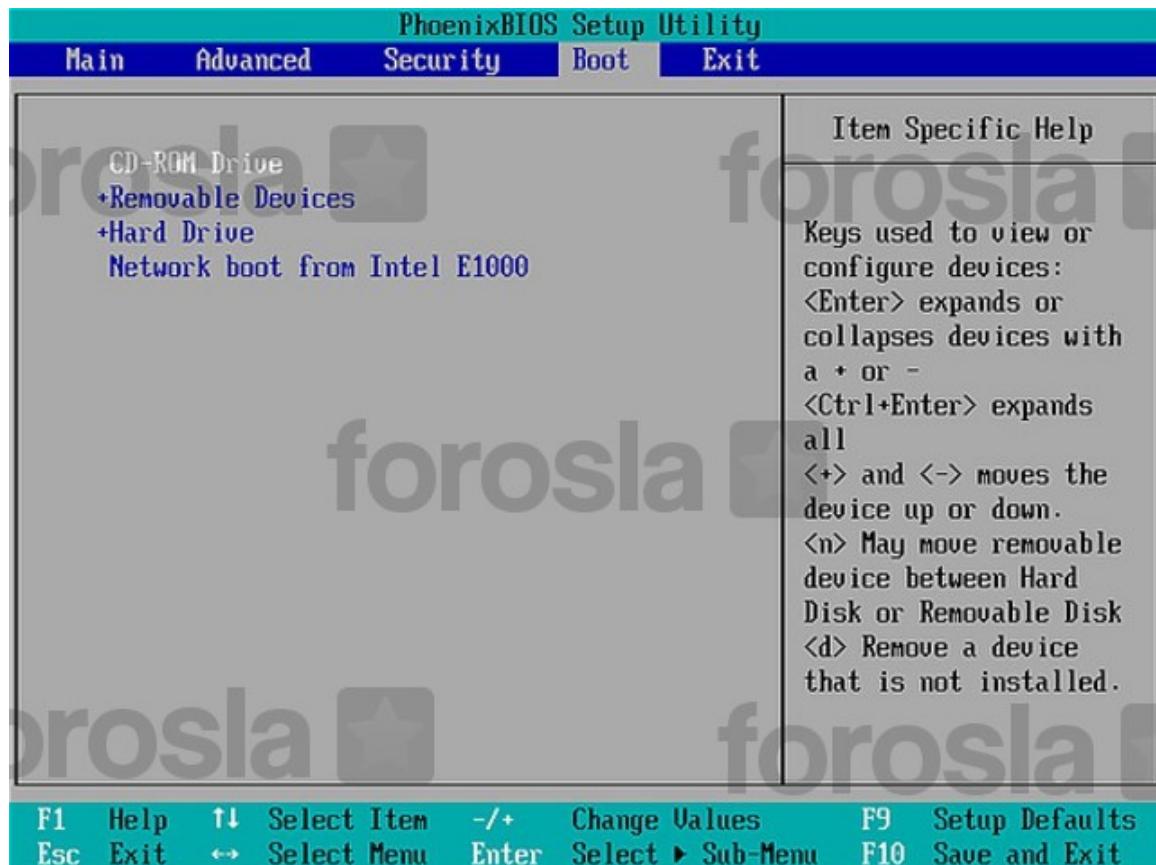
INSTALACIÓN DE SISTEMAS GNU-LINUX

Instalación de Debian Wheezy 7.2

Existen diferentes métodos de instalación de sistemas que se pueden documentar de forma sencilla en la Internet. Se va a proceder a explicar el proceso de instalación desde un dispositivo de CD/DVD de la imagen iso de Debian 7.2 netinstall, esto obliga a tener una conexión de red con la que se cargan los paquetes desde una ubicación de red.

El proceso de instalación.

Este método obliga a indicarle al ordenador que arranque desde un medio extraible. Para ello se ha de acceder a la BIOS y modificar la secuencia de arranque. Este proceso se realiza de manera diferente en función del fabricante del servidor o PC.



La sección en el BIOS en donde se puede configurar el orden de carga.

Menú del Instalador.

Habiendo cargado el sistema desde el medio de instalación, aparecerá una imagen muy similar a la que se muestra abajo. En ella, **Debian** nos ofrece el **Menú** del instalador desde la cual podremos

elegir la interfaz gráfica.

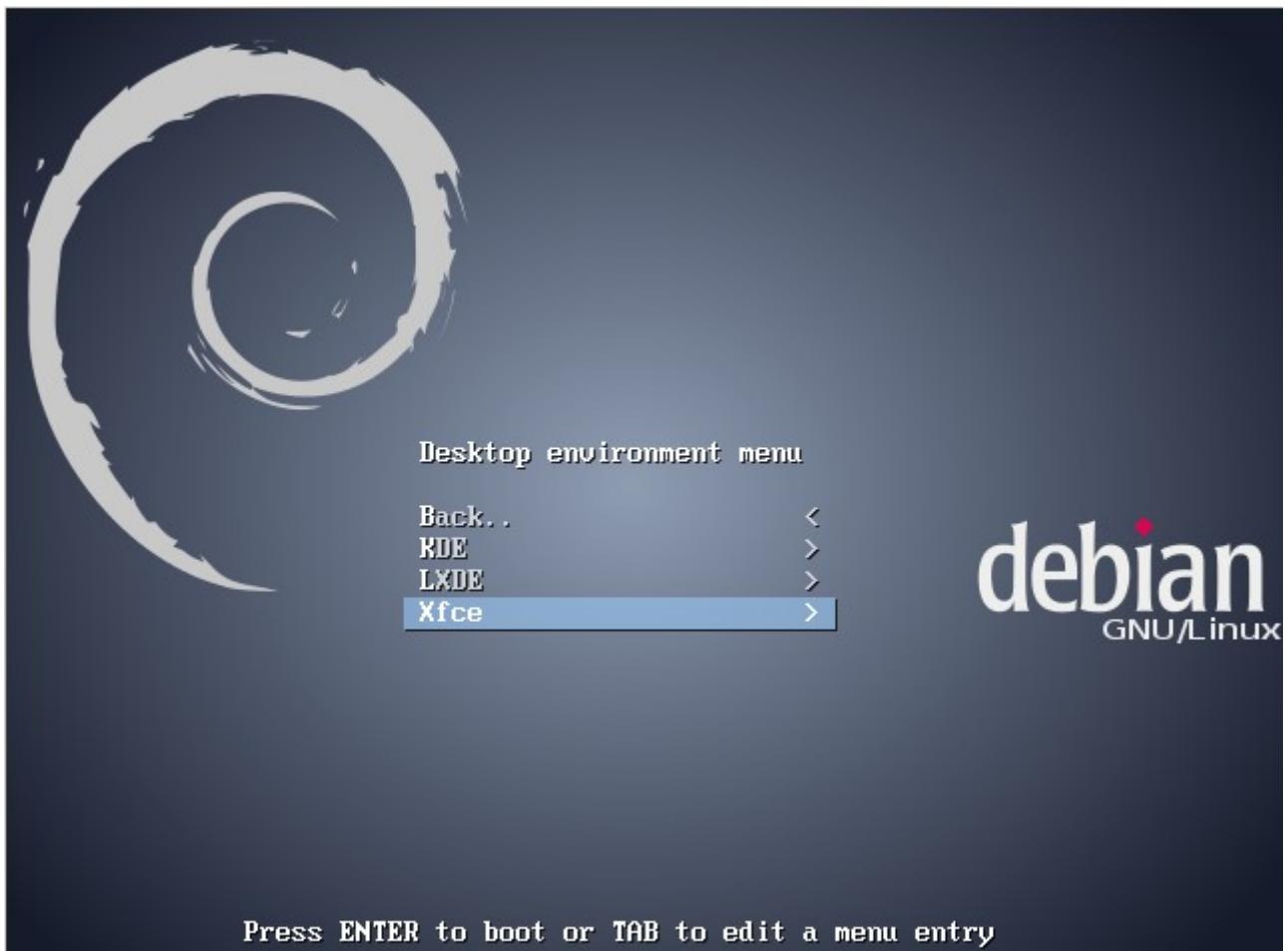
De manera predeterminada (si no se especifica algo distinto), Debian instala Gnome-shell como entorno de escritorio. Si se desea una interfaz diferentemente, haciendo uso de las “flechas” del teclado, se ha de navegar hasta la **Advanced Options** (Opciones Avanzadas) y pulsar **ENTER**.



En este nuevo menú, hay que navegar hasta la opción de **Alternative desktop environments** (Escritorios alternativos o Interfaces gráficas alternativas) y pulsar ENTER.



Existen 3 escritorios adicionales que se pueden escoger por medio del instalador de Debian.
Seleccionar el escritorio y pulsar la tecla **ENTER**.



Una vez seleccionado el escritorio, se debe navegar hasta la opción llamada **Install** y pulsar **ENTER**.



Seleccionando el idioma.

Las pantallas que siguen a continuación, permiten configurar el futuro sistema operativo. En estas pantallas, se puede establecer idioma de la interfaz, teclado, nombres de usuario y contraseñas. Llegado a este punto, es imprescindible una conexión activa de Internet. En la pantalla visible a continuación, seleccionar el lenguaje que se empleará para el escritorio (es decir, los programas, menús, interfaces, etc, todo aparecerá en el lenguaje que escogas) y luego haz clic en **Continue**.

[!!] Select a language

Choose the language to be used for the installation process. The selected language will also be the default language for the installed system.

Language:

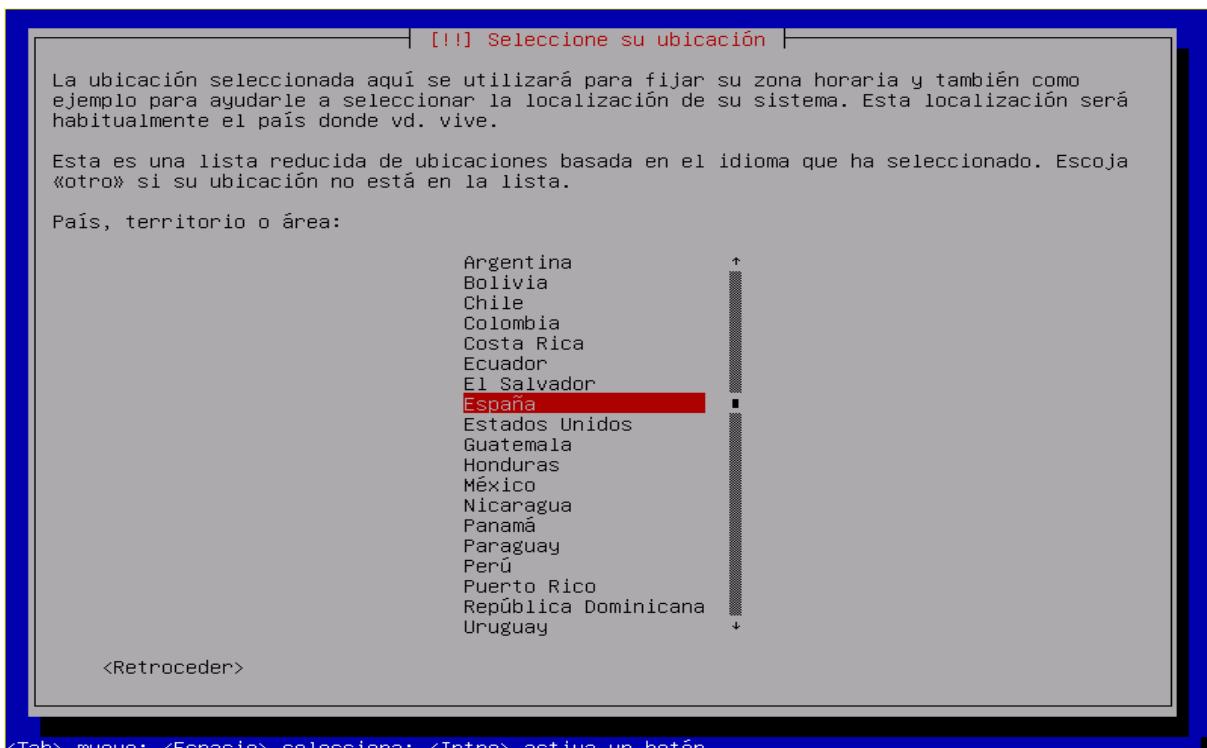
Italian	- Italiano
Japanese	- 日本語
Kazakh	- Қазақ
Korean	- 한국어
Kurdish	- Kurdî
Latvian	- Latviski
Lithuanian	- Lietuviškai
Macedonian	- Македонски
Northern Sami	- Sámegilii
Norwegian Bokmaal	- Norsk bokmål
Norwegian Nynorsk	- Norsk nynorsk
Persian	- فارسی
Polish	- Polski
Portuguese	- Português
Portuguese (Brazil)	- Português do Brasil
Romanian	- Română
Russian	- Русский
Serbian (Cyrillic)	- Српски
Slovak	- Slovenčina
Slovenian	- Slovenščina
Spanish	- Espanol
Swedish	- Svenska
Tagalog	- Tagalog

<Go Back>

<Tab> moves; <Space> selects; <Enter> activates buttons

Selección de la ubicación.

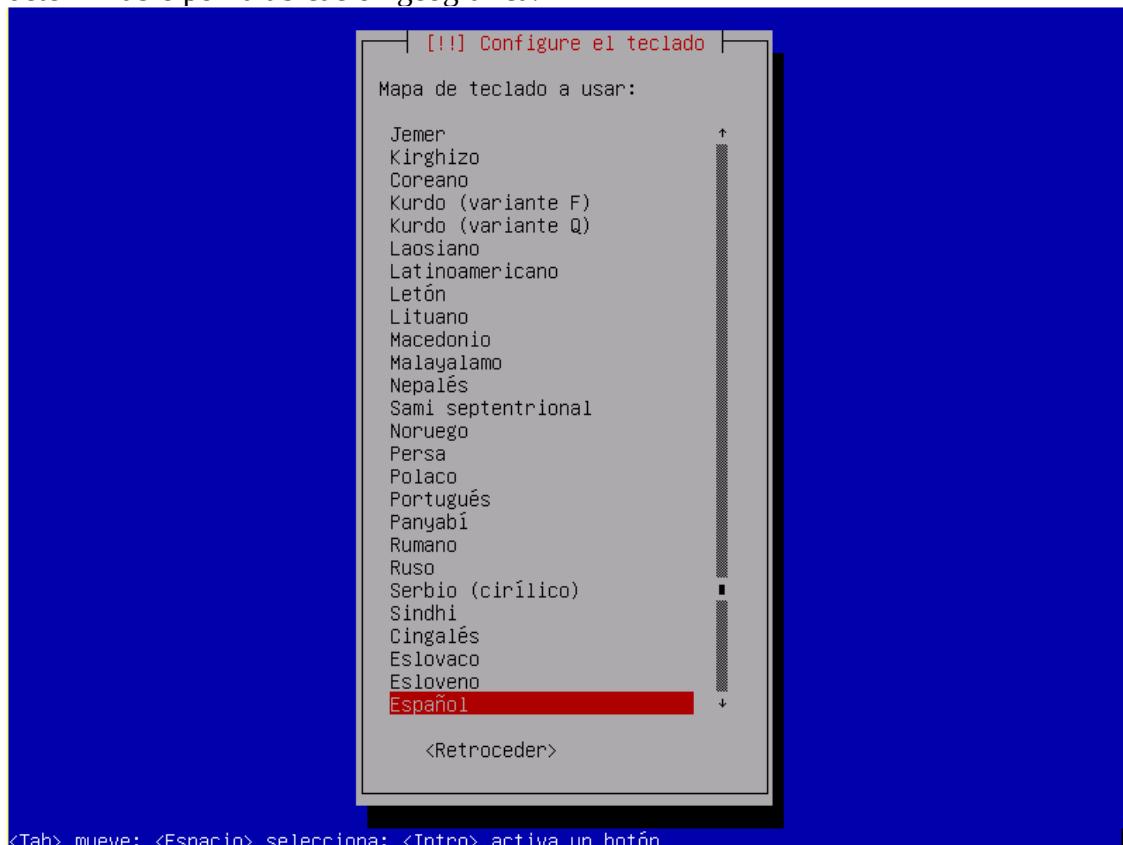
A continuación, seleccionar el país. Esto permite al instalador asignar las opciones de localización (zona horaria, medidas de peso, distancia, moneda, etc) que aplican al país escogido. Seleccionada la ubicación, realizar clic en **Continuar**.



<Tab> mueve; <Espacio> selecciona; <Intro> activa un botón

Distribución del teclado.

En esta pantalla es posible determinar la distribución de teclado que, en principio, no es determinable por la ubicación geográfica.



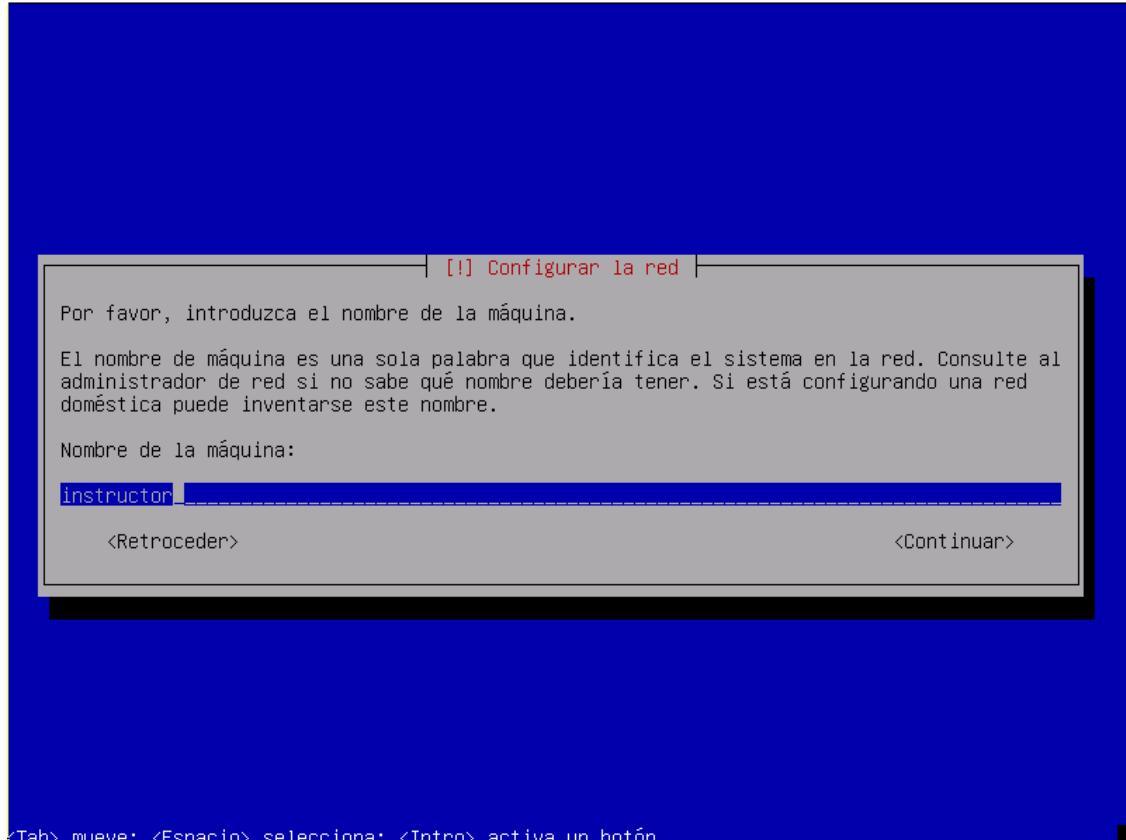
<Tab> mueve; <Espacio> selecciona; <Intro> activa un botón

Trabajando...

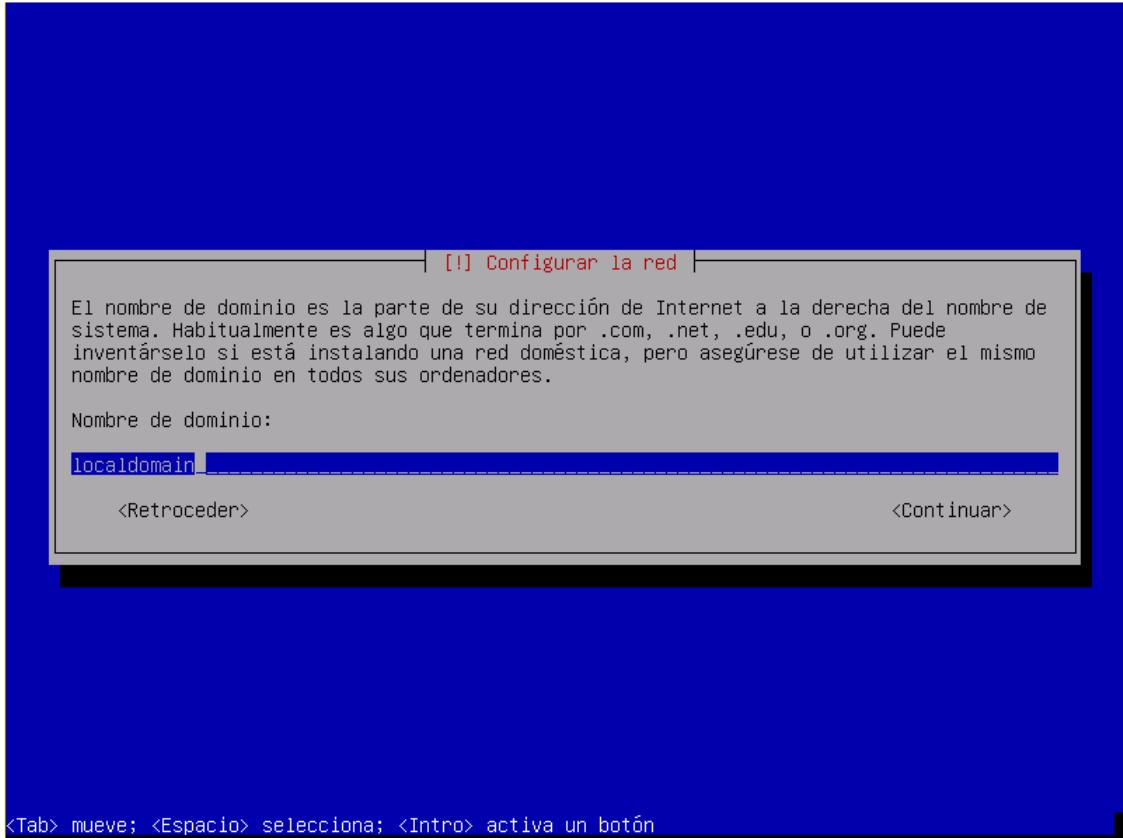
Debian comenzará a descargar o leer del disco los ficheros necesarios para implementar la configuración que se ha definido.

Configuración de la red.

Esta pantalla permite asignar un nombre al equipo. No se admiten espacios en blanco ni caracteres especiales. Establecer un nombre y pulsar **Continuar**.

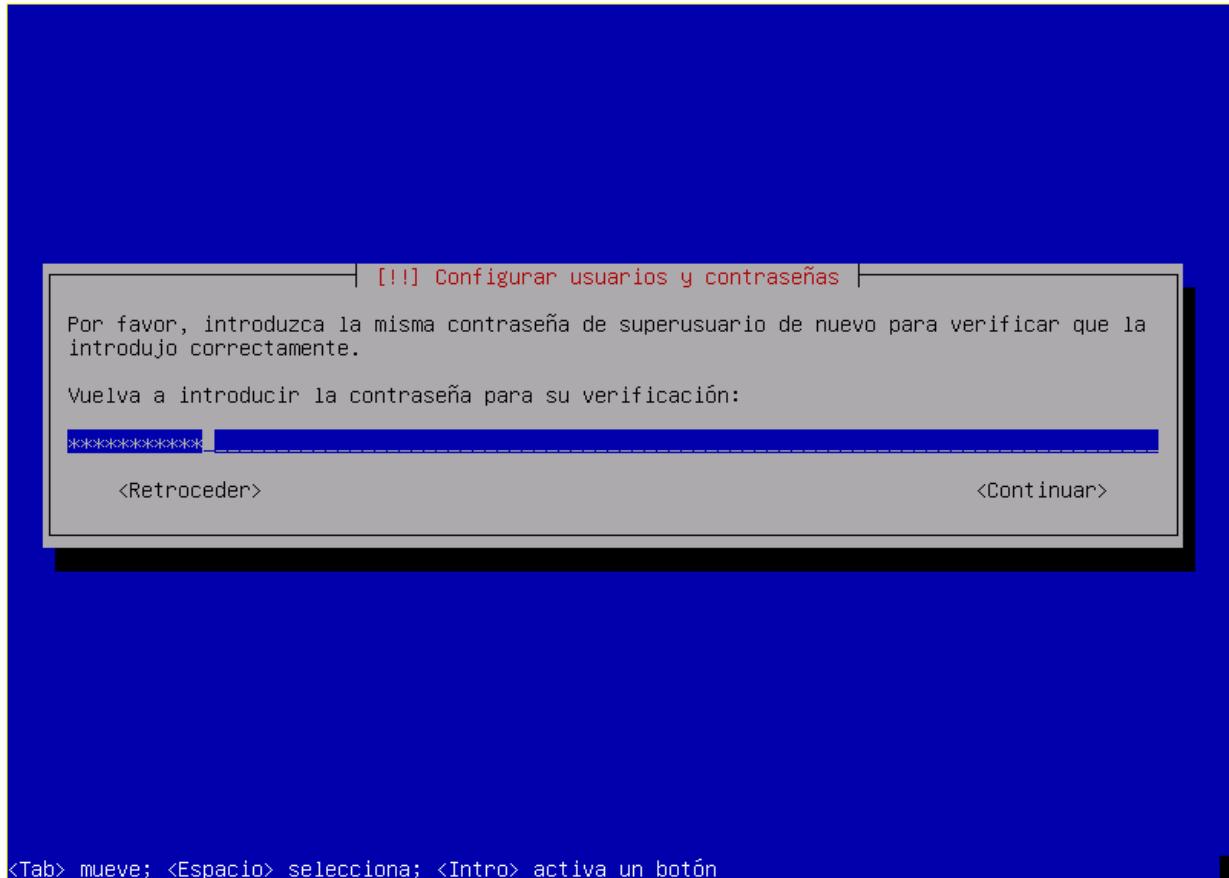


El nombre de dominio es conocido como el sufijo DNS y es muy importante en una red medianamente administrada. Se suele utilizar “localdomain” como una convención cuando no existe un sistema de administración de nombres.



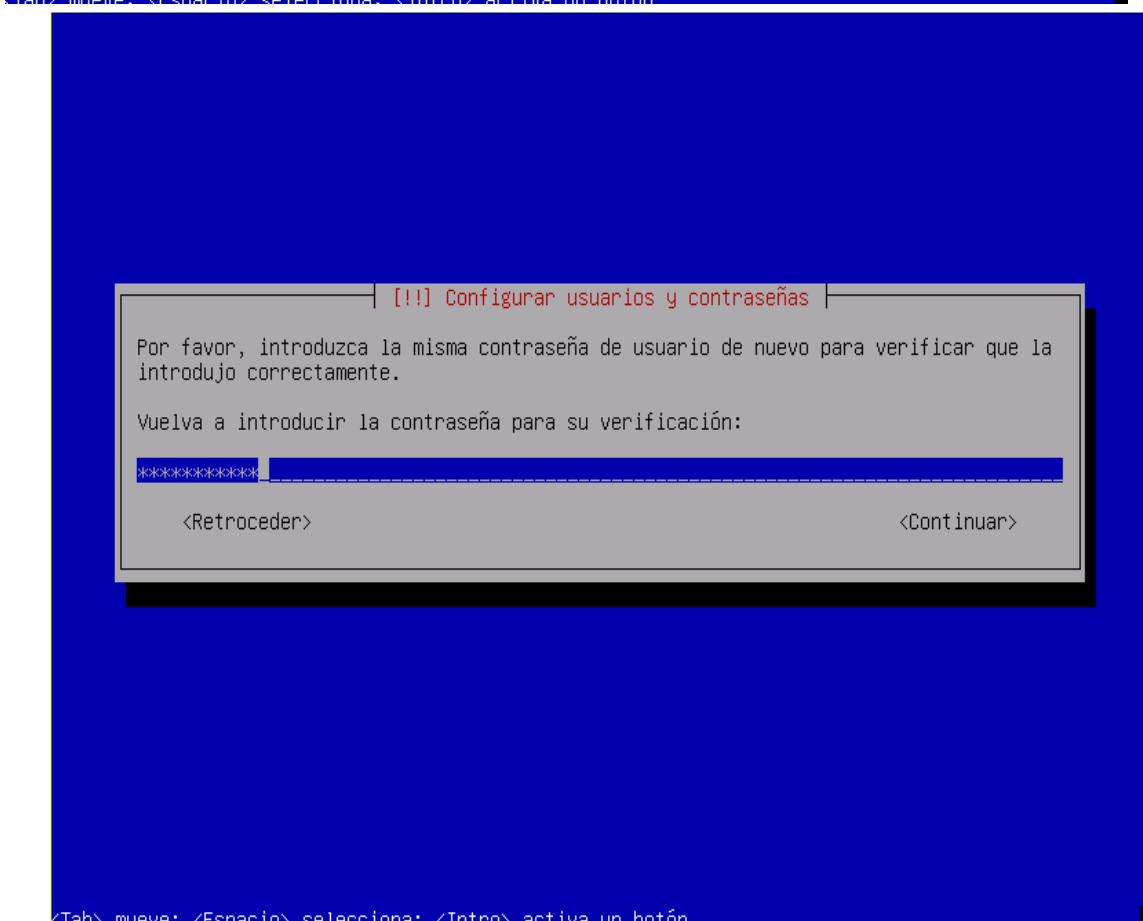
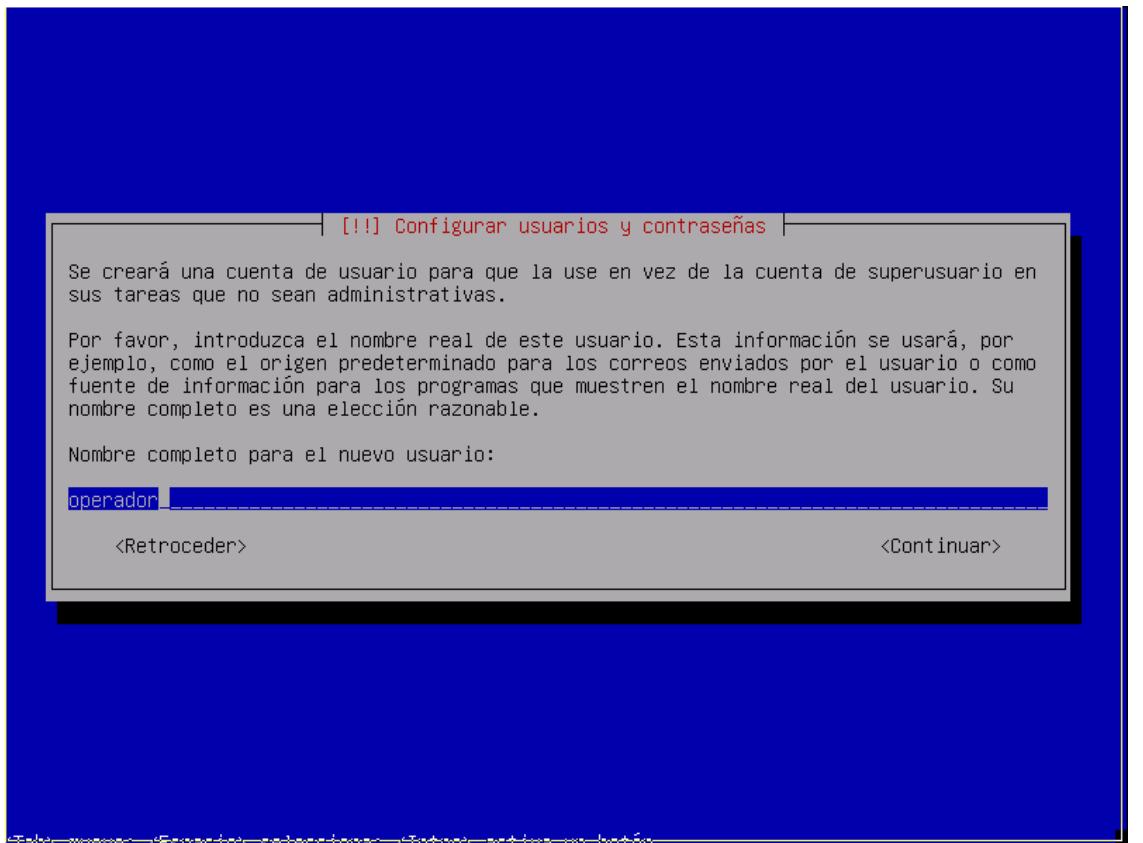
Configurar usuarios y contraseñas

A diferencia de otras distribuciones, Debian permite dejar la contraseña de root “en blanco” lo que supone activar sudo. Esta herramienta permite ejecutar aplicaciones de root por usuarios sin privilegios



<Tab> mueve; <Espacio> selecciona; <Intro> activa un botón

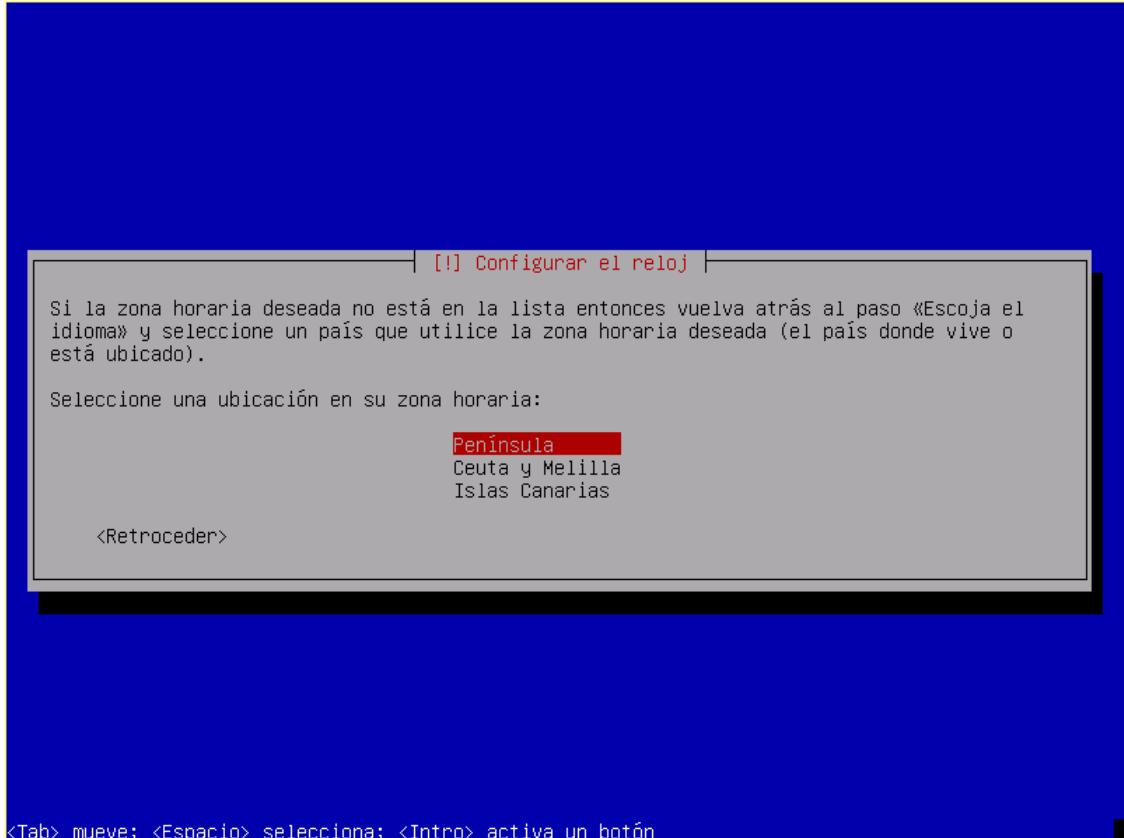
Se solicita la creación de una cuenta de usuario no privilegiado.



<Tab> mueve; <Espacio> selecciona; <Intro> activa un botón

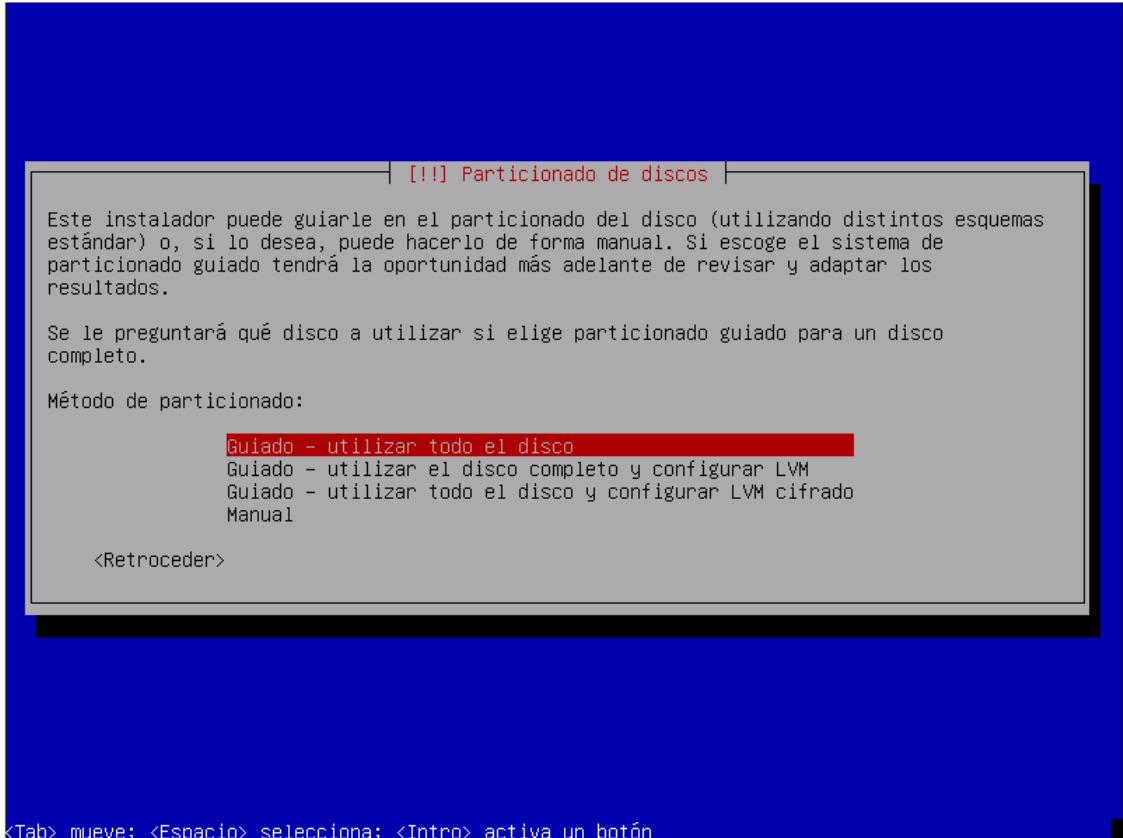
Configuración del reloj.

Selección de la zona horaria en función de la ubicación de referencia



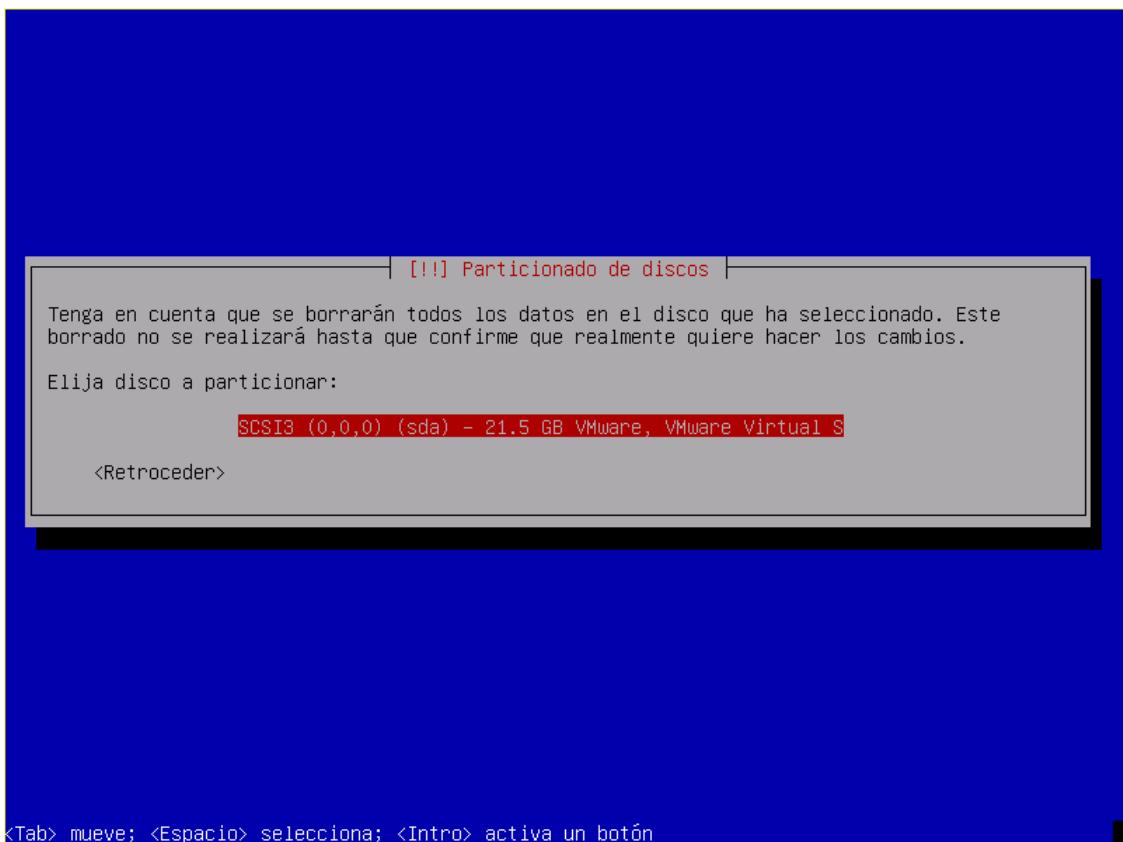
Detección y particionado de discos.

Debian comenzará la detección de los discos presentes en tu sistema y acto seguido te presentará las opciones de particionado de disco. Selecciona la primera opción y pulsa **Continuar**.



<Tab> mueve; <Espacio> selecciona; <Intro> activa un botón

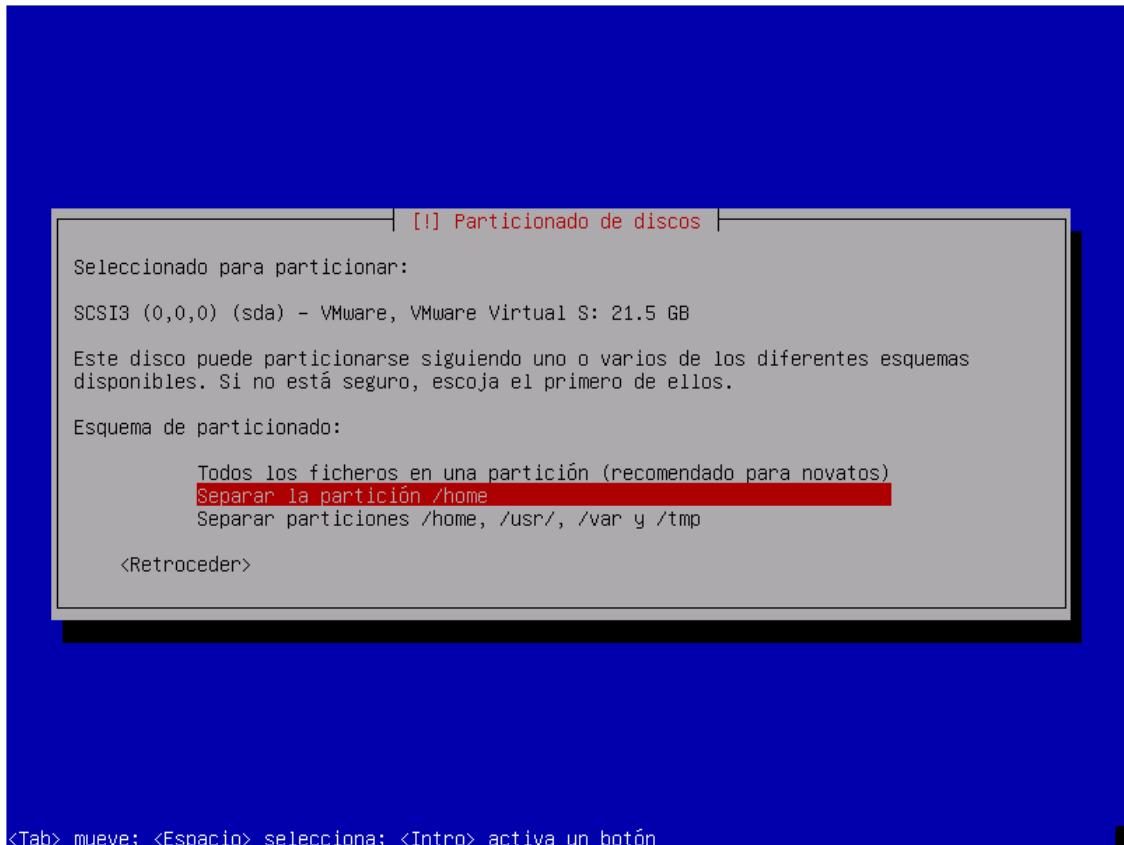
Debian te mostrará una lista con todas las unidades de disco presentes en tu sistema. Si existe más de un disco duro instalado aparecerán en esta pantalla.



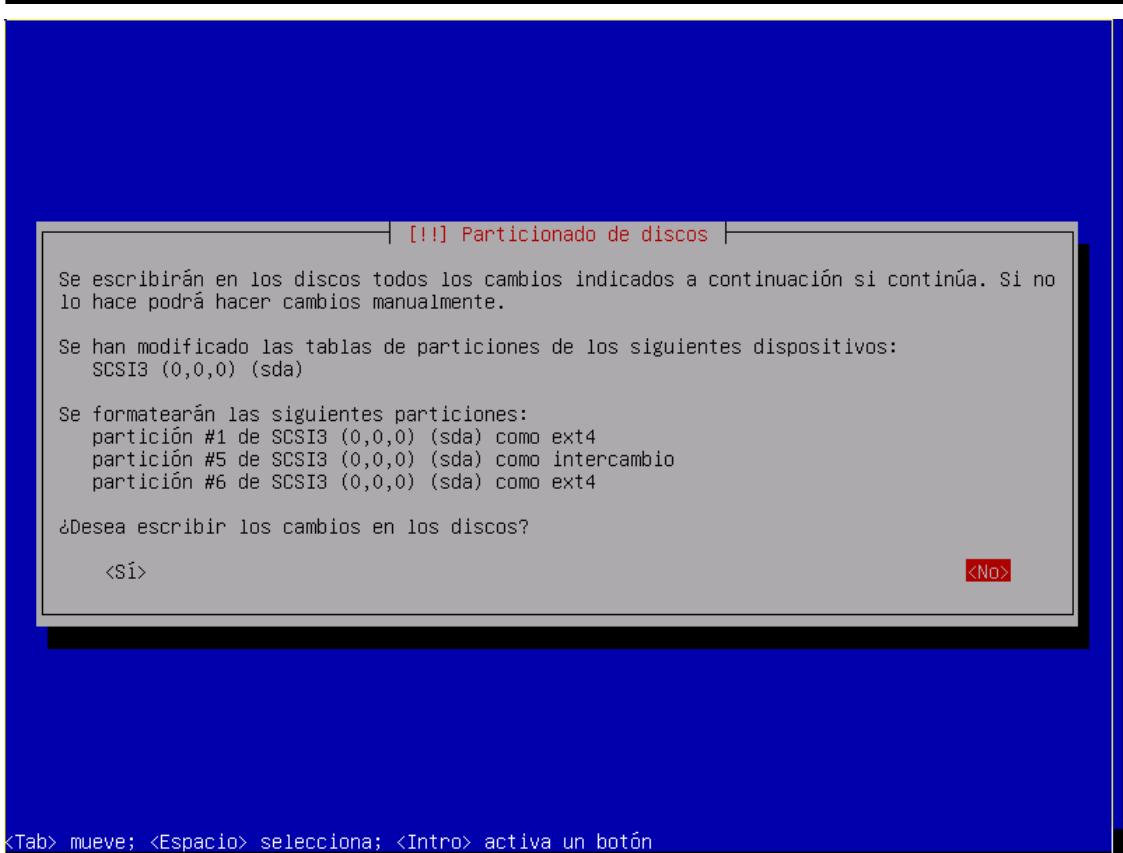
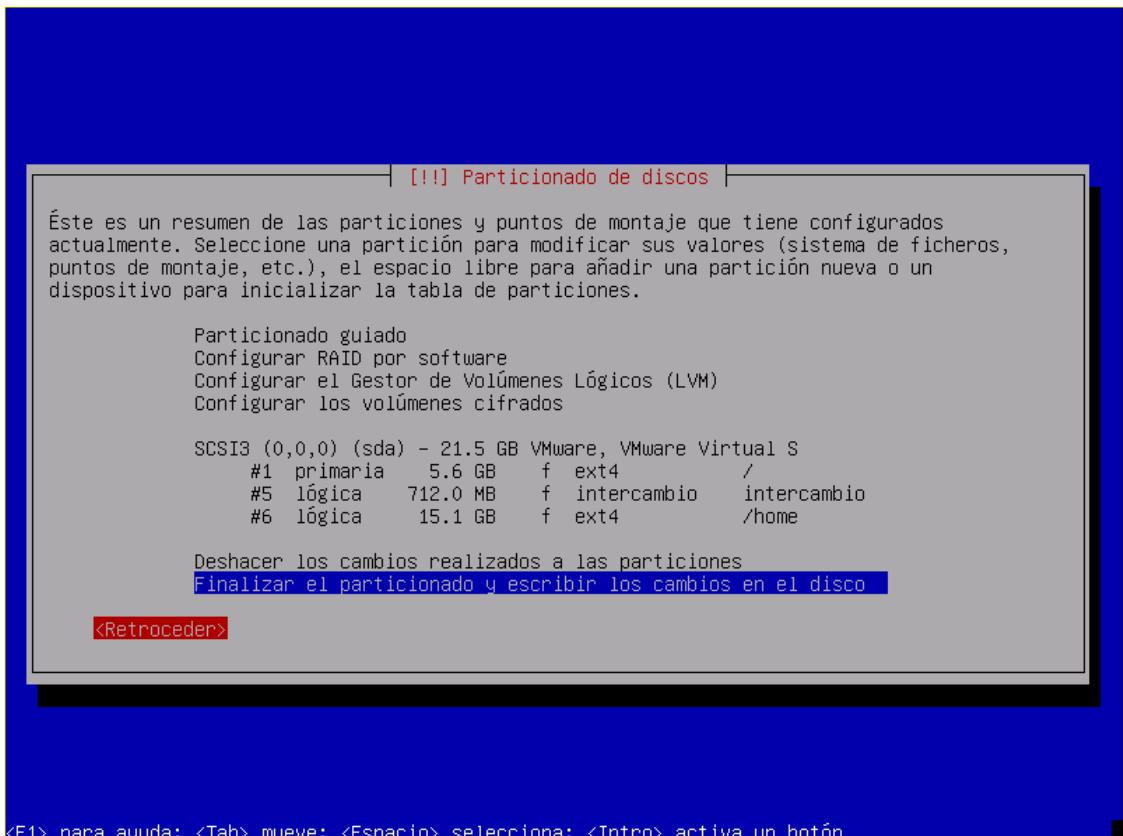
<Tab> mueve; <Espacio> selecciona; <Intro> activa un botón

Existen varias opciones predeterminadas. En esta ocasión se elige usar el directorio de trabajo

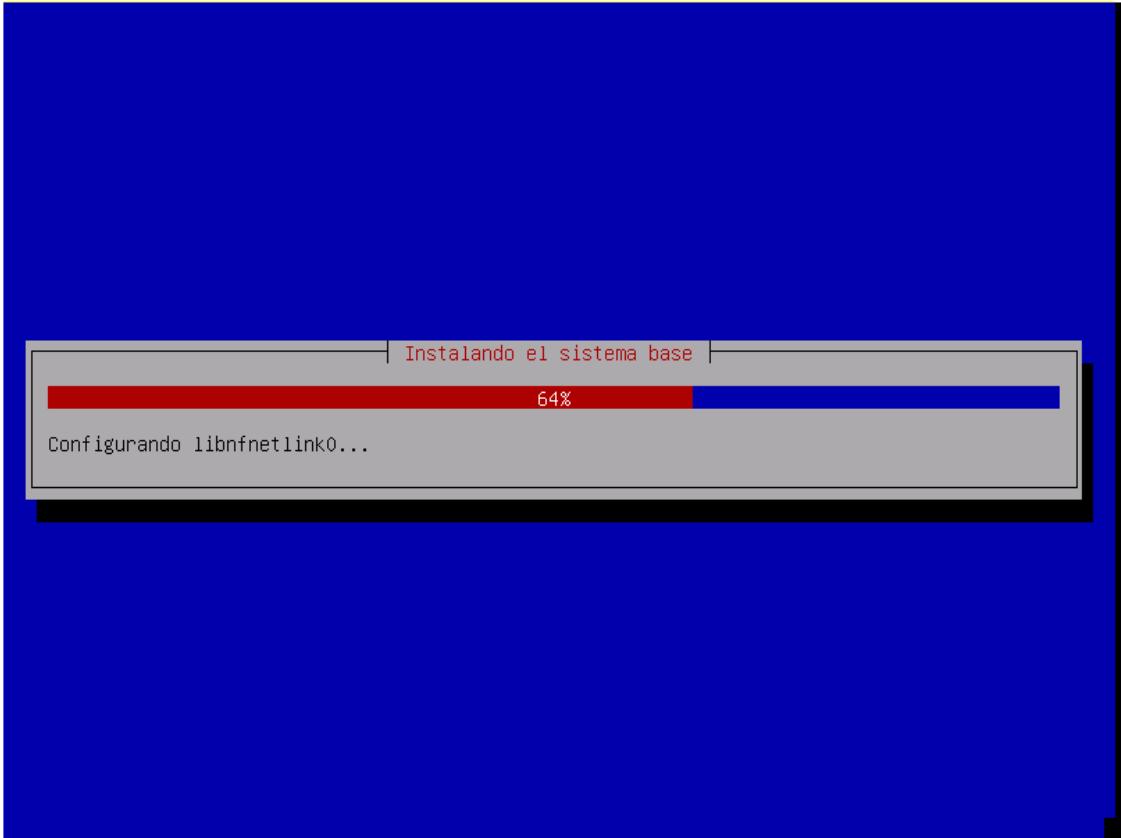
(/home) como una partición distinta.



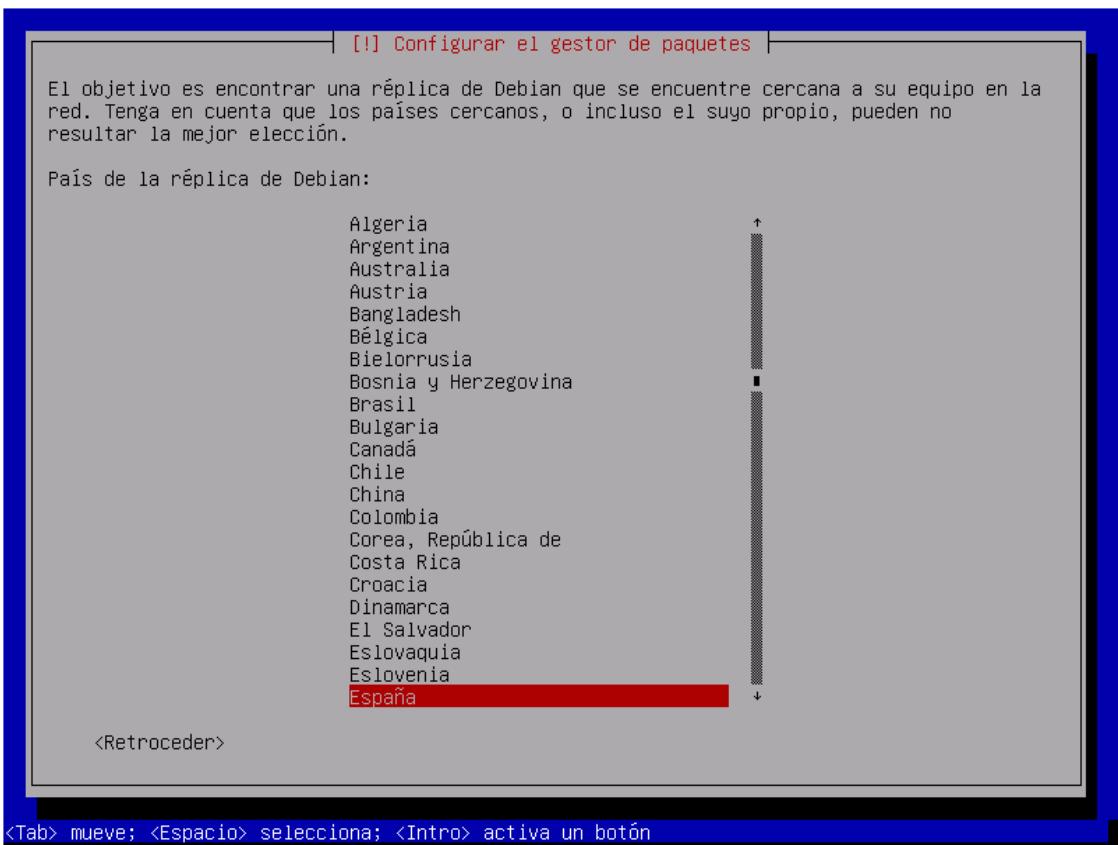
Debian permite hacer una revisión antes de proseguir con los cambios en el disco y presenta un resumen de lo que va a ejecutar. Si es acorde a la planificación de almacenamiento se debe escoger “Finalizar el particionado y escribir los cambios al disco”.



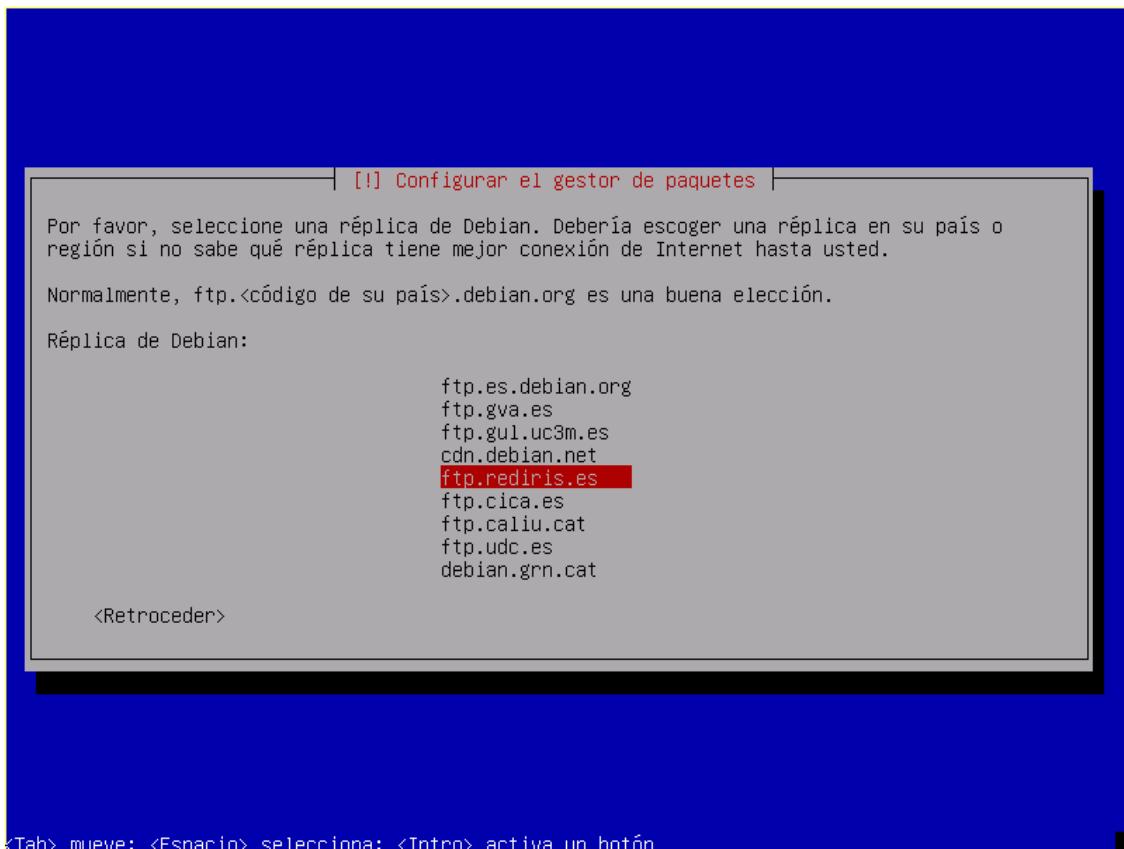
Comienza el copiado de los ficheros del sistema base. El proceso esencial de la instalación de Debian ha dado inicio.



Configurar el gestor de paquetes

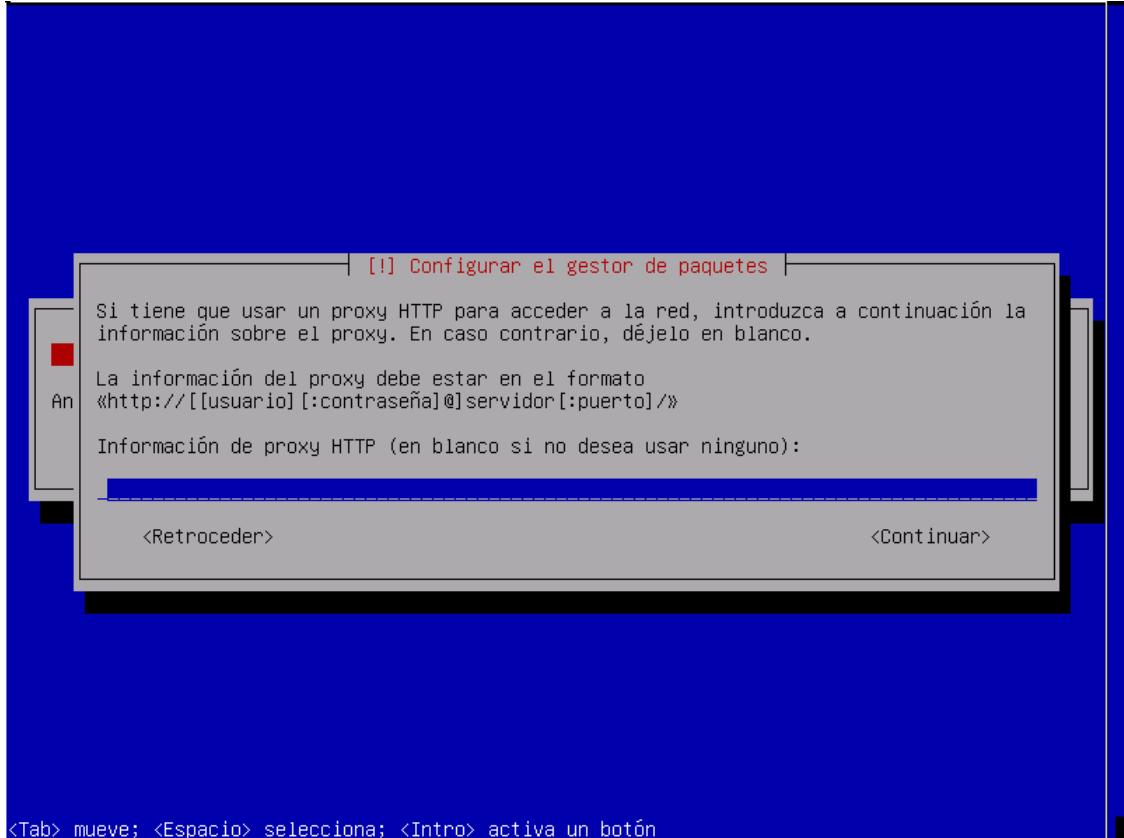


Tras seleccionar el país, Debian presenta una serie de servidores (réplicas) desde las que se pueden descargar los paquetes necesarios para la instalación

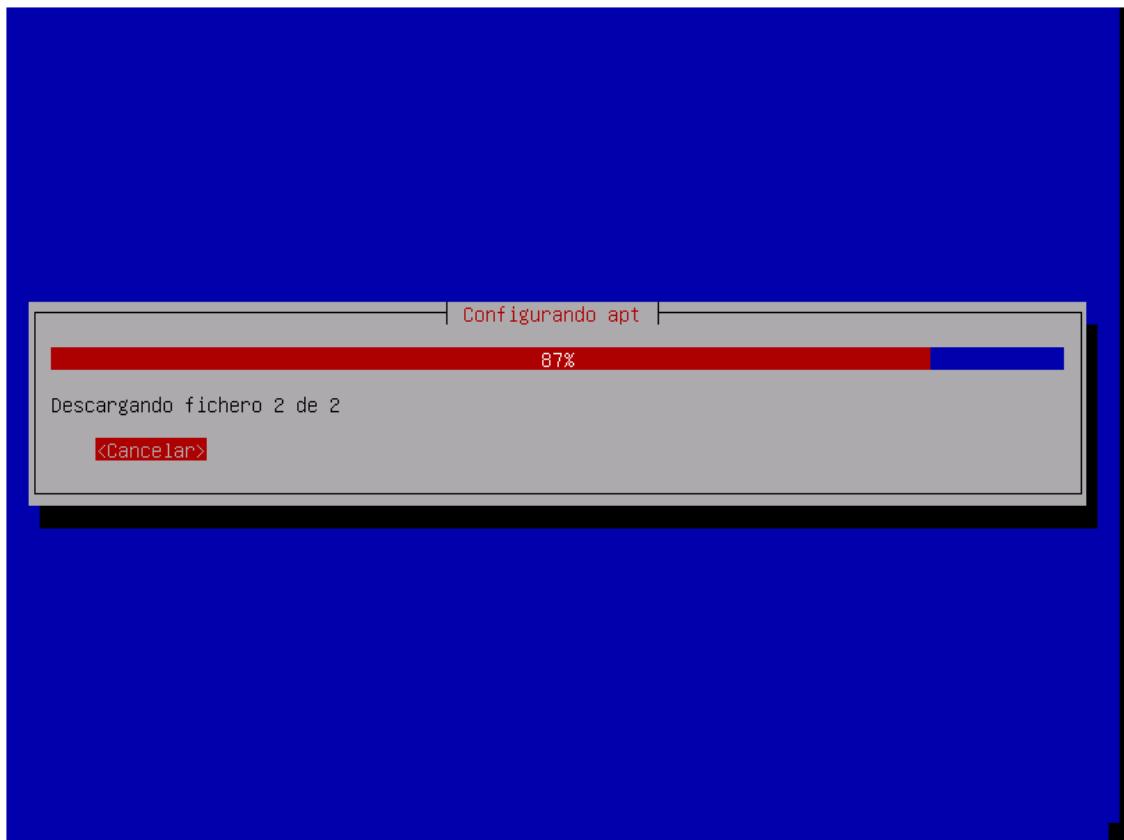


<Tab> mueve: <Espacio> selecciona: <Intro> activa un botón

Si fuera necesario especificar un servidor proxy para conectar con la réplica dbemos incluir la información a continuación.

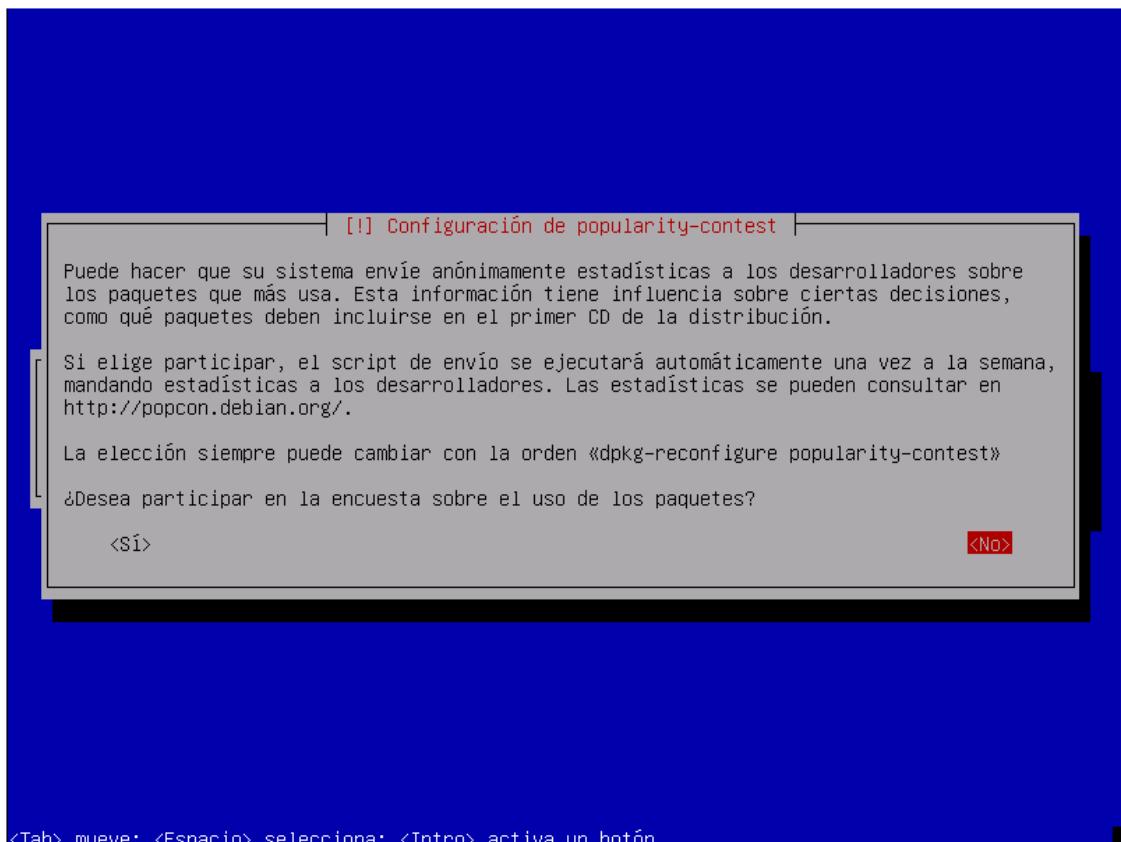


Se descarga y configura la información para la gestión de paquetes en Debian con el mítico apt



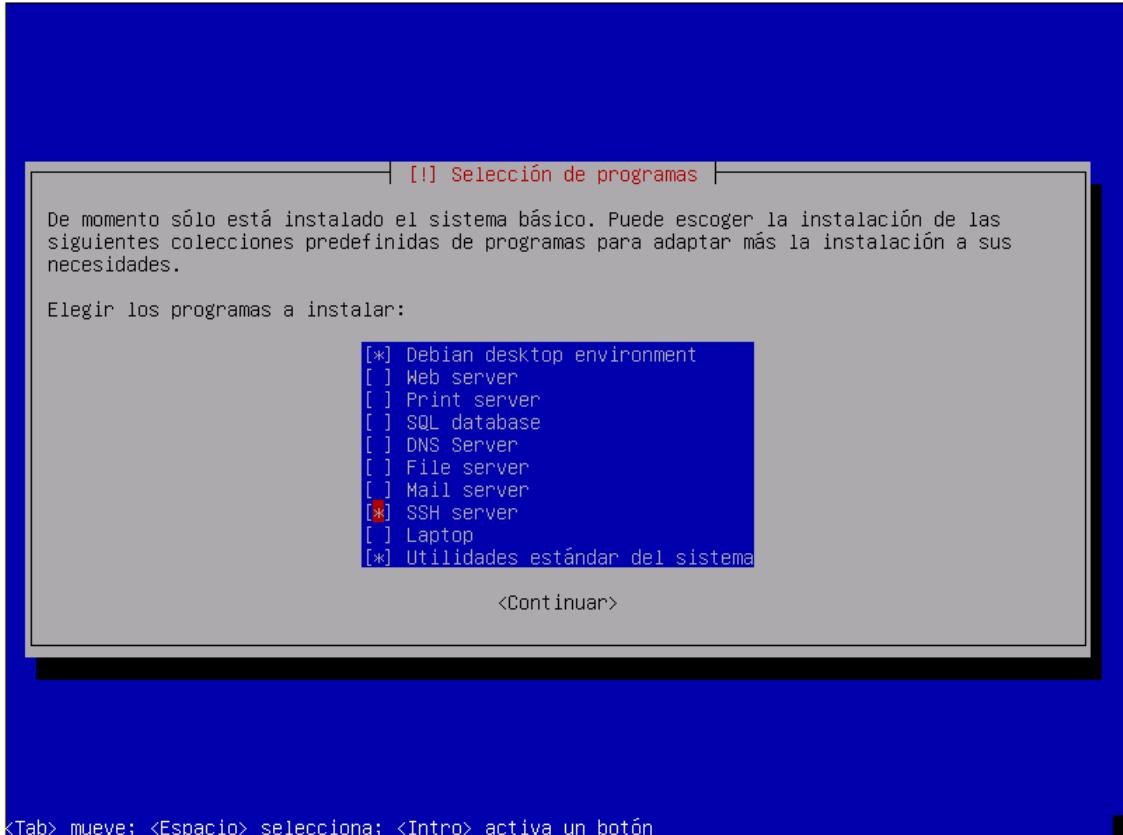
El concurso de popularidad.

Con la finalidad de obtener datos estadísticos que permita a los desarrolladores de Debian determinar que paquetes son los más populares, se ha implementado un script que envía automáticamente esta información 1 vez por semana. Los resultados de la encuesta se pueden consultar <http://popcon.debian.org/> Este script no afecta en absoluto el rendimiento del sistema

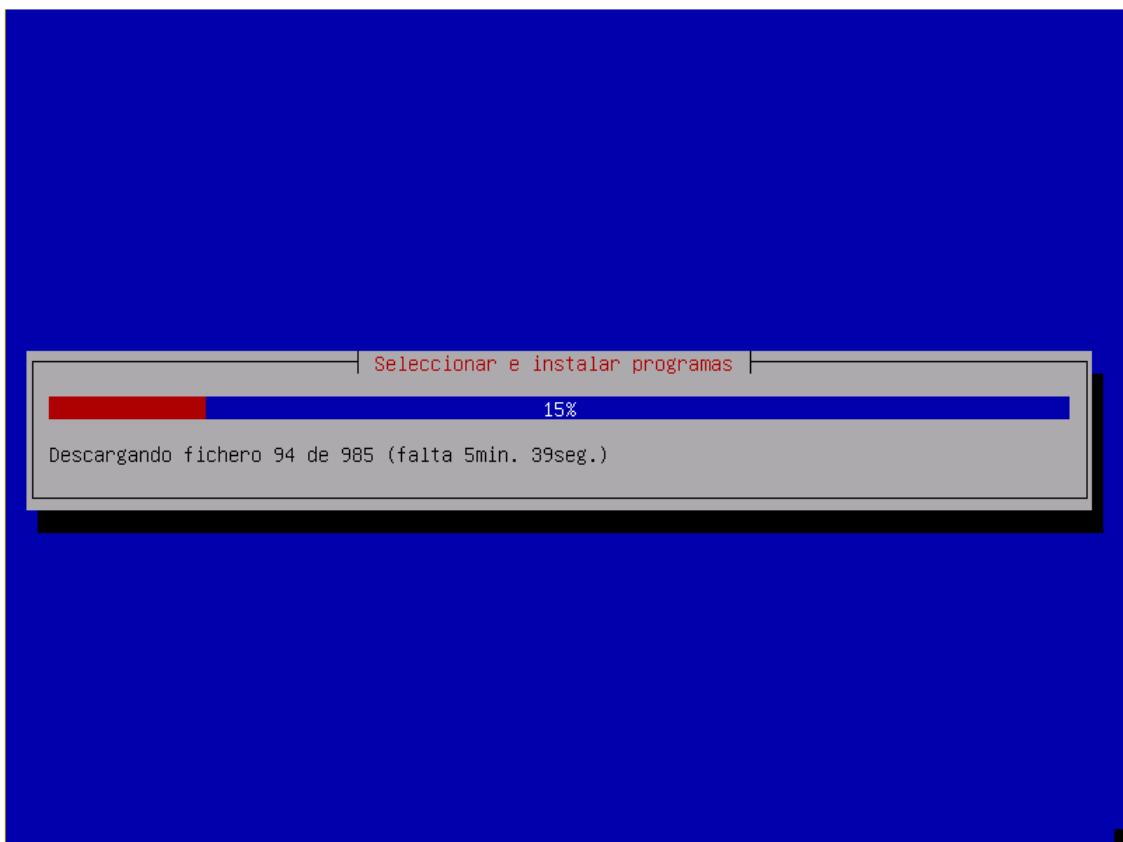


Selección de programas.

Debian permite escoger grupos de software predefinidos para actividades concretas. Para esta instalación se escogerá el servidor de ssh para administración y remota

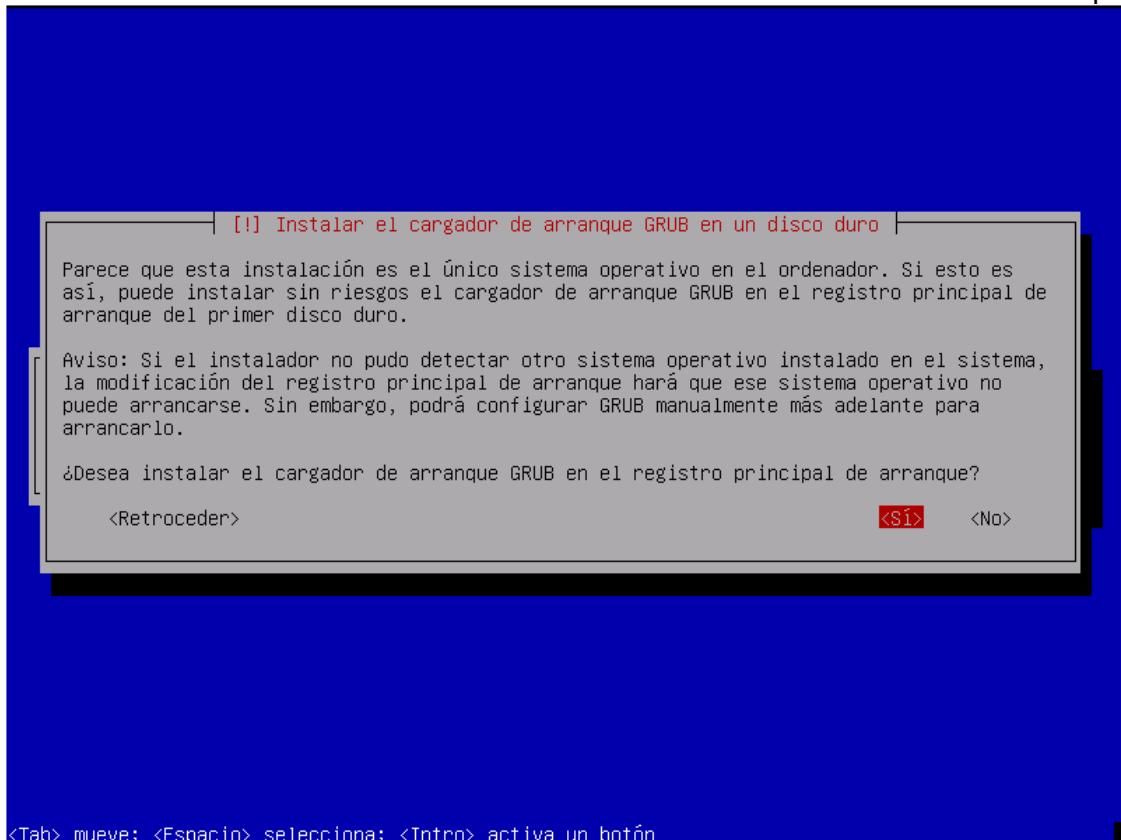


La descarga de los programas dependerá tanto de los grupos escogidos como de la velocidad de la conexión y la réplica escogida

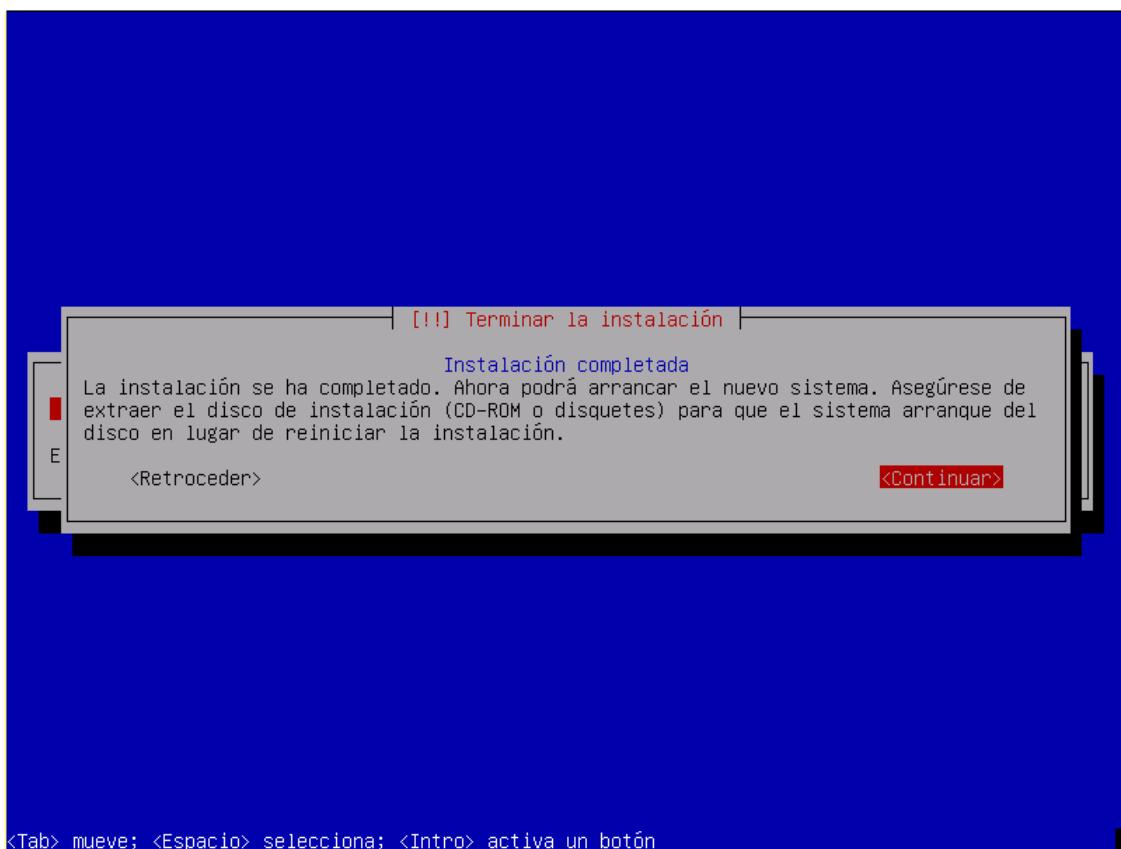


Grub.

Grub es el gestor de arranque de GNU-Linux y en el que se escogerá la sencilla opción de instalación predeterminada



Cuando la instalación ha terminado pulsamos continuar para reiniciar el equipo comenzar con el trabajo.



INSTALACIÓN CENTOS 6.5

Preparación

Al igual que otras distribuciones, CentOS puede instalarse de diferentes maneras. Para mostrar otro método diferente al de la instalación en Debian, se ha optado por escoger una instalación de red, con un arranque local (desde CD con la iso netinstall) y accediendo a un servidor NFS (que se explica en un extra de esta documentación), mediante una conexión de red obligatoria. El proceso solo difiere en los primeros pasos de una instalación desde un DVD local por lo que, si no se desea implementar la solución completa, se puede escoger una imagen iso de dvd y bootear desde ella sin más problemas. Para ello se puede descargar la iso de la web de CentOS o de algún mirror (en este caso http://sunsite.rediris.es/sites/centos.org/6.5/isos/x86_64/) Dado que existen elementos comunes a la instalación de Debian no se reiterarán los conceptos similares.

Menú de instalación.

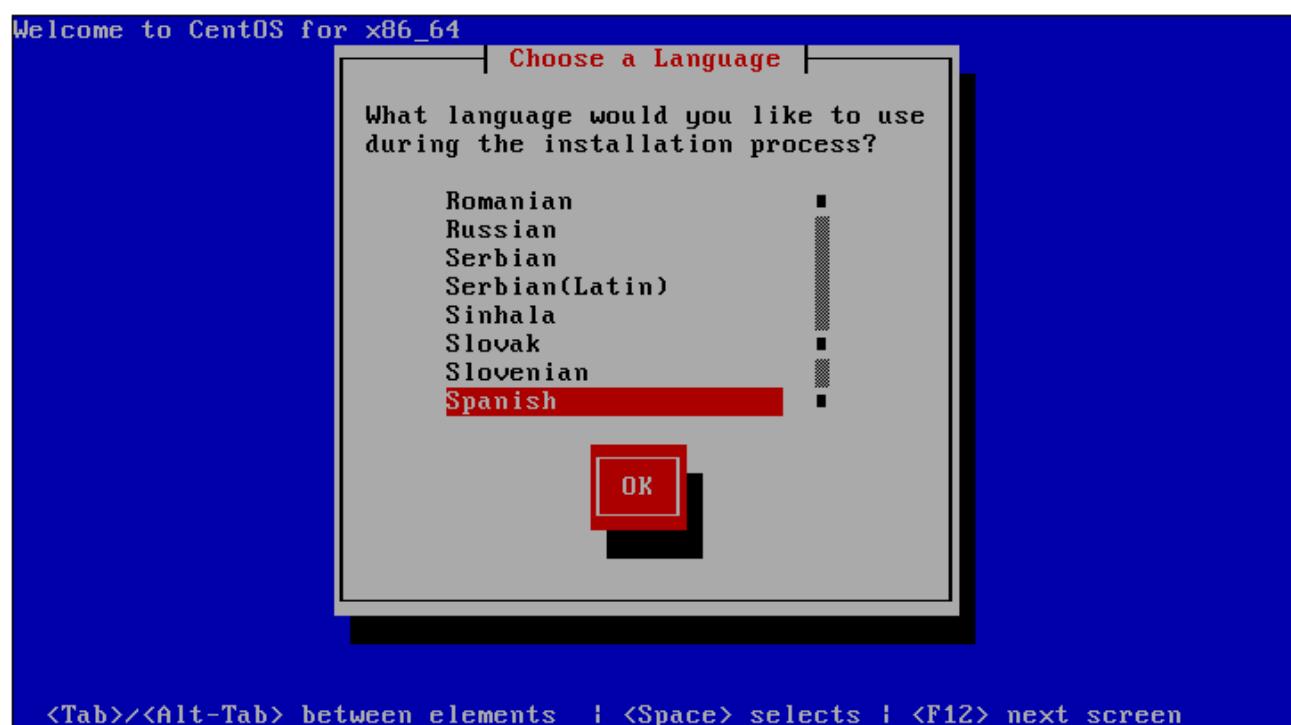


Una vez que se procede al arranque desde el dispositivo extraible CD/DVD (modificando la BIOS si fuera necesario) Aparece un menú inicial con diferentes opciones de recuperación y

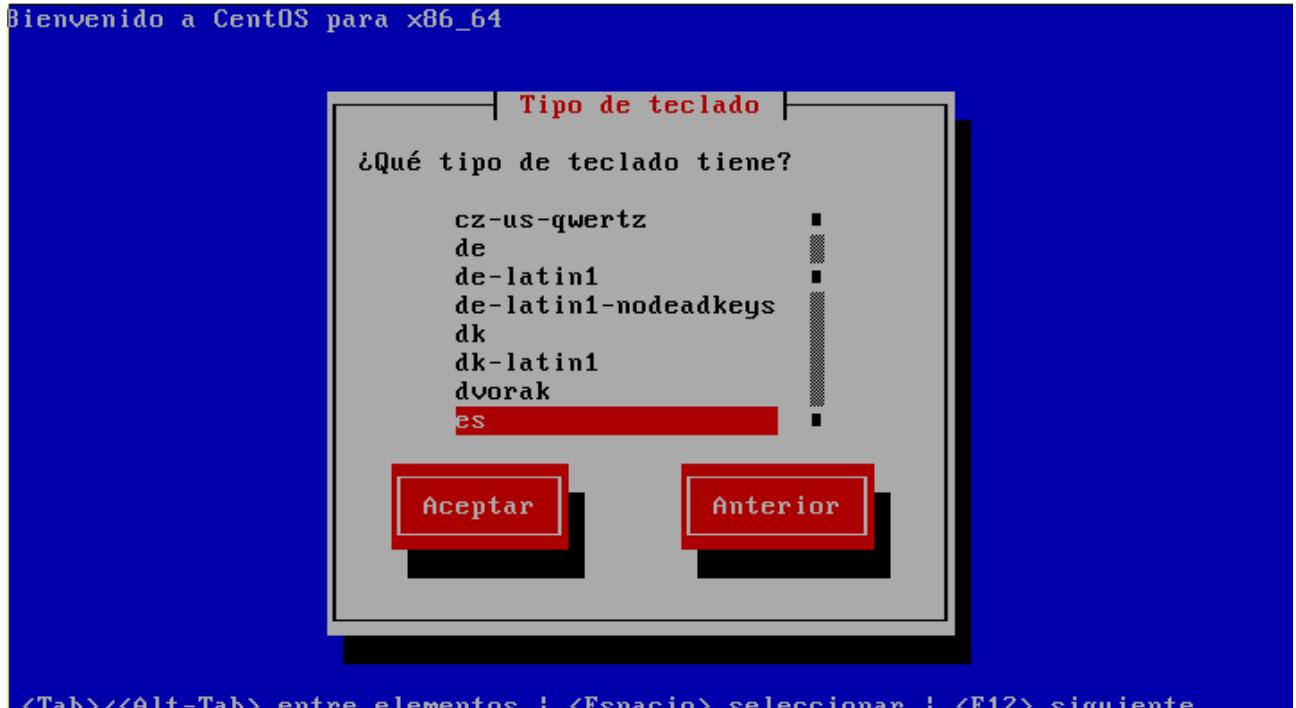
comprobación con una opción predeterminada que se ejecutará en 60" y comienza la instalación. Tras la carga de controladores básicos se nos solicitará intervención para comprobar el medio de instalación (CD) o no. Este proceso es relativamente lento.



Selección del idioma



Tipo de teclado



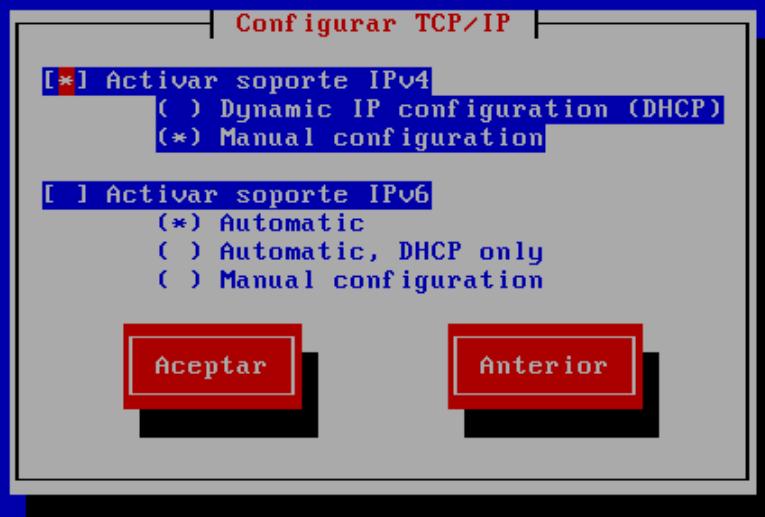
Elección del medio de instalación

En este punto se podría realizar una instalación tomando las fuentes desde un dispositivo local, un servidor http o un recurso NFS. Se escoge la opción NFS



Configurar TCP/IP

Bienvenido a CentOS para x86_64



<Tab>/<Alt-Tab> entre elementos | <Espacio> seleccionar | <F12> siguiente

Para el acceso al servidor de NFS es necesario tener una configuración mínima de red y se opta por la configuración manual para evitar tener que disponer de un servidor de DHCP (de sencilla configuración pero se documenta en otro extra). El soporte IPv6, en este caso, no es necesario. Asignamos los valores correctos de la configuración IP y posteriormente los datos del servidor y recurso NFS.

Configuración NFS

Bienvenido a CentOS para x86_64



<Tab>/<Alt-Tab> entre elementos | <Espacio> seleccionar | <F12> siguiente

Anaconda

Una vez conecta con el servidor NFS, ejecuta el instalador Anaconda y, desde este paso, el proceso es similar a la ejecución desde un DVD.



Almacenamiento básico o SAN

La orientación de CentOS a sistemas de servidor permite, de una forma sencilla, configurar durante la instalación el montaje de volúmenes de una red de almacenamiento.

¿Qué tipo de dispositivos involucra su instalación?

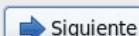
Dispositivos de almacenamiento básicos

- Instalaciones o actualizaciones para tipos comunes de dispositivos de almacenamiento. Si usted no está seguro de la opción apropiada para usted, ésta es probablemente la correcta.

Dispositivos de almacenamiento especializados

- Instala o actualiza dispositivos de empresa tales como Redes de área de almacenamiento (SAN). Esta opción le permitirá añadir discos FCoE / iSCSI / zFCP y filtrar los dispositivos que el instalador debe ignorar.

 Atrás

 Siguiente

Advertencia del dispositivo de almacenamiento



El dispositivo de almacenamiento puede contener datos.



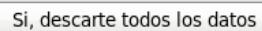
VMware, VMware Virtual S
20480.0 MB pci-0000:00:10.0-scsi-0:0:0:0

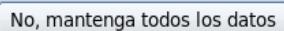
No pudimos detectar particiones o sistemas de archivos en este dispositivo.

Esto pudo deberse a que el dispositivo está en **blanco**, **no particionado**, o **virtual**. Si no, puede haber datos en el dispositivo que no pueden recuperarse si lo utiliza en esta instalación. Podemos retirar el dispositivo de esta instalación para proteger los datos.

¿Está seguro de que este dispositivo no contiene datos valiosos?

Aplicar mi elección a todos los dispositivos con particiones no detectadas o sistemas de archivos

 Si, descarte todos los datos

 No, mantenga todos los datos

Identificación del host

De forma predeterminada se incluye el sufijo DNS. Desde esta pantalla se permite acceder a la configuración TCP/IP

 Por favor, de un nombre a esta computadora. El nombre de host identifica al computador en una red.

Nombre del host:

[Configure la red](#)

[Atrás](#) [Siguiente](#)

Contraseña del root

A diferencia de Debian esta contraseña no puede estar “en blanco” y debe cumplir unos ciertos requisitos de seguridad

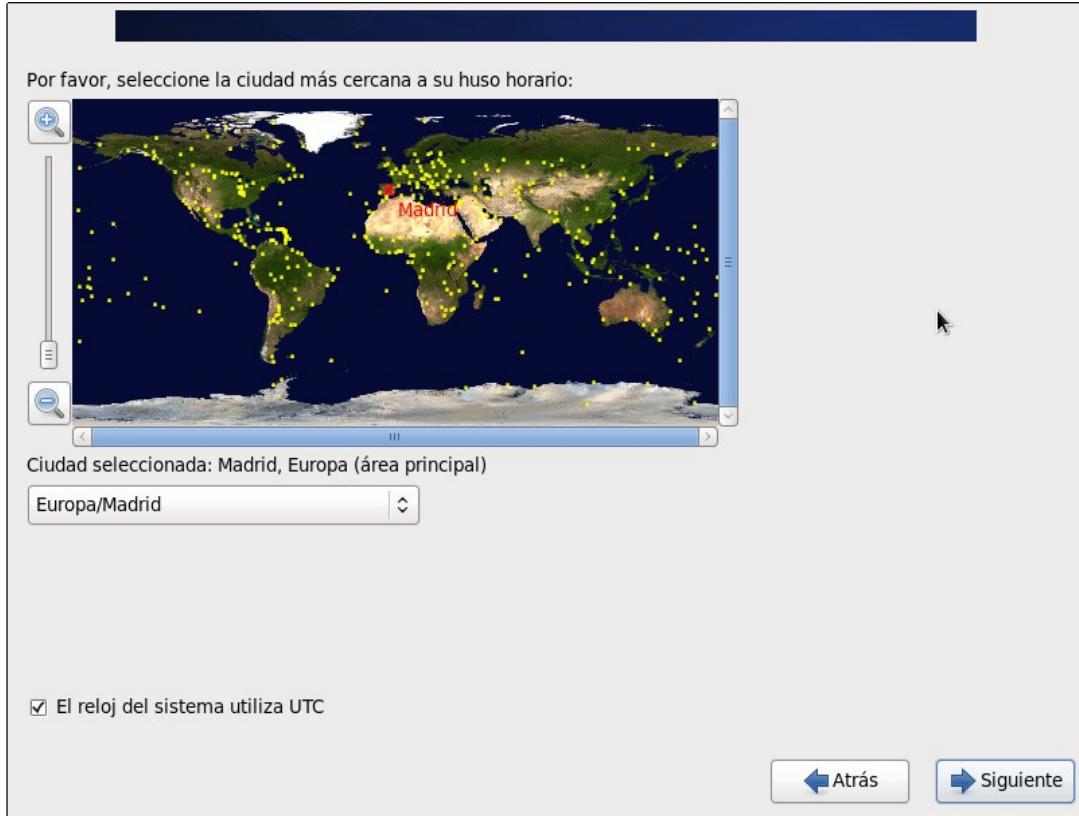
 La cuenta root se utiliza para la administración del sistema. Introduzca una contraseña para el usuario root.

Contraseña de root:

Confirmar:

[Atrás](#) [Siguiente](#)

Zona horaria



Particionado de disco

Existen distintas opciones para la gestión del almacenamiento en CentOS, pero se ha de tener en cuenta que, de forma predeterminada, no utiliza un sistema clásico de particiones o segmentos si no de volúmenes (LVM) que se analizarán en otro extra.

¿Qué tipo de instalación desea?

Usar todo el espacio

Elimina todas las particiones en los dispositivos seleccionados. Esto incluye las particiones creadas por otros sistemas operativos.

Consejo: Esta opción eliminará los datos de los dispositivos seleccionados. Asegúrese de hacer copias de seguridad.

Reemplazar sistema(s) Linux existente(s)

Elimina sólo las particiones Linux (creadas desde una instalación previa de Linux). Esto no elimina otras particiones que tenga en sus dispositivos de almacenamiento (tales como VFAT o FAT32).

Consejo: Esta opción eliminará los datos de los dispositivos seleccionados. Asegúrese de hacer copias de seguridad.

Achicar el sistema Actual

Achica las particiones existentes para dar campo al diseño predeterminado.

Usar el espacio libre

Mantiene sus datos actuales y particiones, y usa solamente el espacio no particionado en los dispositivos seleccionados, asumiendo que hay espacio libre suficiente.

Crear un diseño personalizado.

Crear manualmente su propio diseño en los dispositivos seleccionados usando nuestra herramienta de particionamiento.

Sistema de Encriptado

Revisar y modificar el diseño de particiones

Atrás

Siguiente

Escribiendo la configuración de almacenamiento en el disco...



Las opciones de particionamiento que ha seleccionado, se escribirán ahora en el disco. Cualquier dato en particiones borradas o reformateadas se perderán.

[Volver Atrás](#)

[Escribir cambios al disco](#)

Atrás

Siguiente

Paquetes de software

La instalación predeterminada de CentOS es una instalación mínima. También puede seleccionar un conjunto diferente de software ahora.

- Desktop
- Minimal Desktop
- Minimal
- Basic Server
- Database Server
- Web Server
- Virtual Host

Por favor, seleccione cualquier repositorio adicional que quiera usar para la instalación de software.

- CentOS

 Agregar repositorios de software adicional

 Modificar repositorio

Puede personalizar la selección de software ahora o después de la instalación a través de la aplicación de administración de software.

- Personalizar más adelante Personalizar ahora

 Atrás

 Siguiente





Configuración tras el reinicio



The screenshot shows the "Bienvenido" (Welcome) screen of the CentOS 6.5 setup. On the left, a vertical sidebar lists several configuration steps: "Bienvenido", "Información de Licencia", "Crear Usuario", "Fecha y Hora", and "Kdump". The main area features a large title "Bienvenido" and a descriptive text block. Below this is a dark blue rectangular box containing the "CentOS 6" logo and the text "Community ENTerprise Operating System". At the bottom right of the main area, there are two buttons: "Atrás" (Back) and "Adelante" (Forward).

Bienvenido

Hay algunos pasos más que debe realizar antes de que su sistema esté listo para ser utilizado. El Agente del configuración lo guiará a través de una configuración básica. Pulse "Adelante" en la esquina inferior derecha para continuar

CentOS 6
Community ENTerprise Operating System

Atrás Adelante

Bienvenido
› Información de Licencia
Crear Usuario
Fecha y Hora
Kdump

Información de Licencia

CentOS-6 EULA

CentOS-6 comes with no guarantees or warranties of any sorts, either written or implied.

The Distribution is released as GPLv2. Individual packages in the distribution come with their own licences. A copy of the GPLv2 license is included with the distribution media.

- Sí, Estoy de acuerdo con el Acuerdo de Licencia
 No, no estoy de acuerdo

[Atrás](#) [Adelante](#)

Creación de una cuenta de usuario regular

Bienvenido
Información de Licencia
› Crear Usuario
Fecha y Hora
Kdump

Crear Usuario

Se recomienda crear un 'nombre_de_usuario' para uso normal (no administrativo) de su sistema. Para crear un sistema 'nombre_de_usuario', por favor, provea la información que se pide más abajo.

Nombre de Usuario:	<input type="text" value="operador"/>
Nombre Completo:	<input type="text" value="usuario de uso regular"/>
Contraseña:	<input type="password" value="*****"/>
Confirme la Contraseña:	<input type="password" value="*****"/>

Si necesita usar autenticación de red, tal como Kerberos o NIS, por favor haga clic en el botón Usar Ingreso por Red.

[Usar el Ingreso por Red...](#)

Si necesita más control en la creación de usuario (especificando el directorio principal y o el UID), por favor haga clic en el botón Avanzado.

[Avanzado...](#)

[Atrás](#) [Adelante](#)

Fecha y hora

Si se marca la casilla “sincronizar fecha y hora por la red” activa el protocolo ntp que se estudia en otro apartado.

Bienvenido
Información de
Licencia
Crear Usuario
➤ Fecha y Hora
Kdump

Fecha y Hora

Por favor, ingrese la fecha y hora del sistema.

Fecha y Hora

Fecha y Hora Actual: lun 02 dic 2013 23:48:39 CET
 Sincronizar fecha y hora por la red

Sincronizar la fecha y hora de su computadora con un servidor de hora remoto usando el Protocolo de Hora por Red:

Servidores NTP

0.centos.pool.ntp.org	Añadir
1.centos.pool.ntp.org	Editar
2.centos.pool.ntp.org	Eliminar
3.centos.pool.ntp.org	

▼ Opciones Avanzadas

Acelerar la sincronización inicial
 Usar Fuente de Tiempo Local

[Atrás](#) [Adelante](#)

Bienvenido
Información de
Licencia
Crear Usuario
➤ Fecha y Hora
Kdump

Fecha y Hora

Por favor, ingrese la fecha y hora del sistema.

Fecha y Hora

Fecha y Hora Actual: lun 02 dic 2013 23:41:35 CET
 Sincronizar fecha y hora por la red

Poner manualmente la fecha y hora de su sistema:

Fecha							Hora			
< diciembre >		< 2013 >					Hora :	Minuto :	Segundo :	
lun		mar	mié	jue	vie	sáb	dom	23	7	37
25		26	27	28	29	30	1			
2		3	4	5	6	7	8			
9		10	11	12	13	14	15			
16		17	18	19	20	21	22			
23		24	25	26	27	28	29			
30		31	1	2	3	4	5			

[Atrás](#) [Adelante](#)

Volcado de memoria

Bienvenido
Información de Licencia
Crear Usuario
Fecha y Hora
» Kdump

Kdump

Kdump es un mecanismo de volcado de fallos del kernel. En el evento de una falla del sistema, kdump capturará la información de su sistema que puede ser invaluable para la determinación de la causa del fallo. Observe que kdump no requiere reservar una porción de memoria del sistema que no estará disponible para otros usos.

Habilitar kdump?

Memoria Total del Sistema (MB): 1998

Memoria de Kdump (MB): 128

Memoria de Sistema Utilizable (MB): 1870

Configuración avanzada de Kdump

```
# Configures where to put the kdump /proc/vmcore files
#
# This file contains a series of commands to perform (in order) when a
# kernel crash has happened and the Kdump kernel has been loaded. Di
# this file are only applicable to the Kdump initramfs, and have no effec
# the root filesystem is mounted and the normal init scripts are proces
#
# Currently only one dump target and path may be configured at a time
# to configured dump target fails, the default action will be preformed.
# Default action may be configured with the "default" directive below.
#
# Basics commands supported are:
# path <path> - Append path to the filesystem device which y
#                 dumping to. Ignored for raw device dumps.
#                 If unset, will default to /var/crash.
#
# core_collector_commands continue
```

Atrás Finalizar

INICIO DE SESIÓN EN EL SISTEMA

Cuando deseamos iniciar sesión en un sistema, debemos identificarnos, es decir, el sistema operativo necesita reconocernos como usuarios validos para poder trabajar con él. La forma en que nos identificamos como usuarios es mediante un par de claves que son nuestro nombre de usuario y nuestra contraseña. El nombre de usuario es una cadena de caracteres publica (conocida por el resto de usuarios) y única (no pueden existir dos nombres de usuarios iguales para el mismo sistema). La contraseña, sin embargo es privada (nadie excepto el usuario que la utiliza debe conocerla). Para poder entrar en un sistema es necesario aportar ambos datos de forma correcta, a este proceso se le conoce bajo el nombre de inicio de sesión en el sistema. Al conjunto de nombre de usuario y contraseña se le conoce como credenciales del usuario o cuenta de usuario.

Las cuentas de usuario se almacenan en el sistema en un fichero llamado /etc/passwd (que más adelante veremos). El administrador del sistema es el responsable de crear y mantener las cuentas del sistema. La cuenta de usuario del administrador se crea durante el proceso de instalación del sistema operativo. El nombre de usuario es “root” (nombre de usuario que no se puede modificar), la contraseña, es solicitada por el asistente de instalación. Esta cuenta, por lo tanto, la utiliza el usuario que tiene las funciones de administrador del sistema para poder llevar a cabo las labores de mantenimiento y configuración de la máquina.

Usar la cuenta de root sin conocimientos puede ser peligroso para las tareas del sistema, por ello la mayoría de sistemas GNU/linux permiten crear una cuenta de usuario regular con privilegios (mediante un proceso de autenticación) para las tareas comprometidas.

Incluso cuando existe un arranque automatizado, normalmente en equipos domésticos, se produce el proceso de login aunque en un segundo plano.

En los sistemas basados en unix, el proceso de login se puede producir en un entorno de terminal o en un entorno gráfico (GUI) como pueden ser GNOME, KDE, LXDE, XFCE y otros que comentaremos más adelante. Estos entornos de trabajo suelen tener un servicio de login gráfico que nos permite introducir las credenciales de acceso. Es habitual que no se muestre la contraseña mientras la tecleamos y que aparezcan puntos o asteriscos por motivos de seguridad.

Si el acceso es una terminal o consola de texto se nos solicitará el login y la contraseña, en este caso no se suele mostrar ningún carácter especial ni la contraseña aunque estemos escribiéndola.

```
atlante login: root  
Password:  
Last login: Wed May 29 12:54:45 CEST 2013 on tty1
```

Para terminar una sesión debemos ejecutar el comando

```
$logout
```

(recordamos que el símbolo \$ es el prompt o indicador del sistema, no es parte del comando)

¡En los sistemas Unix se distingue entre mayúsculas y minúsculas. Por lo que es necesario que ambas claves se escriban conforme las generó el administrador.

EDITORES DE TEXTO SIMPLES

Como iremos comprobando, en GNU/linux las operaciones de configuración con ficheros de textos son muy habituales y para ello se debe usar un editor. Existen multiples editores pero ya que comenzamos con un sencillo acceso al entorno gráfico debemos conocer una herramienta bien sencilla llamada nano.

Nano es un editor de texto que nació como un clon libre de Pico, editor de texto un cliente de correo llamado Pine.

Poco a poco llegó a tener más funcionalidades, y hoy es uno de los editores más utilizados desde una terminal.

Como otros editores, está orientado a utilizarlo mediante combinaciones de teclas. La principal diferencia radica en que ofrece en pantalla las opciones más básicas.

Para ejecutarlo debemos acceder a una ventana de terminal y ejecutar:

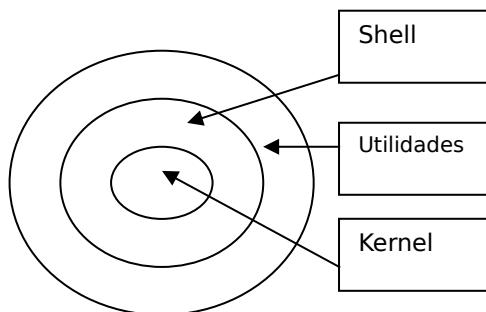
```
$nano
```

En cualquier caso, más adelante aprenderemos a manejar el omnipresente y estandar de facto en todos los sistemas unix y derivados llamado vi.

COMPONENTES DE UN SISTEMA GNU/LINUX

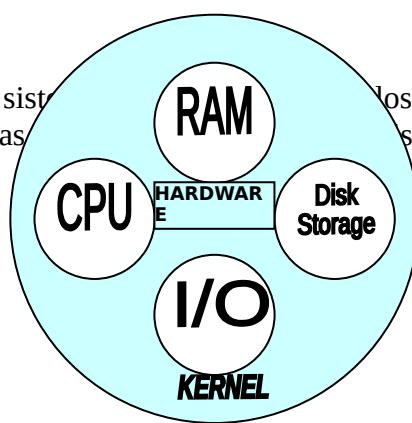
Como cualquier sistema operativo, el sistema GNU/linux es un software que permite administrar los recursos de la máquina, así como, interpretar las instrucciones dadas por el usuario o por una aplicación sobre el hardware del equipo. Además, permite la comunicación con dispositivos externos, como impresoras, disquetes, monitor, etc. Para ello, este sistema operativo se apoya en tres componentes principales que son:

- El kernel
- La Shell
- El árbol de directorios



El kernel

El Kernel es el núcleo del sistema, todo el hardware, disqueteras, teclado, ratón, etc., se controla a través del kernel. Los controladores que permiten administrar el hardware son los controladores que permiten administrar el hardware. El kernel es linux.



La shell

La siguiente capa incluye las utilidades y los comandos del sistema. Estas utilidades o comandos interactúan con el kernel invocando un conjunto bien definido de llamadas al sistema. Las llamadas al sistema ordenan al kernel realizar varias operaciones para la utilidad o comando que llama e intercambiar datos entre el kernel y dicha utilidad o comando. El shell es la parte visible del sistema, es todo lo que el usuario puede ver. El shell no es más que un programa que acepta y ejecuta los comandos de un usuario.

El shell es, por tanto, un procesador de comandos. Pero también tiene otras funciones: es el encargado de la redirección de las entradas y salidas de los datos, es también un lenguaje de programación que hace posible la interacción de varios comandos a la vez, y además tiene la función de ser interface de usuario, es decir aquello que vemos. En definitiva, el shell es el intermediario entre el sistema operativo básico (kernel) y el usuario. Existen varios programas shell, como por ejemplo Bash, tcsh, Bourne, C y Korn. Lo que diferencia a los distintos shell es que cada uno de ellos tiene unas funcionalidades específicas aparte de las comunes que todos presentan. Por ejemplo, Bash presenta la opción de utilizar el backspace o retroceso para borrar el historial, es decir, va guardando los últimos comandos que hemos ido ejecutando para no tener que volver a escribirlos (esta función se realiza con las flechas de arriba y abajo del cursor), mientras que el shell Bourne no presenta ninguna de estas opciones. El shell Bash (Bourne again shell) es el estandar en los sistemas GNU/linux.

Los shells permiten interpretar las instrucciones que le enviamos de forma interactiva y también un conjunto de instrucciones en lote . Eso son los denominados shell scripts (guiones) que el shell lee habitualmente de un fichero y procesa los comandos de forma secuencial. El proceso más simple sería el siguiente:

1. Lee el comando de una terminal o de un fichero
2. Valida el comando
3. Ejecuta el comando enviando la salida a la pantalla o donde esté indicado
4. continua por el paso 1

Los shell suelen tener capacidades similares a los lenguajes de programación tales como definir variables, crear bucles o sentencias condicionales

En un sistema GNU/linux, para la mayoría de las tareas que se realizan se utilizan comandos o instrucciones que se deben teclear en la línea de comandos.

Hay que tener en cuenta que el tipo de prompt que tenemos en pantalla dependerá del usuario con el que hayamos accedido al sistema y del shell que estemos utilizando.

#: Este tipo de prompt nos indica que somos administradores o hemos empezado la sesión como root (superadministradores).

\$: Cuando tenemos este prompt nos indica que somos usuarios regulares (no root) y que estamos utilizando los shell Bourne o Bash.

¡Recuerda que si entramos como root o superusuario, debemos tener en cuenta que el sistema considera que sabemos lo que hacemos y en ningún momento nos pedirá confirmación para ejecutar las órdenes que le demos!

El tipo de sintaxis que vamos a utilizar en GNU/linux va a ser el siguiente:

```
# comando –opciones argumentos
```

comando: La instrucción que queramos utilizar en cada momento.

opciones: operador que variar la ejecución de cada comando.

argumentos: el ámbito de la ejecución del comando que puede ser un nombre del fichero o directorio. Variará en función del comando

Debemos tener en cuenta, que los nombres de los comandos, en general, van en minúsculas.

Para acceder al shell de linux, necesitamos abrir una ventana de terminal.

Vamos a ver una serie de ejemplos para reconocer la diferencia entre comando, argumento y opción:

Si se ejecuta:

```
$ ls
```

Se trata de un comando que esta siendo ejecutado por un usuario que no es root.

Si ejecutamos:

```
$ ls -l
```

Se trata de un comando (ls) que esta siendo ejecutado por un usuario que no es root. Dicho comando lo estamos ejecutando con una opción que la opción “l”

Si ejecutamos la siguiente línea:

```
$ ls dir1
```

En este caso estamos ejecutando el comando ls como usuarios distintos de root con un argumento que es dir1.

Si ejecutamos:

```
$ ls -l dir2
```

Ahora, estamos ejecutando el comando ls como usuario no root, con una opción que es “l” y con un argumento que es dir2.

Si ejecuta la siguiente línea:

```
$ cal 2 2003
```

En este caso, estamos ejecutando el comando cal como usuario no root con dos argumentos que es 2 y 2003.

Si usted ejecuta el siguiente comando:

```
$ ls -ld
```

Estamos ejecutando el comando ls con varias opciones a la vez que son l y d

Otra forma de escribir la línea de arriba sería:

```
$ ls -l -d
```

El efecto en ambos casos seria el mismo. Igual que si invertimos el orden de las opciones.

Las opciones de un comando pueden indicarse de varias formas:

- (guión) es propio de los sistemas unix

-- (doble guión) propio de las aplicaciones GNU. Las opciones definidas así no son acumulativas. sin guión una herencia de los sistemas bsd que suele ser raro en GNU/linux

101 ARQUITECTURA DEL SISTEMA.

- 101.1. Identificar y editar configuraciones hardware.
- 101.2. Inicio del sistema.
- 101.3. Cambiar niveles de ejecución, apagar y reiniciar el sistema.

101.1. Identificar y editar configuraciones hardware.

Peso en el examen de certificación: 2 puntos.

Objetivo: Identificar y configurar el hardware fundamental del sistema.

Conceptos y áreas de conocimiento:

- .Activar y desactivar los periféricos integrados.
- .Configurar sistemas con o sin periféricos externos, tales como los teclados.
- .Diferenciar entre los distintos tipos de dispositivos de almacenamiento masivo.
- .Establecer el ID de hardware adecuado para diferentes dispositivos, especialmente los de arranque.
- .Conocer las diferencias entre dispositivos coldplug y dispositivos hotplug.
- .Determinar los recursos de hardware para los dispositivos.
- .Herramientas y utilidades para mostrar la información de hardware diferente (lsusb, lspci, etc)
- .Herramientas y utilidades para manipular los dispositivos USB.
- . Comprensión conceptual de sysfs, udev, hald, dbus

Términos y utilidades

/sys
/proc
/dev
modprobe
lsmod
lspci
lsusb

101.1.1. La BIOS

Los ordenadores están compuestos por una parte hardware y una parte software. El hardware son los componentes físicos del ordenador, como son el microprocesador (o CPU, *Central Processing Unit*, Unidad Central de Proceso), la memoria RAM (*Random Access Memory*), el disco duro, la tarjeta gráfica, etc. El software en cambio está formado por los programas y los datos que manejan dichos programas.

Dentro del hardware del ordenador encontramos la **BIOS** (Basic Input/Output System, Sistema Básico de Entrada/Salida), que es el software de más bajo nivel que se ejecuta en el ordenador y que se encuentra "grabado" en un chip de memoria de la placa base del ordenador, motivo por el cual se le denomina "firmware". En los primeros ordenadores este software se almacenaba en un chip de tipo **ROM** (*Read Only Memory*, memoria de solo lectura), que para actualizarlo debía ser remplazado por otro chip; posteriormente se emplearon chips de memoria de tipo **EEPROM** (*Electrically Erasable Programmable Read-Only Memory*, memoria de sólo lectura borrable y programable eléctricamente), cuyo contenido podía actualizarse por software, sin necesidad de remplazar el chip. Actualmente se utilizan memorias de tipo **Flash**.

La BIOS suele ir acompañada de otra memoria, en este caso de tipo **CMOS** (Complementary Metal-Oxide Semiconductor), donde se guarda la configuración de la BIOS, cuya información se

mantiene gracias a la pila de botón de la placa base.

Cuando se enciende el ordenador, la BIOS realiza un chequeo del hardware denominado **POST** (*Power On Self Test*, auto diagnóstico de encendido), inicializa el hardware, carga en memoria el cargador de arranque (boot loader) del SO (que suele estar en un disco duro) y luego pasa el control a dicho cargador de arranque, que a su vez inicia el SO.

```
AMIBIOS(C)2001 American Megatrends, Inc.  
BIOS Date: 02/22/06 20:54:49 Ver: 08.00.02  
  
Press DEL to run Setup  
Checking NVRAM..  
  
128MB OK  
Auto-Detecting Pri Channel (0)...IDE Hard Disk  
Auto-Detecting Pri Channel (1)...Not Detected  
Auto-Detecting Sec Channel (0)...CDROM  
Auto-Detecting Sec Channel (1)...Not Detected
```

Uno de los principales objetivos de la BIOS "era" proporcionar servicios de E/S al SO, de forma que el software no accedía directamente al hardware, sino que lo hacía a través de la BIOS. Actualmente los SO hacen uso de sus propios servicios para acceder al hardware, sin usar los que proporciona la BIOS. En el caso de GNU/Linux, su kernel sólo utiliza los servicios de la BIOS al inicio para recopilar información sobre el hardware, pero una vez iniciado ya no los usa más.

Desde el punto de vista de los administradores de sistemas, es importante conocer el papel de la BIOS en la configuración del hardware y en el arranque.

La mayoría de las BIOS disponen de un programa denominado **SETUP**, que permite configurar sus opciones de forma interactiva. Normalmente se accede a esta herramienta pulsando *Supr*, *F1* o *F2* (depende del fabricante de la BIOS) al principio de la secuencia de arranque.

PhoenixBIOS Setup Utility					
Main	Advanced	Security	Power	Boot	Exit
System Time: [05:47:07] System Date: [10/15/2003]					Item Specific Help
Legacy Diskette A: [Disabled] Legacy Diskette B: [Disabled]					
► Primary Master	[105MB]				
► Primary Slave	[None]				
► Secondary Master	[CD-ROM]				
► Secondary Slave	[None]				
System Memory: 640 KB Extended Memory: 131071 KB Boot-time Diagnostic Screen: [Enabled]					
F1 Help Esc Exit	↑↓ Select Item ← Select Menu	-/+ Enter	Change Values Select ► Sub-Menu	F9 Setup Defaults F10 Save and Exit	

Una función importante de la BIOS es que permite activar y desactivar el hardware de la placa base. En la placa suelen venir integrados muchos dispositivos (controladora de disco duro, controladora USB, tarjeta de vídeo, tarjeta de red, tarjeta de sonido, etc.), y puede que en alguna ocasión este dispositivo integrado no sea el más adecuado. En este caso, se podrá desactivar el dispositivo mediante la BIOS evitando que un dispositivo hardware en desuso interfiera con el hardware que se utiliza.

Cómo activar o desactivar dispositivos depende de la BIOS de que se disponga, pero habitualmente se encuentra en los menús *Integrated Peripherals* (periféricos integrados) o *Advanced* (Avanzado).

Cuando GNU/Linux se haya iniciado, utilizará los drivers (controladores) para acceder al hardware del ordenador, en lugar de los servicios de la BIOS, como ya se comentó con anterioridad.

Arrancar sin teclado: Muchos ordenadores funcionan como servidores que no necesitan teclado para su funcionamiento diario. Sin embargo, en ocasiones los ordenadores se niegan a arrancar sin teclado, ya que el proceso de POST lo entiende como un problema. Para desactivar este aviso, en el SETUP de algunas BIOS se dispone de la opción **Halt On** (Detenerse) o similar. Por medio de esta opción puede indicarse a la BIOS bajo qué circunstancias debe negarse a arrancar, y una de ellas sería desactivar la comprobación del teclado.



Teclado USB: Si al sistema se conecta un teclado de tipo USB, o sin cable pero con un adaptador que se conecta al USB, es necesario activar en la BIOS la opción **USB legacy support** (a veces denominada **USB DOS function** o **USB keyboard enable**). Esto permite activar el soporte para teclados e incluso dispositivos de almacenamiento por USB como pendrives, discos duros externos, tarjetas de memoria, etc.

101.1.2. Los sistemas de ficheros proc y sysfs

proc y **sysfs** son sistemas de ficheros virtuales que se montan en los directorios */proc* y */sys* respectivamente. Ambos contienen información relativa al kernel (núcleo) del SO. La versión 2.4 del núcleo sólo conoce */proc*, donde se agrupa toda la información. La versión 2.6 del núcleo ha modificado */proc* para que delegue parte de sus tareas a */sys*.

Los sistemas de archivos */proc* y */sys* son iguales que cualquier otro sistema de archivos, por lo que es posible, por ejemplo, navegar por sus directorios mediante el comando **cd**, listar con **ls** y ver el contenido de los ficheros con **cat**. Sin embargo, ninguno de estos archivos y directorios están almacenados en ningún disco duro o algún tipo de memoria secundaria. Es el núcleo (kernel) del sistema el que genera los contenidos sobre la marcha, según se solicitan. De esta forma el sistema proporciona una forma sencilla de acceder a la información que maneja el kernel como información del hardware, procesos en ejecución, etc.

El siguiente ejemplo muestra el contenido del directorio */proc*:

\$ ls /proc							
1	14	17	1796	2245	26	989	iomem
10	1408	1734	18	2250	3	999	ioports
1007	141	1742	1804	2258	369	acpi	irq
1011	1426	1745	1807	2262	388	asound	kallsyms
1012	1496	1749	1808	2263	4	buddyinfo	kcore
1023	1498	1752	1809	2267	5	bus	keys
1026	1499	1753	1810	2268	6	cgroups	key-users
1042	15	1763	1811	2270	694	cmdline	kmsg
1052	1505	1766	1812	2271	7	cpuinfo	kpagecount
11	1531	1768	1816	2272	706	crypto	kpageflags
1125	1532	1774	1823	23	8	devices	loadavg
1147	1534	1779	184	231	834	diskstats	locks
12	1536	1781	1855	24	837	dma	meminfo
13	16	1783	1862	2482	899	dri	misc
130	1609	1785	1867	2495	9	driver	modules
131	1635	1787	19	25	934	execdomains	mounts
							sched_debug
							scsi
							self
							slabinfo
							softirqs
							stat
							swaps
							sys
							sysrq-trigger
							sysvipc
							timer_list
							timer_stats
							tty
							uptime
							version
							vmallocinfo

132	164	1788	2	2540	949	fb	mtrr	vmstat
133	1670	1790	2043	2587	967	filesystems	net	zoneinfo
135	1671	1792	21	2592	973	fs	pagetypeinfo	
137	1689	1794	22	2595	978	interrupts	partitions	

Los ficheros de */proc* proporcionan mucha información, parte de la cual se tratará más adelante. En la siguiente tabla se expone la información que contienen algunos de los ficheros de */proc*.

Algunos ficheros de <i>/proc</i>	
<i>Nombre del fichero</i>	<i>Descripción</i>
interrupts	Los parámetros IRQ (ver apartado 1.1.3. <i>IRQ</i>).
cpuinfo	Detalles sobre los microprocesadores.
dma	Los parámetro DMA (ver apartado 1.1.5. <i>Canales de DMA</i>).
ioports	Direcciones de entrada y salida (E/S) (ver apartado 1.1.4. <i>Direcciones de E/S</i>).
devices	Periféricos presentes.
meminfo	Estado global de la memoria.
loadavg	Carga del sistema.
uptime	Tiempo transcurrido desde el arranque.
version	Detalles de la versión de GNU/Linux.
modules	Módulos cargados en el núcleo de GNU/Linux.
swaps	Lista y estado de las particiones de intercambio.
partitions	Lista y estado de las particiones conocidas del sistema.
mounts	Sistemas de ficheros montados.
pci	Detalles de los dispositivos PCI.

Es posible listar el contenido de estos ficheros para obtener información del sistema y el hardware; por ejemplo, para conocer los detalles de la CPU:

```
# cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
```

```

cpu family      : 6
model          : 37
model name     : Intel(R) Core(TM) i5 CPU        M 460 @ 2.53GHz
stepping        : 5
cpu MHz         : 2505.710
cache size      : 6144 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level    : 5
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 syscall nx lm constant_tsc up dni monitor
ssse3 lahf_l
bogomips        : 5011.42
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
power management:

```

En `/proc` también hay directorios que agrupan información por tema. En la siguiente tabla se exponen algunos de estos directorios.

Algunos directorios de <code>/proc</code>	
Nombre del directorio	Descripción
<code>/proc/scsi</code>	Información sobre el bus SCSI.
<code>/proc/ide</code>	Información sobre el bus IDE/PATA.
<code>/proc/net</code>	Información sobre la red.
<code>/proc/sys</code>	Parámetros y configuración dinámica del núcleo.
<code>/proc/<PID></code>	Información sobre el proceso con identificador <PID> (<i>Process IDentifier</i>).

Algunos ficheros de `/proc/sys` y `/sys` tienen la particularidad de que se puede modificar su contenido, de forma que el kernel tiene en cuenta estas modificaciones sin que sea necesario reiniciar la máquina.

Un ejemplo sería activar el **forwarding IP** para que el sistema se comporte como un enrutador (router):

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

101.1.3. Controladores de dispositivo (drivers)

Un **controlador de dispositivo** (*device driver*, o simplemente **driver**) es un programa (software) que permite al SO interactuar con un periférico o dispositivo de E/S, haciendo una abstracción del

hardware y proporcionando un medio al SO para comunicarse con él. Se puede decir que es como un manual de instrucciones que le indica al SO cómo debe controlar y comunicarse con un dispositivo en particular. Por tanto, es una pieza esencial, sin la cual no se podría usar el hardware.

Existen tantos tipos de controladores como tipos de periféricos, y es común encontrar más de un controlador posible para el mismo dispositivo, cada uno ofreciendo un nivel distinto de funcionalidades. Por ejemplo, aparte de los oficiales (normalmente disponibles en la página web del fabricante), se pueden encontrar también los proporcionados por el SO, o también versiones no oficiales hechas por terceros.

En GNU/Linux los drivers se encuentran, o bien integrados en el kernel del SO (incluidos en el núcleo en tiempo de compilación), o bien en forma de módulos que se pueden cargar en el kernel (en tiempo de ejecución o durante el inicio del sistema). Para más información consultar el apartado sobre *Módulos del kernel*.

101.1.4. IRQ

IRQ son las siglas de **Interrupt ReQuest** (petición de interrupción). Las IRQ son señales que envían los dispositivos a la CPU para solicitar su atención, de forma que la CPU deje lo que está haciendo y se encargue de tratar dicho evento o interrupción. Cuando la CPU es interrumpida mediante una señal IRQ, la CPU, con el número de la interrupción, consulta el vector de interrupciones. El vector de interrupciones es una tabla almacenada en la memoria principal (RAM) que contiene información que le dice a la CPU lo que tiene que hacer para atender dicha interrupción. Una vez la CPU ha atendido la interrupción, sigue con lo que estaba haciendo.

Un ejemplo serían las interrupciones generadas por el teclado cuando se pulsan sus teclas.

En los microprocesadores de la familia x86, las IRQ van del 0 al 15, y en las arquitecturas más modernas (como la familia x86-64 de 64 bits) hay más de 16 interrupciones.

Algunas interrupciones tienen usos específicos, como el teclado o el reloj en tiempo real (RTC, Real Time Clock), otras tienen usos comunes y se pueden reasignar, y otras se dejan libres (sin uso) para posibles dispositivos que se añadan en el futuro.

Las IRQ y sus usos comunes		
IRQ	Uso	Observación
0	Reloj del sistema	Reservada para uso interno.
1	Teclado	Reservada para el teclado.
2	Para el rango 8-15 de las IRQ	Truco de las arquitecturas x86 para poder gestionar más de 8 interrupciones. Se utiliza para gestionar las IRQ de la 8 a la 15.
3	Segundopuerto serie (COM2)	Comunicaciones vía RS-232. Puede ser compartido con COM4.
4	Primer puerto serie (COM1)	Comunicaciones vía RS-232. Puede ser compartida con COM3.

5	Tarjeta de sonido o segundo puerto paralelo (LPT2)	Al puerto paralelo se conectan impresoras o escáneres.
6	Disquetera	Reservado para la disquetera.
7	Primer puerto paralelo (LPT1)	Al puerto paralelo se conectan impresoras o escáneres.
8	Reloj de tiempo real	RTC.
9	Interrupción abierta	Libre asignación.
10	Interrupción abierta	Libre asignación.
11	Interrupción abierta	Libre asignación.
12	Ratón PS/2	
13	Coprocesador matemático	Reservado para uso interno.
14	Controlador ATA primario	Para dispositivos ATA/IDE: discos duros y unidades ópticas; dentro de GNU/Linux suele ser: /dev/hda y /dev/hdb, ya que IDE soporta hasta dos dispositivos por controladora (maestro y esclavo).
15	Controlador ATA secundario	Para dispositivos ATA/IDE: discos duros y unidades ópticas; dentro de GNU/Linux suele ser: /dev/hdc y /dev/hdd, ya que IDE soporta hasta dos dispositivos por controladora.

En los sistemas x86-64 las IRQ se suelen asignar como en la tabla anterior, y a las IRQ superiores se les puede asignar hardware adicional.

Una vez iniciado el sistema GNU/Linux es posible explorar qué IRQ se están utilizando y para qué fines, examinando el contenido del fichero `/proc/interrupts`:

```
$ cat /proc/interrupts
CPU0
0:      50  IO-APIC-edge    timer
1:     132  IO-APIC-edge    i8042
6:       2  IO-APIC-edge    floppy
8:       0  IO-APIC-edge    rtc0
9:       0  IO-APIC-fasteoi  acpi
12:     188  IO-APIC-edge    i8042
14:       0  IO-APIC-edge    ata_piix
15:     797  IO-APIC-edge    ata_piix
19:     571  IO-APIC-fasteoi  eth0
20:     507  IO-APIC-fasteoi  vboxguest
21:   13449  IO-APIC-fasteoi  ahci, Intel 82801AA-ICH
22:       25  IO-APIC-fasteoi  ohci_hcd:usb1
NMI:        0 Non-maskable interrupts
```

LOC:	18532	Local timer interrupts
SPU:	0	Spurious interrupts
PMI:	0	Performance monitoring interrupts
IWI:	0	IRQ work interrupts
RES:	0	Rescheduling interrupts
CAL:	0	Function call interrupts
TLB:	0	TLB shootdowns
TRM:	0	Thermal event interrupts
THR:	0	Threshold APIC interrupts
MCE:	0	Machine check exceptions
MCP:	2	Machine check polls
ERR:	0	
MIS:	0	

La ejecución del comando anterior muestra en la primera columna los números de IRQ y en la última columna los nombres de los drivers que utilizan cada IRQ.

En el ejemplo anterior se puede ver que la interrupción 12, correspondiente al puerto PS/2 del ratón, está asociada al driver "i8042" (curiosamente, i8042 es el nombre del chip que controla los puertos PS/2), o que la 6 (correspondiente a la disquetera) lleva el driver "floppy".

El fichero */proc/interrupts* lista sólo las IRQ que están en uso en GNU/Linux, y se debe tener en cuenta que hasta que el SO no utiliza una IRQ no se carga el driver correspondiente, por lo que puede que no aparezca en la lista hasta que se intente utilizar el hardware. Por este motivo, puede que haya interrupciones configuradas en el sistema pero que no se estén listando en el fichero *interrupts*.

Si el sistema experimenta conflictos de IRQ, se debe reconfigurar uno o más dispositivos para que utilicen distintas IRQ. Se tratará el tema más adelante.

101.1.5. Direcciones de E/S

La arquitectura de los ordenadores actuales cumple con lo que se denomina MMIO (Memory Mapped Input Output, entrada/salida mapeada en memoria). Esta arquitectura hace que el acceso a los dispositivos hardware de E/S (teclado, ratón, impresora, webcam, etc.) desde la CPU sea igual que un acceso a memoria RAM. Esto es, enviar datos a un dispositivo es como una escritura en RAM, y recibir información desde un dispositivo es como una lectura en RAM. Esto hace que la CPU no tenga que distinguir entre un acceso a RAM o un acceso a un dispositivo de E/S (entrada/salida).

Para hacer esto posible se reserva un conjunto de direcciones de memoria para la comunicación entre la CPU y los dispositivos físicos de E/S. Este conjunto de direcciones de memoria se denomina direcciones de E/S. Cuando la CPU escriba o lea en estas direcciones de memoria, realmente se estará comunicando con un dispositivo de E/S, y no con la memoria RAM.

NOTA: En ocasiones el sistema no detecta el 100% de la memoria RAM que se le ha instalado, y el motivo es que un subconjunto de las direcciones de memoria que se debían emplear para el acceso a la memoria RAM se ha reservado para las operaciones de E/S (comunicación de los dispositivos de E/S).

Al igual que las IRQ, las direcciones de E/S suelen estar asociadas con dispositivos específicos y, por lo general, no se deben compartir.

Dispositivos habituales de GNU/Linux

Dispositivo en GNU/Linux	Nombre en Windows	IRQ habitual	Dirección de E/S
--------------------------	-------------------	--------------	------------------

/dev/ttyS0	COM1	4	0x03F8
/dev/ttyS1	COM2	3	0x02F8
/dev/ttyS2	COM3	4	0x03E8
/dev/ttyS3	COM4	3	0x02E8
/dev/lp0	LPT1	7	0x0378-0x037F
/dev/lp1	LPT2	5	0x0278-0x027F
/dev/fd0	A:	6	0x03F0-0x03F7
/dev/fd1	B:	6	0x0370-0x0377

En la tabla 2 podemos observar que para algunos dispositivos se utiliza más de una dirección de E/S (rango de direcciones).

Cuando GNU/Linux está en ejecución es posible conocer las direcciones de E/S que están en uso examinando el contenido del fichero */proc/ioports*:

```
$ cat /proc/ioports
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-0060 : keyboard
0064-0064 : keyboard
0070-0071 : rtc_cmos
  0070-0071 : rtc0
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : 0000:00:01.1
  0170-0177 : ata_piix
01f0-01f7 : 0000:00:01.1
  01f0-01f7 : ata_piix
0376-0376 : 0000:00:01.1
  0376-0376 : ata_piix
03c0-03df : vesafb
03f2-03f2 : floppy
03f4-03f5 : floppy
[...]
```

En el listado anterior (que ha sido truncado [...]) se puede observar en la primera columna el rango de direcciones de E/S reservado y en la segunda columna el nombre del dispositivo para el que se reservan.

Por ejemplo, se puede observar que para el teclado (keyboard) se han reservado las direcciones 0x0060 y 0x0064.

Como ocurre con las IRQ, si el sistema experimenta conflictos con las direcciones de E/S, será necesario reconfigurar uno o más dispositivos para que sus direcciones no coincidan. Se tratará el tema más adelante.

101.1.6. Canales de DMA

El **DMA** (*Direct Memory Access*, acceso directo a memoria) es otro método para la comunicación con los dispositivos de E/S. En vez de hacer que la CPU se encargue de la transferencia de datos entre un dispositivo de E/S y la memoria RAM, mediante DMA el dispositivo puede transferir

directamente información a la memoria sin intervención de la CPU. Lo que se consigue es dejar libre a la CPU de las tareas de E/S para que pueda ocuparse de otras tareas, mejorando así el rendimiento del sistema.

En la arquitectura x86 existen varios canales DMA, de forma que cada canal puede ser utilizado por un dispositivo concreto.

Para conocer los canales DMA que están en uso en el sistema se puede consultar el fichero `/proc/dma`:

```
$ cat /proc/dma
2: floppy
4: cascade
```

El comando del ejemplo anterior indica que los canales DMA 2 y 4 están en uso.

Al igual que las IRQ y las direcciones de E/S, los canales DMA normalmente no se deberían compartir, ya que puede dar lugar a problemas.

101.1.7. Plug And Play (PnP)

La mayoría de los sistemas actuales disponen de la tecnología **Plug & Play** (PnP, conectar y listo), que permite que los recursos de los dispositivos (IRQ, direcciones de E/S y canales DMA) sean configurados de forma automática por la BIOS.

La forma en que se asignen los recursos a los dispositivos depende de la tecnología de bus de que se trate.

Por ejemplo, el bus **PCI** (Peripheral Component Interconnect) es PnP y los recursos de los dispositivos conectados a este tipo de bus son asignados por la BIOS durante el arranque.

En cambio, el bus **ISA** (Industry Standard Architecture) no es PnP (aunque hubo algún intento de conseguir el PnP en ISA casi al final de la historia de este bus), por lo que en este caso los recursos de los dispositivos conectados son asignados por el SO y no por la BIOS. Las herramientas *isapnptools* se utilizaban por controlar la asignación de recursos en los dispositivos ISA mediante el comando *isapnp* y el fichero de configuración */etc/isapnp.conf*. Desde la versión de kernel 2.4 el soporte para PnP está integrado en el mismo kernel de GNU/Linux y estas herramientas se han vuelto obsoletas. Otro comando que han quedado obsoleto es *lspnp* y el fichero */proc/bus/pnp*.

Actualmente todos los dispositivos PCI y posteriores (AGP, PCI-Express) son PnP.

101.1.8. HotPlug y ColdPlug

Se debe distinguir entre dos tipos de dispositivos:

- *ColdPlug* (conexión en frío): se pueden conectar y desconectar físicamente al equipo únicamente cuando está apagado.
- *HotPlug* (conexión en caliente): se pueden conectar y desconectar físicamente al equipo encendido.

Los dispositivos de conexión en frío están diseñados para conectarse y desconectarse físicamente sólo cuando el ordenador está apagado, por lo que intentar conectarlos o desconectarlos en plena ejecución puede dañar al propio dispositivo o al ordenador.

Los componentes internos del ordenador, como la CPU, la memoria RAM, las tarjetas PCI o los discos duros, son dispositivos de conexión en frío. Aún así, en el caso de PCI, existe una variante de conexión en caliente que normalmente encontramos en servidores y otros sistemas que no se pueden permitir el contratiempo que supone instalar o quitar un dispositivo (*HotSwap*).

Los dispositivos que se conectan al sistema por **USB** (Universal Serial Bus, bus serie universal) o **Firewire** (IEEE-1394), por ejemplo, son de conexión en caliente; se puede conectar y desconectar estos dispositivos según convenga. Este tipo de dispositivos necesitan de un software especial que los detecte según se conecten o desconecten (*usbmgr*, *hotplug*, *hal*, *udev*,...).

101.1.9. Los ficheros de dispositivo (device nodes)

Los periféricos suelen estar vinculados a una controladora, por ejemplo, controladoras IDE o SATA para discos duros y unidades ópticas, o controladoras para USB para dispositivos de este tipo. La controladora debe saber controlar uno o más periféricos vinculados a ella, encargándose de comunicar los periféricos que gestiona con el microprocesador y la memoria RAM.

En cuando a GNU/Linux, gestiona la controladora y sus periféricos con la ayuda de los **drivers** (un driver para la controladora y uno o varios para los periféricos relacionados). Por ejemplo, un driver para la controladora SCSI (también denominado adaptador de host SCSI), otro para el disco duro SCSI, otro para el CD-ROM SCSI, etc.

En GNU/Linux los procesos acceden a los periféricos mediante unos ficheros especiales, denominados **ficheros de dispositivo** (o *device nodes*, nodos de dispositivo). Así, cuando un proceso quieren enviar información a un periférico, lo que hará será escribir en su fichero de dispositivo; por el contrario, si lo que quiere es recuperar información proveniente del dispositivo, lo que hará será leer del fichero de dispositivo correspondiente al periférico. El driver del dispositivo será el que interprete las lecturas/escrituras realizadas en este fichero por los procesos. Este sistema de ficheros especiales simplifica y facilita en gran medida el acceso de los procesos a los periféricos.

Por convención, los ficheros de dispositivo se colocan en el directorio */dev*; estos disponen de un í-nodo único, por lo que es posible conocer sus atributos como cualquier otro fichero:

```
$ ls -l /dev | grep "^[cb]"  
crw----- 1 root      root      5,   1 ene 29 18:34 console  
crw----- 1 root      root     10,  62 ene 29 18:33 cpu_dma_latency  
crw-rw-rw- 1 root      root      1,   7 ene 29 18:33 full  
crw-rw---- 1 root      fuse     10, 229 ene 29 18:34 fuse  
crw----- 1 root      root    252,   0 ene 31 22:35 hidraw0  
crw----- 1 root      root     10, 228 ene 29 18:33 hpet  
crw----- 1 root      root      1,   11 ene 29 18:33 kmsg  
brw-rw---- 1 root      disk     7,   0 ene 29 18:34 loop0  
brw-rw---- 1 root      disk     7,   1 ene 29 18:34 loop1  
brw-rw---- 1 root      disk     7,   2 ene 29 18:34 loop2  
brw-rw---- 1 root      disk     7,   3 ene 29 18:34 loop3  
brw-rw---- 1 root      disk     7,   4 ene 29 18:34 loop4  
brw-rw---- 1 root      disk     7,   5 ene 29 18:34 loop5  
brw-rw---- 1 root      disk     7,   6 ene 29 18:34 loop6  
brw-rw---- 1 root      disk     7,   7 ene 29 18:34 loop7  
crw----- 1 root      root    10, 227 ene 29 18:33 mcelog  
crw-r----- 1 root      kmem     1,   1 ene 29 18:33 mem  
crw----- 1 root      root     10,  61 ene 29 18:33 network_latency  
crw----- 1 root      root     10,  60 ene 29 18:33 network_throughput  
crw-rw-rw- 1 root      root      1,   3 ene 29 18:33 null  
crw-r----- 1 root      kmem     1,   4 ene 29 18:33 port  
crw----- 1 root      root    108,   0 ene 29 18:33 ppp  
crw----- 1 root      root     10,   1 ene 29 18:33 psaux  
crw-rw-rw- 1 root      root      5,   2 feb  1 16:57 ptmx  
crw-rw-rw- 1 root      root     1,   8 ene 29 18:33 random  
[...]
```

NOTA: En el listado anterior se ha filtrado con el comando "grep" las líneas que comienzan por "c" o por "b".

Como se puede observar en el ejemplo anterior, el primer carácter identifica al tipo de periférico:

- **c**: periférico de tipo carácter.
- **b**: periférico de tipo bloque.

Esta distinción determina cómo se comunica el sistema con el driver del periférico en cuestión. En **modo carácter** no se utilizan buffers del sistema y el intercambio se hace byte a byte; es útil para periféricos como el teclado o el ratón. En **modo bloque**, el sistema intercambia con el periférico bloques de bytes, por lo que en este caso si es necesario uso de buffers del sistema, siendo así más rápida la comunicación con periféricos como los discos duros.

Los otros dos atributos esenciales de un fichero de dispositivo son los **números mayor** y **menor**. Esta información la presenta el comando **ls** en lugar del tamaño del fichero (son dos números separados por una coma ",").

```
$ ls -l /dev/sda1  
brw-rw---- 1 root disk 8, 1 ene 29 18:33 /dev/sda1
```

En el ejemplo anterior se muestra el fichero de dispositivo de tipo bloque (b) cuyos números mayor y menor son 8 y 1 respectivamente.

El número mayor identifica al driver del dispositivo (y por consiguiente al tipo de dispositivo: SATA, IDE, SCSI, USB, ...) y el número menor suele identificar al periférico concreto. En otras ocasiones el número menor puede designar una particularidad del periférico, como la partición de un disco duro, el número de tarjeta (en caso de que haya varias tarjetas idénticas, como por ejemplo varias tarjetas de red), etc.

Algunos de los ficheros de dispositivo habituales (según la distribución)

Ruta del fichero	Descripción
/dev/mem	Memoria física.
/dev/kmem	Memoria virtual.
/dev/console	Consola maestra (también /dev/syscon).
/dev/tty	Entrada/salida estándar del proceso en ejecución.
/dev/mouse	El ratón.
/dev/swap	Unidad de intercambio (disco swap).
/dev/null	La basura UNIX. Lo que se escribe en él se elimina.
/dev/root	Sistema de ficheros raíz.
/dev/dump	Disco donde el núcleo hace un volcado de información en caso de entrar en modo pánico.
/dev/rmt0	Lector de cinta magnética en modo carácter.
/dev/fd0	Disquetera en modo bloque.
/dev/pts/1	Entrada/salida del proceso en curso en UNIX System V (y GNU/Linux)
/dev/lp0	Puerto paralelo (LPT1).
/dev/ttyS0	Puerto serie (COM1)

/dev/ttyS1	Puerto serie (COM2)
/dev/psaux	Puerto PS/2 para ratón.
/dev/sound	Tarjeta de sonido.
/dev/dsp	Procesador DSP de la tarjeta de sonido.
/dev/sequencer	Secuenciador MIDI de la tarjeta de sonido.
/dev/usb/*	Periféricos USB.
/dev/hdx	Discos IDE (donde "x" es una letra).
/dev/sdx	Discos SCSI/SATA/USB (donde "x" es una letra).

El comando **mknod** permite crear ficheros de dispositivo. Aunque GNU/Linux dispone de métodos particulares que se encargan de crear automáticamente los ficheros de dispositivo, en ocasiones puede ser necesario crear estos ficheros de forma manual:

```
# mknod /dev/periferico tipo mayor menor
```

Según la distribución, es posible que el directorio */dev* sea un sistema de ficheros completamente dinámico (de tipo **devfs** o **udev**, que se tratará en el siguiente apartado), cuyo contenido cambia en función de la presencia o no de periféricos, por lo que no será posible crear en dicha ubicación los ficheros de dispositivo de forma manual. En estos casos, el driver del periférico y el demonio **devfsd** o **udevd**, según el caso, se encargan de la creación del fichero del dispositivo.

En otras ocasiones, sólo una parte del directorio */dev* es dinámica, como el soporte de USB con el sistema de ficheros **usbdevfs**.

Para conocer si estos sistemas de ficheros de gestión de dispositivos de E/S (periféricos) están montados en el sistema, es posible ejecutar la siguiente orden:

```
# mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc
(rw,noexec,nosuid,nodev)
none on /proc/bus/usb type usbfs (rw)
```

En el ejemplo anterior se puede apreciar como hay un sistema de ficheros **udev** montado en */dev*, por lo que la gestión del contenido del directorio */dev* es completamente dinámica.

101.1.10. Administración de dispositivos

En los sistemas GNU/Linux se dispone de varias utilidades que permiten administrar los dispositivos, tanto de conexión en caliente como en frío:

- **sysfs**: El sistema de ficheros virtual *sysfs*, montado en */sys*, ofrece información sobre los dispositivos para que las aplicaciones del espacio del usuario puedan acceder a ésta. Al igual que con el sistema de ficheros virtual */proc*, podemos explorarlo mediante los comandos **cd**, **ls**, etc.

Modo kernel y modo usuario: En los sistemas GNU/Linux el software se puede ejecutar de dos

modos diferentes: en modo kernel (núcleo) o en modo usuario. Los programas que se ejecutan en modo usuario se dice que están en el *espacio del usuario*, y los que se ejecutan en modo kernel se dice que están en el *espacio del kernel*. Los programas del espacio del kernel no tienen ningún tipo de limitación en cuando al acceso a los recursos del sistema, es por esto que el núcleo del SO se ejecuta en este modo. En cambio, los programas en el espacio del usuario tienen restricciones de cara al acceso a muchos de los recursos del sistema (por ejemplo, de no poder acceder directamente al hardware); los programas de los usuarios (incluido **root**) se ejecutan en este modo.

- **hald (HAL Daemon)**: El demonio HAL o **hald** (Hardware Abstraction Layer, capa de abstracción de hardware) es un servicio que se ejecuta en el espacio del usuario que proporciona información sobre el hardware del sistema a otros programas del espacio del usuario.
- **dbus (D-Bus)**: El Desktop BUS es otro demonio (daemon, servicio) que proporciona acceso a la información del hardware. El D-Bus permite a los procesos comunicarse entre sí mediante eventos; esto es, otros procesos actuarán en función de determinados eventos notificados por otros procesos o incluso por el propio hardware (por ejemplo, un evento provocado al detectar un nuevo dispositivo USB conectado al equipo).
- **udev**: Anteriormente, en los sistemas GNU/Linux, se creaban de forma estática ficheros para los distintos dispositivos (device nodes) dentro del árbol de directorios `/dev` mediante el comando **mknod**. Pero este sistema presentaba una serie de problemas, sobre todo con los dispositivos de conexión en caliente (hotplug). Con **udev** se han resuelto estos problemas a la hora de gestionar el árbol de directorios `/dev` y los ficheros de dispositivos. **udev** es un sistema de archivos virtual, montado en `/dev`, que gestiona los ficheros de los dispositivos de forma dinámica conforme se cargan y descargan los drivers en el kernel. Es posible configurar **udev** mediante los ficheros que hay dentro del directorio `/etc/udev`, pero no suele ser necesario hacerlo.

El núcleo gestiona eventos y mensajes que **udev** lee e interpreta. **udev** dispone de una serie reglas que aplica en respuesta de los mensajes generados por el núcleo. A continuación se presentan tres reglas de ejemplo:

```
KERNEL=="raw1394*",GROUP="video"
KERNEL=="dv1394*",SYMLINK+="dv1394/%n",GROUP="video"
KERNEL=="video1394*",SYMLINK+="video1394/%n",GROUP="video"
```

El significado de las reglas es el siguiente:

- **KERNEL**: nombre del evento generado por el núcleo.
- **GROUP**: nombre del grupo al que pertenecerá el fichero de dispositivo creado.
- **SYMLINK**: crea un enlace simbólico al fichero del dispositivo creado con el nombre indicado. %n representa el número de orden en que se detectó el dispositivo y se creó su fichero.

En el ejemplo anterior, si conectamos una cámara de vídeo digital a un puerto Firewire, el núcleo generará un evento que comienza por "video1394", ejecutándose la regla correspondiente (en nuestro caso, la última que se ha definido):

1. Se creará `/dev/video1394`.
2. Se establecerá como grupo "video".
3. Se creará el enlace simbólico `/dev/video1394/0` (el primero).

Las reglas se colocan en `/dev/udev/rules.d`. Es posible crear reglas propias creando un fichero con las mismas con extensión ".rules" en este directorio: por ejemplo `99-local.rules`.

Todas estas herramientas facilitan la comunicación entre los procesos y los dispositivos, posibilitando el acceso a la información del hardware a los procesos y proporcionando mecanismos para que el hardware notifique a los procesos cualquier cambio en su configuración.

Mucha gente trata los dispositivos de conexión en frío (como los puertos paralelo, serie o PS/2) como si fueran de conexión en caliente, conectándolos o desconectándolos cuando el equipo se encuentra encendido. En la mayoría de las ocasiones no suele ocurrir nada, pero existe el riesgo de dañar el dispositivo o el mismo puerto de conexión, por lo que siempre es recomendable apagar el ordenador antes de conectar o desconectar este tipo de dispositivos. Asimismo, cuando realizamos estas conexiones en caliente los demonios anteriores (como `hal` o `udev`) no lo detectan, ya que el SO sólo gestiona el puerto y no el dispositivo que se conecta a dicho puerto.

101.1.11. Configurar el hardware

Muchos de los dispositivos hardware del ordenador requieren de una configuración: números de IRQ, direcciones de E/S y canales DMA; aunque no todos los dispositivos requieren de estos tres recursos. Inicialmente este proceso de configuración se hacía mediante jumpers. Actualmente, estas opciones se pueden configurar mediante software.

101.1.11.1. Dispositivos PCI

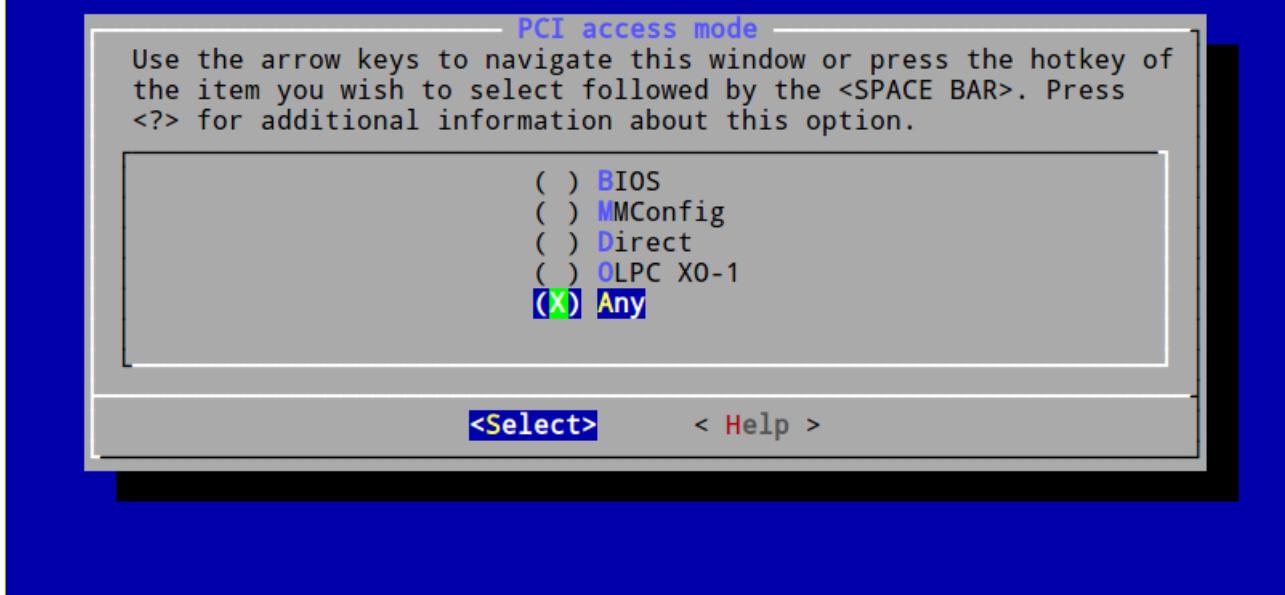
El **bus** de expansión **PCI** se emplea para conectar todo tipo de tarjetas de expansión al ordenador (tarjeta de sonido, tarjeta de red, tarjeta capturadora de TV, tarjeta de ampliación de puertos USB, tarjeta gráfica, etc., aunque esta última ha caído en desuso en favor de otros puertos especializados como AGP o PCI-Express).

101.1.11.1.1. Configuración de dispositivos PCI

Este bus, así como otros buses posteriores como AGP o PCI-Express, tienen la característica de ser *Plug And Play* (*PnP*, conectar y usar), por lo que se configuran de forma automática y no suele ser necesario realizar ningún cambio.

Aún así, es posible cambiar la forma en la que el *kernel* de GNU/Linux detecta estos dispositivos modificando la opción del núcleo "*PCI Devices -> Bus Options -> PCI Access Mode*". Esta opción se establece durante el proceso de compilación del kernel, que no trataremos en este apartado. Los valores que puede tomar la opción "PCI Access Mode" son cuatro:

- **BIOS**: el SO realiza esta tarea a través de los servicios que ofrece la BIOS.
- **MMConfig**: utiliza el protocolo *MMConfig* para detectar los dispositivos.
- **Direct**: mediante un sistema de detección específico de GNU/Linux.
- **OLPC XO-1**: sistema de detección para los portátiles de bajo coste de *One Laptop Per Child*, que requieren un consumo mínimo de energía.
- **Any**: prueba con MMConfig, luego mediante Direct y finalmente BIOS; esta es la opción más adecuada.



La mayoría de las BIOS disponen de opciones accesibles vía SETUP que permiten cambiar la asignación de los recursos a los dispositivos PCI. Esto puede ser útil para resolver problemas con los dispositivos PCI por la asignación de los mismos recursos a distintos dispositivos.

Algunos drivers de GNU/Linux admiten opciones para designar los recursos que debe usar el dispositivo. En este caso es necesario consultar la documentación que viene con los drivers para conocer los detalles sobre las opciones que admite. Luego, se podrán pasar estas opciones al kernel por medio del cargador de arranque (consultar apartado 1.2. *Inicio del sistema*).

También disponemos de una forma de ajustar y consultar directamente las configuraciones de los dispositivos PCI usando el comando **setpci**. Es útil si tenemos el conocimiento suficiente como para comunicarnos con los dispositivos PCI a bajo nivel; por este motivo, no suele ser la forma más habitual de configurar los dispositivos PCI.

101.1.11.1.2. Consultar información sobre dispositivos PCI

Para comprobar cómo están configurados los dispositivos PCI se dispone del comando **lspci**. Este comando muestra toda la información disponible sobre los buses PCI del sistema y los dispositivos conectados a estos buses.

En la tabla 3 podemos ver las distintas opciones que podemos pasarle al comando **lspci** para ajustar su comportamiento.

Opción	Opciones de lspci	Descripción
-v		Incrementa el nivel de detalle de la salida. Se puede obtener más detalle aún con las opciones "-vv" o "-vvv".
-n		Muestra la información en códigos numéricos (Vendor_ID y Device_ID) en vez de traducirlos a

	nombres de fabricante y dispositivo (Vendor y Device).
-nn	Muestra los nombres de fabricante y dispositivo y sus códigos.
-x	Muestra el espacio de configuración de PCI de cada dispositivo como un volcado hexadecimal. Se puede obtener más información con las opciones "-xxx" o "-xxxx".
-b	Muestra los números de IRQ y otros datos según los ve el propio dispositivo en vez de como los ve el kernel (/proc/interrupts, /proc/dma, /proc/ioports,...).
-t	Muestra una vista de árbol que describe la relación entre dispositivos.
-s [[[dominio]:]bus]:][ranura].[función]]	Muestra sólo los dispositivos que coinciden con la especificación dada. Es posible utilizar el comodín "*" para indicar cualquier valor en los parámetros.
-d [fabricante]:[dispositivo]	Muestra datos sobre el dispositivo especificado.
-i <i>fichero</i>	Utiliza el fichero especificado para asociar las ID de fabricante (vendor) y dispositivo (device) con nombres. Por defecto, se utiliza el fichero /usr/share/misc/pci.ids.
-m	Vuelva los datos en un formato más amigable para utilizar desde scripts. La opción "-mm" muestra un formato más reciente.
-D	Muestra los números de dominio de PCI, que por defecto no se muestran.
--version	Muestra la información de versión del comando.

A continuación podemos ver un ejemplo de uso del comando **lspci** con la opción "-nn", donde se listan los dispositivos PCI conectados al sistema mostrándose los nombres y los ID de cada fabricante y dispositivo encontrado:

```
$ lspci -nn
00:00.0 Host bridge [0600]: Intel Corporation 440FX - 82441FX PMC [Natoma]
[8086:1237] (rev 02)
00:01.0 ISA bridge [0601]: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton
II] [8086:7000]
00:01.1 IDE interface [0101]: Intel Corporation 82371AB/EB/MB PIIX4 IDE
[8086:7111] (rev 01)
00:02.0 VGA compatible controller [0300]: InnoTek Systemberatung GmbH VirtualBox
Graphics Adapter [80ee:beef]
00:03.0 Ethernet controller [0200]: Intel Corporation 82540EM Gigabit Ethernet
Controller [8086:100e] (rev 02)
00:04.0 System peripheral [0880]: InnoTek Systemberatung GmbH VirtualBox Guest
Service [80ee:cafe]
00:05.0 Multimedia audio controller [0401]: Intel Corporation 82801AA AC'97
Audio Controller [8086:2415] (rev 01)
00:06.0 USB Controller [0c03]: Apple Computer Inc. KeyLargo/Intrepid USB
[106b:003f]
00:07.0 Bridge [0680]: Intel Corporation 82371AB/EB/MB PIIX4 ACPI [8086:7113]
(rev 08)
00:0d.0 SATA controller [0106]: Intel Corporation 82801HBM/HEM (ICH8M/ICH8M-E)
```

Cada línea de la salida del comando **`lspci`** corresponde a un dispositivo PCI. La forma de interpretar estas líneas es la siguiente:

`bus:ranura.función nombre_dispositivo [id_dispositivo]: nombre_fabricante [id_fabricante]`

La primera columna, que corresponde a bus, ranura y función (bus, slot, function) del dispositivo, identifica el puerto PCI donde se conecta físicamente. Para traducir cada *id_dispositivo* (Device ID) e *id_fabricante* (Vendor ID) se utiliza como base de datos el fichero `/usr/share/misc/pci.ids`.

101.1.11.2. Dispositivos USB

Actualmente la totalidad de los ordenadores traen puertos USB, siendo la interfaz externa de conexión de dispositivos más extendida.

101.1.11.2.1. Fundamentos de USB

USB (Universal Serial Bus, bus serie universal) es un protocolo y un puerto de hardware para transferir datos desde y hacia dispositivos. Admite gran variedad de dispositivos así como soporta la conexión simultánea de más de un dispositivo por puerto (más que ATA o SCSI), así como ofrece unas tasas de transferencia considerablemente superiores a puertos como el paralelo y el serie.

Versiones de USB	
Versión	Velocidad (tasa de transferencia)
1.0	1.5 Mbps
1.1	12 Mbps
2.0	480 Mbps
3.0	4.8 Gbps

Un puerto USB soporta hasta 127 dispositivos y suministra un voltaje de 5V, por lo que muchos dispositivos no requieren de alimentación externa.

Es el puerto preferido para muchos dispositivos externos como impresoras, escáneres, ratones, cámaras web, etc.

Cada puerto puede controlar un dispositivo, pero es posible emplear un **concentrador (hub)** de USB para conectar varios dispositivos en un mismo puerto. De esta forma, se pueden conectar un número enorme de dispositivos a un mismo puerto, aunque en la práctica no es recomendable abusar, sobre todo si los dispositivos que conectamos transmiten grandes cantidades de datos y si los dispositivos sólo reciben alimentación del USB (no habrá suficiente suministro eléctrico para todos con 5V).

101.1.11.2.2. Controladoras USB

Existen muchas controladoras USB diferentes, como UHCI, OHCI, EHCI y R8A66597. Las distribuciones actuales de GNU/Linux traen los drivers para las controladoras USB habituales, por lo que lo normal es que los puertos USB se activen automáticamente al iniciar el ordenador.

Las controladoras **UHCI** (*Universal Host Controller Interface*) y **OHCI** (*Open Host Controller Interface*) gestionan los dispositivos USB 1.x, **EHCI** (*Enhanced Host Controller Interface*) gestiona USB 2.0, y **XHCI** (*eXtensible Host Controller Interface*) gestiona USB 3.0. Las controladoras de

versiones superiores también gestionan los dispositivos de versión inferior.

101.1.11.2.3. Consultar información sobre los dispositivos USB

Se puede conocer los detalles de los dispositivos USB mediante el comando **lsusb**. El siguiente ejemplo muestra un uso simple del comando sin opciones, donde se puede ver información básica sobre los dispositivos:

```
$ lsusb
Bus 002 Device 002: ID 80ee:0021
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 006: ID 1210:2604 DigiTech
Bus 001 Device 004: ID 152d:2338 JMicron Technology Corp. / JMicron USA
Technology Corp. JM20337 Hi-Speed USB to SATA & PATA Combo Bridge
Bus 001 Device 002: ID 0718:0629 Imation Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

En el ejemplo anterior cada línea corresponde a un dispositivo USB. En este caso, se han detectado seis dispositivos, dos conectados al bus o controladora 002 (cuyos ID son 001 y 002) y cuatro dispositivos conectados al bus 001 (con ID 001, 002, 004 y 006). El campo etiquetado con "ID" corresponde al identificador del fabricante y del producto (separados por ":"; <1210:2604> denominado como "DigiTech" en el ejemplo anterior). Al final de cada línea encontramos un pequeño texto descriptivo del dispositivo, aunque en algunas ocasiones no es de mucha ayuda para reconocer el dispositivo del que se trata.

Nótese que los dispositivos con ID 001 en ambos buses son concentradores USB 1.1 y 2.0 respectivamente, lo que se observa fácilmente en el resultado de la orden anterior.

Opciones de lsusb	
Opción	Descripción
-v	Muestra información detallada.
-s [[bus]:][id_dispositivo]	Restringe la información de salida al número de bus y/o dispositivo especificados.
-d [fabricante]:[producto]	Restringe la información de salida a un fabricante y/o a un producto concreto.

-D <i>nombre_fichero</i>	Muestra información sobre el dispositivo al que se puede acceder mediante el fichero <i>nombre_fichero</i> , que debe encontrarse dentro del directorio <i>/proc/bus/usb</i> .
-t	Muestra la lista de dispositivos como un árbol, lo que facilitará saber qué dispositivos están conectados a controladoras específicas.

-V, --version Muestra la información de versión del comando.

En el ejemplo anterior, el dispositivo 006 conectado al bus 001 es un webcam; para obtener información detallada sobre dicho dispositivo USB se puede ejecutar la siguiente orden (-v para mostrar información detallada y -s para indicar un dispositivo concreto):

```
$ lsusb -v -s 001:006
Bus 001 Device 006: ID 1210:2604 DigiTech
Device Descriptor:
  bLength          18
  bDescriptorType   1
```

```

bcdUSB          2.00
bDeviceClass    239 Miscellaneous Device
bDeviceSubClass 2 ?
bDeviceProtocol 1 Interface Association
bMaxPacketSize0 64
idVendor        0x1210 DigiTech
idProduct       0x2604
bcdDevice       0.01
iManufacturer   1 Alcor Micro, Corp.
iProduct        2 WebCam SCB-0380M
iSerial         0
bNumConfigurations 1
Configuration Descriptor:
  bLength          9
  bDescriptorType  2
  wTotalLength     512
  bNumInterfaces   2
  bConfigurationValue 1
  iConfiguration   0
  bmAttributes     0x80
    (Bus Powered)
  MaxPower         500mA
[...]

```

También es posible mostrar la información en forma de árbol, viendo de forma gráfica dónde se conecta cada dispositivo:

```

$ lsusb -t
Bus# 2
`-Dev# 1 Vendor 0x1d6b Product 0x0001
  `-Dev# 2 Vendor 0x80ee Product 0x0021
Bus# 1
`-Dev# 1 Vendor 0x1d6b Product 0x0002
  |-Dev# 2 Vendor 0x0718 Product 0x0629
  |-Dev# 4 Vendor 0x152d Product 0x2338
  `-Dev# 6 Vendor 0x1210 Product 0x2604

```

Al igual que **lspci**, **lsusb** dispone de una base de datos de fabricantes y dispositivos que para traducir cada *id_dispositivo* (Device ID) e *id_fabricante* (Vendor ID). Esta base de datos corresponde al fichero */usr/share/misc/usb.ids*.

Las primeras implementaciones de USB en GNU/Linux necesitaban un driver independiente para cada dispositivo USB. Muchos de estos drivers siguen incluidos en el kernel, siendo aún utilizados por algunos programas. Un ejemplo son los dispositivos USB de almacenamiento, que utilizan drivers USB que interconectan con drivers SCSI (que se trata más adelante), haciendo que discos duros u otros dispositivos de almacenamiento USB (como pendrives) aparezcan como dispositivos SCSI.

GNU/Linux ofrece igualmente un sistema de ficheros virtual denominado **usbfs** que proporciona acceso a información sobre los diferentes dispositivos USB. Este sistema se encuentra montado en */proc/bus/usb*. A los subdirectorios de */proc/bus/usb* se les da nombres numéricos, existiendo un subdirectorio por cada bus USB. En el siguiente ejemplo podemos ver que existen dos buses (controladoras) USB:

```

$ ls -l /proc/bus/usb
total 0
dr-xr-xr-x 2 root root 0 ene 29 18:33 001
dr-xr-xr-x 2 root root 0 ene 29 18:33 002
-r--r--r-- 1 root root 0 ene 29 18:41 devices

```

En el sistema de ficheros virtual **usbfs** también se encuentra el fichero */proc/bus/usb/devices* con información sobre los dispositivos conectados a los buses USB.

Es posible que el sistema de ficheros virtual **usbfs** no se encuentre montado por defecto; esto se puede detectar cuando el directorio */proc/bus/usb* no existe o está vacío. Para montar de forma manual **usbfs** debemos ejecutar la siguiente orden:

```
# mount -t usbfs none /proc/bus/usb
```

Para que se monte de forma automática se debe añadir la siguiente línea al fichero */etc/fstab*:

```
none /proc/bus/usb usbfs defaults 0 0
```

En el interior de los subdirectorios correspondientes a cada bus USB se encuentran los ficheros de dispositivo de cada uno de los dispositivos USB conectados al bus, cuyo nombre de fichero corresponde con el identificador del dispositivo (Device ID). En el siguiente ejemplo se observa que hay cuatro dispositivos (cuyos ID son 001, 002, 004 y 006) conectados al bus 001:

```
$ ls -l /proc/bus/usb/001
total 0
-rw-r--r-- 1 root root 43 ene 29 18:33 001
-rw-r--r-- 1 root root 50 ene 29 18:38 002
-rw-r--r-- 1 root root 50 ene 29 18:40 004
-rw-r--r-- 1 root root 530 ene 29 18:41 006
```

101.1.11.2.4. Ficheros de dispositivos USB

El gestor de dispositivos **udev** que usa el kernel para administrar el sistema de ficheros */dev* detecta la conexión y desconexión de los dispositivos USB, creando los ficheros de dispositivo (device nodes) correspondientes a cada dispositivo USB conectado en el subdirectorio */dev/bus/usb*, siguiendo una estructura similar que */proc/bus/usb*.

Listamos información de los dispositivos USB disponibles, donde se observa que hay dos buses USB (001 y 002):

```
$ lsusb
Bus 002 Device 002: ID 80ee:0021
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 006: ID 1210:2604 DigiTech
Bus 001 Device 004: ID 152d:2338 JMicron Technology Corp. / JMicron USA
Technology Corp. JM20337 [...]
Bus 001 Device 002: ID 0718:0629 Imation Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Listar los buses USB disponibles:

```
$ ls -l /dev/bus/usb
total 0
drwxr-xr-x 2 root root 120 ene 29 18:41 001
drwxr-xr-x 2 root root 80 ene 29 18:33 002
```

Listar los ficheros de los dispositivos conectados al bus USB 001:

```
$ ls -l /dev/bus/usb/001
total 0
crw-rw-r-- 1 root root 189, 0 ene 29 18:33 001
crw-rw-r-- 1 root root 189, 1 ene 29 18:38 002
```

```
crw-rw-r-- 1 root root 189, 3 ene 29 18:40 004  
crw-rw-r-- 1 root root 189, 5 ene 29 18:41 006
```

Listar los ficheros de los dispositivos conectados al bus USB 002:

```
$ ls -l /dev/bus/usb/002  
total 0  
crw-rw-r-- 1 root root 189, 128 ene 29 18:33 001  
crw-rw-r-- 1 root root 189, 129 ene 29 18:33 002
```

101.1.11.2.5. Gestión de USB

El kernel de GNU/Linux no fue concebido originalmente para soportar la tecnología de conexión en caliente, por lo que en un inicio se apoyó de utilidades externas para estos asuntos. En concreto, las herramientas empleadas eran **usbmgr** y **hotplug**.

usbmgr es un programa que se ejecuta en segundo plano para detectar cambios en el bus USB. Cuando detecta algún cambio, carga o descarga los módulos del kernel necesarios para controlar los dispositivos. Este programa emplea los ficheros de configuración del directorio */etc/usbmgr* para gestionar dispositivos específicos y */etc/usbmgr/usbmgr.conf* para la configuración global.

hotplug, por su parte, es otro programa que también detecta cambios en el bus USB. Este sistema utiliza los ficheros almacenados en el directorio */etc/hotplug* para controlar la configuración de dispositivos USB específicos. El fichero */etc/hotplug/usb.usermap* contiene una base de datos de ID de dispositivos y rutas a scripts dentro de */etc/hotplug/usb* que se ejecutan cuando se conectan o desconectan dispositivos. Estos scripts son los que preparaban el sistema atender al nuevo dispositivo conectado.

Actualmente, las distribuciones de GNU/Linux basadas en el kernel 2.6 o posterior, han remplazado **usbmgr** y **hotplug** por los sistemas **udev** y **hal** (comentados antes en este mismo apartado) para la gestión de dispositivos de conexión en caliente.

101.1.11.3. Configurar dispositivos de almacenamiento

Los dispositivos de almacenamiento (discos duros, unidades ópticas, etc.) se encuentran entre los componentes más importantes del sistema. En los ordenadores actuales encontramos habitualmente tres tipos de interfaces para estos dispositivos:

- **PATA** (*Parallel Advanced Tecnology Attachment*, ATA paralelo), también denominados ATA o IDE (*Integrated Drive Electronics*).
- **SATA** (*Serial Advanced Tecnology Attachment*, ATA serie).
- **SCSI** (*Small Computer System Interface*, interfaz de sistema para pequeños ordenadores).

También hay dispositivos de almacenamiento externos conectados por puertos USB y Firewire (IEEE-1394), así como variantes externas de SATA (eSATA) y SCSI.

101.1.11.3.1. Dispositivos PATA

Como su nombre indica, los dispositivos PATA utilizan una interfaz en paralelo, lo que significa que por el cable se transfieren varios bits de forma simultánea. Por este motivo el cable PATA es tan ancho (40 pines para dispositivos de 3.5" y de 44 pines para los de 2.5").

Es posible conectar hasta dos dispositivos por cada conector PATA de una placa base, de forma que los cables PATA suelen tener tres conectores: uno para la placa base y dos para los dispositivos.

Los dispositivos PATA se deben configurar como **maestros** o **esclavos**. Esto se puede hacer

mediante los *jumpers* de los propios dispositivos, estableciendo uno como maestro (*master*) y otro como esclavo (*slave*). Por lo general, el maestro se conecta al final del cable y el esclavo en el conector intermedio. También existe la configuración de selección por cable (*cable select*) que también se establece mediante los jumpers; configurando el dispositivo de esta forma, se autoconfigura en función de su posición en el cable.



Para mejorar el rendimiento, es recomendable conectar los dispositivos en conectores PATA diferentes, en vez de conectarlos al mismo como maestro y esclavo. Esto es, si se dispone de dos conectores PATA en la placa base y dos dispositivos, conectar cada dispositivo a un conector diferente.

Hasta hace poco era habitual encontrar hasta dos conectores PATA en la placa base; aunque, actualmente, están siendo remplazados por los conectores SATA. Hay placas base donde coexisten ambos conectores. Los conectores PATA de la placa base se suelen identificar como IDE1 e IDE2 (si hay más de uno).

Todas las BIOS, excepto las más antiguas, detectan los dispositivos PATA automáticamente y proporcionan información sobre sus capacidades y números de modelo en el SETUP de la BIOS (para más información consultar el apartado 1.1.9.3.6. *Parámetros geométricos de los discos duros*).

En GNU/Linux se identifican los **discos duros PATA** mediante los ficheros **/dev/hda**, **/dev/hdb**, **etc.**, siendo **/dev/hda** el disco duro maestro del primer controlador PATA, **/dev/hdb** el disco esclavo del primer controlador, **/dev/hdc** el maestro del segundo controlador y **/dev/hdd** el esclavo del segundo controlador. Por lo tanto, es posible que en el sistema se disponga de los ficheros **/dev/hda** y **/dev/hdc**, pero no de **/dev/hdb** por no haber un disco duro conectado como esclavo en el primer controlador, habiendo saltos en el esquema de numeración de los ficheros de dispositivos PATA.

Si un disco duro se encuentra particionado, sus **particiones** se identifican mediante ficheros cuyo nombre coincide con el del disco pero acabado en un número, que indica el número de la partición al que se corresponde. Esto es, si el disco duro maestro del primer controlador **/dev/hda** cuenta con tres particiones, éstas se corresponden con los ficheros **/dev/hda1**, **/dev/hda2** y **/dev/hda3**.

En el caso de las **unidades ópticas** (CD-ROM, DVD, ...) también existen los ficheros de dispositivo **/dev/cdrom** o **/dev/dvd**. En este caso, no se suelen particionar. En el caso de las **unidades ZIP**, aunque son dispositivos extraíbles, se les asigna ficheros de dispositivos como a los discos duros PATA, tal y como se vio antes, siguiendo el mismo esquema en caso de particionado.

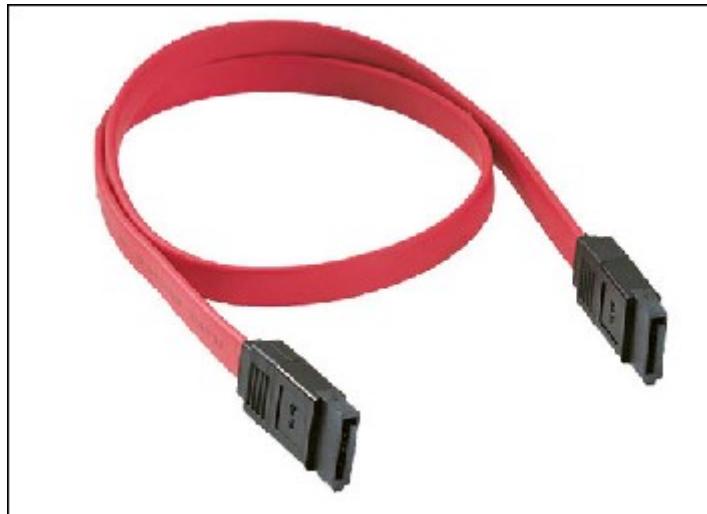
Algunos sistemas GNU/Linux disponen de drivers que tratan los dispositivos PATA como si fueran SCSI, en cuyo caso los nombres de fichero de los dispositivos PATA siguen las reglas de SCSI, que se verán a continuación.

Se puede obtener un listado de los dispositivos PATA detectados durante el arranque:

101.1.11.3.2. Dispositivos SATA

SATA es una interfaz más reciente que está reemplazando a la interfaz PATA. Actualmente, las placas base suelen incluir cuatro o más conectores SATA, coexistiendo en algunos casos con conectores PATA.

A un conector SATA, a diferencia de PATA, sólo se puede conectar un único dispositivo. Esto simplifica la configuración, ya que no habrá que configurar jumpers ni preocuparse por la posición en el cable.



Como indica su nombre, SATA (Serial ATA) transfiere los bits de información de uno en uno (en serie). Motivo por el cual los cables SATA son más estrechos que los PATA. Sin embargo, sus tasas de transferencia son muy superiores a las de PATA.

Comparativa PATA y SATA

<i>Interfaz</i>	<i>Tasa de transferencia</i>
PATA	133 MB/s
SATA I	150 MB/s
SATA II	300 MB/s
SATA III	600 MB/s

Las BIOS detectan los dispositivos SATA de forma automática, al igual que con los PATA, proporcionando información sobre los dispositivos al SO. Las BIOS también ofrecen opciones para utilizar dispositivos SATA para arrancar el sistema; aunque puede que haya BIOS que no lo soporten, sobre si de lo que se dispone es de una tarjeta controladora SATA independiente porque no vino integrada en la placa base.

La mayoría de los drivers de GNU/Linux tratan los **dispositivos SATA como si fueran SCSI**, por lo que conviene consultar el siguiente apartado para conocer la nomenclatura de los dispositivos. Existe algún caso, con drivers antiguos, que se tratan los discos SATA como si fueran PATA, por lo que en este caso se usará la nomenclatura PATA (`/dev/hdx`).

101.1.11.3.3. Dispositivos SCSI

Existen diversas definiciones de **SCSI**, que utilizan cables diferentes y funcionan a varias velocidades. SCSI ha sido tradicionalmente un bus paralelo (como PATA), aunque existe una variante reciente que funciona en serie (como SATA): **SAS** (*Serial Attached SCSI*).

SCSI es un bus superior a PATA y, por consiguiente, de mayor coste, por lo que no está tan extendido.

Admite entre 8 y 16 dispositivos por bus, dependiendo de la variante de SCSI de que se trate. Uno de los dispositivos es el propio **adaptador de host SCSI**, que es el dispositivo que comunica el sistema con los dispositivos de almacenamiento SCSI. Este adaptador de host puede venir integrado en la placa base o puede incluirse como una tarjeta de expansión. Cada dispositivo posee su propio número de ID, que se suele asignar a través de un jumper del dispositivo. Hay que asegurarse de que cada ID de dispositivo es único.

SCSI ID	0	1	2	3
Jumper Block				
SCSI ID	4	5	6 (default)	7
Jumper Block				
SCSI ID	8	9	10	11
Jumper Block				
SCSI ID	12	13	14	15
Jumper Block				

Las BIOS estándar no suelen detectar los dispositivos SCSI; sólo se podrá arrancar desde un dispositivo SCSI si el adaptador de host SCSI lo soporta (si posee una BIOS propia). Los adaptadores de host SCSI de gama alta suelen disponer de su propia BIOS, a diferencia de las de gama baja. Si empleamos un adaptador de host SCSI de gama baja, se podrá utilizar el dispositivo SCSI, pero sólo una vez iniciado el sistema GNU/Linux, ya que no se podrá arrancar el sistema desde dichos dispositivos.

La nomenclatura de los ficheros de dispositivo es la siguiente, dependiendo del tipo de dispositivo de que se trate:

- **Disco duro SCSI:** `/dev/sdx`, donde la "x" es una letra empezando por la "a".
- **Cinta SCSI:** `/dev/stx` y `/dev/nstx`, donde "x" es un número empezando por el "0".
- **CD-ROM/DVD-ROM SCSI:** `/dev/scdX`, donde "x" es un número empezando por el "0".

La numeración de los ficheros de dispositivo se realiza en orden ascendente basado en el ID SCSI. Esto es, si se dispone de un disco duro con ID SCSI 2 y otro con ID SCSI 4, se les asignarán `/dev/sda` y `/dev/sdb` respectivamente. El problema está en si se añade un nuevo dispositivo, en cuyo caso dependerá de si se le asigna ID SCSI 0, 1 ó 3. Si se le asigna 0 ó 1 el nuevo disco se convertirá

en `/dev/sda`, y si se le asigna 3 se convertirá en `/dev/sdb`, incrementando los identificadores de uno o de los dos discos anteriores (según el caso). Por este motivo se recomienda asignar los ID más bajos posibles para poder añadir futuros dispositivos empleando ID superiores.

Otro problema surge si se dispone de más de un adaptador host SCSI, ya que GNU/Linux asigna nombres de fichero de dispositivo a todos los dispositivos del primer adaptador y luego sigue por los dispositivos del segundo adaptador, de forma consecutiva.

Dispositivos de almacenamiento USB y SATA (pseudo-SCSI): Debemos recordar que estos dispositivos son tratados por GNU/Linux como si de dispositivos SCSI se tratara. Esto puede provocar que a un dispositivo SCSI se le asigne un fichero de dispositivo con numeración superior a la esperada.

Los dispositivos conectados a un bus SCSI forman una cadena, que debe ser finalizada por ambos extremos, pero no los dispositivos intermedios. Hay que consultar el manual del adaptador host SCSI o del dispositivo SCSI para saber cómo finalizarlos. Los dispositivos SCSI modernos traen un jumper para activar o desactivar la finalización. En otros casos se emplean unos dispositivos terminadores que deben ser conectados al dispositivo finalizador.



La incorrecta finalización suele derivar en problemas raros, como la imposibilidad de detectar estos dispositivos, bajo rendimiento u operaciones poco fiables. Esto puede ocurrir también al utilizar cables SCSI de baja calidad o demasiado largos.

101.1.11.3.4. Dispositivos externos

Los dispositivos de almacenamiento más comunes son **USB**, **IEEE-1394** (Firewire) y **SCSI**.

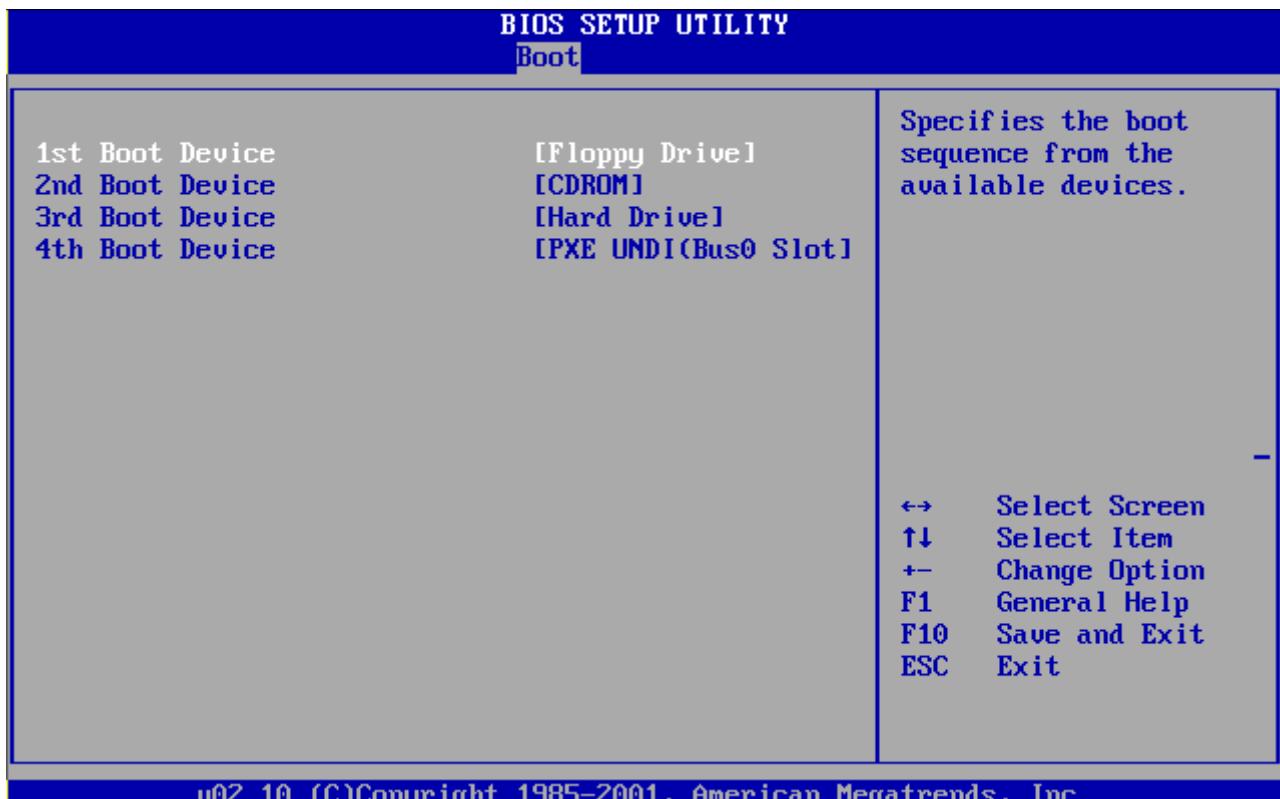
SCSI viene soportando dispositivos externos desde hace ya bastante tiempo, por lo que muchos adaptadores de host SCSI tienen tanto conectores externos como internos. Los dispositivos SCSI externos se configuran de la misma forma que los internos, aunque la asignación del ID SCSI y la finalización puede variar.

GNU/Linux, como ya se comentó anteriormente, trata a los dispositivos USB y IEEE-1394 como dispositivos SCSI desde el punto de vista del software. Lo normal es conectar el dispositivo y ver que aparece un fichero de dispositivo `/dev/sdx` y utilizarlo como si de un dispositivo SCSI se tratara. Esto es el caso de discos duros externos o de los **pendrives**.

101.1.11.3.5. Dispositivos de arranque

Normalmente, la BIOS permite elegir el orden de los dispositivos desde los que arrancar. La BIOS intentará iniciar el sistema desde el primer dispositivo de la lista, y si falla lo intentará con el segundo dispositivo; y así sucesivamente hasta que arranque desde uno de los dispositivos o hasta

agotar las posibilidades, en cuyo caso la BIOS mostrará un mensaje de error.



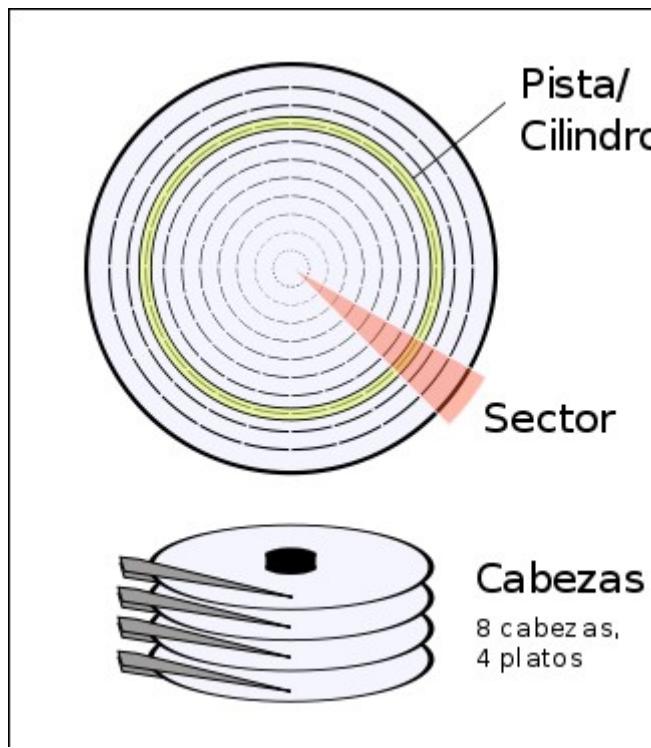
Debemos prestar especial atención al orden de los dispositivos de arranque, ya que en el caso de permitir el uso de medios extraíbles (disquete, pendrive, CD-ROM, etc.) abre la puerta a intrusos que tengan acceso físico al ordenador; sólo tienen que iniciar el sistema con un disquete o un CD-ROM para tomar el control del sistema. Por este motivo es recomendable poner siempre el disco duro como primero o único dispositivo de arranque.

Esta configuración la podremos cambiar en cualquier momento en el SETUP de la BIOS, permitiendo el arranque desde CD-ROM para, por ejemplo, instalar GNU/Linux o para iniciar desde un disco de arranque de emergencia para mantenimiento.

101.1.11.3.6. Parámetros geométricos de los discos duros

En la mayoría de los casos la BIOS detecta y configura los discos duros correctamente. Aún así, en alguna ocasión, puede ser necesario configurar manualmente en la BIOS la geometría de cilindro/cabezal/sector (CHS) del disco, teniendo que establecer a mano el número cilindros, de cabezales y de sectores disponibles en el mismo.

En cuanto a la estructura física de un disco duro, éste está formado por **platos** (*platters*), habiendo dos **superficies** por plato. Sobre cada superficie se mueve un **cabezal** de **lectura/escritura** (*headers*). Cada superficie se divide en círculos concéntricos denominados **pistas** (*tracks*), y cada pista se divide en **sectores**. Asimismo, el mismo número de pista en todas las superficies se denomina **cilindro** (*cylinder*). La información se almacena en bloques de bytes (normalmente 512 bytes) dentro de cada sector. En la siguiente figura se puede apreciar las partes que forman un disco duro.



Con este planteamiento de la estructura de un disco duro, cualquier sector de un disco puede identificarse inequívocamente por tres parámetros: el número de cilindro (*Cylinder*), el número de cabezal (*Header*, que indica la superficie, y junto con el cilindro ya podemos conocer la pista exacta) y el número de sector dentro de la pista (*Sector*). Esto es lo que se denomina direccionamiento o geometría **CHS** (*Cylinder-Header-Sector*).

Desafortunadamente, las BIOS más antiguas ponían un límite al tamaño permitido para cada uno de los valores de C, H y S. Durante la década de los 90, los tamaños de disco rápidamente superaron las limitaciones impuestas por la BIOS. Además, casi todos los discos duros (excepto los más antiguos) utilizan un número variable de sectores por cilindro, incluyendo más sectores en las pistas externas que en las internas, con lo que cabe más información en cada disco. Muchas estrategias intermedias realizaban la traducción de los valores reales de CHS a valores "virtuales" para superar estas limitaciones, ya fuera en la mismo BIOS o por medio de rutinas de software de bajo nivel. Incluso sin los límites artificiales de la BIOS, el diseño de CHS permite hasta 65536 cilindros, 16 cabezales, y 255 sectores/pista. Esto limita la capacidad a 267.386.880 sectores (aproximadamente 137 GB).

La solución consistió en hacer que el sistema ignorara la geometría y permitiera que el propio disco duro la calculara. El sistema, en lugar de pedir un valor de CHS pide una dirección **LBA** (*Logical Block Address*, dirección lógica de bloque), que es un identificador único de cada sector del disco, siendo el propio firmware del disco duro el que se encarga de localizar el cabezal, el cilindro y el sector correspondientes. De esta forma, al acceder a un sector del disco duro no se proporcionan los parámetros CHS, sino el número LBA del sector.

BIOS SETUP UTILITY	
Advanced	
Primary IDE Channel (0)	
Device :	Hard Disk
Vendor :	Virtual HD
Size :	5.2GB
LBA Mode :	Supported
Block Mode:	128Sectors
PIO Mode :	4
Async DMA :	MultiWord DMA-2
LBA/Large Mode	[Auto]
Block (Multi-Sector Transfer)	[Auto]
32Bit Data Transfer	[Enabled]
- Select Screen ↑↓ Select Item +- Change Option F1 General Help F10 Save and Exit ESC Exit	

v02.10 (C)Copyright 1985-2001, American Megatrends, Inc.

Las BIOS modernas suelen permitir elegir en cada caso el modo que se desee utilizar (bien CHS, LBA u otro modo, siendo el modo LBA la mejor opción).

Es posible obtener información detallada sobre los dispositivos de almacenamiento mediante el comando **hdparm**, especificando como parámetro el fichero de dispositivo correspondiente a la unidad:

```
# hdparm -I /dev/sda
/dev/sda:

ATA device, with non-removable media
      Model Number:      VBOX HARDDISK
      Serial Number:    VBbb923065-9ca84d36
      Firmware Revision: 1.0
Standards:
      Used: ATA/ATAPI-6 published, ANSI INCITS 361-2002
      Supported: 6 5 4
Configuration:
      Logical      max      current
      cylinders   16383   16383
      heads        16       16
      sectors/track 63       63
      --
      CHS current addressable sectors:   16514064
      LBA    user addressable sectors:   16777216
      LBA48  user addressable sectors:   16777216
      Logical/Physical Sector size:      512 bytes
      device size with M = 1024*1024:     8192 MBytes
      device size with M = 1000*1000:     8589 MBytes (8 GB)
      cache/buffer size = 256 KBytes (type=DualPortCache)
Capabilities:
      LBA, IORDY(can't be disabled)
      Queue depth: 32
      Standby timer values: spec'd by Vendor, no device specific minimum
```

```

R/W multiple sector transfer: Max = 128 Current = 128
DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 udma5 *udma6
      Cycle time: min=120ns recommended=120ns
PIO: pio0 pio1 pio2 pio3 pio4
      Cycle time: no flow control=120ns IORDY flow control=120ns
Commands/features:
Enabled Supported:
* Power Management feature set
* Write cache
* Look-ahead
* 48-bit Address feature set
* Mandatory FLUSH_CACHE
* FLUSH_CACHE_EXT
* Gen2 signaling speed (3.0Gb/s)
* Native Command Queueing (NCQ)
Checksum: correct

```

Con la opción "-I" se obtiene la información consultando directamente al dispositivo. La opción "-i" muestra la información recuperada durante la inicialización del sistema.

101.1.12. Módulos del kernel

En GNU/Linux, el hardware lo administran los drivers del kernel, algunos de los cuales se encuentran integrados (compilados) en el kernel, y otros, en su mayor parte, son módulos independientes. Estos módulos son ficheros que suelen almacenarse en el árbol de directorios `/lib/modules`, y se pueden cargar o descargar para proporcionar acceso al hardware. Normalmente, GNU/Linux carga los módulos que necesita cuando se inicia, pero puede que en ocasiones se necesite cargar módulos manualmente.

Estos módulos, cuando se cargan, se ejecutan en el espacio del kernel, y están presentes en `/lib/modules/$(uname -r)`:

```
# cd /lib/modules/$(uname -r)
# pwd
/lib/modules/2.6.32-5-686
```

El comando `uname -r` devuelve la versión del kernel del sistema operativo GNU/Linux. En el ejemplo se utiliza para hacer que la ejecución de las órdenes anteriores sean independientes de la versión de kernel de que se disponga.

La extensión de los ficheros de los módulos es "ko" (*Kernel Object*).

En el siguiente ejemplo se muestran los drivers (módulos) disponibles para tarjetas gráficas:

```
# ls /lib/modules/$(uname -r)/kernel/drivers/video/*.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/arcfb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/arkfb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/cirrusfb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/cyber2000fb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/fb_ddc.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/fb_sys_fops.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/hecubaefb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/hgafb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/macmodes.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/metronomefb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/n411.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/neofb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/output.ko
```

```
/lib/modules/2.6.32-5-686/kernel/drivers/video/pm2fb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/pm3fb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/s1d13xxfb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/s3fb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/sm501fb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/sstfb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/svgalib.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/syscopyarea.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/sysfillrect.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/sysimtblt.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/tdfxfb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/tridentfb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/uvesafb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/vfb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/vga16fb.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/vgastate.ko
/lib/modules/2.6.32-5-686/kernel/drivers/video/vt8623fb.ko
```

101.1.12.1. Conocer los módulos cargados en el kernel

Para saber los módulos que hay cargados actualmente en el sistema se dispone del comando **lsmod**, que no recibe opciones:

```
$ lsmod
Module           Size  Used by
uvcvideo          45526  0
videodev          25569  1 uvcvideo
v4l1_compat       10250  2 uvcvideo,videodev
nls_cp437          4489  1
vfat              6578   1
fat               34944  1 vfat
usb_storage        30841  2
nls_utf8            908  1
isofs              24608  0
udf               63002  0
crc_itu_t          1035   1 udf
cpufreq_userspace  1488   0
cpufreq_conservative  4018   0
cpufreq_stats       1940   0
cpufreq_powersave    602   0
vboxvideo          1073   1
drm                112096  2 vboxvideo
ppdev              4058   0
lp                  5570   0
[...]
```

La primera columna, titulada *Module* (módulo), contiene el nombre del módulo cargado. La segunda columna, titulada *Size* (tamaño), indica la cantidad de bytes que consume de memoria el módulo. La última columna, titulada *Used by* (utilizado por), describe quién utiliza el módulo.

En el ejemplo anterior, el módulo *isofs* (utilizado para acceder a sistemas de ficheros de CD-ROM) pone valor 0, indicando que no está en uso actualmente; en cambio, el módulo *vfat* (utilizado para acceder a particiones VFAT) tiene valor 1, indicando que está en uso.

Se puede observar también que si es un módulo el que está haciendo uso de otro (esto es, tiene una dependencia hacia éste), aparece su nombre listado a la derecha de la línea correspondiente al módulo; por ejemplo, el módulo *udf* hace uso del módulo *crc_itu_t*. Esta información permite saber si es posible eliminar un módulo o no, dependiendo si otros módulos dependen de éste.

El comando **lsmod** sólo da forma al contenido del fichero */proc/modules*:

```
# cat /proc/modules
uvcvideo 45526 0 - Live 0xe0ae5000
videodev 25569 1 uvcvideo, Live 0xe0ab6000
v4l1_compat 10250 2 uvcvideo,videodev, Live 0xe0a52000
nls_cp437 4489 0 - Live 0xe0a41000
vfat 6578 0 - Live 0xe09e0000
fat 34944 1 vfat, Live 0xe0a2c000
usb_storage 30841 0 - Live 0xe09e7000
nls_utf8 908 0 - Live 0xe09a7000
isofs 24608 0 - Live 0xe09ca000
udf 63002 0 - Live 0xe0991000
crc_itu_t 1035 1 udf, Live 0xe094e000
cpufreq_userspace 1488 0 - Live 0xe092e000
cpufreq_conservative 4018 0 - Live 0xe0924000
cpufreq_stats 1940 0 - Live 0xe08fd000
cpufreq_powersave 602 0 - Live 0xe08f3000
vboxvideo 1073 1 - Live 0xe082d000
[...]
```

El comando **lsmod** sólo muestra información sobre los módulos del kernel, pero no sobre los drivers que se encuentran compilados directamente en el kernel de GNU/Linux. Por este motivo, es posible que en algunos sistemas sea necesario cargar un módulo y en otros no sea necesario por encontrarse integrado en el kernel.

101.1.12.1. Obtener información sobre un módulo del kernel

Es posible conocer más información sobre los módulos con **modinfo**:

```
# modinfo vfat
filename:      /lib/modules/2.6.32-5-686/kernel/fs/fat/vfat.ko
author:        Gordon Chaffee
description:   VFAT filesystem support
license:       GPL
depends:       fat,nls_base
vermagic:     2.6.32-5-686 SMP mod_unload modversions 686
```

1.1.12.3. Cargar módulos del kernel

GNU/Linux permite cargar módulos del kernel mediante dos comandos: **insmod** y **modprobe**.

101.1.12.3.1. insmod

El comando **insmod** inserta un único módulo en el kernel, de forma que es necesario tener previamente cargados los módulos en los que se basa el que se va a cargar.

Para cargar un módulo con **insmod** hay que especificar la ruta completa de su fichero:

```
# insmod /lib/modules/2.6.32-5-686/kernel/drivers/block/floppy.ko
```

NOTA: El fichero *floppy.ko* es el driver para acceder a la disquetera.

Es posible pasarle opciones al módulo añadiéndolas al final de la orden anterior. Las opciones son específicas de cada módulo, por lo que será necesario consultar la documentación del mismo para saber cuáles pasarle. Se puede indicar desde el número de IRQ que se debe utilizar hasta la resolución de la tarjeta de vídeo.

101.1.12.3.2. modprobe

El comando **modprobe**, al contrario que **insmod**, carga automáticamente los módulos de los que depende, siendo éste el comando preferido para realizar esta tarea.

Para cargar un módulo sólo es necesario indicar su nombre, a diferencia de **insmod**:

```
# modprobe floppy
```

Al igual que **insmod**, es posible pasarle opciones al módulo añadiéndolas al final de la orden.

Es posible utilizar opciones con el comando **modprobe** (especificándolas entre el nombre del comando y el del módulo).

Opciones de modprobe	
Opción	Descripción
-v, --verbose	Mostrar información detallada sobre sus operaciones.
-C ficheroconfig	Cambiar el fichero de configuración. Por defecto, modprobe utiliza el fichero de configuración <i>/etc/modprobe.conf</i> o el directorio <i>/etc/modprobe.d</i> . Esta opción permite especificar un fichero o directorio de configuración alternativo (<i>ficheroconfig</i>).
-n, --dry-run	Realizar una simulación del proceso. Se hacen las comprobaciones y demás operaciones, excepto la inserción real del módulo. Es útil para depurar.
-r, --remove	Eliminar módulos. Invierte el efecto normal de modprobe, haciendo que el módulo especificado se elimine y todo aquél que dependa del mismo.
-f, --force	Forzar la carga del módulo aunque la versión del kernel no coincida con la del módulo. Esta acción es peligrosa, aunque en ocasiones necesaria.
--show-depends	Mostrar todos los módulos de los que depende el módulo especificado. No instala módulo, sólo informa.
-l, --list	Mostrar los módulos disponibles en el sistema (cargados o no). Es posible pasarle el nombre del módulo o parte del nombre para filtrar el resultado, así como es posible utilizar el comodín "*". Por ejemplo, para listar todos los módulos que empiezan por "c": modprobe -l c*

Por ejemplo, para listar todos módulos disponibles que comienzan por "cd":

```
# modprobe -l cd*
kernel/drivers/net/usb/cdc_ether.ko
kernel/drivers/net/usb/cdc_eem.ko
kernel/drivers/net/usb/cdc_subset.ko
kernel/drivers/net/usb/cdc-phonet.ko
kernel/drivers/cdrom/cdrom.ko
kernel/drivers/usb/class/cdc-acm.ko
kernel/drivers/usb/class/cdc-wdm.ko
```

Otro ejemplo, para listar los módulos de los que depende otro módulo:

```
# modprobe --show-depends vfat
insmod /lib/modules/2.6.32-5-686/kernel/fs/nls/nls_base.ko
insmod /lib/modules/2.6.32-5-686/kernel/fs/fat/fat.ko
```

```
insmod /lib/modules/2.6.32-5-686/kernel/fs/fat/vfat.ko
```

101.1.12.4. Eliminar módulos del kernel

En la mayoría de los casos, los módulos se pueden dejar cargados aunque no se usen; lo único que harán será consumir una cierta cantidad de memoria. En otras ocasiones, es necesario descargar un módulo antiguo para cargar uno más actualizado.

Para descargar un módulo del kernel se utiliza el comando **rmmmod**, que es el opuesto a **insmod**. El comando **rmmmod** recibe el nombre del módulo como opción, en lugar del nombre del fichero del módulo:

```
# rmmod floppy
```

En el ejemplo anterior se descargará el módulo floppy.

Opciones de rmmod	
Opción	Descripción
-v, --verbose	Mostrar información detallada sobre sus operaciones.
-f, --force	Forzar la eliminación del módulo aunque esté en uso. Esta acción es peligrosa, aunque en ocasiones es necesaria.
-w, --wait	Esperar hasta que deje de utilizarse el módulo, en lugar de devolver un error si está en uso. rmmod dará la impresión de que no está haciendo nada, esperando a que se deje de usar el módulo, momento en que lo descargará del kernel y terminará su ejecución.

Al igual que **insmod**, **rmmod** trabaja sobre un único módulo. Si se intenta descargar un módulo del que dependen otros módulos o que está en uso, **rmmod** devolverá un mensaje de error (a no ser que se utilice la opción -w). Si del módulo dependen otros módulos, se listarán, por lo que se podrá decidir si descargarlos.

Para descargar módulos con dependencias se puede utilizar el comando **modprobe** con la opción -r, como se comentó anteriormente.

101.1.12.5. Actualizar el árbol de dependencias

El comando **depmod** actualiza el árbol de dependencias entre los módulos modificando el fichero `/lib/modules/$(uname -r)/modules.dep`.

```
# cat /lib/modules/$(uname -r)/modules.dep
kernel/arch/x86/kernel/cpu/mcheck/mce-inject.ko:
kernel/arch/x86/kernel/cpu/cpufreq/powernow-k8.ko:
kernel/drivers/acpi/processor.ko kernel/drivers/thermal/thermal_sys.ko
kernel/arch/x86/kernel/cpu/cpufreq/acpi-cpufreq.ko:
kernel/drivers/acpi/processor.ko kernel/drivers/thermal/thermal_sys.ko
kernel/arch/x86/kernel/cpu/cpufreq/powernow-k6.ko:
kernel/arch/x86/kernel/cpu/cpufreq/powernow-k7.ko:
kernel/drivers/acpi/processor.ko kernel/drivers/thermal/thermal_sys.ko
kernel/arch/x86/kernel/cpu/cpufreq/longhaul.ko:
kernel/arch/x86/kernel/cpu/cpufreq/longrun.ko:
kernel/arch/x86/kernel/cpu/cpufreq/gx-suspmode.ko:
kernel/arch/x86/kernel/cpu/cpufreq/speedstep-ich.ko:
kernel/arch/x86/kernel/cpu/cpufreq/speedstep-lib.ko:
kernel/arch/x86/kernel/cpu/cpufreq/speedstep-lib.ko:
kernel/arch/x86/kernel/cpu/cpufreq/speedstep-smi.ko:
```

```

kernel/arch/x86/kernel/cpu/cpufreq/speedstep-lib.ko
kernel/arch/x86/kernel/cpu/cpufreq/speedstep-centrino.ko:
kernel/arch/x86/kernel/cpu/cpufreq/p4-clockmod.ko:
kernel/arch/x86/kernel/cpu/cpufreq/speedstep-lib.ko
kernel/arch/x86/kernel/cpu/cpufreq/cpufreq-nforce2.ko:
kernel/arch/x86/kernel/msr.ko:
kernel/arch/x86/kernel/cpuid.ko:
kernel/arch/x86/kernel/apm.ko:
kernel/arch/x86/kernel/scx200.ko:
kernel/arch/x86/kernel/microcode.ko:
kernel/arch/x86/crypto/aes-i586.ko: kernel/crypto/aes_generic.ko
[...]

```

Este fichero contiene dos columnas: la primera es la ruta del módulo y la segunda es la lista de dependencias (esto es, la lista de módulos que se deben cargar para que el primero funcione).

Si se utiliza **depmod** con la opción **-a** se vuelven a construir las dependencias de todos los módulos correspondientes a la versión del kernel actual. Esta acción se ejecuta en cada inicio del sistema, pero si se añade algún módulo manualmente será necesario volver a ejecutar esta orden:

```
# depmod -a
```

101.1.13. Otras herramientas

101.1.13.1. hwinfo

La herramienta **hwinfo** detecta el hardware y proporciona un listado con información sobre el mismo (de manera corta con la opción "--short"). Consulta directamente los dispositivos, proporcionando información fiable.

```

# hwinfo --short
cpu:
cpu: Intel(R) Core(TM) i5 CPU M 460 @ 2.53GHz, 2511 MHz
keyboard: /dev/input/event0 AT Translated Set 2 keyboard
mouse:
mouse: /dev/input/mice VirtualBox USB Tablet
mouse: /dev/input/mice VirtualBox Mouse
graphics card:
graphics card: InnoTek Systemberatung VirtualBox Graphics Adapter
sound:
sound: Intel 82801AA AC'97 Audio Controller
storage:
storage: Intel 82371AB/EB/MB PIIX4 IDE
storage: Intel 82801HBM/HEM (ICH8M/ICH8M-E) SATA AHCI Controller
network:
network: eth0 Intel PRO/1000 MT Desktop Adapter
network interface:
network interface: lo Loopback network interface
network interface: eth0 Ethernet network interface
network interface: pan0 Ethernet network interface
disk:
disk: /dev/sda VBOX HARDDISK
partition:
partition: /dev/sda1 Partition
partition: /dev/sda2 Partition
partition: /dev/sda5 Partition
cdrom:

```

```

/dev/sr0           VBOX CD-ROM
usb controller:   Apple KeyLargo/Intrepid USB
                   Intel 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI
Controller       BIOS
bios:
bridge:          BIOS
                  Intel 82371AB/EB/MB PIIX4 ACPI
                  Intel 82801 Mobile PCI Bridge
                  Intel 82801 Mobile PCI Bridge
                  Intel 82801GBM (ICH7-M) LPC Interface Bridge
hub:
memory:
unknown:

```

Para obtener más detalles se puede obviar la opción "--short" y especificar el componente del que se quiere obtener información detallada. Por ejemplo, "--cpu" para el procesador, "--memory" para la memoria, etc. (consultar el manual del comando).

```

# hwinfo --cpu
01: None 00.0: 10103 CPU
[Created at cpu.304]
Unique ID: rdCR.j8NaKXDZtZ6
Hardware Class: cpu
Arch: Intel
Vendor: "GenuineIntel"
Model: 6.37.5 "Intel(R) Core(TM) i5 CPU      M 460 @ 2.53GHz"
Features:
fpu,vme,de,pse,tsc,msr,pae,mce,cx8,apic,sep,mtrr,pge,mca,cmov,pat,pse36,clflush,
mmx,fxsr,sse,sse2,syscall,nx,lm,constant_tsc,up,pni,monitor,ssse3,lahf_lm
Clock: 2511 MHz
BogoMips: 5022.93
Cache: 6144 kb
Config Status: cfg=new, avail=yes, need=no, active=unknown

```

101.1.13.2. dmidecode

La herramienta **dmidecode** no consulta a los periféricos directamente, sino que lee e interpreta la tabla **DMI** (*Desktop Management Interface*) del ordenador, a veces denominada **SMBIOS** (*System Management BIOS*). Proporciona información sobre el estado físico actual de la máquina y sobre las características de sus componentes.

A diferencia de **hwinfo** que consulta directamente a los componentes, **dmidecode** lee la información según la detecta la BIOS y la placa base. Es rápido, a veces más concreto que **hwinfo**, aunque en ocasiones da información errónea o imprecisa.

```
# dmidecode # dmidecode 2.9
SMBIOS 2.5 present.
5 structures occupying 352 bytes.
Table at 0x000E1000.

Handle 0x0000, DMI type 0, 20 bytes
BIOS Information
    Vendor: innotek GmbH
    Version: VirtualBox
    Release Date: 12/01/2006
    Address: 0xE0000
    Runtime Size: 128 kB
    ROM Size: 128 kB
    Characteristics:
        ISA is supported
        PCI is supported
        Boot from CD is supported
        Selectable boot is supported
        8042 keyboard services are supported (int 9h)
        CGA/mono video services are supported (int 10h)
        ACPI is supported

Handle 0x0001, DMI type 1, 27 bytes
System Information
    Manufacturer: innotek GmbH
    Product Name: VirtualBox
    Version: 1.2
    Serial Number: 0
    UUID: 2A57F6B1-60FB-4C9F-B51A-681974B27FC9
    Wake-up Type: Power Switch
    SKU Number: Not Specified
    Family: Virtual Machine

Handle 0x0003, DMI type 126, 13 bytes
Inactive

Handle 0x0002, DMI type 126, 7 bytes
Inactive

Handle 0xFFFF, DMI type 127, 147 bytes
End Of Table
```

Es posible solicitar información concreta con las opciones "-s" y "-t" (consultar el manual).

101.1 - EXTRAS

101.1.1 Reconocimiento de discos scsi

En un sistema de producción un tiempo de parada para agregar un dispositivo de disco es un lujo no permitido, por ello se debe de poder realizar un escaneo de dispositivos nuevos que tengan capacidad de “pinchar en caliente”. Los dispositivos SCSI tiene esta característica y es una tecnología ampliamente utilizada en servidores por ello se va a analizar el procedimiento de reconocimiento en función de su identificación.

```
echo " - - - " > /sys/class/scsi_host/hostx/scan  
fdisk -l  
tail -f /var/log/message
```

Si por el contrario queremos eliminar el dispositivo del árbol, se ejecutarían las siguientes instrucciones:

```
# echo 1 > /sys/block/devName/device/delete  
# echo 1 > /sys/block/sdc/device/delete
```

Para el reescaneo de hardware completo:

```
#find /sys/ -name rescan -exec sh -c 'echo 1 1 1 > {}' \;
```

Mediante el **/proc/partitions** se pueden ver las particiones de los discos:

```
# cat /proc/partitions  
major minor #blocks name  
  
 8      0    8388608 sda  
 8      1    8385898 sda1
```

Y mediante el **/proc/scsi/scsi** se puede listar los discos disponibles. En este caso se trata de una **máquina virtual con VMWare**:

```
# cat /proc/scsi/scsi  
Attached devices:  
Host: scsi0 Channel: 00 Id: 00 Lun: 00  
  Vendor: VMWare, Model: VMware Virtual S Rev: 1.0  
  Type: Direct-Access                      ANSI SCSI revision: 02
```

Para pedir al sistema que haga el rescan del bus deberemos hacer el siguiente echo al fichero **/sys/class/scsi_host/host0/scan**

```
echo - - - > /sys/class/scsi_host/host0/scan
```

En caso de disponer de más de un **dispositivo SCSI** se debe seleccionar el host que corresponda.

En el **dmesg** se puede observar el nuevo disco añadido:

```
# dmesg  
ide: failed opcode was: 0xec  
      Vendor: VMWare, Model: VMware Virtual S Rev: 1.0  
      Type: Direct-Access                      ANSI SCSI revision: 02  
target0:0:1: Beginning Domain Validation  
target0:0:1: Domain Validation skipping write tests  
target0:0:1: Ending Domain Validation  
target0:0:1: FAST-40 WIDE SCSI 80.0 MB/s ST (25 ns, offset 127)  
SCSI device sdb: 2097152 512-byte hdwr sectors (1074 MB)  
sdb: Write Protect is off  
sdb: Mode Sense: 5d 00 00 00  
sdb: cache data unavailable  
sdb: assuming drive cache: write through  
SCSI device sdb: 2097152 512-byte hdwr sectors (1074 MB)  
sdb: Write Protect is off  
sdb: Mode Sense: 5d 00 00 00  
sdb: cache data unavailable  
sdb: assuming drive cache: write through  
  sdb: unknown partition table  
sd 0:0:1:0: Attached scsi disk sdb
```

```
sd 0:0:1:0: Attached scsi generic sg1 type 0
```

Repetiendo los **cat** en el **proc** es posible comprobar la **nueva LUN** añadida:

```
# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: VMware, Model: VMware Virtual S Rev: 1.0
  Type: Direct-Access           ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 01 Lun: 00
  Vendor: VMware, Model: VMware Virtual S Rev: 1.0
  Type: Direct-Access           ANSI SCSI revision: 02
```

De la misma manera en el **/proc/partitions**

```
# cat /proc/partitions
major minor #blocks name

 8      0    8388608 sda
 8      1    8385898 sda1
 8     16   1048576 sdb
```

101.1.2 iSCSI (*target e iniciador*)

iSCSI (Abreviatura de Internet SCSI) es un estándar que permite el uso del protocolo SCSI sobre redes TCP/IP. iSCSI es un protocolo de la capa de transporte definido en las especificaciones SCSI-3. Otros protocolos en la capa de transporte son SCSI Parallel Interface y canal de fibra.

La adopción del iSCSI en entornos de producción corporativos se ha acelerado en estos momentos gracias al aumento del Gigabit Ethernet. La fabricación de almacenamientos basados en iSCSI (red de área de almacenamiento) es menos costosa y está resultando una alternativa a las soluciones SAN basadas en Canal de fibra.

Aunque fuera de las implementaciones domésticas es una solución muy utilizada de forma empresarial. Las soluciones son múltiples y permite acceso a almacenamiento de alta velocidad y centralizado. En este extra del LPIC se tratará de analizar los procedimientos para instalar un servidor (*target*) y su conexión desde un cliente (*initiator*). La plataforma de este baoratorio es Debian 7.2

Servidor: creación del target

Este equipo tiene un disco SCSI agregado que se compartirá con los clientes

```
# cat /proc/partitions
major minor #blocks name

 11      0    227328 sr0
  8      0   20971520 sda
  8      1   20067328 sda1
  8      2        1 sda2
  8      5   901120 sda5
  8     16   1048576 sdb
  8     17   1048376 sdb1
```

```
8      32    1048576 sdc
8      33    1047552 sdc1
```

Se instalan los paquetes necesarios para el servidor y también para el cliente

```
# apt-get install iscsitarget open-iscsi module-assistant
```

Tras la instalación, se debe editar el archivo /etc/default/iscsitarget cuyo contenido debe quedar aproximadamente así:

```
# cat /etc/default/iscsitarget
ISCSITARGET_ENABLE=true
# ietd options
# See ietd(8) for details
ISCSITARGET_OPTIONS=""
```

Para agregar los target que se van a compartir y presentar en la red se edita el fichero /etc/iet/ietd.conf que está muy bien documentado. Se edita para definir el nombre del target, la ruta del disco o volumen y un alias. Es posible definir muchas más opciones que no se verán de forma exhaustiva. En el ejemplo el servidor tiene la IP 172.16.1.150, se puede usar el nombre del host. El puerto de escucha predeterminado es 3260.

```
# tail -3 /etc/iet/ietd.conf
Target iqn.2013-12.deb.local:172.16.1.150
Lun 0 Path=/dev/sdc1
Alias Test
```

Reiniciar el servidor

```
# /etc/init.d/iscsitarget restart
[ ok ] Removing iSCSI enterprise target modules: :.
[ ok ] Starting iSCSI enterprise target service:..
. ok
```

Cliente: configurar el iniciador

El host que desea acceder al target de ISCSI debe tener instalado el paquete open-iscsi

```
# apt-get install open-iscsi
```

posteriormente se debe modificar el proceso de conexión como automático o manual.

```
# grep 'node.startup' /etc/iscsi/iscsid.conf
# node.startup = automatic
node.startup = manual
```

Reiniciar el demonio si se realiza cualquier cambio

```
# /etc/init.d/open-iscsi restart
```

Se usa la herramienta iscsadm para localizar los targets de un servidor y posteriormente mapearlo como un disco

```
# iscsadm -m discovery -t sendtargets -p 172.16.1.150
172.16.1.150:3260,1 iqn.2013-12.deb.local:172.16.1.150
# cat /proc/partitions
major minor #blocks name

      8        0   10485760  sda
      8        1   10005504  sda1
      8        2         1  sda2
      8        5    477184  sda5
     11        0   1048575  sr0

# iscsadm -m node --targetname iqn.2013-12.deb.local:172.16.1.150 -p
172.16.1.150 --login
Logging in to [iface: default, target: iqn.2013-12.deb.local:172.16.1.150,
portal: 172.16.1.150,3260] (multiple)
Login to [iface: default, target: iqn.2013-12.deb.local:172.16.1.150, portal:
172.16.1.150,3260] successful.

# cat /proc/partitions
major minor #blocks name

      8        0   10485760  sda
      8        1   10005504  sda1
      8        2         1  sda2
      8        5    477184  sda5
     11        0   1048575  sr0
      8       16   1047552  sdb
```

Se ha podido verificar el mapeo del disco mediante ISCSI. Posteriormente el disco se trata como si fuera local usando herramientas para particionar, asignar sistema de ficheros y montaje.

101.2. Inicio del sistema.

Peso en el examen de certificación: 3 puntos.

Objetivo: Conocer y controlar el proceso de arranque del sistema.

Conceptos y áreas de conocimiento:

- .Utilizar comandos comunes para el gestor de arranque y trasladar al kernel opciones de inicialización.
- .Conocer la secuencia de ejecución de la BIOS hasta el completo inicio del sistema.
- .Consultar los eventos de inicio en los archivos de registro.

Términos y utilidades

/var/log/messages

BIOS

kernel

dmesg

bootloader

init

101.2.1. Introducción

El proceso de arranque de un sistema X86 y x86-64 puede variar mucho en función de la configuración del sistema y debido en parte a que existen muchas opciones posibles. Pero hay algo común a todas las vías posibles: el proceso de arranque comienza con el BIOS.

La mayoría de los BIOS permiten seleccionar el orden de los dispositivos desde los que poder arrancar, pasando a la segunda entrada si falla la primera, a la tercera si falla la segunda y así sucesivamente. El objetivo es arrancar desde un dispositivo que contenga un MBR (Master Boot Record, Sector de arranque) con la tabla de partición y un cargador de arranque (también denominado gestor de arranque).

El orden habitual de los dispositivos es establecer primero el disquete (identificado como A: en los sistemas operativos MS-DOS y Windows), seguido de la unidad de CD-ROM y/o DVD, según sea el caso y, por último, el primer disco duro. Con este tipo de configuración, el método BOOT del BIOS intentará arrancar desde cada dispositivo, uno tras otro, hasta que uno funcione. Si ninguno de los dispositivos logra arrancar, la BIOS mostrará un mensaje de error.

La etapa del cargador de arranque no es totalmente necesaria. Determinados BIOS pueden cargar y pasar el control a Linux sin hacer uso del cargador. Cada proceso de arranque será diferente dependiendo de la arquitectura del procesador y el BIOS. Desafortunadamente las combinaciones posibles son muy amplias y en constante desarrollo, siendo necesario su estudio particular, ya que, a veces, hay que intervenir en el proceso de inicio Linux de un modo u otro.

Los cargadores de arranque de Linux no sólo permiten iniciar otros sistemas operativos, sino que también permiten configurar su inicio con opciones particulares. Es por ello que se hace necesaria la comprensión y el uso avanzado de todas las opciones, en la referente al inicio del SO, permitiendo una configuración al gusto y que cumpla los objetivos concretos de cada caso.

101.2.2. Ajustes iniciales

Cada uno de los BIOS es diferente en función del diseño de los fabricantes de placas madres (main board) y sus correspondientes interfaces (AMIBIOS, Phoenix, Awrd, etc..) Sin embargo, un gran número de ajustes son semejantes o, en todo caso, sólo hay que saber localizar el menú exacto donde personalizar el ajuste. Estos ajustes se resumen básicamente en:

- **Detección de los discos duros y elección del soporte de inicio:** Linux soporta interfaces de discos IDE, SATA y SCSI. Por ello, no se debería necesitar el disco del controlador para el soporte de almacenamiento concreto, ya que el CD-ROM de instalación de una distribución concreta debería traerlo. Pero en algunos casos, puede no reconocerse algunos tipos de discos cuando se tiene la necesidad de instalar Linux en un soporte concreto. Los discos de controladores son utilizados normalmente por unidades de CD-ROM muy nuevas o que no son estándar, adaptadores SCSI, discos SATA o tarjetas de red. Estos son los únicos dispositivos usados durante la instalación que pueden requerir controladores no incluidos en los CD-ROMs de Red Hat Linux (o disquetes de arranque). En el caso que no se reconozca el disco SATA, la mayoría de los BIOS tienen como opción emular un IDE modificando el controlador de la SATA. Linux los reconoce como tales (es lo que se llama modo nativo). No obstante, es recomendable probar una primera instalación con el soporte de los discos SATA activo.
Del mismo modo que Windows Vista y versiones más recientes de Windows, OpenBSD (versión 4.1 en adelante), NetBSD , FreeBSD , OS X y Solaris 10 (8 / 07 y siguientes), los sistemas Linux (basados en el kernel 2.6.19 en adelante), en principio, gestionan correctamente el soporte de los chipsets SATA compatibles con AHCI (Advanced Host Controller Interface). Se recomienda activar esta opción en la BIOS siempre que se pueda ya que se obtiene una máxima velocidad en el SATA. Además, el modo AHCI soporta la conexión en caliente y el modo NCQ (Native Command Queuing) de los discos SATA II, que permite aprovechar las ventajas del protocolo y ganar algo de velocidad. Sólo si no funciona, se puede intentar un modo combined, y en caso extremo el modo legacy IDE. En lo que respecta a la elección del soporte, desde el BIOS es posible modificar el orden de ejecución de la manera en que se arranca (desde el lector de CD, de DVD, etc.)
- **Detección del soporte para el teclado:** Si el teclado es de tipo USB, o sin cable pero con un adaptador sin cable USB, se debe activar el USB legacy support (a veces llamada función USB DOS function o USB keyboard enable). Activar esta opción es sumamente importante, ya que permite activar en el momento del inicio, además del teclado, los soportes de almacenamiento (discos duros y tarjetas de memoria). Su activación no impide que el sistema (una vez iniciado el sistema operativo, cargados los driver USB del núcleo y de los módulos) se encargue del USB.

Cada proceso de arranque será diferente dependiendo de la arquitectura del procesador y el BIOS. Desafortunadamente las combinaciones posibles son muy amplias y en constante desarrollo, siendo necesario su estudio particular, ya que, a veces, hay que intervenir en el proceso de inicio Linux de un modo u otro. Por ello, como norma general, no se recomienda realizar cualquier otra modificación en la configuración. Sin embargo, con el fin de ahorrar recursos, se puede desactivar los puertos de la placa que no se utilicen: puerto paralelo, puerto serie, etc.

Overclocking: Overclock es un anglicismo de uso habitual en informática que literalmente significa "sobre el reloj", es decir, es el aumento de la frecuencia de reloj de la CPU por encima de las especificaciones del fabricante. La idea es conseguir gratuitamente un rendimiento más alto, o superar las cotas actuales de rendimiento, aunque esto pueda suponer una pérdida de estabilidad o acortar la vida útil del componente. De utilizarse, es evidente que requerirá de componentes hardware de calidad ya que, en particular, pone a prueba la memoria. Es la principal causa de

inestabilidad y cuelgues, tanto en sistemas Windows como en Linux.

101.2.3. Los cargadores de arranque

En la práctica, en Linux existen dos cargadores de arranque (*boot loaders*) importantes: **LILO** (LInux LOader), el cargador de Linux y **GRUB** (GRand Unified Bootloader, Cargador de arranque unificado); donde LILO es el más antiguo y está siendo reemplazado lentamente por GRUB.

Aunque existen otras soluciones, incluso las presentes en otras plataformas, todas tienen la misma finalidad (y puede que incluso se llamen igual), pero no son totalmente idénticas. Tanto LILO como GRUB son cargadores del arranque de la arquitectura x86 y para su extensión del conjunto de instrucciones x86 que maneja direcciones de 64 bits (la arquitectura x86-64). Por ello, en caso de que requiera configurar un cargador de arranque que no pertenezca a x86, se debe consultar la documentación específica de la plataforma.

Siempre se puede consultar información general y técnica para un importante número de gestores de arranque en la dirección URL: http://en.wikipedia.org/wiki/Comparison_of_boot_loaders.

Además, se puede consultar la URL: <http://es.wikipedia.org/wiki/Syslinux>.

101.2.3.1. LILO

El gestor de arranque **LILO** (Linux Loader) permite elegir entre qué sistema operativo se ha de iniciar un equipo con más de un sistema operativo disponible. Fue desarrollado inicialmente por Werner Almesberger, actualmente está a cargo de John Coffman.

LILO fue durante un tiempo el cargador del arranque para la arquitectura de x86. Aunque desde entonces GRUB le ha estado restando popularidad, sigue siendo un pequeño y útil cargador que funciona en una variedad de sistemas de archivos. LILO puede arrancar un sistema operativo desde el disco duro o desde un disco flexible externo. LILO permite seleccionar entre 16 imágenes en el arranque. LILO puede instalarse también en el master boot record (MBR).

LILO es un cargador de arranque casi idéntico a GRUB en su proceso, excepto que no contiene una interfaz de línea de comandos. Por lo tanto, todos los cambios en su configuración deben ser escritos en el MBR y luego se debe reiniciar el sistema. Así, un error en la configuración, o la incorporación de un nuevo kernel, puede dejar el disco inservible para el proceso de arranque hasta tal grado, que sea necesario usar otro dispositivo (disquete, etc.) que contenga un programa capaz de solventar el error. Además, LILO no concibe los sistemas de archivos. En su lugar, la ubicación de los archivos de imagen se almacenan directamente en el MBR y el BIOS se utiliza para acceder a ellos directamente.

101.2.3.1.1. Configuración de LILO

El uso del cargador LILO para realizar su configuración e instalación en el sector de arranque del sistema. Una vez instalado, se puede reiniciar el ordenador e indicarle a LILO qué sistema operativo o kernel se desea activar.

El archivo de configuración LILO es normalmente */etc/lilo.conf*. Este archivo se suele dividir en dos secciones principales muy diferenciadas: las opciones globales y las de cada imagen (éste último tipo de secciones son conocidas como *stanzas* o *estrofas*).

101.2.3.2. GRUB

GNU GRUB (GNU GRand Unified Bootloader) es un gestor de arranque múltiple, desarrollado por el proyecto GNU que se usa comúnmente para iniciar uno de dos o más sistemas operativos instalados en un mismo equipo.

En la mayoría de las distribuciones Linux, es el cargador por defecto, que se puede configurar a medida, en particular la protección mediante contraseña encriptada.

GRUB fue inicialmente diseñado e implementado por el programador Erich Stefan Boleyn, como parte del sistema operativo GNU Hurd desarrollado por la Free Software Foundation. En 1999, Gordon Matzigkeit y Yoshinori Okuji convirtieron a GRUB en un paquete de software oficial del Proyecto GNU y abrieron el desarrollo del mismo al público.

Mientras los gestores de arranque convencionales tienen una tabla de bloques en el disco duro, GRUB es capaz de examinar el sistema de archivos. Actualmente, soporta numerosos sistemas de archivos entre los que están: ext2/ext3/ext4 (Grub2) usado por los sistemas UNIX (incluyendo GNU/Linux), FAT16 y FAT32 usados por Windows 9.x, NTFS usado por los sistemas Windows NT (a partir de Windows NT v.3.51), ZFS de Opensolaris/Solaris, etc..

101.2.3.2.1. Configuración de GRUB

GRUB cuenta con un intérprete de comandos y con una interfaz gráfica, guardando las configuraciones en un archivo de texto, no siendo necesaria su reinstalación tras cada modificación. GRUB puede leer su archivo de configuración en tiempo de arranque. Al igual que LILO, el archivo de configuración de GRUB se divide en secciones globales y de imagen, en las que cada una posee sus propias opciones.

La ubicación normal para el archivo de configuración de GRUB es */boot/grub/menu.lst*, aunque algunas distribuciones como Fedora, Red Hat y Gentoo utilizarán el nombre de fichero grub.conf en lugar de menu.lst.

Hay que tener en cuenta que el archivo de configuración tiene que estar en la partición */boot* que es la única inicialmente accesible por GRUB y dónde espera encontrarlo. Además, hay que tener presente que al arrancar el sistema se estará trabajando con un teclado norteamericano. Por ello, se debe prestar atención a los caracteres especiales intercambiados.

101.2.3.2.2. Ejemplo de configuración de GRUB

El siguiente ejemplo muestra una configuración estándar en la que la primera partición del disco duro es */boot* y la segunda contiene una instalación de Windows.

```
# /boot/grub/grub.conf
#
# Opciones globales
#
default=0
timeout=10
splashimage=(hd0,1)/grub/bootimage.xpm.gz
#
# Opciones de las imágenes Kernel
#
title Debian (2.6.25)
    root (hd0,0)
    kernel /vmlinuz-2.6.25 ro root=/dev/hda3
    initrd /initrd-2.6.25.img
#
```

```
# Otros sistemas operativos
#
title Windows 7
    rootnoverify (hd0,1)
    chainloader +1
```

Como se puede observar, la configuración global de GRUB precede a las configuraciones de cada imagen. A diferencia que en una configuración global de LILO, existen pocas opciones en este tipo de sección global.

Las opciones de las imágenes kernel de GRUB suelen presentar sangrías tras la primera línea, tal como se hace con LILO, que es una convención y no un requisito del formato del fichero. Las stanzas comienzan por una identificación y continúan con opciones que le indican a GRUB cómo gestionar la imagen.

Las líneas que comienzan con el carácter # son líneas de comentarios.

101.2.3.2.3. La configuración global de GRUB

La configuración global de GRUB precede a las configuraciones de cada imagen. A diferencia de lo que ocurre en una configuración global de LILO, existen pocas opciones en este tipo de sección global. El significado de las tres líneas es el siguiente:

- **default=0** : Indica que el sistema predeterminado que se carga es el primero. GRUB indexa desde cero. Si se desea iniciar el segundo sistema operativo listado, se debe emplear default=1 y así sucesivamente para todos los sistemas operativos listados.
- **timeout=10** : Deja un margen de 10 segundos para seleccionar un sistema antes de cargar el predeterminado. Es necesario recalcar que GRUB mide este período en segundos, mientras que LILO lo hace en décimas de segundo.
- **splashimage** : Indica la imagen que se muestra en la pantalla de inicio. La línea es opcional; sin embargo, la mayoría de las distribuciones de Linux apuntan a una imagen que decora el menú de arranque.

101.2.3.2.4. Los kernel en la configuración GRUB

Las opciones de las imágenes kernel de GRUB suelen presentar sangrías tras la primera línea, tal como se hace con LILO, que es una convención y no un requisito del formato del fichero. Las stanzas comienzan por una identificación y continúan con opciones que le indican a GRUB cómo gestionar la imagen. El significado de esas líneas es:

- **title** : Indica el texto que aparece en el menú de pantalla. Además, inicia la stanza de cada imagen y especifica la etiqueta que se mostrará cuando se cargue el cargador del arranque. A diferencia de la opción label de LILO, el title de GRUB admite espacios. De esta forma el title de una stanza puede ser muy descriptivo.
- **root** : Indica qué partición contiene el núcleo linux y puede ser o no la partición raíz del sistema (/). Así root (hd0,0) indica a GRUB que el núcleo se encuentra en la primera partición del primer disco duro (/dev/hda). Sin embargo, GRUB puede residir en una partición FAT, en un disquete o en las particiones de otros sistemas operativos, por lo que no sería de extrañar que la raíz de GRUB se encontrase en algún otro sitio más singular.
- **kernel** : Se utiliza para indicar las características de carga del núcleo. Incluye el fichero del núcleo, la partición raíz del sistema y en general los parámetros que se desean pasar al núcleo. A diferencia de LILO, que separa las opciones del kernel en líneas diferentes, en el caso de GRUB las especificaciones del kernel se incluyen en una misma línea. Por ejemplo: kernel /vmlinuz-2.6.25 ro root=/dev/hda3. La opción ro le indica al kernel que monte

initialmente sus sistema de ficheros raíz en modo sólo lectura y la opción *root*= especifica el sistema de ficheros raíz Linux. Cómo estas opciones se le pasan al kernel, es preciso utilizar los indicadores de dispositivo al estilo de Linux y no al estilo de las propias opciones del fichero de configuración de GRUB.

- **initrd** : Indica dónde se localiza la imagen del disco de memoria, de manera similar a opción de LILo con el mismo nombre.
- **map** : La opción *map*, no presente en el ejemplo, permite modificar la asignación de particiones que ha detectado la BIOS.
- **rootnoverify** : Indica a GRUB que arranque la partición de Windows pero sin intentar montarla. Es decir, es similar a la opción *root*, excepto que en GRUB no se intentará acceder a los archivos de esta partición. La opción *rootnoverify* se utiliza para especificar una partición raíz de sistemas operativos en los que GRUB no puede cargar directamente un kernel, como por ejemplo el DOS y Windows.
- **chainloader +1** : Indica a GRUB que encadene el cargador propio de otro sistema operativo. Normalmente se le pasa la opción +1 para cargar el primer sector de la partición raíz (que se suele especificar con *rootnoverify*) y ceder la ejecución a este cargador del arranque secundario.

Finalmente no está de más indicar que existe una lista bastante exhaustiva de estas opciones en la dirección URL: <http://www.gnu.org/software/grub/manual/legacy/Commands.html#Commands>

101.2.3.2.5. Los dispositivos de disco en GRUB

El cargador de arranque GRUB no hace referencia a los dispositivos de disco por su nombre como lo hace Linux. GRUB numera los dispositivos de manera que, en lugar de */dev/hda*, utiliza (*hd0*). De manera análoga, para */dev/hdb* probablemente utilice (*hd1*). El orden viene determinado por la secuencia en que la BIOS detectó los dispositivos en el momento de la instalación y que es el orden de arranque definido en la BIOS por el administrador del sistema. Las cuatro particiones primarias posibles ocupan los números de particiones 0 a 3. Las particiones lógicas se designan con los números a partir de 4:

- (*hd0,0*) primera partición primaria en el primer disco duro
- (*hd0,1*) segunda partición primaria
- (*hd0,2*) tercera partición primaria
- (*hd0,3*) cuarta partición primaria (y normalmente partición extendida)
- (*hd0,4*) primera partición lógica
- (*hd0,5*) segunda partición lógica
- ...

GRUB no distingue entre dispositivos PATA, SATA, SCSI o RAID, por lo que en un sistema que sólo tiene SCSI, el primer dispositivo SCSI será (*hd0*). En un sistema mixto, los dispositivos ATA, normalmente, reciben los números más bajos, aunque esto no es siempre el caso.

El fichero */boot/grub/device.map* almacena las asociaciones de dispositivos de GRUB.

Por otra parte, el cargador de arranque GRUB numera las particiones de un dispositivo comenzando por el 0 en lugar del 1 que utiliza Linux. GRUB separa los números de partición de los números de dispositivos con una comas. Por ejemplo, (*hd0,0*) es la primera partición lógica del primer disco (que en Linux es normalmente */dev/hd5* o */dev/sda5*. Por el contrario, a las disqueteras se les suele hacer referencia mediante (*fd0*) o, posiblemente, (*fd1*) o superior si se posee más de una disquetera. A las disqueteras no se les hace particiones, razón por la que no reciben números de partición.

101.2.3.2.6. El archivo device.map

El archivo **device.map** contiene la correspondencia entre los nombres de dispositivo GRUB y los nombres de dispositivo Linux. Cuando se dispone de un sistema mixto con discos duros PATA, SATA, SCSI o RAID, el cargador de arranque GRUB debe intentar averiguar el orden de arranque a partir de un procedimiento concreto. En este caso, GRUB no tiene acceso a la información de la BIOS sobre el orden de arranque. GRUB guarda el resultado de esta comprobación en el archivo */boot/grub/device.map*. Por ejemplo, sea el caso en que el orden de arranque definido en la BIOS prioriza los dispositivo PATA antes que los SCSI:

- (fd0) /dev/fd0
- (hd0) /dev/hda
- (hd1) /dev/hdb
- (hd2) /dev/sda
- (hd3) /dev/sdb

Conocido es que el orden de PATA, SATA, SCSI y otros discos duros depende de diversos factores y que Linux no es capaz de detectar dicha correspondencia; luego existe la posibilidad de determinar el orden manualmente en el archivo *device.map*. Si al arrancar el sistema se producen problemas, se puede comprobar si el orden de arranque en el archivo coincide con el orden especificado en la BIOS y modificarlo si es necesario con ayuda de la shell GRUB. Una vez que el sistema Linux haya arrancado, se puede modificar el archivo *device.map* de forma permanente mediante el módulo del cargador de arranque o cualquier otro editor. Tras modificar el archivo *device.map* manualmente, se debe ejecutar el siguiente comando para reinstalar GRUB:

```
# grub --batch --device-map=/boot/grub/device.map < /etc/grub.conf
```

En la actualidad se tiende a que GRUB utilice UUID's o etiquetas del sistema de archivos al generar *grub.cfg*. De esta forma se hace mucho más fácil el reconocimiento de dispositivos. Si el archivo *device.map* no existe, entonces las utilidades GRUB asumirán un mapa temporal del dispositivo, que a menudo es suficiente, particularmente en sistemas con un solo disco.

101.2.3.2.7. El cargador GRUB vs. la shell GRUB

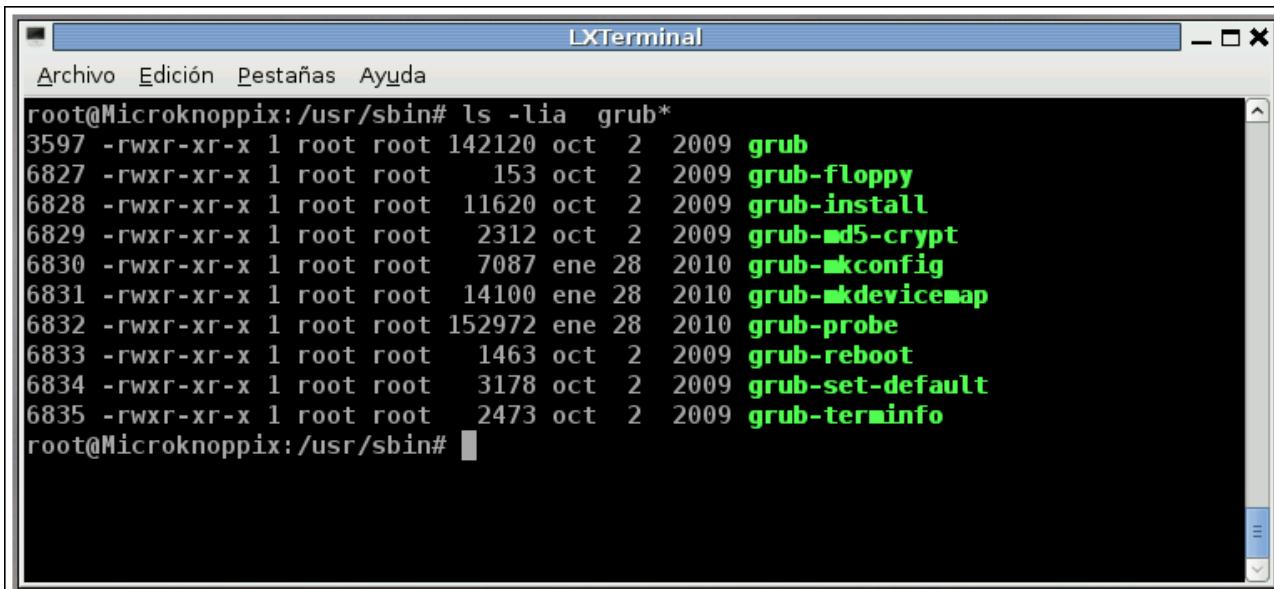
La shell GRUB es un programa más de Linux, ubicado en */usr/sbin/grub*. Este programa se denomina shell GRUB. La funcionalidad de instalar GRUB como cargador de arranque en un disco duro o disquete está directamente integrada en grub en forma del comando *install* o *setup*. De este modo, esta función está disponible en la shell GRUB cuando Linux se está ejecutando.

The screenshot shows an LXTerminal window with the title bar "LXTerminal". The menu bar includes "Archivo", "Edición", "Pestañas", and "Ayuda". The main window displays the GRUB command-line interface. It starts with "GNU GRUB version 0.97 (640K lower / 3072K upper memory)". A note follows: "[Minimal BASH-like line editing is supported. For the first word, TAB lists possible command completions. Anywhere else TAB lists the possible completions of a device/filename.]". The "grub> help" command is run, listing various GRUB commands. The commands are grouped into two columns:

blocklist FILE	boot
cat FILE	chainloader [--force] FILE
clear	color NORMAL [HIGHLIGHT]
configfile FILE	device DRIVE DEVICE
displayapm	displaymem
find FILENAME	geometry DRIVE [CYLINDER HEAD SECTOR [
halt [--no-apm]	help [--all] [PATTERN ...]
hide PARTITION	initrd FILE [ARG ...]
kernel [--no-mem-option] [--type=TYPE]	makeactive
map TO_DRIVE FROM_DRIVE	md5crypt
module FILE [ARG ...]	modulenounzip FILE [ARG ...]
pager [FLAG]	partnew PART TYPE START LEN
parttype PART TYPE	quit
reboot	root [DEVICE [HDBIAS]]
rootnoverify [DEVICE [HDBIAS]]	serial [--unit=UNIT] [--port=PORT] [--
setkey [T0_KEY FROM_KEY]	setup [--prefix=DIR] [--stage2=STAGE2_
terminal [--dumb] [--no-echo] [--no-ed]	terminfo [--name=NAME --cursor-address
testvbe MODE	unhide PARTITION
uppermem KBYTES	vbeprobe [MODE]

At the bottom, "grub>" is visible.

No obstante, estos comandos también están disponibles durante el proceso de arranque sin necesidad de que Linux se esté ejecutando, lo que simplifica en gran medida la recuperación de un sistema defectuoso. Por ello se habla de dos variantes de GRUB: una como cargador de arranque (GRUB) y otra como un programa más de Linux en `/usr/sbin/grub`.



The screenshot shows a terminal window titled "LXTerminal". The menu bar includes "Archivo", "Edición", "Pestañas", and "Ayuda". The command "ls -lia grub*" is run, displaying a list of files in the /usr/sbin directory:

```
root@Microknoppix:/usr/sbin# ls -lia grub*
3597 -rwxr-xr-x 1 root root 142120 oct  2  2009 grub
6827 -rwxr-xr-x 1 root root    153 oct  2  2009 grub-floppy
6828 -rwxr-xr-x 1 root root  11620 oct  2  2009 grub-install
6829 -rwxr-xr-x 1 root root   2312 oct  2  2009 grub-md5-crypt
6830 -rwxr-xr-x 1 root root   7087 ene 28 2010 grub-mkconfig
6831 -rwxr-xr-x 1 root root 14100 ene 28 2010 grub-mkdevicemap
6832 -rwxr-xr-x 1 root root 152972 ene 28 2010 grub-probe
6833 -rwxr-xr-x 1 root root   1463 oct  2  2009 grub-reboot
6834 -rwxr-xr-x 1 root root   3178 oct  2  2009 grub-set-default
6835 -rwxr-xr-x 1 root root   2473 oct  2  2009 grub-terminal
root@Microknoppix:/usr/sbin#
```

La instalación de GRUB es un poco diferente a la de LILO. El comando para instalar GRUB es **grub-install**, al que se le debe especificar el sector de arranque por su nombre de dispositivo en el momento de la instalación. Los comandos básicos tienen el aspecto:

```
# grub-install /dev/hda
```

o bien

```
# grub-install '(hd0)'
```

Ambos comandos instalarán GRUB en el primer sector (o MBR) del primer disco duro. Obsérvese que en el segundo ejemplo, son necesarias las comillas simples que rodean al nombre del dispositivo.

Por el contrario, si se desea instalar GRUB en el sector de arranque de una partición en lugar de en el MBR, se incluirá un identificador de la partición. Es decir:

```
# grub-install /dev/hda1
```

o bien

```
# grub-install '(hd0,0)'
```

101.2.3.2.8. Añadir un kernel a GRUB

El proceso que permite añadir un kernel a GRUB se puede resumir en los siguientes pasos:

1. Como usuario root, se debe cargar el archivo *menu.lst* o *grub.conf* en un editor de texto.
2. Hay que copiar una configuración válida para un kernel de Linux.
3. Se debe modificar la línea **title** para darle a la nueva configuración un nombre único.
4. Se debe modificar la línea **kernel** para apuntar al nuevo kernel. Si fuera necesario, se pueden personalizar las opciones necesarias de la línea **kernel**.
5. Si fuera necesario añadir, eliminar o cambiar un disco RAM, se deben realizar los cambios en la línea **intrd**.
6. Si fuera necesario, en la sección global, se debe cambiar la línea **default** para que apunte al nuevo kernel.
7. Finalmente, hay que guardar los cambios y salir del editor de texto.

A través de los siete pasos anteriores, GRUB estaría configurado para iniciar el nuevo kernel que aparecerá disponible cuando se reinicie el sistema.

Finalmente hay que resaltar que si hubiera problemas, sólo hay que arrancar con una configuración válida para poder resolverlos. No se debe eliminar una configuración válida para un kernel antiguo hasta que se haya determinado que el nuevo funciona correctamente.

Obsérvese cómo en el siguiente ejemplo como se ha añadido una nueva opción para arrancar Debian, en modo a prueba de fallos, sobre todo para sistemas con configuraciones problemáticas.

```
# /boot/grub/grub.conf
#
# Opciones globales
#
default=0
timeout=10
splashimage=(hd0,1)/grub/bootimage.xpm.gz
#
# Opciones de las imágenes Kernel
#
title Debian (2.6.25)
    root (hd0,0)
    kernel /vmlinuz-2.6.25 ro root=/dev/hda3
    initrd /initrd-2.6.25.img
title Debian failsafe (2.6.25)
    root (hd0,0)
    kernel /vmlinuz-2.6.25 ro root=/dev/hda3 ide=nodma \
        apm=off acpi=off vga=normal nosmp maxcpus=0 3
    initrd /initrd-2.6.25.img
#
# Otros sistemas operativos
#
title Windows 7
    rootnoverify (hd0,1)
    chainloader +1
```

101.2.3.3. Diferencias entre LILO y GRUB

El uso de LILO es un método más arriesgado que el de GRUB, porque un MBR que no haya sido configurado adecuadamente deja el sistema sin poder arrancar. Con GRUB, si el archivo de configuración está configurado de forma errónea, se disparará por defecto la interfaz de la línea de comandos de modo que el usuario pueda arrancar el sistema manualmente. Resumiendo algunos aspectos diferenciadores entre LILO y GRUB, se tiene:

1. **LILO** es incapaz de reconocer los sistemas de archivos, por lo que utiliza desplazamientos de disco sin procesar y el BIOS para cargar los datos. Se carga el código del menú y, a continuación, en función de la respuesta, carga, o el sector MBR del disco de 512 bytes como en Microsoft Windows, o la imagen del kernel de Linux.
2. **GRUB** por el contrario comprende los sistemas de archivos comunes ext2 , ext3 y ext4. Debido a que GRUB almacena sus datos en un archivo de configuración en vez de en el MBR y a que contiene un interfaz de línea de comandos, a menudo es más fácil rectificar o modificar GRUB si está mal configurado o corrupto.

Ante la tesisura de cuándo hay que instalar LILO o cuándo GRUB, por norma general será siempre GRUB. En el caso de una nueva instalación se recomienda utilizar GRUB, a no ser que la partición raíz se instale en los siguientes sistemas Raid:

1. Controladora Raid dependiente del CPU (como por ejemplo numerosas controladoras Promise o Highpoint)
2. Software RAID

3. LVM (Logical Volume Manager)

Algunos usuarios prefieren usar LILO sólo porque les es más familiar y otros porque GRUB puede causar problemas al arrancar determinado tipo de hardware. Finalmente, sólo añadir que, en caso de que se actualice desde una versión anterior en la que se utilizaba LILO, se recomienda volver a instalar LILO.

101.3. Cambiar niveles de ejecución, apagar y reiniciar el sistema.

Peso en el examen de certificación: 3 puntos.

Objetivo: Capacidad para gestionar el nivel de ejecución del sistema, lo que incluye cambiar a modos **monousuario**, **apagado** o **reinicio**. Capacidad de avisar a los usuarios antes de cambiar el nivel de ejecución y terminar adecuadamente todos los procesos. Establecimiento del nivel de ejecución predeterminado. Conocimiento básico que permita sustituir el proceso de inicialización del sistema (init) en caso de ser necesario.

Conceptos y áreas de conocimiento:

- .Configurar el nivel de ejecución predeterminado.
- .Cambiar entre los niveles de ejecución, incluido el modo monousuario.
- .Apagado y reinicio desde la línea de comandos.
- .Avisar a los usuarios antes de cambiar los niveles de ejecución u otro evento importante del sistema.
- .Finalizar correctamente los procesos antes de un cambio de nivel de ejecución.
- .Conocimiento de las características básicas de **systemd** y **upstart**

Términos y utilidades

/etc/inittab
init
telinit
shutdown
/etc/init.d

Después de que el núcleo de GNU/Linux ha arrancado, el programa init lee el archivo /etc/inittab para determinar el comportamiento para cada nivel de ejecución (runlevel). A no ser que el usuario especifique otro valor como un parámetro de autoarranque del núcleo, el sistema intentará entrar (iniciar) al nivel de ejecución por defecto.

Los runlevels indican qué cosas se ejecutan en cada momento, define el entorno o modo de ejecución del sistema. Podríamos tener un runlevel en el que la máquina se comportase como servidor web y otro solo para administración. Linux toma a estos runlevels como instrucciones precisas de qué iniciar y qué no. Los runlevels pueden modificarse a gusto del consumidor.

Linux posee ocho runlevels:

Runlevel	Estado del sistema
0	Apaga el sistema. Se detienen todos los procesos y se desmontan todos los sistemas de archivos y se desactiva la partición swap
1	Modo monousuario. Arranca el sistema sin activar la red ni ejecutar ningún servicio. Es el modo utilizado para reparar o depurar el sistema.

- 2 Modo multiusuario. Activa la red y los servicios disponibles. Funciona en la consola de texto
- 3 Modo multiusuario. Activa la red e incluye los *scripts* del directorio **/etc/rc3.d**. Funciona en consola de texto.
- 4 Modo multiusuario. Activa la red e incluye los *scripts* del directorio **/etc/rc4.d**. Funciona en consola de texto.
- 5 Modo multiusuario. Activa la red y los servicios disponibles e inicia **X Window** automáticamente.
- 6 Reinicia el sistema. Se detienen todos los procesos y se desmontan todos los sistemas de archivos y se desactiva *la partición swap*
- S Es el *runlevel* en el que arranca el sistema. Es similar a **1**, pero **1** se utiliza para volver a **S** cuando nos encontramos en otro *runlevel*

Cuando arranca o se reinicia el sistema, *init* no se queda en el *runlevel S* sino que pasa al especificado como defecto: habitualmente **2** o **5**, y se mantendrá en él hasta que forzamos un cambio.

101.3.1. Runlevel 1: single-user mode

Merece la pena dedicarle un apartado en concreto a este runlevel por lo particular de este. Tenemos prácticamente todo desactivado y por tanto como adelantaba en la introducción podemos hacer tareas que requieran no tener en el momento actividad de usuarios. Algo típico de este modo es corregir problemas de corrupción de filesystems.

Para arrancar en modo single user mode podemos hacerlo de dos formas. Por un lado indicarselo al arranque mediante el paso de comandos en el boot como vimos en la lección de ayer, para ello tan solo tendremos que añadir a la línea de arranque por defecto de nuestro sistema el numero 1 o la palabra single detrás. Otra opción es desde un sistema ya arrancado usar el comando init con el parámetro 1 detrás, es decir init 1. No es la mejor forma esta porque pasa automáticamente a este modo por lo que echa de golpe a todos los usuarios sin aviso previo, lo cual no es la mejor idea. No obstante la posibilidad existe y si queremos usar este modo sin necesidad de reinicio podemos usarlo, teniendo en cuenta que deberemos encargarnos antes de los usuarios.

101.3.2. Nivel de ejecución predeterminado

Cuando se inicia un sistema Linux, el nivel de ejecución predeterminado se establece a partir de la entrada **id:** en **/etc/inittab**. Para determinarlo podríamos ejecutar el siguiente comando:

```
#grep "^\d:" /etc/inittab
```

id:5:initdefault: que nos indica que el runlevel predeterminado es el 5.

101.3.3. El archivo /etc/inittab

/etc/inittab es el fichero de configuración de **init**. En este archivo se define, entre otros parámetros, los procesos que se ejecutarán en los distintos *runlevels*.

Cada línea está formada por cuatro campos separados por ":" :

id:runlevels:accion:proceso

id Es el código de identificación de cada línea de fichero, es único y está formado por una cadena de 1 a 4 dígitos, aunque normalmente son dos por compatibilidad con sistemas antiguos.

runlevels indica los runlevels donde se ejecutará la entrada.

accion Describe las acciones a ejecutar. Acciones disponibles:.

<i>wait</i>	No ejecutará otro comando hasta que acabe el actual.
<i>once</i>	Sólo se ejecutará el comando una vez.
<i>respawn</i>	El comando se reiniciará cada vez que termine.
<i>initdefault</i>	Indica el nivel de ejecución por defecto. No es necesario especificar ningún comando.
<i>ctrlaltdel</i>	Define la acción que se ejecutará al pulsar la combinación de teclas.
<i>boot</i>	El proceso será ejecutado durante el arranque. El campo runlevel es omitido.
<i>bootwait</i>	El proceso será ejecutado durante el arranque, mientras <i>init</i> esperará a que finalice. El campo runlevel es omitido.
<i>sysinit</i>	El proceso será ejecutado durante el arranque. Se ejecutará antes que boot o bootwait . El campo runlevel es omitido.
<i>powerfail</i>	El proceso se ejecutará si existe algún problema de alimentación eléctrica. Pero <i>init</i> no esperará a que el proceso termine.
<i>powerokwait</i>	El proceso se ejecutará tan pronto como la alimentación eléctrica este restablecida.
<i>powerfailnow</i>	El proceso se ejecutará cuando se le indique a <i>init</i> que la batería externa se está agotando y el suministro eléctrico está fallando.
<i>kbrequest</i>	El proceso se ejecutará cuando <i>init</i> reciba una señal de que una combinación especial de teclas se ha pulsado.

initdefault Indica el nivel de ejecución por defecto. No es necesario especificar ningún comando.

initdefault Indica el nivel de ejecución por defecto. No es necesario especificar ningún comando.

proceso Especifica los procesos que se ejecutarán.

Ejemplo de archivo */etc/inittab*:

```
#/etc# cat inittab # /etc/inittab: init(8) configuration.
# $Id: inittab,v 1.91 2002/01/25 13:35:21 miquels Exp $
# The default runlevel.
id:2:initdefault:
# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
si::sysinit:/etc/init.d/rcS
# What to do in single-user mode.
~~:S:wait:/sbin/sulogin
# /etc/init.d executes the S and K scripts upon change
# of runlevel.
#
# Runlevel 0 is halt.
# Runlevel 1 is single-user.
# Runlevels 2-5 are multi-user.
# Runlevel 6 is reboot.
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin
# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
# Action on special keypress (ALT-UpArrow).
#kb::kbrequest:/bin/echo "Keyboard Request--edit /etc/inittab to let
#this work."
# What to do when the power fails/returns.
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
# :::
#
# Note that on most Debian systems tty7 is used by the X Window
# System,
# so if you want to add more getty's go ahead but skip tty7 if
# you run X.
#
```

```

1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6

# Example how to put a getty on a serial line (for a terminal)
#
#T0:23:respawn:/sbin/getty -L ttys0 9600 vt100
#T1:23:respawn:/sbin/getty -L ttys1 9600 vt100

# Example how to put a getty on a modem line.
#
#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttys3

```

Nota: Las distribuciones Linux están dejando de utilizar /etc/inittab, los nuevos sistemas emplean /etc/event.d para controlar la secuencia de inicio de init. /etc/inittab se sigue utilizando para especificar el modo de ejecución por defecto.

101.3.4. Los archivos de inicio SysV

El sistema de niveles de ejecución SysV init provee un proceso estándar para controlar cuáles programas inicia o detiene cuando se inicializa un nivel de ejecución. SysV init fué escogido porque es más fácil de usar y es más flexible que el proceso tradicional init estilo BSD. Los archivos de configuración para SysV init están en el directorio /etc/rc.d/. Dentro de este directorio, se encuentran los scripts rc, rc.local, rc.sysinit y, opcionalmente, los scripts rc.serial así como los siguientes directorios:

init.d/ rc0.d/ rc1.d/ rc2.d/ rc3.d/ rc4.d/ rc5.d/ rc6.d/

El directorio init.d/ contiene los scripts que el comando /sbin/init utiliza cuando controla servicios. Cada uno de los directorios numerados representan los seis niveles de ejecución configurados por defecto.

101.3.4.1. El archivo /etc/init.d

En /etc/init.d se encuentran todos los scripts de inicio de todos los *runlevels*.

Cada *runlevel* dispone de un directorio /etc/rcX.d (X=runlevel) donde se encuentran los enlaces correspondientes a los scripts, situados en /etc/init.d/, que se ejecutarán para este *runlevel*.

Si observamos las siguientes líneas de archivos /etc/inittab:

```

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.

si::sysinit:/etc/init.d/rcS
# /etc/init.d executes the S and K scripts upon change
# of runlevel.
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6

```

Vemos que para cada *nivel de ejecución* se ejecuta el *script* : **rc** seguido de un argumento que indica el *runlevel*. De esta manera **rc** ejecutará los enlaces de **/etc/rcX.d** (X=runlevel)

Dentro de **/etc/rcX.d** los *scripts* se organizan mediante un código formado por una letra (S o K) y un número

- Los que empiezan por **S** se ejecutarán al entrar en el *runlevel* (rc les pasa el parámetro start) y los que empiezan por **K** se detendrán (rc les pasa el parámetro stop).
- El número siguiente indica el orden de ejecución: orden ascendente. Esta característica permite a los desarrolladores de las distribuciones controlar el orden en el que se ejecutan los scripts asignándoles números apropiados. Este control es importante porque unos normalmente unos servicios dependen de otros. Por ejemplo los servicios de red se deben iniciar después de que se active la red.

Ejemplos serían: *S10network* y *K35smb*.

Estos *scripts* aceptan de inicio del sistema, inician o detienen los servicios dependiendo del parámetro que se les pase, por lo que la nomenclatura de los scripts controlan si se inician o detienen cuando se pasa a un modo de ejecución. Aparte de estos dos argumentos: **start** y **stop**, suelen aceptar otros como: **reload**, **restart** y **status**.

Para determinar que servicios están activos en un modo de ejecución , podríamos buscar los scripts cuyos nombres de fichero comiencen por S en el directorio de scripts de inicio apropiado, o utilizar una herramienta de administración de modos de ejecución

101.3.5. Estructura del directorio /etc y al proceso init

/etc/rc.sysinit o /etc/init.d/rcS

rc.sysinit es un script de inicialización monolítica. en Debian varia un poco ya que **rcS** corre varios pequeños scripts que están en dos directorios. En ambos casos se lanzan en boot time. Estos son unos scripts que se encargan de cargar digamos las funciones básicas antes de que se levanten los demonios por ejemplo se encarga de montar los filesystems.

Veamos el funcionamiento práctico de Debian en esto con unas pruebas que he estado haciendo. Por cierto os recomiendo que hagáis vosotros pruebas de estas, que entréis en vuestras máquinas del laboratorio y os miréis los directorios **/etc/rc*** ya que hay cosas muy interesante y en general están documentadas dentro de los mismos ficheros de configuración.

Básicamente **/etc/init.d/rcS** nos dice que cargara “**/etc/rcS.d/**” de forma numérica/alfabética. Si entramos en el directorio veremos los scripts que cargarán los servicios y un **README** explicándonos como activar o desactivar servicios. En serio echad un vistazo a estos directorios/ficheros es básico para el aprendizaje el hacerlo. Y dedicarle el tiempo que haga falta a esto, considero fundamental para corregir problemas y configurar el sistema apropiadamente comprender de una manera correcta la forma de arranque del sistema.

/etc/rc.local

No se usa en sistemas Debian, en RedHat si. Cuando ya están todos los scripts y ya con todos los demonios por defecto inicializados se lanza. Contiene las personalizaciones locales que modifican los servicios ya lanzados. El sentido de esto es hacer las modificaciones aquí en lugar del **rc.sysinit** la razón de hacerlo así es que en los upgrades del sistema el **rc.sysinit** se sobre-escribe perdiendo estos cambios pero el **rc.local** se conserva.

/etc/rc

Este script que no existe en Debian se usa para cambiar entre distintos runlevels.

/etc/init.d

En este directorio se encuentran los scripts encargados de inicializar (start), reiniciar (reboot) y parar (stop) los demonios, entre otros. desde la carga de los distintos runlevels se llama a estos scripts pero tambien podemos invocarlos manualmente. Por ejemplo si queremos inicializar a mano el servidor de apache en Redhat usaremos: “/etc/init.d/httpd start”. También podemos tener los parametros de status y de reload en los scripts, es recomendable ver la documentación de cada demonio para entender que hace cada opción en concreto para cada demonio.

En la gran mayoría de los casos al instalar un paquete automáticamente se nos crean estos scripts. En cualquier caso podríamos añadirlos nosotros manualmente o realizar modificaciones en rc.local (recordemos que en tan solo el caso de RH esto último).

/etc/rc0.d – /etc/rc6.d

Ya tratados por encima anteriormente, estos directorios en realidad no contienen ficheros si no enlaces simbólicos a los scripts de inicialización de /etc/init.d . De esta forma en caso de querer cambiar algo en el script de arranque de apache y este estar presente en los init 2, 3 y 5 en lugar de modificarlos en cada uno tan solo lo haremos en /etc/init.d . Cuando se carga un runlevel se cargan todos los scripts que esten dentro de esta carpeta pero con cierta restricción y es que los nombres de los enlaces simbólicos tienen una metodología especial. En caso de empezar por S se cargarán, pero si empiezan por K (kill) se matan. Si realizamos modificaciones y queremos cargar de nuevo el init con los cambios deberemos ejecutar el script: “update-rc.d ”. También como parte de la política de nombres después de la S o la K irá un número secuencial, este indicará el orden en el que se ejecutarán el inicio o parada de los servicios. Este número puede repetirse y significará que esos demonios se pararán o inicializarán a la vez y una vez estén arrancados pasarán al siguiente número. Si renombramos esto podremos por tanto cambiar el orden de parada o de arranque.

101.3.6. Gestión de los servicios de los modos de ejecución

Para gestionar los servicios de los modos de ejecución tenemos varias herramientas: chkconfig, ntsysv y rc-update.

Tenemos que saber que los script de sysv son enlaces simbólicos al original, a fin de no tener que copiar el mismo scripts en cada directorio. Por ello estas herramientas nos permiten modificar el script original sin tener que seguir todas sus copias en todos los directorios de modos de ejecución de SysV. También podríamos modificarlos editando los ficheros de sus enlaces.

101.3.6.1. CHKCONFIG

El comando chkconfig se usa para cambiar, actualizar y consultar información de runlevel para los servicios del sistema. chkconfig es un comando de administrador.

Chkconfig (también localizado en /sbin) nos permite controlar la ejecución de servicios entre reinicios del equipo. Es decir, es el responsable de administrar dentro de la carpeta /etc/rcx.d si un guión se ejecuta (S), o se apaga (K) en determinado nivel de ejecución:

Ejemplos de uso:

chkconfig --list: Lista todos los servicios configurados, así como su estado predeterminado

en cada nivel de ejecución.

```
nfs-common 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
pcmcia 0:off 1:off 2:off 3:on 4:on 5:on 6:off  
xprint 0:off 1:off 2:off 3:on 4:on 5:on 6:off
```

Si estuviéramos interesados en un servicio específico, bastaría con especificar su nombre, por ejemplo **#chkconfig - - list nfs-common** provocará la salida

```
nfs-common 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Si quisieramos desactivar / activar un servicio, bastaría con pasarle el nombre y el parámetro off / on, por ejemplo:

#chkconfig kudzu off: indica que el servicio kudzu no estará disponible para ningún nivel de ejecución.

chkconfig httpd on: Indica que el servidor web estará disponible para todos los niveles de ejecución especificados para dicho servicio

Para especificar los niveles de ejecución en que estará activo un servicio:

chkconfig -level 35 squid on: Indica que el servidor proxy squid estará disponible únicamente para los niveles de ejecución 3 y 5.

Con el parámetro reset le asignaríamos su valor por defecto.

El parámetro --add registra un nuevo script, añadido al directorio principal de scripts de inicio, y añade los enlaces de inicio y final apropiados en los directorios de modos de ejecución adecuados. Chconfig chequeará el script en busca de comentarios especiales que indiquen los niveles de ejecución por defecto.

101.3.6.2. NTSYSV

La utilidad **ntsysv** es una herramienta interactiva en modo texto. Provee una interfaz sencilla para activar y desactivar servicios. Puede usar **ntsysv** para activar o desactivar un servicio. También puede usar **ntsysv** para configurar los niveles de ejecución. Por defecto, únicamente el nivel de ejecución actual es configurado. Para configurar un nivel de ejecución diferente, especifique uno o más niveles con la opción --level. Por ejemplo, el comando ntsysv --level 345 configura los niveles de ejecución 3, 4, y 5.

La interfaz **ntsysv** funciona de forma similar al programa de instalación en modo texto. Utilice las flechas arriba y abajo para desplazarse por la lista. La barra espaciadora selecciona o anula la selección de servicios, y también sirve para "pulsar" los botones **Aceptar** y **Cancelar**. Para desplazarse en la lista de servicios y entre los botones **Aceptar** y **Cancelar**, use la tecla [Tab]. Un asterisco, *, significa que el servicio está activado. Con la tecla [F1] se mostrará una breve descripción de cada servicio.

101.3.7. Cambiar Niveles de Ejecución

A veces es necesario revisar el modo de ejecución actual, normalmente para comprobar el estado si hay algo que no funciona o antes de cambiar el modo de ejecución. Se pueden hacer de dos maneras diferentes: verificar el modo de ejecución por defecto y el modo de ejecución actual.

101.3.7.1. Comprobar y cambiar el modo de ejecución por defecto

Se debe buscar en el archivo /etc/inittab la linea que especifica la acción initdefault (con el comando less, editando el archivo, con el comando grep,...). Por ejemplo:

#grep :initdefault: /etc/inittab producirá una salida similar a id:5:initdefault

Para cambiar el modo de ejecución por defecto la siguiente vez que se inicie el sistema , se debe editar la linea initdefault de /etc/inittab y cambiar el nivel de ejecución por el que se desee.

Si el sistema carece del archivo /etc/inittab, se debe crear uno que sólo contenga la linea :initdefault:

101.3.7.2. Determinar el nivel de ejecución actual

Si el sistema está en ejecución, podemos determinar el nivel de ejecución con el comando runlevel

#runlevel podría producir la salida N 2

El primer carácter es el modo de ejecución y anterior. Cuando es N indica que el sistema no ha cambiado de modo de ejecución desde el arranque.

El segundo carácter es su nivel de ejecución actual.

Se puede cambiar de nivel de ejecución en un sistema en ejecución con init y telinit.

101.3.7.3. Cambiar el nivel de ejecución en un sistema en funcionamiento

Por ejemplo para obtener más servicios o para apagar o reiniciar el ordenador. Esto se puede hacer con los comandos init, telinit, shutdown, halt, reboot y poweroff.

INIT y TELINIT

El proceso init es el primer proceso ejecutado por el kernel, pero podemos utilizarlos para que el sistema vuelva a leer /etc/inittab e implemente los cambios que encuentre allí o que pase a un nuevo nivel de ejecución, pasándole como parámetro el número de runlevel al que pasar, por ejemplo:

init 6 cambia al modo de ejecución 6 (reinicio del sistema)

init es un comando del sistema y como todos acepta parámetros, pero es habitual utilizar telinit que es un enlace a /sbin/init y no init directamente.

Sus parámetros son:

Opciones de telinit

Opciones Descripción

N cambia al runlevel N. Donde N es el nivel al que queremos cambiar, N=[0,1,2,3,4,5,6,s,S]

u, U vuelve a ejecutar el proceso **init**

q, Q vuelve a leer el archivo **/etc/inittab** y aplica los cambios necesarios.

101.3.8. Apagado y reinicio del Sistema

Aunque podemos apagar y reiniciar la máquina mediante init o su alias telinit, esto provocaría un cambio inmediato al nuevo estado, lo que podría provocar molestias a otros usuarios conectados al

sistema, que no recibirían aviso previo. Es más conveniente hacerlo mediante el comando shutdown.

Son varias las razones de esto, la primera es porque shutdown avisa a los usuarios. Indicará a los usuarios que se va a reiniciar o apagar la máquina con un mensaje que es personalizable. En segundo lugar podemos fijar un tiempo de espera, podemos decir que la máquina se apague a una hora en concreto o pasado un tiempo, de esta forma podríamos hacer que nuestro ordenador deje de funcionar después de las 12 que ya no le necesitaremos o tras 10 minutos y así alertar a usuarios conectados y tengan tiempo así a guardar sus trabajos. Shutdown tiene una serie de opciones con las que ejecutarse que resultan interesantes, estan son:

- **-f** arranque rápido, en el siguiente arranque no comprobará los filesystems.
- **-h** después de parar todos los servicios la máquina se apagará.
- **-H** lo detiene (finaliza pero no apaga).
- **-k** en realidad no se apaga pero manda el mensaje de apagado.
- **-r** tras parar los servicios se reiniciará.
- **-F** es el contrario de -f en el siguiente arranque le toque o no se forzará a una comprobación de los filesystems.

Para indicar el tiempo podemos usar lo siguiente:

+ (número): número será el número de minutos tras lo cual se hará shutdown de la máquina now: la máquina hará un shutdown en ese mismo instante
hh:mm: la máquina hará shutdown a la hora indicada (con formato 24h)

Para avisar a los usuarios se puede añadir un mensaje al final del comando, por ejemplo:

```
# shutdown -h +15 "Apagado en 15 Minutos."
```

Si se planifica un apagado, se puede cancelar con la opción -c:

```
# shutdown -c "Cancelado apagado en 15 minutos"
```

Existen otros comandos relacionados con el apagado y el reinicio.

- El comando **halt** detiene el sistema.
- El comando **poweroff** es un enlace simbólico del comando **halt**, que detiene el sistema y luego intenta apagarlo.
- El comando **reboot** es otro enlace simbólico del comando **halt**, que detiene el sistema y luego lo reinicia.

Si cualquiera de ellos es llamado cuando el sistema no se encuentra en los niveles de ejecución 0 ó 6, se invocará el comando **shutdown** correspondiente.

La sintaxis es:

```
halt [-d | -f | -h | -n | -i | -p | -w]  
reboot [-d | -f | -i | -n | -w]  
poweroff [-d | -f | -h | -n | -i | -w]
```

Opción	Descripción
- d	No escribir registro wtmp (en el archivo /var/log/wtmp) El flag -n implica -d
- h	Poner todos los discos duros del sistema en modo de espera antes de que el sistema se detenga o apague
- n	No sincronizar antes de reiniciar o detener
- i	Apagar todas las interfaces de red.

- p Cuando detenga el sistema, lo apaga también. Esto es por defecto cuando el halt se llama como poweroff.
- w No reiniciar o detener, sólo escribir el registro wtmp (en el archivo /var/log/wtmp)

EJEMPLO:

1. Para detener el sistema:

halt

Este comando es similar al **poweroff**, que apaga el sistema.

2. Para apagar el sistema:

poweroff

El comando **poweroff** se utiliza para apagar el sistema.

3. Para reiniciar el sistema:

reboot

El comando **reboot** se utiliza para reiniciar el sistema.

NOTA: Para usar la combinación CTRL+ALT+DEL para el reinicio, debería aparecer en el fichero */etc/inittab* la siguiente linea a fin de ser interpretada por *init*.

```
# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

101.3 EXTRAS

Existen varias soluciones para superar el proceso de inicialización basado en init en los sistemas unix. Solaris incorporó un sistema SMF (basado en servicios) para superar algunas debilidades del demonio init tradicional ya que es estrictamente síncrono, bloqueando futuras tareas hasta que la actual se haya completado. Sus tareas deben ser definidas por adelantado, y solo pueden ser ejecutadas cuando el demonio init cambia de estado (cuando la máquina se arranca o se apaga).

Posteriormente se implementó el sistema upstart (desarrollado por Canonical) en algunos sistemas GNU/linux y recientemente systemd. (desarrolladores de Fedora)

101.3.1 UPSTART

Upstart trabaja de forma asíncrona supervisando las tareas mientras el sistema esta arrancado. También gestiona las tareas y servicios de inicio cuando el sistema arranca y los detiene cuando el sistema se apaga

Su uso es sencillo y compatible con sysvinit fueron objetivos explícitos en el diseño. Por lo tanto, Upstart es capaz de ejecutar scripts de sysvinit sin modificaciones. De esta manera se diferencia de la mayoría de reemplazos de init, que normalmente requieren una transición completa para funcionar correctamente y no son compatibles con un entorno mixto formado por métodos de arranque tradicionales y nuevos.

El núcleo de Upstart es el proceso init, lanzado por el nucleo del SO como primer proceso. init utiliza los ficheros de configuración presentes en /etc/init para saber que trabajos hay que gestionar. Internamente init almacena el estado los trabajos y mantiene una cola de eventos pendientes de procesar. Los trabajos y los eventos son los elementos básicos del funcionamiento de Upstart. La gestión de los eventos da como lugar el arranque y parada de trabajos en función de su especificación. Los trabajos también se pueden arrancar y parar manualmente por el administrador, as como el control de los trabajos puede producir eventos internos

Un trabajo tiene un nombre y un fichero de configuración asociado. El nombre del trabajo se obtiene del nombre base del fichero de configuración sin la extensión. El proceso init esta observando cambios en este directorio constantemente por lo que añadir o modificar trabajos no requiere el envío de señal alguna al proceso init. No obstante, si la sintaxis de un fichero no es correcta, el trabajo sera ignorado de manera aparentemente silenciosa. Aparte del fichero de configuración todo trabajo tiene asociado un estado (parado, arrancando, arrancado, ...) y una meta (arrancar o parar).

Junto con los ficheros de configuración y el binario de init también hay una serie de mandatos para gestionar los trabajos de manera manual, corregir la sintaxis de los ficheros de configuración y depurar su ejecución. Estos mandatos son initctl, start , stop y status. En concreto para habilitar la salida de depuración se ejecuta desde una consola el mandato initctl log-priority debug

Hay varios tipos de trabajos:

1. Tareas: Las tareas son trabajos por los que el evento o mandato que los arranca debe esperar a que finalicen. Se espera que sean trabajos cortos y de duración determinada.
2. Servicios: Trabajos asociados a procesos servidores (demonios). En este caso el lanzador del trabajo sólo espera a que el servidor arranque (fork del proceso).
3. Abstractos: Son trabajos que carecen de cláusula exec o script, por tanto, no tiene un proceso asociado, pero se pueden usar otras cláusulas de configuración para que hagan trabajo útil o ayudar a sincronizar otros trabajos

Los eventos son testigos que el proceso init encola internamente y son procesados según llegan. Las acciones desencadenadas por los eventos no tienen garantía de ejecutarse en ningún orden en particular. Eso quiere decir que si dos trabajos son disparados por el mismo evento, no se puede esperar que ejecuten siempre en el mismo orden. Hay varios tipos de eventos:

1. Métodos: Estos eventos no devuelven el control al proceso que los genera hasta que el evento ha sido procesado por completo. Si, por ejemplo, un evento es causa de la ejecución de una tarea (no un servicio), el proceso que genera el evento no puede continuar su ejecución hasta que la tarea ha finalizado. Por defecto, todos los eventos son de este tipo.
2. Señales: básicamente son iguales que los métodos, pero el llamante no espera a que el evento haya sido procesado por Upstart.
3. Hooks: Eventos que son generados por el sistema a través de otros demonios, como el demonios de udev o el demonio de dbus

.Los eventos se pueden enviar manualmente, incluso con envío de variables.

```
# initctl emit --no-wait evento EMISOR=root DAT01=valor1
```

Un fichero de configuración es una secuencia de cláusulas acompañada de comentarios y líneas de metainformación. Las cláusulas más importantes son:

- Cláusula de arranque de trabajos:

start on <evento> [<VARIABLE>=<PATRON> ...]
start on <expresion-logica-con-eventos-y-patrones>

Ejemplos:

start on startup
start on (started curro and started tajo)
start on stopping curro INSTANCE=currele
start on runlevel [234]
start on un_evento VAR1=valor1 VAR2=valor2

Cuando las condiciones de la cláusula start on se cumplen, la meta del trabajo pasa a ser start .

Las cláusulas con expresiones lógicas son un poco particulares. Una cláusula and bloquea cualquier proceso que emite un evento involucrado en la misma. Cuando la cláusula se hace válida y se dispara el trabajo, entonces los emisores de los eventos pueden ser desbloqueados (que el emisor de un evento sea manualmente desbloqueado depende de si el trabajo disparado es una tarea o un servicio). Lógicamente, esto no se aplica a las señales. Un ejemplo de aplicación de esta peculiaridad es usar un evento abstracto como barrera de sincronización de una serie de trabajos que deben comenzar simultáneamente.

- Cláusula de parada de trabajos:

stop on <expresion-eventos> [VARIABLE=PATRON, ...]

Cuando la condición acerca de los eventos de la cláusula se hace cierta, el objetivo de este trabajo pasa a ser stop

En la cláusula de parada se pueden usar variables del entorno del trabajo. Por ejemplo una especificación de tarea podría ser

start on startevent

stop on stopevent SENDER=\$SENDER

Esta trabajo arrancaría cuando se envía startevent con cualquier valor, y pararía cuando se envía stopevent con la variable SENDER con el mismo valor que tomó en la cláusula start on para startevent

- Cláusula de trabajo principal:

script

...

end script

o

exec mandato args

Esta cláusula indica el código o proceso principal que debe ejecutar este trabajo. No se trata de una cláusula obligatoria. Si no aparece estaríamos ante un trabajo abstracto, que nunca pasaría a parado a no ser que sea parado manualmente o por efecto de la cláusula stop on.

- Cláusula pre-start script ... end script o pre-script exec ...: Se ejecuta antes de lanzar el

proceso principal.

- Cláusula post-start script ... end script o post-script exec: Se ejecuta justo a continuación de lanzar el proceso principal.
- Cláusula post-stop script ... end script o post-stop exec: Se ejecuta cuando el proceso principal ha terminado, independientemente de su estado de terminación.
- Cláusula pre-stop script ... end script o pre-stop exec: Esta cláusula se ejecuta antes de enviar la señal de terminación al proceso principal. Si el proceso ha terminado por sí mismo esta cláusula no se ejecuta.
- Cláusula expect fork . Esta cláusula indica a Upstart que el proceso principal hará exactamente un fork , así que el PID que debe asociar al trabajo es el del hijo. El PID almacenado será el que se usará para enviar señales de terminación
- Cláusula expect deamon . Esta cláusula es como expect fork, pero Upstart espera que haya dos fork exactamente. El PID almacenado será por, tanto, el del nieto del proceso lanzado por upstart como trabajo.
- Cláusula respawn. Esta cláusula relanza cualquier tarea (incluyendo las cláusulas pre-start y post-start) si por cualquier razón el proceso principal ha terminado y la meta no ha sido cambiada a stop, lo que se puede entender como una terminación anómala del trabajo. Para las tareas, una terminación anómala que no puede ser cualquier aquella que produce un valor de retorno distinto de 0.
- Cláusula instance \$VAR. A menos que se incluya esta cláusula, sólo puede haber una instancia de cada trabajo. La variable VAR debe ser alguna variable proporcionada por Upstart al entorno de ejecución del trabajo o alguna variable asociada a los eventos de la cláusula.
- Cláusula task. Cuando se añade esta cláusula, el trabajo se considera tarea, de no existir se considera que es un servicio.
- Cláusulas env VAR=VALOR . Estas cláusulas sirven para declarar variables de entorno para este trabajo.
- Cláusulas export VAR. Estas cláusulas sirven para exportar variables que son parte del entorno de un trabajo a aquellos trabajos que reciben señales internas por su parte.
- Cláusulas emits <patrón>. Estas cláusulas son meramente informativas. Sirven para documentar que eventos pueden ser lanzadas durante la ejecución de un trabajo. Existen ejemplos en los ficheros de arranque del sistema.
- Cláusula manual . Hace que las cláusulas start on y stop on sean ignoradas y sólo se pueda ejecutar el trabajo de manera manual.

La lista completa de todas las cláusulas se puede consultar en la documentación oficial Un trabajo tiene asociados un estado y una meta. En función del estado y de la meta se producirán transiciones de un estado a otro. Dichas transiciones también dependen de si se producen errores en la ejecución de alguna cláusula. En general, un error en la ejecución de cualquier cláusula produce un cambio de meta a stop y la emisión de los eventos stopping y stopped.

La tabla con las transiciones de estados según el estado actual y la meta es la siguiente:

Se emiten los siguientes eventos en algunas transiciones:

- starting: emitido cuando la meta cambia de stop a start en el estado starting

Estado	Objetivo	
	start	stop
waiting	starting	n/a
starting	pre-start	stopping
pre-start	spawned	stopping
spawned	post-start	stopping
post-start	running	stopping
running	stopping	pre-stop or stopping
pre-stop	running	stopping
stopping	killed	killed
killed	post-stop	post-stop
post-stop	starting	waiting

- started: emitido tras la ejecución de post-start si todo ha ido bien.
- stopping: emitido cuando la meta cambia de start a stop y se cambia el estado a stopping
- stopped: emitido cuando el trabajo pasa de post-stop a waiting

Existen varios mandatos para consultar el estado y meta tanto de un trabajo en particular como de todos los trabajos del sistema. Los más importantes son:

initctl status <trabajo> Nos devuelve la meta/estado de un trabajo, así como el PID de la tarea.

initctl list Da una lista con el estado de todos los trabajos que Upstart está trazando

Para arrancar un trabajo: se puede arrancar porque su cláusula start on se dispare a causa de un evento, o por medio de los mandatos *initctl start <trabajo>* o *start <trabajo>*

Para detener un trabajo: puede hacerse manualmente con *initctl stop <trabajo>* o *stop <trabajo>* o puede confiarse en una cláusula stop on disparada por eventos. El efecto en ambos casos es que la meta del trabajo cambia a stop

El primer evento que Upstart genera es startup. Upstart tiene como uno de sus objetivos ser compatible con SysV init. Por ello, simula eventos runlevel y establece un DEFAULT_RUNLEVEL a través del job /etc/init/rc-sysinit.conf.

Ejemplo de fichero de configuración:

```
#cat /etc/init/mysql.conf
# MySQL Service
description      "MySQL Server"
author           "Mario Limonciello <superm1@ubuntu.com>"
start on (net-device-up
          and local-filesystems
          and runlevel [2345])
stop on runlevel [016]
[...]
exec /usr/sbin/mysqld
[...]
```

101.3.2 SYSTEMD

systemd inicia y supervisa todo el sistema y se basa en la noción de *unidades* compuestas de un nombre, tipo y coincidencia de un archivo de configuración con el mismo nombre y tipo (por ejemplo, una unidad avahi.service tiene un archivo de configuración con el mismo nombre y es una unidad de encapsulado del demonio Avahi). Existen siete tipos diferentes de unidades:

1. **service**: este es el tipo más obvio de unidad: demonios que pueden ser iniciados, detenidos, reiniciados, recargados.
2. **socket**: Esta unidad encapsula un socket en el sistema de archivos o en Internet. Actualmente systemd soporta el funcionamiento de los tipos de sockets AF_INET, AF_INET6, AF_UNIX, datagram y paquetes secuenciales. También puede soportar FIFOs clásicos como transporte. Cada unidad socket tiene una unidad de servicio correspondiente, que se inicia si la primera conexión entra en el socket o FIFO (por ejemplo, nsqd.socket inicia nsqd.service en una conexión entrante).
3. **device**: esta unidad encapsula un dispositivo en el árbol de dispositivos de Linux. Si un dispositivo está marcado para ello a través de reglas udev, se expondrá como una unidad device en systemd. Las propiedades establecidas con udev pueden utilizarse como configuración fuente para establecer dependencias para unidades device.
4. **mount**: esta unidad encapsula un punto de montaje en la jerarquía del sistema de archivos.
5. **automount**: este tipo de unidad encapsula un punto de montaje automático en la jerarquía del sistema de archivos. Cada unidad automount tiene una unidad mount correspondiente, que se inicia (es decir, montada) tan pronto como se accede al directorio de automontaje.
6. **target**: este tipo de unidad se utiliza para la agrupación lógica de unidades: en vez de realmente hacer nada por sí misma simplemente hace referencia a otras unidades, que así pueden ser controladas conjuntamente, (por ejemplo, multi-user.target, que es un objetivo que básicamente desempeña el papel de nivel de ejecución 5 en el sistema clásico SysV; o bluetooth.target que es solicitado tan pronto como esté disponible un adaptador bluetooth y que simplemente recibe servicios relacionados con bluetooth que de lo contrario no tendrían que iniciarse: bluetoothd, obexd y cosas por el estilo).
7. **snapshot**: similar a las unidades target, snapshots en realidad no hacen nada ellas mismas y su único propósito es hacer referencia a otras unidades.

systemd tiene muchas características novedosas pero las principales son:

- Capacidades de paralelización agresiva usando socket: para acelerar el arranque completo e iniciar más procesos en paralelo, systemd crea los sockets de escucha antes de iniciar realmente el demonio y sólo pasa el socket al mismo. Todos los sockets para todos los demonios se crean en un solo paso en el sistema de inicio (init) y luego en un segundo paso ejecuta a la vez todos los demonios. Si un servicio necesita de otro, y no se ha iniciado completamente, lo que sucederá es que la conexión esté en la cola del servicio de suministro y el cliente potencialmente se bloqueará en esa única solicitud. Pero sólo ése cliente se bloquea y solo en esa solicitud. También, las dependencias entre servicios ya no tienen que estar configuradas para permitir el correcto arranque paralelizado: iniciando todos los sockets a la vez y un servicio que necesite de otro, seguramente se podrá conectar a su socket.
- Activación D-Bus para iniciar servicios: utilizando la activación bus, un servicio puede ser iniciado la primera vez que se accede. La activación bus también da la sincronización por solicitud mínima necesaria para poner en marcha los proveedores y consumidores de servicios de D-Bus al mismo tiempo: iniciando un servicio al mismo tiempo que otro, si uno

es más rápido, a través de la lógica de activación bus, las colas de D-Bus lo solicitan hasta que el otro consigue establecer su nombre de servicio.

- ofrece inicio de demonios bajo demanda
- Realiza el seguimiento de procesos utilizando Linux **cgroups**: cada proceso ejecutado obtiene su propio cgroup y es muy fácil de configurar systemd para realizar servicios en cgroups que han sido configurados externamente, por ejemplo a través de las utilidades de libcgroups.
- Soporta snapshotting y restauración de estado del sistema: las Snapshots pueden ser utilizadas para guardar/restaurar el estado de todos los servicios y unidades del sistema de inicio (init). Principalmente tiene dos casos de uso: para permitir al usuario temporalmente entrar en un estado específico como «Shell de emergencia», la terminación de los servicios actuales y proporcionar una manera fácil para regresar al estado anterior, activando otra vez todos los servicios que consiguió desactivar temporalmente.
- Mantiene puntos de montaje y automontaje: systemd supervisa cómo vienen y van todos los puntos de montaje y también puede utilizarse para montar o desmontar los puntos de montaje. /etc/fstab puede utilizarse aquí como una fuente de configuración adicional para estos puntos de montaje. Usando la opción **comment=** de fstab incluso puede marcar entradas en /etc/fstab para que sean controladas por systemd a través de los puntos de automontaje.
- Implementa una elaborada lógica de control de servicios transaccional basada en la dependencia: systemd admite varios tipos de dependencias entre servicios (o unidades), utilizando las opciones *After/Before*, *Requires* y *Wants* en los archivos de configuración de la unidad para fijar el orden de cómo se activarán las unidades. *Requires* y *Wants*, expresan una dependencia de requisito positivo, obligatorio u opcional. Existe *Conflicts* que expresa una dependencia de requisito negativo y otras menos utilizadas. Como un control transaccional, si se solicita una unidad de arranque o se apaga, systemd la añadirá con todas sus dependencias a una transacción temporal, verificando si la transacción es consistente (o el orden vía After/Before de todas las unidades es de ciclo libre). Si no es así, systemd intentará solucionarlo y eliminará las tareas no esenciales de la transacción que podrían quitar el bucle.

Las **herramientas de administración** de systemd son:

- **systemctl**: usada para examinar y controlar el estado del sistema systemd y el administrador de servicios.
- **systemd-cgls**: muestra recursivamente el contenido del árbol de jerarquías de un determinado grupo de control de Linux.
- **systemadm**: una interfaz gráfica para el sistema systemd y el administrador de servicios que permite la introspección y el control de systemd.

Ejemplos de uso:

Activa un servicio inmediatamente:

```
systemctl start foo.service
```

Desactiva un servicio inmediatamente:

```
systemctl stop foo.service
```

Reinicia un servicio:

```
systemctl restart foo.service
```

Muestra el estado de un servicio, incluyendo si se está ejecutando o no:

```
systemctl status foo.service
```

Permite un servicio para iniciarse en el arranque:

```
systemctl enable foo.service
```

Deshabilita un servicio para que no se inicie durante el arranque:

```
systemctl disable foo.service
```

Comprueba si un servicio ya está habilitado o no:

```
systemctl is-enabled foo.service; echo $?
```

0 indica que está activado, y 1 indica que está desactivado.

Cambiar a 'nivel de ejecución 3'

```
systemctl isolate multi-user.target (o) systemctl isolate runlevel3.target
```

Cambiar a 'nivel de ejecución 5'

```
systemctl isolate graphical.target (o) systemctl isolate runlevel5.target
```

Conocer el nivel de ejecución actual

```
systemctl list-units --type=target
```

102 INSTALACIÓN DE GNU/LINUX Y ADMINISTRACIÓN DE PAQUETES.

- 102.1. Dimensionar particiones de disco.
- 102.2. Instalar el gestor de arranque.
- 102.3. Gestión de librerías compartidas.
- 102.4. Utilización del sistema de paquetes Debian.
- 102.5. Utilización del sistemas de paquetes RPM y YUM.

102.1. Dimensionar particiones de disco.

Peso en el examen de certificación: 2 puntos.

Objetivo: Diseñar esquemas de particionado de disco para un sistema GNU/Linux.

Conceptos y áreas de conocimiento:

- .Asignar los sistemas de ficheros y el espacio de intercambio en particiones separadas o discos separados.
- .Adaptar el diseño para el uso previsto del sistema.
- .Definir, ubicar y configurar la partición **/boot** para que pueda iniciar el sistema atendiendo a los requerimientos de la arquitectura hardware.
- .Conocimientos básicos sobre LVM.

Términos y utilidades

Sistema de ficheros / (root)

Sistema de ficheros /home

Sistema de ficheros /var

swap

particiones

puntos de montaje

Una partición en un disco duro es cada una de las divisiones que se pueden realizar sobre una unidad física para tener varias unidades lógicas.

Toda partición, para poder ser utilizada se le ha de dar formato mediante un sistema de archivo. Un sistema de archivo es una abstracción que organiza la información almacenada en una unidad de disco en directorios o carpetas y archivos.

102.1.1 Herramienta **fdisk.**

La principal herramienta que se utiliza en Linux para crear particiones es **fdisk**. Esta herramienta crea una tabla de particiones y la almacena en el primer sector del disco duro (sector 0, también conocido como superblock).

Fdisk puede utilizarse de dos maneras, o bien sin parámetros, en cuyo caso se muestra un menú textual o bien mediante parámetros en la línea de comandos.

102.1.1.1 Parámetros de **fdisk.**

Los principales parámetros con los que se puede utilizar fdisk son:

-l: muestra una lista con las tablas de particiones presentes en el sistema.

-v: muestra a la versión de fdisk instalada.

Sin opciones, **fdisk** se arranca sobre el dispositivo por defecto (generalmente **hda** o **sda**).

Si se desea especificar otro disco del sistema hay que indicarlo en la línea de comandos:

```
$ fdisk /dev/hdc
```

102.1.1.2 Línea de órdenes de **fdisk**.

Una vez que se ha lanzado **fdisk**, este nos muestra su propia línea de órdenes con la que se interactúa utilizando comandos de una sola letra. Los comandos más utilizados son los que se muestran en la tabla (se pueden visualizar en **fdisk** mediante el comando de ayuda ‘m’)

Algunos comandos de fdisk	
Comando	Acción
p	Muestra información sobre la partición.
d	Borra la partición.
n	Crea una partición.
q	Sale de la aplicación sin guardar los cambios.
w	Guarda los cambios y sale de la aplicación.
m	Muestra el menú de comandos.
v	Verifica la tabla de particiones.
a	Cambia el indicador de estado de arranque de la aplicación.

A continuación se muestran algunos ejemplos de la utilización de comandos de los comandos fdisk:

Para mostrar particiones se utiliza la orden ‘p’, cuyo resultado será similar al siguiente:

```
Disk /dev/hda: 16 head, 63 sectors, 16383 cylinders  
Units = cylinders of 1008 + 512 bytes  
Device Boot Start End Blocks Id System  
/dev/hda1 + 1 4063 2047720+ 83 Linux  
/dev/hda2 4064 4316 127512 82 Linux swap  
/dev/hda3 4317 16383 6081768 83 Linux
```

Se muestra tres particiones (de 1 a 3) en un único disco IDE (hda). La primera partición es la partición de arranque (tiene una cruz en boot) , la segunda de swap, y la tercera es otra partición con el resto de contenidos.

Borrar una partición: comando ‘d’: Una vez introducida la letra d, se ha de introducir el número de partición que se desea borrar. Por ejemplo si se quiere borrar la partición hda3, tras la d se introduciría un 3.

Para crear una partición nueva se utiliza el comando n, y a continuación el sistema nos preguntará si la partición a crear ha de ser extendida o primaria. Hay que tener en cuenta que solamente se pueden crear cuatro particiones primarias.

e- extendida

p- partición primaria (1-4)

Por ejemplo veamos paso a paso como crear una partición primaria, primero se introduce n, p para indicar que es primaria y por último el número de partición

```
Command (m para ayuda): n
Command action
e extended
p primary partition (1-4)
p
Número de partición (1-4): 4
Primer cilindro (10531-16383, default 10351): (aceptar)
Usando el valor de defecto 10351
Último cilindro o +size o +sizeM o +sizeK _
(10351-16383, defecto 16383): (aceptar)
Usando el valor de defecto 16383
```

Para cambiar el tipo de particiones hay que utilizar el comando t, por ejemplo si queremos poner la partición 4 como swap haremos lo siguiente, en primer lugar introducimos una t, y luego el código de la partición de swap que es el 82.

```
Command (m para ayuda): t
Número de partición (1-4): 3
Hex code ( L para la lista de códigos): 82
Cambiado tipo de sistema de la partición 3 a 82 (Linux swap)
```

Si introducimos cuando se pida el hex code se mostrarán todos los sistemas de ficheros posibles. Esta información también se puede ver poniendo i en el menú principal de fdisk.

Después de efectuar todos los cambios se ha de salir para formatear las particiones

Si se guardan los cambio Se nos mostrará un mensaje indicando que la tabla de particiones ha sido modificada y ha de escribirse en el disco. Para utilizar la nueva tabla habrá que reiniciar el sistema.

También se puede descartar los cambios realizados con fdisk y por tanto no se escribirán en la tabla de particiones. Para ello se ha de salir con q. Para grabar grabar los cambios realizados, se deben salvar con w.

102.2. Instalar el gestor de arranque.

Peso en el examen de certificación: 2 puntos.

Objetivo: Seleccionar, instalar y configurar un gestor de arranque

Conceptos y áreas de conocimiento:

- .Proporcionar lugares alternativos de arranque y las opciones de copia de seguridad de arranque.
- .Instalar y configurar un gestor de arranque como GRUB Legacy.
- .Realice los cambios de configuración básicas para GRUB 2.
- .Interactúa con el gestor de arranque.

Términos y utilidades

/boot/grub/menu.lst
MBR
grub-install
superblock

102.2.1. Introducción.

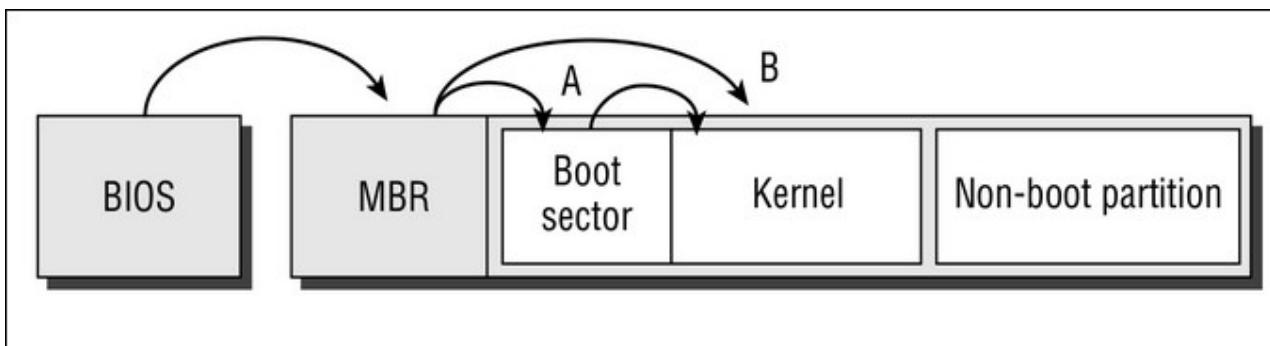
Un disco duro puede contener algo más que particiones y sus contenidos. Una pequeña parte del disco es excepcionalmente importante: el *master boot record (MBR)* contiene la tabla de particiones y el *boot loader* (a veces también llamado *boot manager*, en español, *gestor de arranque*). El gestor de arranque es el software que la BIOS lee y ejecuta cuando el sistema comienza a arrancar. El gestor de arranque es el responsable de la carga del kernel de Linux en memoria y de que este comience a ejecutarse. De hecho, la configuración de un disco duro (o, al menos, de un disco duro de arranque) no está completa hasta que el gestor de arranque está configurado. Aunque las distribuciones de Linux proveen métodos semi-automáticos de configuración de un gestor de arranque durante la instalación del sistema, se debería saber más acerca del tema, particularmente si se recompila el kernel o se necesita preparar una configuración avanzada –por ejemplo, una configuración para seleccionar entre varios sistemas operativos.

En la práctica, hay dos gestores de arranque importantes en Linux: el *Linux Loader (LILO)* (en español, el *Cargador de Linux*) y el *Grand Unified Boot Loader (GRUB)* (en español, el *Magnífico Gestor de Arranque Unificado*).

102.2.1.1. Conceptos previos.

El proceso de arranque de *x86* y *x86-64* puede ser un poco enrevesado, debido en parte a la gran cantidad de opciones disponibles. La figura describe una configuración típica, mostrando un par de rutas de arranque posibles. En ambos casos, el proceso de arranque comienza con la BIOS. A la BIOS se le dice previamente qué dispositivo de arranque debe usar –un disco duro, un disquete, una unidad de CD-ROM, o cualquier otra cosa. Suponiendo que se ha elegido un disco duro como el dispositivo de arranque primario, o si los dispositivos que tienen una prioridad más alta no son *arrancables*, la BIOS carga código desde el MBR. Este código es el código del gestor de arranque primario. En teoría, podría ser cualquier cosa, incluso un (mini) sistema operativo completo.

FIGURA. El sistema de arranque x86 suministra bastantes opciones para redirigir el proceso, pero en última instancia se carga el kernel de un sistema operativo.



En la práctica, el gestor de arranque primario hace una de dos:

- Examinar la tabla de particiones y localizar la partición que está marcada como *arrancable*. A continuación, el gestor de arranque primario carga el sector de arranque de esa partición y lo ejecuta. Este sector de arranque contiene un gestor de arranque secundario, que continúa el proceso localizando un kernel de sistema operativo, cargándolo, y ejecutándolo. Esta opción es descrita en la figura mediante las flechas A.
- Localiza un kernel de sistema operativo, lo carga, y lo ejecuta directamente. Esta opción se salta por completo el gestor de arranque secundario, y se describe en la figura mediante la flecha B.

Tradicionalmente, los sistemas x86 con DOS o Windows siguen el camino A. DOS y Windows 9x/Me se arrancan con cargadores de arranque muy simples que dan muy pocas opciones. Windows NT/200x/XP/Vista proporcionan un gestor de arranque que puede proporcionar una redirección limitada en la segunda etapa del camino A.

Los gestores de arranque de Linux, LILO y GRUB, son mucho más flexibles. Permiten ser instalados tanto en el MBR como en el sector de arranque de una partición de arranque. De hecho, se puede mantener un gestor de arranque primario propio de DOS/Windows y dirigir el sistema para que arranque un kernel desde un sector de arranque (camino A) o saltarse este paso y cargar el kernel directamente desde el MBR (camino B). La primera opción tiene la ventaja de que es poco probable que cualquier otro sistema operativo borre LILO o GRUB, porque está almacenado de forma segura en una partición Linux. Windows tiende a escribir su gestor de arranque MBR estándar cuando es instalado, por lo que, si se necesita reinstalar Windows en un sistema dual (dos o más sistemas operativos), esta acción borrará el gestor de arranque que haya en el MBR. El gestor de arranque permanecerá intacto si está almacenado en el sector de arranque de una partición Linux, aunque Windows podría configurar el sistema para saltárselo. Para reactivar el gestor de arranque de Linux, se debe usar una herramienta como FDISK de DOS/Windows para marcar la partición de Linux como la partición de arranque.

Un inconveniente de situar LILO o GRUB en el sector de arranque de una partición es que esta partición debe ser (como norma general) una partición primaria. (Una excepción es si se usa cualquier otro gestor de arranque en el MBR o en otra partición. Si este gestor de arranque puede redirigir el proceso de arranque a una partición lógica, la restricción anterior desaparece). Por este motivo, mucha gente prefiere ubicar LILO o GRUB en el MBR del disco duro.

En definitiva, ambas opciones funcionan, y, para cuando solo se instala Linux en un sistema, las ventajas e inconvenientes de ambas opciones son mínimas. Algunas distribuciones no dan opción en el momento de instalación. Para ellas, se debería revisar la configuración del gestor de arranque y, cuando se deba añadir un kernel o cualquier otro cambio en el gestor de arranque, modificar la configuración actual en vez de intentar crear una nueva.

LILO y GRUB son mucho más complejos de lo que se muestra en esta guía. Pueden redireccionar el proceso de arranque a sectores de arranque que no sean propios de Linux y, además, son capaces de mostrar menús que ofrezcan la posibilidad de arrancar múltiples sistemas operativos o múltiples kernels de Linux. Se pueden encadenar varios gestores de arranque, incluyendo un tercer grupo de gestores de arranque, como *System Commander* o *BootMagic*. Encadenar gestores de arranque posibilita usar las ventajas propias que cada gestor de arranque provee, tales como la habilidad de *System Commander* de arrancar varias versiones de DOS o Windows en una sola partición.

102.2.2. LILO.

LILO fue, en su momento, el gestor de arranque por defecto de Linux para la arquitectura x86. Aunque GRUB le ha quitado popularidad a LILO, este sigue siendo un gestor de arranque pequeño y útil. Para usar LILO, se debe configurarlo e instalarlo en el sector de arranque. Una vez instalado, se puede reiniciar la máquina y decirle a LILO qué sistema operativo o kernel se quiere arrancar.

102.2.2.1. Configurando LILO.

LILO se configura usando el archivo */etc/lilo.conf*. Este archivo está partido en dos secciones principales: *global* y *por-imagen*. (La segunda se mantiene en secciones conocidas como *stanzas* (en español, estrofas)). Algunas opciones por-imagen están más separadas dependiendo de si son para un kernel Linux o para otro sistema operativo. Para liar más el asunto, muchas de las opciones por-imagen pueden ser usadas en el nivel global para indicar valores por defecto.

102.2.2.1.1. Opciones Globales Esenciales de LILO.

El listado muestra un archivo */etc/lilo.conf* típico, incluyendo opciones globales y tres stanzas para arrancar dos kernels Linux y un sistema operativo distinto a Linux. La mayoría de las líneas de configuración adoptan la forma *opción* u *opción=valor*, pero las líneas que comienzan con almohadilla (#) son comentarios y serán ignoradas. Todo excepto la primera línea de cada stanza está identado de forma tradicional. Esta práctica hace fácil localizar dónde comienza una nueva stanza.

```
# lilo.conf
#
# Global Options:
#
boot=/dev/hda
prompt
timeout=150
default=fedora
lba32
vga=normal
root=/dev/hda5
read-only
#
# Kernel Options (may have multiple):
```

```

#
#stanza 1
image=/boot/vmlinuz-2.6.25
label=fedora
initrd=/boot/initrd-2.6.25
append="mem=2048M"
#stanza 2
image=/boot/bzImage-2.6.26-experimental
label=debian
root=/dev/hda6
#
# Other Operating Systems Options (may have multiple):
#
#stanza 3
other=/dev/hda2
label=dos

```

El archivo *lilo.conf* soporta bastantes opciones. Muchas de estas son un tanto crípticas pero, para alcanzar metas comunes de configuración, sólo se tendrán que usar un puñado:

Ubicación del gestor de arranque La opción *boot=* especifica el nombre del dispositivo que contiene el sector de arranque (*/dev/hda* en el caso del listado anterior). En este ejemplo se usa como sector de arranque el MBR del primer disco duro PATA. En esta configuración, LILO actúa como el gestor de arranque primario –es el primero cargado por la BIOS. Si se quisiera que LILO residiera en una partición en */dev/hda*, se debería proporcionar un identificador de partición específico, por ejemplo, */dev/hda1*. En esta configuración, la BIOS debería cargar un gestor de arranque x86 estándar desde el MBR, el cual debería después cargar LILO.

Stanza por defecto La opción *default=* especifica el kernel o sistema operativo por defecto que arrancará. Si se omite esta opción, se usará la primera imagen listada en *lilo.conf* por defecto.

Mensaje de arranque La linea *prompt* ordena a LILO que muestre el prompt *boot:* y que espere a que el usuario introduzca algo. Esta es una opción poco atractiva, pero puede ser omitida si se quiere que el sistema arranque directamente una configuración simple.

Tiempo de expiración del arranque La opción *timeout=* especifica la cantidad de tiempo, en décimas de segundo, que LILO esperará una entrada desde el teclado antes de arrancar la imagen de un kernel por defecto. Se debe usar la opción *prompt* para habilitar el tiempo de expiración. El listado anterior muestra el valor *150* , indicando un tiempo de expiración de 15 segundos.

Soporte de discos grandes La opción *lba32* habilita a LILO a arrancar desde discos donde la imagen del kernel esté ubicada en una partición que esté más allá del cilindro 1024. Se debe usar esta opción en la mayoría de los casos, pero se puede omitir si se usa un ordenador un tanto antiguo.

Opciones de vídeo La linea *vga=* selecciona el modo texto VGA que se usará cuando arranque. Las opciones son *normal* , *extended* , *ask* y un número. A menos que se tengan problemas con la visualización de vídeo durante el proceso de arranque, esta opción no debería tocarse.

Partición raíz de Linux El valor de la opción *root=* es pasada al kernel de Linux para decirle qué partición debe usarse como partición raíz (/). Se puede ajustar un valor por defecto para esta opción y anularla en las stanzas individuales, como se hace en el listado.

Arranque en modo solo-lectura La opción *read-only* indica que el sistema de archivos raíz debería montarse en modo solo-lectura. Generalmente, el sistema operativo montará el sistema de archivos en modo lectura-escritura.

102.2.2.1.2. Opciones Por-imagen Esenciales de LILO.

LILO admite dos tipos principales de stanzas: las que son para kernels de Linux y las que son para otros sistemas operativos. La opción *image=* es usada para indicar un kernel de Linux, y la opción *other=* es usada para indicar cualquier otro sistema operativo. Cualquiera de las dos opciones sirven para indicar el comienzo de una stanza, y las lineas que le siguen deben ser convenientemente identadas hasta la próxima stanza o el final del archivo. Se pueden ajustar diversos tipos de opciones *por-imagen*:

Imagen de arranque de Linux La línea *image=* indica el archivo kernel de Linux que se usará al arrancar. Se debe suministrar la ruta absoluta al archivo de imagen del kernel de Linux.

Partición de arranque No-Linux Una línea *other=* indica la partición que contiene su propio gestor de arranque. Cuando se selecciona esta opción, LILO pasa el control al gestor de arranque en esa partición. DOS, Windows, OS/2, BeOS, FreeBSD, y otros sistemas operativos pueden tener sus propios gestores de arranque en sus particiones, de modo que esta herramienta (LILO) permite pasar el control a los sistemas operativos citados.

Etiqueta de Sistema Operativo La opción *label=* proporciona un nombre para usar por LILO. Cuando se arranca una máquina, se puede escribir el nombre de la etiqueta o pulsar la tecla *tab* (tabulador) para conseguir una lista de las etiquetas disponibles, como se describe posteriormente en «Interactuando con LILO».

Disco RAM La línea *initrd=* apunta a un disco RAM inicial. Este es un pequeño sistema de archivos en un archivo que el gestor de arranque carga en memoria y entrega al kernel como un substituto basado en RAM de un dispositivo de disco. Las distribuciones Linux utilizan frecuentemente discos RAM como este para almacenar controladores de kernels como manera de mantener pequeño el tamaño del kernel mientras se soportan al mismo tiempo un gran número de controladores. Cuando se construye un kernel propio, generalmente es más sencillo incluir controladores que son necesarios en el proceso de arranque (tales como los necesarios para un disco duro en particular y el sistema de archivos usado) en el archivo principal del kernel.

Opciones extra del kernel Se pueden pasar opciones al kernel a voluntad con la opción *append=*. Por ejemplo, en el listado anterior, la stanza *fedora* pasa la opción *mem=2048M* al kernel de esta forma. (Esta opción le dice a la máquina que tiene 2048MB de memoria RAM. Generalmente, Linux detecta la memoria RAM disponible, pero algunas veces esta u otras opciones son necesarias). Además, algunas de las opciones descritas en la sección «Opciones Globales Esenciales de LILO» pueden ser usadas en las stanzas por-imagen. En particular, *vga=*, *root=* y *read-only* son encontradas comúnmente en stanzas por-imagen. Si no se usan en stanzas, las opciones globales son las que se aplican.

102.2.2.2. Añadiendo un kernel a LILO.

Para añadir un nuevo kernel a LILO, se deben seguir los siguientes pasos como *root*:

1. Abrir el archivo */etc/lilo.conf* con el editor de textos preferido.

2. Copiar una stanza de Linux que funcione.
3. En la stanza copiada, modificar la línea *label*= para darle a la copia un nombre nuevo. El nombre debería ser una cadena de letras, números o ambos, sin espacios.
4. Cambiar la línea *image*= para que apunte al archivo del nuevo kernel.
5. Cambiar cualquier otra opción que necesite ser cambiada. Por ejemplo, si se ha preparado un disco RAM nuevo, cambiar la línea *initrd*= para que apunte a él. Si la nueva configuración no usa un disco RAM, se puede eliminar cualquier línea *initrd*= que esté presente en la stanza .
6. Guardar los cambios y salir del editor de textos.
7. En el *prompt* escribir **lilo**. Este comando instala LILO en el MBR o en el sector de arranque de la partición de arranque. A continuación, se debe ver por pantalla una lista de los nombres de las stanzas . Hay que asegurarse de que la nueva configuración está presente.

Para añadir un nuevo sistema operativo que no sea Linux se debe proceder de igual forma, pero se debería copiar y modificar una configuración *other*= que sepamos que funciona.

Se pueden pasar algunas opciones a **lilo** para modificar lo que hace:

Especificar un archivo de configuración alternativo La opción **-C config-file** especifica un archivo de configuración alternativo que se debe utilizar en lugar de */etc/lilo.conf*.

Comprobar la configuración La opción **-t** comprueba la configuración; no escribe nada en el sector de arranque.

Mostrar abundante información La opción **-v** produce salida por pantalla con abundante información de todo lo que está sucediendo cuando **lilo** se ejecuta.

Especificar un dispositivo de arranque La opción **-b bootdev** especifica un dispositivo de arranque, anulando la opción que esté presente en *lilo.conf*.

102.2.2.3. Interactuando con LILO.

En ocasiones se necesita arrancar con opciones que no están presentes en el archivo */etc/lilo.conf*. Si se ha configurado LILO para suministrar un prompt **boot**: (o **lilo**:), se pueden pasar opciones extra al gestor de arranque en esa línea de comandos.

Algo que se podría hacer es arrancar en modo *single-user* (usuario simple o único usuario, en español). Suponiendo que la imagen que se quiere usar se llama **linux**, se puede escribir algo como esto:

boot: linux 1

También es posible que el programa **init** (generalmente */sbin/init*) se haya corrompido, perdido o configurado erróneamente. En situaciones como esta, es posible especificar un programa **init** alternativo. Una técnica común es usar un intérprete de comandos típico, como por ejemplo **bash**, en lugar de **init**. Se debe hacer en el prompt **boot**: de la siguiente forma:

boot: linux init=/bin/sh

102.2.3. GRUB.

GRUB se ha convertido en el gestor de arranque por defecto para muchas distribuciones Linux porque ofrece numerosas características que le faltan a LILO. Por ejemplo, no se tiene que

reinstalar GRUB después de editar su configuración, y se tienen más opciones interactivas en el momento de arranque. Al igual que con LILO, se debe añadir explícitamente un nuevo kernel a la configuración de GRUB si se tiene intención de usar ese kernel. Si GRUB no está instalado en el sector de arranque, se debe instalarlo, aunque este paso no es necesario si GRUB se encuentra funcionando ya. Al igual que con LILO, se puede interactuar con GRUB durante el proceso de arranque y pasarle opciones para controlar cómo arranca el sistema.

102.2.3.1. Configurando GRUB.

La ubicación habitual para el archivo de configuración de GRUB es `/boot/grub/menu.lst`. Algunas distribuciones (como *Fedora*, *Red Hat* y *Gentoo*) usan el nombre de archivo `grub.conf` en lugar de `menu.lst`. GRUB puede leer su archivo de configuración al arrancar, lo que significa que no se necesita reinstalar el gestor de arranque en el sector de arranque cuando se hagan cambios en el archivo de configuración. Al igual que LILO, el archivo de configuración de GRUB está partido en las secciones *global* y *por-imagen*, teniendo cada una sus propias opciones. Antes de entrar en detalles, se deberían comprender algunas peculiaridades de GRUB.

102.2.3.1.1. Peculiaridades y Nomenclatura de GRUB.

El siguiente listado muestra un archivo de configuración de GRUB de ejemplo. Este archivo es aproximadamente equivalente al archivo de configuración de LILO mostrado anteriormente. En particular, puede arrancar los mismos sistemas operativos usando los mismos kernels –Fedora en `/dev/hda5`, Debian en `/dev/hda6` y DOS en `/dev/hda2`. Fedora y Debian comparten una partición `/boot` (`/dev/hda1`), en donde reside la configuración de GRUB.

```
# grub.conf/menu.lst
#
# Global Options:
#
default=0
timeout=15
splashimage=/grub/bootimage.xpm.gz
#
# Kernel Image Options:
#
title Fedora (2.6.25)
root (hd0,0)
kernel /vmlinuz-2.6.25 ro root=/dev/hda5 mem=2048M
initrd /initrd-2.6.25
title Debian (2.6.26-experimental)
root (hd0,0)
kernel (hd0,0)/bzImage-2.6.26-experimental ro root=/dev/hda6
#
# Other operating systems
```

```

#
title DOS
rootnoverify (hd0,1)
chainloader +1

```

GRUB no se refiere a las unidades de disco mediante el nombre de archivo de dispositivo de la forma en que Linux lo hace. GRUB numera las unidades de disco, de tal forma que en vez de usar `/dev/hda` GRUB usa `(hd0)`. De igual forma, `/dev/hdb` es referenciado con `(hd1)`. GRUB no distingue entre dispositivos PATA, SATA y SCSI, de tal forma que si se dispone de un sistema que solo usa SCSI, el primer dispositivo SCSI es `(hd0)`. En un sistema mixto, los dispositivos ATA reciben por norma general los números más bajos, aunque no siempre es así. El mapeado de dispositivos de GRUB puede encontrarse en el archivo `/boot/grub/device.map`.

Además, GRUB numera las particiones de un dispositivo empezando por `0` en lugar de empezar por `1`, como hace Linux. GRUB separa los números de las particiones de los números de los dispositivos con una coma, como en `(hd0,0)` para la primera partición del primer disco (en Linux, `/dev/hda1` o `/dev/sda1`) o `(hd0,4)` para la primera partición lógica del primer disco (en Linux, `/dev/hda5` o `/dev/sda5`). Las unidades de disquete son referenciadas con `(fd0)`, o posiblemente `(fd1)` o un número más alto si se dispone de más de una unidad de disquete. Los disquetes no tienen particiones por lo que no reciben número de partición.

GRUB define su propia partición raíz, que puede ser diferente de la partición raíz de Linux. La partición raíz de GRUB es la partición donde está el archivo de configuración de GRUB (`menu.lst` o `grub.conf`). Debido a que normalmente este archivo está en el directorio de Linux `/boot/grub/`, la partición raíz de GRUB será la misma que la partición raíz de Linux, a no ser que se use una partición a parte para `/boot` o `/boot/grub`. Si se separa `/boot` en una partición propia, lo que es bastante común, la partición raíz de GRUB será la misma que la partición `/boot` de Linux. Se debe tener en cuenta esta diferencia cuando hagamos referencia a los archivos del directorio de configuración de GRUB.

102.2.3.1.2. Opciones Globales esenciales de GRUB.

La sección *global* de GRUB precede a la sección *por-imagen*. Típicamente, se encontrarán menos opciones en esta sección global que en la sección global de LILO:

Sistema Operativo por defecto La opción `default=` le dice a GRUB qué sistema operativo debe arrancar. En el listado anterior, `default=0` indica que debe arrancarse el primer sistema operativo que aparece en la lista (hay que recordar que el índice que usa GRUB para referenciar a los sistemas operativos comienza por `0`). Si se quiere arrancar el segundo sistema operativo que aparece en la lista, se debe utilizar `default=1` , y así para todos los demás sistemas operativos.

Tiempo de expiración La opción `timeout=` define cuánto tiempo, en segundos, se debe esperar a que el usuario introduzca cualquier valor antes de que se arranque el sistema operativo por defecto. Hay que fijarse en que GRUB mide su tiempo de expiración en segundos, mientras que LILO lo hace en décimas de segundo.

Imagen de fondo La línea `splashimage=` apunta a un archivo de imagen que se muestra como fondo de pantalla en el proceso de arranque. Esta línea es opcional, pero la mayoría de las distribuciones Linux usan una imagen para acentuar el menú de arranque. La referencia al nombre del archivo es relativa a la partición raíz de GRUB, de manera que si `/boot` es una partición separada, esa porción de la ruta se omite. O bien, la ruta puede comenzar con la

especificación de dispositivo propia de GRUB, tal como (*hd0,5*) para referirse a un archivo en esa partición.

102.2.3.1.3. Opciones Por-imagen esenciales de GRUB.

Típicamente, las opciones por-imagen de GRUB son identadas después de la primera línea, parecido a LILO, pero esto es solo una convención, no un requisito de formato de archivo. Las opciones comienzan con una identificación y continúan con las opciones que le dicen a GRUB cómo manejar la imagen a cargar:

Título La línea *title* comienza una *stanza por-imagen* y especifica la etiqueta que se debe mostrar cuando el gestor de arranque se ejecuta. Al contrario que la opción *label* de LILO, la opción *title* de GRUB puede aceptar espacios y es mucho más descriptiva, como se muestra en el listado anterior.

Raíz de GRUB La opción *root* especifica la ubicación de la partición raíz de GRUB. Se trata de la partición */boot* si ésta existe de forma separada de la partición raíz de Linux. Si no, suele coincidir con la partición raíz de Linux (*/*).

Especificación del kernel El parámetro *kernel* define el lugar donde el kernel de Linux está, así como cualquier opción de kernel que se le deba pasar. Las rutas son relativas a la partición raíz de GRUB. Como alternativa, se pueden especificar dispositivos usando la sintaxis propia de GRUB, como, por ejemplo, *kernel (hd0,5)/vmlinuz ro root=/dev/hda5*. La opción *ro* le dice al kernel que monte su sistema de archivos en modo *solo-lectura*, y la opción *root=* especifica el sistema de archivos raíz de Linux. Dado que estas opciones se pasan directamente al kernel, se usan identificadores de dispositivos del estilo de Linux, cuando es necesario, al revés que con otras opciones en el archivo de configuración de GRUB.

Disco RAM de inicio Se debe usar la opción *initrd* para especificar un disco RAM de inicio, como con la opción del mismo nombre en LILO.

Raíz no-Linux La opción *rootnoverify* es similar a la opción *root*, sólo que GRUB no intentará acceder a los archivos de esa partición. Se usa para especificar una partición de arranque para sistemas operativos para los cuales GRUB no puede cargar un kernel, tales como DOS o Windows.

Carga en cadena La opción *chainloader* le dice a GRUB que pase el control a otro gestor de arranque. Normalmente, se pasa la opción *+1* para cargar el primer sector de la partición raíz (generalmente especificado con *rootnoverify*) y transferir la ejecución al gestor de arranque secundario.

Para añadir un kernel a GRUB se deben seguir estos pasos:

1. Como usuario *root*, cargar el archivo *menu.lst* o *grub.conf* en un editor de textos.
2. Copiar una configuración que funcione para un kernel Linux.
3. Modificar la línea *title* para darle a la nueva configuración un nombre único.
4. Modificar la línea *kernel* para que apunte al nuevo kernel. Si se necesita, cambiar las opciones de kernel.
5. Si se está añadiendo, borrando, o cambiando un disco RAM, hacer los cambios apropiados en la línea *initrd*.
6. Si se estima oportuno, cambiar la opción global *default* para que apunte al nuevo kernel.
7. Guardar los cambios y salir del editor de textos.

En este punto, GRUB está configurado para arrancar el nuevo kernel. Cuando se reinicie, el nuevo kernel debería aparecer en el menú, y se debería poder arrancarlo. Si se encuentran problemas,

arrancar una configuración que funcione previamente para depurar el problema.

102.2.3.2. Instalando el gestor de arranque GRUB.

La instalación de GRUB es algo diferente a la de LILO. El comando para instalar GRUB es **grub-install**. También se debe especificar el sector de arranque mediante el nombre de dispositivo cuando se instala el gestor de arranque. El comando básico es

```
# grub-install /dev/hda
```

o

```
# grub-install '(hd0)'
```

Cualquiera de los dos comandos anteriores instalará GRUB en el primer sector (o en el MBR) del primer disco duro. En el segundo ejemplo, el nombre del dispositivo debe ir entre comillas simples. Si se quiere instalar GRUB en el sector de arranque de una partición distinta a la del MBR, se debe incluir un identificador de partición, como en */dev/hda1* o *(hd0,0)*.

No se necesita reinstalar GRUB después de hacer cambios en el archivo de configuración. Solo se necesita instalar GRUB si se hacen ciertos cambios en la configuración de disco, tales como redimensionar o mover la partición raíz de GRUB, mover la instalación por completo a un nuevo disco duro, o, posiblemente, al reinstalar Windows (el cual tiende a borrar los gestores de arranque basados en MBR). En alguno de estos casos, se puede necesitar arrancar Linux usando una copia de seguridad del gestor de arranque, como LILO o GRUB, instalado en un disquete. (Escribir **grub-install /dev/fd0** para crear uno y después etiquetarlo y guardarlo en un lugar seguro).

102.2.3.3. Interactuando con GRUB.

La primera pantalla que el gestor de arranque GRUB muestra es una lista de todos los sistemas operativos que se han especificado con la opción *title* en el archivo de configuración de GRUB. Se puede esperar el tiempo de espiración para que arranque el sistema operativo por defecto. Para seleccionar otro sistema operativo, hay que usar el teclado de flechas para resaltar el sistema operativo que se quiere arrancar. Una vez que la opción está resaltada, presionar la tecla *Enter* para empezar el arranque.

Se deben seguir los siguientes pasos si se quiere cambiar o pasar opciones adicionales a un sistema operativo:

1. Usar el teclado de flechas para resaltar el sistema operativo que más se acerque a lo que se quiere arrancar.
2. Pulsar la tecla **E** para editar esa entrada. Se verá una nueva pantalla que muestra todas las opciones para esa entrada.
3. Usar el teclado de flechas para resaltar la linea con la opción **kernel**.
4. Pulsar la tecla **E** para editar las opciones del kernel.
5. Editar la línea **kernel** para añadir cualquier opción, como por ejemplo **1** para arrancar en modo *single-user* (usuario simple o único usuario, en español). GRUB pasará la opción extra al kernel.
6. Pulsar la tecla **Enter** para completar la edición.
7. Pulsar la tecla **B** para comenzar el arranque.

Se pueden hacer los cambios que se quieran en el paso 5, como usar un programa **init** diferente.

Se puede hacer eso añadiendo **init=/bin/bash** (o cualquiera que sea el programa que se quiera usar) al final de la línea **kernel**.

102.3. Gestión de librerías compartidas.

Peso en el examen de certificación: 1 puntos.

Objetivo: Determinar las librerías compartidas de las que dependen los programas ejecutables e instalarlas cuando sea necesario.

Conceptos y áreas de conocimiento:

- .Identificar las librerías compartidas.
- .Identificar los lugares habituales donde se ubican las librerías del sistema.
- .Carga de librerías compartidas.

Términos y utilidades

ldd
/etc/ld.so.conf
ldconfig
LD_LIBRARY_PATH

102.3.1. Introducción

Dentro de linux, como de cualquier sistema operativo, podemos encontrar dos tipos de programas:

1. *Los programas enlazados estáticamente:* Son programas que contienen todas las funciones de librería en su interior, de tal manera que se bastan solos para poder ejecutarse.
2. *Los programas enlazados dinámicamente:* Estos programas podríamos decir que está “incompletos” en el sentido de que para poder ejecutarse necesitan de una serie de funciones almacenadas en librerías “externas”. Aunque pueda parecer una desventaja que no se puedan ejecutar por sí mismos, esto se compensa con dos ventajas; en primer lugar que los programas en sí son más pequeños y la segunda ventaja es que las librerías pueden ser compartidas entre varios programas, ocupando menos espacio en el disco duro, y también en la memoria cuando el programa se ejecuta. Por este motivo la mayoría de los programas actuales utilizan enlace o vinculación dinámica.

102.3.2. Comando ldd

Para saber si un programa está enlazado dinámicamente, y en el caso de que lo esté, averiguar las librerías que necesita, GNU/Linux nos proporciona el comando **ldd**. Su ejecución es sencilla, basta con aportar como argumento el programa del que deseemos obtener dicha información:

\$ ldd /bin/ls

Si la respuesta es "not a dynamic executable", esto significa que el programa no está enlazado dinámicamente. En caso de sí estarlo la respuesta es distinta, por ejemplo, si ejecutamos la línea de órdenes **\$ ldd /bin/ln** (**ln** es el comando que se utiliza para crear enlaces entre ficheros, un

programa que aparece en todas las distribuciones) el resultado será parecido al siguiente:

```
/bin/ln: libc.so.6 => /lib/tls/libc.so.6 (0x00ebd000) /lib/ld-
linux.so.2 => /lib/ld-linux.so.2 (0x00194000)
```

El comando nos muestra los nombres de las dos bibliotecas (libc.so.6 y ld-linux.so.2) que necesita el comando **ln** y su ruta. La extensión .so indica que se trata de bibliotecas dinámicas. Si hacemos **ls -l** en el directorio **/lib** veremos que realmente estos archivos son enlaces simbólicos a versiones concretas de las bibliotecas que se encuentran en este mismo directorio.

102.3.3. Carga dinámica

Realmente *ld-linux.so*, que parece ser una biblioteca compartida, es un ejecutable con el código necesario para realizar la carga dinámica y leer la información de configuración, y a partir de esta información determinar qué requiere cada programa para poder ejecutarse.

102.3.4. Configuración de bibliotecas compartidas

Si un programa no encuentra una biblioteca que necesita, no podrá ejecutarse y dará un error. Para evitar esto, se puede configurar la variable de entorno **LD_LIBRARY_PATH** de la siguiente manera:

```
$export LD_LIBRARY_PATH=/usr/pathdelbiblioteca:/usr/lib/otropathdelib
```

De este modo se utilizarán las ubicaciones **/usr/pathdelbiblioteca** y **/usr/lib/otropathdelib** para buscar bibliotecas compartidas. Se da por hecho que en dichas ubicaciones están las bibliotecas que el programa en cuestión necesita.

Otra forma de hacer esto es añadiendo las ubicaciones en el fichero **/etc/ld.so.conf**, el cual, almacena una lista de los directorios donde se encuentran las bibliotecas compartidas. Por ejemplo:

```
/usr/lib
/usr/X11R6/lib/Xaw3d
/usr/X11R6/lib
```

El directorio **/lib** no se incluye ya que está incluido por defecto puesto contiene las bibliotecas requeridas por el sistema.

Para aumentar el rendimiento de lectura de bibliotecas existe un fichero caché (**/etc/ld.so.cache**), donde se almacenan todas las bibliotecas de estos directorios. Por tanto, al cambiar el fichero de configuración, el fichero caché debe de ser actualizado mediante el comando **ldconfig**.

102.3. EXTRAS

102.3.Extra Multiarquitectura.

Multiarch es un término usado para referirse a la capacidad del sistema de instalar y correr aplicaciones de diferentes binarios (i386, amd64) en un mismo equipo, por ejemplo ejecutar aplicaciones i386-linux-gnu en una máquina con un sistema base amd64-linux-gnu. Este es el ejemplo más común, pero funciona con otras combinaciones posibles por citar algunos armel y armhf.

Para poder realizar la instalación de paquetes de 32 bits, se debe realizar la instalación completa de la arquitectura, esto es muy sencillo en Debian 7:

```
root@zeus:$ dpkg --add-architecture i386  
root@zeus:$ apt-get update
```

Ya es posible realizar la instalación de paquetes i386.

En Centos se pueden instalar las librerías de 32 bits de esta forma:

```
#yum install glibc.i686
```

102.4. Utilización del sistema de paquetes Debian.

Peso en el examen de certificación: 3 puntos.

Objetivo: Realizar la gestión de paquetes de software usando las herramientas propias de Debian.

Conceptos y áreas de conocimiento:

- .Instalar, actualizar y desinstalar paquetes Debian binarios.
- .Buscar paquetes que contengan archivos o bibliotecas específicas que pueden o no estar instalados.
- .Obtener información del paquete como versión, contenido, dependencias, integridad del paquete y estado de la instalación.

Términos y utilidades

/etc/apt/sources.list
dpkg-reconfigure
apt-cache
dpkg
apt-get
aptitude

102.4.1. Introducción

En esta sección estudiaremos como se gestionan los **paquetes (programas)** en Debian y sus distribuciones derivadas. La forma en que se hará será muy similar a la gestión de paquetes **rpm** que se verá más adelante. Los paquetes, que realmente son ficheros binarios, se llaman así porque no solo contienen el programa que se desea instalar, sino que además incluyen los ficheros (scripts) de configuración, la documentación y sus dependencias.

Es muy importante conocer lo que significa “**dependencias**”, este término pone de manifiesto la necesidad que tienen los programas que otros estén presentes para poder funcionar. Por ejemplo, si pretendemos instalar el paquete “X” que es un entorno gráfico, posiblemente tenga dependencias, es decir, que necesite de otros paquetes sobre los que apoyarse para funcionar correctamente, por ejemplo librerías.

Los nombres de los paquetes Debian tienen la siguiente estructura:

nombre-del-paquete_nombre-del-paquete-build_arquitectura.deb

donde...:

- **nombre-del-paquete:** Será corto y descriptivo, si esta formado por varias palabras suelen estar separadas por guiones.
- **nombre-del-paquete:** Varía en cada revisión y suele ser numérica siguiendo el esquema siguiente: major.minor.patchlevel.
- **build:** Indica la versión del paquete.
- **arquitectura:** Plataforma hardware para la cual fue diseñada la compilación del paquete.

- **.deb**: Es la extensión del fichero que, para los paquetes Debian, es siempre **.deb**

Ejemplos:

ethereal_0.8.13-2_i386.deb; arj_3.10.22-9_i386.deb

102.4.2. Base de datos de los paquetes

La información correspondiente a los paquetes que hay instalados se conserva en una base de datos que suele estar en **/var/lib/dpkg**. Dentro de este directorio tenemos el fichero **/var/lib/dpkg/status** que contiene la totalidad de paquetes conocidos por dpkg con su estado y también el fichero **/var/lib/dpkg/available** que contiene los paquetes que hay disponibles y se pueden instalar. Estos ficheros son muy similares y nos darán bastante información: quién mantiene el paquete, su tamaño, versión, dependencias, descripción, etc. Es muy útil para saber que uso tiene un paquete o ponerse en contacto con su desarrollador.

102.4.3. Herramientas para la gestión de paquetes .deb

El sistema Debian tiene varias herramientas para el manejo de los paquetes, pero los cuatro comandos principales son **dpkg**, **apt-get**, **deselect** y **alien**.

102.4.3.1. dpkg

Esta herramienta permite manejar los paquetes de forma individual tratando directamente con los ficheros **.deb**. Es el núcleo del sistema de empaquetado Debian y equivalente al comando **rpm** en otras distribuciones.

Permite instalar, actualizar, suprimir y gestionar los paquetes. Su uso se suele limitar para forzar instalaciones, arreglar dependencias rotas y el más común, ver los paquetes instalados.

Este comando no gestiona las dependencias. Si al instalar un paquete con **dpkg** faltaran dependencias informará de ello y tendremos que instalarlas previamente o en la misma línea de comando que nos ha dado el aviso. **Un aspecto importante de esta orden es la necesidad de haber descargado previamente los paquetes a instalar.**

La herramienta dpkg usa muchos de los archivos de la carpeta **/var/lib/dpkg** ya mencionada anteriormente. Su sintaxis es:

dpkg orden [paquete/s]

Donde **orden** puede ser:

-i; --install Sirve para instalar el/los paquete/s que se le pase/n como segundo parámetro. (Debe escribirse el nombre completo del paquete). Si el paquete que se pretende instalar tiene dependencias de otros paquetes que no se encuentren instalados el programa lo detectará, y será necesario instalar previamente (o en la misma línea) aquellos abortando la operación.

La opción también se puede usar para actualizar un paquete, pero hay que tener en cuenta que si el paquete estaba instalado lo actualizará si hay una versión más nueva disponible y **si no estaba instalado lo instalará**. Por eso hay que tener cuidado si sólo se quieren actualizar los existentes sin instalar paquetes nuevos.

-r; --remove Elimina el/los paquete/s que se le indique/n como segundo parámetro. No borra los ficheros de configuración para evitar reconfigurar la herramienta si se vuelve a reinstalar. Esta opción también comprueba las dependencias y si el paquete que se pretende desinstalar es una dependencia de otro que esté instalado, abortará la operación.

-P; --purge Elimina todo incluyendo los ficheros de configuración.

-R Esta opción junto con **-i** instala todos los paquetes contenidos en un directorio que se indique.

-l; --list Da el estado de los paquetes Debian que aparecen en la base de datos. Si le añadimos una cadena nos muestra los paquetes que en su nombre contienen esa cadena. Es una opción muy usada que devuelve una sola línea por paquete y permite usar comodines.

En el listado resultante aparecerá a la izquierda de cada paquete unas indicaciones que podrán ser, por ejemplo, “un”, “ii” o “rc”. Esos valores indicarán respectivamente paquete sin instalar, paquete instalado o paquete desinstalado pero del que se conservan sus ficheros de configuración.

-s; --status Da información sobre el paquete que le indicamos a continuación. Si está instalado, tamaño, etc.

-S; --search Busca un paquete de los instalados que contenga la cadena que le indiquemos a continuación. Si en vez de una cadena le pasamos un fichero nos indicará a qué paquete corresponde dicho fichero. Este segundo uso es el más común, encontrar un paquete que contiene un fichero, o lo que es lo mismo, el paquete propietario de un fichero

-L; --listfiles Muestra la lista de ficheros que usa el paquete cuyo nombre indiquemos a continuación, vamos, los ficheros que instala. La mayoría de las veces basta con añadir en nombre del paquete, sin versión, pero a veces pueden existir varias y entonces habría que especificarla en el nombre del paquete.

--configure Ejecuta los scripts de configuración del paquete que indiquemos a continuación.

--get-selections Imprime **sólo los nombres** de los paquetes que estén instalados. Permite comodines.

Ejemplos:

#dpkg -i paquete1.deb paquete2.deb (Instala los dos paquetes indicados. Obviamente los archivos **paquete1.deb** y **paquete2.deb** deben existir en el directorio desde donde se invoca el comando **dpkg**)

#dpkg -r zip (Elimina el paquete ya instalado en el sistema de nombre zip)

#dpkg -P apache (Elimina completamente el paquete apache del sistema, incluyendo los ficheros de configuración, algo que no ocurre con la opción anterior: **-r**)

#dpkg --purge arj (Igual que la anterior pero aplicado al paquete arj)

#dpkg -i -R /var/tmp/packs/ (Instala todos los paquetes que haya en el directorio **/var/tmp/packs/** así como los que pudieran encontrarse en subdirectorios que cuelguen de él a todos los niveles de profundidad)

#dpkg -l (Imprime por la salida estandar la relación ordenada alfabéticamente de todos los paquetes instalados en el sistema)

#dpkg -l apache (igual que la anterior pero sólo la información correspondiente al paquete *apache*)

#dpkg --get-selections xserver* zip* (Imprime por la salida estandar la relación ordenada alfabéticamente de todos los paquetes cuyos nombres coinciden con los patrones proporcionados)

#dpkg -s wawk (Imprime por la salida estandar el estado del paquete proporcionado así como información complementaria de interés)

#dpkg -S /usr/bin basename (Busca entre los todos ficheros de todos los paquetes instalados el archivo proporcionado o patrón de coincidencia)

#dpkg -L coreutils (Imprime por la salida estandar la relación completa de archivos que el paquete

-en este caso coreutils- a depositado en el arbol de directorios después de instalarlo)

Nota: La ejecución y posterior salida de un comando en pantalla puede variar dependiendo de la distribución Linux, pero básicamente contendrá la misma información con distinto formato.

102.4.3.2. Gestor APT (Advanced Packaging Tool)

Con la herramienta anterior, cuando se quiere instalar algo, puede resultar que se necesite a su vez 3 ó 4 paquetes más y, cuando se van a instalar estos, estos segundos necesitan de otros pudiendo la historia convertirse en una auténtico laberinto existiendo incluso dependencias cruzadas. Como ya dijimos antes, **dpkg** instala los paquetes individualmente sólo informando de las dependencias pero no instalándolas. Esta nueva herramienta gestiona esas dependencias instalándose todo lo necesario.

El programa **apt-get** sirve para automatizar la gestión de paquetes en las distribuciones que derivan de Debian. Su principal ventaja es que resuelve él mismo las dependencias y, si quieras instalar un paquete que tiene dependencias y estas a su vez otras resultando que hay que instalar un número determinado de paquetes, nos informará de ello mostrándonos la lista de paquetes que se instalarán, nos consultará si deseamos proseguir y, si se contesta afirmativamente, lo instalará todo. Esta herramienta instala todas las dependencias requeridas (**depends**), pero ignora los paquetes recomendados (**recommends**) y sugeridos (**suggests**) que también puede haberlos.

Para poder controlar todo, **apt-get** en vez de trabajar directamente con paquetes lo hace con **repositorios de paquetes**, que satisfacen las dependencias de manera automática. Leerá una lista de paquetes del repositorio, creará un árbol de dependencias, y determinará que paquetes son requisitos previos obligatorios y que todavía no están instalados. También puede ser que le **sugiera** instalar algún otro paquete.

Estos repositorios contendrán todos los paquetes necesarios y pueden estar en local (en un cd, dvd o directorio), aunque lo más habitual es que sean remotos, estén online y que sean mantenidos y actualizados a diario siendo el inconveniente de esta segunda opción el tiempo de descarga, pero con la gran ventaja de que con un solo comando podemos actualizar todos los paquetes del sistema.

Los paquetes contenidos en los repositorios dependerán unos de otros, o de otros paquetes procedentes de otros repositorios. El sistema **APT** puede gestionar varios repositorios de distintos sitios, y cuando instala un paquete, también instala sus dependencias (si las encuentra).

102.4.3.2.1 El fichero /etc/apt/sources.list

La configuración de los repositorios se encuentra en el fichero **/etc/apt/sources.list** que es el que indica al comando **apt-get** donde coger los paquetes para su instalación.

Un ejemplo de este fichero para una Debian sería:

```
# cat /etc/apt/sources.list
# deb cdrom:[Debian GNU/Linux 7.2.0 _wheezy_ - Official amd64
NETINST Binary-1 20131012-14:04]/ wheezy main
#deb cdrom:[Debian GNU/Linux 7.2.0 _wheezy_ - Official amd64
NETINST Binary-1 20131012-14:04]/ wheezy main
deb http://ftp.es.debian.org/debian/ wheezy main
deb-src http://ftp.es.debian.org/debian/ wheezy main
deb http://security.debian.org/ wheezy/updates main
deb-src http://security.debian.org/ wheezy/updates main
```

```
# wheezy-updates, previously known as 'volatile'  
deb http://ftp.es.debian.org/debian/ wheezy-updates main  
deb-src http://ftp.es.debian.org/debian/ wheezy-updates main
```

La primera línea, que se encuentra comentada, indica el dispositivo desde donde se instaló nuestro Linux y podría interesar quitar el comentario para actualizar desde el CD/DVD. El resto de líneas indican fuentes desde donde se puede recuperar información y paquetes al sistema local.

Posteriormente habría que usar el comando **apt-get update** para sincronizar la información que figura en su base de datos local con las fuentes especificadas en el fichero. La sincronización debería de realizarse siempre antes de instalar o actualizar un paquete y también después de modificar */etc/apt/sources.list*. También se recomienda poner los recursos más rápidos al inicio del fichero. Se pueden añadir comentarios poniendo al principio de la línea el símbolo #.

102.4.3.2.2 apt-get

La sintaxis del comando es:

apt-get [opción/es] orden [paquete/s]

La **orden** podrá ser:

update Actualiza la base de datos, solo la lista de paquetes y versiones disponibles, desde los repositorios. Se suele utilizar esta opción después de modificar el fichero sources.list.

install Instala el paquete que indiquemos a continuación. Si el paquete ya estuviera instalado lo actualiza. Ejecute **apt-get update** antes de actualizar paquetes para asegurarse de que la base de datos local muestre las últimas versiones disponibles. Si el nombre del paquete va seguido de – (menos) indica que en vez de instalarlo hay que desinstalarlo.

Cuando se ordena la instalación de un paquete, el apt-get revisa primero si ya fue descargado, si no lo fue, irá al primer recurso del sources.list a buscar la versión más nueva del programa, y si este tiene dependencias se añadirán a la lista de instalación.

remove Borra el paquete que indiquemos a continuación. Hay que tener en cuenta que borra el paquete indicado pero no los paquetes que tuvieron que instalarse como requisito previo a aquel aunque no vayan a ser necesarios según nuestro árbol de dependencias. Esta orden avisa de esta eventualidad pero no borra esos paquetes. También indica los paquetes no necesarios que pudiera haber de antes. Tampoco borra los archivos de configuración del paquete.

autoremove Eliminará los paquetes que le indiquemos junto con sus dependencias que no vayan a ser necesarias para los paquetes que queden instalados en el sistema. Incluirá también aquellas dependencias innecesarias para el sistema aunque no hallan sido instaladas por el/los que se pretenden eliminar. La opción **remove** junto con **--auto-remove** es equivalente.

Si se usa autoremove sin ningún nombre de paquete, los paquetes que no se estén usando y que se instalaron como dependencias de otros se eliminarán de su sistema de forma automática.

purge Junto con el nombre de un paquete lo desinstala y borra sus archivos de configuración.

upgrade Actualiza **todos** los paquetes instalados de los que haya nuevas versiones disponibles. También es muy conveniente ejecutar apt-get update antes de su uso.

clean Elimina todos los archivos de paquetes descargados y que estén en la cache.

Cuando apt-get instala un programa, guarda una copia del fichero deb en los directorios */var/cache/apt/archives* y */var/cache/apt/archives/partial*. Con el tiempo esos directorios pueden llegar a ocupar mucho espacio, para limpiar ambos directorios se usa esta opción.

check Actualiza la cache y verifica las dependencias.

Entre las **opciones** tenemos las siguientes:

-f Intenta corregir dependencias rotas.

-d Descarga un paquete pero no lo instala.

-s Simula la ejecución de la orden pero no la realiza. Se usa por ejemplo para ver si un paquete depende de otros paquetes o que implicaría un upgrade.

-y Responde a todo que si, cuidado con esta opción.

-h Muestra la ayuda.

Ejemplos:

#apt-get update (Actualiza la cache local con la de los repositorios de referencia indicados en /etc/apt/sources.list)

#apt-get install vim-gtk (Instala el paquete vim-gtk y todas sus dependencias)

#apt-get install tzdata (Instala el paquete tzdata y todas sus dependencias)

#apt-get install libqt* (Instala todos los paquetes cuyos nombres coinciden con el patrón así como las dependencias que necesitan)

#apt-get install -s gcl (Instala el paquete gcl de forma simulada, no produce ningún efecto real en el sistema)

#apt-get install -d gcl (Descarga un paquete sin instalarlo y luego podrá visualizar la información del paquete con **dpkg --info**. Generalmente los archivos descargados están en **/var/cache/apt/archives/**)

#apt-get remove -s gcl (Elimina el paquete gcl de forma simulada, no produce ningún efecto real en el sistema)

#apt-get clean (Borra totalmente el repositorio local que contiene los ficheros de los paquetes descargados)

102.4.3.2.3 apt-cache

Busca información sobre paquetes en nuestro sistema, en la caché local. Puede utilizar expresiones regulares.

Su sintaxis es:

apt-cache orden paquete/patron_de_busqueda

Veamos sus opciones más comunes:

search Busca, por su nombre o comentario, un paquete en la base de datos local APT.

show Muestra la descripción del paquete.

Ejemplos:

#apt-cache search torrent

#apt-cache search “linux loader”

#apt-cache show torrent

102.4.3.3. dselect

Es una interfaz gráfica (front-end) del comando **dpkg** que gestiona las dependencias y los conflictos. Normalmente no suele estar instalada y por tanto se necesita usar el comando anterior para hacerlo. La orden completa sería **apt-get install dselect** y luego se ejecutaría con **dselect**.

Por defecto, al arrancarlo selecciona automáticamente todos los paquetes "Requeridos" ("Required"), "Importantes" ("Important") y "Estándar" ("Standard"), pero podemos configurar otros.

```
Interfaz de manejo de paquetes dselect de Debian `1.16.1.2 (amd64)'.

* 0. [M]étodo Escoger el método de acceso que se usará.
  1. [A]ctualiza Actualizar la lista de paquetes disponibles, si se puede.
  2. [S]elección Solicitar qué paquetes desea en el sistema.
  3. [I]nstalar Instalar y actualizar los paquetes deseados.
  4. [C]onfigura Configurar los paquetes que no estén configurados.
  5. [D]esinstalar Desinstalar los paquetes no deseados.
  6. sa[L]ir Salir de dselect.

Utilice ^P y ^N, las teclas del cursor, letras iniciales, o dígitos;
Pulse <intro> para confirmar la selección. ^L redibuja la pantalla.

Copyright (C) 1994-1996 Ian Jackson.
Copyright (C) 2000,2001 Wichert Akkerman.
Esto es software libre; vea la Licencia Pública General de GNU versión 2 o
posterior para las condiciones de copia. No hay NINGUNA garantía.
```

102.4.3.4. alien (Convertidor de paquetes)

Al igual que **dselect** esta herramienta a veces no está instalada, por lo que la instalaríamos previamente con **apt-get install alien**.

Este comando permite convertir paquetes que no son de Debian sino de Redhat (rpm), Stampede (slp), Slackware (tgz), Solaris (pkg) o genéricos a Debian y viceversa. Lo que hace es generar una salida lo más parecida a Debian y siempre será preferible usar paquetes originales para esta distribución, pero a veces no hay más remedio porque no exista el paquete buscado en los repositorios.

La sintaxis del comando es:

alien opción/es paquete

Las opciones típicas son:

-d o --to-deb Es la opción por defecto. Convierte un paquete rpm a formato debian (.deb).

-r o --to-rpm Para convertir un paquete a formato RPM.

-t o --to-tgz Para convertir un paquete a formato tgz.

--to-slp Para convertir un paquete a formato slp.

-i o --install Instala el paquete tras crearlo.

-c o --scripts Incluye en la conversión los scripts de pre y postinstalación.

-h o --help Muestra la ayuda.

--description="Comentario" Pone una descripción al paquete creado.

Ejemplos:

```
#alien -d lgtoclnt-7.4-1.i686.rpm
```

```
#alien -d --scripts lgtoclnt-7.4-1.i686.rpm
```

```
#alien --scripts -d lgtoclnt-7.4-1.i686.rpm
```

```
#alien -d wget.rpm
```

```
#alien -d lgtoclnt-7.4-1.i686.rpm
```

```
#alien --scripts -d lgtoclnt-7.4-1.i686.rpm
```

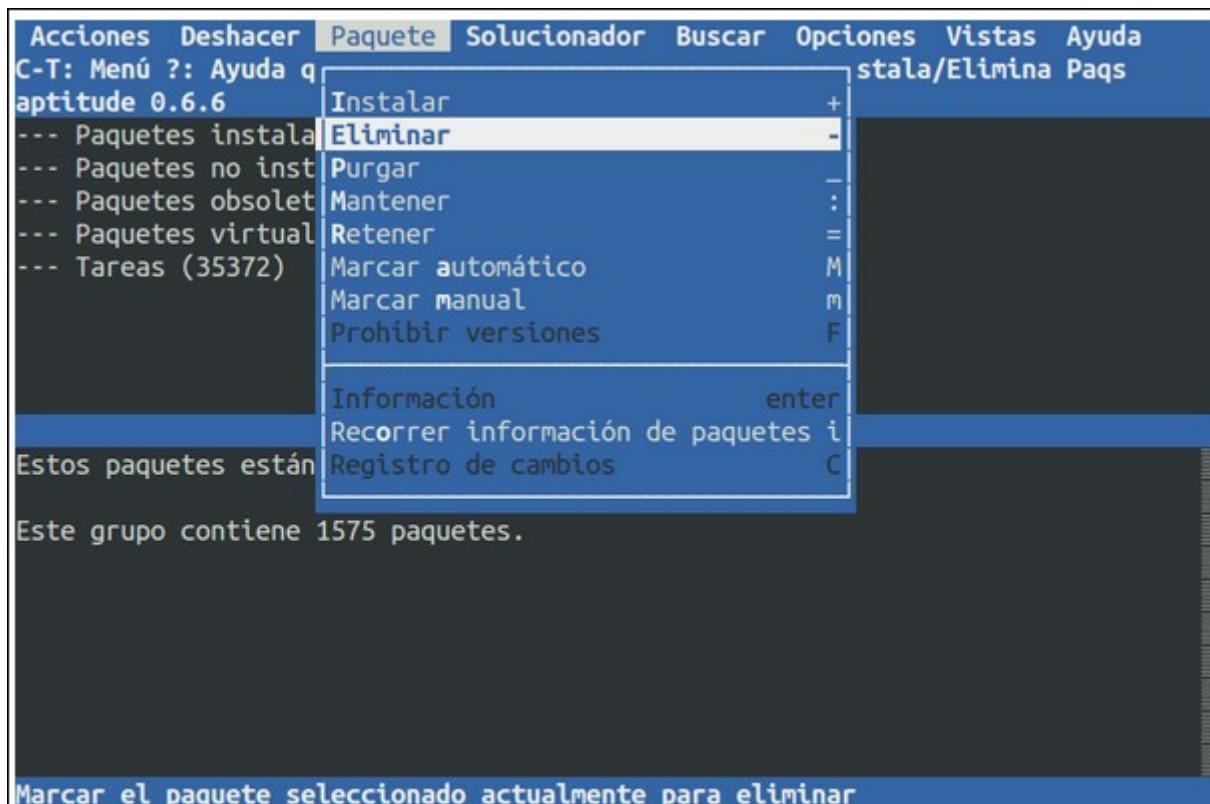
Para este comando debe indicarse el nombre completo del paquete con su versión, arquitectura, etc. Además hay que decir que una conversión simple comprobará las dependencias, pero no incluirá los scripts de pre y postinstalación. Para esto hay que especificar el parámetro **--scripts**.

102.4.3.5. aptitude

Si no está presente la instalaríamos con **#apt-get install aptitude**.

El comando aptitude ofrece una interfaz para las funciones de gestión de APT. Se puede usar para instalar o eliminar paquetes y controlar los indicadores de estado que muestran, por ejemplo, si se deberían actualizar o conservar en su estado actual. Es como una evolución del dselect pero mucho más amigable.

Utilice **Enter** para expandir o comprimir las ramas de selección y llegar hasta los paquetes, y **ctrl-t** para acceder a la barra de menú.



Una "i" en la columna izquierda indica que en el estado actual se hace necesario instalar el paquete.

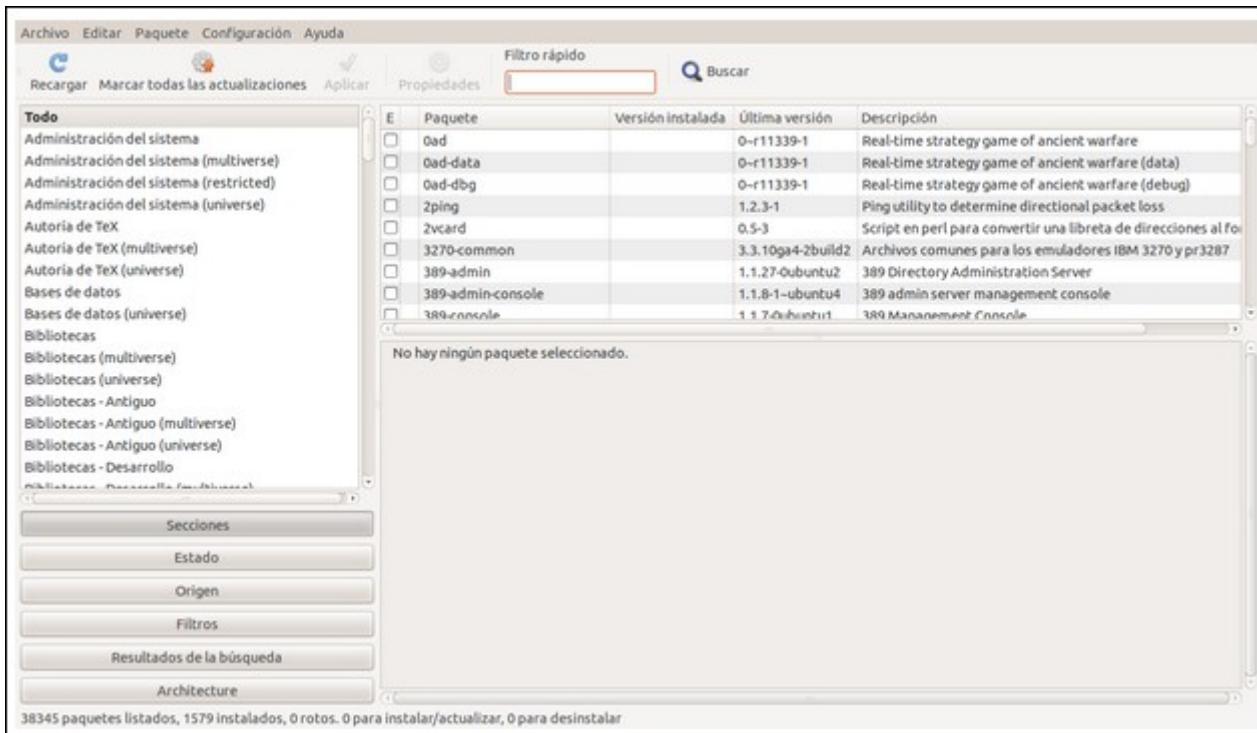
La tecla "?" le ofrecerá ayuda en cualquier momento y "/" le permitirá buscar paquetes. Por ejemplo, si tecleamos "/arj" la búsqueda nos llevaría a ese paquete o algún otro con nombre similar, por ejemplo, arj-doc, si es el caso pulse "n" (next) para ir al próximo resultado. Luego podrá usar "+/-" o el menú **Package** para marcar/desmarcar el paquete para su instalación.

Cuando termine, seleccione **Acciones -> Instalar/Eliminar paquetes** (o presione la tecla "g") para aplicar los cambios. También se puede usar la opción quit (q) para salir sin aplicarlos.

102.4.3.6. synaptic

Además de aptitude existen otras interfaces de gestión de paquetes para los sistemas Debian. Una de ellas es synaptic que está preparada para el X Window System y es un front-end, una interfaz gráfica, que llama a las funciones de APT. Tiene todas las funciones de APT y, además, disfruta de una interfaz muy atractiva.

El botón **Aplicar** instalará los paquetes que se marquen y actualizará todos los paquetes que sea necesario actualizar. El botón **Reload** (Recargar) actualizará la lista de paquetes.



102.4 EXTRAS

102.4 EXTRAS apt-cache

La suite de APT incluye el programa apt-cache que ofrece información sobre la base de datos del paquete Debian (conocida como la “caché del paquete”). Con este comando se permite:

- Mostrar la información del paquete #apt-cache showpkg <nombre_paquete>
- Mostrar las estadísticas de los paquetes #apt-cache stats
- Localizar dependencias no satisfechas: #apt-cache unmet
- Mostrar las dependencias: # apt-cache depends <nombre_paquete>
- Localizar todos los paquetes: #apt-cache pkgnames
- Encontrar en los repositorios información sobre algún paquete: #apt-cache search <cadena>

102.4. EXTRAS portage

Portage es el gestor de paquetes oficial de la distribución de Linux Gentoo y también el de Google Chrome OS.

Portage (implementado en Python y Bash) está inspirado en los Ports BSD, aunque implementa ciertas características avanzadas que no están presentes en éstos: gestión de dependencias, afinamiento preciso de los paquetes a gusto del administrador, instalaciones falsas , entornos de prueba durante la compilación, desinstalación segura, perfiles de sistema, paquetes virtuales, gestión de los ficheros de configuración y múltiples ranuras para distintas versiones de un mismo paquete.

Portage dispone de un árbol local que contiene las descripciones de los paquetes de software, así

como los scripts necesarios para instalarlos. Este árbol se puede sincronizar con un servidor remoto mediante una orden:

```
emerge --sync
```

Cuando un paquete de software es seleccionado para ser instalado, Portage descarga los archivos con el código fuente y los compila en ese momento, generando los archivos ejecutables y documentación correspondiente. Es posible especificar las optimizaciones que emplear en la compilación, así como utilizar una variable llamada *USE* que indica la compatibilidad con otros programas.

La posibilidad de indicar las optimizaciones y el parámetro *USE* permiten crear una distribución a medida. De todas formas, Portage también soporta la instalación de binarios, ya sean paquetes precompilados por el mismo sistema o paquetes que se encuentran exclusivamente en formato binario.

Portage permite mantener el software actualizado y controlar las versiones que se encuentran instaladas, proporcionando unas posibilidades similares a las de APT de Debian. Así, por ejemplo, es posible actualizar todos los paquetes instalados a la última versión estable sin necesidad de intervención del usuario:

```
emerge --update world
```

Como se puede comprobar, la herramienta para la administración de paquetes en gentoo es emerge, que tiene un completo manual.

102.5. Utilización del sistemas de paquetes RPM y YUM.

Peso en el examen de certificación: 3 puntos.

Objetivo: Realizar la gestión de paquetes de software usando las herramientas propias de Debian.

Conceptos y áreas de conocimiento:

.Instalar, volver a instalar, actualizar y eliminar paquetes con RPM y YUM.

.Obtener información sobre paquetes RPM, como versión, estado, dependencias, integridad y firmas.

.Determinar qué archivos proporciona un paquete, así como encontrar qué paquete de un archivo específico está contenido en él.

Términos y utilidades

rpm

/etc/yum.conf

yum

rpm2cpio

/etc/yum.repos.d/

yumdownloader

El gestor de paquetes RPM es típico de las distribuciones GNU/Linux basadas en Red-Hat, como por ejemplo Fedora, CentOS, Suse, Mandriva, etc. Para nuestros ejemplos usaremos Fedora.

El sistema RPM se compone de:

- Ficheros de paquetes (*.rpm)
- La base de datos RPM
- La herramienta rpm

102.5.1. Los paquetes RPM

Los ficheros RPM contienen los paquetes de software compilado, preparado para ser instalado, y que contienen lo siguiente:

- Ficheros comprimidos de la aplicación
- Nombre y versión del paquete
- Fecha, host y autor de la construcción del paquete
- Descripción
- Información de integridad y verificación (MD5 checksum)
- Dependencias

El esquema de nombrado de los paquetes RPM es el siguiente: **package-version-patch.architecture.rpm**. Donde:

- package: nombre de la aplicación
- version: número de versión de la aplicación
- patch: número de compilación del paquete.
- architecture: arquitectura para la que ha sido compilado el paquete.

Por ejemplo el paquete "*ethereal-0.8.9-1.i386.rpm*", contiene la versión 0.8.9 del programa Ethereal, y es la primera compilación para la plataforma i386 (Intel PC).

102.5.2. Comando RPM

El comando rpm lo usaremos para:

- Instalar, actualizar, desinstalar y eliminar paquetes
- Preguntar información a la base de datos RPM
- Verificar paquetes y comprobar los ficheros instalados
- Construir ficheros binarios a partir de código fuente

El comando rpm realiza las siguientes comprobaciones:

- Asegurarse de que existe espacio suficiente en disco para el paquete.
- Comprobar que los ficheros existentes no serán sobreescritos.
- Que se encuentran todas las dependencias.

Veamos un ejemplo de uso del comando rpm:

1. Accedemos a un terminal del sistema
2. Escribimos **rpm -q zsh** para verificar que el paquete zsh no está instalado.
3. Montamos el DVD de instalación del Fedora 16, y hacemos **cd /media/Fedora\ 16\ i386\ DVD/Packages/**
4. Si hacemos **vdir** veremos los ficheros de los paquetes RPM que contiene el DVD que hemos puesto.
5. Obtenemos privilegios de root, mediante el comando **su**.
6. Escribimos **vdir zsh***
7. Escribimos **rpm -qpi zsh-4.3.11-2.fc15.i686.rpm** para instalar el paquete zsh desde el fichero rpm. Con esto veremos información del paquete.
8. Escribimos **rpm -ivh zsh-4.3.11-2.fc15.i686.rpm** para instalar el paquete, y mostrar una barra de progreso con almohadillas.
9. Escribimos **rpm -q zsh** para verificar que está instalado el paquete.
10. Si escribimos **zsh** entraremos a una nueva consola. Escribir **exit** para salir.

Las principales opciones del comando rpm son las siguientes:

- **-i** o **--install**: Instala el paquete.
- **-U** o **--update**: Actualiza o instala el paquete.
- **-F** o **--freshen**: Actualiza el paquete solamente si estuviera instalado.
- **-V** o **--verify**: Verifica el tamaño, MD5, permisos, tipo, integridad, etc.
- **-q** o **--query**: Investiga paquetes y archivos.
- **-e** o **--erase**: Desinstala el paquete.

El comando rpm debe ejecutarse con alguna de las opciones principales, para determinar de este modo qué funcionalidad del comando queremos invocar. Además podemos acompañar a las opciones principales de otras subopciones que modifican la manera en que actúa la opción principal.

Las principales subopciones del comando rpm son las siguientes:

- **a**: aplica una variable a todos los paquetes instalados.

- c: Con la opción principal "q", lista los archivos de configuración.
- d: Con la opción principal "q", lista los archivos de documentación.
- f: Con la opción principal "q", verifica qué paquete instaló el referido archivo.
- h: Muestra el progreso del procedimiento solicitado.
- i: Con la opción principal "q", lista información sobre un determinado paquete.
- l: Con la opción principal "q", lista todos los archivos y directorios del paquete.
- p: Con la opción principal "q", indica que la investigación se realiza en el archivo .rpm.
- v: Modo más descriptivo.

Otras opciones importantes del comando rpm son:

- --nodeps: Instala el paquete sin verificar las dependencias.
- --force: Fuerza la instalación/actualización.
- --test: Muestra cómo sería la instalación, pero no instala.
- --requires: Con la opción principal "q", muestra las exigencias para el paquete especificado.
- --whatrequires: Con la opción principal "q", muestra qué programas dependen del paquete.

102.5.3. Base de datos RPM

La información de los paquetes instalados se guarda en una base de datos. Esta base de datos se guarda en el directorio `/var/lib/rpm`. Cada vez que se usa el comando rpm se consulta la información de esta base de datos.

Normalmente no nos preocuparemos de cómo se guarda la información en la base de datos porque esto se hace de forma automática.

Si tenemos errores extraños instalando o desinstalando paquetes, puede ser debido a una base de datos inconsistente o dañada. Si se corrompe dicha información, podemos reconstruir la base de datos con el comando siguiente:**rpm --rebuilddb**.

102.5.4. Comando YUM

Para poder instalar paquetes RPM con el comando **rpm** es necesario que previamente hayamos localizado y descargado en nuestro equipo el paquete .rpm en cuestión. Las distribuciones de GNU/Linux tiene unas repositorios accesibles desde Internet con todos los paquetes de la propia distribución.

El comando YUM es similar al comando **apt-get** de Debian, haciéndonos más fácil el trabajo. Por ejemplo, para instalar el paquete **zsh** con el comando YUM hacemos **#yum install zsh** y tenemos los siguiente:

- El comando yum se encarga de localizar el fichero rpm que necesitamos buscando en los repositorios de la distribución.
- Una vez localizado el fichero, lo descarga a nuestro equipo.
- Una vez descargado lo instala.
- Si el paquete tuviera dependencias con otros paquetes, automáticamente yum se encarga de localizar los ficheros rpm de las dependencias de los repositorios.
- Descargar los ficheros de las dependencias a nuestro equipo.
- Instalar los paquetes de las dependencias.

El fichero de configuración de YUM es `/etc/yum.conf`. Veamos un ejemplo

Algunas de las opciones de este archivo son:

- **cachedir**: Directorio de almacenamiento de los paquetes y otros archivos de datos. El estándar es `/var/cache/yum`.

- **keepcache**: Valor 1 ó 0. Determina si yum debe mantener los paquetes y archivos relacionados después de una instalación exitosa. El estándar es 1.
- **reposdir**: Lista de directorios donde yum buscará los archivos .repo que definen los repositorios. El estándar es `/etc/yum/repos.d`
- **debuglevel**: Nivel de mensajes de aviso.
- **errorlevel**: Nivel de mensajes de error.
- **logfile**: Fichero de log de yum. Donde se registran los mensajes generados por yum.
- **gpgcheck**: Valor 1 ó 0. Determina si yum debe o no realizar la verificación GPG de los paquetes.

Por defecto, el directorio donde YUM descarga los paquetes rpm es `/var/cache/yum`.

Los archivos .repo definen repositorios de Internet desde donde se pueden descargar los paquete. Y tienen el siguiente formato:

```
[Identificador]
name=Nombre descriptivo del repositorio
baseurl=url:::/camino/para/el/repositorio
```

- [Identificador]: Término único que identifica cada repositorio.
- name: Texto de descripción del repositorio.
- baseurl: URL del directorio donde se encuentra el directorio *repodata* del yum.
- enable: Valor 1 ó 0. Determina si debe utilizarse el repositorio.
- gpgcheck: Valor 1 ó 0. Determina si debe realizarse la verificación GPG para los paquetes de este repositorio.

Opciones más comunes del comando yum:

- **yum search paquete**: Localiza un determinado paquete.
- **yum install paquete**: Instala el paquete.
- **yum remove paquete** o **yum erase paquete**: Desinstala el paquete.
- **yum provides recurso** o **yum whatprovides recurso**: Localiza qué paquete, instalado o no, ofrece un determinado recurso.
- **yum update**: Actualiza los paquetes desactualizados que estén instalados.
- **yum upgrade**: Similar a update, pero se utiliza para actualizar la distribución a una versión más actual.
- **yum info paquete**: Es similar a **rpm -qi paquete**

Existe otro comando llamado **yumdownloader**, similar a yum pero solamente localiza y descarga el paquete RPM de los repositorios sin instalarlo. Con la opción "--source", descarga el código fuente del paquete en lugar del programa compilado.

Existen herramientas de instalación para el entorno gráfico como **yumex**, **kyum**, etc.

102.5.5. Validar la integridad/firma de un paquete

Los paquetes incluyen una firma que podemos usar para verificar su autenticidad.

Con Fedora podemos hacer lo siguiente:

- **rpm --import /usr/share/rhn/RPM-GPG-KEY-FEDORA**, para incorporar las claves a la base de datos rpm.
- **rpm --checksig paquete** o **rpm -K paquete**, para verificar si es correcta la firma del paquete.
- **rpm -V paquete**, para verificar la integridad de un paquete instalado.
- **rpm -Va**, para verificar la integridad de todos los paquetes instalados.

Los caracteres de verificación que se muestran son los siguientes:

- .(punto): Prueba exitosa.
- ?: La prueba no puede realizarse.
- M: El permiso o el tipo de archivo ha cambiado.
- 5: La suma MD5 del archivo es diferente.
- D: El dispositivo se modificó.
- L: El enlace simbólico se modificó.
- U: El dueño del archivo ha cambiado.
- G: El grupo del archivo ha cambiado.
- T: La fecha del archivo ha cambiado.

102.5 EXTRAS

102.5 EXTRAS YAST y Zypper

En derivados de SUSE (como puede ser openSUSE), YaST es la herramienta de sistema que, entre otras muchas cosas, te permite gestionar el software de la distribución con facilidad.

Los apartados que hay dentro de este módulo de gestión de software de YaST son:

- **Añadir productos:** Añadir productos adicionales.
- **Comprobación de medios:** Si surge un problema con la instalación y utiliza un medio de instalación CD o DVD, debería comprobar si el medio esta defectuoso. La comprobación verifica la suma de control MD5.
- **Actualización en línea:** Permite actualizar los paquetes instalados a las últimas versiones.
- **Búsqueda de paquetes:** Habilita la utilización del buscador de paquetes Webpin. Permite buscar entre todos los repositorios del build-service de openSUSE y los repositorios de la comunidad conocidos.
- **Gestión de software:** Permite instalar, actualizar y desinstalar paquetes de forma individual o en grupos, buscar paquetes individuales, consultar por repositorio, grupo o patrón.
- **Repositorios de software:** Uso de distintas fuentes de repositorios, y posibilidad de añadir o eliminar fuentes desde el propio módulo.

Tiene una interfaz gráfica y es muy sencilla. Desde la línea de comandos presenta un sistema de menú, desarrollado en ncurses, muy intuitivo.

La herramienta zypper es la interfaz de línea de comandos para la bibliotecas de gestión del sistema. Zypper se puede utilizar para instalar, actualizar o quitar software, gestionar los repositorios, realizar consultas, y mucho más. Utilizar el comando zypper requiere privilegios de root, por lo que se ha de ingresar al modo root antes de ejecutar cualquiera de los comandos zypper.

Uso

La estructura básica del comando zypper es así:

```
zypper GLOBAL_OPTIONS COMMAND COMMAND_OPTIONS
```

Aunque parece un poco complejo no lo es. Se analizarán las diversas tareas que llevará a cabo con el comando zypper.

Instalación de software

Instalación de software con zypper es simple. El comando completo se vería así:

```
zypper install SOFTWARE
```

Donde SOFTWARE es el nombre del software que deseas instalar.

```
zypper install wireshark
```

Eliminación de software

La estructura de comando para borrar el software se vería así:

```
zypper remove SOFTWARE
```

ejemplo:

```
zypper remove wireshark
```

Búsqueda de software

Es posible usar zypper para buscar la base de datos en caché para los candidatos de instalación de este modo:

```
zypper search wireshark
```

Zypper informará de todas las posibles aplicaciones para la cadena de búsqueda.

Actualización con zypper

Actualización con zypper es increíblemente fácil. Ejecuta el comando `zypper update` y si hay actualizaciones disponibles para el sistema, serán reportadas y solo es necesario escribir "y" (sin comillas) y pulsar Intro para ejecutar todas las actualizaciones.

Actualización de la Distribución con zypper

Si se desea una actualización de la distribución con zypper se podría usar el comando `dup` de este modo:

```
zypper dup
```

103 COMANDOS GNU Y UNIX.

- 103.1. Trabajar en la línea de comandos.
- 103.2. Procesar cadenas de texto por medio de filtros.
- 103.3. Administración básica de archivos
- 103.4. Flujos, tuberías, y redireccionamiento de salida.
- 103.5. Crear, monitorizar y finalizar procesos.
- 103.6. Modificar la prioridad de ejecución de un proceso.
- 103.7. Buscar en archivos de texto, utilizando expresiones regulares
- 103.8. Edición básica de archivos con el vi

103.1. Trabajar en la línea de comandos.

Peso en el examen de certificación: 4 puntos.

Objetivo: Interactuar con shells y comandos usando la línea de órdenes. Se asume que la shell para el aprendizaje de este apartado es bash.

Conceptos y áreas de conocimiento:

- . Uso de comandos simples y líneas de órdenes sencillas para realizar tareas básicas.
- . Uso y modificación del entorno de shell incluyendo la definición, la referencia y la exportación de variables de entorno.
- . Utilización y edición de historial de comandos.
- . Invocar comandos dentro y fuera del path definido.

Términos y utilidades

.

- echo
- exec
- pwd
- unset
- bash
- env
- export
- set
- man
- history

OBJETIVO: Trabajar de forma eficaz con la línea de comandos. Incluye la escritura de comandos válidos y las secuencias de comandos, usando substitución, y aplicando comandos recursivos a través de un árbol de directorios.

Una vez que te has logado dentro del sistema Linux, te enfrentas con el shell. Éste aparece simplemente como una interface de línea de comando. El sistema presenta la shell con un **prompt**, que suele ser simplemente un carácter como \$ o #. El prompt nos dice que está preparado para aceptar comandos, que pueden ocupar una o más líneas de texto.

A partir del prompt del shell escribimos comandos que son interpretados por el shell y enviados al sistema. El shell que estés corriendo en ese momento configura su prompt correspondiente. Aunque el shell inicial era **sh** (por su creador Stephen Bourne), la mayoría de los sistemas Linux tienen como predeterminado el shell **bash** (Bourne Again Shell), una variante suya.

103.1.1.- Conceptos básicos sobre la shell

Durante la ejecución, *bash* mantiene un conjunto de *variables de shell* que contienen información importante para la ejecución de la shell. La mayoría de ellas se inicializan cuando *bash* se inicia, pero pueden ser inicializadas o cambiadas en cualquier momento.

La primera variable de interés es **PS1**, que contiene el contenido visible de la línea de comandos actual en el momento en el que está aceptando comandos (hay también **PS2** usada cuando se requieren múltiples líneas. Se puede ver el contenido de PS1 o de cualquier otra variable simplemente usando el comando **echo** con el nombre de la variable precedido por el símbolo **\$**.

```
$ echo $PS1  
\$
```

La salida **\\$** indica que PS1 tiene dos caracteres: **** y **\$**. La barra invertida **** indica a la shell que no debe interpretar el símbolo siguiente, en este caso el dólar, de ninguna forma especial.

Un carácter dólar es el prompt por defecto para *sh*, pero *bash* permite más opciones para que sea más informativo. Generalmente, el prompt está configurado para mostrar el nombre de usuario, nombre del servidor y directorio actual de trabajo en el prompt. Un ejemplo de este prompt sería:

```
[angie@redhat /etc]$
```

En este ejemplo *angie* es el nombre de usuario, *redhat* es el nombre del servidor y */etc* es el directorio actual de trabajo (llamado **pwd** – present working directory). El prompt para la shell de *bash* es el símbolo **\$**. El prompt se configura a través del archivo **/etc/bashrc**. Por medio de este archivo se puede cambiar la configuración de lo que se muestra en el prompt. Cada usuario puede personalizar el prompt a su gusto, creando para ello el archivo de configuración personalizado en **~/.bashrc** (el carácter **~** equivale al directorio home del usuario)

Generalmente, nuestro prompt estará configurado como lo que hemos visto arriba, cuya línea de configuración sería la siguiente:

```
[\u@\h \w]\$
```

Cada uno de los caracteres precedidos por una barra invertida **** tiene un significado especial para el *bash*, como ya veremos más adelante.

Los comandos escritos en la línea de comandos no son enviados al sistema hasta que no se haya pulsado la tecla *ENTER*. Esto permite editar los comandos antes de que se pasen al sistema.

Los comandos escritos en la línea de comandos deben seguir una sintaxis específica. La primera palabra es el comando que se debe ejecutar; esta puede incluir una ruta absoluta, que es la ruta completa al comando, que debe terminar con el nombre del comando. A continuación deben figurar las opciones que son usadas por el comando. Los argumentos del comando van a continuación de las opciones. Cada uno de estos elementos debe separarse en la línea de comandos por medio de un espacio en blanco. El formato de un comando es el siguiente:

```
$ ls
```

Las opciones son códigos de una letra precedidos por un guion (-), y modifican la acción del comando. Se pueden combinar y usar varias opciones con un mismo comando. Las opciones son sensibles a mayúsculas y, en muchas ocasiones, una letra minúscula significará una opción diferente que su correspondiente mayúscula. Una opción muy importante disponible con muchos comandos es la **-R**, que especifica que el comando se debe ejecutar de forma recursiva a través del árbol de directorios.

Recordatorio de examen: La opción para la función recursiva de un comando es muy importante y

aparecerá en el examen.

Si lo que quieras es listar los archivos y los directorios, y que los directorios aparezcan seguidos de una barra inclinada (/), se puede usar la opción -F (prueba el resultado).

```
$ ls -F
```

Ahora suponed que deseamos mostrar el contenido del directorio /etc desde otra localización. Además queremos que se muestre una barra inclinada tras todos los nombres de directorios que estén dentro de /etc. El directorio /etc se puede utilizar como argumento para el comando ls. Un argumento es otra opción que se puede utilizar con un comando. En el siguiente ejemplo, el argumento es un directorio que será examinado por el comando. Sería algo así:

```
$ ls -F /etc
```

Otro carácter especial que se puede utilizar cuando se introducen comandos es la barra invertida (\). Cuando se introduce a la derecha, antes de pulsar la tecla Enter, la barra invertida permite extender el comando a lo largo de varias líneas. La barra invertida provoca que el sistema ignore la pulsación de la tecla Enter, y trata todos los comandos como si estuvieran en una única línea. Esta función puede ser útil si tecleamos comandos muy largos, para poder partirlas en varias líneas. Un ejemplo sería el siguiente:

```
$ ls -F /etc \ ls -F /usr
```

En el mundo real: Para ejecutar comandos muy largos se utilizan scripts, en lugar de teclearlos directamente sobre la línea de comandos. Sin embargo, es importante recordar el uso de la barra invertida para el examen.

Otras opciones que se pueden usar con el comando ls pueden ser:

```
$ ls -l
```

Devuelve la lista de archivos y directorios en formato extenso, incluyendo nombre, privilegios de acceso, propietario, tamaño, fecha de última modificación, etc.

```
$ ls -a
```

Muestra todos los archivos y directorios del pwd, incluyendo los archivos ocultos.

```
$ ls *.gif
```

Muestra todos los archivos del pwd que terminen por ".gif".

```
$ ls ga*
```

Muestra todos los archivos y directorios del pwd que comiencen por "ga".

103.1.2. Completado de comandos.

El shell bash incluye una característica denominada completado de comandos. Esto nos permite teclear las primeras letras de un comando, pulsar la tecla Tabulador, y dejar que el sistema complete el comando por nosotros. Si queremos ejecutar el comando dmesg para mostrar el buffer del kernel, podríamos teclear:

```
$ dm
```

y pulsar la tecla Tabulador, de forma que el sistema completará el comando:

```
$ dmesg
```

Si existe más de una coincidencia con la cadena tecleada antes de pulsar la tecla Tabulador, el sistema hará sonar un **beep**. Pulsando de nuevo la tecla Tabulador se mostrarán todas las

posibilidades que coincidan con lo tecleado. Pulsar la tecla Esc dos veces produce el mismo efecto que pulsar la tecla Tabulador.

103.1.3. Conectando varios comandos.

En todos los ejemplos utilizados hasta ahora utilizamos la tecla Enter para informar al sistema de que el comando debía ser procesado. Sin embargo, no estamos limitados a ejecutar un único comando de cada vez. Podemos ejecutar varios comandos, sin que estén conectados entre sí de ningún modo, tecleándolos en la misma línea y separándolos por punto y coma (;). Por ejemplo, es posible listar todos los archivos del directorio actual y la fecha de hoy tecleando:

```
$ ls ; date
```

El punto y coma es un carácter especial que siempre significa que hay varios comandos en la misma línea. Debido a que esto último tiene un sentido global, podemos prescindir de los espacios en blanco a ambos lados del punto y coma para obtener el mismo resultado (ls;date).

Si lo que hacen los comandos tiene algo en común, la salida de uno de ellos se convertirá en la entrada del siguiente entonces podemos conectarlos usando una tubería (|). Por ejemplo, si la lista de archivos de un directorio es muy larga como para poder verla en una sola pantalla, podemos ver una pantalla de cada vez usando:

```
$ ls -l | more
```

De este modo, la salida del comando ls -l será la entrada del comando more. Si falla la primera parte de la línea de comando, no se podrá ejecutar la segunda.

103.1.4. Comodines.

Los comodines son caracteres que se utilizan en lugar de otros caracteres que el sistema rellena. Los dos comodines más frecuentes son el asterisco * y la interrogación ?. Aunque en ocasiones se confundan, su significado es diferente y producirán resultados totalmente distintos. El asterisco significa ninguno, alguno o todos los caracteres:

```
$ ls s*
```

Este comando mostrará todas las entradas (archivos o directorios) dentro del directorio actual que comiencen con la letra s, y que tengan cualquier número de caracteres a continuación (incluyendo ninguno). Un posible resultado del comando puede ser:

```
s sa sam samp sampl sample samples samples.gif
```

Hay que prestar atención a que el comando encuentra la s sola y la s seguida de cualquier número de caracteres a continuación. En contraste, la interrogación (?) es un contenedor para un y sólo un carácter. Utilizando las mismas posibilidades que antes, el comando:

```
$ ls s?
```

Encontrará entradas (archivos y directorios) dentro del directorio actual que comiencen por la letra s y que únicamente tengan una letra más. El resultado sería:

```
sa
```

Si quisieramos encontrar las entradas que comiencen por s y cuyo nombre tenga 5 caracteres en total, utilizaríamos:

```
$ ls s????
```

En resumen, el asterisco significa todos o ninguno, y el interrogante siempre significa uno. Estos

dos comodines no son excluyentes, de modo que se pueden combinar según las necesidades. Por ejemplo, para encontrar sólo los archivos que tengan una extensión de tres letras dentro del directorio actual, utilizaremos:

```
$ ls *.???
```

Para complicar un poco más las cosas también podemos utilizar los corchetes ([]) para especificar posibles valores. Todos los valores posibles deben estar dentro de los corchetes, y el shell los tratará individualmente:

```
$ ls [de]*
```

Este ejemplo encontrará todas las entradas que comiencen por d o por e y que contengan un número ilimitado de caracteres. Para encontrar las entradas de longitud de 3 caracteres que comiencen por d o por e, utilizaremos:

```
$ ls [de]??
```

El número de caracteres que podemos incluir dentro de los corchetes es teóricamente ilimitado. Sin embargo, si lo que queremos es encontrar todas las entradas que comiencen por una letra minúscula pero no por un número u otro carácter, podemos utilizar [abcdefghijklmnoprstuvwxyz]. Debido a que esto es un rango, una forma mucho más simple de obtener el mismo resultado es poniendo:

```
$ ls [a-z]*
```

Los rangos no tienen que ser series completas de números o caracteres, podemos expresar subconjuntos de ellos. Por ejemplo, si queremos buscar entradas que comiencen por alguna letra entre la d y la t, podemos utilizar indistintamente [defghijklmnopqrst] o [d-t]. Si la entrada puede comenzar por esas letras tanto en mayúsculas como en minúsculas, podemos usar [DEFGHIJKLMNOPQRSTdefghijklmnopqrst] o [D-Td-t]

Otros ejemplos son:

- Todas las letras (mayúsculas y minúsculas): [A-z] (que es lo mismo que [A-Z] más [a-z])
- Todos los números: [0-9]
- Cualquier carácter que no sea un número: [!0-9]
- Cualquier carácter que no sea una letra: [!A-z]

103.1.5. Path y otras variables .

Cuando tecleas un comando en el prompt la shell primero busca entre sus comandos internos y de no encontrar el comando se buscará una utilidad (comando) externa con ese mismo nombre. Esto se realiza buscando en los directorios incluidos en la variable **PATH** y en el mismo orden en el que han sido definidos en dicha variable hasta que se encuentra el primer archivo ejecutable cuyo nombre coincide con el tecleado, en caso de no encontrarse ninguno después de buscar en todos los directorios indicados aparecerá el mensaje (“*command not found*”).

A continuación se remarcan algunas cosas importantes que deben conocerse respecto al path:

Puedes consultar el valor actual de **PATH** con el siguiente comando:

```
echo $PATH
```

En el path no se incluye por defecto el directorio actual, por tanto, podrías tener un fichero ejecutable en tu directorio, ver que está ahí (tecleando ls) pero al escribir su nombre obtener un error de “*command not found*”. Para solucionar esto podrías escribir el pathname completo del fichero a ejecutar, añadir este directorio a la variable PATH, mover el fichero a un directorio incluido en dicha variable, o añadir el símbolo del directorio actual (.) a la variable PATH.

Las distintas entradas en el path irán separadas por dos puntos (:).

El orden de búsqueda en la variable PATH debería comenzar por los directorios donde se encuentran los comandos más comunes (los directorios bin) y terminar por los directorios donde se encuentren los comandos del usuario, si existiesen.

Para añadir un directorio al path puedes redefinir la declaración completa o, simplemente, añadir el nuevo directorio con el comando:

```
$ PATH=$PATH:{nuevo directorio}
```

Por tanto, para añadir el directorio /home/fulanito al path, el comando sería:

```
$ PATH=$PATH:/home/fulanito
```

Si quieras añadir una entrada tal que el directorio donde estás trabajando en ese momento esté siempre incluido en el path de búsqueda, escribe:

```
$ PATH=$PATH:./
```

Por motivos de seguridad te recomendamos que no hagas ésto último, pero si no tienes más remedio que hacerlo, colócalo siempre al final de la declaración de PATH como se comentó anteriormente y no al principio.

103.1.6. Variables comunes.

Podemos ver el valor de cualquier variable existente con el siguiente comando:

```
$ echo ${nombre_de_variable}
```

Por tanto, el comando:

```
$ echo $MAIL
```

mostrará el directorio de correo, \$HOME, el directorio home, y así sucesivamente. Para obtener una lista completa de todas las variables definidas en el entorno puedes utilizar cualquiera de estos dos comandos: env y set.

Aunque las salidas de los mismos puedan variar ligeramente (variables del entorno contra variables locales), en su mayor parte la salida de env es un subconjunto de la salida de set. Algunas de las variables que podemos visualizar son:

- **HOME**: El directorio donde inicias la sesión y a donde llegas si tecleas cd sin ningún parámetro adicional.
- **LINES**: El número de líneas de pantalla que se visualizarán entre cada pausa (comando more).
- **LOGNAME**: El nombre de usuario con el que has conectado.
- **PWD**: El directorio de trabajo actual, el directorio donde te encuentras en este momento.
- **SHELL**: El intérprete de comandos que estás utilizando.
- **TERM**: El tipo de terminal o emulación seleccionado.

Como regla general, las variables del sistema siempre aparecen en mayúsculas.

Puedes cambiar el valor de las variables según lo necesites o añadir tus propias variables para que sean utilizadas por otros programas o desde la línea de comando. Por ejemplo para crear una nueva variable llamada HOY, solo tendrías que escribir:

```
$ HOY=Viernes
```

Ahora puedes ver el valor de esa variable escribiendo:

```
$ echo $HOY
```

El resultado es Viernes. Si ahora empleas el comando:

```
$ set
```

la variable aparecerá ahí, sin embargo si usas el comando:

```
$ env
```

no aparecerá. La variable ha sido creada localmente y solo podrá ser referenciada localmente. Para que sea accesible por subrutinas o procesos hijos debes utilizar el comando export que hará que pase de ser local a ser una variable del entorno:

```
$ export HOY
```

Esto pondrá la variable en el entorno donde podrá ser encontrada tanto localmente como por las subrutinas y procesos hijos, la variable y su valor serán accesibles durante toda la sesión y se perderán una vez desconectes.

Para que el valor sea permanente debes añadir la definición a un perfil, por ejemplo puedes cambiar el valor de PATH para todos los usuarios del sistema, conectándote como root y editando el archivo */etc/profile* modificando la línea donde se define la variable PATH. Ésto podrías hacerlo utilizando el editor vi con el siguiente comando:

```
$ vi /etc/profile
```

Si quisieses hacer la modificación solo para un usuario en particular tendrías que editar el archivo */home/nombre_del_usuario/.bash_profile* (el punto al inicio del nombre del fichero indica que este es un fichero oculto, tendrías que teclear ls -a para poder verlo). Las modificaciones en los perfiles solo se harán efectivas tras la desconexión y nueva conexión del usuario. Para cambiar el valor de una variable, simplemente, vuelve a definirla con el nuevo valor:

```
$ HOY=Lunes
```

Como ya la habíamos exportado anteriormente ya no tenemos por que hacerlo otra vez y el nuevo valor quedará accesible tanto localmente como en el entorno. Si fuese necesario eliminar una variable puedes utilizar el comando unset.

103.1.7. Alias de comandos.

Aunque el sistema operativo y la shell ofrecen multitud de comandos y utilidades, puedes crear alias con nombres que tengan más sentido para ti o que sean más pequeños y así teclear menos caracteres. Por ejemplo, si estás familiarizado con la línea de comandos de Windows o del DOS estarás acostumbrado a escribir dir para obtener una lista de ficheros de un directorio, para obtener lo mismo en Linux se escribiría ls -l. Podrías crear un alias de tal forma que cada vez que escribieses dir se ejecutase ls -l, utilizando la siguiente expresión:

```
$ alias dir="ls -l"
```

La sintaxis será siempre el alias seguido del comando que habrá de ejecutarse cuando se teclee el alias separados por el signo igual (=). En raras ocasiones podrías hacerlo sin encerrar el comando entre comillas, pero como norma general siempre deberías incluirlas.

¡Ojo! Para que los alias no se pierdan al finalizar la sesión deben añadirse al archivo .bashrc que se encuentra en el directorio home del usuario.

103.1.8. Configurando el prompt.

Entre las variables presentes y definibles, están aquellas que definen el prompt. El prompt es el mensaje que visualiza la shell cuando está lista para recibir un comando.

Los prompts por defecto incluyen:

- \$ El último carácter para sh, bash, y ksh
- % El último carácter para csh y zsh
- > El último carácter para tcsh

El prompt primario puede ser tanto la variable PS1 o la variable prompt, dependiendo de que shell estés utilizando. En bash, un valor típico para PS1 sería:

```
[\u@\h \w]\$
```

Componente por componente, PS1 sería igual a lo siguiente:

- El corchete izquierdo (])
- El nombre del usuario actual (\u)
- La arroba (@)
- El nombre del host actual (\h)
- Un espacio
- El directorio de trabajo actual (\W)
- El corchete derecho (])
- El signo del dolar (\$)

Un ejemplo de este prompt sería:

```
[leko@pentimVIII home]\$
```

La barra invertida () indica la utilización de un valor especial. Algunos de estos valores se muestran en la tabla 3.1.1

Tabla 3.1.1

Valores en el uso de la barra invertida

Valor	Resultado
\d	Fecha actual
\h	Nombre del host hasta el primer punto
\n	Interlínea
\s	Shell
\t	Hora actual
\u	Nombre usuario
\W	Directorio actual

\w	Ruta completa de directorios
!	Número del historial (se comentará mas adelante)
#	Número del comando
\\$	Prompt por defecto—\$ para los usuarios normales y # para root
\`	Barra invertida literal
ABC	Texto libre
\`	Secuencia de caracteres no imprimibles
\`	Fin de secuencia de caracteres no imprimibles
\$date	Salida del comando date (o cualquier otro)

Para modificar el prompt de forma permanente para todos los usuarios hace falta editar el fichero */etc/bashrc* (como root) y modificar su contenido. Teniendo en cuenta que si un usuario tiene en su directorio personal el fichero *~/.bashrc* se sobrescribirá el contenido en */etc/bashrc*.

Si te fijas en las variables del sistema verás que, además de **PS1**, puedes encontrar **PS2**.

Anteriormente se comentó que podía terminarse una línea de comandos con una barra invertida para indicarle a la shell que aun no se había terminado de introducir todo el comando, si observamos un ejemplo podemos ver lo siguiente:

```
[leko@pentimVIII home]$ ls -l *.gif \
> *.fig \
> *.bmp
```

Observa como el prompt cambió de lo indicado en **PS1** a un signo de mayor (>). Si hubiese seguido siendo el mismo **PS1** no podrías saber si se trata del mismo comando o de uno nuevo, por eso se cambia del prompt primario a otro secundario definido, precisamente por la variable **PS2**. Su valor podrá ser cambiado de la misma forma que el de **PS1**, incluyendo los valores especiales de la Tabla 3.1.1. La mayoría de las shells admiten tres o cuatro niveles de prompts.

103.1.9 Otras variables

A estas alturas ya te habrás dado cuenta de que el signo del dolar (\$) se utiliza para obtener el valor de una variable; si tienes una variable llamada **EJEMPLO**, puedes ver su contenido examinando **\$EJEMPLO**.

Hay otras tres variables que pueden resultar prácticas para determinar tu entorno.

La primera—\$\$—muestra el ID del proceso correspondiente a la shell que se ejecuta actualmente:

```
$ echo $$
```

La segunda—\$\$?—muestra el resultado del último comando ejecutado. Este resultado puede ser correcto (**0**) o incorrecto (**1**). Por ejemplo, el comando **ls** admite la opción **-F** que diferenciará entre ficheros y directorios insertando una barra invertida detrás del nombre de los directorios, sin

embargo este comando no incluye la opción **-z**.

Dada esta información, en el ejemplo siguiente podremos comprobar la utilización de la variable \$?

```
[leko@pentimVIII home]$ ls -F  
Desktop\ sample snapshot01.gif snapshot02.gif  
[leko@pentimVIII home]$ echo $?  
0  
[leko@pentimVIII home]$ ls -z  
ls: invalid option - z  
[leko@pentimVIII home]$echo $?  
1
```

La tercera variable—\$!—mostrará el ID del último proceso hijo que se inició en el background . Si no se hubiese iniciado ningún proceso de background, entonces la variable no tendría valor.

Los procesos se describen con mayor detalle en otro capítulo pero para esta explicación es suficiente con saber que poniendo un ampersand (&) al final de un comando le indicamos a la shell que ejecute dicho comando en el background:

```
[leko@pentimVIII home]$ echo $!  
[leko@pentimVIII home]$ ls -F &  
[leko@pentimVIII home]$ echo $!  
19321  
[leko@pentimVIII home]$
```

103.1.10. Usando el historial de comandos.

El archivo del histórico contiene una lista de los comandos introducidos en la línea de comando. La variable **HISTSIZE** define el número de comandos que se almacenarán en dicho archivo durante la sesión actual, esta variable puede estar definida tanto en */etc/profile* como en *~/.profile*) y su valor por defecto es de 1000 entradas. El comando *history* muestra todas las entradas del archivo del histórico que se guarda en *~/.bash_history*.

Puedes navegar por las entradas del histórico con las flechas del teclado. La flecha hacia arriba mostrará las entradas anteriores, mientras que la flecha hacia abajo avanzará hacia adelante en el histórico. De esta forma podemos ahorrarnos volver a escribir comandos que ya habíamos escrito con anterioridad. Mientras navegamos por los comandos podemos editarlos antes de ejecutarlos de nuevo.

La variable **HISTCMD** proporciona el índice dentro del histórico comando que se está ejecutando. La variable **HISTFILE** especifica el nombre del fichero que contendrá el histórico (*~/.bash_history* por defecto). La variable **HISTFILESIZE** especifica el máximo número de líneas que contendrá el fichero especificado en **HISTFILE** y, aunque suele tener el mismo valor que **HISTSIZE**, podría ser diferente ya que ésta última se refiere solo al histórico de la sesión actual y no al tamaño total del archivo histórico.

```
$ fc
```

La utilidad *fc* proporciona otra opción para editar los comandos del fichero histórico antes de ejecutarlos. La utilidad *fc* abre el editor de textos por defecto con el comando especificado y ahí podemos editarla y salvarla antes de ejecutarlo disponiendo de toda la potencia de un editor de textos. Podemos llamar a *fc* utilizando como parámetro el número del comando que queremos editar o, también, con un rango de comandos para, de esta forma, editarlos y ejecutarlos en conjunto.

También es posible especificar el editor de textos a utilizar. Una vez que se ha llamado a fc el fichero de comandos puede ser editado como cualquier otro fichero de texto y los comandos editados se ejecutarán al salir del editor. La opción -l se utiliza para mostrar una lista de los valores especificados a continuación, podéis escribir, por ejemplo, fc -l 50 60 y obtendréis la lista de los comandos del 50 al 60 en el historial.

Tabla 3.1.2

Operadores para el manejo del historial	
Operador	Descripción
!!	También conocido como bang-bang,[1] este operador hace referencia al comando más reciente del historial.
!n	Hace referencia al comando número n del historial. Puedes utilizar el comando history para conocer estos números.
!-n	Hace referencia al comando actual menos n en el historial.
!cadena	Hace referencia al comando más reciente que comience por cadena.
!?cadena	Hace referencia al comando más reciente que contenga cadena.
^ cadena1^cadena2	Sustitución rápida. Repite el último comando reemplazando la primera aparición de cadena1 por cadena2.

[1] Es común llamar bang al signo de admiración en los sistemas Linux y Unix.

Al trabajar con Linux habrá ocasiones en las que necesites una información más amplia sobre la utilización de algún comando, utilidad o configuración del sistema. Aunque este y otros libros pueden ser muy útiles, ningún libro puede tener la información totalmente actualizada.

Afortunadamente existen magníficos recursos para cuando necesitemos más información. Algunas de estas fuentes de información las podremos encontrar en nuestro sistema local, mientras que otras estarán disponibles en Internet. Este capítulo te informará sobre algunos de los lugares más útiles para buscar información. Este conocimiento te hará ahorrar mucho tiempo cuando trabajes con sistemas Linux y es esencial para la preparación del examen.

103.1.11. Obteniendo ayuda con las páginas 'man'.

Uso y administración de la documentación del Sistema Local. El uso y administración de los recursos de man y del material de */usr/doc/*. Incluye el encontrar las páginas man más relevantes, buscar por las secciones de man, encontrar comandos y las páginas man relativas a éstos, configurar el acceso a las fuentes de man y al sistema man, utilizar la documentación del sistema almacenada en */usr/doc/* y otros lugares relacionados y determinar qué documentación mantener en */usr/doc/*.

Las páginas del manual o páginas man de Linux son el mejor lugar para resolver cuestiones relativas a la sintaxis y opciones de los comandos y utilidades del sistema. Los documentos de las páginas man están almacenados en un formato comprimido. El comando man descomprime y formatea estas páginas para su correcta visualización. Se accede a estas páginas utilizando el

comando man seguido del nombre del comando que se quiere consultar. Un ejemplo de la sintaxis te este comando es el siguiente:

```
# man ls
```

Este comando buscará todas las páginas del manual relativas al comando ls. Cuando abras las páginas del manual lo primero que se visualizará es un banner con el comando y la página man que está siendo consultada. También se muestra aquí el logo FSF de Free Software Foundation.

Para desplazarse hacia abajo por las páginas man se utiliza la barra espaciadora, que avanzará una página por cada pulsación. La tecla Q provoca la salida del proceso de visualización de la página. Si quieras buscar algún texto dentro de la página man puedes utilizar las expresiones regulares, a continuación se muestra un ejemplo de como buscar la palabra option.

```
/option
```

103.1.11.1 Encontrando las páginas man.

Las páginas man se almacenan en el sistema. La variable MANPATH indica la ubicación de estos archivos. Por defecto las páginas man se guardan en los siguientes lugares.

- /usr/man/man1
- /usr/man/man2
- /usr/man/man3
- /usr/man/man4
- /usr/man/man5
- /usr/man/man6
- /usr/man/man7
- /usr/man/man8
- /usr/man/man9

El significado de los números se comentará en la siguiente sección del capítulo, “Buscando secciones de las páginas man”

Pregunta de Examen: Asegurate que sabes la ubicación por defecto de los ficheros fuentes de las páginas man. Esta pregunta es muy probable que salga en el examen.

El usuario puede especificar un MANPATH diferente. Esto permitiría utilizar un conjunto diferente de páginas man. Esto es práctico porque algunos comandos podrían almacenar sus páginas man en lugares distintos a los estándar. El comando man admite distintas opciones, una de ellas permite usar un path distinto al indicado en MANPATH. Las opciones del comando man se muestran en la tabla 3.1.3.

Tabla 3.1.3

Opciones del comando *man*

Opción	Uso
-C fichero-de- configuración	Indica un fichero de configuración distinto a /etc/man.conf
-M path	Indica en que directorios se buscarán las páginas man.

-P paginador	Indica el paginador o el programa utilizado para formatear y visualizar las páginas man. El paginador por defecto es el indicado en la variable de entorno PAGER Los paginadores more y less son los más frecuentemente utilizados.
-S Lista-de-secciones	Indica una lista de las secciones a buscar separadas por dos puntos (:)
-a	Indica que han de mostrarse todas las entradas coincidentes y no solo la primera.
-c	Indica que la página fuente ha de ser reformateada.
-d	Indica que debe mostrarse información de debug en lugar de las páginas man.
-f	Indica que el programa man debe comportarse como el programa whatis.(se explicará mas adelante).
-h	Muestra información sobre el comando man.
-k	Indica que el programa man debe comportarse como el programa apropos.(se explicará mas adelante).
-K	Busca una cadena especificada en las páginas man. Por cada entrada encontrada se le pregunta al usuario si desea verla.
-m	Indica un conjunto alternativo de páginas man basado en el sistema especificado.
-w	Indica que ha de visualizarse el path de las páginas man en lugar de las páginas.

A continuación se muestra un ejemplo del uso de la opción -a. Ésta hace que las páginas coincidentes sean mostradas en el orden en el que han sido encontradas. En primer lugar se le muestra al usuario la entrada correspondiente a crontab en la sección uno. Cuando el usuario pulsa la tecla Q para salir de ésta página , se mostrará la entrada encontrada en la sección cinco.

```
$ man -a crontab
```

La opción -w es práctica para encontrar la ubicación de las entradas de las páginas man. Si utilizásemos esta opción con la utilidad crontab obtendríamos lo siguiente:

```
$ man -w crontab
/usr/man/man1/crontab.1.gz
```

Pregunta de Examen: Asegurate que conoces las opciones de búsqueda y sus funciones, entre ellas -a, -K, y -k.

103.1.11.2. Buscar secciones en las páginas man.

La información de las páginas *man* de Linux están contenidas en un conjunto de archivos. Estos archivos están agrupados en secciones y cada sección contiene un tipo específico de información. La Tabla 3.1.4 lista éstas secciones y su uso:

Tabla 3.1.4	
Secciones de las páginas <i>man</i>	
Sección	Uso
1	Comandos y aplicaciones del usuario.
2	Llamadas del sistema y errores del Kernel.
3	Llamadas a librerías.
4	Drivers de dispositivos y protocolos de red.
5	Formatos estándar de archivos.
6	Juegos y demos.
7	Ficheros y documentos misceláneos.
8	Comandos de administración del sistema.
9	Especificaciones e interfaces oscuros del kernel.

Cuando se le pasa un argumento al comando *man*, éste busca dentro de las secciones siguiendo un orden específico y se retorna la primera coincidencia. El orden de búsqueda por defecto es el siguiente:

1, 8, 2, 3, 4, 5, 6, 7, 9.

También es posible indicar la sección en la que quieras buscar. Si quisieses buscar información sobre la utilidad *crontab* en la sección cinco utilizarías el siguiente comando:

```
# man 5 crontab
```

Si usas la opción -a podrás examinar todas las entradas coincidentes. Siguiendo el ejemplo de la utilidad crontab tendrías que utilizar el siguiente comando:

```
# man -a crontab
```

103. 1.11.3 Buscando con whatis.

La utilidad whatis no se menciona específicamente en los objetivos del examen, pero conocer su uso puede venir bien en el examen. La utilidad whatis se usa para buscar entradas coincidentes en la base de datos whatis. Esta base de datos se crea utilizando el comando `/usr/bin/makewhatis`. Esta base de datos contiene las descripciones cortas que se encuentran en las páginas man de los comandos del sistema. Un ejemplo de su uso es el siguiente:

```
# whatis passwd
passwd (1) - update a user's authentication tokens(s)
passwd (1ssl) - compute password hashes
passwd (5) - password file
passwd.nntp [passwd] (5) - passwords for connecting to remote NNTP
servers
```

Como puedes ver en este ejemplo el comando passwd tiene entradas en las secciones uno y cinco de las páginas man. También ha sido encontrado en la sección uno de las páginas man del comando ssl.

El comando man -f busca en esta base de datos entradas coincidentes con la palabra clave indicada. A continuación tenemos un ejemplo de la salida producida por este comando.

```
# man -f passwd
passwd (1) - update a user's authentication tokens(s)
passwd (1ssl) - compute password hashes
passwd (5) - password file
passwd.nntp [passwd] (5) - passwords for connecting to remote NNTP
servers
```

Estos comandos realizan la misma búsqueda. Se muestran los comandos y las páginas man donde han sido encontrados. Esto puede ser práctico para localizar secciones de páginas man y variantes de los comandos.

103.1.11.4 Buscando con apropos.

Al igual que whatis, el comando apropos no se menciona específicamente en los objetivos del examen, pero conocer su uso puede venir bien en el examen. Y también como la utilidad whatis, el comando apropos utiliza la base de datos whatis. Este comando se emplea para buscar tanto los nombres de comando como las descripciones para la palabra clave indicada. A continuación vemos un ejemplo del comando apropos:

```
# apropos password
chpasswd (8) - update password file in batch
gpasswd (1) - administer the /etc/group file
htpasswd (1) - Create and update user authentication files
nppasswd (1) - Change a user's password
passwd (1) - update a user's authentication tokens(s)
passwd (1ssl) - compute password hashes
passwd (5) - password file
```

```
passwd.nntp [passwd] (5) - passwords for connecting to remote NNTP servers
pg_passwd (1) - Manipulate the flat password file
pwupdate (8) - updates passwd and shadow NIS map
rpc.yppasswdd [rpc] (8) - NIS password update daemon
smbpasswd (5) - The Samba encrypted password file
smbpasswd (8) - change a users SMB password
ypchfn [yppasswd] (1) - change your password in the NIS database
ypchsh [yppasswd] (1) - change your password in the NIS database
yppasswd (1) - change your password in the NIS database
```

El siguiente comando:

```
# man -k password
```

obtiene el mismo resultado. Esto puede ser práctico cuando busques comandos a partir de determinadas palabras clave.

103.21.11.5. Configurando el acceso a páginas man.

Como se mencionó con anterioridad en este capítulo, en el directorio */usr/man* se guardan por defecto los ficheros fuente de las páginas man. La variable de entorno MANPATH puede ser utilizada para cambiar el path de búsqueda por defecto de los ficheros fuente de las páginas man. La variable MANPATH sobreescribirá el path de búsqueda por defecto de las páginas man, así que es importante incluir el path de las páginas man existentes. Abajo tenemos un ejemplo de una definición de la variable MANPATH añadida a un fichero */home/user/.profile*.

```
# export
MANPATH=/usr/local/man:/usr/man/preformat:/usr/man:/usr/X11R6/man
```

La mayoría de los documentos almacenados en */usr/man* están comprimidos y sin formatear. El comando man utiliza el fichero */etc/man.config* para obtener información acerca de la correcta visualización de esos fichero.

Este fichero contiene la definición de MANPATH, así como, información relativa a las opciones de compresión, formateo y paginación. Mediante la opción -C podemos especificar un fichero de configuración diferente.

El comando man se encuentra en */usr/bin*. Este directorio debe estar incluido en la variable de entorno PATH, de lo contrario deberemos ejecutar el comando utilizando el path absoluto */usr/bin/man*.

103.1.EXTRA'S

103.1.EXTRA'S AYUDA

Cuando se trabaja con una interfaz gráfica el proceso de ejecución de aplicaciones es más intuitiva que desde una terminal. Si desconoces un comando y sus opciones es complicado realizar una tarea correctamente. Por ello se han implementado diversos métodos para averiguar información de ayuda acerca de los comandos y elementos del sistema.

El shell estandarizado en GNU/linux es bash o alguna modificación del mismo, de esta manera al usar éste, el propio binario incorpora un comando interno que aporta información de ayuda sobre

los comandos internos únicamente. El comando es help; comprobemos su funcionamiento con este ejemplo:

```
atlante:~$ help pwd
pwd: pwd [-LP]
    Muestra el nombre del directorio de trabajo actual.
```

Opciones:

- L muestra el valor de \$PWD si nombra al directorio de trabajo actual
- P muestra el directorio físico, sin enlaces simbólicos

Por defecto, `pwd' se comporta como si se especificara `-L'.

Estado de Salida:

Devuelve 0 a menos que se de una opción inválida o no se pueda leer el directorio actual.

```
atlante:~$ help man
bash: help: no hay temas de ayuda que coincidan con `man'. Pruebe
`help help' o `man -k man' o `info man'.
```

La última salida del ejemplo, da pie a verificar que el mismo sistema es capaz de mostrar otros métodos de extracción de ayuda mediante el comando man e info.

Para crear los manuales teniendo ya los ficheros fuentes existe una aplicación denominada catman. El propio manual aporta una idea de funcionamiento de esta aplicación.

```
root@cli:~# man catman
CATMAN(8)          Útiles para las Páginas del Manual
CATMAN(8)

NOMBRE
      catman - crea o actualiza las páginas del manual
preformatteadas

SINOPSIS
      catman [-dhV] [-M ruta] [sección] ...

DESCRIPCIÓN
      catman se usa para crear un conjunto actualizado de
páginas de manual preformatteadas, conocido como páginas cat. Las
páginas cat se muestran generalmente mucho más rápido que las
páginas del manual originales. La decisión de admitir páginas cat
es del administrador local, que debe suministrar directorios
apropiados para contenerlas.
(...)
```

El proyecto GNU desarrolló un sistema de ayuda que modifica las fuentes de los manuales del formato groff al formato de hipertexto, con la ventaja de poder usarla mediante un navegador. El

comando es info:

```
atlante:~$ info echo
File: coreutils.info,      Node: echo invocation,      Next: printf
invocation, Up: Pr\
inting text
```

15.1 `echo': Print a line of text

`echo' writes each given STRING to standard output, with a space between each and a newline after the last one. Synopsis:

```
echo [OPTION]... [STRING]...
```

Due to shell aliases and built-in `echo' functions, using an unadorned `echo' interactively or in a script may get you different functionality than that described here. Invoke it via `env' (i.e., `env echo ...') to avoid interference from the shell.

The program accepts the following options. Also see *note Common options::: Options must precede operands, and the normally-special argument `--' has no special meaning and is treated like any other STRING.

`-n'

Do not output the trailing newline.

`-e'

Enable interpretation of the following backslash-escaped
--zz-Info: (coreutils.info.gz)echo invocation, 92 líneas --
Top-----
Este es Info, versión 4.13. Teclee h para ayuda, m para
seleccionar un ítem

El manual de info ayuda a comprender el funcionamiento del comando:

```
#man info

INFO(1)                                         User Commands
INFO(1)
NAME
    info - read Info documents
SYNOPSIS
    info [OPTION]... [MENU-ITEM...]
DESCRIPTION
    Read documentation in Info format.
OPTIONS
```

```

-k, --apropos=STRING
      look up STRING in all indices of all manuals.
-d, --directory=DIR
      add DIR to INFOPATH.
--dribble=FILENAME
      remember user keystrokes in FILENAME.
-f, --file=FILENAME
      specify Info file to visit.
( . . . )

```

Recordar también que la mayoría de los comandos responden a la opción `-h --help` (con las excepciones que confirmán la regla) para mostrar una ayuda rápida.

Por último, cabe destacar, que en el mundo GNU/linux existen unos documentos que se conocen como los **HOWTO** que resuelven problemas no orientado al uso de un comando sino como una solución a un problema. Es habitual encontrar estos documentos en la internet tratando de localizar una solución concreta como por ejemplo “instalación de opensuse”. Una URL paradigmática es <http://www.howtoforge.com/>.

103.1 EXTRAS type

Los comando pueden ser internos (funciones incluidas en el binario del shell) o externos (llamadas a programas. En ocasiones (y dependiendo del shell) es posible ejecutar un comando que sea interno y tener su equivalente externo. Para saber de qué tipo es la instrucción se puede usar el comando `type`, que lo indica.

```

atlante:~$ type cd
cd es una orden interna del shell
atlante:~$ type bash
bash es /bin/bash
atlante:~$ type quote
quote: es una función
quote ()
{
  echo '\${1//\'/'\\\'\\\'\\\'}\\'
}

```

Como se puede comprobar, el comando `type` para los comandos externos muestra la ruta del ejecutable. También señala la secuencia de una función.

```

atlante:~# type type
type es una orden interna del shell

```

103.1 EXTRAS

103.1 EXTRAS Atajos de teclado de bash

Comandos con la tecla CTRL

Atajo	Significado
CTRL + A	Se envía el puntero al comienzo de la línea del prompt.
CTRL + C	Se aborta la ejecución del proceso actual. Útil para frenar la ejecución de un script.
CTRL + D	Se envía una señal de EOF. Útil para <i>detener la ejecución de intérpretes</i> como los de Python, Ruby o Java.
CTRL + E	Se envía el puntero al final de la línea del prompt.
CTRL + K	Se elimina el contenido de la línea desde la posición del puntero hacia la derecha.
CTRL + L	Se limpia la pantalla de la misma forma que con el comando <i>clear</i> o <i>cls</i> (ms-dos).
CTRL + O	Ejecutar comando (igual que el pulsar <i>intro</i>).
CTRL + R	Se busca en el historial el último comando que corresponda a la búsqueda que introduzcamos.
CTRL + U	Se corta el texto a la izquierda del puntero. Útil si tecleamos un comando que vamos a usar con posterioridad o queremos borrar una línea.
CTRL + Y	Se pega el texto que ha sido cortado con CTRL+U.
CTRL + Izquierda	Se desplaza el puntero una palabra a la izquierda.
CTRL + Derecha	Se desplaza el puntero una palabra a la derecha.

Otros comandos

Atajo	Significado
SHIFT + Insert	Se pega el contenido del portapapeles.
Flecha Arriba	Se muestra el comando anteriormente ejecutado.
Flecha Abajo	Se muestra el siguiente comando que hemos ejecutado.

103.2. Procesar cadenas de texto por medio de filtros.

Peso en el examen de certificación: 3 puntos.

Objetivo: Aplicación de filtros a cadenas y flujos de caracteres.

Conceptos y áreas de conocimiento:

.Envío de ficheros y flujos de texto a través de utilidades de filtros de texto para su modificación posterior con los comandos incluidos en el paquete de utilidades GNU "textutils".

Términos y utilidades

- cat
 - cut
 - expand
 - fmt
 - head
 - od
 - join
 - nl
 - paste
 - pr
 - sed
 - sort
 - split
 - tail
 - tr
 - unexpand
 - uniq
 - wc
-

103.2.1.- Comandos básicos para mostrar archivos

cat

Muestra el contenido de un archivo de texto, ese es su uso principal. Puede emplearse como redireccionador de lo que entra por su entrada standard y lo envía a su salida standard, pero eso se verá más adelante (3.4 pipes).

Un ejemplo sería:

```
$ cat /etc/apt/sources.list  
$ cat apuntes.html
```

tac

Es como **cat** pero al revés, es decir, muestra el archivo de atrás hacia adelante, mostrando las líneas del texto de la última a la primera. Lo da la vuelta como un calcetín. Se emplea en archivos *log* (registro de cosas que han acontecido).

Un ejemplo sería:

```
$ tac /var/log/apt/history.log
```

En este ejemplo podemos hacer un seguimiento de instalaciones y actualizaciones pero desde las últimas a las primeras (sentido inverso al archivo, donde lo último se registra al final).

head

En inglés significa cabeza o cabecera, está claro que nos mostrará el principio de un archivo de texto, por defecto las 10 primeras líneas.

Tiene dos opciones principales:

- n <nº lineas> Muestra las n primeras líneas
- c muestra los n primeros caracteres.

Un ejemplo sería:

```
$ head lista.txt -n 5
```

tail

En inglés significa cola, que es lo contrario de cabeza. Muestra las líneas finales, por defecto las 10 últimas. Pero mucho ojo, **las muestra en su orden correcto**, no invierte el orden como pasaba en **tac**. Dispone de opciones **-n** y **-c**, con significado similar a **head**. Además dispone de dos opciones adicionales:

- f muestra las últimas líneas y todas las nuevas que vayan apareciendo. Muy útil para archivos log.
- + <nº linea> muestra las líneas a partir del número de linea indicado

Un ejemplo sería:

```
$ tail /var/log/apt/history.log -n
```

Nos mostrará la última instalación o actualización de paquetes realizada.

103.2.2.- Comandos para contar u numerar

wc

El comando proviene del término "word counter", contador de palabras. Cuenta líneas, palabras y caracteres. Dependiendo de la opción que se emplee.

-l cuenta lineas (lines)

- w cuenta palabras (words)
- c cuenta caracteres (char) pero de un byte, por tanto cuenta bytes
- m cuenta caracteres independientemente de si el código es de un byte (ANSI) u varios (UTF)

Vamos a ver un ejemplo de la opción más comprometida, la de contar caracteres, viendo la diferencia entre -c y -m. En este ejemplo usamos un archivo UTF-8 con 'ñ' y vocales acentuadas (que ocupan dos bytes en vez de uno). El contador arroja valores diferentes

```
$ cat ejemplo.txt
España ha ganado el mundial de fútbol.
El equipo campeón es España.
$ wc ejemplo.txt -c
73 ejemplo.txt
$ wc ejemplo.txt -m
69 ejemplo.txt
```

En un caso arroja 73 (bytes) que corresponden a 69 caracteres.

nl

Numerador de lineas (numbered lines) Equivalente a **cat -b**. Con la opción -bt muestra sólo las lineas que no estén en blanco. No requiere mayor explicación.

103.2.3.- Comandos manejadores de tabuladores

Estos comandos recogen la entrada de un archivo o de la entrada standard y vuelcan el resultado en la salida estándar. Son estos dos:

expand

Sustituye los tabuladores que aparecen en el archivo por su equivalente en espacios en blanco. Por defecto emplea 8 espacios, con la opcion -t se puede modificar.

unexpand

Mas o menos lo contrario de expand, sustituye dos o mas espacios en el texto por tabuladores.

103.2.4.- Comandos para volcado de archivos como números hexadecimales y otros formatos

hexdump Vuelca el contenido de un archivo binario por la salida standard. Existen diversas maneras de emplear este comando, la más empleada es -C que hace la salida más clara. Lo que hace es mapear el contenido de un archivo como hexadecimal indicando posiciones y valores almacenados. Veamos un ejemplo.

```
$ cat ejemplo.txt
Estaba la pastora, lará lará larito.
$ hexdump ejemplo.txt
00000000 7345 6174 6162 6c20 2061 6170 7473 726f
00000010 2c61 6c20 7261 a1c3 6c20 7261 a1c3 6c20
00000020 7261 7469 2e6f 0a0a
00000028
$ hexdump ejemplo.txt -C
```

```
00000000 45 73 74 61 62 61 20 6c 61 20 70 61 73 74 6f 72 |
Estaba la pastor|
00000010 61 2c 20 6c 61 72 c3 a1 20 6c 61 72 c3 a1 20 6c |
a, lar.. lar.. l|
00000020 61 72 69 74 6f 2e 0a 0a |arito...|
00000028
```

Como se puede apreciar, permite ver la codificación interna del archivo en hexadecimal, mostrando que está en UTF-8, donde un carácter a veces son varios bytes.

od

Abreviatura de 'octal dump' vuelca archivos a la salida standard en formato octal. Tiene diversas opciones para hacer los volcados en otros formatos. Las opciones más comunes son:

- c hace el volcado a caracteres ASCII
- d en enteros sin signo
- i a enteros (integer)

Un ejemplo con el mismo archivo anterior sería:

```
$ od ejemplo.txt -c
0000000 E s t a b a l a p a s t o r
00000020 a , l a r 303 241 l a r 303 241 l
00000040 a r i t o . \n \n
00000050
```

103.2.5.- Comandos para dividir y juntar archivos

split

Corta o trocea un archivo en trozos, de un número de líneas determinado o de un tamaño determinado. Split como adjetivo se emplea para por ejemplo la leche cuando está cortada, significa separar.

Se puede trocear el archivo empleando -l y el número de líneas de cada parte o -b y el tamaño en bytes de cada parte. Pongamos un ejemplo:

```
$ split -b 1024k cancion.mp3 parte_
```

Esto genera los archivos:

```
$ ls parte_*
parte_aa parte_ab parte_ac parte_ad
```

Estos archivos pueden recomponer una copia del original usando:

```
$ cat parte_*> copia.mp3
```

cut

Se emplea en archivos de texto donde en cada linea hay campos separados por un separador. Corta determinadas columnas del archivo y las muestra por la salida standard.

El separador de campo se especifica con -d y -f de la posición del campo o campos que deseamos "cortar". Vamos a ver un ejemplo con el archivo */etc/group* que contiene lista de grupos separados por ":". Vamos a cortar el primer y el tercer campo.

```
$ cut -d ':' -f 1,3 /etc/group
```

```
root:0
daemon:1
bin:2
sys:3
adm:4
tty:5
disk:6
lp:7
mail:8
news:9
uucp:10
man:12
...
...
```

Ahora vamos a emplear la opción -c que corta los n primeros caracteres:

```
$ cut -c 2-6 /etc/group
oot:x
aemon
in:x:
ys:x:
dm:x:
ty:x:
isk:x
...
...
```

paste

Combina archivos donde una parte de cada linea está en cada uno de los archivos. En definitiva monta un archivo donde las columnas proceden de otros archivos.

Vamos a ver un ejemplo con dos archivos de texto, uno con un número en cada línea y otro con las letras "a", "b" y "c" en cada línea. Este es el resultado de aplicar el comando.

```
$ paste numeros.txt letras.txt
uno a
dos b
tres c
...
```

join

Similar a paste, pero mas potente. Permite especificar campos directamente. Su significado en inglés podríamos traducirlo como "enrolarse", o "formar parte de", aunque en este caso sería más acertado "combinar". Lo que hace es meter datos de un archivo junto con los de otro pero permite establecer condiciones.

El comando nos permite combinar dos archivos con líneas que contienen campos eligiendo el orden que ocuparán en el texto resultante, teniendo en cuenta que alguno de los campos (el que yo elija) de un fichero sea igual a alguno (elegido también) de los campos del otro fichero a combinar.

Vamos "la caña". Hay que tener en cuenta que el puntero recorre ambos archivos hacia adelante, no volviendo atrás, por tanto los campos a comparar deben ser en orden ascendente o descendente.

El formato del comando es **join** -1 <nº campo> -1 <campo> <archivo uno> <archivo dos>

Veamos un ejemplo:

```
$ cat paises
1 España español
...
```

```

2 Francia francés
3 Japón japonés
4 China chino
$ cat países abreviaturas
1 ESP
2 FRA
4 CHN
$ join -1 1 -2 1 países abreviaturas
1 España español ESP
2 Francia francés FRA
4 China chino CHN

```

En este ejemplo hemos mezclado el archivo *países* con el archivo *abreviaturas de manera que si el campo primero (el número) de cada archivo era igual se mezclaran los campos, unos a continuación de los otros. Al no disponer de abreviatura para japon (4), este no ha salido.*

Si deseamos que aparezca primero la abreviatura y luego el país, y nada más, el comando a aplicar sería:

```

$ join -1 1 -2 1 -o '2.2 1.2' países abreviaturas
ESP España
FRA Francia
CHN China

```

Como puede apreciarse me he sacado de la manga la opción -o, no explicada antes. la 'o' viene de 'order' y sirve para indicar los campos y el orden en el que aparecerán, se ponen dos números separados por un punto, el primero el archivo y el segundo el campo dentro del archivo.

La utilidad de este comando es para combinar archivos con información sobre procesos, usuarios, etc , formados por campos donde el primero es un número. Aparte puede tener muchas otras, por ejemplo para responder correctamente al cuestionario LPI.

103.2.6.- Comandos para ordenar o modificar

sort

El comando sort ordena las líneas de uno o más ficheros imprimiendo el resultado en pantalla. Su sintaxis es:

sort -opciones fichero (s)

Las opciones disponibles para este comando son:

-n Ordena las líneas de texto del fichero desde el principio (+n), por el final (-n), según el campo representado por el numero n.

Por ejemplo:

```

$ cat fich1
Pepe 48486
Juan 48481
Maria 48487
Luis 48483
Pedro 48482

```

```
$ sort +1n fich1
Juan 48481
Pedro 48482
Luis 48483
Pepe 48486
Maria 48487
```

- r Invierte el orden.
- f Ignora las mayúsculas y minúsculas en el momento de ordenar el texto.
- d Ordena según el abecedario. Primero las letras, después los dígitos y espacios, el resto de símbolos se ignoran.
- o **fichero** El resultado lo almacena en el fichero y no lo muestra en pantalla.
- t **carácter** Utiliza este carácter como separador. Si no se especifica nada, se utiliza el espacio en blanco.

uniq

Este comando muestra el contenido de archivos suprimiendo líneas secuenciales repetidas. Es decir, líneas iguales consecutivas.

Veamos un ejemplo:

```
$ cat letra
debajo del olivo
debajo del olivo
debajo del olivo
el sol calienta
debajo el olivo
el sol calienta
debajo del olivo el sol calienta
$ uniq letra
debajo del olivo
el sol calienta
debajo el olivo
el sol calienta
debajo del olivo el sol calienta
```

Con la opción -u sólo pone las líneas que no se repiten:

```
$ uniq letra -u
el sol calienta
debajo el olivo
el sol calienta
debajo del olivo el sol calienta
```

Alguno dirá ¿pero si son dos frases que no paran de repetirse? Si, pero no de forma secuencial, consecutiva, varias veces seguidas. Ahora se entiende ¿no?

tr

Convierte caracteres del archivo original en otros siguiendo el patrón establecido. El nombre del comando procede de "translate", traducir. Este comando sólo recoge datos de la entrada standard, no de archivos directamente, por tanto siempre va detrás de un pipe.

Veamos un ejemplo:

```
$ echo España ha ganado el mundial| tr '[a-z]' '[A-Z]'
ESPAÑA HA GANADO EL MUNDIAL
```

En este caso tratamos de convertir todo a mayúsculas, pero en el caso del español nos fallará con los acentos y las 'ñ'.

En este ejemplo vamos a convertir un carácter concreto en varios:

```
$ echo España | tr '[ñ]' '[gn]'  
Espagna
```

Esto puede ser útil para eliminar eñes en caso de pasar texto a formatos que no las soportan bien.

103.2.7.- Comandos previos a la impresión

Estos comandos son algo prehistóricos en cierto modo, pensados para impresoras basadas en caracteres, que debían soportar grandes listados.

fmt

Abreviatura de 'format', modifica el formato a determinado número de caracteres por línea, pensado para preprocessar la impresión .Configura el texto por defecto a 75 caracteres por linea. Tiene tres opciones:

- w : indica el número de caracteres por línea
- s: divide líneas grandes, pero no las rellena
- u: Establece un espacio entre palabras y dos entre sentencias (por si se nos quedó la mano pegada en el espaciador)

Veamos un pequeño ejemplo de la última opción (ya sé que ese C no es correcto):

```
$ cat prueba  
printf("Perico los palotes") ;  
if else then  
$ fmt prueba -u  
printf("Perico los palotes") ; if else then
```

Es importante saber que no actúa con la entrada standard, por tanto no nos lo encontraremos detrás de un pipe.

pr

Divide el archivo para impresión en páginas de 66 líneas y 72 caracteres de ancho. Tiene dos opciones:

- l para especificar líneas por página
- w para especificar caracteres por línea

Al ejecutarlo nos genera al menos una página, le añade una cabecera con fecha (formato año-mes-dia), la hora, el nombre del archivo y la página.

103.3. Administración básica de archivos

Peso en el examen de certificación: 4 puntos.

Objetivo: Uso de comandos básicos para la gestión de ficheros y directorios.

Conceptos y áreas de conocimiento:

- Copiar, mover y eliminar ficheros y directorios individualmente.
- Copiar múltiples ficheros y directorios recursivamente.
- Eliminar ficheros y directorios recursivamente.
- Uso de comodines sencillos y avanzados en la conjugación de comandos.
- Uso de **find** para la localizar y actuar sobre ficheros en base a las opciones de **find**: type, size, or time.
- Uso de **tar**, **cpio** y **dd**.

Términos y utilidades

- cp
- find
- mkdir
- mv
- ls
- rm
- rmdir
- touch
- tar
- cpio
- dd
- file
- gzip
- gunzip
- bzip2
- file globbing

El objetivo de este tema es la Administración básica de los ficheros en LINUX. Linux es una colección de ficheros almacenados en su disco duro. Debido a ello, la capacidad de administrar los ficheros contenidos en sus sistemas de ficheros es un factor importante para cualquier administrador de sistemas. Por esto, aprenderemos a copiar, mover, visualizar, renombrar etc. ficheros y directorios.

NOTA: En el formato de los comando los elementos que estén entre corchetes significa que son elementos optativos.

103.3.1. Reglas de nomenclaturas de ficheros.

Los nombres de los ficheros de Linux son como los de cualquier otro SO. Sin embargo, cada SO posee sus propias peculiaridades en la nomenclatura, diferencias que pueden ser un problema para

aquellos que se suelen mover entre sistemas o sí desean pasar ficheros de un sistema a otro. Las reglas para nombrar a los directorios y a los ficheros de LINUX, son las siguientes:

- Los caracteres que se pueden utilizar son letras mayúsculas/minúsculas, dígitos y caracteres especiales; aunque no se recomienda el uso de caracteres especiales, sobre todo si tienen un significado especial para el LINUX. Por ejemplo algunos ficheros de copia de seguridad terminan en virgulilla (~).
- Aunque los nombres de ficheros de Linux pueden contener espacios, y a pesar de que éstos son habituales en algunos SO, los espacios se deben indicar mediante caracteres de escape en la línea de comandos de Linux, anteponiendo una barra invertida (\) al espacio o rodeando todo el nombre de fichero con comillas("). Este requisito hace que resulte un poco incómodo utilizar espacios en Linux, por lo que se suelen sustituir por guiones (-) o subrayados (_).
- Si el nombre comienza por punto (.), se convierte en oculto.
- Linux distingue entre mayúsculas y minúsculas en los nombres de los ficheros.
- La longitud de los nombres depende del sistema de ficheros. En ext2, ext3 y muchos otros el límite es de 255 caracteres.
- Hay dos nombres de ficheros particularmente especiales. Un nombre de fichero consistente en solo punto (.) hace referencia al directorio actual, mientras que uno consistente en un punto doble(..) hace referencia al directorio padre.
- Existen una serie de recomendaciones que son las siguientes:
 - Si nuestro ordenador tiene otro S.O. además del LINUX, y existe un traspaso de información de un sistema al otro, es recomendable que se sigan las reglas del S.O. que posea mayores restricciones a la hora de nombrar a los ficheros y directorios. Lo mismo ocurre con otro tipo de sistemas o a la hora de utilizar lenguajes de programación o paquetes concretos.
 - Se intentará que el nombre, describa el contenido del fichero o directorio.

103.3.2. Visualizar contenidos de directorios.

Para listar el contenido de un directorio poseemos varias órdenes, pero todas son pequeñas modificaciones de la orden **ls**. Las opciones que veremos para una orden valdrán para las demás.

COMANDO: ls

FORMATO: **ls [-opciones] [directorios o archivos]**

FUNCIÓN : Visualiza los nombres de los ficheros existentes en el/los directorios indicados en columnas. Si indicamos el/los nombre/s de fichero/s visualiza información de dicho/s fichero/s.

OPCIONES:

1 Visualiza el nombre de los ficheros/directorios en 1 columna

C Visualiza el directorio en columnas

F Distingue los directorios con / al final del nombre, los enlaces simbólicos con @ y los ficheros ejecutables con un * al final del nombre

R Visualiza en recursivo. Es decir visualiza toda la estructura de ficheros y directorios del

directorio indicado

- a** Visualiza además ficheros y directorios ocultos
- b** Visualiza caracteres no gráficos del nombre de los ficheros
- d** Visualiza el nombre del directorio no su contenido
- l** Visualiza la información más común de los ficheros
- n** Visualiza el nº de propietario y el nº del grupo
- m** Visualiza el nombre de los ficheros seguidos y separados por una coma
- o** Visualiza toda la información excepto el nombre del grupo
- x** Visualiza el nombre de los ficheros ordenados por filas
- i** Visualiza el nº de enlace o inodo de cada fichero

Ejemplo: \$ ls /

```
$ ls -l/
total 84
drwxr-xr-x 2 root root 4096 2008-06-03 12:43 bin
drwxr-xr-x 3 root root 4096 2008-06-03 12:43 boot
lrwxrwxrwx 1 root root 11 2008-06-03 12:32 cdrom -> media/cdrom
drwxr-xr-x 12 root root 13940 2012-02-06 11:03 dev
drwxr-xr-x 121 root root 12288 2012-02-06 11:15 etc
drwxr-xr-x 23 root root 4096 2011-05-27 12:01 home
drwxr-xr-x 2 root root 4096 2008-04-22 19:48 initrd
lrwxrwxrwx 1 root root 33 2008-06-03 12:43 initrd.img ->
boot/initrd.img-2.6.24-16-generic
drwxr-xr-x 16 root root 4096 2008-06-03 12:43 lib
drwx----- 2 root root 16384 2008-06-03 12:32 lost+found
drwxr-xr-x 4 root root 4096 2012-02-06 11:03 media
drwxr-xr-x 3 root root 4096 2011-11-11 12:55 mnt
drwxr-xr-x 2 root root 4096 2008-04-22 19:48 opt
dr-xr-xr-x 118 root root 0 2012-02-06 11:03 proc
drwxr-xr-x 5 root root 4096 2008-10-27 11:26 root
drwxr-xr-x 2 root root 4096 2008-06-03 12:43 sbin
drwxr-xr-x 2 root root 4096 2008-04-22 19:48 srv
drwxr-xr-x 12 root root 0 2012-02-06 11:03 sys
drwxrwxrwt 12 root root 4096 2012-02-06 11:15 tmp
drwxr-xr-x 11 root root 4096 2008-04-22 19:51 usr
drwxr-xr-x 15 root root 4096 2008-04-22 20:07 var
lrwxrwxrwx 1 root root 30 2008-06-03 12:43 vmlinuz ->
boot/vmlinuz-2.6.24-16-generic
```

103.3.3. Crear directorios.

Para crear directorios utilizaremos la orden **mkdir**.

COMANDO: **mkdir**

FORMATO: **mkdir [-opciones] directorios**

FUNCIÓN: Crear los directorios indicados. En **directorios** puede indicar rutas o una lista de directorios separados por blancos.

OPCIONES:

m Tras **m** especificaremos los permisos que deseamos que tenga el directorio a crear. Los permisos **m** los indicaremos según las reglas del comando **chmod**

p Al indicar esta opción se crean los directorios padres del directorio a crear antes de crear éste

Ejemplo:

\$ mkdir D1 D2

Crea el directorio D1 y el D2 en el directorio activo.

\$mkdir -p D1/D2

En caso de no existir el directorio llamado D1 en el directorio activo crea el directorio D1 en el directorio activo y dentro de éste el D2. En caso de existir el directorio llamado D1 en el directorio activo crea el directorio D2 dentro del D1.

103.3.4. Borrar directorios.

Para borrar directorios utilizaremos la orden **rmdir**.

COMANDO: **rmdir**

FORMATO : **rmdir [-opciones] directorios**

FUNCIÓN : Borra los directorios indicados en **directorios**. Un directorio no se puede borrar si no está vacío o es el directorio activo.

OPCIONES:

p Borra si es posible los directorios de la ruta del directorio indicado

Ejemplo: **\$ rmdir -p D1/D2/D3**

Borra todos los directorios indicados si están vacíos.

103.3.5. Cambiarse de directorio.

Para hacer que un directorio sea el activo utilizaremos la orden **cd**.

COMANDO: **cd**

FORMATO : **cd [directorio]**

FUNCIÓN : Cambiar el directorio indicado. En caso de no indicar el directorio al que deseamos cambiarnos, se cambiará siempre al directorio HOME. HOME es el directorio de conexión de cada usuario, es decir; siempre que un usuario se conecta, su directorio activo en ese momento será HOME. Los directorios HOME se encuentran en el directorio **/home**. El directorio HOME de un usuario tiene el mismo nombre que el usuario; por lo tanto, en nuestro caso el directorio HOME de un usuario llamado **user1** será: **/home/user1**

Ejemplo: **\$ cd D1/D2**

El directorio activo será el D2 que se encuentra dentro del D1 del directorio activo.

103.3.6. Saber en que directorio nos encontramos.

Para conocer la ruta del directorio activo utilizaremos la orden **pwd**.

COMANDO: **pwd**

FORMATO : **pwd**

FUNCIÓN : Visualizar la ruta del directorio en el que nos encontramos.

Ejemplo:

\$ cd

S **pwd**

/home/aeg

103.3.7. Visualizar contenidos de ficheros.

Para visualizar el contenido de ficheros poseemos varios comandos como **cat, more, less, head, tail**.

COMANDO: **cat**

FORMATO: **cat [-opciones] nom-ficheros**

FUNCIÓN: Visualiza el contenido de los ficheros indicados.

OPCIONES:

A Visualiza todos los caracteres no gráficos

s Si existen varias líneas en blanco seguidas sólo visualiza 1

T Visualiza el tabulador como ^I

E Visualiza los <INTRO> como \$

n Visualiza las líneas numeradas

Ejemplo: \$cat aeg.txt

NOTA: Para visualizar ficheros la orden más sencilla es la orden "cat" anteriormente vista. Con "cat" podemos visualizar cualquier archivo, pero su salida es continua hasta que se encuentra el fin de fichero. Par visualizar ficheros largos poseemos una serie de órdenes (**more,less, head y tail**) que nos facilitarán la visualización de los mismos.

COMANDO: **more**

FORMATO: **more [-opciones] ficheros**

FUNCIÓN: Visualiza el contenido de los ficheros indicados por páginas, o por pantallas. Si indicamos varios ficheros, el contenido de éstos los separa por unas líneas de 2 puntos.

OPCIONES:

c La visualización de cada página, comienza por la 1^a línea, es decir, no hace scroll.

n Siendo **n**, un número que especifica el número de líneas que se visualizan, al utilizar la barra espaciadora.

s En caso de existir varias líneas en blanco seguidas, visualiza únicamente una.

+n Siendo **n** un número de línea. Comienza a visualizar el fichero desde la línea indicada.

+/cadena Comienza a visualizar el contenido del fichero, 2 líneas antes de la línea que contiene la

primera aparición de la cadena indicada.

d Visualiza por cada pantalla, un mensaje de lo que hay que hacer, para continuar la visualización del fichero.

Una vez dentro del more, es decir, visualizando el contenido del fichero, podemos utilizar una serie de teclas entre ellas las siguientes:

<RETURN> Visualiza una línea más del fichero.

<barra espaciadora> Provoca la visualización de la siguiente pantalla del fichero.

q ó **Q** Para terminar de visualizar el fichero.

h ó ? Visualiza la pantalla de ayuda de estas teclas.

b Va a la pantalla anterior

v Comienza la ejecución del editor vi, por lo que nos permite modificar el fichero.

. Repite la última acción realizada.

= Visualiza el número de la línea en curso, (la última visualizada).

!comando Ejecuta el comando del shell indicado.

:n Salta al siguiente fichero

:p Salta al fichero anterior

:f Visualiza el nombre del fichero.

Ejemplo \$ more aeg.txt

COMANDO: less

FORMATO: less [-opciones] ficheros

FUNCIÓN: Visualiza el contenido de los ficheros indicados por pantallas o por páginas.

OPCIONES:

+n Igual que con more.

/cadena Comienza a visualizar el fichero a partir de la línea que contiene la cadena.

Como en el more, se pueden utilizar una serie de teclas, entre las que se encuentran las siguientes:

(h, q, , !) Estas teclas se utilizan igual que con more.

ny Va **n** líneas hacia atrás.

b Va una pantalla hacia atrás.

g Va a la primera línea del fichero.

G Va a la última línea del fichero.

/cadena Visualiza como 1ª línea, aquella línea que contenga la cadena indicada de las páginas siguientes a la actual.

n Repite la búsqueda anterior hacia delante.

?cadena Visualiza como 1ª línea, aquella línea que contenga la cadena indicada de las páginas anteriores a la actual.

N Repite la búsqueda anterior hacia atrás.

Para visualizar las primeras líneas de los ficheros utilizaremos el comando **head**.

COMANDO: head

FORMATO: head [-opciones] ficheros

FUNCIÓN: Visualizar las 10 primeras líneas de los ficheros indicados.

OPCIONES:

n Siendo **n** el número de líneas primeras que se desean visualizar.

Ejemplo: \$ head -3 aeg.txt

Para visualizar las últimas líneas de los ficheros utilizaremos el comando **tail**.

COMANDO: tail

FORMATO: tail [-opciones] ficheros

FUNCIÓN: Visualizar la cola del fichero indicado, (las últimas 10 líneas por defecto).

OPCIONES:

n Visualiza las **n** últimas líneas.

Ejemplo: \$tail -3 aeg.txt

103.3.8. Copiar ficheros.

Para realizar copias de ficheros utilizaremos el comando **cp**.

COMANDO: cp

FORMATO: cp [-opciones] origen destino

FUNCIÓN: Copiar el fichero indicado en **origen**, en el **destino** indicado. Como destino podemos tener un fichero si el origen también lo es o un directorio si el origen es un conjunto de ficheros o un directorio.

OPCIONES:

I Realizará un enlace duro (Hard link).

r Realiza una copia en recursivo. Es decir si indicamos un directorio copia la estructura del mismo.

i Si el fichero destino existe pide confirmación para sobrescribirlo.

p Mantiene los atributos de origen.

u Sólo copiara el fichero si el origen es más reciente que el destino.

Ejemplo: \$ cp aeg.txt d1 \$ cp -r d1 d2

103.3.9. Mover/Renombrar ficheros.

Para renombrar, mover ficheros y directorios utilizaremos el comando **mv**

COMANDO: mv

FORMATO: mv [-opciones] origen destino

FUNCIÓN: Renombrar o mover ficheros y directorios. Si **origen** y **destino** son directorios, renombra. Si **origen** y **destino** son ficheros, renombra.

OPCIONES:

v Visualiza la acción que está realizando.

Ejemplo: \$ mv aeg.txt aeg12.txt Renombra aeg.txt por aeg12.txt

\$ mv aeg.txt d1 Mueve aeg.txt a d1

\$ mv d1 d2 Renombra d1 como d2(si d2 no existe). Mueve d1 a d2 (si d2 existe)

103.3.10. Borrar ficheros.

Para borrar ficheros y directorios utilizaremos el comando **rm**.

COMANDO: **rm**

FORMATO: **rm [-opciones] ficheros**

FUNCIÓN: Borrar los ficheros indicados

OPCIONES:

r Borra en recursivo. Es decir si indicamos un directorio borra la estructura del mismo.

v Visualiza el nombre del fichero mientras lo borra.

i Antes de borrar el fichero, preguntará si queremos borrarlo. Si contestamos "s", lo borra , si contestamos "n" no lo borra.

Ejemplos: \$ rm aeg.txt \$ rm -i aeg1.txt aeg2.txt

103.3.11. Modificar los datos cronológicos de ficheros.

Los sistemas de ficheros nativos de Linux mantienen tres tipos de marcas temporales para cada ficheros (Hora de creación, Hora de última modificación y Hora de último acceso). Hay varios programas que se basan en estas marcas temporales (Por ejemplo **make**), Por consiguiente, a veces tendrá que modificar las marcas temporales de los ficheros. Para ello utilizaremos el comando **touch**.

COMANDO: **touch**

FORMATO: **touch ficheros**

FUNCIÓN: Cambia la hora de modificación y acceso de los ficheros indicados. Si los ficheros no existen los crea vacíos.

OPCIONES:

a Cambia la hora de acceso, pero no la de modificación.

c No crea el fichero en caso de que no exista.

t tiempo En **tiempo** indicaremos un tiempo concreto. El formato de **tiempo** es:

[[CC]YY]MMDDhhmm [ss]. Donde MM es el mes, DD el día, hh es la hora(en formato 24h), mm son los minutos, **[[CC]YY]** es el año (como 2012 o 12, ambos son equivalentes) y **[ss]** son los segundos.

r fich La marca temporal del fichero indicado en **fich** será la marca que se le asigna a los ficheros.

Ejemplo: \$touch aeg.txt

103.3.12. Caracteres comodín.

Para hacer referencia a un conjunto de ficheros simultáneamente, utilizaremos los wildcards o caracteres comodines.

Wildcards son caracteres, por los cuales podemos simbolizar parte del nombre de un archivo, y de esta forma hacer referencia a más de un archivo al mismo tiempo.

Los caracteres comodines que poseemos en LINUX son:

* Representa cualquier cadena de caracteres incluso la cadena vacía.

? Representa a un único carácter.

[]

1.- Si dentro de los corchetes escribimos una lista de caracteres, el ordenador elige cualquier carácter dentro de dicha lista.

Ejemplo: \$ ls F[abc]

Visualiza los nombres de los ficheros o directorios, que empiecen por F , y luego tengan un único carácter, (este tendrá que ser "a", "b" o "c").

2.- Si dentro de los corchetes utilizamos un guión estamos indicando un rango de caracteres.

Ejemplo: \$ ls F[1-4]

Visualiza los nombres de los ficheros o directorios, que empiecen por F, y luego tengan un único carácter, (este tendrá que estar comprendido entre el 1 y el 4 ambos inclusive), es decir, visualizará si existen F1, F2, F3, F4.

3.- Si dentro del corchete, incluimos la , (!), admiración como primer carácter, significa que excluimos los caracteres.

Ejemplo: \$ ls F[!1-3]

Visualizará los ficheros que empiecen por F, y que luego tengan 1 carácter que no sea ni el 1, ni el 2, ni el 3.

Ejemplo: \$ ls F[!1-3]

Visualizará los ficheros que empiecen por F, y que luego tengan 1 carácter que no sea ni el 1, ni el 3.

103.3 EXTRAS

103.3 EXTRAS Más comandos

Algunos comandos útiles:

COMANDO cal:

El comando cal se utiliza para mostrar el calendario.

SINTAXIS:

La sintaxis es

cal [opciones] [mes] [año]

OPCIONES:

- 1 Muestra un sólo mes como salida.
- 3 Muestra el mes previo/actual/siguiente como salida.
- s Muestra el domingo como primer día de la semana.
- m Muestra el lunes como primer día de la semana.
- j Muestra fechas julianas (días ordenados, numerados desde el 1 de Enero).
- y Muestra un calendario para el año actual.

EJEMPLO:

```
root@cli:~# cal
```

```

Diciembre 2013
do lu ma mi ju vi sá
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
root@cli:~# cal 1 14
      Enero 14
do lu ma mi ju vi sá
 1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

```

COMANDO bc:

El comando bc se utiliza como calculador de la línea de comandos. Es similar a una calculadora básica. Usándolo podemos hacer cálculos matemáticos básicos.

SINTAXIS:

La sintaxis es
bc [opciones]

OPCIONES:

- c Sólo compilar. El output son comandos dc que son enviados al salida estándar.
- l Define las funciones matemáticas e inicializa la escala a 20, en vez de al cero por defecto.
- filename Nombre del archivo que contiene los comandos básicos de cálculo, éste no es un comando necesario.

EJEMPLO:

```

root@cli:~# bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
9*8
72
root@cli:~# bc -l
bc 1.06.95

```

```
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type `warranty'.  
1+2  
3  
root@cli:~# cat > ./calculo.txt  
8*5-3  
root@cli:~# bc ./calculo.txt  
bc 1.06.95  
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type `warranty'.  
37  
root@cli:/tmp$ bc  
bc 1.06.95  
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type `warranty'.  
obase=16  
255  
FF  
obase=2  
255  
11111111
```

COMANDO dirname:

El comando dirname espera que su primer argumento sea el nombre de un fichero del sistema operativo compuesto por una serie de directorios seguidos del nombre del fichero, separados por / como es habitual. dirname muestra por la salida estándar el camino de directorios que preceden al fichero, eliminando el nombre de éste

SINTAXIS:

La sintaxis es
dirname [arg]

EJEMPLO:

```
root@cli:~# dirname /etc/ssh/sshd_config  
/etc/ssh
```

COMANDO basename:

El comando basename espera que su primer argumento sea el nombre de un fichero del sistema

operativo compuesto por una serie de directorios seguidos del nombre del fichero, separados por / como es habitual. basename muestra por la salida estándar el nombre del fichero, eliminando el path a éste.

SINTAXIS:

La sintaxis es
basename [arg]

EJEMPLO:

```
root@cli:~# basename /etc/ssh/sshd_config
sshd_config
```

COMANDO date:

El comando date muestra la hora y la fecha.

SINTAXIS:

date [opciones] [+formato] [fecha]

OPCIONES:

- a Ajusta lentamente la hora en sss.fff segundos (fff representa fracciones de segundo). Este ajuste puede ser positivo o negativo. Sólo el administrador de sistema o superusuario puede ajustar la hora.
- s date-string Establece la fecha y hora al valor especificado en el datestring. El datestr puede contener los nombres de los meses, zona horaria, "am", "pm", etc.
- u Muestra (o establece) la fecha en Greenwich Mean Time (GMT-hora universal).

Formato:

- %a Día de la semana abreviado(Tue).
- %A Día de la semana completo(Martes).
- %b Nombre del mes abreviado(Jan).
- %B Nombre del mes completo(Enero).
- %c Formato de hora y fecha específico del país.
- %D Fecha en formato %m/%d/%y.
- %j Día del año juliano (001-366).
- %n Inserta una nueva línea.
- %p Cadena para indicar a.m. o p.m.
- %T Hora en formato %H:%M:%S.

%t Espacio de tabulación.

%V Número de la semana en el año (01-52); comienzo de la semana en Lunes.

EJEMPLO:

```
root@cli:~# date
mié dic 18 01:08:03 CET 2013
root@cli:~# date +\''La Fecha es '%D%t'La Hora es '%T\'"
"La Fecha es 12/18/13    La Hora es 01:08:06"
```

COMANDO w:

El comando w es similar al comando who y no está presente en todos los sistemas. who proporciona información relativa a entradas en el sistema. El comando w informa de qué están haciendo los usuarios. El listado estándar es:

SINTAXIS:

La sintaxis es

w [opciones]

EJEMPLO:

```
root@cli:~# w
 01:12:09 up 22:15,  2 users,  load average: 1,00, 1,01, 1,05
USER     TTY      FROM          LOGIN@    IDLE     JCPU    PCPU WHAT
root     tty1          mar02   8:58m  0.06s  0.05s -bash
root     pts/2    172.16.1.2  20:06   1.00s  0.27s  0.00s w
```

COMANDO cmp:

El comando de linux cmp compara dos archivos y te dice qué números de línea son distintos.

SINTAXIS:

cmp [opciones..] file1 file2

OPCIONES:

- c Muestra los octetos distintos como caracteres.
- l Muestra el número de octetos (decimal) y el valor del octeto distinto (octal) para cada diferencia.
- s No muestra nada para archivos distintos, devuelve el estado de salida únicamente.

EJEMPLO:

```
root@cli:~# cp pass1 pass2
```

```
root@cli:~# cmp pass1 pass2
root@cli:~# echo 0000 >> pass2
root@cli:~# cmp pass1 pass2
cmp: fin de fichero encontrado en pass1
```

COMANDO diff:

El comando diff se utiliza para encontrar diferencias entre dos archivos.

SINTAXIS:

diff [opciones..] de-archivo a-archivo

OPCIONES:

- a Trata todos los archivos como texto y los compara línea-a-línea.
- b Ignora cambios en la cantidad de espacios blancos.
- c Usa el formato de salida del contexto.
- e Hace que el salida sea un script ed válido.
- H Usa la heurística para acelerar el manejo de grandes archivos que tienen pequeños cambios dispersos.
- i Ignora los cambios mayúsculas y minúsculas, las considera equivalentes.
- n Mostrar en formato RCS, como -f excepto que cada comando especifica el número de líneas afectadas.
- q Mostrar diffs en formato RCS, como -f excepto que cada comando especifica el número de líneas afectadas.
- r Cuando compara directorios, compara repetidamente cualquier subdirectorio encontrado.
- s Informa cuando dos archivos sean iguales.
- w Ignora los espacios en blanco cuando compara líneas.
- y Utiliza el formato de salida uno junto al otro.

EJEMPLO:

```
root@cli:~# diff pass1 pass2
24a25
> 0000
```

COMANDO yes:

El comando yes muestra repetidamente la cadena dada separada por un espacio y seguida de una nueva línea hasta que se le detiene. Si no se da ninguna cadena, sólo muestra "y" repetidamente hasta que se le detiene. Se utiliza normalmente en scripts, su salida se une a un comando o programa que pida confirmación para hacer una u otra cosa (quieres borrar este archivo pulsa "y" o "n")

SINTAXIS:

yes [cadena..]
yes [opciones..]

OPCIONES:

--help Mostrar mensaje de ayuda y salir.
--version Mostrar versión y salir.

EJEMPLO:

```
root@cli:~# yes  
y  
y  
y  
(...)  
root@cli:~# yes REPETIDO  
REPETIDO  
REPETIDO  
(...)  
root@cli:~# yes n | rm -i fichero01.sh
```

COMANDO dmesg:

dmesg (diagnostic message, mensajes de diagnóstico) es un comando presente en los sistemas operativos Unix que lista el buffer de mensajes del núcleo. Este buffer contiene una gran variedad de mensajes importantes generados durante el arranque del sistema y durante la depuración de aplicaciones. La información ofrecida por dmesg puede guardarse en el disco duro mediante un demonio de registro, como syslog.

SINTAXIS:

dmesg

EJEMPLO:

```
root@cli:~# dmesg |more  
[ 0.000000] Initializing cgroup subsys cpuset  
[ 0.000000] Initializing cgroup subsys cpu  
[ 0.000000] Linux version 3.2.0-4-amd64 (debian-kernel@lists.debian.org) (gcc  
version 4.6.3 (Debian 4.6.3-14) ) #1 SMP Debian 3.2.51-1  
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-3.2.0-4-amd64 root=UUID=ec  
f98603-4439-4cee-8400-6bc9ccf84b98 ro quiet  
[ 0.000000] Disabled fast string operations  
[ 0.000000] BIOS-provided physical RAM map:  
[ 0.000000]   BIOS-e820: 0000000000000000 - 000000000009f000 (usable)  
[ 0.000000]   BIOS-e820: 000000000009f000 - 00000000000a0000 (reserved)  
[ 0.000000]   BIOS-e820: 00000000000ca000 - 00000000000cc000 (reserved)  
[ 0.000000]   BIOS-e820: 00000000000dc000 - 0000000000100000 (reserved)  
[ 0.000000]   BIOS-e820: 0000000000100000 - 0000000003fee0000 (usable)  
[ 0.000000]   BIOS-e820: 0000000003fee0000 - 0000000003feff000 (ACPI data)
```

103.4. Flujos, tuberías, y redireccionamiento de salida.

Peso en el examen de certificación: 4 puntos.

Objetivo: Redirigir flujos de datos y conectarlos con el fin de procesarlos eficientemente.

Redirigir la entrada estandar, salida estandar y salida de errores, canalizando la salida de un comando hacia la entrada de otro. Utilizar la salida de un comando como argumento de entrada de otro, enviando el resultado hacia la salida estandar y a un fichero simultaneamente.

Conceptos y áreas de conocimiento:

- Redireccionamiento de entrada estandar, salida estandar y salida de errores.
- Canalizar la salida de un comando hacia la entrada de otro comando.
- Utilizar la salida de un comando como argumentos de otro.
- Enviar los resultados de un comando hacia la salida estandar y un fichero.

Términos y utilidades

- tee
- xargs

El objetivo de este apartado es el de explicar las distintas formas de redireccionamiento de salida existentes en LINUX. Redireccionar significa modificar la dirección de los datos de un comando. El redireccionamiento de salida modifica en concreto la dirección de los resultados de un comando.

Los flujos, la redirección, y los pipes son algunas de las herramientas más potentes de la línea de comandos. Linux trata lo que entra y lo que sale de los programas como un flujo, que es una entidad de datos que se pueden manipular. Generalmente, la entrada viene del teclado y la salida va hacia la pantalla. No obstante, puede redirigir estos flujos de entrada y salida para que vengan de otras fuentes (como ficheros) o se dirijan a éstas. De igual manera, pueden canalizar (*pipe*) la salida de un programa hacia otro programa. Estos recursos pueden suponer una magnifica herramienta de conexión entre varios programas.

103.4.1. Tipos de flujos.

Para empezar a entender el redireccionamiento y los *pipes*, primero debe conocer los distintos tipos de flujos de entrada y salida, de entre los que destacan tres:

- **Entrada estándar:** Los programas aceptan la entrada de datos a través de la entrada estándar, o **stdin**, que en la mayoría de los casos es la información que llega al ordenador desde el teclado.
- **Salida estándar:** Los programas de modo texto envían a sus usuarios la mayoría de los resultados a través de la salida estándar, o **stdout**, que se suele mostrar en modo texto a pantalla completa o bien en una ventana GUI, como pueden ser *xterm* (los programas que son totalmente GUI, como los procesadores de texto GUI, no emplean la salida estándar para sus interacciones normales, aunque pueden emplear la salida estándar para mostrar mensajes en el *xterm* desde el que fueron iniciados, La salida de los GUI no se gestiona a través de un flujo de salida en el sentido aquí descrito).
- **Error estándar:** Linux proporciona un segundo tipo de flujo de salida, conocido como error

estándar o **stderr**. Este flujo de salida está pensado para llevar información de alta prioridad, como los mensajes de error.

Normalmente, los mensajes de error se envían al mismo dispositivo de salida que la salida estándar, por lo que cuesta diferenciarlos, aunque algunos se pueden redirigir independientemente, lo cual puede resultar muy útil. Por ejemplo, puede redirigir el error estándar a un fichero a la par que la salida estándar se sigue mostrando en la pantalla, de manera que puede interactuar con el programa y estudiar los mensajes de error mucho más adelante.

Internamente, los programas tratan estos flujos como simples ficheros de datos; los abren, leen los ficheros o escriben en ellos y los cierran cuando terminan. Dicho de otro modo, los ficheros normales son flujos desde el punto de vista de los programas. Lo que ocurre es que estos flujos son los que se utilizan para interactuar con los usuarios.

103.4.2. Redireccionamiento de salida.

Los símbolos de redireccionamiento de salida son los siguientes:

> Redirecciona la salida estándar, que es la pantalla. Al dirigir la salida a un fichero, si éste existiera perdería la información que poseyera. Si el fichero no existe lo crea.

Ejemplo: `$ls >memorias/aeg.txt`

Creamos el fichero **aeg.txt** en el directorio **memorias** del directorio activo, cuyo contenido será la lista de ficheros del directorio activo.

>> Redirecciona la salida standard. En este caso si se redirecciona a un fichero y éste ya existe, lo redireccionado se añade al final del fichero, sin borrar el contenido anterior.

Ejemplo: `$ls / >>memorias/aeg.txt`

Añade la lista de ficheros y directorios de la raíz al fichero **aeg.txt** del directorio **memorias** del directorio activo. Si ejecutamos estos dos comandos en este orden el contenido del fichero **aeg.txt** será la lista de ficheros y directorios del directorio activo y después la lista de ficheros y directorios de la raíz.

>> Redirecciona el error estándar, que es la pantalla. Crea un nuevo fichero que contiene el error estándar. Si el fichero existe lo sobrescribe.

Ejemplo: `$cat texto.txt 2>aeg.txt`

Creamos el fichero **aeg.txt** en el directorio activo, cuyo contenido será el error si el fichero **texto.txt** no existe. Si el fichero **texto.txt** existe el fichero **aeg.txt** se crea vacío.

>>> Redirecciona el error estándar, que es la pantalla. Si el fichero existe añade el error al fichero indicado, sino lo crea.

Ejemplo: `$cat texto.txt 2>>aeg.txt`

Añade el error al fichero **aeg.txt** en el directorio activo. Si el fichero **texto.txt** existe al fichero **aeg.txt** no se le añadirá nada.

&> Redirecciona tanto la salida estándar como el error estándar. Si el fichero existe lo sobrescribe

Ejemplo: `$cat texto.txt &>aeg.txt`

Creamos el fichero **aeg.txt** en el directorio activo, cuyo contenido será el error si el fichero **texto.txt** no existe y el contenido del fichero **texto.txt** en caso de que exista.

NOTA: Un truco muy habitual es redirigir la salida estándar o el error estándar hacia `/dev/null`. Este

fichero es un dispositivo que no está conectado a nada; se utiliza para deshacerse de datos.

Ejemplo: Suponiendo que no poseemos ningún fichero en el directorio activo, ejecutaremos una serie de comandos y mostraremos las respuestas de los mismos y los contenidos de los ficheros que se van creando o modificando.

```
$cat >aeg.txt
El contenido de este fichero son estas tres líneas
Segunda línea
Tercera línea
^d
$cat aeg.txt >nuevo.txt
$cat nuevo.txt
El contenido de este fichero son estas tres líneas
Segunda línea
Tercera línea
$cat aeg1.txt >nuevo.txt
cat: aeg1.txt: No existe el fichero o el directorio
$cat nuevo.txt
$cat aeg1.txt 2>nuevo.txt
$cat nuevo.txt
cat: aeg1.txt: No existe el fichero o el directorio
$cat aeg.txt 2>nuevo.txt
El contenido de este fichero son estas tres líneas
Segunda línea
Tercera línea
$cat nuevo.txt
$cat aeg.txt &>error
$cat error
El contenido de este fichero son estas tres líneas
Segunda línea
Tercera línea
$cat aeg1.txt &>error1
$cat error1
cat: aeg1.txt: No existe el fichero o el directorio
$
```

103.4.3. Tuberías.

Con frecuencia, los comandos pueden operar con la salida de otros comandos. Por ejemplo, puede utilizar un comando de filtro de texto (como el **egrep** que se explica a continuación) para que otro programa manipule la salida de texto (ordenar los datos seleccionados anteriormente). Esto se puede hacer con la ayuda de los operadores de redirecciónamiento, enviando la salida estándar del primer comando (**egrep**) y ejecutar el segundo programa (**sort**) tomando como dato el fichero creado anteriormente. Esta solución es poco elegante, ya que conlleva la creación de ficheros que luego si no los borramos pueden provocar desorden entre nuestros datos. La solución es el uso de **pipes** de datos (o **pipelines**).

El **pipeline** |, es un tipo de redirecciónamiento especial; con él se indica, que la salida obtenida de un comando, sirve como entrada de datos a otro comando. Es decir, los datos de salida los redireccionamos a otro comando.

Ejemplo:

La orden **ls** visualiza toda la información del directorio sin parar, por lo que algunas veces no llegamos a ver el contenido de todo el directorio. Como la orden **ls** no posee ninguna opción para paginar esta información, deberemos combinar la orden **ls** y la orden **more**, (que es la que pagina), para ello escribiremos: **\$ ls | more**

```
$egrep -i 'donostia' alumnos.txt | sort
```

Visualiza la lista de los alumnos que son de Donostia ordenados.

NOTA: Podemos utilizar el número de | que deseemos e incluso hacer que el resultado final se almacene en un fichero.

```
$egrep -i 'donostia' alumnos.txt | cut -d: -f1,2 | sort >aeg.txt
```

103.4.4. Comandos más utilizados con redirecciónamientos.

Para elegir líneas de los ficheros utilizaremos el comando grep, egrep o fgrep.

COMANDO: grep, egrep, fgrep

FORMATO: grep [-opciones] cadena ficheros

FUNCIÓN: Visualiza las líneas de los ficheros indicados que contengan la cadena indicada.

OPCIONES:

v Visualiza las líneas del fichero que no poseen la cadena indicada.

x Visualiza las líneas que poseen solo la cadena indicada.

w Visualiza las líneas que posee la cadena como palabra completa.

c Visualiza cuántas líneas posee la cadena a buscar.

n Visualiza además de las líneas que posee la cadena, el nº de la línea.

l Visualiza únicamente el nombre del fichero en caso de que la cadena exista en dicho fichero.

H En caso de realizar la búsqueda en varios ficheros, no visualiza el nombre del fichero delante de la línea que posee la cadena.

R Realiza una búsqueda en recursivo

y ó i No distingue entre mayúsculas y minúsculas.

f<fich> La cadena a buscar será el contenido del fichero indicado en <fich>.

e<exp> Se utiliza cuando se desea buscar una cadena que comience con "-", para no confundirlo con una opción.

```
Ejemplo: $ egrep 'aeg' ventas.txt
```

Hasta el momento la cadena a buscar era una cadena fija, es decir, buscábamos los caracteres especificados. Pero con estos comandos también podemos buscar cadenas que sean expresiones regulares.

La primera diferencia que existe entre estos 3 comandos es que **fgrep**, solo puede buscar cadenas fijas, en cambio **egrep** y **grep**, pueden buscar expresiones regulares. En caso de utilizar expresiones, es conveniente encerrar estas, entre apóstrofes (').

Los caracteres, (metacaracteres), que podemos utilizar para definir una expresión regular son los siguientes:

. Indica que en dicha posición puede existir cualquier carácter con uno de longitud.

^ Indica que la cadena se va a buscar al comienzo de la línea. Este carácter se coloca antes de la cadena a buscar.

\\$ Indica que la cadena a buscar se encuentra al final de la línea. El \\$ se coloca después de la cadena a buscar.

[] Dentro de los corchetes indicaremos:

Un rango de caracteres: buscará los caracteres correspondientes, a dicho rango.

Una lista de caracteres: buscará los caracteres correspondientes a dicha lista.

Si tras el corchete de abrir, "[", escribimos un circunflejo, "^", significa que buscará todos los

caracteres excepto los indicados o por el rango o por la lista de caracteres.

Si deseamos buscar varias cadenas en un fichero, utilizaremos el comando **egrep**, y para separar las cadenas utilizamos el pipe , |, ó el <intro>.

En caso de utilizar <intro>, en la nueva línea se visualizará el símbolo <, que significa que el comando no ha finalizado, en este momento teclearemos la 2^a cadena a buscar, cerraremos el apóstrofe y escribiremos el nombre del fichero.

Ejemplos

```
$ egrep '^a' aeg.txt  
$ egrep 'a$' aeg.txt  
$ egrep 'F[0-4]' aeg.txt  
$ grep 'F[^0-4]' aeg.txt  
$ grep 'F.L' aeg.txt  
$ grep 'F[abc]' aeg.txt
```

Para conseguir caracteres o campos de los ficheros utilizaremos el comando **cut**.

COMANDO: **cut**

FORMATO: **cut [-opciones] ficheros**

FUNCIÓN: Toma de los ficheros indicados, las columnas o los campos indicados.

OPCIONES:

c Tras esta opción, indicaremos los números de las columnas que queramos elegir. Los números de las columnas se pueden indicar separándolos por comas ó grupos de columnas separando las columnas mediante guiones. Cada carácter es una columna.

f Tras esta opción, indicaremos los números de campos, (de forma similar a las columnas). Hay que tener en cuenta, que el separador de campo por defecto es el tabulador.

d Tras esta opción indicaremos el carácter que deseamos sea el separador de campos.

Ejemplo: `$ cut -c1,4,7-9 aeg.txt`

Elige las columnas 1 y 4 y de la 7 a la 9. Sólo visualiza las columnas elegidas.

```
$ cut -f 1,2,7-9 aeg.txt
```

Elige los campos 1, 2 y del 7 al 9 ambos inclusive.

Para unir las líneas de varios ficheros utilizaremos el comando **paste**.

COMANDO: **paste**

FORMATO: **paste [-opciones] fichero1 fichero2**

FUNCIÓN: Visualiza 2 ó más ficheros lateralmente, es decir, coloca la 1^a línea del **fichero1** junto a la 1^a línea del **fichero2**. El elemento de unión es el tabulador.

OPCIONES:

d Tras la opción indicaremos un carácter, dicho carácter será la unión de los ficheros.

Ejemplo: `$ paste aeg11.txt aeg12.txt`

```
$ paste -d* aeg11.txt aeg12.txt
```

Para conocer los caracteres, palabras y líneas de los ficheros utilizaremos el comando **wc**.

COMANDO: **wc**

FORMATO: **wc [-opciones] ficheros**

FUNCIÓN: Cuenta líneas, palabras y caracteres de los ficheros indicados.

OPCIONES:

- l** Cuenta el nº de líneas
- w** Cuenta el nº de palabras
- c** Cuenta el nº de caracteres

Si no indicamos las opciones, visualiza el nº de líneas, de palabras y de caracteres en este orden. Al indicar las opciones, visualiza el nº de las opciones en el orden indicado.

Ejemplo: `$ wc aeg.txt`

Visualiza el nº de líneas, de palabras, y de caracteres del fichero **aeg.txt**.

`$ wc -c aeg.txt`

Visualiza solo, el nº de caracteres del fichero **aeg.txt**

`$ wc aeg1.txt aeg2.txt`

Visualiza 3 líneas; en la primera línea, visualiza el nº de líneas, palabras y caracteres del fichero **aeg1.txt**, en la 2ª línea la misma información pero del fichero **aeg2.txt**, y en la 3ª línea el total de líneas, palabras y caracteres de los 2 ficheros.

Para ordenar las líneas de los ficheros utilizaremos el comando sort.

COMANDO: sort

FORMATO: `sort [-opciones] ficheros`

FUNCIÓN: Ordenar el contenido de los ficheros indicados.

OPCIONES:

- o** Tras esta opción escribiremos el nombre del fichero en el cual se grabará el fichero ordenado.
- r** Ordena de mayor a menor.
- t** Tras esta opción indicaremos el carácter que hemos considerado como separador de campos. El separador de campos por defecto es el espacio.
- g** Trata los números como tales.
- +n1 -n2** Si en lugar de tomar como campo a ordenar toda la línea, queremos hacerlo por un campo en particular, tenemos que especificarlo indicando **+n1** indica los campos a saltar y **-n2** indica a partir de que campo se ignora la línea.

Ejemplos: `$sort aeg11.txt aeg12.txt`

Ordena tomando los contenidos de los 2 ficheros y visualiza una única salida común, es decir, no ordena primero el contenido del **aeg11.txt** y luego el contenido del **aeg12.txt**.

`$sort -r aeg.txt`

Ordena de mayor a menor el contenido del fichero **aeg.txt**.

`$sort +1 -3 aeg.txt`

Ordena el contenido del fichero **aeg.txt** tomando para ordenar los campos 1 y 2 sabiendo que el separador de campos sea el espacio.

OPCIONES:

- o** Tras esta opción escribiremos el nombre del fichero en el cual se grabará el fichero ordenado.
- r** Ordena de mayor a menor.
- t** Tras esta opción indicaremos el carácter que hemos considerado como separador de campos.

El separador de campos por defecto es el espacio.

g Trata los números como tales.

+n1 -n2 Si en lugar de tomar como campo a ordenar toda la línea, queremos hacerlo por un campo en particular, tenemos que especificarlo indicando **+n1** indica los campos a saltar y **-n2** indica a partir de que campo se ignora la línea.

Ejemplos: `$sort aeg11.txt aeg12.txt`

Ordena tomando los contenidos de los 2 ficheros y visualiza una única salida común, es decir, no ordena primero el contenido del **aeg11.txt** y luego el contenido del **aeg12.txt**.

`$sort -r aeg.txt`

Ordena de mayor a menor el contenido del fichero **aeg.txt**.

`$sort +1 -3 aeg.txt`

Ordena el contenido del fichero **aeg.txt** tomando para ordenar los campos 1 y 2 sabiendo que el separador de campos sea el espacio.

103.5. Crear, monitorizar y finalizar procesos.

Peso en el examen de certificación: 4 puntos.

Objetivo: Realizar tareas de gestión de procesos.

Conceptos y áreas de conocimiento:

- Ejecutar trabajos en primer y segundo plano.
- Envío de señal a un programa para continuar la ejecución después del cierre de sesión (logout).
- Monitorización activa de procesos.
- Seleccionar y clasificar procesos para su visualización en pantalla.
- Envío de señales a procesos.

Términos y utilidades

- &
- bg
- fg
- jobs
- kill
- nohup
- ps
- top
- free
- uptime

103.5.1. Introducción.

El objetivo principal es la gestión básica de procesos en Linux. Los principales apartados del mismo son los siguientes:

- Ejecutar trabajos en primer plano (foreground) y segundo plano (background).
- Hacer que un programa continúe su ejecución tras haber cerrado la sesión de un terminal.
- Monitorizar los procesos activos.
- Seleccionar y ordenar procesos para su visualización.
- Enviar señales a los procesos.

Algunos términos y comandos que manejarás al finalizar este tema son:

&	bg	fg	jobs	kill	killall
---	----	----	------	------	---------

nohup	ps	top	free	uname	uptime
-------	----	-----	------	-------	--------

103.5.2. Monitorización de procesos:

El término proceso hace referencia a un programa que está en ejecución. Generalmente también empleamos el término “tarea” para referirnos a un proceso.

Linux es un sistema operativo multitarea y multiusuario. Esto quiere decir que múltiples procesos pueden operar simultáneamente sin interferirse unos con los otros. Cada proceso cree que es el único en el sistema pero en realidad está compitiendo con el resto por obtener los recursos.

Debes diferenciar entre proceso y programa. Imagina que varios usuarios están visionando documentos con un procesador de textos. Cada instancia del programa de procesador de textos es un proceso separado. Nosotros distinguimos cada proceso por un número entero positivo de identificación conocido como PID (Process Identifier). Si además un proceso ha sido creado por otro decimos que este otro es “su padre” y podemos obtener su identificador mediante el atributo PPID(Parent Process Identifier). De esta forma los procesos pueden estar relacionados mediante una relación jerárquica de tipo árbol. De hecho existe un primer proceso que inicia los demás llamado init que corresponde al sistema operativo.

Además del PID y en caso de haberlo del PPID los procesos iniciados por un usuario cuentan con un UID(User Identifier) y si éstos han sido ejecutados en un terminal de comandos o consola cuentan con un valor de TTY (o bien aparecerá en esta columna un ? indicando que no tiene terminal).

Prácticamente todo lo que se está ejecutando en el sistema en cualquier momento es un proceso, incluyendo el shell, el entorno gráfico (que habitualmente son varios procesos), etc. La excepción a lo anterior es el kernel o corazón de Linux, que es un conjunto de rutinas que residen en memoria y a los cuales los procesos pueden acceder mediante llamadas al sistema.

103.5.2.1. Comando uname:

Antes de comenzar a analizar y monitorizar los procesos conviene hablar del kernel, núcleo de todo sistema Linux. Podemos averiguar información del kernel con el comando uname cuyas opciones básicas son las siguientes:

Versión corta	Versión Larga	Información generada
-n	--nodename	Nombre del equipo en la red
-s	--kernel-name	Nombre del kernel (suele ser simplemente Linux)
-v	--kernel-version	Fecha y hora de compilación del kernel. Curiosamente no suele aportar la versión real.
-r	--kernel-release	Número de la versión real del kernel.
-p	--procesor	Información del procesador (modelo, velocidad, fabricante). En muchos sistemas sólo se obtiene <i>unkown</i> .

-i	--hardware-platform	Información sobre el hardware, a menos que devuelva <i>unkown</i> .
-o	--operating-system	Información del sistema operativo. Con frecuencia devuelve GNU/Linux.
-a	--all	Información completa del comando uname.

Los ejemplos de este tema han sido probados en una distribución Debian 7.1 de Linux. Ejecutemos uname -a para ver toda la información:

```
root@Master01:~# uname -a
```

```
Linux Master01 3.2.0-4-amd64 #1 SMP Debian 3.2.46-1+deb7u1 x86_64 GNU/Linux
```

103.5.2.2. Comando ps:

Para la monitorización de procesos el comando más utilizado es ps (literalmente significa “process status” - estado de los procesos). Sirve para listar por pantalla los procesos activos. Este comando mostrará por pantalla listas de procesos y recuerda que esta salida puedes redirigirla a un fichero si te interesa guardar la información.

El comando ps acepta una cantidad elevada de opciones, cada una de ellas para mostrar datos concretos. En general existen tres tipos de sintaxis para escribir estas opciones:

- Opciones de Unix98: comienzan por un guión (-) y tienen un único carácter. Se pueden agrupar.
- Opciones de BSD: también tienen un único carácter y se pueden agrupar pero no comienzan por guión.
- Opciones GNU largas: comienzan por doble guión (--), son palabras y nunca las agruparemos.

Cuando hablamos de agrupar opciones nos referimos a que es equivalente escribir ps -a -f que escribir ps -af. Recuerda que si tienes dudas con las opciones comunes del comando puedes emplear ps --help o bien utilizar el comando man ps.

En cualquier caso siempre obtendremos una tabla cuyas columnas podrán tener los siguientes rótulos:

p o PID	Process ID, número único o de identificación del proceso.
P o PPID	Parent Process ID, padre del proceso.
U o UID	User ID, usuario propietario del proceso.
t o TT o TTY	Terminal asociada al proceso, si no hay terminal aparece entonces un '?'.
T o TIME	Tiempo de uso de CPU acumulado por el proceso.

c o CMD	El nombre del programa o comando que inició el proceso.
RSS	Resident Sise, tamaño de la parte residente en memoria en kilobytes.
SZ o SIZE	Tamaño virtual de la imagen del proceso.
NI	Valor nice (prioridad) del proceso, un número positivo significa menos prioridad y un número negativo más prioridad (valores de -19 a 19).
C o PCPU	Porcentaje de tiempo de procesador utilizado por el proceso.
STIME	Starting Time, hora de inicio del proceso.
S o STAT	Estado del proceso. Los valores pueden ser: <ul style="list-style-type: none"> • R (runnable), en ejecución, corriendo o ejecutándose. • S (sleeping), proceso en ejecución sin actividad por el momento, o esperando por algún evento para continuar. • T (topped), proceso detenido totalmente, pero puede ser reiniciado. • Z (zombie), proceso que por alguna razón no terminó de manera correcta, no deberíamos tener procesos zombies. • D (uninterruptible sleep), son procesos generalmente asociados a acciones de entrada/salida del sistema. • X (dead), muerto, proceso terminado pero que sigue apareciendo. Al igual que los Z no deberían verse nunca.

La forma más básica de uso de ps consiste en indicarle un conjunto de cero o más PID como argumentos y obtendremos una salida mostrando el estado de los procesos especificados. Ten en cuenta que si no especificamos PID se mostrarán todos los procesos ejecutados desde nuestra terminal y no todos los del sistema. Para mostrar todos los procesos el sistema recurriremos a opciones como **-a** y **-e**.

Ejemplo 1: mostrar los procesos asociados a nuestra consola (que en este caso se llamará **terminal controlador**).

```
root@Master01:~# ps
  PID TTY      TIME CMD
3683 pts/2  00:00:00 bash
4038 pts/2  00:00:00 sleep
4039 pts/2  00:00:00 ps
```

Observa que en este ejemplo los procesos que aparecen son bash (es nuestra ventana de terminal o consola), el proceso sleep y el propio comando ps. Para cada proceso podemos observar su PID. También observa que todos los procesos se ejecutan en el terminal pts/2. Antes de ver las opciones más utilizadas mostremos un poco de ayuda con **--help**:

```
root@Master01:~# ps --help
```

Usage:

```
ps [options]
```

Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.

For more details see ps(1).

Comprobad que la salida es diferente a la de la versión web que se ejecuta en una distribución de Fedora antigua (la última revisión, 19 al escribir este manual, soporta el mismo formato). Ejecutamos la ayuda completa.

```
root@Master01:~# ps --help all
```

Usage:

```
ps [options]
```

Basic options:

-A, -e	all processes
-a	all with tty, except session leaders
a	all with tty, including other users
-d	all except session leaders
-N, --deselect	negate selection
r	only running processes
T	all processes on this terminal
x	processes without controlling ttys

Selection by list:

-C <command>	command name
-G, --Group <gid>	real group id or name
-g, --group <group>	session or effective group name
-p, --pid <pid>	process id
--ppid <pid>	select by parent process id
-s, --sid <session>	session id
-t, t, --tty <tty>	terminal
-u, U, --user <uid>	effective user id or name
-U, --User <uid>	real user id or name

selection <arguments> take either:

comma-separated list e.g. '-u root,nobody' or
blank-separated list e.g. '-p 123 4567'

Output formats:

- F extra full
- f full-format, including command lines
- f, --forest ascii art process tree
- H show process hierarchy
- j jobs format
- j BSD job control format
- l long format
- l BSD long format
- M, Z add security data (for SE Linux)
- O <format> preloaded with default columns
- O <format> as -O, with BSD personality
- o, o, --format <format>
 - user defined format
- s signal format
- u user-oriented format
- v virtual memory format
- X register format
- y do not show flags, show rrs vs. addr (used with -l)
- context display security context (for SE Linux)
- headers repeat header lines, one per page
- no-headers do not print header at all
- cols, --columns, --width <num>
 - set screen width
- rows, --lines <num>
 - set screen height

Show threads:

- H as if they where processes
- L possibly with LWP and NLWP columns
- m, m after processes

-T possibly with SPID column

Miscellaneous options:

-c show scheduling class with -l option
c show true command name
e show the environment after command
k, --sort specify sort order as: [+|-]key[,[+|-]key[,...]]
L list format specifiers
n display numeric uid and wchan
S, --cumulative include some dead child process data
-y do not show flags, show rss (only with -l)
-V, V, --version display version information and exit
-w, w unlimited output width

--help <simple|list|output|threads|misc|all>
display help and exit

For more details see ps(1).

Ejemplo 2:mostrar todos los procesos del sistema (**opción -e**):

```
root@Master01:~# ps -e
```

PID	TTY	TIME	CMD
1 ?		00:00:01	init
2 ?		00:00:00	kthreadd
3 ?		00:00:00	ksoftirqd/0
5 ?		00:00:00	kworker/u:0
6 ?		00:00:00	migration/0
7 ?		00:00:00	watchdog/0
8 ?		00:00:00	cpuset
9 ?		00:00:00	khelper
10 ?		00:00:00	kdevtmpfs
11 ?		00:00:00	netns
12 ?		00:00:00	sync_supers
13 ?		00:00:00	bdi-default
14 ?		00:00:00	integrityd

```
15 ? 00:00:00 kblockd
17 ? 00:00:00 khungtaskd
```

(...) Lista cortada

Ejemplo 3:mostrar todos los procesos (**-e**) con más columnas de información (**opción -f**):

```
root@Master01:~# ps -ef
UID      PID  PPID  C STIME TTY      TIME CMD
root      1    0  0 22:47 ?    00:00:01 init [2]
root      2    0  0 22:47 ?    00:00:00 [kthreadd]
root      3    2  0 22:47 ?    00:00:00 [ksoftirqd/0]
root      5    2  0 22:47 ?    00:00:00 [kworker/u:0]
root      6    2  0 22:47 ?    00:00:00 [migration/0]
root      7    2  0 22:47 ?    00:00:00 [watchdog/0]
root      8    2  0 22:47 ?    00:00:00 [cpuset]
root      9    2  0 22:47 ?    00:00:00 [khelper]
root     10    2  0 22:47 ?    00:00:00 [kdevtmpfs]
root     11    2  0 22:47 ?    00:00:00 [netns]
root     12    2  0 22:47 ?    00:00:00 [sync_supers]
root     13    2  0 22:47 ?    00:00:00 [bdi-default]
```

Ejemplo 4: mostrar todos los procesos con más información extra (**opción -F**):

```
root@Master01:~# ps -eF
UID      PID  PPID  C   SZ RSS PSR STIME TTY      TIME CMD
root      1    0  0 2662 832  0 22:47 ?    00:00:01 init [2]
root      2    0  0    0  0 0 22:47 ?    00:00:00 [kthreadd]
root      3    2  0    0  0 0 22:47 ?    00:00:00 [ksoftirqd/0]
root      5    2  0    0  0 0 22:47 ?    00:00:00 [kworker/u:0]
root      6    2  0    0  0 0 22:47 ?    00:00:00 [migration/0]
root      7    2  0    0  0 0 22:47 ?    00:00:00 [watchdog/0]
root      8    2  0    0  0 0 22:47 ?    00:00:00 [cpuset]
root      9    2  0    0  0 0 22:47 ?    00:00:00 [khelper]
root     10    2  0    0  0 0 22:47 ?    00:00:00 [kdevtmpfs]
root     11    2  0    0  0 0 22:47 ?    00:00:00 [netns]
root     12    2  0    0  0 0 22:47 ?    00:00:00 [sync_supers]
root     13    2  0    0  0 0 22:47 ?    00:00:00 [bdi-default]
```

Ejemplo 5: si queremos mostrar los usuarios recurriremos a la opción **-u**. También podremos filtrar

el listado por uno o varios usuarios concretos. Usaremos -u usuario, U usuario o bien --User usuario. Cuando decimos usuario nos referimos a un UID o bien a una cadena con el nombre del usuario. Por ejemplo ps -u operador mostrará los procesos del usuario operador.

Ten en cuenta que la opción BSD u minúsculas (sin guión) es empleada para mostrar información adicional (típicamente la columna USER). Es decir si hacemos ps U operador mostraremos los procesos del usuario operador pero puede que no aparezca la columna USER. Si quieres mostrar esta columna puedes hacer ps u U operador.

```
root@Master01:~# ps u -u operador
USER    PID %CPU %MEM   VSZ RSS TTY STAT START TIME COMMAND
operador 3427  0.0  0.2 123440 4324 ?      Sl 22:48 0:00 /usr/bin/gnome-
operador 3445  0.0  0.5 210148 10976 ?      Ssl 22:48 0:00 x-session-manag
operador 3485  0.0  0.0 12384 332 ?      Ss 22:48 0:00 /usr/bin/ssh-ag
operador 3488  0.0  0.0 24184 588 ?      S 22:48 0:00 /usr/bin/dbus-l
operador 3489  0.0  0.0 31396 1920 ?      Ss 22:48 0:00 /usr/bin/dbus-d
operador 3497  0.0  0.9 454832 18540 ?      Sl 22:48 0:00 /usr/lib/gnome-
operador 3505  0.0  0.1 64404 2724 ?      S 22:48 0:00 /usr/lib/gvfs/g
operador 3515  0.0  0.2 227080 4652 ?      S<l 22:48 0:00 /usr/bin/pulsea
operador 3518  0.0  0.2 163612 4416 ?      S 22:48 0:00 /usr/lib/gvfs/g
operador 3524  0.0  0.1 60496 2488 ?      S 22:48 0:00 /usr/lib/gvfs/g
operador 3526  0.0  0.1 76872 2620 ?      Sl 22:48 0:00 /usr/lib/gvfs/g
```

Ejemplo 6: Mostremos todos los procesos (opción BSD a) sin terminal asociado (opción BSD x):

```
root@Master01:~# ps ax
PID TTY    STAT TIME COMMAND
 1 ?    Ss  0:01 init [2]
 2 ?    S  0:00 [kthreadd]
 3 ?    S  0:00 [ksoftirqd/0]
 5 ?    S  0:00 [kworker/u:0]
 6 ?    S  0:00 [migration/0]
 7 ?    S  0:00 [watchdog/0]
 8 ?    S< 0:00 [cpuset]
 9 ?    S< 0:00 [khelper]
10 ?    S  0:00 [kdevtmpfs]
11 ?    S< 0:00 [netns]
12 ?    S  0:00 [sync_supers]
13 ?    S  0:00 [bdi-default]
```

Ejemplo 7: Y ahora todos los procesos (opción BSD a) sin terminal (opción BSD x) con la columna

usuario (opción BSD u, que sirve para mostrar la columna USER y otras columnas):

```
root@Master01:~# ps aux
USER      PID %CPU %MEM   VSZ   RSS TTY      STAT START  TIME COMMAND
root      1  0.0  0.0 10648  832 ?        Ss  22:47  0:01 init [2]
root      2  0.0  0.0     0   0 ?        S   22:47  0:00 [kthreadd]
root      3  0.0  0.0     0   0 ?        S   22:47  0:00 [ksoftirqd/0]
root      5  0.0  0.0     0   0 ?        S   22:47  0:00 [kworker/u:0]
root      6  0.0  0.0     0   0 ?        S   22:47  0:00 [migration/0]
root      7  0.0  0.0     0   0 ?        S   22:47  0:00 [watchdog/0]
root      8  0.0  0.0     0   0 ?        S<  22:47  0:00 [cpuset]
root      9  0.0  0.0     0   0 ?        S<  22:47  0:00 [khelper]
root     10  0.0  0.0     0   0 ?        S   22:47  0:00 [kdevtmpfs]
root     11  0.0  0.0     0   0 ?        S<  22:47  0:00 [netns]
root     12  0.0  0.0     0   0 ?        S   22:47  0:00 [sync_supers]
root     13  0.0  0.0     0   0 ?        S   22:47  0:00 [bdi-default]
root     14  0.0  0.0     0   0 ?        S<  22:47  0:00 [kintegrityd]
```

Ejemplo 8: ¿Te interesa obtener una salida con una lista de columnas concreta? Puedes utilizar -o indicando la lista de columnas separada por comas. En este ejemplo obtendremos todos los procesos (-e) con las columnas (-o) usuario, pid y terminal:

```
root@Master01:~# ps -eo user,pid,tty
```

```
USER      PID TT
root      1 ?
root      2 ?
root      3 ?
root      5 ?
root      6 ?
root      7 ?
root      8 ?
root      9 ?
root     10 ?
root     11 ?
root     12 ?
```

Ejemplo 9: Sólo vamos a aclarar un aspecto. La opción -e muestra todos los procesos del sistema como vimos en el ejemplo anterior. Sin embargo la opción -a muestra todos los procesos pero de nuestra sesión (todas nuestras terminales).

```
root@Master01:~# ps -ao user,pid,tty
```

```
USER      PID TT
root      4191 pts/2
root@Master01:~#
```

Ejemplo 10: Para mostrar el árbol jerárquico de los procesos tenemos las opciones -H, -f y --forest. Podríamos ejecutar ps -u operador --forest. En muchos sistemas tenemos un comando llamado pstree que también muestra el árbol de relaciones. Típicamente con pstree se usan las opciones -A y -G para obtener un árbol con líneas Ascii. La opción -u mostrará además el usuario entre paréntesis. Ejemplo de uso: pstree -AGu.

```
root@Master01:~# pstree -AGu
init——NetworkManager——{NetworkManager}
    |————accounts-daemon——{accounts-daemon}
    |————acpid
    |————apache2——apache2(www-data)
    |           |————2*[apache2(www-data)——26*[{apache2}]]
    |————atd(daemon)
    |————avahi-daemon(avahi)——avahi-daemon
    |————bluetoothd
    |————colord(colord)——{colord}
    |————colord-sane(colord)——2*[{colord-sane}]
    |————console-kit-dae——64*[{console-kit-dae}]
    |————cron
    |————cupsd
    |————daemon——mpt-statusd——sleep
    |————dbus-daemon
    |————dbus-daemon(messagebus)
    |————dbus-daemon(operator)
    |————dbus-launch
    |————dbus-launch(operator)
```

Utilizando ps es habitual emplear ps -eH. Prueba estas opciones en tu consola. Aquí te mostramos un listado con las opciones -j (jobs o trabajos de los que hablaremos más adelante) y --forest.

```
root@Master01:~# ps -j --forest
  PID  PGID   SID TTY      TIME CMD
3683  3683  3683 pts/2  00:00:00 bash
4202  4202  3683 pts/2  00:00:00 \_ ps
root@Master01:~#
```

Otros ejemplos: Un problema que puede presentarse en muchas terminales es que el tamaño máximo de las líneas soportadas es de 80 caracteres, y si estamos mostrando muchas columnas

podemos obtener líneas truncadas. Puedes redirigir la salida a un fichero e indicar mediante las opciones `-w` o `w` que se evite este truncamiento. El ejemplo sería `ps -w > procesos.txt`.

Por otra parte puede que necesites obtener los PID de los procesos creados por algún comando concreto. Imagina que quieras obtener los procesos de todas las instancias de bash. Utiliza una tubería (pipe) y el comando grep: `ps ax | grep bash`.

Las opciones `-f` (full) y `-l` (longitud) permiten controlar la cantidad de información de salida.

```
root@Master01:~# ps -f
UID      PID  PPID  C STIME TTY      TIME CMD
root    3683  3678  0 22:50 pts/2  00:00:00 -bash
root    4214  3683  0 23:58 pts/2  00:00:00 ps -f
```

103.5.2.3. Repaso de ps y uso para ordenar procesos en pantalla:

Ya hemos visto usos del comando ps. Las opciones que soportan son muchas pero habitualmente usaremos unas más que otras. Recuerda que ps mostrará por defecto sólo la lista de procesos que fueron ejecutados desde un terminal de tu sesión. Para ver todos nuestros procesos con sus correspondientes terminales controladores utilizaremos la opción `-a`.

Por otra parte la opción `-x` mostrará procesos que no tienen un terminal controlador. Y la opción `-e` mostrará información para todos (every) los procesos (similar a `-a`). Hay opciones que incrementan la cantidad de información obtenida, como `-f` y `-l`

El terminal controlador se muestra en la columna TTY.

Para hacer el siguiente ejemplo hemos cambiado a un nuevo terminal, luego tendremos (pts/0) y (pts/1). Será la opción `-a` la que nos permite ver los procesos de nuestra sesión, aunque estén en terminales diferentes.

```
root@Master01:~# ps -af
UID      PID  PPID  C STIME TTY      TIME CMD
root    4227  3674 13 00:02 pts/1  00:00:05 iceweasel
root    4280  3683  0 00:02 pts/2  00:00:00 ps -af
```

Como ves en el código anterior iceweasel se está ejecutando en pts/1 mientras que ps está en pts/2.

Continuando con el repaso: si quieras seleccionar aquellos procesos de un usuario particular del sistema utilizarás la opción `-u`. Y si quieras mostrar los procesos correspondientes a un comando concreto puedes utilizar la opción `-c`.

Por último, la opción `-o` permite indicar exactamente las columnas que deseamos en el listado. Por ejemplo, si quisieramos obtener una lista de procesos que corresponden al comando iceweasel con las columnas usuario,PID,terminal,tiempo de CPU y comando haríamos:

```
root@Master01:~# ps -C iceweasel -o user,pid,tty,comm
USER      PID TT      COMMAND
root    4227 pts/1  iceweasel
```

Utilizando ps para ordenar procesos en pantalla: En ocasiones querrás ordenar la salida por algunos campos particulares. Esto puede hacerse con la opción `--sort` y a continuación especificar los campos. El orden por defecto es ascendente (+) pero podemos especificar que sea descendente en algún campo anteponiendo un signo menos (-).

El siguiente listado muestra todos los procesos de nuestra sesión (-a) utilizando el formato de trabajos o jobs (-j). La salida está ordenada por la sesión (columna SID – session ID) en orden descendente y por el nombre del comando en orden ascendente

```
root@Master01:~# ps -aj --sort -sid,+comm
 PID PGID SID TTY      TIME CMD
 4287 4287 3683 pts/2  00:00:00 ps
 4227 4227 3674 pts/1  00:00:06 iceweasel
```

Ahora un listado similar al anterior pero queremos los dos órdenes ascendentes: dado que este es el comportamiento por defecto no hará falta indicarlo.

```
root@Master01:~# ps -aj --sort sid,comm
 PID PGID SID TTY      TIME CMD
 4227 4227 3674 pts/1  00:00:07 iceweasel
 4290 4290 3683 pts/2  00:00:00 ps
```

Como siempre te recomendamos que consultes las páginas del comando man para ver detalles particulares del comando ps, así como las opciones para especificar campos concretos. También tendrás un resumen corto utilizando ps --help.

103.5.2.4. Comando free:

El comando free muestra la cantidad de memoria disponible (free) y utilizada (used) en nuestro sistema. Por defecto muestra los datos en KiloBytes pero podemos modificar esta apariencia usando las opciones -b (para bytes), -k (para kilobytes), -m (para megabytes) o -g (para gigabytes). La opción -t muestra una línea con los totales y la opción -s hará que los valores se actualicen con la frecuencia especificada que será un número en segundos pero ojo, en formato de punto flotante. Veamos dos ejemplos:

Ejemplo 1: uso de free sin parámetros: los datos aparecerán en KiloBytes.

```
root@Master01:~# free
      total    used    free   shared  buffers   cached
Mem:  2054132  947156 1106976       0   63296  443248
-/+ buffers/cache:  440612 1613520
Swap:  901116      0  901116
```

Ejemplo 2: mostrar la memoria en MegaBytes y con una línea de totales.

```
root@Master01:~# free -mt
      total    used    free   shared  buffers   cached
Mem:  2005      924    1081       0      61     432
-/+ buffers/cache:  430    1575
Swap:  879      0    879
Total: 2885    924   1961
```

103.5.2.5. Comando uptime:

El comando uptime muestra en una única línea la hora actual, cuánto tiempo ha estado el sistema en ejecución, cuántos usuarios tienen actualmente sesión iniciada y la carga media del sistema en los últimos 1 minuto, 5 minutos y 15 minutos. Observa el listado de ejemplo:

```
root@Master01:~# uptime  
00:13:19 up 1:25, 3 users, load average: 0,00, 0,01, 0,05
```

En relación a la carga media, si los procesos no demandan CPU la carga media será cero. Si hay procesos haciendo uso intensivo de la CPU la carga media será uno. Si el valor es mayor que uno significa que hay procesos compitiendo por el procesador. Si el valor fuera muy elevado podría significar que hay algún proceso colgado, en ese caso utiliza top para analizar la situación.

103.5.2.6. Comando top:

Los procesos son entidades dinámicas: cambian de estado, consumen recursos, crean procesos hijo, mueren. Si ejecutamos varias veces el comando ps podríamos ir viendo los cambios pero esto resulta engorroso. El comando top nos permite ver dinámicamente qué está pasando con nuestros procesos. Lo que hace es mostrar una lista constantemente actualizada de procesos con información útil de los mismos. El siguiente es un listado de ejemplo de uso de top. Dado que es un comando dinámico será necesario detenerlo para volver a la consola: teclea **q** para finalizar.

```
top - 00:15:33 up 1:27, 3 users, load average: 0,00, 0,01, 0,05  
Tasks: 132 total, 1 running, 131 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0,0 us, 0,3 sy, 0,0 ni, 99,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st  
KiB Mem: 2054132 total, 955952 used, 1098180 free, 63336 buffers  
KiB Swap: 901116 total, 0 used, 901116 free, 443348 cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3678	root	20	0	92112	4060	3176	S	0,3	0,2	0:00.37	sshd
1	root	20	0	10648	832	696	S	0,0	0,0	0:01.05	init
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.12	ksoftirqd/0
5	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/u:0
6	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
7	root	rt	0	0	0	0	S	0,0	0,0	0:00.02	watchdog/0
8	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	cpuset
9	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	khelper
10	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
11	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	netns
12	root	20	0	0	0	0	S	0,0	0,0	0:00.01	sync_supers
13	root	20	0	0	0	0	S	0,0	0,0	0:00.00	bdi-default

14	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kintegrityd
15	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kblockd
17	root	20	0	0	0	0	S	0,0	0,0	0:00.00	khungtaskd
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kswapd0

El comando top, mientras está en ejecución, cuenta con un número de subcomandos. Los más prácticos son los siguientes:

h y ? nos ofrece ayuda (help).

q nos permite detener el comando top (quit).

k nos permite matar un proceso (kill). Nos pedirá el PID del mismo.

r nos permite cambiar la prioridad de un proceso (renice). También pedirá un PID.

s nos permite cambiar la velocidad (speed) de refresco de los datos. Nos pedirá el nuevo valor en segundos. Por defecto top se actualiza cada 5 segundos.

p provoca que los datos se ordenen por el uso de CPU. Por defecto este es el comportamiento habitual de top.

m es similar a p pero provoca que los datos se ordenen por el uso de memoria.

f nos permite agregar o eliminar campos del listado

o ordena el listado

F selecciona campos para realizar la ordenación.

Consulta las páginas del comando man para ver detalles de las opciones de uso de top incluyendo cómo ordenar por uso de memoria o cualquier otro criterio. No obstante ya te habrás dado cuenta de que este comando resulta muy útil para detectar procesos que pueden estar “colgados” requiriendo mucho uso de CPU o de memoria.

Las opciones anteriores se utilizan una vez que estamos ejecutando top. Si directamente quieres ejecutar este comando con alguna modificación tienes las opciones de comando. Veamos algunas (no están todas):

top -d retardo Especifica el tiempo de refresco entre actualizaciones. Tendrás que especificar el valor de retardo en segundos. Habitualmente este valor es 5.

top -p PID La opción -p es muy útil cuando quieres monitorizar sólo algunos procesos específicos. Puedes usar esta opción hasta 20 veces. Ej: top -p 3812 -p 1529

top -u usuario La opción -u es muy útil cuando quieres monitorizar sólo los procesos específicos de un usuario. Ej: top -u Gonzalo mostrará los procesos del usuario Gonzalo

top -n veces La opción -n indica que el comando se actualizará n veces y luego terminará su ejecución.

top -b La opción -b sirve para especificar el modo por lotes, en el que top no emplea los comandos normales de actualización de la pantalla. Se puede usar, por ejemplo, para registrar en un fichero el uso de CPU de algún proceso.

103.5.2.7. Comando jobs:

Es un comando algo menos utilizado que ps pero en ocasiones muy útil. El término jobs (trabajos) hace referencia a los procesos que dependen de un terminal. Linux, internamente, establece un

identificador para cada uno de estos trabajos, que corresponde con un número entero que comienza en 1. Recuerda: sólo son procesos asociados al terminal controlador desde el que ejecutamos. Observa el siguiente listado:

```
root@Master01:~# jobs -l
[1] 4325 Ejecutando      sleep 120 &
[2]- 4326 Ejecutando      yes > /dev/null &
[3]+ 4327 Hecho          ls -R / > /dev/null
```

La primera columna es el número de trabajo. En este ejemplo tenemos tres (el 1,2 y el 3). El signo más junto al trabajo 3 indica que en este momento era el que estaba ejecutándose. Hemos utilizado la opción `-l` para ver más información, concretamente la segunda columna con los PID (lo que nos permitiría utilizar `ps` con estos procesos). A continuación hay una columna con el estado (Running o Ejecutando en el ejemplo). Por último tenemos el comando de cada proceso.

Este es uno de los usos de `jobs`: averiguar los PID de los procesos de nuestro terminal para luego usarlos con `ps` o matar estos procesos. También hay muchas utilidades en Linux que utilizan el número de tarea en lugar del PID, y `jobs` nos ayuda a obtenerlo. Por último `jobs` tiene otra utilidad: si estamos trabajando con una sesión remota al cerrar la sesión podemos dejar colgados procesos en el terminal remoto. Con `jobs` podemos comprobarlo y cerrar estos procesos que han quedado sin control.

Si ejecutamos `jobs -p` obtendremos como salida simplemente la lista de PID de procesos correspondientes a cada trabajo.

```
root@Master01:~# jobs -p
4325
4326
```

Podemos usar esta salida como entrada del comando `ps`, tal y como se muestra en el siguiente código:

```
root@Master01:~# ps $(jobs -p)
 PID TTY      STAT   TIME COMMAND
 4326 pts/2    R      0:58 yes
[1]- Hecho      sleep 120
```

103.5.3. Procesos en primer plano y en segundo plano:

Imagina que abres un terminal y ejecutas algún comando dinámico como `top`. En ese momento descubrirás que ya no tenemos control sobre el terminal porque el programa lanzado está monopolizándola. Estos procesos se dice que están ejecutándose en primer plano (foreground) y paralizan el terminal hasta que finalicen.

Para poder volver a tener la consola disponible para otros comandos nos haría falta finalizar el proceso que está en foreground. También podemos pausarlo pulsando Control-Z, pero el proceso quedaría suspendido sin realizar su tarea y volvemos a tener el terminal operativo. Probémoslo:

Ejecuta `top` en un terminal. Se está ejecutando en primer plano, con lo que el terminal queda inutilizado, no puedes escribir otros comandos. Podríamos terminar `top` (recuerda que para eso basta con escribir `q`), pero no queremos finalizar el proceso sino utilizar el terminal para otros comandos. Entonces pausaremos `top` escribiendo: Control-Z. Aparecerá un mensaje indicando que `top` ha sido

detenido (estado Stopped) y tiene un número de tarea 1. La consola está disponible (y top pausado). Escribe el comando ls para listar el contenido de tu carpeta. A continuación escribe el comando fg que volverá a traer el proceso top a primer plano. Termina esta prueba saliendo de top con la opción q.

¿Entonces ocurre que en un terminal sólo podremos ejecutar un proceso cada vez, o bien tendremos que pausarlo con Ctrl-Z para ejecutar otro? La respuesta es no. Existe la ejecución de procesos en segundo plano (background). Los procesos que estén en segundo plano se estarán ejecutando con el resto de procesos del sistema, y estarán realizando su tarea sin que nosotros aparentemente lo advirtamos (a menos, claro, que requieran una entrada de datos), dejando el terminal libre para otras actividades. Colocar un proceso en segundo plano es útil si se trata de procesos que hacen uso intensivo de la CPU y requieren muy poca información de entrada.

En este apartado veremos cómo colocar procesos en primer y segundo plano.

103.5.3.1. Uso de &:

Si al final de un comando escribimos el símbolo & estaremos ejecutándolo directamente en background. Observa este código: ejecutamos el comando yes (y lo redirigimos a /dev/null para que no haga nada) y por otra parte ejecutamos el comando ls con -laR para listar todos los ficheros y guardarlos en misarchivos.txt. Ambos procesos en background

```
root@Master01:~# yes > /dev/null &
[1] 4340
root@Master01:~# ls -laR > misarchivos.txt &
[2] 4341
root@Master01:~# jobs
[1]- Ejecutando      yes > /dev/null &
[2]+ Hecho            ls -laR > misarchivos.txt
```

Los dos comandos, yes y ls, se han ejecutado en background. Por lo tanto la consola la hemos tenido operativa todo el tiempo. Al ejecutar un comando en background observa cómo obtenemos una línea que nos informa del número de trabajo (1 y 2 en el ejemplo) y de los PID de los procesos (4340 y 4341 en el ejemplo). Después ejecutamos jobs para ver el estado de los trabajos, que es “running” (ejecutando) y “done” (hecho). Nuestra segunda tarea ha finalizado.

Si quisieras ejecutar en background un conjunto de comandos (a modo de script) lo que debes hacer es encerrarlos entre paréntesis e indicar el & al final. Ejemplo:

```
root@Master01:~# (while sleep 5;do date;done) &
[2] 4343
root@Master01:~# vie oct 18 00:32:43 CEST 2013
vie oct 18 00:32:48 CEST 2013
vie oct 18 00:32:53 CEST 2013
vie oct 18 00:32:58 CEST 2013
vie oct 18 00:33:04 CEST 2013
vie oct 18 00:33:09 CEST 2013
(...)
```

Este pequeño código hace un bucle infinito que consiste en dormir durante 5 segundos y después imprimir la fecha actual. Éste es ahora nuestro segundo trabajo [2] con PID 3065. Se está ejecutando en background e imprime cada 5 segundos la fecha y hora. Para pausarlo debemos hacer lo siguiente:

1º) Traer el proceso al primer plano: escribe fg %2

2º) Pausa este proceso con Ctrl-z.

RECUERDA: para pausar un proceso con Ctrl-z éste debe estar en primer plano.

```
root@Master01:~# fg
```

```
( while sleep 5; do date; done )
vie oct 18 00:46:16 CEST 2013
vie oct 18 00:46:21 CEST 2013
vie oct 18 00:46:26 CEST 2013
vie oct 18 00:46:31 CEST 2013
^Z
[2]+ Detenido      ( while sleep 5; do date; done )
```

Si ejecutas jobs verás que el primer trabajo está en ejecución y el segundo está detenido. Eso sí, el segundo trabajo ya no está en background (observa que en la descripción del comando ha desaparecido la & final). Esto es porque el comando fg trajo el trabajo 2 al primer plano. Para terminar esta prueba mataremos el trabajo 2 con kill %2. Hacemos jobs y quedará un único trabajo en background.

```
root@Master01:~# jobs
[1]- Ejecutando      yes > /dev/null &
[2]+ Detenido      ( while sleep 5; do date; done )
```

103.5.3.2. Comandos fg y bg:

Con los comandos **fg** (foreground) y **bg** (background) es posible manipular procesos, transfiriéndolos de un plano al otro.

Continuando con nuestro ejemplo anterior tenemos un único trabajo, el comando yes, en background.

Agregaremos otro trabajo al background:

Con iceweasel & se crea un segundo trabajo (el navegador iceweasel) que está en ejecución en el background. Se abrirá la ventana del navegador (minimízala) y el terminal estará operativo:

```
root@Master01:~# jobs
[1] Ejecutando      yes > /dev/null &
[2]+ Detenido      ( while sleep 5; do date; done )
[3]- Ejecutando      iceweasel &
```

El comando **fg** (foreground) permite transferir un proceso al primer plano. Recuerda pausarlo con Ctrl-Z. Observa el ejemplo:

```
root@Master01:~# fg 3
iceweasel
^Z
[3]+ Detenido      iceweasel
```

Ahora ejecutamos jobs para ver el estado de nuestros trabajos. Recuerda el símbolo [+] que indica que está al frente (primer plano) o es el proceso actual. Observa también que ha desaparecido el & en el proceso de iceweasel.

```
root@Master01:~# jobs
[1] Ejecutando      yes > /dev/null &
[2]- Detenido       ( while sleep 5; do date; done )
[3]+ Detenido       iceweasel
```

Ahora queremos enviar de vuelta al background nuestra ventana de iceweasel. Es la tarea 3. Emplearemos el comando **bg** y le indicamos el número de tarea.

```
root@Master01:~# bg 3
[3]+ iceweasel &
root@Master01:~# jobs
[1] Ejecutando      yes > /dev/null &
[2]+ Detenido       ( while sleep 5; do date; done )
[3]- Ejecutando     iceweasel &
```

Vuelve a aparecer & al final de la tarea 3. Está en el background. El comando **bg** permite enviar un proceso al background.

Si utilizamos **fg** sin ningún número de tarea traeremos al frente la tarea más reciente. También podríamos traer el frente un proceso sin conocer su número de tarea, utilizando el nombre del comando. Por ejemplo:

fg %iceweasel traerá al frente todos los procesos que sean navegadores iceweasel.

fg %ice Traerá al frente todos los procesos que empiecen por ice.

El **comando bg** tiene la misma sintaxis que el **comando fg**. Los comandos **fg**, **bg** y **jobs** son internos a la Shell o terminal donde se utilizan. Por este motivo utilizan los números de tarea y no los PID.

Para terminar nuestro ejemplo mataremos los trabajos 1, 2 y 3 con **kill %1**, **kill %2** y **kill%3**.

En resumen: utilizando **fg** traemos un proceso al frente y reactivamos su ejecución. Los procesos que están en el frente monopolizan la consola.

Utilizando **bg** sin parámetros enviamos el proceso que esté al frente en ese momento al segundo plano.

Ambos comandos, **bg** y **fg**, permiten indicarle un número de trabajo, que podemos obtener con **jobs**.

Si en un momento dado necesitas saber el PID de un trabajo de nuestra sesión utiliza **jobs -l**.

Al utilizar **jobs** el proceso que en ese momento es el trabajo actual ejecutado en el terminal vendrá marcado con un signo +.

Los tres comandos fg, bg y jobs operan con los procesos del terminal, que se llaman trabajos o tareas.

Para ejecutar un comando directamente en background utilizamos un & al final.

103.5.3.3. Entrada/salida de procesos en background.

Los procesos que están en modo background se están ejecutando en segundo plano, pero ¿qué ocurre si necesitan información de la entrada o bien generar información de salida?

A menos que se usen los mecanismos de redirección, la salida estándar (stdout) y la salida estándar de errores (stderr) serán redirigidos hacia el terminal controlador.

De forma análoga si el proceso está a la espera de información de entrada (stdin) será de este terminal controlador, pero nuestra consola no tiene forma de dirigir los caracteres que escribamos hacia la entrada estándar de un proceso si éste está en background.

En este caso la Shell Bash suspende el proceso, de forma que no se ejecutará más tiempo. Es decir: **si un proceso en background** necesita información de entrada automáticamente se pausará.

Lo que tendremos que hacer sería traer el proceso al primer plano con fg, aportarle la entrada necesaria y volverlo a llevar al background con bg.

El siguiente código ilustra un ejemplo sencillo. Se trata de un pequeño script que imprime la fecha, después redirige lo que escribamos hacia un fichero llamado entradas.dat, y por último vuelve a imprimir la fecha. El segundo paso, que utiliza el comando cat, requiere entrada por parte del usuario. Lanzamos el script en background:

```
root@Master01:~# (date; cat > entradas.dat;date) &
```

```
[4] 4767
```

```
root@Master01:~# vie oct 18 01:32:05 CEST 2013
```

```
[4]+ Detenido      ( date; cat > entradas.dat; date )
```

Debes tener en cuenta que en algunas shell es necesario pulsar Enter en el terminal para ver el estado de los procesos en Background. En Debian hemos tenido que hacerlo. En ese momento observamos cómo nuestro script está detenido. Se imprimió la primera fecha y se detuvo el proceso pues el comando cat requiere entrada por teclado.

Para finalizar el script lo traemos al primer plano (con fg) y escribiremos unas líneas de texto seguida de Ctrl-D para señalar el final de la entrada.

```
root@Master01:~# fg
```

```
( date; cat > entradas.dat; date )
```

```
este es el texto para la entrada, da igual lo que sea
```

```
se grabará en el fichero entradas.dat tras pulsar ctrl +d
```

```
vie oct 18 01:35:59 CEST 2013
```

```
root@Master01:~#
```

Ahora el proceso ha finalizado su ejecución imprimiendo la última fecha. Si quisieramos podríamos abrir el fichero creado entradas.dat.

En la práctica a los procesos que tengamos en background les colocaremos la entrada y la salida redirigida hacia ficheros para evitar este problema.

103.5.4. Ejecutar un proceso después de terminar la sesión:

Imagina ahora que tenemos varios procesos en background que están ejecutándose. En ese momento cerramos nuestro terminal. Lo normal es que se aborte la ejecución de estos procesos, aunque esto depende de la Shell que estemos utilizando.

El sistema operativo se comunica con los procesos mediante el envío de señales. En este caso al cerrar el terminal es muy probable que la Shell envíe a sus procesos hijo la señal SIGHUP (que significa “colgar”) y los cierre.

Si lo que deseamos es que un proceso concreto continúe su ejecución aún cuando se cierre el terminal utilizaremos el comando **nohup**, que provocará que este proceso ignore la señal SIGHUP.

Hagamos una pequeña prueba: En un terminal nuevo ejecuta `firefox &`. Se abre la ventana del navegador y se ejecuta firefox en segundo plano. Vuelve a la consola. Vamos a ver nuestros procesos (comando `ps`) con su relación jerárquica (opción `--forest`), en un listado donde aparezcan sólo las columnas PID, PPID, CMD (opción `-o pid,ppid,cmd`):

```
[profesor@localhost ~]$ firefox &
[1] 3668
[profesor@localhost ~]$ ps --forest -o pid,ppid,cmd
  PID  PPID  CMD
  1450   1444  bash
  3668   1450  \_ /usr/lib/firefox/firefox
  3687   3668  |  \_ [firefox] <defunct>
  3702   1450  \_ ps --forest -o pid,ppid,cmd
[profesor@localhost ~]$ █
```

Aquí observamos que nuestro terminal (que es el comando bash con PID 1450) es el proceso padre (PPID) para otros dos procesos: el firefox y el propio comando `ps` que hemos ejecutado. Si ahora cierras el terminal observarás que se cierra también la ventana de firefox.

Repetimos el proceso pero usando **nohup**. Abrimos un terminal nuevo pero en este caso ejecutamos: `nohup firefox &`

```
[profesor@localhost ~]$ nohup firefox &
[1] 3827
[profesor@localhost ~]$ nohup: se descarta la entrada y se añade la salida a «nohup.out»
█
```

También se ha lanzado el proceso firefox en segundo plano y el terminal ha mostrado un mensaje que dice “`nohup: se descarta la entrada y se añade la salida a nohup.out`”. En un momento hablaremos de este mensaje. Prueba ahora a cerrar el terminal. Observarás que firefox permanece en ejecución.

El comando **nohup** hace una tarea más. De forma automática redirige las salidas estándar `stdout` y `stderr` del proceso a un fichero llamado **nohup.out** o bien `$HOME/nohup.out`. En el caso de que se detecte que no se puede escribir en este fichero entonces `nohup` no se ejecutará.

Si queremos escribir en otro fichero que no sea `nohup.out` tendremos que redirigir `stdout` y/o `stderr` hacia el fichero deseado.

Es importante resaltar que el comando nohup no ejecutará tuberías (pipelines) o listas de comandos. Para lograrlo tendríamos previamente que guardar la tubería o la lista de comandos en un fichero y después ejecutarlo utilizando **sh** (la Shell por defecto) o el comando **Bash** (recuerda que para eso debes haber convertido el fichero en un fichero ejecutable).

En el siguiente código haremos esto mismo con un pequeño script que guardaremos en el fichero micomando.dat. Lo que hace el script es escribir por pantalla la hora cada 15 segundos.

Para guardar el script en el fichero haz lo siguiente:

```
root@Master01:~# echo "while sleep 15;do date;done" > micomando.dat
```

Ya hemos creado el fichero. Ahora vamos a ejecutarlo mediante el comando sh:

```
root@Master01:~# nohup sh micomando.dat &
```

```
[1]35678
```

```
nohup: se descarta la entrada y se añade la salida a «nohup.out»
```

Tras unos minutos mata el trabajo 1 con kill %1 y muestra el contenido del fichero nohup.out (utilizando cat nohup.out). Observarás que fue ahí donde se grabó las fechas puesto que nohup redirigió la salida hacia ese fichero:

```
root@Master01:~# cat nohup.out
vie oct 18 02:10:38 CEST 2013
vie oct 18 02:10:53 CEST 2013
vie oct 18 02:11:08 CEST 2013
vie oct 18 02:11:23 CEST 2013
root@Master01:~#
```

No olvides la utilización de kill %1 para matar el proceso. En caso contrario tendremos un comando escribiendo permanentemente en un fichero.

Nota: otra forma de ejecutar el fichero sin utilizar el comando sh sería utilizar el comando bash.

103.5.5. Envío de señales a los procesos:

Las señales en Linux son mecanismos para comunicarnos con los procesos. Generalmente estas señales las envía el propio sistema operativo y decimos que son **asíncronas** puesto que se envían cuando ha ocurrido un evento concreto.

El ejemplo más evidente acabamos de estudiarlo. Al cerrar un terminal éste proceso envía a todos sus procesos hijo la señal SIGHUP (“colgar”). También hemos empleado intuitivamente otra señal: cuando teníamos un proceso en primer plano y queríamos pausarlo escribíamos Ctrl-Z. Esto envía al proceso la señal SIGTSTP (“stop o detener”).

En este apartado veremos cómo enviar señales a los procesos.

103.5.5.1. El comando kill.

El comando kill es el que nos permite enviar una señal a cualquier proceso. Aunque el término kill signifique matar (y este sea el comportamiento por defecto de este comando, que ya hemos

empleado en ejemplos anteriores) utilizaremos el comando kill para enviar cualquier señal.

Si ejecutamos kill -l obtendremos una lista de las señales existentes:

```
root@Master01:~# kill -l
 1) SIGHUP   2) SIGINT   3) SIGQUIT   4) SIGILL   5) SIGTRAP
 6) SIGABRT  7) SIGBUS   8) SIGFPE   9) SIGKILL  10) SIGUSR1
11) SIGSEGV 12) SIGUSR2 13) SIGPIPE  14) SIGALRM   15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT  19) SIGSTOP  20) SIGTSTP
21) SIGTTIN 22) SIGTTOU23) SIGURG  24) SIGXCPU25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO   30) SIGPWR
31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42)
SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47)
SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52)
SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57)
SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62)
SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
root@Master01:~#
```

Observa que cada señal tiene un número o identificador asociado. Podríamos usar la opción –s seguida del nombre de la señal, o bien la opción –n seguida del número, o incluso –nombre_señal. Las siguientes tres líneas son equivalentes (en todas estamos enviando la señal SIGHUP al trabajo con id de tarea 6:

kill –s SIGHUP %6 con opción –s se escribe el nombre de la señal.

kill –n 1 %6 con opción –n escribimos el número de la señal.

kill –SIGHUP %6 también podemos indicar directamente –Nombre_Señal.

El último parámetro de kill ha sido %id_trabajo. Pero si quitamos el símbolo % nos estaremos refiriendo a un PID. Por ejemplo:

kill –SIGTSTP 5847 mandará la señal de Stop al proceso con PID 5847.

Hagamos una prueba de envío de señales en un nuevo terminal:

```
[profesor@localhost ~]$ firefox &
[1] 4287
[profesor@localhost ~]$ jobs
[1]+ Ejecutando firefox &
[profesor@localhost ~]$ kill -SIGTSTP 4287
[profesor@localhost ~]$ jobs
[1]+ Detenido firefox
[profesor@localhost ~]$ kill -SIGCONT 4287
[profesor@localhost ~]$ jobs
[1]+ Ejecutando firefox &
[profesor@localhost ~]$ █
```

Analicemos este código. Primero ejecutamos firefox en background. El id de trabajo es 1 y su PID es 4287. Comprobamos que está en ejecución con jobs.

Después ejecutamos kill –SIGTSTP 4287. Hemos enviado la señal de Stop al proceso. Aquí nos damos cuenta de que kill no muestra ningún mensaje por pantalla. Para comprobarlo utilizamos jobs y vemos que el proceso está detenido y ha desaparecido la & final. La señal SIGTSTP equivale a traer el proceso al frente (como el comando fg) y detenerlo (con Ctrl-Z).

Por último enviamos al mismo proceso la señal SIGCONT (“continuar”), que ha colocado de nuevo el proceso en background en ejecución (luego es equivalente a utilizar el comando bg).

Un apunte: en el desarrollo de este tema estamos ejecutando el comando kill de un terminal correspondiente a una Shell Bash. Las Shell bash y csh incluyen su propia versión del comando kill. No obstante podríamos utilizar el comando externo empleando /bin/kill. En ese caso trabajaremos indicando los PID al tratarse de un comando externo. Estos apuntes han sido probados en una distribución Debian y en Fedora). En las mismas el comando externo /bin/kill no soporta identificadores de trabajo sino PIDs.

Antes de terminar este apartado analicemos algunas señales típicas:

- La señal SIGINT (número 2) equivale a interrumpir un proceso con Control-C.
- La señal SIGTERM (número 15) es la señal empleada para matar definitivamente un proceso, aunque le permite cerrar antes sus ficheros abiertos. Esta es la señal enviada por defecto cuando usamos kill sin especificar señales. Ejemplo: kill 8950 envía SIGTERM al proceso 8950. En muchos casos esta señal es equivalente a SIGINT pues ambas finalizan el proceso.
- La señal SIGKILL (número 9) también mata el proceso inmediatamente, sin que éste termine sus tareas como cierre de ficheros abiertos.

Ten en cuenta que el comando kill te permitirá enviar señales a tus procesos, pero raramente podrás enviar señales a procesos de otros usuarios, a menos, claro, que seas el usuario root.

Por último también se soporta omitir la palabra SIG cuando usamos la opción –nombre_señal. Las dos líneas siguientes son equivalentes:

kill –SIGTERM 8839

kill –TERM 8839

103.5.2. El comando killall.

El comando **killall** es una variante de kill empleada para matar un grupo de procesos por su nombre

y no por su PID. Por ejemplo si quisiéramos eliminar todos los procesos de firefox de nuestra sesión haríamos:

```
killall firefox
```

Y se enviaría a todos la señal SIGTERM (15), ya que este comando funciona igual que kill.

También podríamos enviar otra señal usando la notación abreviada:

```
killall -HUP firefox
```

 enviará la señal SIGHUP a todos los procesos de firefox.

Opciones interesantes son:

-l lista todas las opciones disponibles en el comando killall.

-i antes de enviar la señal a cada proceso nos pedirá una confirmación de tipo (y/n), de forma que para cada proceso decidimos si enviar o no la señal. Esta opción es obligatoria si estamos utilizando el usuario root ya que podríamos destruir procesos de algunos usuarios de nuestro sistema.

103.5 EXTRAS

103.5.1 EXTRAS pkill

Este comando finaliza procesos enviando la señal 15 de forma predeterminada. Permite finalizar un proceso utilizando el nombre. La sintaxis de este comando es:

```
pkill [- opciones] nombredeproceso
```

Las opciones disponibles con pkill son parecidas a las que utilizamos con pgrep. No obstante la opción -u permite especificar el usuario propietario del proceso. Esto es muy útil para que no matemos procesos con el mismo nombre pero pertenecientes a otros usuarios.

-u usuario

-o el primero ejecutado (oldest)

-n El más reciente (newest)

-l lista

```
root@cli:~# ps -f
UID      PID  PPID  C STIME TTY          TIME CMD
root      5273  5261  0 dic17 pts/2      00:00:00 -bash
root     24500  5273  0 08:41 pts/2      00:00:00 sleep 1000
root     24501  5273  0 08:41 pts/2      00:00:00 sleep 1100
root     24502  5273  0 08:41 pts/2      00:00:00 sleep 1200
root     24503  5273  0 08:41 pts/2      00:00:00 sleep 1300
root     24504  5273  0 08:41 pts/2      00:00:00 sleep 1400
root     24506  5273  0 08:42 pts/2      00:00:00 sleep 1500
root     24507  5273  0 08:42 pts/2      00:00:00 sleep 1600
root     24508  5273  0 08:42 pts/2      00:00:00 sleep 1700
root    24546  5273  0 08:49 pts/2      00:00:00 ps -f
root@cli:~# pkill sleep
```

```
[1] Terminado           sleep 1000
[2] Terminado           sleep 1100
[3] Terminado           sleep 1200
[4] Terminado           sleep 1300
[5] Terminado           sleep 1400
[7] Terminado           sleep 1500
[8]- Terminado          sleep 1600
[9]+ Terminado          sleep 1700
```

104 DISPOSITIVOS, SISTEMAS DE ARCHIVOS EN GNU/LINUX Y ESTÁNDAR FHS

- 104.1. Crear particiones y sistemas de archivos
- 104.2. Mantenimiento de la integridad de sistemas de archivos.
- 104.3. Control de montaje y desmontaje de los sistemas de archivos.
- 104.4. Administrar cuotas de disco.
- 104.5. Controlar permisos y propiedades de archivos.
- 104.6. Crear y modificar enlaces simbólicos y duros.
- 104.7. Encontrar archivos de sistema y conocer su localización correcta.

104.1. Crear particiones y sistemas de archivos

Peso en el examen de certificación: 2 puntos.

Objetivo: Configurar particiones de disco y después crear sistemas de ficheros. Ambas tareas sobre soportes como discos duros. Se incluyen las particiones de intercambio (swap)

Conceptos y áreas de conocimiento:

- usar varios comandos mkfs para particionar y crear sistemas de ficheros:
- ext2/ext3/ext4
- xfs
- reiserfs v3
- vfat

Términos y utilidades

- fdisk
- mkfs
- mkswap

104.1.0. Introducción.

Para manejar particiones se pueden utilizar herramientas como **fdisk**, **cfdisk**, **sfdisk**, **parted** y sus interfaces gráficas **qtparted** o **gparted**.

- **fdisk**: es la más antigua y más utilizada.
- **cfdisk**: es un poco más visual y se utiliza con flechas direccionales.
- **sfdisk**: es más complicada y precisa.
- **parted**: permite operaciones avanzadas como el redimensionamiento. Se puede utilizar con las interfaces gráficas **qtparted** o **gparted**.

Una vez se ha creado una partición, el sistema de ficheros debe ser añadido para que Linux pueda hacer uso de ese espacio. La utilidad **mkfs** se usa para crear sistemas de ficheros en particiones vacías.

104.1.1. Particiones

La herramienta principal usada para crear particiones de disco es **fdisk**. La utilidad **fdisk** divide el disco en particiones y escribe la tabla de particiones en el sector 0 (conocido como superblock).

Sintaxis básica:

fdisk [-l] [disco]

Nota: para utilizar **fdisk** es necesario ejecutarlo como *root*.

fdisk puede utilizarse de dos maneras, de modo interactivo o no-interactivo. El modo interactivo ofrece menús y opciones y permite modificar, además de explorar, particiones y tablas de particiones.

Se puede evitar el menú ejecutando **fdisk** con las siguientes opciones:

- **-l**: lista las tablas de particiones para todos los discos del sistema, si no se indica ninguno, o

las particiones del disco que se indique. La información obtenida es la misma que en modo interactivo con la entrada *p* del menú.

- **-v:** da únicamente la versión de **fdisk**.

Si no se utiliza ninguna de esas opciones, **fdisk** comprueba si el número de cilindros del dispositivo por defecto (*hda1*) es mayor de 1024 y avisa de ello, si es así. Entonces espera una instrucción, que se ejecutará sobre el primer disco del sistema.

Se puede iniciar **fdisk** con un dispositivo distinto al de por defecto especificándolo en la línea de comandos. Por ejemplo, para arrancar **fdisk** con el tercer driver IDE, se debe poner:

```
# fdisk /dev/hdc
```

Una vez la utilidad está iniciada, pulsando *m* se presenta una ayuda en forma de menú, algunas de las cuales se listan en la tabla 1.

Tabla 1
Algunas de las opciones de *fdisk*

Opción	Función
p	Muestra información sobre la partición o imprime la tabla de particiones.
d	Borra una partición.
n	Crea una partición.
q	Sale de la aplicación sin guardar los cambios.
w	Guarda los cambios y sale de la aplicación.
m	Muestra los comandos disponibles.
v	Verifica la tabla de particiones.
a	Cambia el indicador de estado de arranque de la partición.
l	Lista los tipos de particiones conocidos
t	Cambia el identificador de sistema de una partición
x	Funciones adicionales (sólo para usuarios avanzados)

Listar particiones

Para ver una tabla de un único disco, lo añadimos al comando como parámetro, por ejemplo:

```
# fdisk -l /dev/sda
```

El mismo resultado se obtiene ejecutando *fdisk* con un disco como argumento, en este caso **/dev/sda**, y pulsando la tecla *p* (print):

```
root@ubunserv:/# fdisk /dev/sda
...
Orden (m para obtener ayuda): _
```

```
Orden (m para obtener ayuda): p

Disco /dev/sda: 4294 MB, 4294967296 bytes
255 cabezas, 63 sectores/pista, 522 cilindros
Unidades = cilindros de 16065 * 512 = 8225280 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador de disco: 0x0002cf99

Dispositivo Inicio     Comienzo      Fin      Bloques  Id Sistema
/dev/sda1    *           1          32      248832   83  Linux
La partición 1 no termina en un límite de cilindro.
/dev/sda2        32         523      3942401   5  Extendida
La partición 2 no termina en un límite de cilindro.
/dev/sda5        32         523      3942400   8e  Linux LVM
```

Las siete columnas que la impresión de la tabla de particiones nos muestra la siguiente información:

- En la primera columna se encuentra el dispositivo resultado de la partición de disco *sda*. Así, podemos observar que en este momento contamos con 3 particiones: *sda1*, *sda2* y *sda3*.
- La segunda columna nos informa que es la primera partición del disco la que contiene la parte *booteable*.
- La tercera y cuarta columna nos dan información acerca de los cilindros donde empieza y termina cada partición respectivamente.
- La quinta nos dice cuantos bloques ocupa la partición.
- La sexta y séptima columna informan del tipo de partición (identificador y sistema respectivamente).

Eliminar

Para eliminar una partición se pulsa la tecla *d* (delete) y el número de partición (*sdbX*, siendo X el número). Si sólo hay una partición la toma por defecto.

```
Orden (m para obtener ayuda): d
Número de partición (1-5): 1_
```

Crear particiones

Para crear una partición se pulsa la tecla *n* (new).

```
Orden (m para obtener ayuda): n
Acción de la orden
  l  Partición lógica (5 o superior)
  p  Partición primaria (1-4)
```

Se solicitarán los siguientes datos:

1. Tipo de partición: hay que pulsar *p* o *l*, dependiendo del tipo de partición que se quiera crear, primaria o extendida.
2. Selección de la partición: hay que indicar el número de partición, que no tiene porqué ser consecutivo, ni 1, en el caso de que no exista ninguna partición previamente.
3. Primer cilindro: se selecciona la posición de inicio de la partición, por defecto **fdisk** propone

el primer cilindro libre desde principio del disco.

4. Último cilindro o tamaño de la partición: por último hay que establecer dónde termina la partición, para ello se indica cual es el último cilindro o el tamaño que tendrá la partición.

```
Orden (m para obtener ayuda): n
Acción de la orden
  e  Partición extendida
  p  Partición primaria (1-4)
p
Número de partición (1-4): 1
Primer cilindro (1-2610, valor predeterminado 1):
Se está utilizando el valor predeterminado 1
Último cilindro, +cilindros o +tamaño{K,M,G} (1-2610, valor predeterminado 2610)
:
```

Antes de guardar los cambios, pulsando **w**, se puede comprobar el nuevo estado de la tabla de particiones pulsando **p**.

Guardar particiones

Para que las acciones realizadas con *fdisk* tengan efecto hay que pulsar **w** (write) y la tabla de particiones se escribirá en el MBR.

```
Orden (m para obtener ayuda): w
!Se ha modificado la tabla de particiones!
Llamando a ioctl() para volver a leer la tabla de particiones.
[ 984.429293] sd 2:0:1:0: [sdb] Assuming drive cache: write through
Se están sincronizando los discos.
```

Si el disco en cuestión está en uso Linux no puede volcar la nueva tabla ni crear las nuevas particiones, aparecerá un *mensaje de advertencia*. Para corregir éste problema hay que forzar al núcleo a leer de nuevo la tabla de particiones.

Forzar la sincronización

Para forzar al núcleo a leer la tabla de particiones se pueden utilizar los comandos **blockdev** y **partprobe**.

blockdev con el parámetro **--rereadpt** (re-read partition table)

```
# blockdev --rereadpt /dev/sdb
```

partprobe sólo está disponible si **parted** está instalado. Lee las tablas de todas las particiones si no se le indica como argumento un disco determinado.

Este comando tendrá el mismo efecto que el anterior:

```
# partprobe /dev/sdb
```

Modificar el tipo

El tipo de una partición se puede modificar en cualquier momento ya que no implica la modificación de su tamaño. Hay que pulsar la tecla **t**, indicar el número de partición sobre la que se quiere trabajar y el código correspondiente al tipo que se le quiera asignar. Pulsando **L** se muestra la lista de códigos y tipos de sistemas.

```
Orden (m para obtener ayuda): t  
Número de partición (1-5): 5  
Código hexadecimal (escriba L para ver los códigos):
```

Marcar un partición como activa

Para que una partición sea **booteable** hay que pulsar *a* en el menú de **fdisk** e indicar el número de partición que se quiere marcar como activa.

Cuando se lista la tabla de particiones aparecerá con un *asterisco* junto al nombre.

Salir

Para salir sin guardar los cambios se pulsa *q*.

104.1.2. Sistemas de ficheros

El comando **mkfs** crea un nuevo sistema de ficheros en un dispositivo de bloque especificado como una partición en un disco duro. Las opciones usadas por **mkfs** están seguidas por un argumento especificando la partición que debe ser formateada. Después de la ejecución del comando, el código de salida de 0 (cero) indicará que se ha llevado a cabo con éxito mientras que el código de salida 1 (uno) indicará fallo.

Ejemplo de la sintaxis usada:

```
# mkfs -opciones argumentos
```

El uso básico es:

```
# mkfs -t tiposistemaficheros opciones periférico
```

Donde **tiposistemaficheros** es un tipo de sistema de ficheros soportado por Linux (por ejemplo, *ext2* o *xfs*) y **periférico** es el lugar de la partición del disco objetivo (por ejemplo, */dev/hda1* o */dev/sdc3*). Las opciones específicas del sistema de ficheros se añaden después de **tiposistemaficheros**.

La utilidad **mkfs** se utiliza con las opciones que aparecen en la tabla 2.

Tabla 2
Opciones usadas con *mkfs*

Opción	Función
-t ftype	Especifica el tipo de sistema de ficheros a crear. Por defecto se usa ext2.
fs - options	Opciones específicas de sistema de ficheros para ser pasados al sistema real de ficheros que vamos a crear.
-c	Comprueba el dispositivo en busca de bloques defectuosos antes de crear el sistema de ficheros.
-l fichero	Lee los bloques defectuosos del fichero.
-v	Produce una salida con más información, incluyendo todas las órdenes específicas del sistema de ficheros concreto que se ejecutan. Es útil para comprobaciones.

El comando básico para crear un sistema de ficheros de tipo *ext2* en la partición */dev/sdb1* es el siguiente:

```
# mkfs -t ext2 /dev/sdb1
```

La ejecución del comando proporciona la siguiente información:

```
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=4096 (bitácora=2)
Tamaño del fragmento=4096 (bitácora=2)
Stride=0 blocks, Stripe width=0 blocks
1310720 nodos-i, 5241198 bloques
262059 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=0
Número máximo de bloques del sistema de ficheros=0
160 bloque de grupos
32768 bloques por grupo, 32768 fragmentos por grupo
3192 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Escribiendo las tablas de nodos-i: hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 35 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

Es posible dar una etiqueta al sistema de ficheros.

- Cada bloque es de 4096 bytes.
- Hay 5241198 bloques.
- Los inodos representan el número máximo de ficheros: 1310720
- Se reserva el 5% del espacio de disco para el superusuario.
- Se reparten las tablas de inodos por grupos.
- Hay 10 superbloques de emergencia.
- Se pueden modificar algunos parámetros del sistema de ficheros con el comando **tune2fs**.

mkfs encarga la creación del sistema de ficheros a una de las múltiples utilidades especializadas, dependiendo del tipo de sistema de ficheros que especificamos: **mkfs.ext2**, **mkfs.xfs** o **mkfs.vfat**, por ejemplo.

En la tabla 3 se pueden ver algunas de las utilidades prácticas de **mkfs**. La mayoría de dichas utilidades implementan las mismas opciones, aunque varían según las funcionalidades implementadas en los distintos sistemas de ficheros. A pesar de que **mkfs** invoca a esas otras utilidades, generalmente se deberá ejecutar el comando estándar **mkfs** cuando se crea un sistema de ficheros en vez de ejecutar cualquiera de las utilidades especializadas directamente.

A pesar de las diferencias, existen unas cuantas opciones que son comunes a todas las utilidades **mkfs.***. Añadiendo el parámetro **-c** verificará el dispositivo especificado en busca de bloques malos, que se omitirán durante el paso de creación del sistema de ficheros actual. Añadir los parámetros **-v** o **-V** produce salidas detalladas o extremadamente detalladas, respectivamente.

Tabla 3
Utilidades para la creación de Sistemas de Ficheros

Comando	Uso
mkfs.ext2 o mke2fs	Crea un sistema de ficheros ext2

mkfs.ext3	Crea un sistema de ficheros ext3
mkfs.reiserfs	Crea un sistema de ficheros reiserfs
mkfs.vfat	Crea un sistema de ficheros VFAT
mkfs.ntfs	Crea un sistema de ficheros NTFS
mkfs.msdos o mkdosfs	Crea un sistema de ficheros MS-DOS
mkswap	Crea un sistema de ficheros de Linux swap
mkraid	Inicializa y actualiza cadenas de dispositivos RAID
mkfs.minix	Crea un sistema de ficheros Minix
mkfs.bfs	Crea un sistema de ficheros SCO BFS

Para obtener más detalles de las opciones al crear cada tipo de sistemas de ficheros, se puede hacer uso de las páginas *man* de Linux.

Para crear un sistema de ficheros debe ser usada la herramienta correcta especificada en la tabla 3. Por ejemplo, un uso apropiado de estas utilidades es crear una partición *ext2* usando **mkfs.ext2**:

```
# mke2fs /dev/hda3
```

ext2 y ext3

Los sistemas de ficheros *ext2* y *ext3* son compatibles. *ext3* es una extensión de *ext2*, un diario en el que se registran todos los cambios del sistema de ficheros antes de ser efectuados realmente, a este mecanismo se le conoce como *journaling*, en caso de parada abrupta, el sistema repasa las grabaciones del diario y comprueba si se realizaron las operaciones y si es necesario las vuelve a ejecutar. El journal es el archivo *.journal* que se encuentra en el directorio raíz del sistema de archivos.

ext2 es el primer sistema de ficheros desarrollado específicamente para Linux, es rápido y necesita menos escritura que *ext3*, por lo tanto, lleva menos desgaste de los soportes de almacenamiento.

Se puede transformar fácilmente un sistema *ext2* en *ext3* y viceversa.

ext3 es sucesor de *ext2*, se puede considerar como **ext2** con algunas mejoras, y por lo tanto, soportando los mismos comandos y los mismos parámetros, los más usuales son:

Tabla 4
Parámetros ext2 y ext3

Parámetro Significado

-b	Tamaño de los bloques en bytes, múltiplo de 512. Cualquier fichero creado en el disco ocupa al menos un bloque y, por lo tanto, si se prevé un gran número de pequeños ficheros hay que poner un valor bajo.
-c	Verifica los bloques defectuosos antes de crear el sistema de ficheros. También se puede utilizar el comando badblocks .
-i	Relación bytes/inodo. Se calcula el tamaño de la tabla de inodos en función del tamaño total del sistema de ficheros. Un inodo ocupa 128 bytes.
-m	Seguido de un número n, donde n es el porcentaje de bloques reservado al superusuario, por defecto el 5%. Si n es 0 se gana espacio, no es aconsejable para sistemas críticos.
-L	Establece una etiqueta para el sistema de ficheros, útil para el montaje
-j	Añade la zona de <i>journaling</i> (diario) a un sistema de ficheros <i>ext2</i> , convirtiéndolo en <i>ext3</i>

De ext2 a ext3

ext3 es un sistema de ficheros *ext2* con soporte transaccional, para convertir un sistema de ficheros *ext2* en *ext3* se utiliza *tune2fs*.

```
# tune2fs -j /dev/sdb1
```

Si el sistema de archivos está montado mientras se realiza la migración, el journal estará visible como *.journal* en el directorio raíz del sistema de archivos. Si el sistema de archivos no está montado, el journal se ocultará y no aparecerá en el sistema de archivos.

De ext3 a ext2

Para pasar un sistema de ficheros *ext3* a *ext2* hay que suprimir el soporte transaccional con *tune2fs*, previamente el sistema de archivos debe ser desmontado.

```
# tune2fs -O ^has_journal /dev/sdb1
```

(el parámetro es la letra o mayúscula)

Luego se debe eliminar el fichero *.journal* de la raíz de la partición.

Label

Con este comando se muestra la etiqueta del un sistema de ficheros que se especifique como argumento

```
# e2label /dev/sdb1
```

Y si a continuación se le indica una etiqueta se cambia la que tuviese por la nueva.

```
# e2label /dev/sdb1 NUEVAETIQUETA
```

Reiserfs

Estos sistemas de ficheros son muy rápidos trabajando con archivos pequeños, pero con ficheros grandes son lentos. La creación de un sistema de ficheros en *reiserfs* es tan simple como con *ext2* y *ext3*. No es posible convertir un sistema de ficheros *ext2/ext3* en *reiserfs* ni viceversa.

```
# mkfs -t reiserfs /dev/sdb1
```

reiserfs acepta parámetros diferentes de *ext2* y *ext3*.

Tabla 5
Parámetros reiserfs

Parámetro Significado

-b	Tamaño de los bloques en bytes, múltiplo de 512, incluido entre 512 y 8192. Si no se especifica, se determina según el tamaño de la partición.
-l	Label, etiqueta que se le da al sistema de ficheros.
-f	Fuerza la ejecución del comando, incluyendo un disco y no una partición.
-d	Modo debug, proporciona más información.

Label

Para modificar la etiqueta de un sistema *reiserfs* se utiliza el comando *reiserfstune*.

```
# reiserfstune -l NUEVAETIQUETA /dev/hda6
```

Vfat

Se crea el sistema de ficheros de la misma forma que los anteriores, Si no se especifica el tipo de

FAT (12, 16 o 32), se seleccionar automáticamente en función del tamaño de la partición.

```
# mkfs -t vfat -v /dev/sdb1
```

```
mkfs.vfat 3.0.9 (31 Jan 2010)
Auto-selecting FAT32 for large filesystem
/dev/sdb1 has 255 heads and 63 sectors per track,
logical sector size is 512,
using 0xf8 media descriptor, with 41929586 sectors;
file system has 2 32-bit FATs and 32 sectors per cluster.
FAT size is 10240 sectors, and provides 1309658 clusters.
There are 32 reserved sectors.
Volume ID is 2846e0ff, no volume label.
```

En este ejemplo se ve como se ha seleccionado automáticamente el sistema FAT32.

Se pueden especificar varios parámetros:

Tabla 6 Parámetros vfat	
Parámetro	Significado
-c	Verifica el periférico antes de la creación.
-F	Tamaño de la FAT (12, 16, 32)
-l	Permite utilizar un disco completo y no una partición.
-n	Etiqueta, nombre del volumen.
-v	Visualización de los detalles de la creación.

mkswap

Linux utiliza el código 0x82 para identificar el área de intercambio, *swap*.

mkswap es el comando usado para inicializar particiones de intercambio, por ejemplo:

```
# mkswap /dev/hda3
```

Para activar la partición, se usa el comando **swapon**, en el caso anterior:

```
# swapon /dev/hda3
```

104.1.EXTRAS

104.1.EXTRAS LVM (Logical Volume Management)

Fuente : <http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento/-/wiki/Base%20de%20Conocimiento/Volumenes+L%C3%B3gicos+%28LVM%29>

Breve retrospectiva de los LVM

LVM es una implementación de un administrador de volúmenes lógicos para el kernel Linux. Se escribió originalmente en 1998 por Heinz Mauelshagen, que se basó en el administrador de volúmenes de Veritas usado en sistemas HP-UX. LVM incluye muchas de las características que se esperan de un administrador de volúmenes, incluyendo:

- Re-dimensionado de grupos lógicos
- Re-dimensionado de volúmenes lógicos
- Instantáneas de sólo lectura (LVM2 ofrece lectura y escritura)
- RAID 0 de volúmenes lógicos

Entendiendo que es un LVM

Te ha pasado alguna vez que instalas algún sistema operativo Linux en tu equipo (Fedora, Debian, Red Hat, Centos , Ubuntu, Mandriva, Slackware) y te vas con la idea de que has elegido el tamaño adecuado para cada partición, pero con el paso del tiempo te percatas que una de tus particiones se queda sin espacio libre, que haces entonces? ¿Como resuelves este problema? ¿re-instalas nuevamente tu Linux en tu equipo o servidor? Aquí es donde los LVM entran en acción, ya que la implementación de estos nos facilitaran la administración de nuestros sistemas de almacenamiento. Un LVM (Logical Volume Manager) es una implementación de un administrador de volúmenes lógicos para el kernel de Linux que nos permite:

- Re-dimensionado de grupos lógicos
- Re-dimensionado de volúmenes lógicos
- Respaldos de lectura y escritura (solo para LVM2)
- RAID0 de volúmenes lógicos.

Supongamos que en una partición de nuestro disco duro en donde hemos colocado la partición de /home se está quedando sin espacio, para resolver este problema solo habría que aumentar el espacio en dicha partición, pero el problema radica en como hacerlo. LVM nos permite añadir más espacio a esa partición de forma que no sea necesario tener que reinstalar el sistema operativo en un disco duro de mayor capacidad. Si no que asociamos un nuevo disco a la partición /home. Por lo que cuando un nuevo disco se añade al sistema, no es necesario mover los datos de los usuarios, simplemente se añade el nuevo disco al grupo lógico correspondiente y se expanden los volúmenes lógicos a todo lo que se considere adecuado, o bien se migran los datos de discos antiguos a otros nuevos de forma totalmente transparente al usuario.

Niveles de un LVM

Un LVM se divide en 5 niveles, estos son:

1.Volúmenes Físicos o Physical Volumes 2.Volúmenes Lógicos o Logical Volumes 3.Grupos de volúmenes o Volume Groups 4.Extensión Lógica o Logical Extents 5.Extensión Física o Physical Extents

Volúmenes Físicos (Physical Volumes)

Son los discos duros físicos o particiones de un disco duro

Volúmenes Lógicos (Logical Volumes)

Es el equivalente a una partición de un disco duro, por lo que puede contener un sistema de archivos como por ejemplo /home.

Grupos de volúmenes (Volume Groups)

Es la parte que engloba los volúmenes lógicos (Logical Volumes) y nuestros volúmenes físicos (Physical Volumes), es decir, es una especie de contenedor donde se sitúan los volúmenes lógicos (Logical Volumes) y los volúmenes físicos (Physical Volumes).

Extensión Lógica (Logical Extents)

Cada volumen físico esta divido en pequeños trozo de datos llamados extensión lógica(logical extents).

Extensión Física (Physical Extents)

Cada volumen físico(LV) esta divido en pequeños trozos datos que se llaman extensión física(physical extents), que es del mismo tamaño que una extensión lógica(LE).

Instalación y Configuración de un LVM

Instalando el paquete lvm

A partir de este punto empezaremos a descargar los paquetes necesarios para configurar un equipo o servidor con LVM. La forma en que se instalara este paquete sera tecleando en un BASH lo siguiente:

```
[BASH] # yum install -y lvm2
```

Esto para distribuciones Linux derivadas de la familia Red Hat -> Fedora, CentOS, WhiteBox, Mepis Para instalarlo en distribuciones Linux de la familia Debian -> Ubuntu,Linux Mint y derivados sera de la siguiente forma:

```
[BASH] # aptitude install -y lvm2
```

Preparando el escenario LVM con fdisk

Ahora que ya tenemos instalado el paquete de LVM es hora de configurar nuestro disco duro, para ello haremos uso del comando fdisk el cual nos servirá para primero asignar un espacio en el disco duro para el LVM en el cual posteriormente crearemos los tres niveles básicos de un LVM, nos referimos a:

1.El Volumen Físico 2.El Grupo de Volúmenes 3.El Volumen Lógico

Supongamos que tenemos la siguiente distribución de espacio en el siguiente disco duro hipotético. Donde hda es un disco duro de 20 GB el cual alberga la siguiente distribución

Raíz 10 Giga-bytes asignados

Swap 256 Megaabytes asignados

Boot 256 Megaabytes asignados

Espacio Libre 9.5 Giga-bytes de espacio libre

NOTA:La partición /boot no puede ser asignada a un LVM, esto debido a que el sistema de boteo de Linux no reconoce las particiones de LVM y por lo tanto no puede ser iniciado el proceso de arranque, se recomienda que esta partición quede separada.

El lugar donde implementaremos el LVM sera en el espacio libre del disco duro, para ello haremos uso del comando fdisk, el cual servirá para preparar la instalación del LVM. Fdisk es una aplicación disponible para Linux el cual permite dividir en forma lógica un disco duro, siendo denominado este nuevo espacio como partición. La sintaxis para lanzar esta aplicación es la siguiente:

```
[BASH]# fdisk /dev/[h|s] d [a|b|c]
```

Donde :

- h La letra 'h' hace referencia a un disco duro PATA. Eje: hd
- s La letra 's' hace referencia a un disco duro SATA. Eje: sd
- a La letra 'a' hace referencia al primer disco duro del equipo
- b La letra 'b' hace referencia al segundo disco duro del equipo
- c La letra 'c' hace referencia al tercer disco duro del equipo

En nuestro caso, contamos con un solo disco duro PATA en el equipo, por lo que ejecutaremos fdisk de la siguiente manera:

```
[BASH]# fdisk /dev/hda
```

Una vez que la aplicación esta iniciada, se nos presenta el siguiente mensaje:

Command (m for help):

Si usted presiona la tecla 'm' se imprimirá el menú con las herramientas propias del comando fdisk. Estas herramientas son:

- a Conmuta el indicador de inicable
- b Modifica la etiqueta de disco bsd
- c Conmuta el indicador de compatibilidad con DOS
- d Suprime una partición
- l Lista los tipos de particiones conocidos
- m Imprime este menú
- n Añade una nueva partición
- o Crea una nueva tabla de particiones DOS vacía
- p Imprime la tabla de particiones
- q Sale sin guardar los cambios
- s Crea una nueva etiqueta de disco Sun
- t Cambia el identificador de sistema de una partición
- u Cambia las unidades de visualización/entrada
- v Verifica la tabla de particiones

- w Escribe la tabla en el disco y sale
- x Funciones adicionales (sólo para usuarios avanzados)

Como podemos notar, con la opción "m" podemos imprimir nuevamente este menú. Seleccione del menú, la opción "**“Imprimir tabla de particiones”**", para ello teclee la letra 'p' , esto nos mostrara la distribución actual de nuestras particiones en nuestro disco duro.

Observe que

- **En /dev/hda1 esta la partición de /boot**
- **En /dev/hda2 esta la partición de /**
- **Y por ultimo en /dev/hda3 la memoria de intercambio**

Lo primero que haremos sera crear una partición extendida, posteriormente crearemos sobre la partición extendida la partición lógica, dicha partición contendrá nuestro LVM.

ATENCION: En un disco IDE o SATA podemos crear únicamente hasta 3 particiones primarias en el disco duro. Si queremos de más particiones podemos crear una extendida y ahí seguir particionando (hasta un total de 60 particiones lógicas). Si queremos más, no queda otra que utilizar un segundo disco duro. En el caso de un disco SCSI, éste admite hasta 15 particiones primarias.

Para agregar las particiones al disco duro teclee la letra 'n' Recordemos:

n -> Añade una nueva partición al disco duro

El menú emergente nos preguntara si queremos añadir una partición extendida o primaria, obviamente daremos de alta primero una partición extendida, para ello teclee la letra 'e' lo cual le indica al sistema que hemos elegido generar una partición extendida. Si te das cuenta, el sistema nos pregunta el espacio libre del disco duro que queremos ocupar para generar la partición extendida. Por default, la aplicación “fdisk” nos sugiere utilizar los valores predeterminados, estos valores corresponden al espacio libre del disco duro, estos valores son 1372 y 2610 (cilindros) los cuales corresponden al inicio y final del espacio libre disponible del disco. También podemos asignar manualmente estos valores, ya sea en cilindros o en unidades de Mega-bytes o Giga-bytes. El siguiente paso sera generar la partición lógica, para ello teclearemos nuevamente la letra 'n', los cual nos propondrá utilizar nuevamente los valores predeterminados del espacio disponible del disco duro, los cuales aceptamos nuevamente. Si eres observador habrás notado que '**fdisk**' ya no pregunta si deseamos añadir una partición primaria o extendida, esto es debido a que ya tenemos generada una partición extendida y por default las nuevas particiones que sean generadas serán tratadas como particiones lógicas. Al terminar, revisa nuevamente la tabla de particiones del disco duro y verifica los cambios

Como se muestra en la imagen, fueron generadas dos nuevas particiones:

/dev/hda4 Partición Extendida

/dev/hda5 Partición LVM

Solo nos resta cambiar el identificador a la partición hda5 ya que este tiene asignado como identificador a **“Linux”** el cual es incorrecto ya que esta partición sera de tipo **“LVM”**.

Para hacerlo teclee la letra 't' , opción que nos servirá para cambiar el identificador a la partición /hda5

Command (m for help):t

Fdisk, nos preguntara a que partición queremos cambiar el ID, como ya sabemos sera a la partición

hda5 o sea la partición 5., Como no conocemos el código hexadecimal para las particiones LVM lanzamos la ayuda para poder visualizar todos los códigos hexadecimales disponibles, para ello teclee la letra 'L' y localiza el código hexadecimal para las particiones LVM

El código hexadecimal para las particiones LVM es el siguiente:

8e Linux LVM

Una vez localizado el código hexadecimal, solo restara teclearlo.

Con esto habremos concluido la generación de la particiones extendidas y lógicas para nuestro LVM, solo nos restara guardar los cambios al disco Para guardar los cambios al disco teclee la letra 'w'. Solo restara reiniciar el equipo para que los cambios hechos al disco duro sean visibles.

Manejo practico de LVM

Creando y destruyendo Volúmenes Físicos (PV)

Para los ejemplos de las siguientes secciones entenderemos que el disco duro sobre el que estaremos trabajando sera /dev/hda. Lo primero que haremos para comenzar a crear y administrar un LVM sera iniciar el sistema operativo en el nivel de ejecución 1, para hacerlo teclee en consola lo siguiente:

```
[BASH]# init 1
```

Una vez terminado, procederemos a revisar el estado de las particiones que recien añadimos en el capitulo anterior. Para hacerlo usaremos nuevamente el comando fdisk, esto para obtener la configuración de la nueva tabla de particiones, para ello teclee lo siguiente:

```
sh-3.2# fdisk /dev/hda
```

Para imprimir la tabla de particiones simplemente teclee la letra 'p' Como podemos observar , tenemos la partición /dev/hda5 con su respectivo identificador de LVM, sera sobre esta partición sobre la cual crearemos el Volumen Físico de nuestro LVM. Para crear el volumen físico sobre la partición /dev/hda5 , ejecutaremos:

```
sh-3.2# pvcreate /dev/hda5
```

Listando volúmenes físicos (PV)

Para verificar que se creo de manera correcta nuestro Volumen Físico ejecute

```
sh-3.2# pvdisplay
```

Borrando volúmenes físicos (PV)

Si quisieramos borrar el Volumen Físico que recien hemos creado ejecute

```
sh-3.2# pvremove /dev/hda5
```

El siguiente paso sera crear el Grupo de Volúmenes (VG) al cual pertenecerá nuestro Volumen Físico

Creando y destruyendo Grupos de Volúmenes (VG)

Para crear un grupo de volúmenes físicos, es decir, un almacén de espacio libre para nuestro Volumen Físico /dev/hda5, ejecutaremos:

```
sh-3.2# vgcreate serverLVM /dev/hda5
```

La palabra serverLVM es el nombre que yo asigne a mi Grupo de Volúmenes, usted puede asignar el nombre que mas le convenga.

Listando volúmenes físicos (VG)

Para verificar que se creo de manera correcta nuestro Grupo de Volúmenes ejecute

```
sh-3.2# vgdisk
```

Borrando volúmenes físicos (VG)

Para borrar un grupo de volúmenes (VG) primero se deberá haber borrado cualquier volumen lógico (LV), de otra forma no se lograra borrar el grupo de volúmenes. Si quisiéramos borrar el Grupo de Volúmenes que recien hemos creado ejecute

```
sh-3.2# vgremove serverLVM
```

Con esta acción lograremos que los Volúmenes Físicos (PV) queden huérfanos y con ello podrán ser borrados o asignados a otro grupo de volúmenes El siguiente paso sera crear el Volumen Lógico (LV)

Creando y destruyendo Volúmenes Lógicos (LV)

Para crear y agregar un Volumen Lógico (VG) a un Grupo de Volúmenes (VG) debemos tener en cuenta el espacio disponible en nuestro Grupo de Volúmenes (VG), usted puede verificar este valor ejecutando el siguiente comando:

```
sh-3.2# vgdisk
```

Tome en cuenta el valor ubicado dentro del rectángulo en blanco Como podemos apreciar, tenemos un espacio de 9.49 Giga-bytes para implementar nuestros Volúmenes Lógicos Para crear y agregar un Volumen Físico (VG) a nuestro Grupo de Volúmenes (VG) serverLVM teclee lo siguiente:

```
sh-3.2# lvcreate -n HOME-LVM -L6G serverLVM
```

-n HOME-LVM El parámetro -n indica que asignaremos un nombre a ese Volumen Lógico, en este ejemplo asignamos el nombre de HOME-LVM

-L6G El parámetro -L indica la cantidad de espacio que utilizaremos para nuestro Volumen Lógico, en este ejemplo asignamos 6 Giga-bytes de los 9.49 Giga-bytes disponibles

serverLVM Este parámetro indica a que Grupo de Volúmenes estamos asignando nuestro Volumen Lógico , es este ejemplo se lo estamos asignando a serverLVM

Listando volúmenes lógicos (LV)

Para verificar que se creo de manera correcta nuestro Volumen Lógico ejecute

```
sh-3.2# lvdisplay
```

Esta acción mostrara el siguiente mensaje:

Borrando volúmenes lógicos (LV)

Para destruir un volumen lógico, siempre que este desmontado, ejecutaremos:

```
sh-3.2# lvremove /dev/serverLVM/HOME-LVM
```

Note que hay que indicar el camino completo al volumen, no basta solo con el nombre ya que podría haber volúmenes lógicos con el mismo nombre pero pertenecientes a distintos grupos de volúmenes. El siguiente paso sera dar formato a los Volúmenes Lógicos recien creados. Solo restara reiniciar el equipo para que los cambios hechos al disco duro sean visibles.

Utilizando los LVM

Dando formato a los volúmenes lógicos

El siguiente paso sera dar formato al Volumen Lógico (VG) que recien creamos, para hacerlo haremos uso del comando mkfs. mkfs (“Make Filesystem”) es una herramienta estándar de Unix para formatear una partición de un disco duro con un específico sistema de ficheros, pudiendo ser ext2, ext3, ext4, msdos, vfat, ntfs, reiserfs y xfs. La sintaxis básica de este comando es la siguiente:
mkfs.[ext2|ext3|ext4|msdos|vfat|ntfs|reiserfs|xfs] partición

Una vez entendido el concepto de mkfs pasaremos a dar formato a nuestro Volumen Lógico, recordemos que nuestro Volumen Lógico esta alojado en la siguiente ruta:

/dev/serverLVM/HOME-LVM

Daremos formato al Volumen Lógico de la siguiente manera:

```
sh-3.2# mkfs.ext3 /dev/serverLVM/HOME-LVM
```

Lo cual deberá arrojar el siguiente mensaje:

Montando la partición de /home en el Volumen Lógico

1. El primer paso sera crear la carpeta en la cual montaremos nuestra partición de /home. Naturalmente, el lugar donde crearemos esta carpeta sera en la partición de /mnt lo cual haremos de la siguiente forma

```
[BASH]# mkdir /mnt/HOME-LVM
```

2. El segundo paso sera montar el Volumen Lógico en la carpeta que recien creamos

```
[BASH]# mount -v /dev/serverLVM/HOME-LVM /mnt/HOME-LVM
```

3. El siguiente paso sera copiar el contenido de la partición /home a la carpeta donde tenemos montado el Volumen Lógico, para ello usaremos el siguiente comando:

```
[BASH]# cp -Ra/home/* /mnt/HOME-LVM
```

Opciones	Descripción
cp	Copia el fichero o carpeta
-R	La opción -R indica que copiara recursivamente el contenido de una carpeta

-a La opción -a indica que preservara los permisos y enlaces simbólicos de cada fichero
 4.A continuación moveremos el contenido de la carpeta /home a otra carpeta, el nombre que yo asigne a esta carpeta sera “oldhome”

```
[BASH]# mkdir /oldhome
```

```
[BASH]# mv /home/ /oldhome/}}}} El motivo de esta acción es porque nuestra nueva /home sera la ubicada en la partición de /mnt/LVM-HOME.
```

5.Solo nos restara indicar al sistema que nuestra nueva /home sera montada en /mnt/LVM-HOME para ello abra y agregue al fichero /etc/fstab la siguiente linea

```
[BASH]# vi /etc/fstab
```

Device	Montaje	F.type	M.Options	DF	PN
LABEL=/1	/	ext3	defaults	1	1
LABEL=/boot1	/boot	ext3	defaults	1	2
tmpfs	/dev/shm	tmpfs	defaults	0	0
devpts	/dev/pts	devpts	gid=5,mode=620	0	0
sysfs	/sys	sysfs	defaults	0	0
proc	/proc	proc	defaults	0	0
/dev/serverLVM/HOME-LVM	/home	ext3	defaults	0	2
LABEL=SWAP-hda3	swap	swap	defaults	0	0

Opciones Descripción

Device	Este campo especifica la partición del sistema de ficheros, por ejemplo /dev/hda1.
Montaje	Este campo no puede contener el dispositivo entero (/dev/sda)
Filesystem type	Aquí se introduce el directorio donde se quiere que el dispositivo sea montado. Por ejemplo si la partición /dev/hda1 tiene el sistema de ficheros root, donde está el sistema se montará en /
Mount options	En este campo se indica que tipo de partición se trata, ext2, ext3, ext4, reiserfs, swap, iso9660
Dump Frecuency	Opciones de Escritura, Lectura y ejecución
Pass Number para el fsck	El programa dump consulta la entrada del /etc/fstab para ver cada cuanto tiempo debe hacer el backup. Normalmente tiene el valor 1 para filesystems basados en ext2 y ext3 y 0 para los demás.
	Este campo es usado por la utilidad fsck cuando la opción -A se especifica, normalmente en tiempo de arranque. Tiene valores:
	0 - No chequear el sistema
	1 - Debe ponerse para el filesystem root (/) y ordena al fsck a chequearlo primero
	2 - Hace el chequeo de la unidad, después del chequeo de los marcados con un 1

Expandiendo Volúmenes Lógicos

Expandir un Volumen Lógico(LV) es sencillo, para ello primero tendremos que verificar el espacio disponible en el Grupo de Volúmenes (VG), para ello teclee lo siguiente:

Nota: Esto debe ejecutarse en el nivel de ejecución 1

```
[BASH]# vgdisplay
```

Esta acción desplegará información acerca del Grupo de Volúmenes, tendremos que poner atención a la siguiente linea ya que esta nos informará el espacio disponible en el disco duro

```
Free PE/ Size      893 / 3,49 GB
```

Como podemos observar tanto en el recuadro en negro como en el recuadro en gris observamos que tenemos 3.49 GB de espacio disponible, por lo que expandiremos nuestro Volumen Lógico (VG) a 3 GB, para ello solo debemos teclear el siguiente mandato.

```
[BASH]# lvresize -L 3G /dev/serverLVM/HOME-LVM
```

Con esta acción habremos añadido el tamaño al Volumen Lógico (VG), por lo que ahora tendremos que hacer coincidir el sistema de archivos, en este caso EXT3 con el nuevo tamaño del volumen lógico, por lo que tenemos que hacer lo siguiente

```
[BASH]# resize2fs /dev/serverLVM/HOME-LVM
```

Ahora para comprobar que todo ha ido bien, ejecutamos el siguiente comando:

```
[BASH]# lvdisplay
```

```
LV Size 9.00 GB (Y nos fijamos en este valor) }}
```

Con lo que podemos observar que ahora tenemos 9 GB asignados a nuestro directorio /home, y ya no tendremos problemas de espacio en dicho directorio.

Reduciendo Volúmenes Lógicos

Como es natural con el paso del tiempo puede darse el caso que tengamos un Volumen Lógico (VG) que no necesite parte del espacio que habíamos reservado en un principio, y por lo tanto queremos quitar un trozo de ese Volumen Lógico (VG) para tenerlo disponible para asignarlo a otro de nuestros discos. Supongamos que tenemos que nuestra partición /home tiene demasiado espacio y queremos reducir el tamaño de dicho directorio la cual actualmente posee 9 GB de disco y queremos quitar 3 GB, con lo que nos quedaremos con 6 GB Lo primero que tenemos que hacer es verificar la información disponible en el Grupo de Volúmenes (VG), por lo tanto ejecutaremos el siguiente comando para comprobar cuánto espacio tenemos:

- Nota: Esto debe ejecutarse en el nivel de ejecución 1

```
[BASH]# lvdisplay /dev/serverLVM/HOME-LVM
```

LV Size 9.00 GB (Y nos fijamos en este valor) }} Ahora que conocemos el tamaño exacto vamos a reducir el volumen lógico, para eso tenemos que hacer:

```
[BASH]# resize2fs /dev/serverLVM/HOME-LVM 6GB
```

```
[BASH]# lvreduce -L -3G /dev/serverLVM/HOME-LVM (Reducimos el volumen lógico)}}
```

Con lo que ya tendremos nuestro volumen lógico reducido, ahora para comprobar que el proceso a funcionado correctamente volvemos a ejecutar el mismo comando de antes y observamos que el nuevo tamaño es 30 GB:

```
[BASH]# lvdisplay /dev/serverLVM/HOME-LVM
```

```
[BASH]# LV Size 6.00 GB (Y nos fijamos en este valor)}}
```

104.1 EXTRAS Btrfs

Btrfs (B-tree FS o normalmente pronunciado "Butter FS") es un sistema de archivos copy-on-write desarrollado por Oracle Corporation (entre otras) para GNU/Linux. Su objetivo es sustituir al actual sistema de archivos ext3, eliminando el mayor número de sus limitaciones, en especial con el tamaño máximo de los ficheros; además de la adopción de nuevas tecnologías no soportadas por ext3. Se afirma también que se "centrará en la tolerancia a fallos, reparación y fácil administración".

Este sistema de ficheros ha recogido ideas de ReiserFS y del sistema desarrollado por la desaparecida Sun Microsystems: ZFS. La capacidad de trabajar con la lógica y el almacenamiento físico recuerda mucho al tratamiento de volúmenes de ZFS, propiedad de Oracle en la actualidad. Varias distribuciones están tratando de implantarlo como sistema de ficheros predeterminado. Algunas de las características de Btrfs:

Empaquetado eficiente en espacio de archivos pequeños y directorios indexados

- Asignación dinámica de inodos (no se fija un número máximo de archivos al crear el sistema de archivos)
- Snapshots escribibles y snapshots de snapshots
- Subvolúmenes (raíces del sistema de archivos internas separadas)
- Mirroring y Stripping a nivel de objeto
- Comprobación de datos y metadatos (alta seguridad de integridad)
- Compresión
- Copy-on-write del registro de todos los datos y metadatos
- Gran integración con device-mapper para soportar múltiples dispositivos, con varios algoritmos de RAID incluidos
- Comprobación del sistema de archivos sin desmontar y comprobación muy rápida del sistema de archivos desmontado
- Copias de seguridad incrementales eficaces y mirroring del sistema de archivos
- Actualización desde ext3 a Btrfs, y reconversión a ext3 al momento de la actualización
- Modo optimizado para SSD (activado a través de una opción de montaje)
- Defragmentación sin desmontar

Btrfs en Debian

Para realizar tareas con btrfs es necesario realizar la instalación de éste

```
root@cli:~# apt-get install btrfs-tools
```

Btrfs en CentOS

```
# yum update
# yum install btrfs-progs
```

Una vez instaladas las herramientas de btrfs, los procedimientos son similares.

Para crear un volumen con varios discos (que evidentemente han de existir y ser reconocidos por el sistema):

```
root@cli:~# mkfs.btrfs /dev/sdb /dev/sdc /dev/sdd
WARNING! - Btrfs Btrfs v0.19 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using
adding device /dev/sdc id 2
adding device /dev/sdd id 3
fs created label (null) on /dev/sdb
        nodesize 4096 leafsize 4096 sectorsize 4096 size 6.00GB
```

```
Btrfs Btrfs v0.19
```

La herramienta btrfs es el comando único de gestión de este sistema de ficheros pero con muchos subcomandos para las diferentes operaciones. Para ver información del sistema de ficheros creado en el ejemplo anterior se puede usar como argumento cualquiera de las unidades de disco:

```
root@cli:~# btrfs filesystem show
failed to read /dev/sr0
Label: none  uuid: 43e0f114-fbda-4f5a-906e-4b80ae216522
          Total devices 3 FS bytes used 28.00KB
          devid      3 size 2.00GB used 519.94MB path /dev/sdd
          devid      2 size 2.00GB used 212.75MB path /dev/sdc
          devid      1 size 2.00GB used 531.94MB path /dev/sdb
```

```
Btrfs Btrfs v0.19
```

Es posible crear redundancia de datos y de metadatos o de configurarlo para mejor rendimiento usando las opciones -m (metadatos) o -d (datos o ambos) y palabras clave como single, raid0, raid1. Algunos ejemplos:

```
root@cli:~# mkfs.btrfs -d raid0  /dev/sdd /dev/sdc
WARNING! - Btrfs Btrfs v0.19 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using

failed to read /dev/sr0
failed to read /dev/sr0
adding device /dev/sdc id 2
fs created label (null) on /dev/sdd
          nodesize 4096 leafsize 4096 sectorsize 4096 size 4.00GB
Btrfs Btrfs v0.19
root@cli:~# mkfs.btrfs -m single /dev/sdb

WARNING! - Btrfs Btrfs v0.19 IS EXPERIMENTAL
WARNING! - see http://btrfs.wiki.kernel.org before using

fs created label (null) on /dev/sdb
          nodesize 4096 leafsize 4096 sectorsize 4096 size 2.00GB
Btrfs Btrfs v0.19
root@cli:~# btrfs filesystem show
failed to read /dev/sr0
Label: none  uuid: 4448d878-ab90-43de-8fb0-f268899544a2
          Total devices 2 FS bytes used 28.00KB
```

```

        devid      1 size 2.00GB used 437.50MB path /dev/sdd
        devid      2 size 2.00GB used 417.50MB path /dev/sdc

Label: none  uuid: 058b97cf-cf08-4982-98d1-9ef0a4772245
        Total devices 1 FS bytes used 28.00KB
        devid      1 size 2.00GB used 20.00MB path /dev/sdb

Btrfs Btrfs v0.19

```

Para montar el volumen se usa cualquier dispositivo de éste.

```

root@cli:~# mount /dev/sdc /mnt/
root@cli:~# df -T | grep btrfs
/dev/sdc                                btrfs          4194304
184      3725696   1% /mnt

```

Para que sea permanente agregar a /etc/fstab

```

[...]
/dev/sdb /mnt           btrfs defaults 0      1
[...]

```

Se extrae información del volumen con :

```

root@cli:~# btrfs filesystem df /mnt
Data, RAID0: total=409.50MB, used=128.00KB
Data: total=8.00MB, used=0.00
System, RAID1: total=8.00MB, used=4.00KB
System: total=4.00MB, used=0.00
Metadata, RAID1: total=204.75MB, used=24.00KB
Metadata: total=8.00MB, used=0.00

```

Btrfs incorpora la compresión en el sistema de ficheros con varios algoritmos (lzo, zlib) y se define en el momento de montar.

```
root@cli:~# mount -o compress=lzo /dev/sdc /mnt/
```

Agregar un nuevo disco al volumen:

```

root@cli:~# btrfs device add /dev/sdb /mnt
root@cli:~# btrfs filesystem show
failed to read /dev/sr0
Label: none  uuid: 4448d878-ab90-43de-8fb0-f268899544a2
        Total devices 3 FS bytes used 156.00KB
        devid      1 size 2.00GB used 437.50MB path /dev/sdd
        devid      2 size 2.00GB used 417.50MB path /dev/sdc

```

```
devid      3 size 2.00GB used 0.00 path /dev/sdb
```

```
Btrfs Btrfs v0.19
```

Se pueden crear subvolúmenes que se muestran como directorios pero tienen su propio identificador en el sistema de ficheros. Estos subvolúmenes se pueden montar con -o y el identificador del subvolumen o la cadena (path). Ejemplo:

```
root@cli:~# btrfs subvolume create /mnt/svol
Create subvolume '/mnt/svol'
root@cli:~# btrfs subvolume list /mnt/
ID 259 top level 5 path svol
root@cli:~# ls /mnt
1 2 3 4 5 svol
root@cli:~# btrfs subvolume create /mnt/svol/vol2
Create subvolume '/mnt/svol/vol2'
root@cli:~# ls /mnt/svol/
vol2
root@cli:~# btrfs subvolume list /mnt/
ID 259 top level 5 path svol
ID 260 top level 5 path svol/vol2
root@cli:~# mkdir /borrame
root@cli:~# mount -o subvolid=260 /dev/sdd /borrame
root@cli:~# ls /borrame/
root@cli:~# touch /borrame/nuevoFile
root@cli:~# ls /mnt/svol/vol2/
nuevoFile
```

Es posible crear instantáneas del sistema de ficheros (snapshots) que nos permite tomar un momento de referencia. Esta snapshot tiene características de subvolumen y es un objeto más en el volumen. Las instantáneas tiene un espacio referenciado pero no ocupan en tanto no se modifica el FS original.

```
root@cli:~# btrfs subvolume snapshot /mnt/svol/ /mnt/svol/snap01
Create a snapshot of '/mnt/svol/' in '/mnt/svol/snap01'
root@cli:~# ls -l /mnt/svol/snap01/
total 0
drwxr-xr-x 1 root root 0 dic 18 12:10 vol2
```

El comando btrfs-convert permite convertir FS ext2,3,4 a Btrfs y viceversa.

104.1 EXTRAS XFS

XFS es un sistema de ficheros de 64 bits desarrollado por Silicon Graphics y liberado en el año 2000. En wikipedia se puede encontrar una breve información sobre las características de este

sistema de ficheros:

“XFS soporta un sistema de archivos de hasta 8 exabytes, aunque esto puede variar dependiendo de los límites impuestos por el sistema operativo. En sistemas GNU/Linux de 32 bits, el límite es 16 terabytes.

Registro de bitácora (*journaling*)

*XFS provee soporte para llevar un registro (*journaling*), donde los cambios al sistema de archivos primero son escritos a un diario o journal antes de que se actualicen los datos del disco. El journal es un buffer circular de bloques del disco que no son parte del sistema de archivos. En XFS el registro (journal) contiene entradas ‘lógicas’ que describen a un alto nivel las operaciones que se están realizando, al contrario de otros sistemas de archivo con un registro (journal) ‘físico’, que guardan una copia de los bloques modificados durante cada transacción. Las actualizaciones del registro (journal) se realizan asincrónicamente para evitar una baja en el rendimiento. En el caso de una caída repentina del sistema, las operaciones inmediatamente anteriores a la caída pueden ser terminadas, garantizando así la consistencia del sistema. La recuperación se realiza automáticamente a la hora del montaje del sistema de archivos y la velocidad de recuperación es independiente del tamaño del sistema de archivos. Incluso si alguna información que fuese modificada inmediatamente antes de la caída del sistema no fuese escrita al disco, XFS se encarga de borrar todos los bloques de datos sin escribir, eliminando así cualquier compromiso de seguridad.*

Grupos de asignación

Los sistemas de archivos XFS están particionados internamente en grupos de asignación, que son regiones lineares de igual tamaño dentro del sistema de archivos. Los archivos y los directorios pueden crear grupos de asignación. Cada grupo gestiona sus inodos y su espacio libre de forma independiente, proporcionando escalabilidad y paralelismo — múltiples hilos pueden realizar operaciones de E/S simultáneamente en el mismo sistema de archivos.

LVM

Es posible aumentar la capacidad de sistemas de ficheros XFS: `xfs_growfs` es ideal para particiones LVM”

Para poder usar este sistema de ficheros es necesario instalar:

```
[root@instructor ~]# yum install xfsprogs
```

o en Debian

```
root@cli:~# apt-get install xfsprogs
```

XFS es más “tradicional” en su administración que btrfs:

```
root@cli:~# mkfs.xfs -f /dev/sdd
meta-data=/dev/sdd              isize=256    agcount=4, agsize=131072 blks
                                =          sectsz=512   attr=2, projid32bit=0
data     =              bsize=4096   blocks=524288, imaxpct=25
                                =          sunit=0    swidth=0 blks
naming   =version 2            bsize=4096   ascii-ci=0
log      =internal log         bsize=4096   blocks=2560, version=2
```

```
        =          sectsz=512    sunit=0 blks, lazy-count=1
realtime =none          extsz=4096   blocks=0, rtextents=0
root@cli:~# mount /dev/sdd /mnt
root@cli:~# df -T --type xfs
S.ficheros     Tipo 1K-bloques Usados Disponibles Uso% Montado en
/dev/sdd       xfs      2086912  32928      2053984  2% /mnt
```

Se asigna un sistema de ficheros con mkfs.xfs, procede al montaje sin mayor problema y se verifica su elevado uso ya que es un disco pequeño y XFS, es muy rápido con archivos "relativamente grandes", mientras más grande sean los archivos a manejar con XFS, mejor.

104.2. Mantenimiento de la integridad de sistemas de archivos.

Peso en el examen de certificación: 2 puntos.

Objetivo: Mantener un sistema de ficheros estandar, así como los datos extras asociados al "journaling" del sistema de ficheros.

Conceptos y áreas de conocimiento:

- Verificar la integridad del sistema de ficheros.
- Monitorizar el espacio libre y los inodos.
- Reparar pequeños problemas en un sistema de ficheros.

Términos y utilidades

- du
- df
- fsck
- e2fsck
- mke2fs
- debugfs
- dumpe2fs
- tune2fs
- xfs tools

104.2.0. Introducción

Nuestros sistemas de archivos no son infalibles aunque trabajemos en las mejores condiciones posibles. Es por ello que este apartado nos presenta algunos comandos para monitorizar el tamaño libre y la comprobación y reparación del sistema de archivo si tenemos algún tipo de error.

Aunque estemos trabajando en las mejores condiciones posibles, tarde o temprano los sistemas de ficheros terminan fallando y por tanto presentando problemas. Estos problemas suelen aparecer por:

- Apagar el ordenador inadecuadamente, cortes de alimentación o fallos en el hardware, que puede provocar que una operación de escritura no pueda completarse y por tanto los datos se perderían impidiendo al kernel sincronizar la cache de memoria con el disco duro, además de que parte del sistema de ficheros quedaría marcados como que está en uso.
- Si un sistema de archivos se llena hasta el límite de su capacidad, podría causar que un programa/demonio deje de funcionar. Por ejemplo en Linux muchos demonios (apache, mysql, etc.) usan el directorio /tmp para guardar información mientras se ejecutan y por tanto si no tienen espacio en el sistema, dejarían de funcionar correctamente.
- Además si un sistema de archivos se queda sin inodos libres (número finito en cualquier sistema de archivos) normalmente cuando dicha partición contiene muchos ficheros pequeños (por ejemplo archivos de logs), el sistema operativo no podrá crear nuevos objetos provocando su caída. Tenga en cuenta que un sistema puede tener espacio disponible y no tener inodos libres, y al contrario tener inodos libres pero no espacio disponible.

Como habrá podido observar es muy importante regularmente (cron) monitorizar el espacio libre y verificar la integridad del disco duro para prevenir posibles problemas. En caso de tener problemas existen utilidades en Linux para repararlos. Cuando un sistema de archivos está muy cerca de agotar su capacidad, tendremos que eliminar ficheros que no sean necesarios o redimensionar dicho sistema de archivo si esto es posible, pero en el caso de que agotemos (improbable) deberíamos volver a crear el sistema de archivos con un número mayor de inodos ó bien borrar una gran cantidad de archivos.

Los comandos más utilizados en Linux para mantener la integridad de los sistemas de archivos son:

- **df**: Monitorización del espacio libre y inodos libres del sistema de archivos.
- **du**: Monitorización del uso del espacio en disco.
- **fsck**: Verificación y reparación si se desea de la integridad de un sistema de archivos.

104.2.2. Monitorizar el espacio e inodos libres del disco (df).

El comando **df** proporciona información sobre el espacio en disco disponible y sobre los inodos libres de los sistemas de archivos montados en directorio (como por ejemplo /, /proc, /home, etc.). Si ningún directorio/fichero es proporcionado como argumento mostrará información de todos los sistemas de archivos montados incluidos en el fichero */etc/fstab*. La sintaxis de este comando es:

df [Opciones] [Fichero]

- **[Opciones]**: Lista de argumentos en formato corto o largo para obtener información más detallada en la ejecución del comando.
- **[Fichero]**: Normalmente se le indica un directorio, en el cual se encuentra montada una partición, pero también podemos indicar el nombre de un fichero. En este último caso se mostrará información sobre la partición en la que se ubica dicho fichero o directorio.

Las opciones más importantes que se usan junto con este comando son:

- **-h, --human-readable**: Añade automáticamente un sufijo (M (MegaBytes), K (KiloBytes), T (TeraBytes), etc.) haciendo que los resultados estén en un formato legible para las personas. Si no se coloca esta opción los resultados se muestra en kilobytes.
- **-i, --inodes**: Muestra información de los inodos disponibles en vez de la información por **defecto** de espacio libre.
- **-l, --local**: Se limita a mostrar únicamente los sistemas de archivos locales, excluyendo aquellos montados via nfs, sshfs, samba, etc.
- **-T, --print-type**: Muestra junto a cada sistema de archivos su tipo (ext3, ext3, minix, vfat, etc.)
- **-a, --all**: Muestra sistemas de archivos ficticios (/proc, /sys, etc.).
- **--help**: Muestra ayuda del comando y relación de todas los argumentos que se pueden usar.

Algunos ejemplos del uso de este comando y de sus opciones son las siguientes:

1.- Mostrar el uso del espacio en disco de todos los sistemas de archivos (excluyendo

los ficticios) del sistema. Se puede comprobar rápidamente que el directorio raíz está a un 39 % de uso y que el de arranque (/boot) está al 11 % de uso. Al no decirle nada en relación al formato legible para las personas los datos se muestra en bloque de 1K, por ejemplo el tamaño de /boot es de 233191 KBytes, lo que equivale aproximadamente a 233 MBytes si usamos 1 KB equivale a 1000 B ó de 228 MB si usamos 1 KB equivale a 1024 B (233191 / 1024).

S.ficheros	Bloques de 1K	Usado	Dispon	Uso%	Montado en
/dev/mapper/cabinadisco-root	2814080	1020668	1650464	39%	/
udev	503356	4	503352	1%	/dev
tmpfs	204260	296	203964	1%	/run
none	5120	0	5120	0%	/run/lock
none	510648	0	510648	0%	/run/shm
/dev/sda1	233191	23523	192217	11%	/boot

2.- Mostrar el uso del espacio en disco de todos los sistemas de archivos montados en el sistema en un formato legible. Se puede observar como el directorio raíz (/) es de 2,7 Gigabytes, que tiene usado 997 MegaBytes, por tanto tiene disponible 1,6 GigaBytes que equivale al 39 % de espacio libre. Por el contrario la partición (/dev/sda1) que se monta en el directorio /boot es de 38 MegaBytes, de los cuales 23 MegaBytes están usados, lo que equivale a 11% de espacio usado.

S.ficheros	Tam.	Usado	Disp.	% Uso	Montado en
/dev/mapper/cabinadisco-root	2,7G	997M	1,6G	39%	/
udev	492M	4,0K	492M	1%	/dev
tmpfs	200M	296K	200M	1%	/run
none	5,0M	0	5,0M	0%	/run/lock
none	499M	0	499M	0%	/run/shm
/dev/sda1	228M	23M	193M	11%	/boot

3.- Mostrar el uso del espacio de inodos libres de todos los sistemas de archivos montados en el sistema en un formato legible. Se puede observar como el directorio raíz (/) tiene usado el 33 % de sus inodos, por el contrario el sistema de archivos /dev/sda1 únicamente tiene usado el 1 % de sus inodos (frente a su 11 % de ocupación en relación a su espacio en disco). Está claro que ninguno de estos sistemas de archivos se está acercando en su consumo de inodos. Si el sistema continúa con esta tónica de utilización, lo más probable es que el volumen /boot agote su capacidad antes de agotar los inodos libres.

S.ficheros	Inodos	IUsado	ILibre	IUso%	Montado en
/dev/mapper/cabinadisco-root	175K	58K	118K	33%	/
udev	123K	496	123K	1%	/dev
tmpfs	125K	372	125K	1%	/run
none	125K	2	125K	1%	/run/lock
none	125K	1	125K	1%	/run/shm
/dev/sda1	122K	230	122K	1%	/boot

4.- Mostrar el uso del espacio en disco disponible de la partición en donde se encuentra el fichero /etc/fstab en formato legible por las personas. Se puede observar como dicho fichero se encuentra en el directorio raíz (/) y por tanto este comando sólo muestra información de dicha partición.

S.ficheros	Tam.	Usado	Disp.	% Uso	Montado en
/dev/mapper/cabinadisco-root	2,7G	997M	1,6G	39%	/

104.2.3. Monitorizar el uso del espacio en disco (du).

El comando **df** nos muestra información del espacio usado de un sistema de archivo, pero no podemos saber en que está gastado, es decir que ficheros y directorios. Para eso podemos usar el comando **du**, que nos muestra directorio por directorio (**recursivamente**) el uso del espacio en disco. La sintaxis de este comando es:

du [Opciones] [Directorios]

- **[Opciones]:** Lista de argumentos en formato corto o largo para obtener información más detallada en la ejecución del comando.
- **[Directorios]:** Normalmente se le indica un directorio o una lista de directorios (con espacios en blanco). Si no se indica el comando dà como salida la información del directorio de trabajo actual.

Las opciones más importantes que se usan junto a este comando son:

- **-h, --human-readable:** Añade automáticamente un sufijo (M (MegaBytes), K (KiloBytes), T (TeraBytes), etc.) haciendo que los resultados esten en un formato legible para las personas. Si no se coloca esta opción los resultados se muestra en Kilobytes.
- **-a, --all:** Además de mostrar información del tamaño de los directorios, muestra el de los archivos contenidos en el directorio.
- **-c, --total:** Realiza una sumatoria total del tamaño ocupado por todos los directorios mostrados en el informe.
- **-s, --summarize:** Muestra un sumario por cada uno de los directorios especificados (en dicho total introduce la suma de todos los subdirectorios) pero no incluye los totales encontrados en cada directorio.
- **-S, --separate-dirs:** Excluye los subdirectorios de las sumas y por tanto de los totales, limitándose a mostrar el tamaño de los archivos contenidos en el directorio, pero no de los subdirectorios contenidos en el directorio.

Veamos algunos ejemplos para comprender la diferencia que existe entre los argumentos mas importantes. En todos los ejemplos se ha usado la opción de realizar una sumatoria y que el formato sea legible para que sea más fácil reconocer las diferencias.

```
LPIC-1# du -cha
4,0K ./directorio1/subdirectorio
8,0K ./directorio1
4,0K ./directorio2
4,0K ./archivo.txt
20K .
20K total
LPIC-1# du -ch
4,0K ./directorio1/subdirectorio
8,0K ./directorio1
4,0K ./directorio2
20K .
20K total
LPIC-1# du -chs
20K .
20K total
LPIC-1# du -chS
4,0K ./directorio1/subdirectorio
4,0K ./directorio1
4,0K ./directorio2
8,0K .
20K total
LPIC-1# du -chSs
8,0K .
20K total
```

En el primer ejemplo se muestran los archivos, directorios y subdirectorios del directorio actual (con el tamaño de cada uno de ellos). Como se muestra todo, observará que la suma coincide ($4\text{ K (subdirectorio)} + 8\text{ K (directorio)} + 4\text{ K (directorio)} + 4\text{ K (archivo)} = 20\text{ K}$).

- En el segundo ejemplo (se quita la opción -a) no se muestran los archivos, sin

embargo el comando si realizar la suma de todos los elementos, por eso sigue diciendo que tiene 20K.

- En el tercer ejemplo le hemos añadido la opción -s, lo que implica que va a realizar la suma de todos los directorios y archivos, pero no los muestra, sin embargo puede observar como el total sigue diciendo que es de 20K.
- En el cuarto ejemplo le hemos añadido la opción -S, lo que implica que a realizar la sumatoria no se incluirá a los subdirectorios (opcion -S), observé como ahora en el directorio1 su tamaño es de 4K, en vez de 8K como había aparecido en los anteriores ejemplos. Puede observar tambien que el total (20K total) siempre muestra todo indistintamente de la opción que se ponga al comando.
- Y por ultimo en el quinto ejemplo al añadir la opción -sS, no se muestra el tamaño de los directorios (aunque si que se suma) y ádemas no se tiene en cuenta el tamaño de los subdirectorios. Puede observar tambien que el total (20K total) siempre muestra todo indistintamente de la opción que se ponga al comando.

104.2.4. Verificar y reparar la integridad de un sistema de archivos (fsck).

Tal y como se adelanto en la introducción de este tema, los sistemas de archivos tarde o temprano terminan por fallar. Es por este motivo que los sistemas operativos Linux disponen del comando fsck para chequear y reparar (si se cree conveniente) uno, varios o todos los sistemas de archivos encontrados. Este comando utiliza la información de lo que se conoce como superbloque que se encuentra en el bloque 1 de la partición. Debido a la importancia de este superbloque se realizan copias del mismo en intervalos regulares del sistema de archivo (por defecto se guarda cada 8192 bloques.). Al arrancar un sistema operativo Linux se realizará una comprobación de todos los sistemas archivos relacionados en */etc/fstab* (**antes de que se monten**). En el caso de que no se especifique ningún sistemas de archivo como argumento se verificará y repararán aquellos indicados en el fichero */etc/fstab*.

En el caso de que el directorio padre del sistema de archivo no pudiera ser encontrado los ficheros con errores serán salvados al directorio */lost+found*.

Es normal usar sistemas operativos en formato Live-CD para realizar verificaciones y reparaciones sobre los sistemas de archivos con fallos. Cuando se ejecuta este comando devuelve un valor que es la suma de las condiciones de salida. Estos valores son:

- 0 (no existe errores en el sistema de archivos).
- 1 (existían errores y se han corregido).
- 2 (el sistema operativo debe reiniciarse).
- 4 (errores del sistema de ficheros sin corregir).
- Etc.

Durante la ejecución de este comando se realizan los siguientes pasos:

- Comprobar inodos, bloques y tamaños.
- Comprobar la estructura de los directorios.
- Comprobar la conectividad de los directorios.
- Comprobar las referencias.
- Comprobar el total de la información.

La sintaxis de este comando es:

fsck [Opciones] -t fstype [Sistemas de archivos]

- **[Opciones]:** Lista de argumentos en formato corto o largo para ejecutar dicho comando con diferentes parámetros.
- **-t fstype:** Tipo o tipos de archivos que serán chequeados.
- **[Sistemas de archivos]:** Sistema/s a chequear. Podemos utilizar para indicar el sistema de archivo, el nombre del dispositivo (por ejemplo /dev/sdb1), punto de montaje (por ejemplo /home), el parámetro UUID (por ejemplo: UUID=8868abf6-67c6-24g3-87d9-bgf7826276).

Las opciones más importantes que se usan junto a este comando son:

- **-A:** ejecución del comando fsck con todos los sistemas de archivos que se encuentren en el fichero /etc/fstab. Esta opción se suele en la carga del sistema operativo.
- **-p:** repara automáticamente el sistema de archivos sin hacer preguntas.
- **-y:** responde automáticamente "yes" a todas las preguntas interactivas. Hay que tener mucho cuidado con esta opción.
- **-v:** despliga más información (verbose).
- **-f:** fuerza una comprobación aunque parezca que el sistema de archivos parece limpio.
- **-b:** Se usa este argumento cuando se desea usar una copia de superbloque alternativa (no la del superbloque 1). Es usado normalmente para restaurar un superbloque defectuoso.

Algunos ejemplos del uso de este comando y de sus opciones son las siguientes:

1.- Se realiza el checkeo de un sistema de archivo y se comprueba que está limpio.

```
LPIC-1# fsck /dev/sdb1
fsck desde util-linux 2.19.1
e2fsck 1.41.14 <22-Dec-2010>
/dev/sdb1: recuperando el fichero de transacciones
/dev/sdb1: limpio, 11/131072 ficheros, 27083/523264 bloques
```

2.- Se fuerza la comprobación aunque esté limpio. Si hubiéramos puesto la opción -y, no hubiera preguntado la pregunta del final.

```
LPIC-1# fsck -f /dev/sdb1
fsck desde util-linux 2.19.1
e2fsck 1.41.14 <22-Dec-2010>
Paso 1: Verificando nodos-i, bloques y tamaños
Paso 2: Verificando la estructura de directorios
Paso 3: Revisando la conectividad de directorios
Paso 4: Revisando las cuentas de referencia
Paso 5: Revisando el resumen de información de grupos
/dev/sdb1: 11/131072 ficheros <0.0% no contiguos>, 27083/523264 bloques
LPIC-1# fsck -f /dev/sd
sda sda1 sda2 sda5 sdb sdb1
LPIC-1# fsck -f /dev/sda5
fsck desde util-linux 2.19.1
fsck: fsck.LVM2_member: no se encontró
fsck: Error 2 mientras se ejecutaba fsck. LVM2_member para /dev/sda5
LPIC-1# fsck -f /dev/sda1
fsck desde util-linux 2.19.1
e2fsck 1.41.14 <22-Dec-2010>
/dev/sda1 está montado.

WARNING!!! The filesystem is mounted. If you continue you ***WILL***
cause ***SEVERE*** filesystem damage.

De verdad quiere continuar? <s/n>?
```

104.2 EXTRA

104.2 EXTRA Comprobación en XFS y Btrfs

Existen herramientas especializadas en la comprobación de los sistemas de ficheros XFS y btrfs.

```
root@cli:~# xfs_
xfs_admin      xfs_db        xfs_growfs     xfs_mdrestore  xfs_quota
xfs_bmap       xfs_estimate   xfs_info       xfs_metadump   xfs_repair
xfs_check      xfs_freeze    xfs_io         xfs_mkfile    xfs_rtcp
xfs_copy       xfs_fsr       xfs_logprint  xfs_ncheck
```

para la comprobación de un sistema de ficheros XFS ejecutaremos:

```
root@cli:~# xfs_check -v /dev/sde1 | more
setting block 0/0 to sb
setting block 0/4 to freelist
setting block 0/5 to freelist
setting block 0/6 to freelist
setting block 0/7 to freelist
setting block 0/1 to btbno
setting block 0/118 to free1
setting block 0/119 to free1
setting block 0/120 to free1
setting block 0/121 to free1
setting block 0/122 to free1
setting block 0/123 to free1
setting block 0/124 to free1
(...)
```

Para visualizar información del FS cuando está montado:

```
root@cli:~# xfs_info /dev/sde1
meta-data=/dev/sde1              isize=256    agcount=4, agsize=30518 blks
                                =          sectsz=512  attr=2
data      =             bsize=4096   blocks=122070, imaxpct=25
          =          sunit=0    swidth=0 blks
naming    =version 2            bsize=4096   ascii-ci=0
log       =internal             bsize=4096   blocks=1200, version=2
          =          sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                 extsz=4096   blocks=0, rtextents=0
```

Para ver los nombres de los ficheros ligados a su inodo:

```
root@cli:~# xfs_ncheck /dev/sde1 | more
256 fich.93.doc
257 fich.94.doc
258 fich.95.doc
```

```
259 fich.96.doc
260 fich.97.doc
261 fich.98.doc
262 fich.99.doc
263 fich.100.doc
131 fich.0.doc
132 fich.1.doc
133 fich.2.doc
```

Para tener información detallada por grupos de cilindros:

```
root@cli:~# xfs_logprint /dev/sde1 | more
xfs_logprint: /dev/sde1 contains a mounted and writable filesystem
xfs_logprint:
    data device: 0x841
    log device: 0x841 daddr: 488320 length: 9600

cycle: 1      version: 2          lsn: 1,0      tail_lsn: 1,0
length of Log Record: 20          prev offset: -1          num ops: 1
uuid: 8a137379-69c2-48ad-8eaf-9a8589335380  format: little endian linux
h_size: 32768
-----
Oper (0): tid: b0c0d0d0  len: 8  clientid: LOG  flags: UNMOUNT
Unmount filesystem
-----
cycle: 1      version: 2          lsn: 1,2      tail_lsn: 1,2
length of Log Record: 512         prev offset: 0          num ops: 5
uuid: 8a137379-69c2-48ad-8eaf-9a8589335380  format: little endian linux
h_size: 32768
-----
Oper (0): tid: 75c9dc7d  len: 0  clientid: TRANS  flags: START
-----
Oper (1): tid: 75c9dc7d  len: 16  clientid: TRANS  flags: none
TRAN:     type: CHECKPOINT        tid: 75c9dc7d        num_items: 2
Oper (2): tid: 75c9dc7d  len: 56  clientid: TRANS  flags: none
INODE: #regs: 2  ino: 0x80  flags: 0x1  dszie: 0
        blkno: 64  len: 16  boff: 0
Oper (3): tid: 75c9dc7d  len: 96  clientid: TRANS  flags: none
INODE CORE
magic 0x494e mode 040755 version 2 format 1
nlink 2 uid 0 gid 0
```

```
atime 0x52b26370 mtime 0x52b26364 ctime 0x52b26364
size 0x6 nblocks 0x0 extsize 0x0 nextents 0x0
naextents 0x0 forkoff 0 dmevmask 0x0 dmstate 0x0
flags 0x0 gen 0x0
-----
Oper (4): tid: 75c9dc7d len: 0 clientid: TRANS flags: COMMIT
```

Aún con la seguridad ante posibles errores de Btrfs, existe una herramienta de comprobación no integrada en btrfs:

```
root@cli:/# btrfsck /dev/sde2
checking extents
checking fs roots
checking root refs
found 28672 bytes used err is 0
total csum bytes: 0
total tree bytes: 28672
total fs tree bytes: 8192
btree space waste bytes: 23875
file data blocks allocated: 0
referenced 0
Btrfs Btrfs v0.19
```

104.2 EXTRA debugfs

Herramienta que permite acceder a un sistema de ficheros ext y realizar operaciones de control sobre los objetos del filesystem:

Se ejecuta con:

```
root@cli:~# debugfs
```

Se muestra un nuevo prompt y ya estamos en la aplicación. En esta práctica se va identificar un fichero y se modificará algún campo. Posteriormente se eliminara cualquier valor en el inodo y se pasará la herramienta de comprobación de ext para repararlo.

```
root@cli:# debugfs -w /dev/sde1
debugfs 1.42.5 (29-Jul-2012)
debugfs: ls

2 (12) . 2 (1012) .. 13 (20) fichero2.txt 15 (20) fichero4.txt
20 (984) fichero9.txt 19 (248) fichero8.txt 0 (776) 2.txt
11 (20) fichero0.txt 14 (20) fichero3.txt 16 (20) fichero5.txt
17 (20) fichero6.txt 18 (128) fichero7.txt 0 (816) 1.txt
12 (1024) fichero1.txt
(END)

debugfs: show_inode_info fichero2.txt
```

```
Inode: 13 Type: regular Mode: 0644 Flags: 0x80000
Generation: 2444731394 Version: 0x00000001
```

```

User: 0 Group: 0 Size: 133
File ACL: 0 Directory ACL: 0
Links: 1 Blockcount: 2
Fragment: Address: 0 Number: 0 Size: 0
ctime: 0x52b257b7 -- Thu Dec 19 03:19:35 2013
atime: 0x52b257b7 -- Thu Dec 19 03:19:35 2013
mtime: 0x52b257b7 -- Thu Dec 19 03:19:35 2013
EXTENTS:
(0):8454
(END)

debugfs: set_inode_field fichero2.txt mode 0770
debugfs: show_inode_info fichero2.txt

Inode: 13 Type: bad type Mode: 0770 Flags: 0x80000
Generation: 2444731394 Version: 0x00000001
User: 0 Group: 0 Size: 133
File ACL: 0 Directory ACL: 0
Links: 1 Blockcount: 2
Fragment: Address: 0 Number: 0 Size: 0
ctime: 0x52b257b7 -- Thu Dec 19 03:19:35 2013
atime: 0x52b257b7 -- Thu Dec 19 03:19:35 2013
mtime: 0x52b257b7 -- Thu Dec 19 03:19:35 2013
EXTENTS:
(0):8454
(END)
debugfs: clri fichero2.txt
debugfs: show_inode_info fichero2.txt

Inode: 13 Type: bad type Mode: 0000 Flags: 0x0
Generation: 0 Version: 0x00000000
User: 0 Group: 0 Size: 0
File ACL: 0 Directory ACL: 0
Links: 0 Blockcount: 0
Fragment: Address: 0 Number: 0 Size: 0
ctime: 0x00000000 -- Thu Jan 1 01:00:00 1970
atime: 0x00000000 -- Thu Jan 1 01:00:00 1970
mtime: 0x00000000 -- Thu Jan 1 01:00:00 1970
BLOCKS:

(END)
debugfs: quit

root@cli:~# fsck -f /dev/sde1
fsck de util-linux 2.20.1
e2fsck 1.42.5 (29-Jul-2012)
Paso 1: Verificando nodos-i, bloques y tamaños
Paso 2: Verificando la estructura de directorios
La entrada 'fichero2.txt' que está en / (2) tiene un nodo-i 13 borrado/no
utilizado. Borrar<s>? si
Paso 3: Revisando la conectividad de directorios
Paso 4: Revisando las cuentas de referencia
Paso 5: Revisando el resumen de información de grupos
Diferencias del mapa de bits del bloque: -8454
Arreglar<s>? si
La cuenta de bloques libres es incorrecta para el grupo #1 (7923,
contados=7924).
Arreglar<s>? si
La cuenta de bloques libres es incorrecta (462323, contados=462324).
Arreglar<s>? si
Diferencias del mapa de bits del nodo-i: -13

```

```
Arreglar<s>? si
La cuenta de nodos-i libres es incorrecta para el grupo #0 (2019,
contados=2020).
Arreglar<s>? si
La cuenta de nodos-i libres es incorrecta (122379, contados=122380).
Arreglar<s>? si
```

```
/dev/sde1: ***** EL SISTEMA DE FICHEROS FUE MODIFICADO *****
/dev/sde1: 20/122400 files (0.0% non-contiguous), 25956/488280 blocks
```

104.3. Control de montaje y desmontaje de los sistemas de archivos.

Peso en el examen de certificación: 3 puntos.

Objetivo: Configurar el montaje de sistemas de ficheros.

Conceptos y áreas de conocimiento:

- Montar y desmontar sistemas de ficheros de forma manual.
- Configurar el montaje de sistemas de fichero en tiempo de arranque.
- Configurar el montaje de sistemas de ficheros desde dispositivos de usuario removibles.

Términos y utilidades:

- /etc/fstab
- /media
- mount
- umount

104.3.1. Introducción.

La estructura de los sistemas de ficheros en GNU/Linux están generalmente divididos en particiones, unidas todas ellas en el punto de montaje raíz (/).

Los FS de los dispositivos removiles, tales como CD-ROM, diskettes, discos ZIP, etc., se unen a la raíz del sistema de la misma manera, como directorios (o puntos de montaje). En principio estos directorios destinados a los dispositivos están vacíos, a la espera de su montaje, puede darse el caso de que el directorio destinado a este fin contenga subdirectorios o archivos, en cuyo caso quedarán ocultos hasta que el dispositivo se desmonte.

104.3.2. Administrar la tabla de FS

Para que las diferentes particiones estén disponibles desde un primer momento es necesario montarlas durante el inicio del sistema, los dispositivos removiles también se usan frecuentemente y es aconsejable tenerlos preparados para usar los comandos de montaje.

Toda esta información se guarda en el fichero /etc./fstab. Los FS definidos en este fichero son revisados y montados durante el arranque del sistema. Sus entradas se consultan como fuente de información por defecto cuando los usuarios quieren montar dispositivos removiles.

En el siguiente ejemplo de /etc./fstab se puede ver que se trata de un fichero de texto con 6 campos en cada línea:

Device Mount point F. type M. Options DF PN
/dev/sda1 / ext2 defaults 1 1

```
/dev/sda5 /boot ext2 defaults 1 2  
/dev/sda9 /home ext2 defaults 1 2  
/dev/sda10 /tmp ext2 defaults 1 2  
/dev/sda11 swap swap defaults 0 0  
/dev/fd0 /mnt/floppy ext2 noauto,users 0 0  
/dev/hdc /mnt/cdrom iso9660 noauto,ro,users 0 0
```

Device: Este campo especifica la partición del filesystem, por ejemplo /dev/hda1. Este campo no puede contener el dispositivo entero (/dev/hda)

Mount point: Aquí se introduce el directorio donde se quiere que el dispositivo sea montado. Por ejemplo si la partición /dev/hda1 tiene el filesystem root, donde está el sistema se montará en /

Filesystem type: En este campo se indica que tipo de partición se trata, ext2, reiserfs, swap, iso9660 (CD-ROMS) etc. etc.

Mount options: Se explican más adelante, se separan por comas.

Dump frequency: El programa dump (para hacer backups) consulta la entrada del /etc/fstab para ver cada cuanto tiempo debe hacer el backup.

Normalmente tiene el valor 1 para FS basados en ext2 y 0 para los demás.

Pass number para el fsck: Este campo es usado por la utilidad fsck cuando la opción -A se especifica, normalmente en tiempo de arranque. Tiene valores:

0 - No chequear el sistema

1 - Debe ponerse para el FS root (/) y ordena al fsck a chequearlo primero

2 - Hace el chequeo de la unidad, después del chequeo de los marcados con un 1

En el ejemplo se tiene un disco duro SCSI (dev/sda) La primera partición

/dev/sda1 tiene el directorio root (/), la quinta contiene las imágenes del kernel para el arranque, la 9 para el directorio de los usuarios, la partición 10 es la temporal, la particion 11 para el sistema de swap. Luego se tiene la /dev/fd0 para diskettes y la /dev/hdc para el CD-ROM.

104.3.3. Montar un FS

Los FS se montan con el comando mount. Durante el arranque, los FS que no contienen un 0 en el pass number son chequeados y luego montados.

Después del arranque se pueden añadir más sistemas de archivos manualmente con el comando mount.

Sintaxis

Mount [opciones] device

Mount [opciones] directorio

Mount [opciones] device directorio

Se usa para montar (y así poder usar) FS dentro de la estructura del árbol del sistema. La primera y segunda opción consulta al fichero /etc/fstab para montar los dispositivos y así tomar las opciones que se le especifiquen en el /etc/fstab. La tercera opción es independiente del fichero /etc/fstab y monta el FS (device) en el directorio (directorio)

Opciones del comando mount:

Opción	Función
-a	Monta todos los filesystems especificados en el /etc/fstab menos los que tengan la opción

	noauto
-h	Ayuda del comando mount
-o	Especifica las opciones del mount en la linea de comandos
-r	Monta filesystems en modo de solo lectura
-t fstype	Especifica un tipo de filesystem
-v	Salida interactiva
-w	Monta filesystems de lectura/escritura

Opciones de montaje con mount, estas opciones se especifican en el fichero /etc/fstab o bien en la línea de comandos con la opción -o:

Opción	Función
async	Toda la E/S al sistema de ficheros debería hacerse asíncronamente.
auto	Puede montarse con la opción -a
defaults	Establece las opciones: rw, suid, dev, exec, auto, nouser y async. Es la opción por defecto en sistemas ext2
dev	Interpretar dispositivos especiales de caracteres o bloques en el sistema de ficheros
exec	Permitir la ejecución de binarios
noauto	Sólo puede montarse explícitamente (esto es, la opción -a no hará que el sistema de ficheros se monte)
noexec	No permitir la ejecución de ningún binario en el sistema de ficheros montado. Esta opción puede ser útil para un servidor que tiene sistemas de ficheros que contienen binarios para otras arquitecturas distintas de la suya.
nosuid	No permitir el efecto de los bits SUID ni SGID
nouser	Prohibir a un usuario ordinario (esto es, distinto de root) montar el sistema de ficheros. Esto es lo predeterminado
ro	Montar el sistema de ficheros en modo de sólo lectura.
rw	Montar el sistema de ficheros de lectura y escritura

suid	Permitir el efecto de los bits SUID y SGID
sync	Toda la E/S al sistema de ficheros debería hacerse síncronamente.
user	Permitir a un usuario ordinario montar el sistema de ficheros
users	Permite a cualquier usuario el montaje/desmontaje del sistema de ficheros

104.3.4. Desmontar un FS

Los FS pueden ser desmontados usando el comando umount. Cuando un FS es desmontado, los contenidos del árbol principal se actualizan, no pudiéndose usar (el umount) si el sistema de ficheros que se quiere desmontar está en uso.

Si el sistema de ficheros está en uso el comando umount dará un error. Esto puede ocurrir por ejemplo cuando tenemos abierto un fichero de un CDROM o un proceso está haciendo uso del mismo. Otros errores pueden surgir si quitamos dispositivos removibles sin antes desmontarlos: perdida de datos, corrupción de los mismos...

Sintaxis:

Umount [opciones] device

Umount [opciones] directorios

Desmonta un FS de un dispositivo o un directorio

Opciones de umount:

Opción	Uso
-a	Desmonta todos los filesystems descritos en /etc/mtab. Este fichero está mantenido por los comando mount y umount en tiempo real, se usa normalmente cuando se apaga/reinicia el PC.
-t fstype	Desmonta sólo los filesystems del tipo especificado

104.4. Administrar cuotas de disco.

Peso en el examen de certificación: 1 punto.

Objetivo: Gestión y mantenimiento de cuotas de disco para los usuarios del sistema.

Conceptos y áreas de conocimiento:

- Levantar cuotas de disco para un sistema de ficheros.
- Editar, chequear y generar informes de cuotas de disco de los usuarios.

Términos y utilidades

- quota
- edquota
- repquota
- quotaon

104.4.0. Introducción

Los sistemas operativos GNU/Linux son **multiusuario**, es decir, que pueden ser utilizados por varios usuarios de forma simultánea.

El modo en el que el sistema identifica a los distintos usuarios es mediante la asignación de cuentas de usuario.

Cada usuario se identifica por un identificador de usuario que es único en el sistema. Cada usuario puede pertenecer a uno o varios grupos.

En todo sistema GNU/Linux, deberían existir al menos dos cuentas: la del administrador (**root**) y la de un usuario sin privilegios de administración. La cuenta de **root** debe reservarse exclusivamente para las tareas de administración, mientras que para el trabajo diario debe emplearse una cuenta de usuario sin privilegios.

La existencia de múltiples usuarios condiciona el espacio de almacenamiento disponible, de tal forma que, atendiendo al identificador de cada usuario, se determinará la cantidad de espacio de almacenamiento que se tendrá disponible.

Dentro de la administración de las unidades de almacenamiento y de los sistemas de archivos, las políticas de cuotas de disco, son una necesidad en sistemas donde el número de usuarios puede ser alto, y el espacio de almacenamiento es limitado.

En GNU/Linux el único usuario que puede modificar las políticas de cuotas de disco, es el **root**.

A lo largo de este apartado, vamos a diferenciar dos funcionamientos básicos: **política de cuotas en GNU/Linux** y **gestión de las cuotas de disco**.

104.4.1. Política de cuotas en GNU/Linux.

En GNU/Linux, el sistema de cuotas provee un mecanismo para el control y uso del espacio de disco duro disponible en un sistema de ficheros.

Cada usuario tendrá su propia política de cuotas de disco y cada grupo también dispondrá de su propia política de cuotas.

La activación de la gestión de cuotas de disco, se realiza para un sistema de archivo específico, esto quiere decir, que no se implementa por defecto para todas las unidades de almacenamiento disponibles, sino que hay que activar las cuotas de forma explícita unidad por unidad.

Se pueden establecer límites en la cantidad de espacio (**limita el número bloques**) y el número de ficheros (**limita el número de inodos**) que puede disponer un usuario o grupo.

Para cada limitación (bloques e inodos) existen dos tipos de límites:

- **Límite duro (hard)**: es la cantidad máxima de espacio en disco (bloques o inodos) que un usuario o grupo puede usar. Una vez alcanzado el límite, en ningún caso podrá usar más espacio.
- **Límite suave (soft)**: también es la cantidad máxima de espacio en disco (bloques o inodos) que un usuario o grupo puede usar. Sin embargo, a diferencia del límite duro, el límite suave **puede ser excedido durante cierto tiempo**, conocido como período de gracia.
- **Periodo de Gracia**: Especifica el periodo de tiempo que un usuario puede superar el límite suave de espacio en disco, pero siempre sin superar el límite duro.

La finalidad de este periodo es dar margen de uso de espacio en disco a los usuarios, para que antes de la finalización del plazo, vuelvan a tener un uso de espacio en disco inferior al límite suave. Una vez finalizado el periodo de gracias, el límite suave tendrá los mismos efectos que el límite duro.

El periodo de gracia se puede expresar en segundos, minutos, horas, días, semanas o meses.

104.4.2. Gestión de las cuotas de disco

El procedimiento que vamos a seguir para activar las cuotas de disco para los usuarios del sistema es el siguiente:

1. Seleccionaremos el sistema de ficheros en el que se van a controlar las cuotas de disco.
2. Habilitaremos las cuotas en ese sistema de ficheros.
3. Especificaremos cuotas para un usuario que nos servirá de plantilla para establecer cuotas para el resto de usuarios del sistema.
4. Finalmente habilitaremos para todos los usuarios nuevos que se añadan al sistema, que tengan establecida automáticamente la cuota de disco que pueden usar.

Previamente debemos instalar el paquete **quota** que contiene todas las herramientas para implementar el sistema de cuotas. Adicionalmente podría instalarse el paquete **quotatools**.

```
#apt-get install quota
```

```
# yum install quota
```

104.4.2.1. Elección del sistema de ficheros sobre el que se aplican las cuotas.

Lo más usual es que sólo el sistema donde están los directorios de trabajo de los usuarios tenga cuotas, aunque es recomendable que tenga cuotas todo sistema de ficheros donde los usuarios puedan escribir.

Para comenzar, debemos decidir qué particiones dentro de */etc/fstab* se desean tener bajo límite de uso de espacio o cuota de disco.

Para habilitar las cuotas en un sistema de ficheros hay que editar el fichero */etc/fstab* e incluir la opción **usrquota** o **grpquota**, de manera que si el archivo contiene:

```
/dev/hda1 / ext3 defaults 1 2
```

```
/dev/hda5 /home ext3 defaults 1 2
```

se deben agregar las opciones anteriores:

```
/dev/hda1 / ext3 defaults,grpquota 1 2
```

```
/dev/hda5 /home ext3 defaults,usrquota 1 2
```

para especificar las particiones con límite de uso de espacio.

Después de agregar la opción, vuelva a montar cada sistema de archivos cuyas entradas en */etc/fstab* hayan sido modificadas.

Si el sistema de archivos no está siendo usado por ningún proceso, use el comando **umount** seguido de **mount** para volver a montar el sistema de archivos (o sólo **remount**). Si el sistema de archivos está siendo usado actualmente, el método más fácil para volver a montar el sistema de archivos es reiniciando el sistema.

104.4.2.2. Habilitar las cuotas de disco.

Para habilitar las cuotas de disco en el sistema de archivos cuyas entradas en */etc/fstab* hayan sido modificadas, la manera más sencilla es ejecutar el comando:

```
#quotacheck -acug
```

para crear el archivo de cuotas **aquota.user** y **aquota.group** en la raíz del sistema de archivos al que hayamos aplicado cuotas. Seguidamente ejecutaremos:

```
#quotacheck -avug
```

para generar la tabla de uso actual del sistema de archivos con cuotas activadas.

Opciones del Comando quotacheck	
Opción	Significado
-a -all	Verifica todos los sistemas de archivos con cuotas
-c -create	Especifica que se creen los archivos de cuota para cada sistema de archivos con cuotas activadas
-g -group	Verifica el uso actual de cuotas para grupos
-m -no-remount	Evita que el sistema se remonte como de solo lectura
-v -verbose	Reporta lo que hace conforme progresá, son los mensajes que salen a la terminal
-u -user	verifica por soporte de cuotas para usuarios

Por último, para iniciar o parar la gestión de cuotas en los sistemas de archivos, utilizaremos **quotaon** o **quotaoof**, de la siguiente forma:

```
# quotaon /home
```

para iniciar la gestión de cuotas en el sistema de archivos indicado

```
# quotaon -a
```

para iniciar la gestión de cuotas en todos los sistemas de archivos que las tengan habilitadas.

Esta operación no es necesaria después de haber reiniciado el sistema, ya que la habilitación e inicio de cuotas de disco, está incluida en los scripts de inicio más concretamente en */etc/rc.d/rc.local*, donde debemos tener unas líneas como estas:

```
# Actualizar y activar cuotas
```

```
if [ -x /sbin/quotacheck ] ; then
```

```

echo "Actualizando Cuotas ...."
/sbin/quotacheck -avug
echo "Terminado."
fi
if [ -x /sbin/quotaon ]; then
echo " Activando Cuotas ... "
/sbin/quotaon -avug
echo "Terminado."
fi

```

104.4.2.3. Establecer cuotas para un usuario.

Para establecer las cuotas de disco para un usuario o grupo se utiliza el comando ***edquota***.

```
# edquota antonio
```

Se abrirá entonces el editor de texto definido en la variable de entorno EDITOR o VISUAL, habitualmente el editor vi, pero si deseamos modificar el editor por defecto:

```
# export VISUAL=nano
```

y nos mostrará algo similar a:

```
Disk quotas for user antonio (uid 1000):
```

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hda5	440436	0	0	37418	0	0

- La primera columna es el nombre del sistema de archivos que tiene una cuota activada.
- La segunda columna indica los bloques, en kilobytes usados por el usuario actualmente.
- Las siguientes dos columnas son usadas para colocar límites duros y suaves para los bloques del usuario en el sistema de archivos.
- La columna inodes indica los inodes usados por el usuario actualmente.
- Las últimas dos columnas son usadas para colocar los límites duros y suaves para los inodes del usuario en el sistema de archivos.
- Si cualquiera de los valores está especificado a 0, ese límite no está configurado.

Desde el editor, podemos modificar los límites duros y suaves para bloques e inodos:

```
Disk quotas for user antonio (uid 1000):
```

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hda5	440436	1048576	1572864	37418	0	0

En los bloques se ha establecido para el usuario antonio un límite suave de 1GB y un límite duro de 1'5GB de espacio de almacenamiento en el sistema de archivos,

Para verificar que la cuota de disco de un usuario ha sido especificada, podemos utilizar el comando:

```
#quota antonio
```

El período de gracia puede cambiarse con

```
# edquota -t
```

Filesystem	Block grace period	Inode grace period
/dev/hda5	7days	7days

Si deseamos un resumen del estado de las cuotas de disco de todos los usuarios para un sistema de archivos, podemos utilizar el comando:

```
# repquota /home
```

104.4.2.4. Establecer automáticamente la cuota de disco para usuarios nuevos.

Para no tener que editar las cuotas de todos los usuarios existentes en el sistema, podemos establecer límites de uso de un usuario desde las cuotas de otro usuario mediante el comando:

```
#edquota -p antonio pepe
```

De esta forma al usuario pepe se le asignan las cuotas de disco que tiene establecidas el usuario antonio.

Si se tiene muchos usuarios ya creados en el sistema, y desea crear cuotas para ellos iguales a las del usuario antonio con un solo comando, podemos utilizar:

```
#edquota -p antonio 'awk -F: '$3 >999 {print $1}' /etc/passwd'
```

Asumiendo que los usuarios tienen un número de UID a partir del 1000, lo que por otra parte es lo más habitual.

Finalmente, para los usuarios nuevos, modificaremos el fichero de configuración del programa **adduser**, de forma que todos los usuarios creados de ahora en adelante tengan asignadas las mismas cuotas de disco que tiene establecidas el usuario antonio.

```
# nano /etc/adduser.conf
```

```
QUOTAUSER="antonio"
```

104.5. Controlar permisos y propiedades de archivos.

Peso en el examen de certificación: 3 puntos.

Objetivo: Controlar el acceso a los ficheros y directorios mediante el uso adecuado de permisos y propietarios.

Conceptos y áreas de conocimiento:

- Mantenimiento de los permisos de acceso a ficheros regulares y especiales así como a directorios.
- Utilización de modos de acceso como "suid", "sgid" y "sticky bit" para mantener la seguridad.
- Saber como cambiar la mascara de permisos para la creación de ficheros.
- Utilizar el grupo para conceder permisos de acceso a grupos de usuarios.

Términos y utilidades:

- chmod
- umask
- chown
- chgrp

104.5.0. Introducción

Los sistemas operativos GNU/Linux son **multiusuario**, es decir, que pueden ser utilizados por varios usuarios de forma simultánea.

El modo en el que el sistema identifica a los distintos usuarios es mediante la asignación de cuentas de usuario.

Cada usuario se identifica por un nombre de usuario que es único en el sistema. Cada usuario puede pertenecer a uno o varios grupos.

En todo sistema GNU/Linux, deberían existir al menos dos cuentas: la del administrador (**root**) y la de un usuario normal. La cuenta de root debe reservarse exclusivamente para las tareas de administración, mientras que para el trabajo diario debe emplearse una cuenta de usuario normal.

La identidad del usuario y los grupos a los que pertenece determinan los permisos de acceso a los ficheros y otros recursos del sistema.

La administración de las políticas de seguridad, es una tarea fundamental para el administrador de un sistema.

En GNU/Linux la único usuario que puede cambiar los permisos y propiedades de un archivo o directorio es su propietario. El usuario **root** podrá cambiar los permisos de cualquier objeto de otro usuario del sistema.

A lo largo de este apartado, vamos a diferenciar dos funcionamientos básicos: **política de permisos en GNU/Linux y gestión de los permisos**.

104.5.1. Política de permisos en GNU/Linux.

En GNU/Linux, cada fichero o directorio tiene un propietario identificado por su UID (Identificador

de usuario), y cada usuario pertenece al menos a un grupo identificado por su GID (Identificador de grupo).

Basado en esta estructura, el sistema crea **permisos de acceso a tres niveles**:

- Permisos para el propietario del objeto
- Permisos para el grupo al que pertenece el usuario
- Permisos para el resto de los usuarios del sistema

A su vez los **permisos básicos** son tres:

- R --> Permiso de lectura (Read)
- W --> Permiso de escritura (Write)
- X --> Permiso de ejecución (eXecution)

Los permisos posibilitan diferentes acciones si **se aplican a un fichero**,

- R --> El usuario puede leer el contenido del fichero
- W --> El usuario puede modificar el contenido del fichero
- X --> El usuario puede ejecutar el contenido del fichero

o **se aplican a un directorio**:

- R --> El usuario puede leer el contenido del directorio, es decir los números de inodo y los nombres de los ficheros (podríamos ejecutar **ls**).
- W --> El usuario puede modificar el contenido del directorio, es decir, crear, renombrar y eliminar ficheros y subdirectorios.
- X --> El usuario puede recorrer el directorio y acceder a los objetos que contiene (podría ejecutar **cd** pero no **ls**).

104.5.2. Interpretando los permisos y propiedades de los archivos.

Podemos hacernos una primera idea de los permisos y propiedades de los archivos en GNU/Linux, escribiendo en nuestro intérprete de comandos **\$ls -l**, y obteniendo la siguiente salida:

```
$ ls -l
```

```
drwxr-xr-x 2 antonio antonio 4096 2012-01-29 20:11 Descargas
drwxr-xr-x 2 antonio antonio 4096 2012-01-29 20:11 Documentos
drwxr-xr-x 3 antonio antonio 4096 2012-02-12 21:48 Escritorio
-rw-r--r-- 1 antonio antonio 179 2012-01-29 20:00 examples.desktop
drwxr-xr-x 2 antonio antonio 4096 2012-01-29 20:11 Imágenes
drwxr-xr-x 2 antonio antonio 4096 2012-01-29 20:11 Música
drwxr-xr-x 2 antonio antonio 4096 2012-01-29 20:11 Plantillas
drwxr-xr-x 2 antonio antonio 4096 2012-01-29 20:11 Público
drwxr-xr-x 2 antonio antonio 4096 2012-01-29 20:11 Vídeos
```

Cada línea identifica un objeto contenido en la ruta actual, y está dividida en varios campos:

{Tipo} {Permisos Propietario} {Permisos Grupo} {Permisos Otros} {Número} {Propietario}
{Grupo} {Tamaño} {Fecha} {Hora} {Nombre}

Vamos a analizar la información que nos muestra el comando empezando por la izquierda.

- **{Tipo}** --> (1 carácter) Nos indica que tipo de objeto es:

-	El guión nos indica que el objeto es un fichero
d	Directorio
l	Enlace Simbólico
b	Fichero que representa a un dispositivo de tipo bloque
c	Fichero que representa a un dispositivo de tipo carácter

- **{Permisos Propietario}** --> (3 caracteres)(rwx) Nos indica los permisos que tiene sobre el objeto el usuario propietario del mismo.
- **{Permisos Grupo}** --> (3 caracteres)(rwx) Nos indica los permisos que tiene sobre el objeto los usuarios que pertenecen al mismo grupo del propietario.
- **{Permisos Otros}** --> (3 caracteres)(rwx) Nos indica los permisos que tiene sobre el objeto el resto de usuarios del sistema.
- **{Número}** --> (1 carácter) Nos indica el número de objetos que contiene. Si es un fichero aparecerá 1, si se trata de un directorio aparecerá como mínimo 2 (los directorios . y ..).
- **{Propietario}** --> Nos indica el nombre del usuario al que pertenece el objeto.
- **{Grupo}** --> Nos indica el nombre del grupo del usuario propietario del objeto.
- **{Tamaño}** --> Nos indica el tamaño del objeto en bytes.

En ficheros de dispositivos nos mostrará 2 números, el primero identifica al tipo de dispositivo y el segundo identifica un dispositivo concreto dentro de un tipo.

```
$ ls -l /dev
```

```
lrwxrwxrwx 1 root root 8 2012-01-29 20:12 cdrom -> /dev/hdc
crw-rw-rw- 1 root root 1, 7 2012-01-29 20:11 full
brw-rw---- 1 root disk 3, 0 2012-01-29 20:11 hda
brw-rw---- 1 root disk 3, 1 2012-01-29 20:11 hda1
brw-rw---- 1 root disk 3, 2 2012-01-29 20:11 hda2
brw-rw---- 1 root cdrom 22, 0 2012-01-29 20:11 hdc
```

- **{Fecha}** --> Nos indica la fecha de creación del objeto.
- **{Hora}** --> Nos indica la hora de creación del objeto.
- **{Nombre}** --> Nos indica el nombre del objeto.

En los enlaces simbólicos nos muestra donde apunta, es decir, cuál es el objeto real del enlace (enlace --> objeto real).

104.5.3. Modificar los permisos de los archivos.

Lo más habitual es utilizar el comando **chmod** (change mode) para modificar los permisos de los archivos. Su sintaxis básica es:

```
chmod [opción] permisos fichero
```

Las opciones modifican el comportamiento del comando, como **-R**, para cambiar los permisos de un modo recursivo dentro de los directorios.

Desde una **notación simbólica** los permisos los podemos dividir en 3 campos:

usuario	operador	permisos
		r --> Lectura
u --> propietario (user)	+ --> añade permisos a los ya existentes	w --> Escritura
g --> grupo (group)	- --> quita permisos de los que ya había	x --> Ejecución
o --> resto usuarios (other)	= --> establece los permisos, borrando los que tuviese anteriormente.	s --> Set uid o gid
a --> todos (all)		t --> Sticky bit bit persistente

La manera de aplicar estos permisos será:

{Usuario a quién se aplica}{Modo de aplicar}{Permisos que aplica}

Ejemplos:

a+r --> Permisos de lectura para todos.

+r --> Si no se indica el usuario, se supone **all**, luego sería idéntica a la anterior.

og-x --> Quita permiso de ejecución a todos menos al propietario.

u+wx --> Da todos los permisos al propietario.

o-rwx --> Quita todos los permisos al resto de usuarios.

Ejemplos de ejecución:

#chmod o=r fichero

Establece los permisos del fichero para el resto de usuarios a sólo lectura.

#chmod u+x fichero

Añadimos el permiso de ejecución para el propietario del fichero.

#chmod -R a=rwx /home/compartido

Establece los permisos de lectura, escritura y ejecución al directorio *compartido* y todo su contenido para el propietario, el grupo y el resto de usuarios.

104.5.4. Notación Octal de los permisos en GNU/Linux.

Para entender cómo gestionar los permisos con una notación octal, nos interesa recordar cómo es el sistema binario.

En un sistema **BINARIO** sólo pueden haber dos valores para cada dígito: **[0-1]**.

En todos los sistemas de numeración, incluido el binario, el valor de cada número depende del valor de cada dígito, de la posición que ocupan y de la base en la que estamos trabajando.

$$\text{Valor Decimal} = \sum_{n=0} (\text{Dígito}_n * \text{Base}^{\text{Posición}})$$

El número $00010111_{(2)}$ (base 2 --> binario) equivale a $23_{(10)}$ (base 10 --> decimal), equivalencia que se explica según la siguiente tabla:

Posición del BIT:	7	6	5	4	3	2	1	0
Número Binario	0	0	0	1	0	1	1	1
Base ^{Posición}	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Dígito _n * Base ^{Posición}	$0*2^7$	$0*2^6$	$0*2^5$	$1*2^4$	$0*2^3$	$1*2^2$	$1*2^1$	$1*2^0$
Valores a Sumar	0	0	0	16	0	4	2	1

$$\text{Valor Decimal} = 16 + 4 + 2 + 1 = 23$$

Siguiendo con esta transformación de sistemas de numeración, nos encontramos con que cada **dígito octal** se corresponde con **tres dígitos binarios**.

De esta forma, podríamos utilizar 3 dígitos en binario, el primero para la lectura, el segundo para la escritura y el tercero para la ejecución. Si el dígito está a 1 entonces el permiso está habilitado, si está a 0 entonces el permiso está deshabilitado.

Permisos			Binario			Octal
-	-	-	0	0	0	0
-	-	x	0	0	1	1
-	w	-	0	1	0	2
-	w	x	0	1	1	3
r	-	-	1	0	0	4
r	-	x	1	0	1	5
r	w	-	1	1	0	6
r	w	x	1	1	1	7

Finalmente, para asignar los permisos a los 3 niveles de usuarios utilizaremos 3 dígitos en octal, el primer dígito octal corresponde a los permisos para el **propietario**, el segundo para el **grupo** y el tercero para el **resto de los usuarios**.

Ejemplo:

```
$chmod 754 fichero.txt
```

Establece al fichero los permisos `rwxr-xr--`, donde 7 corresponde a `rwx`, 5 a `r-x`, y 4 a `r--`.

104.5.5. Máscara de Permisos.

En GNU/Linux cuando un usuario crea un fichero o directorio, se le asigna unos permisos por defecto. Usualmente, para un fichero nuevo se le asignan los permisos `rw-r--r--` (644), mientras que para un directorio nuevo `rwxr-xr-x` (755).

Para conseguir esta asignación por defecto, se utiliza una máscara de permisos, cuya misión no es asignar los permisos, sino restringirlos. Esto quiere decir que la máscara especifica los permisos que no se asignarán a los nuevos objetos que se vayan creando.

El valor de la máscara se puede consultar y modificar mediante el comando **umask**. El valor por defecto de la máscara suele ser **0022**.

El procedimiento de aplicación de permisos para nuevos objetos es el siguiente:

1. Por defecto, se crean todos los ficheros con los permisos 666 (`rw-rw-rw-`)
2. Por defecto, se crean todos los directorios con los permisos 777 (`rxrwxrwx`)
3. Se aplica la máscara (por defecto 0022)
4. Se utiliza la misma máscara para todos los ficheros y directorios.
5. La máscara no modifica los permisos de los objetos existentes, sino que se aplica solamente a los nuevos objetos.

Para consultar el valor de la máscara hay que utilizar el comando **umask** sin parámetros:

```
$umask
```

0022

Para asignar un nuevo valor de máscara, indicamos el nuevo valor como parámetro:

```
$umask 0026
```

```
$umask
```

0026

Ejemplos de aplicación de la máscara:

- Para un fichero

Predeterminado `rw-rw-rw-` (666)

Retirar `----w--w-` (022)

Resultado `rw-r--r--` (644)

- Para un directorio

Predeterminado `rxrwxrwx` (777)

Retirar `----w--w-` (022)

Resultado `rxr-xr-x` (755)

- Suprime permisos, no realiza la resta aritmética

Predeterminado `rw-rw-rw-` (666)

Retirar `----wxrwx` (037)

Resultado `rw-r-----` (640) (y no 629, imposible en octal)

104.5.6. Cambiar el propietario y el grupo.

Como hemos referido al principio de este apartado, en GNU/Linux la única persona que puede cambiar los permisos y propiedades de un archivo o directorio es su propietario o el **root**, por lo que es posible que en el transcurso de las operaciones realizadas en el sistema, necesitemos modificar el propietario o grupo de un fichero, para que este pueda ser utilizado por otros usuarios.

Para cambiar el propietario de un fichero, utilizamos el comando **chown** (change owner), siempre y cuando o bien seamos **root** o bien seamos el propietario del fichero. Para cambiar el grupo del fichero, utilizamos el comando **chgrp** (change group).

La sintaxis de los comandos:

chown usuario fichero --> Cambia únicamente el usuario

chown usuario:grupo fichero --> Cambia el usuario y el grupo

chown -R usuario:grupo directorio --> Cambia el usuario y el grupo de un directorio y de todos los objetos que contenga.

chgrp grupo fichero --> Cambia únicamente el grupo

Ejemplo:

```
#ls -l  
-rw-r--r-- 1 root root 0 2012-02-28 01:19 ejemplo.txt
```

```
#chown antonio ejemplo.txt  
#ls -l  
-rw-r--r-- 1 antonio root 0 2012-02-28 01:19 ejemplo.txt
```

```
#chgrp antonio ejemplo.txt  
#ls -l  
-rw-r--r-- 1 antonio antonio 0 2012-02-28 01:19 ejemplo.txt
```

El fichero *ejemplo.txt* pertenece al usuario **root**, cambia el propietario del fichero al usuario *antonio*, y a continuación cambia el grupo del fichero al grupo *antonio*.

104.5.7. Permisos de acceso especiales.

GNU/Linux dispone de unos permisos especiales que se utilizan para solucionar situaciones específicas sin alterar las políticas de seguridad. Estos permisos son:

- Sticky bit (Bit pegajoso, o bit persistente)
- SUID (Establecer el ID del usuario)
- SGID (Establecer el ID del grupo)

BIT PERSISTENTE

El sticky bit tiene un sentido diferente según se aplique a un fichero o a un directorio.

Este permiso se aplica a un fichero cuando este se utiliza con bastante frecuencia, y debería ser retenido en el área de swap aun cuando no se esté ejecutando en ese momento. Esto consume memoria swap pero reduce notablemente el tiempo de ejecución.

Si este permiso se aplica a un directorio, entonces los usuarios no podrán borrar ficheros en este directorio, salvo los ficheros de los que sean propietarios.

Este permiso puede ser útil aplicado a un directorio compartido, en el que queremos que todos los usuarios puedan leer y escribir, pero no puedan modificar o borrar nada más que los ficheros que cada usuario ha creado y no los de los otros usuarios.

Este permiso aplicado se mostraría con el comando **ls -l** con una **t** en lugar que correspondería a los permisos de ejecución para el resto de usuarios del sistema.

```
#ls -l  
drwxrwxrwt 2 root root 4096 2012-01-29 20:11 Compartida
```

El sticky bit corresponde al valor octal 1000 así que si queremos asignarlo:

```
#chmod 1000 Compartida
```

Esto añadiría el bit persistente a *Compartida* pero eliminaría los demás permisos. Si queremos añadirlo sin eliminar los permisos anteriores y suponiendo que tuviera permisos 777:

```
#chmod 1777 Compartida
```

SUID - ESTABLECER EL ID DEL USUARIO

El SUID le dice al kernel que el usuario que ejecuta el fichero que tiene ese permiso aplicado, adquiera durante la ejecución la identidad del propietario del fichero.

De este modo cuando un usuario ejecuta un fichero, durante el tiempo que dura la ejecución, adquiere la personalidad del propietario y operar como tal.

Pongamos un ejemplo. Es deseable que cualquier usuario del sistema pueda cambiar su contraseña sin tener que recurrir al root. El problema es que cuando se cambia la contraseña las modificaciones se guardan en un par de ficheros */etc/password* y */etc/shadow*.

Estos ficheros son propiedad del root y sólo él puede escribir en ellos.

Una solución sería permitir que todo el mundo pueda escribir en estos ficheros. La solución es malísima porque de este modo no sólo cada usuario podría cambiar su propia contraseña, sino también la de cualquier otro usuario.

La solución pasa por activar el bit **suid** al fichero ejecutable */usr/bin/passwd* que es propiedad del **root**. Con este comando, cambiamos las contraseñas escribiéndolas en */etc/password*.

De este modo, cuando un usuario normal ejecuta */usr/bin/passwd*, durante el tiempo que dura la ejecución del programa, adquiere la personalidad del **root** y por tanto puede escribir en los ficheros ya mencionados.

Este permiso aplicado se mostraría con el comando **ls -l** con una **s** en lugar que correspondería a los permisos de ejecución para el propietario del fichero.

```
#ls -l /usr/bin/passwd  
-rwsr-xr-x 1 root root 26616 2012-01-29 20:11 /usr/bin/passwd
```

El sticky bit corresponde al valor octal 4000 así que si queremos asignarlo sin eliminar los permisos anteriores:

```
#chmod 4755 /usr/bin/passwd
```

SGID - ESTABLECER EL ID DEL GRUPO

El SGID tiene una aplicación muy parecida a la del SUID, pero referido al grupo del propietario del

fichero

Este permiso aplicado se mostraría con el comando **ls -l** con una **s** en lugar que correspondería a los permisos de ejecución para el grupo del propietario del fichero.

```
#ls -l /usr/bin/passwd  
-rwxr-sr-x 1 root root 26616 2012-01-29 20:11 /usr/bin/passwd
```

El sticky bit corresponde al valor octal 2000 así que si queremos asignarlo sin eliminar los permisos anteriores:

```
#chmod 2755 /usr/bin/passwd
```

Establecer el SUID o el SGID tiene sentido únicamente si se han establecido los permisos de ejecución previamente (permiso **x** en el propietario o el grupo). Si no se ha establecido el permiso de ejecución, se sustituye la **s** por **S**.

104.5 EXTRAS

1045 EXTRAS Atributos de los sistemas de ficheros ext

Conjuntamente con los permisos, los atributos de ficheros en Unix es una opción muy válida para restringir aún más, las operaciones a realizar con nuestros ficheros y directorios.

En sistemas **Linux**, se usa el comando **chattr** para cambiar estos atributos. Una vez “dominado” es uno de los comandos mas útiles en **linux** para salvaguardar la integridad de muchos de sus ficheros importantes conjuntamente con los permisos.

Es un comando poco conocido por muchos usuarios e incluso administradores de sistemas. Con chattr, y en sistemas de ficheros ext2 o posteriores, es posible asignarle atributos a los ficheros y directorios que residan en los mismos. El uso del comando está restringido naturalmente a root y en algunos casos al propietario del fichero (consultar la página man)

La tabla con los atributos y sus significados, la podemos ver aquí:

Atributo	Significado	Ejemplo de uso
A	El valor de la fecha de acceso no será cambiado en cada lectura del fichero.	Puede incrementar los tiempos de lectura al ahorrarse la actualización de este dato.
s	El espacio que ocupaba el fichero, será llenado por bloques de ceros cuando el fichero sea eliminado.	Muy util para realizar un borrado pseudoseguro de forma rápida. No obstante, es recomendable el uso de srm.
a	El fichero únicamente podrá ser abierto para añadir datos al mismo.	Este atributo está pensado principalmente para usarlo con logs. Podemos modificarlos para añadir líneas pero no modificar más.
c	Activa la compresión de los datos del fichero.	En kernels con soporte de compresión, se comprime el espacio del fichero en disco de forma transparente para las aplicaciones.

D	Este atributo hace que los datos escritos en un directorio, se sincronicen en el disco de forma automática.	Es muy util en el caso de discos en memoria RAM o bien en aquellos que la escritura sea en formato raw. No es muy seguro para equipos de uso habitual.
d	Elimina el fichero o directorio de las copias de seguridad realizadas con la utilidad dump.	Util para directorios como /tmp o aquellos de los que no queramos hacer backup o se hagan con otras herramientas.
I	Suele venir por defecto en ext4 y ext3. Está relacionado con la utilización de la indexación vía htree de estos sistemas.	Desactivando este atributo, podemos ahorrar tiempo de acceso en el caso de estar utilizando otros sistemas de indexación, aunque rara vez se dará el caso.
i	Pone el fichero en modo solo lectura y no es posible crear enlaces hacia el.	Interesante atributo para activarlo en ficheros que rara vez son escritos. Binarios, ficheros de un servidor web, repositorios de consulta, o incluso ficheros de BBDD que no son accedidos vía web para su modificación.
j	En sistemas con ext3 o superior, es posible realizar el “journaling” de los ficheros con este atributo en el caso de que la partición no sea montada con tal opción.	Puede ahorrar tiempos de acceso a disco montar un sistema de estas características y activar el journaling solo para determinados ficheros. No es muy recomendable.
S	Este atributo tiene el mismo significado para los ficheros, que el D para los directorios.	Es muy util en el caso de discos en memoria RAM o bien en aquellos que la escritura sea en formato raw. No es muy seguro para equipos de uso habitual.
T	Activa el denominado Orlov block allocator en un directorio. Esto es, que el directorio con este atributo, se escribirá en las partes mas “rápidas” del disco.	Este atributo es muy util para utilizarlo con algunos directorios con gran número de accesos como un directorio de un servidor web, el servidor de ficheros, etc.
t	Los ficheros con este atributo, no presentan fragmentación en el sistema de ficheros.	Realmente, no tiene mucho uso salvo para antiguos sistemas. Actualmente, los sistemas modernos, cuentan con FS resistentes a la fragmentación.

La forma de asignar un atributo es con el signo “+” y retirarlo es con el signo “-”. Para listar los atributos de los ficheros, se puede utilizar el comando **lsattr**.

Un ejemplo:

```
-bash-3.2# touch prueba
-bash-3.2# chattr +i prueba
-bash-3.2# rm prueba
rm: remove write-protected regular empty file `prueba'? y
rm: cannot remove `prueba': Operation not permitted
-bash-3.2#id uid=0(root) gid=0(root) groups=0(root)....
```

104.6. Crear y modificar enlaces simbólicos y duros.

Peso en el examen de certificación: 2 puntos.

Objetivo: **Crear y mantener enlaces duros y simbólicos a ficheros.**

Conceptos y áreas de conocimiento:

- Crear enlaces.
- Identificar enlaces simbólicos y/o duros.
- Copiar en contraposición a enlazar ficheros.
- Utilizar enlaces para dar soporte a tareas administrativas en el sistema.

Términos y utilidades

- In

104.6.0. Introducción

En este apartado vamos a ver la posibilidad de crear enlaces a ficheros, distinguiendo entre enlaces duros (hard links) y enlaces simbólicos (symbolic links o symlinks), también conocidos como enlaces blandos.

Recordemos que en GNU/Linux los ficheros son identificados por el sistema a través del número de inodo. Por otra parte, un directorio, que contiene ficheros, es una simple lista de números de inodo con sus correspondientes nombres de fichero. Cada nombre de fichero, por lo tanto, enlaza a un inodo particular.

El comando utilizado para la creación de enlaces es: **In**

Existen muchos ámbitos en los que puede ser interesante la utilización de enlaces, ya que nos permiten disponer de un mismo fichero en distintas ubicaciones, para acceder, por ejemplo, con diferentes permisos dependiendo precisamente de esta ubicación.

104.6.1. Enlaces duros.

Los enlaces duros comparten el mismo número de inodo, es decir, son referencias a un mismo fichero. Por ello no es posible distinguir un enlace de su fichero original. Lógicamente, los cambios que realicemos sobre cualquiera de los enlaces, será reflejado en el resto, debido a que es el mismo fichero.

Si borramos algún enlace esto no afecta al fichero mientras existan enlaces que apunten a su inodo. Un fichero es eliminado cuando borramos el último enlace a su inodo.

Los enlaces duros tienen la limitación que no pueden ser utilizados para directorios y que sólo pueden ser creados en el mismo sistema de ficheros donde se encuentre el fichero original.

El formato de la instrucción es el siguiente:

In [fichero] [enlace]

Comprendamos mejor su funcionamiento: Para ello vamos a crear un fichero de texto con el nombre "original". Mediante el comando **\$ ls -i** veremos el número de inodo asociado a este fichero.

A continuación creamos un enlace duro a este fichero, utilizando el comando:

\$ ln original enlaceduro

Si nuevamente efectuamos un listado **\$ ls -i** podremos comprobar que los dos archivos poseen el mismo inodo. Si efectuamos un listado en formato largo (**\$ ls -l**) podremos observar que, tras los permisos del fichero, hay un número que es el que determina el número de enlaces duros asociados al inodo del fichero, en este caso 2. Este contador, cuando llega cero, es cuando se elimina definitivamente tanto el inodo como el contenido del fichero.

Si deseamos conocer los enlaces duros de un inodo podemos utilizar la instrucción **find**, ejecutada desde la raíz del sistema de ficheros con la opción **-inum**. Por ejemplo, si el inodo es el 145394:

```
$ find -inum 145394
```

104.6.2. Enlaces simbólicos.

El enlace simbólico no enlaza con el inodo del fichero origen, sino que es un acceso directo al fichero origen. Si estamos acostumbrados a trabajar con los accesos directos del sistema Windows, su funcionamiento es similar.

En este caso el inodo del fichero origen y del enlace simbólico son diferentes. Su tamaño también, ya que el enlace simbólico únicamente contiene la ruta para poder acceder al fichero origen.

Los enlaces simbólicos sí que pueden ser creados para directorios y pueden extenderse a otros sistemas de ficheros.

Si borramos el fichero origen el enlace simbólico permanece, pero sin ninguna utilidad, ya que el fichero al que apuntaba ha desaparecido. Si utilizamos el enlace simbólico realmente accederemos al fichero origen, por lo que las modificaciones que realicemos afectarán a su contenido. Eliminar el enlace no afecta al fichero origen.

El formato de la instrucción es el siguiente:

ln -s [fichero] [enlace]

Es recomendable ejecutar siempre la instrucción en el directorio donde queremos crear el enlace, de esta forma el enlace forzosamente incluirá la ruta para poder acceder al fichero.

Utilizando los mismos ficheros que en el apartado anterior, construyamos un enlace simbólico sobre el fichero "original":

\$ ln -s original simbolico

Si ejecutamos el comando **\$ ls -i** podremos observar que el número de inodo es distinto para cada fichero. Por otra parte, en un listado largo, comprobaremos que el tamaño de ambos ficheros es distinto. Además el enlace simbólico se indica de la siguiente forma:

simbolico -> original

Si eliminas el fichero original puedes verificar que el enlace simbolico no tiene utilidad, en cambio, el enlace duro construido anteriormente sigue operativo.

104.7. Encontrar archivos de sistema y conocer su localización correcta.

Peso en el examen de certificación: 2 puntos.

Objetivo: Estar totalmente familiarizado con el estandar de jerarquía del sistema de ficheros (FHS), incluyendo las ubicaciones de archivos típicos y clasificaciones del directorio.

Conceptos y áreas de conocimiento:

- Entender la correcta localización de ficheros bajo FHS (*Filesystem Hierarchy Standard - Estandar de la Jerarquía del Sistema de Ficheros*).
- Localizar ficheros y comandos en un sistema GNU/Linux.
- Conocer la localización y el propósito de los ficheros y directorios importantes tal y como se definen en FHS.

Términos y utilidades:

- find
- locate
- updatedb
- whereis
- which
- type
- /etc/updatedb.conf

104.7.0. Introducción

En este apartado se van a tratar los comandos básicos para poder encontrar archivos de sistema en los sistemas GNU/Linux.

FHS (Filesystem Hierarchy standard)ofrece un excelente recurso que asegura la coherencia entre las distintas distribuciones y otros sistemas operativos. Sin embargo, en la práctica, la localización exacta de un archivo puede ser frustrante, y surge la necesidad de buscar archivos en el sistema rápidamente.

Linux dispone de una serie de herramientas de localización de archivos, a saber, find, locate, whereis y which.

104.7.1. find.

El comando **find** se usa para encontrar archivos en el árbol de directorios de Linux. **find** requiere un punto de partida y las características del archivo a encontrar. Comienza buscando por un directorio especificado y continúa por todos los subdirectorios que están bajo el mismo, en busca de nombres de archivos que coincidan con el patrón especificado. Cuando no se especifica ningún directorio, comienza por el directorio actual (pwd) y continúa por todos los subdirectorios dentro de éste. Las búsquedas por medio de la utilidad **find** pueden ser lentas y utilizar muchos recursos del sistema, dependiendo de la búsqueda requerida.

La sintaxis del comando es la siguiente:

find /directorio expresión

Tabla 4.7-1 Opciones del comando **find**

Opción	Uso
-atime	Búsqueda basada en el número de días desde el último acceso.
-ctime	Búsqueda basada en el número de días desde el último cambio en la entrada del directorio.
-group	Búsqueda de archivos pertenecientes al grupo especificado.
-newer	Búsqueda de archivos más recientes que el archivo especificado.
-name	Búsqueda de archivos cuyo nombre coincide con la cadena especificada.
-user	Búsqueda de archivos pertenecientes al usuario especificado.

Algunos ejemplos serían:

1. Busca en el directorio actual todos los archivos o directorios de nombre a buscar y muestra el resultado en pantalla.

```
# find . -name abuscar -print
```

2. Busca, a partir del directorio /usr/usuarios, todos los archivos que terminen en .f. El uso de las comillas es indispensable porque de lo contrario, el shell sustituye por el asterisco los nombres de todos los archivos en el directorio de partida.

```
# find /usr/usuarios -name '* .f' -print
```

3. Buscar a partir del directorio raíz todos los ficheros regulares cuyo nombre termine en .lo

```
# find / -type f -name "* .log"
```

4. Buscar a partir del directorio raíz todos los ficheros terminados en .tmp y eliminarlos

```
# find / -type f -name "* .tmp" -delete
```

La opción -delete sirve para eliminar los ficheros que **find** encuentra a partir de los parámetros que le damos (cuidado, cualquier error puede ser letal).

5. Encontrar en el directorio actual (.) los ficheros regulares, y copiarlos a /tmp

```
# find . -type f -exec cp '{}' /tmp \;
```

6. Encontrar en el directorio /usr/src los archivos cuyo nombre terminen en .c (código fuente) y cuyo tamaño sea mayor a 100k, e imprimirlas (en pantalla).

```
# find /usr/src -name '* .c' -size +100k -print
```

104.7.2. **locate**.

Locate es un comando de búsqueda de archivos, bastante parecido al comando **find**. La diferencia de **locate** es que la búsqueda la hace en una base de datos indexada para aumentar significativamente la velocidad de respuesta. **Locate** proporciona un método seguro para indexar y

buscar rápidamente archivos en el sistema. Utiliza codificación incremental para comprimir su base de datos y hacer las búsquedas más veloces, pero también almacena los permisos y propietario del archivo, de modo que los usuarios que no puedan ver esos archivos, no podrán acceder a ellos. Este comando utiliza por defecto la base de datos **slocate** para encontrar los archivos. Es posible especificar otra base de datos para ser usada.

La sintaxis del comando es la siguiente:

locate -opciones argumentos

La base de datos se actualiza por medio del comando **updatedb**, sin argumentos. Sólo el usuario root está autorizado a utilizar este comando en el sistema.

Tabla 4.7-2 Opciones del comando **locate**

Opción	Uso
-u	Comienza por el directorio raíz / cuando se crea la base de datos slocate.
-U /PATH	Comienza por el directorio especificado cuando se crea la base de datos slocate.
-e DIR	Excluye los directorios especificados cuando se crea la base de datos slocate.
-c	Analiza el archivo /etc/updatedb.conf cuando se crea la base de datos slocate.
-i	Busca sin tener en cuenta mayúsculas y minúsculas.
-o FILE	Especifica el archivo de salida a crear.
-d PATH	Especifica la ruta a las bases de datos a buscar.

Algunos ejemplos serían:

1. Listar todos aquellos archivos que posean la palabra «**locate**» en su nombre.

```
#locate locate
```

2. Buscar los archivos comprimidos con gzip que además estén debajo de /home.

```
# locate *.gz | grep ^/home
```

En este ejemplo se pone de manifiesto que es posible combinar los comandos de búsqueda con la utilidad **grep**.

104.7.3.- whereis.

El comando **whereis** busca un archivo específico dentro de archivos fuente, binarios y páginas del manual. Los nombres especificados, antes de ser buscados, se separan de la definición de la ruta y de las extensiones del tipo .ext (por ejemplo .c). Así pues, la utilidad **whereis** trata de encontrar los programas deseados dentro de las localizaciones de código.

Tabla 4.7-3 Opciones del comando **whereis**

Opción	Uso
--------	-----

- b Búsqueda de binarios.
- m Búsqueda de entradas manuales.
- s Búsqueda de fuentes.
- u Búsqueda de entradas inusuales que no tienen una entrada por cada tipo.

Algunos ejemplos serían:

1. Localizar información sobre el comando **ls**.

```
# whereis ls
```

2. Pedir la página del manual de este mismo comando.

```
#whereis man
```

3. Pedir todos los archivos que tengan que ver con el comando **passwd**.

```
#whereis passwd
```

104.7.4.- which.

El comando which para cada uno de sus argumentos, muestra por la salida estándar (stdout) la ruta completa a los ejecutables que se pueden ejecutar cuando dicho argumento se escribe en la línea de comandos. Ésto lo hace buscando ficheros ejecutables o scripts en los directorios especificados en la variable de entorno PATH.

Esta utilidad nos permite ver la ruta completa a un comando antes de ejecutarlo. Puede ser útil para comprobar que estamos haciendo uso del comando que queremos ejecutar.

La utilidad utiliza la siguiente sintaxis:

which -opciones nombre_de_programa

Algunos ejemplos serían:

1. Averiguar dónde se encuentra instalado el programa **find**.

```
#which find
```

Nos devolverá:

/usr/bin/find

2. Averiguar dónde se encuentra instalado el programa **find** y todas las ocurrencias que encuentre.

```
#which -a find
```

Nos devolverá todas las ocurrencias que encuentre:

/usr/bin/find

/usr/bin/X11/find

105 SHELLS, SCRIPTS Y ADMINISTRACIÓN DE DATOS.

- 105.1. Personalizar y trabajar en el entorno shell.
- 105.2. Editar y escribir scripts simples.
- 105.3. Administración de datos SQL.

105.1. Personalizar y trabajar en el entorno shell.

Peso en el examen de certificación: 4 puntos.

Objetivo: Personalizar el entorno de ejecución de la shell para satisfacer las necesidades de los usuarios. Modificar las especificaciones de los perfiles de usuarios, tanto a nivel global como a nivel usuario.

Conceptos y áreas de conocimiento:

- Inicializar variables de entorno (p.e. PATH) al iniciar la sesión o cuando se crea una nueva shell
- Escribir funciones BASH que incorporen secuencias de comandos de uso frecuente.
- Mantenimiento de directorios esqueleto para los nuevos que se inserten en el sistema.
- Establecer ruta de búsqueda de comandos con el directorio adecuado.

Términos y utilidades

- /etc/profile
- env
- export
- set
- unset
- ~/.bash_profile
- ~/.bash_login
- ~/.profile
- ~/.bashrc
- ~/.bash_logout
- function
- alias
- lists

105.1.1. Introducción

El shell espera entradas por el teclado (comandos) en una línea llamada **Línea de comandos** o **prompt**. El prompt proporciona información en el terminal y su posición en el sistema de ficheros.

Ejemplo: juan@lpi:/home/public>

En esta línea, obtiene cuatro datos:

- juan: es el nombre de conexión o login del usuario actualmente conectado.
- lpi: es el nombre del hostname, el nombre lógico de la máquina.
- /home/public: es la posición actual del shell en el sistema de ficheros.
- >: es la terminación estándar del bash para un usuario sin privilegios.

El carácter de terminación puede tener otros significados:

- \$ indica que el usuario no tiene privilegios particulares, como con >.
- # indica que el usuario es el administrador root que tiene todos los privilegios.

Los comandos tienen la siguiente estructura:

Comando [parámetros] [argumentos] donde un parámetro es una opción del comando.

Existen dos tipos de comandos:

1. **Los comandos externos** son programas binarios presentes como ficheros en su disco duro.
Cuando ejecuta el comando, se carga este fichero en memoria y se inicia como proceso.
2. **Los comandos internos** son propios del shell. Estos comandos forman parte del programa shell, del bash.

Se pueden ejecutar varios comandos en una sola línea, basta con separarlos con el carácter punto y coma (;

Ejemplo: **\$pwd;date;ls**

Además, se pueden ejecutar los comandos de manera condicional. La condición de ejecución de un comando es el éxito o no del comando anterior. Una vez ejecutado, cada comando devuelve un código de retorno, 0 si todo ha salido bien, 1 o 2 en caso de error. El shell puede recuperar este valor con la variable **\$?**.

Ejemplo: **\$ ls -l**

\$echo \$?

Los caracteres **&&** y **||** permiten efectuar una ejecución condicional.

comando1 **&&** comando2 --> Se ejecutará el comando2 colocado después de **&&** únicamente si el comando1 anterior ha devuelto 0 (éxito).

comando1 **||** comando2--> Sólo se ejecutará el comando2 colocado después de **||** si el comando1 anterior ha devuelto algo diferente a 0.

Ejemplo: **\$ grep "juan" lista && echo "Alumno matriculado" || echo "Alumno no matriculado"**

Si juan está en el contenido del fichero lista visualiza el texto "Alumno matriculado" y si no "Alumno no matriculado "

105.1.2. Variables

Un nombre de variable obedece a ciertas reglas:

1. Se puede componer de letras minúsculas, mayúsculas, cifras, caracteres de subrayado.
2. El primer carácter no puede ser una cifra.
3. El tamaño de un nombre suele ser ilimitado.

Se declara una variable en cuanto se le asigna un valor. Se efectúa la asignación con el signo **=**, sin espacio antes ni después del signo.

Ejemplo: **var=Lpi**

Se accede al contenido de una variable colocando el signo **\$** delante del nombre de la variable.

Cuando el shell encuentra el **\$**, intenta interpretar la palabra siguiente como si fuera una variable. Si existe, entonces se sustituye el **\$nombre_variable** por su contenido, o por un texto vacío en el caso contrario.

105.1.3. Uso de las llaves{}

Las llaves permiten sustituir el nombre de una variable por su contenido

Ejemplo:\$fichero=lista

\$cp \${fichero}1 {fichero}2

Esto equivale a:\$cp lista1 lista2

- **\$ {VARIABLE?texto}**

Si la VARIABLE ha sido asignada, esta expresión representa su contenido. En caso contrario, el texto es visualizado por la pantalla y el shell termina su ejecución.

Si no se especifica texto, el shell emite su propio mensaje.

Ejemplo: \$echo \${EDITOR?'Por favor, especifique EDITOR'}

Si la variable EDITOR ha sido "asignada, este ejemplo visualiza su contenido,y en caso contrario visualizará el mensaje expresado, terminando el shell su ejecución.

- **\$ {VARIABLE +texto }**

Si la VARIABLE ha sido asignada, esta expresión representa el texto. En caso de que la variable no haya sido asignada, esta expresión representa el string nulo (vacio).

Ejemplo: \$ echo \${USUARIO+"Buenos dias, \$USUARIO"}

Este comando visualiza saludo, pero sólo en caso de que la variable USUARIO haya sido asignada.

- **\$ {VARIABLE -texto }**

Si la VARIABLE está vacía o es inexistente, el texto cogerá su sitio. En caso contrario, es el contenido de la variable la que prevalece.

- **\$ {VARIABLE =texto }**

Si la VARIABLE está vacía o es inexistente el texto cogerá su sitio y se convertirá en el valor de la variable.

105.1.4. Variables Standard del SHELL

Hay ciertas variables predefinidas por el Shell, que son creadas inmediatamente después de hacer login. Como tales variables, se les puede asignar otro contenido en cualquier momento que sea necesario.

Algunas de las variables standard del shell son:

Variables Standard del SHELL

Variable	Descripción
HOME	Es el nombre de camino (pathname) del usuario actual. Es el directorio en el "se encuentra" después de hacer login. El contenido de esta variable indica el directorio al que se cambiará cuando se utilice el comando cd sin argumentos
PATH	Esta variable debe contener una lista de nombres de camino (pathnames) separados entre sí mediante dos puntos (:) y sin espacios. Estos nombres de camino indican al shell dónde buscar los comandos que teclea el usuario. Por defecto, un comando sería buscado solo en el directorio actual (que puede no ser el asignado en principio al usuario) y mediante la especificación de otros directorios alternativos, el usuario tendrá acceso más fácil a más comandos. Se puede añadir fácilmente un directorio más a la variable PATH: <code>\$ PATH=\$PATH:/usr/comandos</code>
PS1	El nombre de esta variable es la abreviatura de "prompt string one". El prompt es el string de caracteres que muestra el shell como aviso al operador para que indicar que está listo para recibir comandos. El valor de PS1 es \$ para usuarios normales y # para el superusuario. <code>\$ PS1="Hola >" Hola > Hola >PS1="Teclee comando :" Teclee comando : Teclee comando :PS1="S "</code>
PS2	Esta variable representa el "prompt string two". Este es el prompt que el shell visualiza cuando se ha introducido un comando incompleto que continúa en otra línea. Al principio de todas las líneas de continuación y hasta que el comando se termina, el shell visualiza este prompt en lugar de PS1. El valor inicial de PS2 es el carácter >. <code>S VARIABLE='linea uno > linea dos > linea tres' S PS2="Siga: = S VARIABLE=' linea uno siga: linea dos Siga: linea tres' \$</code>
IFS	Esta variable contiene un string compuesto por los caracteres que el shell interpreta como separador entre palabras. Los caracteres que componen el string asignado normalmente a IFS son el espacio, el TAB y el salto de linea (newline). Son todos ellos caracteres no visibles, de modo que si intentamos visualizar el contenido de IFS parecerá que está vacío. Dado el cometido de esta variable, puede ser peligroso cambiar su contenido.
TERM	Esta variable contiene el nombre del terminal. No es utilizada por el shell, sino por el editor vi y otros programas que explotan las posibilidades del terminal a través del paquete curses de rutinas de manejo de pantalla. El valor de esta variable se utiliza para localizar características del terminal en los ficheros /etc/termcap y /usr/lib/terminfo
TERMCAP	Todas las características del terminal (TERMinal CAPabilities) que aparecen en el fichero /etc/termcap pueden ser asignadas a la variable TERMCAP para ahorrar tiempo al programa vi y a otros, evitándoles la necesidad de abrir y leer el fichero /etc/termcap
LOGNAME	Esta variable contiene el nombre mediante el cual el usuario ha entrado al sistema a través de login
USER	Contiene el nombre del usuario en curso
MAIL	Ruta y fichero que contiene los mensajes del usuario
PWD	Ruta actual

105.1.5. Variables locales y Variables de entorno.

Existen dos listas de variables mantenidas por el shell:

- variables locales

- variables de entorno

Comandos para uso con variables:

- El comando **set**: Utilizando el comando set sin argumentos, se visualiza una lista de todas las variables locales y sus valores asignados.

Cuando se asigna una variable, ésta es ubicada en el grupo de las variables locales.

- El comando **unset**:

Suprime la variable especificada.

- El comando **readonly**: Protege una variable en modo solo lectura (no se puede eliminar, salvo saliendo del shell)
- El comando **env**: El comando env sirve para visualizar la lista de variables de entorno y sus valores asignados.

A diferencia de las variables locales, las variables de entorno son pasadas a todos los programas y subshells.

El mantenimiento de estas dos listas de variables separadas permite cambiar una variable en un shell sin afectar a otros programas y subshells.

- El comando **export**: El comando export sirve para "exportar" una variable de la lista de variables locales a la lista de variables de entorno.

Ejemplo: \$ export VARIABLE1 VARIABLE2 VARIABLE3 ...

Los nombres de variables de la lista a exportar deben ir expresados sin el carácter \$.

Uso: export [-fn] [nombre[=valor] ...] ó export -p

- El comando **Alias**: Un alias es un atajo a un comando y a sus posibles parámetros. Utilizado el comando alias sin argumentos , lista los alias disponibles.

Uso: alias [-p] [nombre[=valor] ...]

Ejemplo: alias deltree='rm -rf'

- El fichero **.profile**: Desde luego no es necesario especificar los valores de las variables cada vez que se hace login.

Basta con realizar estas asignaciones en un fichero llamado .profile ubicado en el directorio del usuario.

Cada usuario podrá tener un fichero .profile en su directorio inicial, y asignar de esta forma las variables que necesite, tanto locales como de entorno.

El fichero /etc/profile es común para todos los usuarios, es decir, las asignaciones de variables que se hagan en este fichero afectarán a todos y cada uno de los usuarios.

NOTA: Cuando el nombre de un fichero empieza por un punto, como es el caso de .profile, no es "visto"(oculto) por algunos comandos como ls, rm, etc.

- **\$?** (status de retorno de comandos): Como ya hemos comentado previamente la variable \$? contiene el status de terminación del último comando ejecutado.

Valor 0 terminación normal, un valor distinto de cero para indicar la terminación con algún tipo de problema.

- **\$\$** (identificador de proceso): En Unix, cada proceso en ejecución tiene un número único en

ese momento que es su identificador de proceso (process identifier o PID). Consultando el valor de \$S podremos saber el identificador del proceso actual. Dada su unicidad se suele utilizar en la creación de ficheros temporales cuyo nombre es preciso que no lo tenga ningún otro fichero en ese momento.

Ejemplo:

```
$ TMP=tmp$$  
echo $TMP  
tmp1488
```

- \$! (identificador de proceso background): Esta variable contiene el identificador de proceso del último comando que se ha ejecutado en background.

Puede ser necesario "matar" un proceso en background. Para ello es necesario el comando kill y saber el número de identificación del proceso

Ejemplo: \$ Kill \$|

- \$-(las opciones del Shell)

105.1.5. Ficheros de configuración BASH

Se puede lanzar el shell bash en varios modos:

1. shell interactiva de conexión
2. shell interactiva simple
3. shell no interactiva
4. modo sh

Según su modo de lanzamiento, el shell va a buscar y ejecutar varios scripts y ficheros de configuración.

Un fichero de configuración es un script de shell, una secuencia de comandos individuales que tiene como meta configurar el entorno del usuario.

1. Shell de conexión:

Se ejecuta el shell de conexión después de la validación del login y de la contraseña en la consola.

En este modo, el Shell busca ejecutar, en este orden y si están presentes:

/etc/profile (define las variables de entorno importantes como, LOGNAME, USER, PATH, HOSTNAME...)

~/.bash_profile. Está en el directorio home del usuario.

Puede definir variables adicionales y llama a otro script: ~/.bashrc, que a su vez llama a /etc/bashrc (sirve para definir las funciones y alias para todo el sistema y todos los usuarios en bash)

~/.bash_login

~/.profile

En el momento de desconectarse, intenta ejecutar: ~/.bash_logout

2. Shell simple:

El shell simple corresponde a la ejecución del bash en una ventana (xterm, konsole), una consola o

manualmente (teclear bash en una consola). En este caso, sólo se ejecutará el fichero siguiente si existe:

`~/.bashrc` (Puede definir alias y funciones)

3. Modo no interactivo:

Se puede iniciar el shell en modo interactivo. Es cuando un usuario ejecuta un script. En principio, no hay ningún script ejecutado por defecto en el momento del inicio, salvo si se especifica una variable `BASH_ENV` que contiene la ruta de un script. En este caso, bash carga y ejecuta este fichero antes de comenzar con la ejecución del script o del comando.

4. Modo Bourne Shell:

Cuando se inicia el bash en modo Bourne Shell mediante el comando `sh`, se intenta ejecutar los ficheros en este orden:

`/etc/profile`

`~/.profile`

105.1 EXTRAS

105.1 EXTRAS opciones del shell

Comando interno shopt

Permite controlar comportamientos opcionales del shell

- sin opciones, muestra una lista de características indicando si están activas (on) o no (off)
- opciones:
 - `-s optname` activa la opción indicada
 - `-u optname` desactiva la opción indicada
- Ejemplo:

```
$ touch hola Hola
$ ls h*
hola
$ shopt -s nocaseglob
$ ls h*
hola Hola
```

- para más información, manual de `bash-builtins`

Opciones del shell con set

Con `set -o` se visualizan las opciones (con valores booleanos) que se pueden activar o desactivar en el shell. Para cambiar el valor se usa `#set -o <opción>` `#set +o <opción>`

```
root@cli:~# set -o
allexport      off
braceexpand    on
emacs         on
errexit       off
errtrace      off
functrace     off
```

```
hashall          on
histexpand       on
history          on
ignoreeof        off
interactive-comments  on
keyword          off
monitor          on
noclobber        on
noexec           off
noglob           off
nolog            off
notify           off
nounset          off
onecmd           off
physical         off
pipefail         off
posix             off
privileged       off
verbose          off
vi                off
xtrace           off
```

Ejemplo:

```
root@cli:~# set -o | grep ignoreeof
ignoreeof      off
root@cli:~# set -o ignoreeof
root@cli:~# set -o | grep ignoreeof
ignoreeof      on
root@cli:~# Use "logout" para dejar el shell.
root@cli:~# set +o ignoreeof
root@cli:~# set -o | grep ignoreeof
ignoreeof      off
root@cli:~#
```

105.2. Editar y escribir scripts simples.

Peso en el examen de certificación: 4 puntos.

Objetivo: Editar scripts existentes o escribir nuevos BASH-scripts sencillos.

Conceptos y áreas de conocimiento:

- Usar sintaxis estandar de shell (loops, tests).
- Usar sustitución de comandos.
- Analizar valores de retorno identificando el éxito o el fracaso de la ejecución y cualquier información proporcionada por el comando.
- Realizar correo condicional hacia el superusuario.
- Seleccionar el interprete de shell correcto haciendo uso de la línea *shebang* (#!).
- Gestionar la ubicación, el propietario, la ejecución y el permiso especial *suid* de los scripts.

Términos y utilidades:

- for
- while
- test
- if
- read
- seq

105.2.1. Programación SHELL

El Shell dispone de un lenguaje de programación. Se agrupan todas las instrucciones y comandos dentro de un script. Durante su ejecución, cada línea se leerá una por una y se ejecutara.

Una línea puede componerse de comandos internos o externos.

Una línea de comentarios siempre empieza con el carácter #.

La primera línea reviste una importancia especial, ya que permite indicar que Shell va a ejecutar el script

Ejemplos: `#!/bin/bash` Shell Script Bourne Again

`#!/bin/ksh` Shell Script Korn Shell

105.2.2. Argumentos de un Script

1. Parámetros posicionales S1, S2, S3, etc

Cuando se llama a un fichero de comandos ahell para su ejecución, se le pueden pasar parámetros de igual forma que se hace con los ommandos de Unix:

`$ comando arg1 arg2 arg3`

El Shell pasa estos parámetros al interior del programa para ser tratados. Los nombre de variables que asigna a estos parámetros posicionales son:

\$1 primer argumento

\$2 segundo argumento

\$3 tercer argumento

2. \$* (todos los argumentos) y \$# (contador de argumentos)

La variable \$* contiene todos los argumentos de llamada al programa de shell o shellscript.

La variable \$# contiene el número de argumentos con que se ha llamado al programa de shell o shellscript

3. \$@ (todos los argumentos)

La diferencia entre esta variable y \$* es que cuando va entrecomillada, su valor es una lista de argumentos separados, mientras que "\$*" tiene como valor un solo string con la lista de todos los argumentos.

Es fácil apreciar la diferencia con un ejemplo:

```
$ cat numarg
$ echo $#
$
$ set a b c
$ numarg $*
3
$ numarg "$*"
1
$ numarg $@
3
$ numarg "$@"
3
```

4. Reorganización de los parámetros

El comando **shift** permite modificar la posición de los parámetros. Al ejecutar shift desplaza todos los parámetros una posición, suprimiendo el primero: \$2 se convierte en \$1, \$3 se convierte en \$2, y así sucesivamente. El \$1 original desaparece. \$#,\$* y \$@ se vuelven a definir en consecuencia.

El comando **shift** seguido de un valor n efectúa un desplazamiento de n elementos. Así, con shift 4 \$5 se convierte en \$1, \$6 se convierte en \$2...

5. Salida de script

El comando exit permite terminar un script. Por defecto el valor devuelto es 0 (no error) , pero se puede especificar cualquier otro valor de 0 a 255.

Se puede recuperar el valor de salida mediante la variable \$?

105.2.3. Sustitución de Comandos

El lenguaje de shell contempla la posibilidad de poder sustituir la salida de un comando por el comando en si. De esta forma podemos visualizar la salida del comando o asignarla a una variable para tratarla posteriormente, etc.

Para hacer este tipo de utilización de un comando basta con encerrarlo entre ` (comilla inversa).

Ejemplos:

\$ echo `date` --> Visualiza la fecha y hora actual.

\$ fecha=`date` --> Asigna a la variable fecha un texto consistente en la salida del comando date.

\$ echo "Hay " `who | wc -l` "usuarios activos" --> Visualiza el número de usuarios activos.

105.2.3. Control de Flujo

Hasta ahora hemos visto la forma en que el shell permite ejecutar cadenas de comandos, bastando para ello incluir una secuencia de los mismos en un fichero de texto y darle permiso de ejecución.

Ahora vamos a ver las distintas formas en que se puede alterar la secuencia de comandos a ejecutar dentro de un programa de shell, tomando decisiones sobre diversas condiciones:

- **La sentencia if:**

```
if [ condicion ]
then
    lista de comandos
else
    if [ condicion2 ]
    then
        lista de comandos 2
    fi
fi
```

```
if [ condicion ]
then
    lista de comandos
elif [ condicion2 ]
then
    lista de comandos
fi
```

La sentencia if ejecuta una lista de comandos, y si todos ellos devuelven un status de salida 0, es decir, terminan su ejecución sin problemas, entonces se ejecutará otra lista de comandos bajo esta condición.

Ejemplo:

```
$ if grep '^manuel:' /etc/passwd > /dev/null then
echo "manuel esta registrado en /etc/passwd"
fi
$
```

En este ejemplo se visualiza un determinado mensaje en caso de que el comando grep encuentre "manuel:" como principio de alguna de las líneas del fichero /etc/passwd.

Si el comando grep no lo encuentra, dará un status de salida distinto de cero y el comando echo no será ejecutado.

La sentencia elif o abreviatura de "else if", es útil en caso de tener que chequear más condiciones cuando la condición del primer if no se cumple.

Ejemplo:

```
$ if test - fich1
then
cat fich1
elif test - fich2
cat fich2
fi
$
```

Si existe el fichero "fich1", visualizará su contenido, en caso contrario, visualizar "fich2" si existe.

Finalmente, la sentencia else sirve para ejecutar una lista de comandos en caso de que las condiciones de los if y elif anteriores hallan fallado.

Ejemplo:

```
$ if test -r fich1
then
cat fich1
elif test -r fich2
cat fich2
else
echo -no existe ni fich1 ni fich2
fi
$
```

- **Comando test**

El comando test no forma parte de la Shell, es un comando de Unix, pero esta pensado para usarlo dentro de programas de shell.

Básicamente, los argumentos de test forman una expresión. Si esta expresión es cierta, test devuelve un valor de salida = 0, y si es falsa, devuelve un valor distinto de cero, con lo que resulta especialmente apropiado para usarlo en construcciones de tipo if.

Las dos formas de realizar los test son:

test expresion
o bien [expresion]

Utilización de test sobre ficheros:

Utilización de Test sobre Ficheros	
Comando	Descripción
-r fichero	ver si existe fichero y se puede leer.
-w fichero	ver si existe fichero y se puede grabar.
-x fichero	ver si existe fichero y es ejecutable.
-f fichero	ver si existe fichero y es un fichero ordinario
-d fichero	ver si existe fichero y es un directorio
-c fichero	ver si existe fichero y es un fichero especial de tipo "carácter".
-b fichero	ver si existe fichero y es un fichero especial de tipo "bloque".
-p fichero	ver si existe fichero y es de tipo FIFO (los FIFOs sirven para comunicar datos entre procesos, en la forma "primera entrada primera salida o First-In-First-Out").
-u fichero	ver si existe fichero y tiene activado el bit "set-user-Id"
-g fichero	ver si existe fichero y tiene activado el bit "set-group-Id"
-k fichero	ver si existe fichero y tiene activado el "sticky bit"
-s fichero	ver si existe fichero y tiene tamaño superior a cero
-t [descriptor]	ver si el fichero abierto relacionado con el descriptor (1 por defecto) está asociado con un periférico de tipo terminal
-L fichero	El fichero es un vínculo simbólico
-e fichero	El fichero existe

Utilización de test sobre argumentos numéricos:

El comando test puede aplicarse a valores numéricos – ya sean representados de forma literal o mediante variables para establecer comparaciones-.

Téngase en cuenta que el Shell sólo maneja cadenas de caracteres y siempre las interpreta como tales; por eso se recurre a un comando externo (test) para dar un tratamiento numérico a las cadenas de caracteres que simbolizan números.

Utilización de Test sobre Argumentos Numéricos	
Comando	Descripción
n1 -eq n2	ver si n1 es numéricamente igual a n2.
n1 -ne n2	ver si n1 no es igual que n2.
n1 -gt n2	ver si n1 es mayor que n2.
n1 -ge n2	ver si n1 es mayor o igual que n2.
n1 -lt n2	ver si n1 es menor que n2.
n1 -le n2	ver si n1 es menor o igual que n2.

Este tipo de comparaciones pueden combinarse con los operadores:

! operador de negación

-a operador “and” binario

-o operador “or” binario

(expr) paréntesis para agrupar expresiones

NOTA: Téngase en cuenta que todos los operadores son argumentos separados para el comando

test, y no deben juntarse.

Utilización de test sobre strings:

Utilización de Test sobre Strings	
Comando	Descripción
-z str	ver si la longitud del string str es cero
-n str	ver si la longitud del string str no es cero
str1=str2	ver si los strings str1 y str2 son idénticos
str1!=str2	ver si los strings str1 y str2 no son idénticos
str	ver si el string str no es un string nulo

La sentencia while

While es una de las sentencias de control estructurado de bucles. Los comandos de la lista se ejecutan repetidamente mientras que el comando que representa la condición devuelva un status de 0, lo que equivale a decir que la condición sea cierta.

While [condicion]

do

lista de comandos

done

Ejemplo: Toma la lista de argumentos de llamada como nombres de usuario, y los busca mediante el comando grep en el fichero /etc/passwd indicando si han sido localizados.

```
if test $# -eq 0
then echo " Utilizacion: $0 nom1 nom2 etc"
exit 1
fi
BB=/dev/null
while test $# -gt 0
do
if grep "^${1}:" /etc/passwd >SBB 2>$BB
then
echo "$1 esta en /etc/passwd"
else
echo "$1 no esta en /etc/passwd"
fi
shift
done
```

- **Sentencia until**

Las construcciones de tipo until son funcionalmente semejantes a los de tipo while, con la única diferencia de que el bucle se repite mientras que la condición sea falsa

```
until [condicion]
do
lista de comandos
done
```

Un bucle de este tipo equivale a:

```
while [ ! condicion ]
do
lista de comandos
done
```

- **Sentencia for**

En la mayoría de los lenguajes, los bucles de tipo "for" se repiten incrementando o decrementando el valor de cierta variable numérica.

La utilización de for en el shell es algo diferente. Las construcciones de tipo for permiten repetir una lista de comandos tantas veces como palabras se incluyan después de la palabra clave in, asignando cada una de estas palabras a la VARIABLE en cada repetición del bucle for VARIABLE in lista

```
for VARIABLE in lista
do
lista de comandos
done
```

Ejemplo con do: Este bucle clasifica en orden alfabético todos los ficheros del directorio actual.

```
$ for fichero in *
do
sort <$fichero >tmp
mv tmp $fichero
done
```

Ejemplo con for:

```
COMANDO=$1
shift
for fichero in $*
do
eval $COMANDO < $fichero >tmp
```

```
mv tmp $fichero  
done
```

- **Sentencia case**

La sentencia case permite ejecutar de forma selectiva una lista de comandos elegida entre varias listas.

case palabra in

Patrón_1

Lista comandos

;;

Patrón_2

Lista de comandos

;;

esac

dónde Patrón_i es una expresión regular que utiliza los caracteres especiales.

Cualquiera de estos patrones puede estar compuesto por varias expresiones regulares separadas por una barra vertical (|). El shell interpreta la barra vertical como el operador OR, de tal forma que la palabra de cabecera de la sentencia case coincidirá con "patrón_i" si coincide con cualquiera de las expresiones regulares que lo forman.

Cuando el shell ejecuta la sentencia case, compara la "palabra" con cada patrón, y ejecuta la lista de comandos que van detrás del mismo si éste coincide con la palabra.

Solo se ejecutará una de las listas de comandos, ya que, cuando se encuentra un patrón que coincide con la palabra, después de ejecutar la lista de comandos correspondiente, el control salta hasta la sentencia inmediatamente posterior a esac.

Observaciones sobre la sentencia case: Si en una sentencia case se quiere especificar una lista de comandos por defecto, ésta deberá ir asociada al último ,patrón, que puede ser un asterisco (*), ya que éste coincide con cualquier palabra.

La sentencia break no afecta a la sentencia case directamente, pero puede terminar un case si dicha sentencia reside dentro de un bucle estructurado.

Ejemplo de utilización de case: Construir un menú de opciones para ejecutar varios posibles comandos.

```
echo "Seleccione una de las siguientes opciones"  
echo "-----"  
echo  
echo "(1) Listar el directorio  
echo "(2) Ver usuarios activos  
echo "(3) Visualizar Fecha y hora
```

```

echo
echo "Seleccione: \c"
read opcion
case opcion in
1) ls
2) who
3) date
*) echo "ninguna opción seleccionada"
esac

```

- **Sentencias break [n] y continue [n]**

Existen dos sentencias que permiten interrumpir la secuencia de comandos que se ejecutan dentro de un bucle for, while o until.

- **Break [n]**

La sentencia break interrumpe la ejecución del bucle y desvía el control a la sentencia posterior a la de terminación del bucle.

Se puede salir de un bucle que esté incluido dentro de otro, y éste a su vez dentro de otro, etc, especificando el número de bucles a abandonar mediante un número entero inmediatamente después de la palabra break.

Ejemplo:

```

$ while true
do
until false
do
if true
then
break
fi
done
echo bucle 1
done
echo bucle 2
Bucle2
$
```

Este ejemplo ilustra la salida con break 2 desde un bucle que está incluido en otro. Ello ocasiona que el comando "echo bucle 1" no sea ejecutado, ya que la ejecución salta al comando siguiente al "done" de terminación del bucle mas externo.

- **Continue [n]**

La sentencia continue interrumpe la secuencia de ejecución de comandos dentro de un bucle,

desviando el control a la cabecera del bucle para comenzar una nueva iteración.

Se puede especificar el número de bucles a saltar, de forma similar a la sentencia break.

- **Comando read**

El comando **read** permite al usuario insertar una cadena y colocarla en una o varias variables.

```
read var1 [var2 ...]
```

Si se especifican varias variables, la primera palabra irá en var1; la segunda, en var2, y así sucesivamente. Si hay menos variables que palabras, las últimas palabras van en la última variable.

Ejemplo:

```
echo "Inserte dos palabras o más :\"  
read palabra1 palabra2
```

- **Funciones**

Las funciones son trozos de scripts con nombre, directamente llamados por su nombre, que pueden aceptar parámetros y devolver valores. Los nombres de funciones siguen las mismas reglas que las variables, excepto que no se pueden exportar.

```
nombre_funcion ()  
{  
comandos  
return  
}
```

Las funciones pueden escribirse o bien en el script actual, o bien en otro archivo que puede incluirse en el entorno. Para ello, teclee: . nombreflic

El punto seguido de un nombre de fichero carga su contenido (funciones y variables) en el contexto actual.

El comando **return** permite asignar un valor de vuelta a una función. Bajo ningún concepto se debe utilizar el comando **exit** para salir de una función porque esta instrucción interrumpe también el script invocante.

- **Comando expr**

El comando **expr** permite efectuar cálculos sobre valores numéricos, comparaciones, así como la búsqueda en cadenas de texto.

Ejemplo:

```
$ expr 7 + 3  
10  
$ expr 7 \* 3  
21
```

Operadores	
Operador	Descripción
+	Suma.
-	Sustracción.
*	Multiplicación. Como el shell reconoce la estrella en tanto que comodín, hay que cerrarla con una contrabarra: *.
/	División.
%	Módulo.
!=	Diferente. Visualiza 1 si diferente, 0 en caso contrario.
=	Igual. Visualiza 1 si igual, 0 en caso contrario.
<	Inferior. Visualiza 1 si inferior, 0 en caso contrario
>	Superior. Visualiza 1 si superior, 0 en caso contrario.
<=	Inferior o igual. Visualiza 1 si inferior, 0 en caso contrario.
>=	Superior o igual. Visualiza 1 si superior, 0 en caso contrario
:	Búsqueda en una cadena. P. ej.: expr Julio: J* devuelve 1, ya que Julio empieza por J. Sintaxis particular: expr "Julio": ".*" devuelve la longitud de la cadena

105.2 EXTRAS

105.2 EXTRAS sed y awk

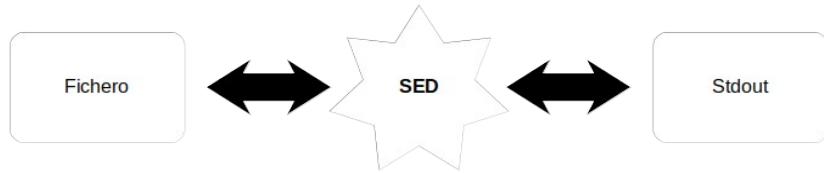
Índice

- ✓ *El Stream Editor*
- ✓ *Formato del comando*
- ✓ *Introducción a las expresiones regulares*
- ✓ *Uso del Stream Editor*
- ✓ *Procesamiento del texto mediante el comando AWK*

El Stream Editor

El programa sed (o stream editor) se utiliza para editar datos de ficheros sin necesidad de abrirlos en un editor, tal como el editor vi. Permite que se puedan modificar los datos del fichero desde la línea de comandos y enviar la salida a la pantalla de forma predeterminada. Todo esto permite que el administrador pueda de forma repetitiva, realizar cambios rápidos.

El editor sed no cambia el contenido del fichero de origen. Para guardar la salida de pantalla se puede redirigir la salida a un nuevo fichero. Este editor es el más utilizado para realizar cambios en varios ficheros de forma rápida.



Formato del comando

`sed [-options][address] command file... [>newfile]`

El comando **sed** se puede utilizar con la tubería de la siguiente forma:

`ls -l | sed '/2/d'`

Opciones del comando

Las opciones de este comando sirven únicamente para controlar los eventos de salida. Las opciones más comunes son:

- e** : Permite varias ediciones en la misma línea de comando
- n** : Suprime la salida predeterminada

Introducción a las Expresiones Regulares

Similar al comando **grep**, **sed** utiliza caracteres especiales de búsqueda. La siguiente tabla describe dichos caracteres especiales.

Carácte r	Propósito	Ejemplo	Resultado
^	Inicio de línea	'^patrón'	Concuerdan todas las líneas que empiecen por "patrón"
\$	Final de línea	'patrón\$'	Concuerdan todas las líneas que acaben por "patrón"
.	Sustituye a un carácter	'p.....n'	Concuerda con las líneas que contengan una "p" seguida de cinco caracteres y luego una "n".
[]	Concuerda un carácter del patrón	'[Pp]atrón'	Concuerda con aquellas líneas que contengan "Patrón" o "patrón"
*	Coincidir con el carácter anterior 0 o n veces.	'[a-z]*'	Concuerda con cualquier carácter alfabético en minúsculas.
[^]	Concuerda un carácter que no esté en el patrón	'[^a-m]atrón'	Concuerda con aquellas líneas que no contengan desde la letra "a" hasta la "m" seguidas de "atrón".

Uso del Stream Editor

El comando **sed** utiliza expresiones regulares como parámetros para realizar ediciones.

Borrado de líneas mediante el comando d

El siguiente ejemplo, muestra como **sed** busca líneas que contengan un modelo y las elimina:
`sed '/ pattern/d' filename`

Para borrar todas las líneas que contengan la palabra root del fichero /etc/group ejecutamos el siguiente comando:

```
$ sed '/root/d' /etc/group
```

Para borrar todas las líneas que contengan 3 en la salida del comando **ls**, escribimos:

```
$ ls -l | sed '/3/d'
```

Un ejemplo de la salida del comando ls y mediante sed eliminamos la línea 5 será el siguiente, guardando finalmente los resultados en el nuevo fichero:

```
$ ls -l | sed '5,$d' > newfile  
$ cat newfile
```

Visualización de las líneas mediante el comando pcommand

El siguiente ejemplo, muestra como sed imprime todas las líneas de salida estándar de forma predeterminada.

```
$ sed '/Dante/p' dante
```

Por defecto, sed imprime todas las líneas de la salida estándar. Si no se encuentra el modelo, dante, sed imprimirá una imagen de las líneas en la salida. Para suprimir esta acción predeterminada utilizaremos la acción –n con el comando p, de la siguiente forma:

```
$ sed -n '/Dante/p' dante
```

De esta forma, solo las líneas que contienen dante serán mostradas.

Agregar caracteres de final de línea

Para agregar caracteres de final de líneas para cada línea que se muestre en la salida estándar se puede utilizar la siguiente sintaxis:

```
$ ls -l | sed 's/$/ EOL/'
```

Reemplazar los espacios por columnas

Se pueden reemplazar cada espacio por un carácter de columna, para ello, ejecutamos el siguiente comando:

```
$ ls -l | sed 's/ *:/g'
```

Múltiples ediciones con sed

El siguiente ejemplo, muestra como se puede utilizar un único comando para editar varias líneas:

```
$ sed -e 's/Dante/DANTE/g' -e 's/poet/POET/g' dante
```

En este comando, se genera un nuevo fichero de la siguiente forma: La primera para reemplaza Dante por DANTE, la segunda parte reemplaza la palabra poet por POET . El resultado se muestra en pantalla. Podemos realizar ambas operaciones a la vez gracias a la utilización de la opción –e.

Procesamiento del texto mediante el comando AWK

El comando awk es un procesador de texto muy sencillo y fácil. Escanea un fichero (o entrada) línea a línea, desde la primera hasta la ultima, buscando por líneas que coincidan con un modelo y seleccionando acciones en estas líneas.

La sintaxis del comando es:

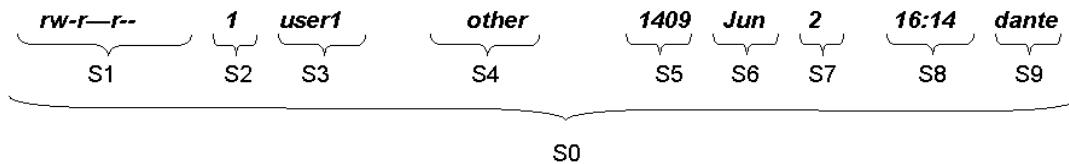
```
awk '{ action}' filename
```

Formato básico del comando awk

Si no se especifica ningún fichero se utiliza la entrada estándar, por ejemplo, el teclado. Un ejemplo de este comando, es el que mostraremos a continuación, en el que la salida del comando ls –l es pipeada con awk. Para cada línea recibida por awk, la acción imprimir se ejecutara, sobre la pantalla.

```
$ ls -l | awk '{print $0}'
```

El resultado de esta acción será exactamente el mismo que el de ejecutar el comando ls -l.



Cuando awk lee una línea, automáticamente la rompe y la guarda en campos. Cada campo se asigna a una variable. Los espacios o tabuladores se utilizan para delimitar los distintos campos. Los nombres de las variables son el signo del \$ seguido del número de campo contando de izquierda a derecha. Así, la variable \$1 representa el contenido del campo 1. La línea entera se representa por la variable \$0.

Utilización de awk para mostrar datos específicos

Vemos el siguiente ejemplo:

```
$ ls -l | awk '{print $3 $5 $9}'  
user154120dante  
user1368dante_1  
user1176dat  
user1512dir1  
user1512dir2  
user1512dir3  
user1512dir4  
user1235file1  
user1105file2  
user1218file3  
user1137file4  
user156fruit  
user157fruit2
```

Observamos, que la salida estándar no incluye espacio entre los tres campos. El siguiente ejemplo, muestra espacios entre los distintos campos:

```
$ ls -l | awk '{print $3, $5, $9}'  
user1 54120 dante  
user1 368 dante_1  
user1 176 dat  
user1 512 dir1  
user1 512 dir2  
user1 512 dir3  
user1 512 dir4  
user1 235 file1  
user1 105 file2  
user1 218 file3
```

```
user1 137 file4
user1 56 fruit
user1 57 fruit2
user1 512 practice
user1 28738 tutor.vi
```

En este ejemplo, a pesar de dejar espacios en blanco, los campos no se alinean. Para proveer una alineación correcta utilizamos el siguiente comando:

```
$ ls -l | awk '{print $3 "\t" $5 "\t" $9}'
user1 54120 dante
user1 368 dante_1
user1 176 dat
user1 512 dir1
user1 512 dir2
user1 512 dir3
user1 512 dir4
user1 235 file1
user1 105 file2
user1 218 file3
user1 137 file4
user1 57 fruit2
user1 512 practice
user1 28738 tutor.vi
```

Utilización de awk para cambiar el formato de los datos

Se puede utilizar el comando awk para reordenar los campos y cambiar su formato. El siguiente comando muestra los datos ordenados:

```
$ ls -l | awk '{print $9,$5,$3}'
dante 54120 user1
dante_1 368 user1
```

105.3. Administración de datos SQL.

Peso en el examen de certificación: 2 puntos.

Objetivo: Realizar accesos de cualquier naturaleza a bases de datos y manipular sus datos usando comandos SQL. Este objetivo incluye la realización de accesos que impliquen la unión de dos tablas y/o subconsultas.

Conceptos y áreas de conocimiento:

- Use of basic SQL commands.
- Perform basic data manipulation.

Términos y utilidades

- insert
- update
- select
- delete
- from
- where
- group by
- order by
- join

105.3.1 Introducción

El SQL, *Structured Query Language*, es un lenguaje estandarizado ISO de consulta y tratamiento de bases de datos relacionales. Se pueden descomponer sus posibilidades en cuatro funciones:

- El lenguaje de **definición de datos**: creación (CREATE TABLE), modificación (ALTER TABLE) y supresión de las tablas (DROP TABLE), índice (CREATE INDEX), etc.
- El lenguaje de **tratamiento de datos**: Concerne a las peticiones «clásicas»: añadido, supresión (delete), modificación update) y selección (Select) de datos.
- El lenguaje de **control de datos**: instalación y gestión de los privilegios de los usuarios de la base de datos.(GRANT y REVOKE)
- El lenguaje de **control de las transacciones**: gestión en particular de los «commit» y de las posibles vueltas atrás, procedimientos, etc.

105.3.2 Selección de datos

- **Select**

La instrucción SELECT permite extraer datos de una o varias tablas según determinados criterios.

SELECT apellido_campo1, apellido_campo2...FROM table;

Se devuelve el resultado en forma de tabla cuyos encabezamientos son los nombres de los campos

seleccionados. Es posible renombrarlos con AS. Por ejemplo:

```
SELECT apellido, nombre, id AS identificador FROM usuarios;
```

La utilización del asterisco «*» como campo permite seleccionar todos los campos de la tabla. La petición siguiente lista todo el contenido de la tabla:

```
SELECT * FROM usuarios;
```

- **From**

La cláusula FROM lista las tablas o ficheros que contienen los datos a recuperar por la consulta. El formato de esta cláusula es:

```
FROM nombretabla [alias_tabla] ...
```

- *nombretabla* puede ser una o mas nombres de tabla en el directorio de trabajo si se omite este, o en un directorio distinto si se especifica.
- *alias_tabla* es un nombre que se usa para referirse a la tabla en el resto de la sentencia SELECT para abreviar el nombre original y hacerlo más manejable, en el caso de existir más de una tabla en la consulta y, también para poder realizar consultas uniendo varias veces la misma tabla. Por ejemplo,

```
SELECT A.NOMBRE, A.APELLIDOS FROM MATRICUL M, ALUMNOS A WHERE  
M.MATRICULA = A.MATRICULA AND M.GRUPO = '1A' AND ANNO = 1995
```

es mucho más práctico y sencillo que:

```
SELECT ALUMNOS.NOMBRE, ALUMNOS.APELLIDOS FROM MATRICUL, ALUMNOS  
WHERE MATRICUL.MATRICULA = ALUMNO.MATRICULA AND MATRICUL.GRUPO = '1A'  
AND ANNO = 1995
```

Las dos sentencias son idénticas .

Si no se pudiera utilizar alias, no se podría unir una tabla consigo misma y la siguiente consulta no se podría llevar a cabo:

```
SELECT A.NOMBRE, A.APELLIDOS, H.NOMBRE FROM ALUMNOS A, ALUMNOS H WHERE  
A.PADRE = H.PADRE
```

que nos devolverá el nombre de aquellos alumnos/as que tienen el mismo padre, es decir, son hermanos.

- **Distinct**

Si la petición devuelve líneas idénticas, puede suprimir los duplicados con DISTINCT. En el caso siguiente, es lógico pensar que varios usuarios tienen el mismo nombre. La causa DISTINCT va a suprimir en salida los duplicados (o triplicados, etc.).

```
SELECT DISTINCT nombre FROM usuarios;
```

- **Where**

La cláusula WHERE especifica las condiciones de selección de las líneas.

```
SELECT apellido_campo1, apellido_campo2...
```

FROM table

WHERE condición;

Es posible aplicar una condición a cualquier campo de la tabla, incluyendo aquellos que no están presentes en la selección de campos.

Teniendo en cuenta el tipo de datos del campo o de los campos afectados por la condición, puede utilizar distintos operadores: $=$, \geq , \leq , $>$, $<$, \neq (diferente) y la negación con el signo de exclamación: \neq (diferente), $\not>$ (no superior), $\not<$ (no inferior). Se pueden relacionar los criterios de condiciones con los operadores lógicos AND, OR y NOT. Si el criterio es un texto, se pone entre comillas simples: 'titi'.

SELECT nombre FROM usuarios WHERE apellido='SANCHEZ';

Puede utilizar los atributos LIKE, BETWEEN o IN en una cláusula WHERE.

- LIKE compara el valor de un campo con un valor de texto especificado con posibles caracteres de sustitución: «%» para una cadena cualquiera y «_» para un carácter cualquiera.
- BETWEEN especifica un intervalo de datos.
- IN propone una lista de elementos.

Ejemplos:

Petición extrae todos los registros cuyos id están incluidos entre 0 y 100

*SELECT * FROM usuarios WHERE id BETWEEN 0 AND 1000;*

Ésta extrae todos los registros en los que el nombre corresponde a un nombre compuesto «Juan »: Juan Jacobo, Juan Carlos, Juan María, etc.

*SELECT * FROM usuarios WHERE nombre LIKE 'Juan %';*

Finalmente, ésta última extrae los usuarios que viven en Madrid, Barcelona, Valencia y Sevilla.

*SELECT * FROM usuarios WHERE ciudad IN ('Madrid', 'Barcelona', 'Valencia', 'Sevilla');*

105.3.3 Las expresiones y las funciones

Se puede utilizar expresiones, en particular las aritméticas, dentro de una instrucción SELECT, tanto en la selección de los campos como en la cláusula WHERE.

Ejemplo: La petición siguiente calcula el precio NETO de un producto considerando que el campo iva contiene el porcentaje de IVA aplicado al producto.

*SELECT precio+(precio*iva) AS precio_net FROM t_productos WHERE id_producto='1245';*

Si los campos son de texto, el símbolo «+» permite hacer concatenaciones:

SELECT apellido+' '+nombre FROM usuarios;

Las funciones se aplican como las expresiones en los campos seleccionados o en la cláusula WHERE. Son de tres tipos principales: matemáticas y estadísticas, fechas y cadenas de caracteres.

Ejemplos: `SELECT count(*) FROM usuarios;` Esta consulta cuenta el número de líneas en una tabla.

`SELECT min(precio), max(precio), avg(precio) FROM t_productos;` Extrae los precios mínimo, máximo y medio del conjunto de la tabla de los productos.

`SELECT sum(precio*ctd) from t_productos;` Extrae el importe total de las existencias de productos:

Las funciones de fecha se aplican sobre los campos de tipo fecha, salvo `current_date`, que devuelve la fecha del día: `SELECT current_date;`

Ejemplo: `SELECT * from usuarios WHERE month(f_inscripción)=’2012’;` Extrae todos los usuarios que se han inscrito en 2012.

105.3.4 Cláusula ORDER BY

La cláusula ORDER BY permite ordenar los resultados.

Es posible indicar el tipo de ordenación, creciente con ASC, decreciente con DESC.

Ejemplo: La petición siguiente ordena los productos de más barato a más caro teniendo en cuenta el IVA:

`SELECT precio+(precio*iva) AS precio_net FROM t_productos ORDER BY precio_net ASC;`

105.3.5 Cláusula GROUP BY

La cláusula GROUP BY agrupa los resultados por campos de valores idénticos.

Ejemplo: La petición siguiente visualiza el id de proveedor y número total de productos en reserva por id de proveedor:

`SELECT id_proveedor, sum(ctd) FROM t_productos GROUP BY id_proveedor;`

105.3.6 Cláusula HAVING

La cláusula HAVING dice a SQL que incluya solo ciertos grupos producidos por la cláusula GROUP BY en los resultados de la consulta.

Al igual que la cláusula WHERE, utiliza una condición de búsqueda para especificar los grupos deseados. En otras palabras, especifica la condición que deben de cumplir los grupos. Sólo es válida si previamente se ha especificado la cláusula GROUP BY. La cláusula HAVING tiene la forma:

HAVING expresión1 operador expresión2

- *expresión1* y *expresión2* pueden ser nombres de campos, valores constantes o expresiones y estas no deben coincidir con una expresión de columna en la cláusula SELECT.
- *operador* es un operador relacional que une las dos expresiones

Ejemplo: La sentencia siguiente nos mostrará el número de alumnos en cada grupo de 1995 cuyo numero de integrantes supera los 30:

```
SELECT GRUPO, COUNT(*) FROM MATRICULA WHERE AÑO = 1995 GROUP BY GRUPO  
HAVING COUNT(*) > 30 ;
```

105.3.7 Las combinaciones

Con SQL es posible recuperar registros de varias tablas al mismo tiempo. Para ello las tablas implicadas deben compartir campos que las puedan relacionar.

La combinación de tablas se basa en la relación de varias tablas a través de uno (o varios) campos (atributos) de cada tabla. No hay límite al número de tablas ni al de combinaciones.

```
SELECT tabla1.campo1, tabla2.campo1 FROM tabla1, tabla2  
WHERE tabla1.campocomun=tabla2.campocomun;
```

Ejemplo: consulta que extrae el nombre de cada proveedor para cada producto:

```
SELECT DISTINCT t_proveedores.nombre FROM t_proveedores, t_productos  
WHERE t_proveedores.id_proveedor=t_proveedores.id_proveedor;
```

Se puede utilizar el AS para renombrar las tablas en la salida de la consulta.

Si el nombre del campo es único en todas las tablas, entonces no hace falta especificar la tabla. Se puede transformar la consulta de la manera siguiente:

```
SELECT DISTINCT nombre FROM t_proveedores as t1, t_productos as t2  
WHERE t2.id_proveedor=t1.id_proveedor;
```

Se pueden efectuar combinaciones con más de dos tablas. Así, para obtener la lista de los proveedores por clientes, necesitará consultar la tabla de los clientes llamada "usuarios", la tabla de los pedidos, la tabla de los productos y la tabla de los proveedores:

```

SELECT DISTINCT t_proveedores.nombre
FROM usuarios, t_pedidos, t_productos
WHERE usuarios.id=t_pedidos.id
AND t_pedidos.id_producto=t_productos.id_producto
AND t_productos.id_proveedor=t_proveedores.id_proveedor
AND usuarios.id='1';

```

Es posible combinar una tabla consigo misma. El truco consiste en duplicar temporalmente la tabla después de FROM, pero con nombres diferentes especificados con AS.

Ejemplo: El ejemplo siguiente extrae los apellidos y nombres de alias del usuario 2:

```

SELECT B.apellido, B.nombre FROM usuarios AS A, usuarios AS B
WHERE B.id = A.id_alias AND A.id = '2';

```

105.3.8 Las Subconsultas

Es posible hacer selecciones anidadas: que la devolución de una primera selección sirve de criterio para una segunda selección.

Por ejemplo, veamos cómo extraer los apellidos y nombres de todos los alias:

```

SELECT id FROM usuarios
WHERE id IN (select id_alias from usuarios WHERE id_alias!="")

```

Si la subconsulta devuelve un único valor, puede usar un igual para pasarle el valor a la consulta superior. En caso de varios valores devueltos, utilice IN.

105.3.9 Insertar

El comando INSERT Inserta registros en una tabla.

```
INSERT INTO table (campo1, campo2, etc.) VALUES ('valor1', 'valor2'...);
```

Ejemplo: insertar un nuevo cliente en la tabla usuarios:

```

INSERT INTO usuarios (id, apellido, nombre, ciudad, id_alias)
VALUES ('3', 'Aguilera', 'Julio', 'Olmo', NULL);

```

5.3.10 Actualizaciones

La instrucción UPDATE actualiza uno o varios registros de una tabla. La cláusula WHERE es opcional.

UPDATE tabla SET campo1='valor', campo2='valor'[WHERE campon='valor']

Ejemplo en el que se actualizan todos los registros aumentando todos los precios un 5%:

*UPDATE t_productos SET precio=precio*1.05;*

105.3.11 Supresión

La instrucción DELETE suprime uno o varios registros de una o varias tablas. Cuidado, según el modelo básico de datos, con las restricciones de integridad: en algunos casos, suprimir una referencia puede suprimir n registros en cascada.

DELETE FROM tabla WHERE campo1='valor';

Ejemplo: Veamos cómo suprimir el usuario 3 creado más arriba:

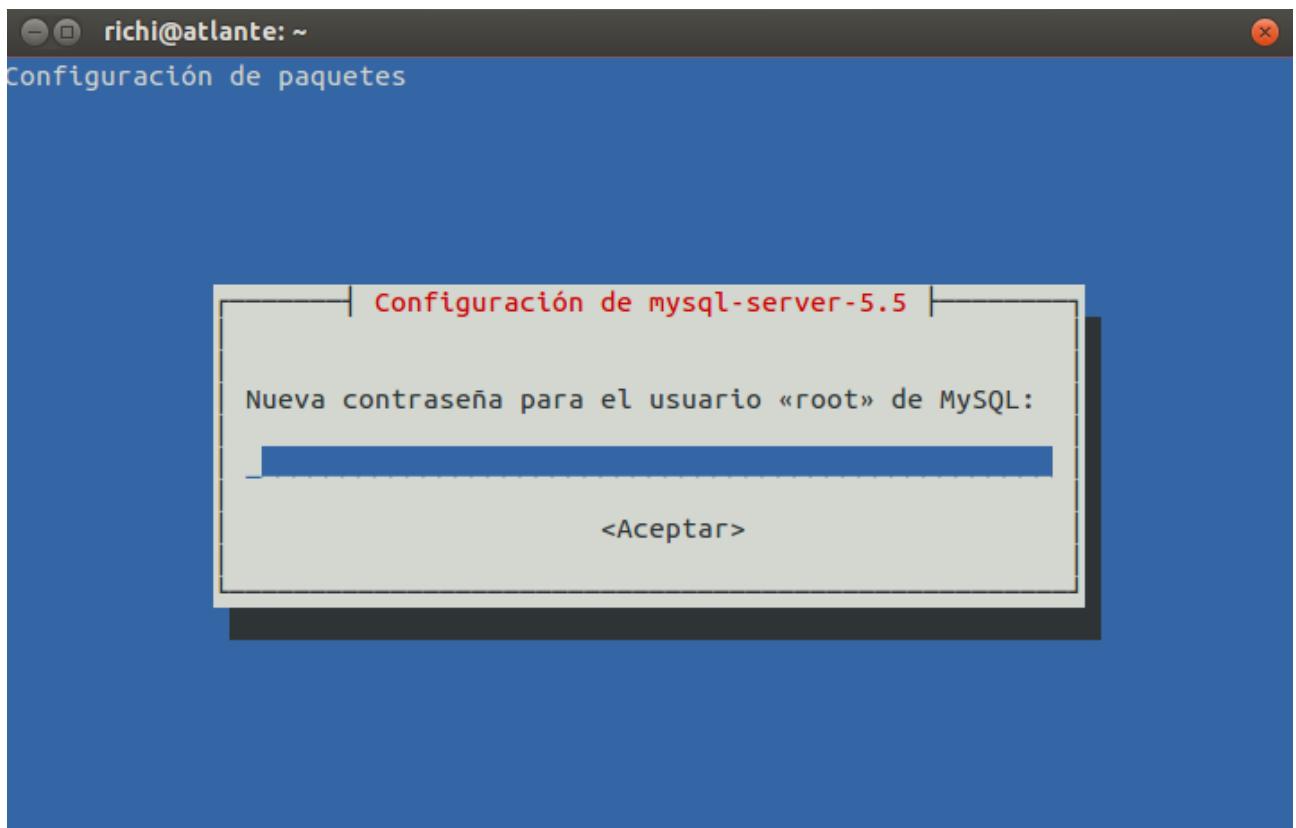
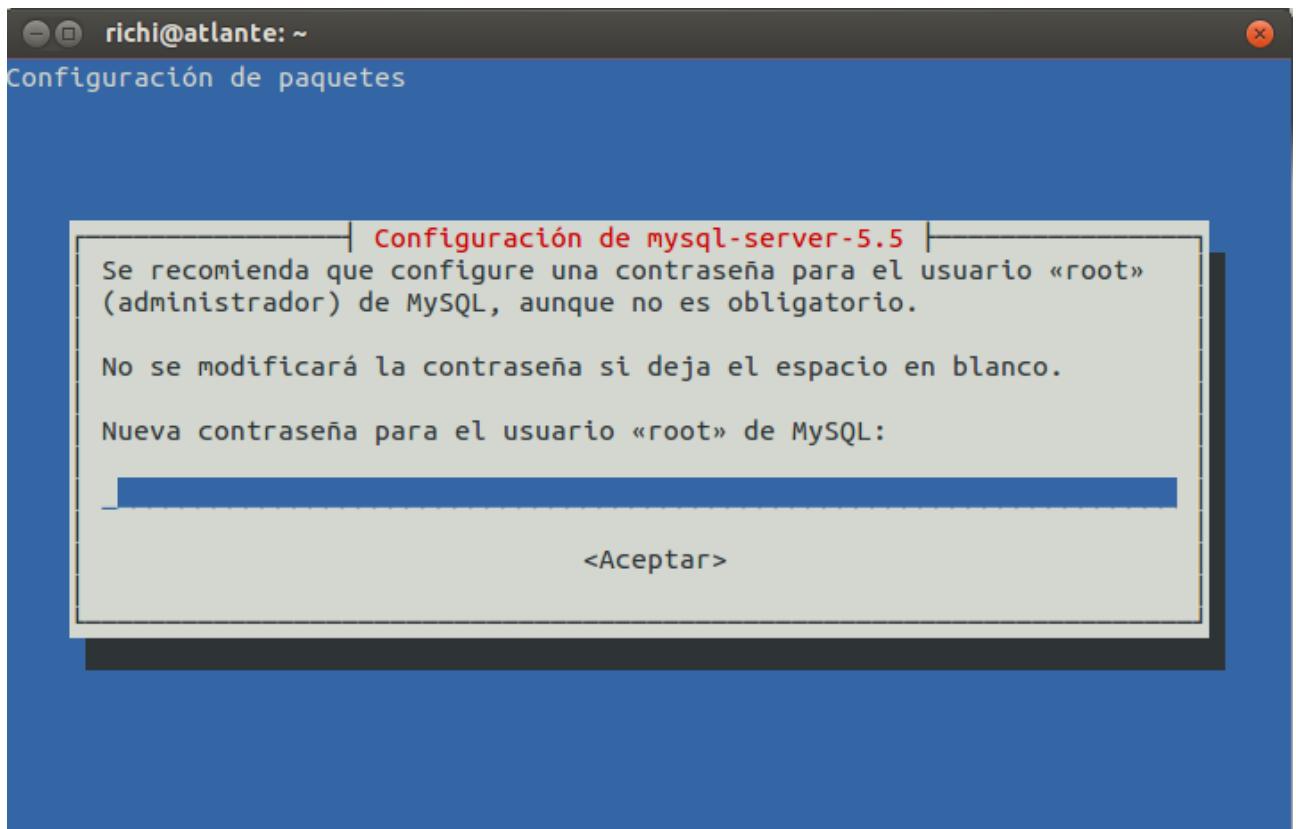
DELETE from usuarios WHERE id=3;

105.3 EXTRAS

105.3 EXTRAS Instalación mysql

```
root@server:~# apt-get install mysql-server mysql-client
```

Durante la instalación, el instalador solicita la contraseña del *root* del servidor MySQL. Este usuario *root* es específico de MySQL, por tanto debe tener una contraseña diferente a la del *root* del sistema.



Configuración

Por seguridad y mejor compatibilidad, la configuración predefinida de *MySQL* sólo aceptará conexiones locales (dirección 127.0.0.1). En caso de que se pretenda acceder al servidor *MySQL* desde la red interna, podemos cambiar el *bind-address* por la dirección de nuestro servidor (192.168.1.100).

/etc/mysql/my.cnf

```
# [...]
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
# [...]
```

Verificación

A partir de este momento, es posible acceder al monitor de *MySQL*:

```
root@server:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.30-1.1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> quit
Bye
root@server:~#
```

106 INTERFACES DE USUARIO Y ESCRITORIOS.

106.1 Instalar y configurar X11.

106.2. Configuración del gestor de login gráfico.

106.3. Accesibilidad

106.1 Instalar y configurar X11.

Peso en el examen de certificación: 2 puntos.

Objetivo: Instalar y configurar X11.

Conceptos y áreas de conocimiento:

- Verificar que la tarjeta gráfica y el monitor están soportados por el servidor X.
- Conocimiento del servidor de fuentes Xfonts.
- Entender y conocer el fichero de configuración de X Window.

Términos y utilidades:

- /etc/X11/xorg.conf
- xhost
- DISPLAY
- xwininfo
- xdpyinfo
- X

106.1.1. Introducción

El sistema gráfico de GNU/Linux se denomina **X Window System**, o de manera mas habitual **X Window, X11** o sencillamente **X**.

X es un sistema gráfico cliente/servidor que se encarga de dibujar y gestionar los componentes habituales de un entorno gráfico de usuario, también llamado GUI (Graphical User Interface), como son, las ventanas, los botones, los menús, las listas, las casillas de selección, el cursor de ratón, etc.

- El servidor X se encarga de recibir y responder las órdenes de visualización en la pantalla de la estación de trabajo local y de controlar peticiones de entrada vía teclado o ratón de los clientes X.
- Los clientes X son programas gráficos como el administrador de ficheros o el gestor de ventanas, con los que el usuario interactúa. El cliente se comunica con el servidor a través de un protocolo llamado *XProtocol*, el cual usa un componente llamado *Xlib* (biblioteca con los comandos básicos para generar ventanas, posicionarlas, o controlar eventos).

Al contrario que el servidor X que siempre se encuentra en el equipo local, un cliente X puede estar tanto en el sistema local como en un sistema remoto.

Entre las muchas peticiones gráficas que recibe el servidor X, la más importante es la de crear una ventana y dibujarla. Como X sólo facilita lo imprescindible, se limita a dibujar la ventana, pero no se ocupa de la decoración ni del manejo de la misma, por ello, a X también se le llama "Sistema de ventanas".

X no es un gestor de ventanas, sino que necesita de uno para controlar el manejo de ventanas, permitiendo al usuario instalar uno o más gestores de ventanas de su preferencia, como clientes de X. A los gestores de ventanas también se les denomina Window Manager, entorno de gráfico o entorno X. El servidor X visualiza el resultado dibujado por este gestor: las ventanas, movimientos, estilos, colores, etc. El Window Manager inicia una sesión de usuario directamente en el sistema X Window.

106.1.2. Sistema X Window

La primera implementación del sistema X Window para GNU/Linux se basaba en la adaptación distribuible de X Window versión 11 release 6 (habitualmente conocida como X11R6). Esta adaptación de libre distribución se conoce como *Xfree86*.

Un cambio de licencia en el 2004 hizo que XFree86 fuera un poco menos libre y dejó de ser compatible con la GPL de la "Free Software Foundation", por ello se creó una nueva rama de desarrollo, agrupándose la mayoría de los antiguos desarrolladores de XFree86, y crearon *X.org*, o dicho de otra manera, *X.org* es una "fork" de XFree86.

Por lo tanto, *X.org*, también llamado *Xorg*, se convirtió en el entorno X Window principal de GNU/Linux. *X.org* es rápido, modular y está disponible con numerosos drivers.

Este sistema es cada vez más popular, y forma parte de la mayoría de las distribuciones. Gentoo Linux, Fedora Core, Slackware, SuSE, Mandriva, Cygwin/X, Ubuntu y FreeBSD ya lo utilizan y Debian sólo en su versión estable.

106.1.3. Instalación de Xfree86

Generalmente no habrá que descargar los códigos fuente de Xfree86 y proceder a su instalación por el conocido procedimiento de descompresión, desempaquetado, configuración y compilación ya que, para aquellas distribuciones GNU/Linux que lo proporcionan queda disponible durante el proceso de instalación.

Pero si fuese necesario, se pueden encontrar distribuciones binarias de Xfree86 para GNU/Linux en varios servidores FTP. Si se opta por la instalación de alguna de ellas se hará lo siguiente:

Descargar y ejecutar el script *Xinstall.sh*, para conocer cualquier requisito necesario que haya que cumplir antes de continuar con la instalación. Satisfechos dichos requisitos se descarga la distribución binaria, se crea el directorio */usr/X11R6* (como root), y de desempaquetan los ficheros descargados en */usr/X11R6*. Una vez hecho esto hay que añadir */usr/X11R6/bin* al path de búsqueda editando el *.profile* o el script de inicio de la shell que se esté utilizando, ya sea *.profile*, *.cshrc*, *.login* o *.bashrc*. Si el sistema tiene varios usuarios la mejor solución es editar los ficheros de inicio generales del sistema como */etc/profile*.

También se debe verificar que el enlazador en modo de ejecución "runtime linker", *ld.so*, puede encontrar las librerías compartidas que utiliza el sistema X Window. Esto se hace añadiendo la línea */usr/X11R6/lib* al fichero */etc/ld.so.conf*. Ahora podemos configurar el sistema X Window XFree86.

106.1.4. Instalación de Xorg

También podemos instalar *X.org* a partir de su código fuente, pero como requiere mucho tiempo de compilación, es preferible instalar *Xorg* desde los CD, DVD o repositorios de su distribución.

Por defecto, se instala *Xorg* en */usr/X11R6*. Sin embargo, algunas distribuciones colocan los ficheros binarios, las librerías, los ficheros compartidos, etc., en el árbol clásico */usr* y en particular los módulos y drivers en */usr/lib/xorg* (o en */usr/lib64/xorg*).

Sin embargo, la configuración se encuentra siempre en el mismo sitio: */etc/X11*. Las últimas versiones son capaces de autodetectar el hardware (tarjeta gráfica, monitor, teclado y ratón) evitando así tener que realizar complejas configuraciones.

106.1.5. Equivalencias entre Xfree86 y Xorg

Debido a que la mayoría de las distribuciones actualmente usan X.org es conveniente conocer cómo se denominan las herramientas y los archivos de configuración en cada una de estos productos. La tabla siguiente marca estas diferencias.

Xfree 86	Xorg
/etc/X11/XF86Config	/etc/X11/xorg.conf
XF86config	xorgconfig
xf86cfg	xorgcfg

106.1.6. Configuración del sistema X Window

Al principio, el sistema X Window, era difícil configurarlo correctamente. Actualmente, con las nuevas versiones de Xfree86 y sobre todo de Xorg, la configuración se ha convertido en algo mucho más sencillo.

El servidor X cambió para soportar módulos cargables dependientes de la tarjeta de vídeo y del chipset utilizados, haciendo más fácil su manejo. Actualmente aunque es posible configurar manualmente los sistemas gráficos de GNU/Linux, son las distribuciones, mediante sus herramientas, las que se encargan de realizarlo.

Para determinar qué versión de X se está ejecutando, simplemente teclee el siguiente comando *X -version*:

\$ X -version

X.org X Server 1.6.1.901 (1.6.2 RC1)

Release Date 2009-5-8

X Protocol Version 11, Revision 0

En Xorg: El comando *xorgcfg* detecta de forma automática todos los parámetros de configuración del servidor X y crea el fichero de configuración *xorg.conf*. Una vez que el servidor X dispone de una configuración básica, se inicia X con un gestor de ventanas básico (*twm*) y una herramienta gráfica que permite al usuario realizar modificaciones en la configuración. El fichero de configuración de Xorg, "*xorg.conf*", está, como lo hemos comentado anteriormente, en */etc/X11: /etc/X11/xorg.conf*.

La herramienta *xorgconfig* es similar a *xorgcfg*, pero está basada en un asistente en modo de cuestionario de preguntas, debiendo contestar, en orden, a preguntas referentes al monitor, la tarjeta gráfica, el ratón y el teclado.

Se puede crear y editar manualmente el fichero **xorg.conf** mediante el comando *Xorg -configure*, que crea un nuevo fichero **xorg.conf** utilizando la información que es detectada en el sistema. Este archivo se creará en el directorio local.

Ejemplo creación del archivo **xorg.conf** en Fedora.

Xorg -configure

X.Org X Server 1.6.1.901 (1.6.2 RC 1)

Release Date: 2009-5-8

X Protocol Version 11, Revision 0

Build Operating System: Linux 2.6.18-128.1.6.el5 i686

Current Operating System: Linux Suffolk 2.6.29.6-213.fc11.i686.PAE #1 \\
SMP Tue Jul 7 20:59:29 EDT 2009 i686

Kernel command line: ro root=/dev/mapper/vg_suffolk-lv_root rhgb quiet

Build Date: 18 May 2009 02:47:59PM

Build ID: xorg-x11-server 1.6.1.901-1.fc11

Before reporting problems, check <http://wiki.x.org>

to make sure that you have the latest version.

Markers: (--) probed, (**) from config file, (==) default setting,

(++) from command line, (!!) notice, (II) informational,

(WW) warning, (EE) error, (NI) not implemented, (??) unknown.

(==) Log file: "/var/log/Xorg.1.log", Time: Wed Aug 12 06:32:31 2009

List of video drivers:

glint

nv

vmware

voodoo

radeon

mach64

geode

sisusb

intel

s3virge

```
ati
mga
amd
savage
ast
v4l
i128
neomagic
sis
dummy
ztv
trident
tdfx
cirrus
openchrome
apm
fbdev
vesa
(++) Using config file: "/root/xorg.conf.new"
Xorg detected your mouse at device /dev/input/mice.
Please check your config if the mouse is still not operational, as by default
Xorg tries to autodetect the protocol.
Your xorg.conf file is /root/xorg.conf.new
To test the server, run 'X -config /root/xorg.conf.new'
El fichero que crea se llama xorg.conf.new. Para utilizarlo la próxima vez que se inicie el sistema necesitamos verificarlo concienzudamente y modificarlo en caso necesario. Este nuevo fichero deberemos copiarlo al directorio /etc/X11/ con el nombre del archivo de configuración xorg.conf, haciendo:
# cp /root/xorg.conf.new /etc/X11/xorg.conf
En Xfree86: Los comandos de configuración equivalentes a xorgcfg de Xorg son XF86Setup o
```

XF86config. Estos programas crean el fichero de configuración */etc/X11/XF86Config*, que define los parámetros y el comportamiento del servidor X, incluyendo la localización de los ficheros, definiciones de fuentes, resoluciones de pantalla y configuraciones del monitor.

Las distribuciones Mandriva y Debian, utilizan un programa llamado *xf86config* o *Xconfigurator*, que no inician el servidor X hasta que no estén listos para chequear la configuración. El programa *Xconfigurator* es una versión mejorada de *xf86config* que está totalmente basado en texto.

Todos los programas de configuración de X Window van pasando por una serie de pasos para verificar el hardware de vídeo, el ratón o mouse, el tipo de monitor, las resoluciones deseadas y la profundidad de color. Con esta información, la herramienta de configuración intenta determinar que conjunto de valores de configuración son los apropiados y, a continuación, los somete a un test para averiguar cual de ellos es el que funciona mejor. El usuario debe decidir si la visualización es aceptable antes de que finalice el test o el programa decidirá que no es aceptable.

La selección de monitor es fundamental para que el servidor X conozca que prestaciones y que límites existen para la resolución y las frecuencias de refresco. Si no estamos seguros del tipo del monitor o no apareciese en el listado, siempre será mejor intentar seleccionar los tipos genéricos. La mayoría de los monitores se ajustan a la configuración de pantalla estándar, XGA (1024×768 píxeles a 60 Hz de frecuencia vertical).

Por otra parte, las tarjetas de vídeo de distintos fabricantes pueden llevar el mismo chipset, por lo que sería conveniente antes de iniciar el proceso de instalación verificar que el chipset de su tarjeta de vídeo es compatible con Xorg. Los Chipsets compatibles con Xorg se enumeran en la wiki de Xorg.

Distintas distribuciones de GNU/Linux ofrecen herramientas de configuraciones propias. Entre ellas podemos mencionar el programa *system-config-display* distribuido por "Red Hat Software". Es una herramienta GUI basada en X que comprueba las características del chipsets de la tarjeta gráfica y que puede instalarlo usando un gestor de paquetes como *yum*. Además, utilizando *system-config-display*, con permisos de **root**, en Fedora 10, puede crear el fichero *xorg.conf*, si es que necesita configurar un componente del sistema de forma manual, puesto que esta distribución descarto usar el fichero *xorg.conf*. Para ejecutar el programa *system-config-display* en la terminal, escriba:

```
# system-config-display
```

Algunos de los programas de configuración X llaman a *xvidtune*, que nos permite realizar ajustes para obtener la mejor visualización. **xvidtune** es un programa para ajustar la configuración de la pantalla. Proporciona una interfaz con diversos controles para configurar las frecuencias de actualización y la resolución de la pantalla.

106.1.6.1. El fichero *xorg.conf*

Todos los programas de configuración X terminan creando el fichero de configuración de X Window, normalmente en */etc/X11*. Durante la inicialización, el servidor mira la configuración existente en */etc/X11/xorg.conf* antes de presentar el interfaz X al usuario.

El fichero */etc/X11/xorg.conf* se estructura internamente en forma de secciones y de subsecciones. Cada una corresponde o a una funcionalidad del servidor X o a un periférico de entrada o salida.

Sección	Descripción
Files	Localización de los ficheros.
Server Flags	Banderas o switches del servidor.

InputDevice	Descripción de los dispositivos de entrada.
Monitor	Descripción del monitor.
Device	Descripción de los dispositivos gráficos.
Screen	Configuración de la pantalla.
ServerLayout	Presentación general.
Module	Carga dinámica de módulos.

Files

Esta sección se utiliza para especificar la ruta de la fuente (fonts) por defecto y de la base de datos de los colores RGB. Utilizando la directiva *FontPath* crea una lista de directorios donde el servidor X busca fuentes. La base de datos RGB es una tabla de equivalencia de valores numéricos de color rojo / verde /azul y los nombres de los colores. Cientos de nombres están definidos y pueden ser utilizados en la configuración de las X aplicaciones en las que los nombres de colores son obligatorios.

Ejemplo:

```
Section "Files"

InputDevices "/dev/gpmdata"
InputDevices "/dev/input/mice"
FontPath "/usr/share/fonts/misc:unscaled"
FontPath "/usr/share/fonts/75dpi:unscaled"
FontPath "/usr/share/fonts/100dpi:unscaled"
FontPath "/usr/share/fonts/Type1"
FontPath "/usr/share/fonts/URW"
FontPath "/usr/share/fonts/Speedo"
FontPath "/usr/share/fonts/cyrillic"
FontPath "/usr/share/fonts/truetype"
FontPath "/usr/share/fonts/uni:unscaled"
FontPath "/opt/kde3/share/fonts"
FontPath "/usr/local/share/fonts"
```

```
EndSection
```

Las rutas son vínculos simbólicos hacia los directorios correspondientes de /usr/share/fonts y retoman el formato de las entradas *FontPath*.

ServerFlags

Esta sección permite la personalización de las opciones del servidor X, tales como el manejo de teclas de acceso rápido.

El ejemplo muestra una sección que permite iniciar X sin ratón (nada impide utilizar luego un ratón USB o inalámbrico), con prohibición de parar el servidor, cambiar de resolución y pasar a consola. Es la configuración por defecto de un nodo de Internet en GNU/Linux.

```
Section "ServerFlags"
```

```
    Option "AllowMouseOpenFail" "on"
```

```
    Option "DontZap" "on"
```

```
    Option "DontZoom" "on"
```

```
    Option "DontVTSwitch" "on"
```

```
EndSection
```

InputDevice

En esta sección se utiliza una vez por cada tipo de dispositivo de entrada conectado al sistema. Aparecerá por lo menos dos veces: una para el teclado y otra vez para el ratón.

Ejemplo que describe la sección de un teclado:

```
Section "InputDevice"
```

```
    Identifier "Keyboard[0]"
```

```
    Driver "kbd"
```

```
    Option "Protocol" "Standard"
```

```
    Option "XkbLayout" "es"
```

```
    Option "XkbModel" "pc105"
```

```
    Option "XkbOptions" "caps:shiftlock"
```

```
    Option "XkbRules" "xfree86"
```

```
EndSection
```

Ejemplo que describe la sección de un ratón:

```
Section "InputDevice"
```

```
    Identifier "Mouse[1]"
```

```
Driver"evdev"

Option"Buttons" "10"

Option"InputFashion" "Mouse"

Option"Name" "Logitech USB R*"

Option"Protocol" "event"

Option"SendCoreEvents" "on"

Option"Vendor" "Sysp"

Option"ZAxisMapping" "4 5"

EndSection
```

Monitor

Se utiliza para definir las especificaciones de los monitores y para definir una lista de los modos de vídeo que se pueden manejar. Puede haber varias de ellas.

Ejemplo de sección que describe un monitor genérico de tipo Vesa que soporta una resolución de 1280x1024 a 60Hz.

```
Section "Monitor"

Identifier "Monitor[0]"

VendorName "--> VESA"

ModelName "1280X1024@60HZ"

UseModes "Modes[0]"

DisplaySize 340 270

HorizSync 31.0 - 64.0

VertRefresh 50.0 - 60.0

Option "DPMS"

EndSection
```

Device

Se utiliza para especificar las características del hardware de la tarjeta gráfica instalada. También puede haber varias de ellas.

El ejemplo siguiente muestra la configuración de una tarjeta gráfica NVidia.

```
Section "Device"

Identifier"Device[0]"
```

```
Driver"nvidia"
VendorName"Nvidia"
BoardName"GeForce 8600 GT"
Option"NoLogo" "0"
Option"DPI" "86 x 86"
Option"RenderAccel" "True"
Option"AddARGBGLXVisuals" "True"
EndSection
```

Otro ejemplo con una tarjeta ATI y el driver fglrx propietario:

```
Section "Device"
Identifier"aticonfig-Device[0]"
Driver"fglrx"
Option"XAANoOffscreenPixmaps" "true"
Option"TexturedVideo" "On"
Option"UseFastTLS" "1"
Option"Textured2D" "on"
Option"TexturedXRender" "on"
Option"BackingStore" "on"
Option"VideoOverlay" "Off"
Option"OpenGLOverlay" "Off"
BusID"PCI:1:0:0"
EndSection
```

Screen

La sección *Screen* une una tarjeta gráfica con su monitor correspondiente e incluye algunas opciones de configuración para ellos. Una sección Screen contiene una o varias subsecciones llamadas *Display* que determinan, para un tipo de visualización en n bits (8 bits: 256 colores, 16 bits: 65.536 colores, 24 bits: 16 millones de colores), cuáles deberían ser las resoluciones adecuadas. *Depth* define la profundidad de los colores en número de bits. *DefaultDepth*, la profundidad de los colores por defecto. *Modes*, las resoluciones soportadas. Además, la primera resolución de la lista es la resolución por defecto. Se puede pasar de una resolución a otra con la combinación de teclas "Ctrl Alt +", "Ctrl Alt -" o mediante las posibilidades propuestas por su

entorno de escritorio. Si no soporta una resolución, se desactiva automáticamente.

En el siguiente ejemplo, se han agrupado las secciones anteriores *Device* y *Monitor*. La sección *Screen* se llama *Screen[0]*. Cuatro subsecciones *Display* configuran la visualización en 8, 15, 16 y 24 bits, cada una, con cuatro resoluciones posibles. En este caso, la resolución por defecto es de 1.280x1.024 y la visualización por defecto será en 1.280x1.024 y 16 millones de colores.

Section "Screen"

Identifier "Screen[0]"

Device "Device[0]"

Monitor "Monitor[0]"

DefaultDepth 24

SubSection "Display"

Depth 15

Modes "1280x1024" "1024x768" "800x600" "640x480"

EndSubSection

SubSection "Display"

Depth 16

Modes "1280x1024" "1024x768" "800x600" "640x480"

EndSubSection

SubSection "Display"

Depth 24

Modes "1280x1024" "1024x768" "800x600" "640x480"

EndSubSection

SubSection "Display"

Depth 8

Modes "1280x1024" "1024x768" "800x600" "640x480"

EndSubSection

EndSection

ServerLayout

En esta sección une pareja monitor-tarjeta (*Screen*) con uno o más *InputDevices*. Pueden ser utilizados múltiples secciones *ServerLayout* para realizar configuraciones de sistemas con más de

un monitor. X gestiona muy bien las visualizaciones multipantalla y multitarjeta mediante una extensión llamada *Xinerama*. Un caso sencillo podría ser:

```
Section "ServerLayout"  
Identifier"Layout[all]"  
Screen"Screen[0]" 0 0  
InputDevice "Keyboard[0]" "CoreKeyboard"  
InputDevice "Mouse[1]" "CorePointer"  
EndSection
```

Si dispone de varios ratones (por ejemplo un ratón USB y un touchpad) o teclados (por ejemplo un teclado USB enchufado en la base de un portátil, y además el teclado del portátil), se puede declarar los otros periféricos de entradas en la sección *ServerLayout*, pero con el modificador "*SendCoreEvents*". Esto permite al periférico enviar peticiones al servidor X como los otros periféricos.

```
Section "ServerLayout"  
Identifier"Layout[all]"  
Screen"Screen[0]" 0 0  
InputDevice"Keyboard[0]" "CoreKeyboard"  
InputDevice"Mouse[1]" "CorePointer"  
InputDevice"Touchpad" "SendCoreEvents"  
EndSection
```

Module

La sección *Module* facilita al servidor X una lista de módulos complementarios y optativos que se han de cargar para añadirle nuevas funcionalidades. Se declara un módulo con una línea *Load*.

```
Section "Module"  
Load"dbe"  
Load"type1"  
Load"freetype"  
Load"extmod"  
Load"glx"  
EndSection
```

Una lista de módulos habitualmente usados, puede ser:

- *dbe* (Double Buffer Extension) dos búferes de visualización. Uno, principal, al que le corresponde la visualización actual, y otro, secundario, la visualización en segundo plano. Una vez que está listo, se conmutan los dos búferes. Este mecanismo evita que la imagen se cuelgue.
- *extmod* : módulo de extensión del protocolo X, que casi todo el mundo utiliza.
- *freetype* : permite utilizar fuentes de caracteres TrueType (ttf).
- *type1* : permite utilizar las fuentes Type1.
- *bitmap* : permite utilizar las fuentes bitmap (innecesario para las últimas versiones).
- *GLcore* : modo básico de añadido de las extensiones OpenGL.
- *glx* : extensiones GLX (extensión a OpenGL).
- *dri* (Direct Rendering Infrastructure): OpenGL llama a las funciones de la tarjeta gráfica y se acelera considerablemente la visualización en 3D.
- *i2c* : instalación del bus serie i2c, para comunicarse, entre otros, con el monitor.
- *ddc* : protocolo DDC para los monitores (Display Data Channel), que pasa por el bus i2c.
- *int10* : capa de emulación/acceso en tiempo real a la interrupción 10 de la tarjeta gráfica, en particular, para acceder a las funcionalidades VESA de la tarjeta y del monitor
- *vbe* (Vesa Bios Extension): extensiones Vesa para los accesos a algunos modos y resolución de la tarjeta.

La mayoría son optativos, pero algunos componentes como periféricos de entrada, aplicaciones y tarjetas gráficas, pueden necesitar ciertas extensiones para funcionar. Por ejemplo para hacer funcionar OpenOffice.org con el soporte de las fuentes de caracteres.

Extensions

Esta sección es optativa, y permite activar o desactivar extensiones de **Xorg**. Las más conocidas son *Damage* y *Composite*. La extensión *Damage* permite señalar a las ventanas que se deben volver a dibujar en parte. Por ejemplo, se puede volver a dibujar una zona de la pantalla mientras permanece cubierta por una ventana cuyo contenido no ha cambiado pero que puede ser transparente: es la de debajo la que ha cambiado. La opción compuesta es la que permite tener los efectos del mismo nombre: sombras, transparencia de las ventanas, alpha-blending, etc. asociada a las 3D, en particular AIGLX, que le permite obtener, con los efectos propuestos por Compiz-fusion, cubos 3D como efectos de escritorio.

Section "Extensions"

Option "Composite" "Enable"

EndSection

106.1.6.2. El fichero XF86Config

El fichero XF86Config también está dividido en secciones según la información específica de cada una:

<i>Sección</i>	<i>Descripción</i>
Files	Localización de los ficheros.
Server Flags	Banderas o switches del servidor.
Module	Carga dinámica de módulos.

InputDevice	Descripción de los dispositivos de entrada.
Device	Descripción de los dispositivos gráficos.
VideoAdaptor	Descripción del adaptador de vídeo Xv.
Monitor	Descripción del monitor.
Modes	Descripción de los modos de vídeo.
Screen	Configuración de la pantalla.
ServerLayout	Presentación general.
DRI	Configuración específica de DRI.
Vendor	Configuración específica del proveedor/fabricante.

Las secciones *ServerLayout* enlazan los dispositivos de entrada y de salida utilizados en una sesión. Los dispositivos de entrada se describen en las secciones *InputDevice*.

Los dispositivos de salida, normalmente, están formados por varios componentes independientes (como una tarjeta gráfica y un monitor). Las tarjetas gráficas se describen en las secciones *Device* y los monitores en *Monitor*. El monitor y la tarjeta de vídeo se tratan como uno solo en las secciones *Screen*, y a esos mismos conjuntos se hace referencia en la sección *ServerLayout*.

En general, todo lo que se ha explicado en el apartado anterior (6.1.6.1. *El fichero xorg.conf*) es de aplicación aquí. Se puede encontrar una documentación mas extensa sobre el formato y contenido del fichero XF86Config en el sitio oficial del proyecto **xfree86** <http://www.xfree86.org/>

106.1.6.3. XFonts

Los sistemas X Window se distribuyen con una colección básica de fuentes, como el texto de las ventanas de la terminal y de los navegadores, pero se pueden añadir fuentes adicionales al sistema. Existe gran variedad de fuentes disponibles, tanto gratuitas como comerciales. Algunas fuentes muy creativas son creadas por personas y distribuidas gratuitamente en Internet

En Xorg: Las distribuciones suelen buscar las fuentes de caracteres de manera automática en */usr/lib/X11/fonts*, como fuentes básicas del sistema, por lo que, la sección *Files* es optativa. X cuando se inicia, analiza los directorios de fuentes que encuentra en la sección *Files* del xorg.conf, si los hubiese, e incluye su contenido en la lista de fuentes disponibles durante la sesión de X. Se puede especificar rutas de fuentes propias con la directiva *FontPath* en la sección *Files* de xorg.conf. La sintaxis es simple: *FontPath "camino"*

Por ejemplo:

Section "Files"

```
FontPath "/usr/share/X11/fonts/misc"
```

```

FontPath "/usr/share/X11/fonts/cyrillic"
FontPath "/usr/share/X11/fonts/100dpi/:unscaled"
FontPath "/usr/share/X11/fonts/75dpi/:unscaled"
FontPath "/usr/share/X11/fonts/Type1"
FontPath "/usr/share/X11/fonts/100dpi"
FontPath "/usr/share/X11/fonts/75dpi"
FontPath "/usr/share/fonts/X11/misc"
# path to defoma fonts
FontPath "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
EndSection

```

Aunque las fuentes de caracteres por defecto se cargan de manera automática desde */usr/lib/X11/fonts*, algunas distribuciones modifican a menudo los scripts de ejecución de X para tener otras rutas. En las últimas de Mandriva, por ejemplo, todas las fuentes están en */usr/share/fonts*.

Añadir nuevas fuentes es muy sencillo. En primer lugar, sería recomendable crear un nuevo directorio para las nuevas fuentes, como por ejemplo */usr/share/X11/fonts/local* o */usr/local/fonts*, para separar las fuentes particulares de las fuentes de X.Org, y así, protegerlas durante las actualizaciones. Después de que las fuentes se instalen en el nuevo directorio, se ejecuta la utilidad *mkfontdir* para agregar las nuevas entradas al archivo *xorg.conf* y así incluir la ruta de las nuevas fuentes en el fichero de configuración de X. Por ejemplo, *FontPath "/usr/local/fonts"*. Ahora, el servidor X puede ser reiniciado para reconocer las nuevas fuentes, o bien las podemos agregar de manera dinámica con el comando *xset*:

```
# xset fp+ /usr/local/fonts
```

En una red con múltiples estaciones de trabajo, la gestión de las fuentes de forma manual puede llevar mucho tiempo. Para simplificar este problema, el administrador puede instalar todas las fuentes que desee en un solo sistema y ejecutando *xfs*, crear un servidor de fuentes X en la red. El servidor de fuentes X es un pequeño demonio (daemon) que envía las fuentes a los clientes locales y a los sistemas remotos.

Algunas distribuciones de GNU/Linux utilizan exclusivamente *xfs*, sin generar una lista de fuentes en la *FontPath* de la *xorg.conf*. Para incluir en la ruta de fuente de su sistema *xfs*, agregar una directiva *FontPath* como esta:

```

Section "Files"

RgbPath "/usr/share/X11/fonts/rgb"
FontPath "unix/-1"
EndSection

```

Si instala *xfs* desde un paquete de su distribución, es probable que sea automáticamente configurado para iniciarse al arrancar y para funcionar continuamente sirviendo fuentes a los clientes locales y remotos. Para iniciar *xfs* manualmente, simplemente escriba el comando *xfs*. Por razones de seguridad, es posible que desee ejecutar *xfs* como un usuario no *root*.

xfs está configurado en el fichero de configuración, */etc/X11/fs/config* y se inicia normalmente durante la inicialización del sistema en */etc/rc.d/init.d/xfs*. Este script puede ser utilizado además para parar y/o reiniciar el servidor de fuentes X.

En Xfree86: El servidor de fuentes X proporciona al servidor X el trazado de las fuentes. Aunque normalmente esto siempre ha estado unido a la aplicación del servidor X, desde RedHat 6.0 se ha independizado del servidor y se ejecuta por separado. El fichero XF86config también tiene una sección que identifica la localización de las fuentes en el sistema, esta sección, igual que para Xorg, se llama *File*.

Consideremos el siguiente ejemplo de un fichero XF86config file:

```
# Multiple FontPath entries are allowed (they are concatenated together)

# By default, Red Hat 6.0 and later now use a font server

# independent of the X server to render fonts.

FontPath "/usr/X11R6/lib/X11/fonts/TrueType"

FontPath "unix/:7100"
```

Estas líneas indican, por una parte, la ruta a las fuentes *TrueType* y por otra, se especifica una conexión con un servidor de fuentes. Las especificaciones del servidor de fuentes utilizan la sintaxis: *<trans>/<hostname>:<port-number>* donde *<trans>* es el tipo de transporte utilizado para conectar con el servidor de fuentes (*unix* para Unix-domain sockets o *tcp* para una conexión TCP/IP), *<hostname>* es el nombre de la máquina que ejecuta el servidor de fuentes, y *<port-number>* es el número de puerto donde el servidor de fuentes escucha las peticiones (normalmente 7100).

Cuando no se especifica la entrada *FontPath* en el fichero XF86Config, el servidor retorna al modo histórico de fuentes precompiladas:

```
/usr/X11R6/lib/X11/fonts/misc/
/usr/X11R6/lib/X11/fonts/Speedo/
/usr/X11R6/lib/X11/fonts/Type1/
/usr/X11R6/lib/X11/fonts/CID/
/usr/X11R6/lib/X11/fonts/75dpi/
/usr/X11R6/lib/X11/fonts/100dpi/
```

Las rutas de fuentes que se detecten como incorrectas se eliminarán de la lista de rutas durante la inicialización del servidor. Si hubiese directorios adicionales de fuentes instaladas deben añadirse a la lista mediante entradas *FontPath*. No obstante, el fichero XF86Config es utilizado por el servidor X y no por el servidor de fuentes X. El servidor de fuentes X, conocido normalmente como *xfs*, se arranca durante la inicialización del sistema y lee su propio fichero de configuración que define su modo de operación independientemente del servidor X. Este fichero se encuentra en */etc/X11/fs/config*. A continuación tenemos un ejemplo de un fichero de configuración XFS:

```
#  
  
# Default font server configuration file for Red Hat Linux  
  
#  
  
# allow a max of 10 clients to connect to this font server  
# client-limit = 10  
  
# when a font server reaches its limit, start up a new one  
# clone-self = on  
  
# alternate font servers for clients to use  
# alternate-servers = foo:7101,bar:7102  
  
# where to look for fonts  
  
#  
  
catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,  
/usr/X11R6/lib/X11/fonts/75dpi:unscaled,  
/usr/X11R6/lib/X11/fonts/misc,  
/usr/X11R6/lib/X11/fonts/Type1,  
/usr/X11R6/lib/X11/fonts/Speedo,  
/usr/X11R6/lib/X11/fonts/75dpi,  
/usr/share/fonts/default/Type1  
  
# in 12 points, decipoints  
  
default-point-size = 120  
  
# 100 x 100 and 75 x 75  
  
default-resolutions = 75,75,100,100  
  
# use lazy loading on 16 bit (usually Asian) fonts  
  
deferglyphs = 16  
  
# how to log errors  
  
use-syslog = on  
  
# don't listen to TCP ports by default for security reasons
```

```
no-listen = tcp
```

Particularmente interesante es la palabra clave *catalogue*, ya que es esencialmente equivalente a la palabra clave *FontPath* de XF86Config. Por tanto, si se hacen cambios en la directiva *FontPath* de XF8Cconfig, también habría que hacer los mismos cambios en *catalogue* de /etc/X11/xfs/config para que el servidor de fuentes se entere.

106.1.7. Control del servidor

Muchas aplicaciones X están programadas para que, en lugar de tener una utilidad de configuración integrada, examinen el contenido de un archivo en el directorio *home* del usuario llamado *.Xresources*.

El servidor X pueden proporcionar determinadas configuraciones cuando arranquen los clientes, utilizando el fichero *.Xresources*. El fichero *.Xresources* no se crea automáticamente, hay una configuración por defecto válida para todo el sistema. Este fichero contiene directivas que se aplicarán al cliente X cuando se inicie. El siguiente ejemplo de un fichero *.Xresources* indica como se mostrara el color en un *xterm*.

```
xterm_color*background: Black  
xterm_color*foreground: Wheat  
xterm_color*cursorColor: Orchid  
xterm_color*reverseVideo: false  
xterm_color*scrollBar: true  
xterm_color*saveLines: 5000  
xterm_color*reverseWrap: true  
xterm_color*font: fixed  
xterm_color.geometry: 80x25+20+20  
xterm_color*fullCursor: true  
xterm_color*scrollTtyOutput: off  
xterm_color*scrollKey: on  
xterm_color*VT100.Translations: #override\n\  
<KeyPress>Prior : scroll-back(1,page)\n\  
<KeyPress>Next : scroll-forw(1,page)  
xterm_color"titleBar: false
```

Cada una de estas directivas es una directiva del sistema X Window que describe como se visualizará el cliente. Cada línea consiste de un nombre de cliente seguido por un asterisco y del parámetro X.

A través de un fichero *.Xresources* cuidadosamente ensamblado, se puede manipular y definir el aspecto que presentará cada aplicación al iniciarse.

106.1.8. Comprobar la configuración de X

Una vez terminada la configuración de X, es el momento de probar el servidor e iniciararlo. Para ello, obviamente, el sistema no se debe encontrar en un nivel de ejecución de modo gráfico. Se puede pasar a modo texto con *init 2 o 3* (según su distribución) y se prueba mediante:

```
$ X -probeonly
```

```
X Window System Version 7.2.0
```

```
Release Date : TRue Jan 22 17:08:26 UTC 2008-05-31
```

```
X Protocol Version 11, Revision 0, Release 7.2
```

```
Build Operating System: openSUSE SUSE LINUX
```

```
Current Operating System : Linux opensuse 2.6.22.17-0.1-default #1
```

```
SMP 2008/02/10 20:01:04 UTC i686
```

```
Build Date: 22 Junuary 2008
```

```
Before reporting problems, check http://wiki.x.org
```

```
To make sure that you have the lastest version.
```

```
Module Loader present
```

```
Markers: (--) probed, (**) from config file, (==) default setting,
```

```
(++) from command line, (!!) notice, (II) informational,
```

```
(WW) warning, (EE) error, (NI) not implemented, (??) unknown.
```

```
(==) Log file: "/var/log/Xorg.0.log", Time: Sat May 31 10:16:06 2008
```

```
(==) Using config file: "/etc/X11/xorg.conf"
```

```
(II) Module already build-in
```

```
No se ha detectado ningún error. De haber habido uno, le saldría esto:
```

```
(EE) No drivers available.
```

```
Fatal server error:
```

```
no screens found
```

En este caso:

- el fichero */etc/X11/xorg.conf* contiene un error de sintaxis, o
- un periférico se ha configurado mal: pantalla inexistente, no soporta la resolución indicada,

driver incorrecto, etc.

Si el motivo del error fuese una configuración erronea, tendríamos que analizar el registro de sucesos del servidor X que se encuentra en el fichero `/var/log/Xorg.0.log`, para resolver este problema. Las entradas en el registro contienen todos los detalles de la carga y de la configuración de X Window y son, a menudo, muy largas. En caso de error, las últimas líneas indican dónde se encuentra el problema. Si X funciona pero no reacciona como debería, se tendría que analizar toda la información de las trazas.

Solucionados los problemas, si los hubiera, deberíamos probar el servidor X tecleando el comando `X` (no tiene porque ser root) de la siguiente manera:

`$ X`

Si `X` funciona, debe aparecer una pantalla gris, en realidad una sucesión de puntos negros y blancos, y una cruz en medio que es el cursor del ratón. Mueva el ratón. Si funciona, es que se ha configurado bien su servidor. Luego detenga el servidor X con `[Ctrl][Alt][Retroceso]`, porque sin gestor de ventanas no podremos trabajar.

106.1 EXTRAS

106.1 EXTRAS HAL/Udev

Las últimas versiones de las distribuciones más habituales ya no usan `xorg.conf`. Éste fue sustituido incorporando en el servidor X la recepción de la información de la HAL (Hardware Abstraction Layer). Permitía a las aplicaciones de escritorio detectar y usar el hardware a través de una API simple y portable, sin importar el hardware sobre el que se estuviera ejecutando. De hecho HAL fue muy usado por KDE o GNOME para automontar dvds, camaras o memorias usb.

Para forzar la configuración hardware se usan archivos `fdi` que se pueden localizar (si los hubiere) en `/etc/hal/fdi/policy`. En el directorio `/usr/share/hal/fdi/policy/` hay "ejemplos" de archivos `fdi`.

Esos archivos están escritos en xml y son hasta fáciles de entender. En cualquier caso HAL es un proyecto discontinuado y su funcionalidad se ha pasado a udev.

udev sustituye a devfs en la gestión de los dispositivos /dev añadiendo nuevas características muy interesantes que se aplican también a los dispositivos gráficos y de entrada/salida.

- **udev mantiene en /dev sólo las entradas correspondientes a los dispositivos que hay conectados** al sistema. Así se soluciona el problema del /dev superpoblado.
- **No se usa el número mayor y menor para reconocer a cada dispositivo.** Puede funcionar incluso aunque estén elegidos al azar. Por tanto, no le afecta el que se acaben las combinaciones mayor/menor asignables.
- **Permite dar un nombre fijo para cada dispositivo**, por ejemplo cámara, sin que éste dependa de qué otros dispositivos hay conectados ni del orden en que se han conectado. Un disco duro, por ejemplo, se reconoce por el identificador de su sistema de ficheros, el nombre del disco, y el conector físico en el que está.
- **Avisa mediante mensajes D-BUS para que cualquier programa del espacio de usuario pueda enterarse cuando un dispositivo se conecta o desconecta** (esto es útil para HAL). También permite a los programas consultar la lista de dispositivos conectados y la forma de acceder a cada uno.
- **udev hace que toda la política de nombres de dispositivo esté en espacio de usuario, y**

no en el kernel. Esto hace posible que un programa cualquiera pueda decidir el nombre de un dispositivo, por ejemplo basándose en sus características. No es necesario modificar el kernel si no se está conforme el nombre que se le da a un dispositivo. Otra ventaja de tener el demonio de udev (udevd) en espacio de usuario es que esta memoria se puede llevar a disco (al espacio de intercambio), a diferencia de la memoria de kernel. Esto lo hace apropiado para sistemas embebidos con poca memoria física.

- udev **respeta la forma de nombrar dispositivos definida en el LSB**, aunque permite que los usuarios usen otros nombres.
- El proceso (udevd) **ocupa poca memoria y no necesita ejecutarse siempre**. Esto también favorece a los sistemas embebidos y equipos poco potentes.

udev se configura mediante una serie de reglas que se pueden encontrar en /etc/udev/rules.d con la terminación .rules. Los archivos rules son analizados en busca de reglas que sean de aplicación al hardware del sistema. Esto se hace por orden alfabético. Las líneas que comienzan con un # son comentarios y el sistema las ignora. También ignora las líneas en blanco. Cualquier otra línea es una regla. Las reglas deben ocupar una sola línea.

Al encontrar una regla que coincide con el hardware presente, udev hace lo que indica esa regla y sigue buscando otras reglas que puedan ser de aplicación.

Sintaxis:

Las reglas se componen de una serie de pares clave/valor, separados por comas. Las claves pueden ser de coincidencia y de asignación. Se distinguen porque las de coincidencia llevan dos signos igual (==) mientras que las de asignación llevan un solo signo igual (=). Cada regla debe tener al menos una coincidencia y una asignación. Cuando los valores de todas las claves de coincidencia coinciden con un dispositivo, se invocan todas las acciones especificadas por las claves de asignación.

Un ejemplo de regla bien sencillo:

```
KERNEL=="hdb", NAME="disco_extra"
```

Lo que la regla señala es que cuando el kernel encuentre un dispositivo al que corresponda "hdb", le asigne un nombre "disco_extra", en vez del "hdb" que el kernel le iba a dar por defecto.

Reglas básicas:

udev proporciona varias claves de coincidencia. Las más usadas son:

- KERNEL - compara con el nombre que el kernel da al dispositivo, por ejemplo, sda.
- SUBSYSTEM - compara con el subsistema en el que integra el dispositivo, por ejemplo, usb o ata.
- DRIVER - compara con el nombre del driver que gobierna el dispositivo, por ejemplo, usb-storage

Las claves de asignación más usadas son:

NAME - el nombre que se usará para crear el nodo. Por ejemplo, si usamos NAME=="rdppk", udev creará un nodo llamado /dev/rdpk

SYMLINK - permite especificar el enlace simbólico que se creará.

También en este caso, existen más claves de asignación que estas dos. La lista completa está en la página man de udev.

En el caso típico, nuestras reglas no tendrán ninguna asignación NAME. De este modo el nodo se crea siempre con el nombre del kernel por defecto. Lo que interesa es crear una asignación SYMLINK="pendrive" o SYMLINK="camara". Por ejemplo:

```
KERNEL=="sdb", SYMLINK="pendrive"
```

En realidad, esta regla no es muy útil, ya que eso sólo nos deja acceder al pendrive en sí. Cosa que puede ser útil para particionarlo pero para nada más. Lo que queremos es acceder a las particiones que pueda tener el pendrive/disco. Lo conseguiremos con una regla como ésta:

```
KERNEL=="sdb", SYMLINK+="pendrive%n"
```

Nótese el símbolo + delante del igual, indicando que lo que sigue es una lista de enlaces, no uno solo. Nótese también el "%n" añadido después de "pendrive". Esta regla creará una serie de enlaces simbólicos sustituyendo automáticamente "%n" por el número de partición. Por tanto, si el pendrive tenía tres particiones, aparecerán cuatro enlaces simbólicos: /dev/pendrive, /dev/pendrive1, /dev/pendrive2 y /dev/pendrive3, que apuntarán respectivamente a /dev/sdb, /dev/sdb1, /dev/sdb2 y /dev/sdb3

Información de /sys:

Las claves de coincidencia analizadas hasta ahora (KERNEL, SUBSYSTEM y DRIVER) se quedan un poco cortas. Para tener un control más fino son necesarias claves que permitan identificar dispositivos usando códigos de vendedor, número de serie, etc. De esta manera, incluso con dos discos exactamente iguales será posible distinguir uno del otro.

La mayoría de los drivers exportan esta información usando sysfs, lo que significa que se puede acceder a ella en algún lugar dentro de la carpeta /sys. La clave de coincidencia que sirve para comparar contra cualquiera de los atributos presentes en /sys es ATTR{atributo}. Por ejemplo, ATTR{size}=="398297088" coincidirá con un dispositivo de ese tamaño. Para saber cuál es el size de nuestro disco duro, hacemos:

```
[root@instructor ~]# udevadm info --query=all --path=/sys/block/sda/sda1/ | grep -i size
E: UDISKS_PARTITION_SIZE= "398297088"
```

Y con esto ya es viable qué tamaño indicar en la regla, que podría quedar como:

```
KERNEL=="sda1", ATTR{size}=="398297088", SYMLINK+="disco_duro%n"
```

106.1 EXTRAS MIR y Wayland

Las X window han quedado anticuadas y tienen unas capacidades innumerables pero que rara vez se llegan a usar. Es por ello que surgió Wayland como un protocolo de servidor gráfico y una biblioteca para Linux que implementa este protocolo, según se indica en la wikipedia.

Wayland proporciona un método para que los gestores de composición de ventanas se comuniquen directamente con las aplicaciones y el hardware de vídeo. Se espera que también sea posible la comunicación con hardware de entrada usando otras bibliotecas. Las aplicaciones renderizan los gráficos en sus propios buffers y el gestor de ventanas se convierte en el servidor gráfico, haciendo una composición con esos buffers para formar la visualización en pantalla de las ventanas de las aplicaciones. Este es un enfoque más simple y más eficiente que usar un gestor de composición de ventanas con el X Window System.

Wayland consiste en un protocolo (en gran parte completo) y una implementación de referencia llamada Weston. Para el renderizado, Weston puede usar OpenGL ES o software (la biblioteca pixman). En la actualidad, los clientes se limitan a OpenGL ES en lugar de OpenGL completo porque "libGL utiliza GLX y todas las dependencias de X". El proyecto también está desarrollando versiones de GTK+ y Qt que renderizan hacia Wayland en vez de a X. Se espera la mayoría de las aplicaciones obtengan soporte para Wayland a través de una de estas bibliotecas sin modificar la aplicación. Weston, el compositor Wayland de referencia, solo trabaja con el núcleo Linux debido a su dependencia de las características específicas de Linux como udev.

Canonical, desarrollador de ubuntu, ha optado por no apoyar el proyecto wayland y crear su propio servidor gráfico llamado Mir. Este correrá también en dispositivos de telefonía que implementen ubuntu. En cualquier caso, la renovación del servidor gráfico está en marcha.

106.2. Configuración del gestor de login gráfico.

Peso en el examen de certificación: 2 puntos.

Objetivo: Inicializar y personalizar un gestor de pantalla (display manager). Este objetivo contempla los gestores de pantalla XDM (X Display Manager), GDM (Gnome Display Manager) y KDM (KDE Display Manager).

Conceptos y áreas de conocimiento:

- Activar o desactivar el gestor de pantalla.
- Cambiar el mensaje de inicio del gestor de pantalla.
- Cambiar la profundidad de color por defecto del gestor de pantalla.
- Configurar el gestor de ventanas para usarlo en terminales X.

Términos y utilidades:

- /etc/inittab
- ficheros de configuración xdm.
- ficheros de configuración kdm.
- ficheros de configuración gdm.

106.2.1. Introducción.

El Display Manager, o gestor de ventanas o gestor de login gráfico, es un elemento de X Window que se encarga de realizar la conexión de usuarios, tanto locales como remotos, su autenticación, y la carga de componentes en su entorno de trabajo, al abrir una nueva sesión.

El gestor de login gráfico representa más o menos el equivalente gráfico de los servicios propuestos por *init*, *getty* y *login*: pide identificadores y contraseñas, para autenticar a los usuarios y poder abrir una sesión. Es posible abrir varias sesiones X desde un solo gestor de login gráfico, incluso en una misma máquina.

El gestor de login gráfico por defecto se llama *XDM*: X Display Manager. Su estilo gráfico no resulta muy atractivo, pero es ligero y funciona con todos los servidores X. En Linux además son muy populares:

- GDM: versión del entorno de escritorio GNOME.
- KDM: versión del entorno de escritorio KDE.

Las versiones *gmd* y *kgm* de GNOME y KDE han evolucionado, proponiendo lo mismo que XDM con funcionalidades adicionales.

106.2.2. Iniciar una sesión gráfica.

Hay, principalmente, dos métodos para iniciar una sesión X. Uno es, manualmente, conectándose en un entorno de texto que, después, arranca el servidor X. Y el otro, es automáticamente, utilizando el sistema de *login gráfico de xdm*.

106.2.2.1. Arranque manual de una sesión X.

Si la sesión actual se encuentra en modo comando y tiene instalado el servidor X, puede usar el comando *startx* para iniciar una sesión del servidor X y el entorno asociado. Este a su vez, llamará a los programas necesarios para inicializar el entorno e inicializar el servidor X.

Para tener acceso al servidor X y a los clientes locales relacionados, la ruta `/usr/X11R6/bin` ha de encontrarse en la variable `PATH` del sistema, si no estuviese habría que añadirlo en el `.profile`.

\$startx

Cuando el sistema no encuentra ningún fichero de configuración, visualizará X, una pantalla gris con un cursor de ratón que no ofrece acción alguna.

Una vez arrancado una sesión en el servidor X, mediante `startx`, se ejecuta el programa `xinit`. El programa `xinit` se usa para dos cosas: iniciar el servidor del sistema de los gestores de ventanas X y un primer programa cliente. Esto lo realiza en sistemas que no pueden arrancar X directamente a partir de `init` o en entornos que usan múltiples sistemas gestores de ventanas.

Si no se suministra un programa cliente en la línea de comandos, `xinit` buscará y ejecuta el fichero `xinitrc` situado en el directorio home del usuario. Si no existiese se ejecutaría por defecto el fichero `/etc/X11/xinit/xinitrc`.

Dado que es recomendable no tocar el fichero `/etc/X11/xinit/xinitrc` para que se mantenga fiel al estándar definido por su distribución, se podría modificar el fichero personal `.xinitrc`. Como son iguales, se puede usar el primero como modelo del segundo.

Por ejemplo, el fichero `xinitrc` podría contener lo siguiente:

```
#!/bin/sh
```

```
if [ "`whoami`" != root ]; then
    xsetroot -solid "#21449C"
fi
exec /etc/X11/Xsession $*
```

La última línea ejecuta `Xsession`, va a buscar un gestor de ventanas para iniciarla, como en el siguiente ejemplo inicia el entorno de escritorio KDE, en Mandriva mediante:

```
$ cat $HOME/.xinitrc
```

```
/usr/bin/startkde
```

O en openSUSE mediante:

```
/opt/kde3/bin/startkde
```

Y para lanzar GNOME:

```
exec gnome-session
```

El último comando en `.xinitrc` debe ser arrancado con `exec` y no debe ser enviado nunca al background. Si no se utilizase `exec` o se enviase el comando al background, el servidor X finalizaría justo después de completar la ejecución de los clientes del fichero `xinitrc`.

Algunas distribuciones hacen que el fichero `xinitrc` llame al script `Xclients`. Algunas distribuciones utilizan este fichero para chequear y saber qué gestores de ventanas están instalados y para arrancar aquel que el usuario ha configurado como favorito. Un usuario puede tener su propio fichero `Xclients` en `~/.Xclients`.

106.2.2.1.1. Arranque manual de xdm

El xdm utiliza el servidor X para correr en la pantalla local. Por lo tanto, se debe tener una configuración de X en la estación de trabajo antes de utilizar un gestor de login gráfico. Para empezar xdm, simplemente ingresar como root:

```
# xdm
```

xdm lanza el servidor X y muestra el inicio de sesión, tras lo cual se puede iniciar una sesión de forma habitual. Entonces, xdm inicia el entorno gráfico. Después de la sesión, xdm restablece y de nuevo muestra la pantalla de inicio de sesión.

La mayoría de distribuciones Linux activan las consolas virtuales. Se puede cambiar entre ellas utilizando las combinaciones de teclas Ctrl-Alt-F1, Ctrl-Alt-F2, etc. (El Ctrl sólo es necesario cuando se cambia de una consola X a una de texto o a una consola X de otros.) Por lo general, las primeras seis consolas se configuran como pantallas en modo texto, y X se lanza en la consola 7 (Ctrl-Alt-F7) o el primer TTY que no execute *mingetty* o algún otro proceso *getty*. Es decir, como con *startx*, la consola original en modo texto se mantiene sin cambios después de iniciar manualmente xdm. Por lo tanto, debe salir de su consola en modo texto para dejar el sistema sin supervisión, con xdm corriendo manualmente.

Si desea dejar xdm, primero debe asegurarse de que todas las sesiones de X se cierren correctamente, de lo contrario, podría perder datos, al cerrarse todos los procesos activos al cerrarse xdm. Simplemente debería detener todos los procesos xdm utilizando *kill* o *killall* desde una consola de texto:

```
# Killall xdm
```

Por supuesto, xdm no es muy útil para el sistema local, si siempre hay que empezar de forma manual. Es por eso que la mayoría de distribuciones Linux incluyen una opción de tiempo de arranque para empezar xdm por usted, eliminando el inicio de sesión en modo texto completo.

106.2.2.2. Arranque automático de una sesión X.

Dentro de los diversos tipos de arranques en modo gráfico, *xdm* es el más sencillo. xdm presenta una GUI que permite a una sesión gráfica comenzar en un servidor de X. xdm se distribuye como parte de Xorg y está configurado por una serie de ficheros que se encuentran en */etc/X11/xdm*.

Existen dos opciones, en la mayoría de las distribuciones, para realizar un arranque automático de xdm:

- mediante el fichero */etc/inittab*
- o iniciando xdm como un servicio.

Mediante el fichero */etc/inittab* : Esta línea en la parte inferior del fichero */etc/inittab* indica al *init* que inicie xdm para el runlevel 5

```
# Run xdm in runlevel 5
```

```
x:5:respawn:/usr/X11R6/bin/xdm -nodaemon
```

Con esta configuración, cuando el sistema entra en el nivel de ejecución 5, xdm se inicia y se presenta el inicio de sesión gráfica. Pudiera ocurrir que aún teniendo el fichero *inittab*, tal y como hemos descrito, el sistema continué accediendo en modo consola. Esto puede ser debido a que X este mal configurado, en cuyo caso, la pantalla parpadearía varias veces antes de ponerse en modo consola, o esta en el nivel de ejecución (runlevel) incorrecto. En este último caso, se comprueba y modifica la línea siguiente para iniciar la sesión en nivel 5 (para las distribuciones rpm) por defecto:

```
id:5:initdefault:
```

y pase como root al nivel de ejecución 5:

```
# telinit 5
```

También es posible que se inicie automáticamente xdm simplemente agregándolo al final de una script de inicialización, como *rc.local*. Este método ofrece menos control sobre xdm pero puede ser adecuada para algunas situaciones y para las distribuciones de Linux que no ofrecen los niveles de ejecución.

Iniciando xdm como un servicio: Para iniciar xdm como servicio deberíamos verificar la siguiente línea en el fichero */etc/init.d*:

```
$ ls /etc/init.d/xdm
```

```
/etc/init.d/xdm
```

Podemos verificar la configuración del nivel asociado mediante *rcupdate.d* o *chkconfig*, o manualmente leyendo el fichero de configuración del nivel:

```
# chkconfig --list xdm
```

```
xdm0:off 1:off 2:off 3:off 4:off 5:on 6:off
```

```
# ls -l /etc/rc.d/rc?.d/S*xdm
```

```
lrwxrwxrwx 1 root root 6 may 14 12:22 /etc/rc.d/rc5.d/S10xdm ->../xdm
```

En varias distribuciones, se colocan la elección y los ajustes por defecto de varios gestores de ventanas en ficheros de configuración dentro de la carpeta */etc/sysconfig*. En Mandriva, por ejemplo, se trata del fichero */etc/sysconfig/desktop*.

```
$ cat /etc/sysconfig/desktop
```

```
DISPLAYMANAGER=xdm
```

En openSUSE, sin embargo, los ajustes son más precisos (concretamente para la gestión de la resolución y de los temas) y el fichero de configuración es */etc/sysconfig/displaymanager*.

106.2.2.3. Configuración de xdm.

El primer fichero cargado por *xdm* es *xdm-config*, el fichero de configuración principal de *xdm*. *xdm-config* puede ser difícil y complicado de configurar, puesto que tiene varios ficheros asociados que deben ser modificados. Este fichero establece la operativa básica de xdm, que para la mayoría de instalaciones no deberá ser cambiado nunca, pero si fuese necesaria la configuración del sistema para permitir logins remotos, se debería realizar algún que otro ajuste, en ella, de forma manual.

El fichero de configuración *xdm-config* determina la configuración del login gráfico, mediante un conjunto de entradas que, especifican la utilización de otros scripts.

Sus entradas son del tipo: *DisplayManager.recurso: valor*

<i>DisplayManager.recurso</i>	<i>Valor</i>
DisplayManager.*.setup	/etc/X11/xdm/Xsetup

DisplayManager.*.chooser	/etc/X11/xdm/RunChooser
DisplayManager.*.startup	/etc/X11/xdm/Xstartup
DisplayManager.*.session	/etc/X11/xdm/Xsession
DisplayManager.*.reset	/etc/X11/xdm/Xreset
DisplayManager*resources	/etc/X11/xdm/Xresources
DisplayManager.servers	/etc/X11/xdm/Xservers
DisplayManager.accessFile	/etc/X11/xdm/Xaccess

El fichero *Xsetup* de */etc/X11/xdm/* o cualquier otro fichero asociado a la entrada *DisplayManager.*.setup* del fichero *xdm-config*, se ejecuta incluso antes de que el sistema muestre la ventana de login. En este fichero se determina la presentación gráfica de XDM. El contenido por defecto de este fichero depende del editor de la distribución de Linux, pero en general todos ellos lo usan para:

- modificar el fondo de pantalla con el comando *xsetroot*
- modificar los colores, con *xrdb*
- cambiar con los recursos X, la geometría (las posiciones y tamaños) de los cuadros de diálogo
- visualizar una consola con *xconsole*
- activar el teclado numérico con *numlockx*
- modificar el teclado con *xmodmap*

El fichero *RunChooser* de */etc/X11/xdm/* o cualquier otro fichero asociado con la entrada *DisplayManager.*.chooser* del fichero *xdm-config*, permite visualizar el cuadro de diálogo de bienvenida de los servidores X remotos para conectarse a ellos. Se parece al fichero *Xsetup*, pero se limita a la configuración y al inicio del programa *chooser* (*/usr/X11R6/bin/chooser*, */usr/lib/X11/chooser* o cualquier otro programa que tenga una función similar). Este programa tiene interés únicamente si el servidor X remoto acepta las conexiones mediante el protocolo XDMCP.

Tras la autenticación del usuario, el sistema busca y ejecuta, con privilegios de root, el fichero *Xstartup* en */etc/X11/xdm/* o cualquier otro fichero asociado con la entrada *DisplayManager.*.startup* del fichero *xdm-config*. Sirve, entre otras cosas, para:

- borrar la pantalla
- escribir la información de conexión en los ficheros */var/log* adecuados
- comprobar si la conexión es local o remota
- comprobar si se autoriza al usuario a conectarse

Tras la autenticación del usuario y con el conjunto de privilegios del usuario (no hace falta tener las del *root*), se inicia el fichero *Xsession* desde */etc/X11/xdm/* o cualquier otro fichero asociado con la entrada *DisplayManager.*.session* del fichero *xdm-config*. El sistema comprobará la instalación del servidor X; si no está instalado, ejecutará un comando *Xterm* y lo asociará el proceso de sesión del usuario. De hecho, si una sesión de usuario no se pudiera iniciar con normalidad, este fichero contiene lo necesario para iniciar una consola. *Xsession* sirve, entre otras cosas, para:

- Recoger los errores durante la inicialización de la sesión y redirigirlo al fichero \$

{HOME}/.xsession-errors.

- Cargar variables.
- Cargar los mapas de teclado correspondientes.
- Y averiguar que escritorio debe proporcionar a cada usuario para su sesión (KDE, Gnome ...)

Después de una configuración básica, Xsession intenta iniciar en este orden los siguientes ficheros:

- el fichero \$HOME/.xsession, propio del usuario
- si no lo encuentra, el fichero \$HOME/.xinitrc del usuario
- si no lo encuentra, el fichero /etc/X11/xdm/sys.xsession
- si no lo encuentra, el fichero /etc/X11/xinit/xinitrc

Mediante una bandera (o flag) se puede forzar un arranque por defecto e ir directamente a la carga de un entorno de escritorio, también llamado sWindow Manager, como KDE o GNOME.

Cuando un usuario cierra la sesión, el sistema inicia el fichero Xreset desde /etc/X11/xdm/ o cualquier otro fichero asociado con la entrada *DisplayManager.*.reset* del fichero *xdm-config*. En el registro del sistema encontrará entradas procedentes de este fichero.

El fichero Xresources de /etc/X11/xdm/ o cualquier otro fichero asociado con la entrada *DisplayManager*resources* del ficheros *xdm-config*. define los recursos que permiten personalizar la pantalla de login para xdm. Se usa el formato estándar del .Xresource, visto anteriormente, para cambiar colores, tipos de fuente y demás ajustes de la pantalla de login. En particular, en este fichero se puede modificar el mensaje de bienvenida y el de error:

```
xlogin*titleMessage:Xlogin  
xlogin*greetColor:darkgray  
xlogin*promptColor: darkgray  
xlogin*failColor:red  
xlogin*greeting:Bienvenido/a a CLIENTHOST  
xlogin*fail:-- Conexión denegada -
```

Con una configuración como la anterior, el sistema recibirá al usuario con el mensaje "Bienvenido/a a HOSTNAME" visualizado en gris oscuro. Si no se autentica, se visualiza en rojo el mensaje "Conexión denegada".

El fichero Xservers de /etc/X11/xdm/ o cualquier otro fichero asociado con la entrada *DisplayManager.servers* del ficheros *xdm-config*, proporciona la lista de las especificaciones de los servidores locales de X o, más bien, de los que no necesitan conexión mediante XDMCP. Este archivo asocia el nombre de visualización X (:0, 1, ...) con el servidor X local o con una terminal de X.

El fichero Xaccess de /etc/X11/xdm/ o cualquier otro fichero asociado con la entrada *DisplayManager.accessFile* del ficheros *xdm-config*, proporciona la lista de los hosts anfitriones autorizados a conectarse mediante XDMCP a su servidor X. Por ejemplo, si deseáramos permitir logins remotos a través de xdm, deberíamos cambiar la siguiente línea:

DisplayManager.requestPort:0 Y ponerla como un simple comentario mediante (!) para permitir a xdm escuchar las consultas recibidas: !DisplayManager.requestPort:0

106.2.2.4. ARRANQUE AUTOMATICO DE UNA SESIÓN KDM O GDM

kgdm es la versión del entorno de escritorio KDE del *xdm*, y el *gdm* la versión de Gnome, con un aspecto mejorado y con más opciones. Excepto para todos los recursos gráficos propios, *gdm* y *kdm* emplean (aunque no sea una obligación) los mismos ficheros que *xdm*: *Xsetup*, *Xstartup*, *Xsession*, *Xreset*, *Xresources*, *Xaccess*, etc.

Depende de las distribuciones la manera de escoger qué gestor de ventanas utilizar en el arranque: en Mandriva, por ejemplo, hay que editar el fichero */etc/inittab*, buscar la cadena *xdm* y reemplazarla con *kdm* o *gdm*. En Debian, sin embargo, hay que editar el fichero */etc/X11/default-display-manager* y poner el gestor deseado.

La configuración avanzada de *gdm* y *kdm* configura de forma automática los ficheros comunes de *xdm*, permitiendo configurar manualmente aquéllos ajustes propios del gestor, ya sea mediante ficheros de texto o, si las modificaciones son sencillas, pasando por la interfaz gráfica asociada.

- La configuración de KDM se guarda en */etc/X11/kdm/kdmrc* o en *<prefix-kde>/share/config/kdmrc*, por ejemplo en openSUSE el fichero se coloca en */opt/kde3/share/config/kdm/kdmrc*.
- La configuración de GDM, sin embargo, se guarda en */etc/X11/gdm/gdm.conf*, o */etc/gdm/gdm.conf* o también en */usr/share/gdm/gdm.conf*.

Aunque *gdm* y *kdm* utilizan por defecto las opciones de los ficheros de *xdm*, se puede modificar la configuración de manera que no se haga uso de ellas. Por ejemplo, en *kdm*, se puede activar el protocolo XDMCP así:

[Xdmcp]

Enable=true

Port=177

KeyFile=/opt/kde3/share/config/kdm/kdmkeys

Xaccess=/etc/X11/xdm/Xaccess

También podemos autorizar la exportación de las ventanas X hacia su servidor X (visualización remota) mediante el comando *xhost+* suprimiendo el parámetro *nolisten* de la configuración.

ServerArgsLocal=-nolisten tcp

En **gdm** ejecute, desde la línea de comandos, *gdmsetup*, que inicia la ventana de configuraciones.

106.2.3. Utilización del entorno X.

Hay disponible una amplia variedad de gestores de ventanas, unos más conocidos que otros. Muchos utilizan un determinado gestor de ventanas, mientras que otros utilizan un entorno completo de escritorio como KDE o GNOME, que combina un gestor de ventanas junto con una serie de aplicaciones y herramientas integradas en el mismo.

El gestor de ventanas se arranca normalmente desde el fichero *Xclient*, *Xsession*, o *.xinitrc*, dependiendo de la distribución y de la elección del usuario. Mandriva utiliza el fichero *Xclient* mientras que Debian usa un fichero *Xsession* genérico para los usuarios locales y remotos. Para ejecutar un gestor de ventanas distinto a los incorporados en esos ficheros se puede especificar manualmente en el fichero *.xinitrc* dentro del directorio *home*. Este script se ejecutará cada vez que

se arranque el proceso *xinit*.

Una vez elegido el entorno X podemos personalizar el sistema, por ejemplo, podemos utilizar temas para cambiar el aspecto general del sistema.

106.2.3.1. Personalización de xdm.

Podemos personalizar el aspecto de la pantalla gráfica de login mediante la manipulación de los recursos en */etc/X11/xdm/Xresources*. Por ejemplo, el siguientes script muestra los ajustes para controlar el saludo de bienvenida, mensajes de error de login y los colores:

```
xlogin*borderWidth: 10  
  
xlogin*greeting: Welcome to Linux on CLIENTHOST  
  
xlogin*namePrompt: Login:\040  
  
xlogin*fail: Login incorrect - try again!  
  
xlogin*failColor: red  
  
xlogin*Foreground: Yellow  
  
xlogin*Background: MidnightBlue
```

También se puede incluir opciones en la línea de comandos del servidor X en */etc/X11/xdm/Xservers* para sustituir los que se encuentran en */etc/X11/xorg.conf*.

Para incluir aplicaciones X adicionales o realizar ajustes en la pantalla de login gráfica, se utiliza el fichero */etc/X11/xdm/Xsetup_0*. Por ejemplo, para modificar color de fondo de la pantalla X y poner un color sólido (en formato hexadecimal) y añadir un reloj en la esquina de la pantalla, se haría el ajuste siguiente en el fichero *xsetup_0*:

```
#!/bin/sh  
  
# Xsetup  
/usr/X11R6/bin/xsetroot -solid "#356390"  
  
/usr/X11R6/bin/xclock -digital -update 1 -geometry -5-5 &
```

106.2.3.2. Personalización de kdm.

KDM es el gestor de ventanas del entorno de escritorio KDE y es responsable de la pantalla de acceso gráfico que se encarga de la autenticación de usuarios y de iniciar una sesión de usuario. KDM es distribuido por KDE.org y está configurado por una serie de archivos que se encuentran en */etc/X11/kdm*. Para ver la versión que está disponible para instalar, usar el gestor de paquetes *yum*, y luego instalarlo ejecutaríamos los siguientes comandos:

```
# yum info kdm  
Loaded plugins: refresh-packagekit  
  
Available Packages  
  
Name : kdm
```

Arch : i586
Version : 4.2.4
Release : 5.fc11
Size : 1.5 M
Repo : updates
Summary : The KDE login manager
URL : <http://www.kde.org/>
License : GPLv2
Description: KDM provides the graphical login screen, shown shortly after boot : up, log out, and when user switching.
Use the *yum* package manager to install the *KDM* interface.

yum install kdm

En este proceso de instalación, además del *kdm*, también instalará el entorno de escritorio KDE. El fichero de configuración principal, para controlar la forma en que funciona *kdm*, se llama *kmrc*, y se encuentra en */etc/kde/kdm*.

106.2.3.3. Personalización de gdm.

gdm es el gestor de ventanas para el entorno de escritorio GNOME. GNOME es el entorno por defecto de escritorio gráfico de Fedora y Ubuntu. El *gdm* será cargado automáticamente durante la instalación gráfica de estos sistemas operativos. Si se necesita instalar el entorno GNOME, se puede hacer tecleando:

yum groupinstall "GNOME Desktop Environment"

El archivo de configuración principal de *gdm* es *gdm.conf* o *custom.conf*, según sobre la distribución de Linux. El archivo de configuración se encuentra en */etc/gdm/gdm.conf*. Este archivo contiene secciones para configurar la forma en que funciona el proceso de inicio de sesión, los entornos de la sesión, y la apariencia del administrador o la "bienvenida" que se le presentan al usuario en la pantalla de inicio de sesión.

106.2.3.4. Cambiar de entorno de escritorio.

Si están instalados, tanto el entorno KDE como el GNOME, puede cambiar entre ellos durante el inicio de la sesión gráfica, seleccionando el entorno a usar en cada momento, mediante el menú de sesión inicial. Tanto el *kdm* y como el *gdm* proporcionan dicho menú de sesión. Es posible ejecutar sólo uno de los gestores de ventanas, cambiando el valor predeterminado. Para ello se edita el fichero */etc/sysconfig/desktop* y se realiza, por ejemplo, la siguiente modificación:

desktop= "kde"

displaymanager= "kdm"

Otra forma de cambiar entre los gestores de *kdm* y *gdm*, es la instalación de la herramienta *switchdesk* utilizando un gestor de paquetes y luego ejecutar la aplicación. *Switchdesk* permite al usuario cambiar sólo entre los entornos de escritorio instalados en el sistema. No todos los gestores de visualización son compatibles, sin embargo, es compatible con KDE y GNOME:

```
$ switchdesk kde
Red Hat Linux switchdesk 4.0

Copyright (C) 1999-2004 Red Hat, Inc

Redistributable under the terms of the GNU General Public License

Desktop now set up to run KDE.
```

106.2.4. Terminales X.

Las Terminales X son un tipo de dispositivos de visualización de bajo costo para X. Estos dispositivos se pueden configurar para acceder a un host remoto, encontrar un demonio xdm en el "host dispuesto" a ofrecer los servicios de xdm, y ejecutar una sesión X en la pantalla de destino. Con esta configuración, un gran número de terminales X puede hacer uso, a un coste relativamente bajo, de unos pocos y poderosos sistemas host para ejecutar los clientes gráficos en ellos.

106.2.4.1. xdm para terminales X.

Para utilizar una terminal de X con su host anfitrión, xdm debe estar ejecutándose en la máquina host. Para que el host anfitrión escuche las peticiones de xdm de los terminales X, usa el protocolo *XDMCP*, (el puerto por defecto para xdmcp es 177). Cuando se recibe una solicitud, xdm responde con la misma pantalla de login gráfica que se utiliza en el sistema local. La diferencia es que el servidor X se ejecuta en el hardware del terminal de X y no en el host xdm, transmitiendo toda la información gráfica necesaria por la red.

Se puede configurar el acceso al demonio xdm de su sistema en el fichero de configuración *Xaccess* del directorio */etc/X11/xdm/*. Este fichero es una simple lista de los host que van a ser restringidos o habilitados. Para permitir el acceso a un host, simplemente escriba su nombre. Para restringir un host, escriba su nombre con un signo de exclamación (!) antes. El comodín (*) también permite manejar grupos de dispositivos. El siguiente ejemplo permite el acceso a todos los terminales X en el dominio local, pero prohíbe el acceso de xterm1 en un dominio externo:

```
*.ejemplo.com
```

```
!xterm1.otroejemplo.com
```

106.2.4.2. Emuladores de terminales X.

Para acceder a línea de comandos desde X o para ejecutar una aplicación X que necesita ejecutarse en una terminal concreta, debe utilizar un emulador de terminal. El emulador básico del servidor X se llama *xterm* y funciona sea cual sea su entorno de trabajo X, ya que, el sistema lo inicia por defecto cuando encuentra problemas con el servidor X.

\$ xterm &

Existen diferentes emuladores de terminal como para satisfacer las necesidades de todo el mundo. A continuación se indican varios de los más populares.

xterm

El emulador de terminal "estandard" xterm existe desde hace mucho tiempo. Proporciona emulaciones para aplicaciones X. Soporta un gran número de opciones de línea de comandos para configurar el emulador en tiempo de ejecución. Además, incluye soporte para las opciones de X Toolkit. Para configurar Xterm, tiene que utilizar los botones del ratón y la tecla [Ctrl]:

- [Ctrl] + botón izquierdo: opciones principales
- [Ctrl] + botón derecho: elección de las fuentes de caracteres
- [Ctrl] + botón del medio: opciones del terminal.

rxvt

Para aquellos usuarios que no necesiten toda la funcionalidad de xterm existe rxvt. Ya que fue diseñado para ser utilizado en sistemas con muchos emuladores de terminal abiertos simultáneamente, debe hacer un menor uso de memoria, y por lo tanto, el emulador rxvt emula menos sistemas que xterm y además no incluye soporte de las opciones X Toolkit.

aterm

aterm es similar a rxvt puesto que no soporta tantas emulaciones como xterm y tampoco soporta las opciones X Toolkit. Pero aterm, ofrece varias opciones que no ofrece xterm, como un modo de transparencia en el cual se ve el fondo del escritorio a través de la ventana. aterm fue diseñado para trabajar con el manejador de ventanas *AfterStep*, pero no es indispensable su utilización.

Eterm

Si buscamos el emulador de terminal de mejor aspecto y más configurable, Eterm es el que elegiremos. Fue diseñado para trabajar con el manejador de ventanas *Enlightenment*, y se nota. Necesita más memoria que otros emuladores pero admite tantas configuraciones y personalizaciones como se pueda desear. También soporta temas.

gnome-terminal

El entorno de escritorio GNOME viene con su propio emulador de terminal gnome-terminal. Está escrito utilizando las librerías GTK (GIMP Tool Kit) así que tiene el mismo aspecto y funcionamiento del tema GNOME que se tenga instalado. Es un emulador de terminal muy completo y una buena elección si se utiliza GNOME.

konsole

Para no ser menos, el grupo KDE incluye en su entorno de escritorio un emulador de terminal llamado konsole. Se trata de un emulador completo que hereda el aspecto y funcionamiento de KDE.

106.2.5. Clientes X.

Hay varios parámetros de línea de comandos utilizados por la mayor parte de las aplicaciones clientes X. Estos parámetros nos permiten establecer la configuración de inicio mediante una sintaxis estándar. Son conocidos como "X Toolkit options" (Opciones de la colección de herramientas de X), y aunque puedan ser un poco pesados para teclearlos a mano si que son muy prácticos para utilizarlos en un script o en un menú.

La siguiente tabla muestra estos parámetros:

Opción	Función
<code>-bg "color" o -background "color"</code>	Establece el color de fondo por defecto para la aplicación.
<code>-fg "color" o -foreground "color"</code>	Establece el color de texto por defecto para la aplicación.
<code>-bd "color" o -bordercolor "color"</code>	Establece el color del borde de ventana por defecto para la aplicación.
<code>-bw "número" o -borderwidth "num"</code>	Establece la anchura en pixels del borde de la ventana.
<code>-display nombrehost:display.pantalla</code>	Especifica el nombre del host y los números de <code>display</code> y pantalla en los que se mostrará la aplicación.
<code>-fn "fuente" o -font "fuente"</code>	Especifica la fuente utilizada para el texto.
<code>-geometry "anchura"x"altura"+"x"+"y"</code>	Especifica el tamaño y posición inicial de la ventana.
<code>-iconic</code>	Le indica a la aplicación que se inicie en modo ícono si fuese posible.
<code>-name "nombre"</code>	Especifica el nombre bajo el cual se encontrarán los recursos de las aplicaciones.
<code>-rv o -reverse</code>	Le indica a la aplicación que simule vídeo inverso si fuese posible.
<code>+rv</code>	Le indica a la aplicación que no intente simular vídeo inverso.

106.2.5.1. PERSONALIZACIÓN DE CLIENTES X

El entorno X es altamente personalizable. Las aplicaciones escritas para utilizar entornos de escritorio como GNOME y KDE heredan los aspectos y funcionalidades que hayan sido configurados en esos entornos. Pero hay muchas aplicaciones que no han sido escritas para un entorno de escritorio y solamente utilizan las librerías estandard *X Toolkit*, pero aun así nos proporcionan métodos para personalizar su apariencia.

Las aplicaciones que utilizan las librerías *X Toolkit* pueden personalizarse utilizando el formato *Xresource*. En */usr/X11R6/lib/X11/appdefaults* o en */etc/X11/app-defaults* podemos encontrar numerosos ficheros de ejemplo de *Xresource*. El nombre de cada uno de ellos es el de la aplicación a la que se corresponden.

Al cambiar algún ajuste en un fichero de los que se encuentran en *apps-default*, este ajuste quedará alterado para todos los usuarios del sistema. Para cambiar el comportamiento de la aplicación solo para un usuario deberíamos poner las entradas del fichero *Xresource* en otro fichero llamado *.Xdefaults* dentro del directorio *home* del usuario correspondiente.

106.2.5.2. TECLAS ESPECIALES

Hay varias combinaciones especiales de teclas que pueden utilizarse dentro de X. Los usuarios que solían cambiar de consolas virtuales en Linux con la combinación Alt-Tecla de Función se

sorprenderán al ver que esto ya no funciona en X.

La siguiente tabla nos muestra las combinaciones especiales.

Combinación	Función
Ctrl-Alt-<+ del Teclado numérico>	Cambia a una resolución de mayor vídeo si estuviese configurada.
Ctrl-Alt-<- del Teclado numérico>	Cambia a una resolución menor de vídeo si estuviese configurada.
Ctrl-Alt-Retroceso	Salida rápida de X a no ser que esta opción estuviese desactivada en XF86Config.
Ctrl-Alt-<F1 hasta F6>	Cambia a las consolas de texto.
Ctrl-Alt-<F7>	Cambia de nuevo al modo gráfico desde una consola de texto.

106.2.5.3. CONFIGURACIÓN DE LA SEGURIDAD DE X

X soporta varios mecanismos de autentificación, algunos muy complejos. Para la mayoría de los usuarios el método básico basado en host funciona bien y es fácil de configurar y manejar. La seguridad basada en el host se configura con el comando *xhost*.

El comando *xhost* se usa para decirle al servidor X que permita a las aplicaciones remotas de otro ordenador acceder a su pantalla, teclado, y mouse. Como no hay distinción de quién puede usar el servidor X en el ordenador remoto, el método *xhost* no es una manera muy segura de habilitar el acceso remoto. Es mejor usar *xauth* o *ssh*.

Se pueden consultar los ajustes de seguridad actuales ejecutando *xhost* sin opciones.

```
# xhost  
access control enabled, only authorized clients can connect  
INET:host1  
INET:host2
```

También podemos utilizar el comando *xhost* para ver si disponemos de permisos para mostrar clientes en un sistema remoto y de esta manera que las aplicaciones X sepan donde mostrar su salida. Esto se consigue cambiando el valor de la variable DISPLAY al del host remoto y después ejecutando *xhost*. Por ejemplo para ver si podemos mostrar aplicaciones en el host host2.the-nashes.net escribiríamos:

```
export DISPLAY=host2.enterprise.net:0.0  
xhost
```

El comando *xhost* emplea solo unas pocas opciones de línea de comandos. Para habilitar la autentificación basada en host y limpiar todos los hosts permitidos utilizaremos:

```
# xhost -  
access control enabled, only authorized clients can connect
```

Aunque no sería buena idea, podemos desconectar todas las autentificaciones y permitir a todo el mundo que se conecte a nuestro sistema y a nuestras aplicaciones, escribiendo:

```
# xhost +  
access control disabled, clients can connect from any host
```

Para autorizar a determinados hosts a conectarse utilizamos también la opción +. Por ejemplo, para dar permisos de conexión a tres hosts escribiríamos lo siguiente:

```
# xhost +host1 host2 host3  
host1 being added to access control list  
host2 being added to access control list  
host3 being added to access control list
```

Para eliminar un host de la lista de permisos utilizaremos la opción -. Por ejemplo, para eliminar a host2 escribiríamos lo siguiente:

```
# xhost -host2  
host2 being removed from access control list
```

Finalmente, para asegurarnos de que nuestras autentificación es correcta escribiríamos solamente xhost:

```
# xhost  
access control disabled, clients can connect from any host  
INET:host1  
INET:host3
```

106.2.5.4. CONFIGURACIÓN DE CLIENTES REMOTOS

Utilizando la variable de entorno *DISPLAY* podemos indicar a una aplicación que se muestre en un sistema remoto. El formato de uso de la variable DISPLAY es el siguiente:

DISPLAY=Nombre_del_Servidor:display.screen

Por ejemplo:

DISPLAY=host1.the-nashes.net:0.0

Donde Nombre_del_Servidor es el nombre de la máquina en la cual está corriendo el servidor X (también puede ser la dirección IP), display es el número de monitor conectado al servidor (se empieza a contar desde 0), screen es el número de pantalla virtual (se comienza a contar desde 0). Si solo hubiese un usuario utilizando un servidor X en el sistema remoto el número de display sería 0. El número de screen se utiliza solo en entornos multipantalla y puede omitirse si solo se utiliza una única pantalla.

Después de modificar la variable DISPLAY, cualquier cliente X que se inicie se mostrará automáticamente en el sistema especificado en la variable, si la autentificación así lo permitiese.

Otra manera de mostrar clientes remotamente, consiste en emplear la opción de línea de comando -

display con el comando *xclient*. La sintaxis es la siguiente:

xclient -display nombrehost:display.pantalla [argumentos de xclient]

Éste método es práctico para cuando solo deseamos mostrar uno o dos clientes en otro sistema.

106.2.5.5. CONFIGURACIÓN DEL LOGIN REMOTO

En *xdm* podemos conectarnos remotamente a otro sistema y utilizar nuestro sistema local solo para realizar la visualización. También podemos configurar un sistema para proporcionar un cliente con un menú de distintos sistemas de login. La funcionalidad lo proporciona y controla el protocolo XDMCP (X Display Manager Control Protocol). Estos ajustes deberían funcionar con todos los clientes X y servidores con soporte XDMCP.

Si un sistema remoto ejecuta *xdm*, es fácil conectar con dicho sistema desde X y ejecutar aplicaciones desde él, utilizando la siguiente sintaxis:

X -query <servidor nombre>

X también soporta la posibilidad de buscar en la red local cualquier servidor que ruede XDM, mediante el siguiente comando:

X -broadcast

Los sistemas Linux también pueden configurarse para proporcionar una lista de los hosts con los que podemos conectarnos, esta lista se llama *chooser*. Para solicitar un *chooser* emplearíamos el siguiente comando:

X -indirect <nombre servidor>

106.2.5.6. RECURSOS DE X

Los recursos de X se almacenan en el servidor X en lugar de en un fichero de configuración, para que las aplicaciones que se inicien en los clientes los usen. Durante la ejecución del servidor X, los recursos de éste se almacenan en por defecto en dos sitios, dependiendo de a quien queremos aplicárselos. Será en la propiedad *RESOURCE_MANAGER* del directorio raíz de la pantalla 0, si se aplican a todas las pantallas clientes, o en la propiedad *SCREEN_RESOURCES* de la ventana raíz de una pantalla en concreto si tratamos de aplicárselo un cliente en particular.

Los recursos son manipulados por el programa *xrdb*. Muchas configuraciones de X ejecutan *xrdb* en el arranque, para cargar los recursos del fichero *.Xresources* que se encuentra en el directorio *home* del usuario. De esta manera, cada aplicación busca los recursos en las dos ubicaciones indicadas anteriormente para poderlas usar.

Se pueden ver los recursos actuales desde una consola haciendo:

\$ xrdb -query

106.2 EXTRAS

106.2 EXTRAS LXDE

Cuando de velocidad y rendimiento se trata, no hay nada como el confiable LXDE (*"Lightweight X11 Desktop Environment"*). Rápido, sencillo pero muy bien equipado, LXDE permite implementar una interfaz gráfica, allí donde los recursos están al límite de su capacidad. Ya sea se requiera para una computadora fabricada en la década pasada o actual, LXDE es una excelente opción a considerar. Pero sobre todo, su eficiencia y simplismo lo hacen ideal para ordenadores de bajos recursos y es el preferido por administradores o usuarios para los que el rendimiento lo es todo. En la actualidad, LXDE es capaz de correr en hardware tan antiguo como el Pentium II de 266 Mhz con 192 de RAM!.

A diferencia de otros ambientes de escritorio, los componentes no se integran firmemente. Al contrario, los componentes son independientes, y cada uno de ellos se puede utilizar independientemente con muy pocas dependencias.

LXDE usa Openbox como gestor de ventanas predeterminado y apunta a ofrecer un escritorio ligero y rápido basado en componentes independientes que pueden ser utilizados en otros entornos.

Al parecer se va a unir a otro proyecto de escritorio ligero escrito en QT, razor-qt, lo que augura mayor difusión.

Instalación en Debian

En CentOS aún no existen rpm para instalar este escritorio, por ello sólo se analizará sobre la plataforma Debian aunque hay muchas distros que lo soportan.

Varias formas para realizar la instalación en función de las necesidades:

```
#apt-get install lxde-core // mínima  
#apt-get install lxde // completa  
#apt-get install task-lxde-desktop //todo el escritorio
```

El gestor de sesiones de este escritorio es propio (lxdm lxsessions) y no se instala en el core.

106.3. Accesibilidad

Peso en el examen de certificación: 1 puntos.

Objetivo: Demostrar conocimiento y conciencia de la tecnología de accesibilidad.

Conceptos y áreas de conocimiento:

- Configuración de las características de accesibilidad del teclado.
- Configuración visual y de los temas.
- Tecnología de asistencia (ATs)

Términos y utilidades

Repetición de teclas por pulsación permanente.

- Enlentecer/rebotar/commutar teclas.
- Botones del ratón.
- Temas de Escritorio de alto contraste.
- Temas de impresión de gran tamaño.
- Lector de pantalla.
- Línea Braille.
- Engrandecer pantalla.
- Teclado en pantalla.
- Gestos.
- Orca
- GOK
- emacspeak

106.3. Accesibilidad

106.3.0. Introducción

Si bien la mayoría de los usuarios no tienen problemas para leer una pantalla, escribir con un teclado y utilizar un ratón, otros pueden padecer trastornos o discapacidades que les impidan trabajar de manera normal con estos dispositivos. Hay una amplia gama de discapacidades físicas que pueden afectar la capacidad del usuario para interactuar con las computadoras y las aplicaciones.

La mayoría de las distribuciones de Linux vienen con algunas herramientas de ayuda incorporadas para facilitar el uso de los sistemas informáticos a aquellos usuarios que, por motivos de sufrir alguna discapacidad, ya sea visual, auditiva o motora, no puedan desarrollar sus tareas con los medios habituales que les proporcionan los sistemas operativos. En este apartado estudiaremos dichas herramientas.

106.3.1. Asistencia visual

Las personas con problemas de visión apreciarán la posibilidad de utilizar los siguientes tipos de herramientas:

- Lectores de pantalla

- Lupas virtuales o Aumentadores de pantalla
- Lectores de Braille
- Los entornos de escritorio

Lectores de pantalla: son aplicaciones de software que proporcionan traducción de la información visible en la pantalla del ordenador a un formato de salida de audio. La traducción se pasa al sintetizador de voz, y las palabras se escuchan en voz alta. En la actualidad, están disponibles para Linux lectores de pantalla completamente funcional sólo en modo de consola. Los siguientes son algunos de los lectores de pantalla más comunes:

Emacspeak: Una de las primeras herramientas fue Emacspeak (actualmente en la versión 31), un lector de pantalla gratis que permite a los usuarios interactuar de forma independiente con la computadora. Esta herramienta está clasificada como un lector de pantalla, pero el creador lo llama un "escritorio de audio."

Se trata de un excelente, interfaz basada en texto no gráfico para los usuarios que tengan discapacidad visual. Esta aplicación se puede utilizar como un lector de pantalla en conjunción con un sintetizador hardware o IBM ViaVoice®. Se encuentra disponible para la mayoría de versiones de Linux. El escritorio de Emacspeak trabaja con una variedad de aplicaciones, incluyendo los navegadores.

Jupiter Speech System: Un lector de pantalla de Linux en modo consola. Este paquete también incluye la capacidad de leer archivos de registro de una sesión interactiva personalizable y contiene comandos de voz.

Speakup: Un paquete lector de pantalla para el sistema operativo Linux. Se requiere un hardware sintetizador de voz, tales como el DECTalk Express. Permite interacción con el ordenador por órdenes verbales, además de comentarios de voz.

Orca: Un lector de pantalla diseñado para trabajar con aplicaciones y herramientas de asistencia al servicio del proveedor de la interfaz (AT-SPI). Está incluido en el entorno de escritorio GNOME y en sus aplicaciones, OpenOffice, Firefox, y la plataforma Java.

Orca puede ser activado en el sistema/menú de preferencias de los entornos GNOME. Orca incluye soporte para las herramientas de ayuda para el habla, Braille, y ampliación de la pantalla.

Lupas Virtuales: Otros productos que sirven como aumentadores de pantalla, que permiten a los usuarios que son parcialmente ciegos ampliar las áreas seleccionadas de la pantalla, similar al uso de una lupa de vidrio:

SVGATextMode: Este producto amplía o reduce el tamaño de fuente para los usuarios que prefieren trabajar en modo consola. La pantalla de texto normal que proporciona Linux es de 80 caracteres horizontalmente y 25 verticalmente. Después de que SVGATextMode esté instalado, el texto se puede mostrar con un tamaño mucho más grande, por ejemplo, de 50 caracteres horizontalmente y 15 verticalmente. El programa no ofrece la posibilidad de acercar y alejar, pero el usuario puede cambiar el tamaño cuando lo considere necesario. No intente ejecutar SVGATextMode en una ventana Xterminal, ya que, debe estar en modo de consola la pantalla para que funcione correctamente.

Xzoom: Un amplificador de pantalla que permite al usuario aumentar, rotar o duplicar una parte de la pantalla.

Lectores de Braille: Algunas aplicaciones adicionales que pueden ser utilizados para apoyar los dispositivos Braille son:

BrLTty: Compatible con puerto paralelo y pantallas Braille USB, proporciona acceso a la Linux

en modo consola. Son drivers de terminal, que proporcionan completas capacidades de revisión de pantalla.

Blind + Linux = Blinux: Proporciona una lista de correo que se centran en los usuarios que son ciegos, además de documentación y descargas.

Entornos de escritorio: Además, todos los entornos de escritorio dignos de este nombre, como KDE o GNOME (por ejemplo), permiten, para aquellos usuarios que tengan una visión deficiente, modificar la visualización de:

- Colores: aumentando por ejemplo los contrastes o con temas según la representación de los colores (daltonismo).
- Estilos: modificando el tamaño de varios elementos visuales, como los botones, las casillas, los campos, etc.
- Fuentes de caracteres: eligiendo una fuente adaptada, de mayor tamaño, de un estilo determinado.
- Temas: existen temas específicos para los invidentes que modifican el conjunto de los ajustes anteriores de una vez.

Hay herramientas sencillas que permiten ampliar zonas de la pantalla con una lupa virtual. KDE propone *Kmagnifier*. También es posible asociar acciones (teclas del teclado o del ratón) a la lupa para tener un acceso más rápido.

106.3.2. Asistencia auditiva

Las personas con problemas auditivos también pueden trabajar con Linux, ya que existen soluciones para "hacer hablar" al ordenador. Productos tales como *espeak* (sintetizador muy "computerizado") y sobre todo *mbrola* con *freetts* dan resultados muy cercanos a una voz humana.

Además, existen varios medios para poder realizar "**Notificaciones visuales de alerta**" que activan alertas visuales como iluminar la pantalla o la ventana, o parpadear la pantalla, cuando se reproduzca una alerta en el sistema.

106.3.3. Asistencia motora

Las personas con problemas motoras también trabajar con Linux, ya que existen soluciones para facilitarles el trabajo con los teclados y los ratones.

Teclado : Un simple ajuste del teclado puede ayudar a una persona con problemas motoras. Por ejemplo, una persona lenta va a dedicar mucho tiempo a buscar una tecla, con el riesgo de equivocarse; luego apretará demasiado tiempo la tecla deseada, lo que tendrá como efecto repetir el carácter tecleado varias veces.

El sistema operativo Linux tiene funciones integradas que permiten la configuración de teclados adicionales. En algunos de los equipos de escritorio X Windows, estos ajustes se pueden cambiar desde el menú de preferencias.

Una aplicación desarrollada para XWindows, llamada *AccessX*, proporciona una interfaz gráfica de usuario que permite muchos ajustes. Estos ajustes son accesibles desde los centros de configuración de los entornos gráficos. *AccessX* permite ajustar a los usuarios con diferentes problemáticas con los teclados el siguiente conjunto de opciones configurables:

StickyKeys: Permite al usuario bloquear las teclas modificadoras (por ejemplo, Ctrl y Shift), lo que permite las operaciones con un solo dedo, en lugar de múltiples combinaciones de teclas.

SlowKeys: Esta configuración requiere que el usuario mantenga presionada la tecla durante un período determinado de tiempo para que la tecla sea aceptada. Esto evita que las pulsaciones de teclas que se pulsen de forma accidental sean consideradas como pulsaciones válidas.

ToggleKeys: Suena una alerta de sonido que advierte al usuario que una combinación de teclas ha creado un estado de bloqueo de teclas, tales como el bloqueo de mayúsculas y el bloqueo numérico.

RepeatKeys: Permite a un usuario con limitada coordinación un tiempo adicional para liberar las teclas antes de que considere la aplicación que se trata de una secuencia repetitiva de teclas.

BounceKeys o Delay Keys: Permite tener un retraso entre pulsaciones de teclas. Esta función puede ayudar a prevenir la aceptación de las pulsaciones involuntarias.

En KDE4, el módulo "Accesibilidad" de la pestaña "General" de la configuración del sistema permite modificar numerosos ajustes por defecto, entre los cuales se encuentra el funcionamiento del teclado. Por ejemplo, en la pestaña "Filtros de teclado", seleccione "Usar teclas lentas". Si una persona quiere efectuar una acción, no se tendrá en cuenta hasta que se cumpla el plazo elegido (0,5 segundos por defecto). Otra opción configurable es, "Usar teclas rebotantes" que espera el plazo indicado antes de repetir una tecla en caso de apretar durante demasiado tiempo.

También disponemos de los teclados en pantalla que permiten al usuario seleccionar teclas mediante un dispositivo señalizador, como un ratón, trackball o pantalla táctil, y puede ser utilizado en lugar de un teclado estándar.

Gtkeyboard: Es un teclado gráfico en pantalla, fácil de instalar.

GNOME Onscreen Keyboard (GOK): El teclado gráfico en pantalla, que permite a los usuarios controlar sus ordenadores sin depender de un teclado estándar o de ratón.

Ratón : Como no todo el mundo puede utilizar el ratón, es posible desplazar el cursor en la pantalla usando el teclado.

El AccessX, también, permite a los usuarios con problemas motoras que no pueden hacer un uso adecuado del dispositivo de ratón configurar el ajuste de:

MouseKeys: Que proporciona las secuencias alternativas de teclado para el movimiento del cursor y las operaciones del botón del ratón.

En KDE4, en el módulo "Teclado y Ratón", en la pestaña "Navegación de ratón", si marca "Mover ratón con el teclado", podrá mover el cursor del ratón con las teclas direccionales del teclado numérico. Las teclas de 1 a 9, salvo la 5, desplazan el cursor en la dirección deseada. La tecla 5 simula un simple clic (5 dos veces rápidamente simula un doble clic). Es posible cambiar de botón, mantenerlo, etc.

107 TAREAS ADMINISTRATIVAS.

- 107.1. Administrar usuarios, grupos y archivos de sistema relacionados.
- 107.2. Automatizar y programar tareas administrativas del sistema.
- 107.3. Localización e Internacionalización.

107.1. Administrar usuarios, grupos y archivos de sistema relacionados.

Peso en el examen de certificación: 5 puntos.

Objetivo: Añadir, eliminar, suspender y cambiar cuentas de usuario.

Conceptos y áreas de conocimiento:

- Añadir, modificar y eliminar usuario y grupos.
- Mantener información de usuario/grupos en las bases de datos password/group.
- Crear y mantener cuentas de propósito especial y limitadas.

Términos y utilidades:

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/skel
- chage
- groupadd
- groupdel
- groupmod
- passwd
- useradd
- userdel
- usermod

107.1.1. Introducción.

Linux es un sistema operativo multiusuario donde cada usuario del sistema se identifica mediante una cuenta. Además de estas cuentas de usuario, Linux permite utilizar cuentas de grupo. En este punto vamos a tratar cómo administrar estas cuentas de usuario y de grupo mediante comandos del sistema Linux y también vamos a estudiar los archivos de sistema donde se almacenan estas cuentas.

107.1.2. Administrar cuentas de usuarios.

107.1.2.1. Nombres de usuario.

Un nombre de usuario en Linux debe comenzar obligatoriamente por una letra y puede estar formado por cualquier combinación de letras mayúsculas, minúsculas, números y muchos de los símbolos de puntuación. Se desaconseja utilizar espacios en blanco en los nombres de usuario, ya que, aunque se admiten, causan problemas en determinadas utilidades de Linux. La longitud máxima de un nombre en Linux es de 32 caracteres.

Linux distingue entre mayúsculas y minúsculas en los nombres de usuario, aunque se recomienda escribir siempre los nombres de usuario en minúsculas; puede ser confuso contar en el sistema con varios usuarios con el mismo nombre y que se diferencien únicamente en el uso de mayúsculas y minúsculas.

107.1.2.2. ID de usuarios. UID.

Linux asigna a cada usuario un número de identificación que se conoce como ID de usuario (UID). Linux en realidad no utiliza los nombres de usuario para llevar el control de los mismos sino que utiliza los UID. El UID 0 (cero) es el utilizado para el usuario **root**, los UID de 1 a 99 los reserva el sistema para su propio uso asignándolos a sus propias cuentas internas. Mas allá del ID 100, las ID de usuario se encuentran disponibles para los usuarios normales aunque muchas distribuciones reservan hasta el UID 500 o incluso el 1000 para fines especiales. Así, es corriente que el primer usuario que se cree en el sistema por nuestra parte reciba por parte del sistema el UID 1000. El límite de ID de usuarios es de 65.536 en los kernel 2.2.x y sobre 4,2 billones con los kernel 2.4.x y posteriores.

Se pueden crear varias cuentas de usuario y asignarles el mismo UID, consiguiendo que el sistema los trate como el mismo usuario a la hora de asignarles permisos. Esta práctica se desaconseja ya que puede dificultar la gestión de usuarios y provocar efectos indeseados. Muchas veces los intrusos de sistema intentan modificar el UID de un usuario, asignándole el número 0, con lo que a efectos de permisos se transforman en el usuario **root**.

107.1.2.3. Añadir usuarios.

El comando que se utiliza para añadir usuarios al sistema es **useradd**.

useradd [opciones] nombre-usuario

- **nombre-usuario:** El nombre de usuario que vamos a crear.

Las opciones más frecuentemente utilizadas en este comando son las siguientes:

- **-c comentario:** Añade un comentario al usuario, usado normalmente para indicar el nombre completo del usuario.
- **-d directorio-home:** Usa directorio-home como el directorio inicial (home) del usuario. Es en este directorio donde el usuario almacenará sus ficheros propios de configuración, y donde estará situado cuando se inicie sesión.
- **-e fecha:** La fecha en la que expirará la cuenta, llegado el sistema a esa fecha, la cuenta se desactivará automáticamente. Por defecto las cuentas en Linux no expiran.
- **-g grupo:** El nombre o GID del grupo por defecto o principal del usuario.
- **-G grupo:** Nombres o GID de grupos adicionales del usuario, separados por coma.
- **-m :** Crea el directorio home del usuario.
- **-M:** No crea el directorio home del usuario, aunque en el fichero de configuración */etc/login.defs* se especifique que debe crearse por defecto.
- **-s shell:** Usa shell como el intérprete de comandos (shell) por defecto para el usuario.
- **-u número:** Establece el UID del usuario. Si el UID ya existe en el sistema, obtendremos un mensaje de error.
- **-o :** Permite establecer UID de usuarios repetidos con la opción **-u**.
- **-n :** No crear un grupo con el mismo nombre que el usuario. Esta se realiza automáticamente en muchas distribuciones, y este grupo creado suele ser el grupo principal del usuario.
- **-D :** Lista (y opcionalmente cambia) los valores por defecto del sistema para crear usuarios. Estos valores por defecto son establecidos en el fichero */etc/default/useradd*.

Muchas distribuciones de Linux cuentan con un comando **adduser**, que se encarga de ejecutar **useradd** pero de una forma más amigable. Por ejemplo, siempre después de utilizar el comando **useradd** tendremos que ejecutar el comando **passwd** para asignar una contraseña al usuario, mientras que el comando **adduser** directamente realiza esta función por nosotros.

107.1.2.4. Definir contraseñas.

Las contraseñas para los usuarios se definen con el comando **passwd**. Las contraseñas en Linux pueden contener letras, números y signos de puntuación, Linux distingue entre mayúsculas y minúsculas.

passwd [opciones] nombre-usuario

- **nombre-usuario:** El nombre de usuario al que le vamos a asignar (o cambiar) una contraseña. Si se omite este nombre de usuario, se asigna (o cambia) la contraseña al usuario actual del sistema.

Las opciones más frecuentemente utilizadas con este comando es:

- **-k :** Actualiza una cuenta de usuario expirada.
- **-l :** Bloquea la cuenta de usuario. Esta opción solo puede ser utilizada por el usuario root.
- **-u :** Desbloquea una cuenta de usuario.
- **-d :** Borra la contraseña de una cuenta.
- **-S :** Muestra información sobre la contraseña de una cuenta, indicando si esta definida y qué tipo de encriptación utiliza.

107.1.2.5. Modificar usuarios.

Una vez creada, una cuenta de usuario puede modificarse mediante el comando **usermod**.

usermod [opciones] nombre-usuario

- **nombre-usuario:** El nombre de usuario que vamos a modificar.

Este comando acepta la mayoría de las opciones del comando useradd. Otras opciones frecuentemente utilizadas en este comando son:

- **-d directorio:** Cambia el directorio home o inicial del usuario.
- **-m :** Usada con la opción -d, mueve los ficheros del directorio home antiguo al nuevo.
- **-l nombre:** Cambia el nombre de la cuenta del usuario al nombre especificado
- **-L :** Bloquea la cuenta de usuario.
- **-U :** Levanta el bloqueo de la cuenta de usuario.

Un uso especial del comando **usermod** consiste en añadir usuarios a un grupo mediante la opción -G. El uso general de esta opción es el siguiente:

usermod -G grupo1,grupo2,grupo3... nombre-usuario

Hay que tener en cuenta que la opción -G añade a nombre-usuario en los grupos indicados y elimina la pertenencia a cualquier otro grupo de dicho usuario. Por esto, no se suele utilizar este formato ya que nos obliga a indicar todos los grupos del usuario.

También podemos modificar los parámetros de las cuentas relacionados con la expiración de la misma con el comando **chage**

chage [opciones] nombre-usuario

Las opciones usadas en este comando son:

- **-l :** Muestra la información de expiración de la cuenta de usuario y de la contraseña.
- **-m número :** Define el número mínimo de días entre cambios de contraseñas. Un 0 indica que no hay límite.
- **-M número :** Define el número máximo de días que una contraseña puede estar activa antes de expirar.

- **-E fecha** : Define la fecha de expiración de la cuenta.

107.1.2.6. Eliminar usuarios.

Una vez creada, una cuenta de usuario puede eliminarse mediante el comando **userdel**.

userdel [opciones] nombre-usuario

- **nombre-usuario**: El nombre de usuario que vamos a eliminar.

Las principales opciones utilizadas con este comando son:

- **-f** : Fuerza la eliminación de la cuenta del usuario, aunque este tenga sesión abierta en el sistema.
- **-r** : Elimina además de la cuenta de usuario, el directorio inicial (home) del usuario junto con todo su contenido.

Hay que tener cuidado al borrar usuarios, ya que es posible que queden en el sistema ficheros que pertenezcan al usuario eliminado.

107.1.3. Administrar grupos de usuarios.

107.1.3.1. Uso de los grupos.

Los grupos se utilizan para organizar a los usuarios. Cada fichero en un sistema Linux está asociado a un grupo específico, pudiéndose asignar permisos a dicho grupo. Esto permite que un usuario pueda tener permisos en un fichero simplemente por pertenecer al grupo que tiene permisos en dicho fichero. Un usuario puede pertenecer a múltiples grupos y un grupo puede contar con varios, con uno o con ningún usuario miembro. Así, si el grupo **profesores** cuenta con permisos para leer un fichero, todo usuario miembro del grupo podrá leerlo.

Cada usuario pertenece a un grupo por defecto o grupo primario que viene definido en la configuración del usuario (fichero /etc/passwd). Cuando un usuario realiza acciones en el sistema (crear un fichero, ejecutar un programa...) estas acciones se asocian con el grupo primario del usuario de modo que si por ejemplo un usuario crea un fichero, ese fichero pasa a ser propiedad del grupo principal del usuario. Aparte de este grupo primario, cada usuario de Linux puede pertenecer a todos los grupos secundarios que necesite.

Es común en varias distribuciones Linux crear automáticamente una cuenta de grupo por cada cuenta de usuario usando el mismo nombre para ambas y asignar dicho grupo como grupo principal del usuario. Así, si creamos una cuenta de usuario con nombre **alumno**, se crea también una cuenta de grupo con nombre **alumno** que será su grupo primario.

107.1.3.2. ID de grupos. GID.

Al igual que las cuentas de usuario, cada cuenta de grupo es identificada por el sistema mediante un número de identificación (GID). Todo lo que se ha explicado para los ID de usuario en el punto 7.1.2.2 se aplica a los ID de grupo, así por ejemplo el GID del grupo root es el 0.

107.1.3.3. Añadir grupos.

El comando que se utiliza para añadir grupos al sistema es **groupadd**.

groupadd [opciones] nombre-grupo

- **nombre-grupo:** El nombre de grupo que deseamos crear.

Las principales opciones de este comando:

- **-g número:** Especifica el GID con el que se creará el grupo. Por defecto no se permite usar un GID para varios grupos.
- **-o :** Permite usar un GID para varios grupos, desactivando el error de la opción –g.

107.1.3.4. Modificar grupos.

El comando **groupmod** modifica los datos de un grupo ya existente.

groupmod [opciones] nombre-grupo

Las opciones usadas con este comando son:

- **-g número:** Establece número como el nuevo GID del grupo. Devolverá error si se introduce un GID que ya existe.
- **-o :** Permite establecer un GID al grupo con la opción –g aunque este GID ya exista en el sistema, de modo que tendremos 2 grupos con el mismo GID.
- **-n nombre-grupo-nuevo:** Permite cambiar nombre-grupo por nombre-grupo-nuevo.

107.1.3.5. Cambiar grupo primario.

Podemos cambiar el grupo principal de un usuario mediante el comando **newgrp**.

newgrp nombre-grupo

Este comando modifica el grupo primario del usuario actual, pasando a ser el grupo primario el indicado como nombre-grupo. Normalmente el intentar cambiar de grupo principal se nos solicitará una contraseña que se crea mediante el comando **gpasswd**.

107.1.3.6. Definir contraseñas para los grupos.

El comando **gpasswd** nos permite establecer contraseña para los grupos, y además podemos modificar otras características de los mismos y asignar administradores a los grupos.

gpasswd [opciones] nombre-grupo

- **-a usuario:** añade el usuario especificado al grupo indicado.
- **-d usuario:** elimina el usuario especificado del grupo indicado.
- **-R:** Impide que los usuarios se unan a este grupo mediante el comando **newgrp**.
- **-r:** borra la contraseña del grupo.
- **-A usuario:** Transforma a usuario en el administrador del grupo. Este administrador de grupo puede añadir y borrar miembros del grupo y cambiar la contraseña del mismo. Se puede escribir más de un nombre de administrador separándolos con comas. Estos usuarios ya deben de ser miembros del grupo antes de poder ser administradores.
- **-M usuario:** Funciona igual que la opción –A pero añade a los usuarios al grupo antes de hacerlos administradores.

Si se ejecuta **gpasswd** sin indicarle ninguna opción, nos permitirá cambiar la contraseña del grupo indicado. Esta contraseña se usa para controlar el uso del comando **newgrp**. Esta opción no funciona adecuadamente en algunas distribuciones.

107.1.3.7. Eliminar grupos.

El comando que se utiliza para eliminar grupos del sistema es **groupdel**.

groupdel nombre-grupo

- **nombre-grupo:** El nombre de grupo que se desea eliminar del sistema.

Hay que tener cuidado al borrar grupos, ya que es posible que queden en el sistema ficheros que pertenezcan al grupo eliminado.

107.1.4. Archivos de sistema relacionados con usuarios y grupos.

Las cuentas de usuario y grupo, así como las contraseñas de los mismos se almacenan en el sistema en los ficheros */etc/passwd*, */etc/shadow*, */etc/group* y */etc/gshadow*. Estos ficheros pueden ser modificados manualmente para gestionar usuarios y grupos, pero no es una práctica recomendable.

107.1.4.1. Fichero de usuarios *passwd*.

Cuando se añade un usuario al sistema se añade una línea al fichero de usuarios */etc/passwd*. Cada línea de este fichero contiene información sobre una cuenta de usuario, con campos separados por el símbolo dos puntos. La estructura de cada línea es la siguiente:

nombre-cuenta:contraseña:UID:GID:nombre-completo-usuario:directorio-home-usuario:shell-por-defecto

Así por ejemplo, la línea de */etc/passwd* siguiente:

root:x:0:0:root:/root:/bin/bash

se descompone de la siguiente forma:

- **root** Nombre de la cuenta de usuario.
- **x** Contraseña del usuario
- **0** UID
- **0** GID
- **root** Nombre completo del usuario
- **/root** Directorio home o inicial del usuario
- **/bin/bash** Shell por defecto que utiliza el usuario

Veamos un poco más en profundidad cada uno de los campos:

- **Nombre-cuenta:** Es el nombre de la cuenta de usuario y debe ser único en el fichero.
- **Contraseña:** Cada nombre de usuario tiene asignada una contraseña. La contraseña aparece en este campo de una forma cifrada (ininteligible e irreversible). Por razones de seguridad la mayoría de las distribuciones almacenan las contraseñas en un fichero separado */etc/shadow*, apareciendo en este campo de */etc/passwd* una letra X.
- **UID:** Identificador del usuario. Este UID es un entero no negativo, asignándose al root el UID 0 que otorga permisos especiales en el sistema. Normalmente los valores de 0 a 99 son reservados para uso de administración. En muchas distribuciones este número para usuarios normales comienza en 500 o 1.000.
- **GID:** Identificador del grupo. Este GID es un entero no negativo. La correspondencia entre este número y el nombre del grupo se especifican en el fichero */etc/group*.
- **Nombre-completo:** El nombre completo del usuario. Muchas distribuciones utilizan este campo para almacenar otro tipo de comentarios. Este campo puede contener espacios en blanco.
- **Directorio-home-usuario:** Indica el directorio inicial del usuario. Este directorio suele estar

situado en `/home/nombre-usuario` y suele incluir ficheros que sirven para configurar y personalizar el entorno del usuario.

- **Shell-predeterminado:** El programa intérprete de comandos (Shell) que el usuario utiliza por defecto. En la mayoría de los casos este Shell suele ser `/bin/bash`, pero puede ser cualquier otro. En ocasiones se especifica aquí un programa no ejecutable como `/bin/false`, a efectos de impedir que ese usuario pueda abrir ninguna Shell al hacer login, y por lo tanto, no pueda ejecutar ningún comando en el sistema.

107.1.4.2. Fichero de contraseñas de usuario shadow.

Como hemos indicado anteriormente, las contraseñas habitualmente suelen estar almacenadas en el fichero `/etc/shadow`. Este fichero cuenta con unos permisos mucho más restrictivos que `/etc/passwd` y solo puede ser accedido por el usuario **root**. En cada línea de este fichero se almacena la contraseña y otra información referente a un usuario. Una línea de `/etc/shadow` suele ser algo parecido a lo siguiente:

`pedro:H/huFkiTSnTaiQ:26472:0:-1:7:-1:-1:`

Como vemos, al igual que en el fichero `/etc/passwd` cada línea cuenta con varios campos separados por el símbolo dos puntos. Cada uno de estos campos almacena lo siguiente:

- **Nombre de usuario:** El nombre de usuario, tal como existe en `/etc/passwd`.
- **Contraseña:** La contraseña se almacena de forma encriptada, de modo que viendo lo aquí almacenado no se puede averiguar la verdadera contraseña. Si no aparece aquí ninguna contraseña (`::`) el usuario podrá abrir sesión sin contraseña. Un asterisco o un símbolo de exclamación (`*`) indican que dicho usuario está bloqueado y no puede abrir sesión en el sistema.
- **Ultimo cambio de contraseña:** (26472 en nuestro ejemplo). La fecha en la que se cambió la contraseña por última vez. Esta fecha se almacena indicando el número de días que pasan desde el 1 de enero de 1970.
- **Días hasta que se permita un cambio de contraseña:** Número de días que deben transcurrir para que al usuario le sea permitido cambiar la contraseña.
- **Días antes de que sea requerido un cambio de contraseña:** Número de días desde el último cambio de contraseña antes de que la contraseña caduque y se le solicite otra al usuario.
- **Días entre la expiración y la desactivación:** Linux permite unos días entre la expiración de una cuenta y su desactivación. En este campo se indica este periodo de gracia.
- **Fecha de expiración de la cuenta:** Fecha en la que la cuenta expirará. La fecha se almacena como el número de días transcurridos desde el 1 de enero de 1970.
- **Campo reservado:** Este campo se reserva para futuros usos, y no se utiliza actualmente para nada. Normalmente está vacío.

En los campo relativos a número de días, un valor de -1 o de 99999 indica que esa característica está deshabilitada en el sistema. Estos valores se modifican con los comandos **usermod** y **chage**.

107.1.4.3. Fichero de grupos group.

Las cuentas de grupo están almacenadas en el fichero `/etc/group`. Al igual que en el fichero `/etc/passwd` cada línea contiene una serie de campos separados por el símbolo dos puntos. Cada uno de estos campos almacena lo siguiente:

- **Nombre de grupo:** Cada grupo debe contar con un nombre de grupo único.
- **Contraseña del grupo:** Los grupos pueden contar con contraseñas, al igual que las cuentas

de usuario. De la misma forma, en este campo no se almacena la contraseña del grupo, quedando para esta labor el fichero */etc/gshadow*.

- **GID:** Identificador del grupo. Un número no negativo que identifica al grupo en el sistema.
- **Miembros:** Aquí podemos encontrar una serie de nombres de usuario, separados por comas. Todos los usuarios en esta lista son miembros del grupo. La lista puede estar vacía o contar con decenas de usuarios.

Aquí vemos un ejemplo de una línea del fichero */etc/group*.

trabajadores:x:1005:luis,pedro,carmen

En esta línea se almacena un grupo con nombre trabajadores, con GID 1005 y que cuenta con 3 usuarios como miembros del grupo (luis, pedro y carmen).

Cuando se crea un nuevo usuario ya vimos como es habitual que se cree también un grupo con su mismo nombre. En este caso, el nuevo usuario es automáticamente asignado por el sistema Linux como miembro del grupo creado con su mismo nombre. Sin embargo, el usuario no aparecerá como miembro en el fichero */etc/group*. Este es el comportamiento por defecto para estos grupos unipersonales.

107.1.4.4. Fichero de contraseñas de grupo gshadow.

Al igual que las cuentas de usuario pueden estar protegidas por contraseñas encriptadas, los grupos también pueden estar protegidos del mismo modo. El fichero donde se almacenan las contraseñas encriptadas de los grupos es */etc/gshadow*.

La contraseña de un grupo puede ser usadas para permitir el acceso a dicho grupo a un usuario que no es miembro del mismo. Los usuarios pueden usar el comando **newgrp** para cambiar el grupo inicial al que pertenecen. Si se establece una contraseña para el grupo, **newgrp** pedirá dicha contraseña y permitirá que dicho usuario pase a ser miembro del grupo y que dicho grupo se establezca como grupo principal del usuario.

Cada línea de este fichero almacena varios campos separados por el símbolo dos puntos. Estos son los campos almacenados:

- **Nombre del grupo:** Nombre del grupo que debe coincidir con el indicado en el fichero */etc/group*.
- **Contraseña encriptada:** Aquí se almacena la contraseña encriptada si ha sido establecida con el comando **gpasswd**. Si no se ha establecido contraseña aparecerá este campo como vacío (::) y lo mismo ocurre si aparece un símbolo de exclamación (!) o un asterisco (*).
- **Usuarios administradores:** Una lista de nombres de usuarios separados por comas. Cada uno de estos usuarios se considera administrador del grupo, y podrán cambiar la contraseña del grupo o bien la lista de miembros del mismo.
- **Usuarios miembros:** Una lista de nombres de usuarios separados por comas. Cada uno de estos usuarios es un miembro del grupo y podrán usar **newgrp** sobre el grupo. Esta lista debe ser la misma que se encuentra en */etc/group*.

Si no establece ninguna contraseña para un grupo, se imposibilita que los usuarios puedan unirse a dicho grupo con **newgrp**.

107.1 EXTRAS

107.1 EXTRAS newgrp

Si un usuario quisiera comprobar a qué grupos pertenece bastaría que ejecutara la orden **groups**.

Por ejemplo, la usuaria julia ejecuta:

```
$ groups  
julia admin usuarios
```

lo que quiere decir que su grupo principal es julia y que también pertenece a los grupos admin y usuarios.

Acceso a un grupo: newgrp

Usuarios miembros de un grupo

Un usuario puede pertenecer a varios grupos y puede operar con los recursos que sean propiedad de ese grupo con los permisos que tenga asignados.

Pero para ciertas operaciones como creación de ficheros nuevos o al lanzar un proceso el grupo que se asigna es el grupo principal del usuario; es lógico, por ejemplo un fichero sólo pertenece a un grupo.

Si queremos que los nuevos ficheros que se crean o los procesos que se lancen pertenezcan a otro grupo deberíamos poder cambiar el grupo principal. Este cambio de grupo principal lo realiza la orden

newgrp

Por ejemplo, la usuaria del sistema julia tiene como grupo principal predeterminado el grupo julia y también pertenece al grupo admin.

Si quisiera que su grupo principal fuera admin tendría que ejecutar:

```
$ newgrp admin
```

y la ejecución de la orden groups quedaría como:

```
$ groups  
admin julia usuarios
```

Usuarios no miembros de un grupo

Los usuarios no miembros de un grupo también pueden acceder a un grupo al que no pertenecen utilizando la orden

newgrp

En este caso al ejecutar la orden newgrp el sistema le solicita una contraseña para aprobar o no el acceso del usuario.

Para que un grupo admita a usuarios que no son miembros es necesario que el administrador del grupo le haya asignado una contraseña al grupo.

107.1 EXTRAS su

El comando *su* permite iniciar una sesión con otro usuario sin necesidad de cerrar la sesión actual.

La sintaxis del comando es:

`su [-] [usuario]`

La ejecución de este comando permite el inicio de sesión bajo las credenciales del usuario escrito como argumento. Si no se está utilizando la cuenta de root, el sistema pedirá la contraseña del usuario para poder iniciar la sesión bajo dicho usuario. Ahora bien, el usuario root puede iniciar la sesión con cualquier usuario sin necesidad de aportar la contraseña. Si los datos son correctos, es decir, login de acceso y contraseña (en caso de tener que escribirla), el comando *su* inicia un nuevo proceso shell, bajo el usuario especificado. Si además se usa la opción - cambiara el entorno de trabajo, mostrando el del usuario con el que estamos iniciando, es decir, cargara el perfil del usuario especificado.

Cuando se hace uso de este comando, el usuario poseerá dos UID y dos GID. Es decir, tendrá un UID y GID correspondientes a su inicio de sesión en el sistema, y además, tendrá un UID y GID efectivos que se corresponderán con el usuario con el que ha iniciado una sesión mediante el comando *su*. Cuando dicho usuario intente acceder al sistema lo hará con los UID y GID efectivos, es decir, EUID y EGID que en el caso de la sesión *su* serán los del nuevo usuario y fuera de esta sesión se corresponden con su sesión iniciada. Para ver los EUID y EGID utilizamos el comando *whoami*, un ejemplo de esto es:

```
jose@cli:~$ id  
uid=1006(jose) gid=1007(jose) grupos=1007(jose)  
jose@cli:~$ pwd  
/home/jose  
jose@cli:~$ su - jose2  
Contraseña:  
$ pwd  
/home/jose2  
$ whoami  
jose2  
$ who am i  
jose      pts/2          2013-12-19 12:24 (172.16.1.2)
```

107.1 EXTRAS sudo

Se verá de forma más sencilla en secciones posteriores de este manual.

autor: sergio.gonzalez.duran@gmail.com

Fuente:http://www.linuxtotal.com.mx/?cont=info_admon_014

En ambientes donde varios usuarios usan uno o más sistemas GNU/Linux, es necesario otorgar distintos permisos o privilegios para que estos puedan hacer uso de comandos propios del usuario administrador 'root'. Totalmente fuera de lugar e impensable es 'entregar' la contraseña de root para

que los usuarios puedan hacer uso de los programas propios de sus funciones pero que son propiedad de 'root'. Por otro lado, hacer uso del comando tampoco es práctico porque es lo mismo, necesitan la contraseña de root, así que la mejor alternativa es hacer uso de .

¿Exáctamente qué es y qué hace ?. sudo permite implementar un control de acceso altamente granulado de que usuarios ejecutan qué comandos. Si un usuario normal desea ejecutar un comando de root (o de cualquier otro usuario), verifica en su lista de permisos y si está permitido la ejecución de ese comando para ese usuario, entonces se encarga de ejecutarlo. Es decir, es un programa que basado en una lista de control (/etc/sudoers) permite (o no) la ejecución al usuario que lo invocó sobre un determinado programa propiedad de otro usuario, generalmente del administrador del sistema 'root'.

, para fines prácticos se puede dividir en tres partes:

- sudo, el comando con permisos de SUID, que los usuarios usan para ejecutar otros comandos a los que se les permite usar.
- visudo, el comando que permite al administrador modificar /etc/sudoers.
- /etc/sudoers, el archivo de permisos que le indica a qué usuarios ejecutan cuáles comandos.

sudo

(SUpерuser DO) lo ejecuta un usuario normal, al que se supone tiene permisos para ejecutar cierto comando. Entonces, requiere que los usuarios se autentifiquen a sí mismos a través de su contraseña para permitirles la ejecución del comando. Veamos un ejemplo:

```
$ sudo /sbin/ifconfig  
Password:  
eth0      Link encap:Ethernet HWaddr 4C:00:10:60:5F:21  
          inet addr:200.13.110.62 Bcast:200.13.110.255 Mask:255.255.255.0  
          inet6 addr: fe80::4e00:10ff:fe60:5f21/64 Scope:Link  
...
```

Como se podrá observar se usa el comando seguido del comando (con toda su ruta si es que este no está en el PATH del usuario) al que se tiene permiso. pregunta por la contraseña del usuario que ejecuta el comando y listo.

Por defecto, después de hacer lo anterior tendrás 5 minutos para volver a usar el mismo comando u otros a los que tuvieras derecho, sin necesidad de ingresar la contraseña de nuevo. Si se quiere extender el tiempo por otros 5 minutos usa la opción sudo -v (validate). Por el contrario, si ya terminaste lo que tenías que hacer, puedes usar sudo -k (kill) para terminar con el tiempo de gracia de validación.

Ahora bien, ¿Qué comandos son los que puedo utilizar?, pues la opción -l es la indicada para eso:

```
$ sudo -l  
User sergio may run the following commands on this host:  
    (root) /sbin/ifconfig  
    (root) /sbin/lspci
```

En el caso anterior se ejecutó un comando de root, pero no tiene que ser así, también es posible ejecutar comandos de otros usuarios del sistema indicando la opción -u:

```
$ sudo -u ana /comando/de/ana
```

Una de las opciones más interesantes es la que permite editar archivos de texto de root (claro, con el permiso otorgado en 'sudoers' como se verá más adelante), y esto se logra con la opción -e, esta opción está ligada a otro comando de llamado que invoca al editor por defecto del usuario, que

generalmente es 'vi'.

```
$ sudo -e /etc/inittab  
(Permitirá modificar el archivo indicado como si se fuera root)
```

Cuando se configura se tienen múltiples opciones que se pueden establecer, estás se consultan a través de la opción **-L**

```
$> sudo -L  
Available options in a sudoers ``Defaults'' line:
```

```
syslog: Syslog facility if syslog is being used for logging  
syslog_goodpri: Syslog priority to use when user authenticates successfully  
syslog_badpri: Syslog priority to use when user authenticates unsuccessfully  
long_otp_prompt: Put OTP prompt on its own line  
ignore_dot: Ignore '.' in $PATH  
mail_always: Always send mail when sudo is run  
mail_badpass: Send mail if user authentication fails  
mail_no_user: Send mail if the user is not in sudoers  
mail_no_host: Send mail if the user is not in sudoers for this host  
mail_no_perms: Send mail if the user is not allowed to run a command  
tty_tickets: Use a separate timestamp for each user/tty combo  
lecture: Lecture user the first time they run sudo  
lecture_file: File containing the sudo lecture  
authenticate: Require users to authenticate by default  
root_sudo: Root may run sudo  
...  
varias opciones más
```

Bastante útil, ya que nos muestra las opciones y una pequeña descripción, estás opciones se establecen en el archivo de configuración 'sudoers'.

Una de las opciones más importantes de consulta es **-V**, que permite listar las opciones (defaults) establecidas por defecto para todos los usuarios, comandos, equipos, etc. Más adelante en este tutorial, aprenderemos como establecer opciones específicas para ciertos usuarios, comandos o equipos. NOTA: tienes que ser 'root' para usar esta opción.

```
# sudo -V  
Sudo version 1.6.9p5  
  
Sudoers path: /etc/sudoers  
Authentication methods: 'pam'  
Syslog facility if syslog is being used for logging: local2  
Syslog priority to use when user authenticates successfully: notice  
Syslog priority to use when user authenticates unsuccessfully: alert  
Send mail if the user is not in sudoers  
Lecture user the first time they run sudo  
Require users to authenticate by default  
Root may run sudo  
Log the hostname in the (non-syslog) log file  
Allow some information gathering to give useful error messages  
Visudo will honor the EDITOR environment variable  
Set the LOGNAME and USER environment variables  
Reset the environment to a default set of variables  
Length at which to wrap log file lines (0 for no wrap): 80  
Authentication timestamp timeout: 5 minutes  
Password prompt timeout: 5 minutes  
Number of tries to enter a password: 3  
Umask to use or 0777 to use user's: 022
```

```
Path to log file: /var/log/sudo.log
```

```
...  
varias opciones más listadas
```

Con intención, trunque el listado anterior en la línea "Path to log file: /var/log/sudo.log", donde se indica cual es el archivo 'log' o de bitacora por defecto de , en este archivo se loguea absolutamente todo lo que se haga con , que usuarios ejecutaron que, intentos de uso, etc.

visudo

Permite la edición del archivo de configuración de sudoers. Invoca al editor que se tenga por defecto que generalmente es 'vi'. cuando es usado, bloquea el archivo /etc/sudoers de tal manera que nadie más lo puede utilizar, esto por razones obvias de seguridad que evitarán que dos o más usuarios administradores modifiquen accidentalmente los cambios que el otro realizó.

Otra característica importante de es que al cerrar el archivo, verifica que el archivo este bien configurado, es decir, detectará si hay errores de sintaxis principalmente en sus múltiples opciones o reglas de acceso que se tengan. Por esta razón no debe editarse /etc/sudoers directamente (perfectamente posible ya que es un archivo de texto como cualquier otro) sino siempre usar .

Si al cerrar detecta un error nos mostrará la línea donde se encuentra, y la pregunta "What now?":

```
>>> sudoers file: syntax error, line 15 <<  
What now?
```

Se tienen tres opciones para esta pregunta:

- e - edita de nuevo el archivo, colocando el cursor en la línea del error (si el editor soporta esta función.)
- x - salir sin guardar los cambios.
- Q - salir y guarda los cambios.

Por defecto el archivo de configuración es /etc/sudoers pero se pueden editar otros archivos que no sean ese y que se aplique la sintaxis de , y esto se logra con la opción -f().

Si tan solo se desea comprobar que /etc/sudoers esta bien configurado se usa la opción -c, toma por el archivo de configuración por defecto o si no se indica algún otro.

```
#> visudo -c  
/etc/sudoers file parsed OK
```

La opción -s activa el modo 'estricto' del uso de , es decir no solo se comprobará lo sintáctico sino también el orden correcto de las reglas, por ejemplo si se define el alias para un grupo de comandos y este se usa antes de su definición, con esta opción se detectará este tipo de errores.

Sudoers

Archivo de configuración de , generalmente ubicado bajo /etc y se modifica a través del uso de . En este archivo se establece quien (usuarios) puede ejecutar que (comandos) y de que modo (opciones), generando efectivamente una lista de control de acceso que puede ser tan detallada como se deseé.

Es más fácil entender si dividimos en tres partes su posible configuración, estas son:

- Alias

- Opciones (Defaults)
- Reglas de acceso

Por extraño que parezca ninguna de las secciones es obligatoria, o tienen que estar en algún orden específico, pero la que al menos debe de existir es la tercera, que es la definición de los controles o reglas de acceso. Se detallará cada uno de estos en un momento. Para los que les gusta saber más la cuestión técnica es interesante saber que la construcción de un archivo esta basado en la forma BNF (Backus-Naur Form), concretamente en versión extendida (EBNF), si estudiaste algún curso de informática universitario seguramente sabes de lo que hablo. EBNF describe de una forma precisa y exacta la gramática de un lenguaje, esta se va creando a través de reglas de producción que a la vez son la base para ser referenciadas por otras reglas. Afortunadamente no necesitas saber nada de esto, solo entender como se aplican estas reglas.

Alias

Un alias se refiere a un usuario, un comando o a un equipo. El alias engloba bajo un solo nombre (nombre del alias) una serie de elementos que después en la parte de definición de reglas serán referidos aplicados bajos cierto criterio. Es decir, regresando a EBNF estamos creando las reglas de producción inicial. La forma para crear un alias es la siguiente:

tipo_alias NOMBRE_DEL_ALIAS = elemento1, elemento2, elemento3, ... elementoN

tipo_alias NOMBRE1 = elemento1, elemento2 : NOMBRE2 = elemento1, elemento2

En el segundo caso, separado por ":" es posible indicar más de un alias en una misma definición.

El tipo_alias define los elementos, es decir, dependiendo del tipo de alias serán sus elementos. Los tipos de alias son cuatro y son los siguientes:

- Cmnd_Alias - define alias de comandos.
- User_Alias - define alias de usuarios normales.
- Runas_Alias - define alias de usuarios administradores o con privilegios.
- Host_Alias - define alias de hosts o equipos.

El NOMBRE_DEL_ALIAS puede llevar letras, números o guión bajo (_) y DEBE de comenzar con una letra mayúscula, se acostumbra a usarlos siempre en mayúsculas.

Los elementos del alias varian dependiendo del tipo de alias, así que veámoslos por partes así como varios ejemplos para que comience a quedar claro todo esto.

Cmnd_Alias

Definen uno o más comandos y otros alias de comandos que podrán ser utilizados después en alias de usuarios. Ejemplos:

Cmnd_Alias WEB = /usr/sbin/apachectl, /usr/sbin/httpd, sudoedit /etc/httpd/

Indica que a quien se le aplique el alias WEB podrá ejecutar los comandos apachectl, httpd y editar todo lo que este debajo del directorio /etc/httpd/, nótese que debe de terminar con '/' cuando se indican directorios. También, la ruta completa a los comandos debe ser indicada.

Cmnd_Alias APAGAR = /usr/bin/shutdown -h 23\:00

Al usuario que se le asigne el alias APAGAR podrá hacer uso del comando 'shutdown' exactamente con los parámetros como están indicados, es decir apagar -h (halt) el equipo a las 23:00 horas. Nótese que es necesario escapar el signo ':', así como los símbolos ' : , = \

Cmnd_Alias NET_ADMIN = /sbin/ifconfig, /sbin/iptables, WEB

NET_ADMIN es un alias con los comandos de configuración de interfaces de red ifconfig y de firewall iptables, pero además le agregamos un alias *previamente* definido que es WEB, así que a quien se le asigne este alias podrá hacer uso de los comandos del alias WEB.

Cmnd_Alias TODO_BIN = /usr/bin/, !/usr/bin/rpm

A quien se le asigne este alias podrá ejecutar todos los comandos que estén dentro del directorio /usr/bin/ menos el comando 'rpm' ubicado en el mismo directorio. *NOTA IMPORTANTE: este tipo de alias con un permiso muy amplios menos '! algo, generalmente no son una buena idea, ya que comandos nuevos que se añadan después a ese directorio también podrán ser ejecutados, es mejor siempre definir específicamente lo que se requiera.*

User_Alias

Definen a uno o más usuarios, grupos del sistema (indicados con %), grupos de red (netgroups indicados con +) u otros alias de usuarios. Ejemplos:

User_Alias MYSQL_USERS = andy, marce, juan, %mysql

Indica que al alias MYSQL_USERS pertenecen los usuarios indicados individualmente más los usuarios que formen parte del grupo 'mysql'.

User_Alias ADMIN = sergio, ana

'sergio' y 'ana' pertenecen al alias ADMIN.

User_Alias TODOS = ALL, !samuel, !david

Aquí encontramos algo nuevo, definimos el alias de usuario TODOS que al poner como elemento la palabra reservada 'ALL' abarcaría a todos los usuarios del sistema, pero no deseamos a dos de ellos, así que negamos con '!', que serían los usuarios 'samuel' y 'david'. Es decir, todos los usuarios menos esos dos. *NOTA IMPORTANTE: este tipo de alias con un permiso muy amplios menos '! algo, generalmente no son una buena idea, ya que usuarios nuevos que se añadan después al sistema también serán considerados como ALL, es mejor siempre definir específicamente a los usuarios que se requieran. ALL es válido en todos los tipos de alias.*

User_Alias OPERADORES = ADMIN, alejandra

Los del alias ADMIN más el usuario 'alejandra'.

Runas_Alias

Funciona exactamente igual que User_Alias, la única diferencia es que es posible usar el ID del usuario UID con el carácter '#'.

Runas_Alias OPERADORES = #501, fabian

Al alias OPERADORES pertenecen el usuario con UID 501 y el usuario 'fabian'

Host_Alias

Definen uno o más equipos u otros alias de host. Los equipos pueden indicarse por su nombre (si se encuentra en /etc/hosts) por nombre de dominio, si existe un resolvidor de dominios, por dirección IP, por dirección IP con máscara de red. Ejemplos:

Host_Alias LANS = 192.168.0.0/24, 192.168.0.1/255.255.255.0

El alias LANS define todos los equipos de las redes locales.

Host_Alias WEBSERVERS = 172.16.0.21, web1 : DBSERVERS = 192.168.100.10, dataserver

Se define dos alias en el mismo renglón: WEBSERVERS y DBSERVERS con sus respectivas listas de elementos, el separador ':' es válido en cualquier definición de tipo de alias.

Opciones (defaults)

Las opciones o defaults permiten definir ciertas características de comportamiento para los alias previamente creados, para usuarios, usuarios privilegiados, para equipos o de manera global para todos. No es necesario definir opciones o defaults, ya tiene establecidas el valor de cada uno, y es posible conocerlas a través de *sudo -V* (ver en la sección sudo de este tutorial).

Sin embargo, la potencia de está en su alta granularidad de configuración, así que es importante conocer como establecer opciones específicas.

Las opciones o defaults es posible establecerlos en cuatro niveles de uso:

- De manera global, afecta a todos
- Por usuario
- Por usuario privilegiado
- Por equipo (host)

Se usa la palabra reservada 'Defaults' para establecer las opciones y dependiendo del nivel que deseamos afectar su sintaxis es la siguiente:

- Global: Defaults opcion1, opcion2 ...
- Usuario: Defaults:usuario opcion1, opcion2 ...
- Usuario Privilegiado: Defaults>usuario opcion1, opcion2 ...
- Equipo: Defaults@equipo opcion1, opcion2 ...

La lista de opciones es algo extensa, pueden consultarse en las páginas del manual (*man sudoers*) o en el excelente manual sobre del sitio web de www.rpublica.net <http://www.rpublica.net/sudo/indice.html#defaults>, está en español y define muy claramente lo que significa cada opción.

Los defaults los divide el manual (*man sudoers*) en cuatro: flags o booleanos, enteros, cadenas y listas. Veamos entonces algunos ejemplos de uso para cada uno de ellos:

flags o booleanos

Generalmente se usan de manera global, simplemente se indica la opción y se establece a 'on' para desactivarla 'off' se antepone el símbolo '!' a la opción. Es necesario consultar el manual para saber el valor por defecto 'on' o 'off' para saber si realmente necesitamos invocarla o no.

Defaults mail_always

Establece a 'on' la opción 'mail_always' que enviara un correo avisando cada vez que un usuario utiliza , a la vez, este opción requiere que 'mailto_user' este establecida.

Defaults !authenticate, log_host

Desactiva 'off' el default 'authenticate' que por defecto esta activado 'on' e indica que todos los usuarios que usen deben identificarse con su contraseña, obviamente esto es un ejemplo y sería una pésima idea usarlo realmente, ya que ningún usuario necesitaría autenticarse, esto es porque estamos usando Defaults de manera global. La segunda opción 'log_host' que por defecto está en 'off' la activamos y bitacoriza el nombre del host cuando se usa un archivo (en vez de syslog) como bitácora de .

Defaults:ana !authenticate

Aquí se aprecia algo más lógico, usamos opciones por usuario en vez de global, indicando que el usuario 'ana' no requerirá autenticarse. Pero todos los demás si.

Defaults>ADMIN rootpw

Opciones para usuarios privilegiados, en vez de usar una lista de usuarios, usamos un alias 'ADMIN' que se supone fue previamente definido, y establecemos en 'on' la opción 'rootpw' que indica a que los usuarios en el alias 'ADMIN' deberán usar la contraseña de 'root' en vez de la propia.

Enteros

Tal como su nombre lo indica, manejan valores de números enteros en sus opciones, que deben entonces usarse como *opción = valor*.

Defaults:fernanda, regina passwd_tries = 1, passwd_timeout = 1

Ejemplo donde se aprecia el uso de opciones con valores enteros. En este caso se establecen opciones para los usuarios 'fernanda' y 'regina' solamente, que solo tendrán una oportunidad de ingresar la contraseña correcta 'passwd_tries' el valor por defecto es de 3 y tendrán un minuto para ingresarla 'passwd_timeout' el valor por defecto son 5 minutos.

La mayoría de las opciones de tiempo o de intentos, al establecerlas con un valor igual a cero entonces queda ilimitado la opción.

Defaults@webserver umask = 011

Se establecen opciones solo para los usuarios que se conectan al servidor 'webserver' y el valor 'umask' indica que si mediante la ejecución del comando que se invoque por es necesario crear archivos o directorios, a estos se les aplicará la máscara de permisos indicada en el valor de la opción.

Cadenas

Son valores de opciones que indican mensajes, rutas de archivos, etc. Si hubiera espacios en el valor es necesario encerrar el valor entre comillas dobles (" ").

Defaults badpass_message = "Intenta de nuevo: "

Para todos los usuarios, cuando se equivoquen al ingresar la contraseña, es el mensaje que saldría. En este caso la opción por defecto es "Sorry: try again".

Listas

Permite establecer/eliminar variables de entorno propias de . Los 'Defaults' para variables es de los menos usados en las configuraciones de y ciertamente de los más confusos. Para entender como se aplican es más fácil si primero ejecutas como 'root' el comando *sudo -V*, y al final del listado encontrarás en mayúsculas las posibles variables de entorno que se pueden establecer o quitar y que vienen del shell.

Solo existen tres opciones de listas: *env_check*, *env_delete* y *env_keep*, las listas pueden ser remplazadas con '=', añadidas con '+='; eliminadas con '-=' o deshabilitadas con '!'. Con un par de ejemplos quedará más claro.

Defaults env_delete == HOSTNAME

Elimina la variable de entorno 'HOSTNAME', (pero preserva todas las demás que hubiera) y comandos que se ejecuten bajo y que requieran de esta variable no la tendrían disponible.

Defaults env_reset

Defaults env_check += DISPLAY, PS1

La primera opción 'env_reset' reinicializa las variables de entorno que utilizará o tendrá disponibles, y solo quedan disponibles LOGNAME, SHELL, USER y USERNAME. La siguiente línea indica

que agregue (+=) a lo anterior, también la variable de entorno DISPLAY a su valor establecido antes del reset.

Reglas de acceso

Aunque no es obligatorio declarar alias, ni opciones (defaults), y de hecho tampoco reglas de acceso, pues el archivo /etc/sudoers no tendría ninguna razón de ser si no se crean reglas de acceso. De hecho podríamos concretarnos a crear solamente reglas de acceso, sin opciones ni alias y podría funcionar todo muy bien.

Las reglas de acceso definen que usuarios ejecutan que comandos bajo que usuario y en que equipos. La mejor y (según yo, única manera) de entender y aprender a configurar sudoers es con ejemplos, así que directo al grano:

usuario host = comando1, comando2, ... comandoN

Sintaxis básica, 'usuario' puede ser un usuario, un alias de usuario o un grupo (indicado por %), 'host' puede ser ALL cualquier equipo, un solo equipo, un alias de equipo, una dirección IP o una definición de red IP/máscara, 'comandox' es cualquier comando indicado con su ruta completa. Si se termina en '/' como en /etc/http/ entonces indica todos los archivos dentro de ese directorio.

daniela ALL = /sbin/iptables

Usuario 'daniela' en cualquier host o equipo puede utilizar iptables.

ADMIN ALL = ALL

Los usuarios definidos en el alias 'ADMIN' desde cualquier host pueden ejecutar cualquier comando.

%gerentes dbserver = (director) /usr/facturacion, (root) /var/log/*

Un ejemplo más detallado. Los usuarios que pertenezcan al grupo del sistema llamado 'gerentes' pueden en el equipo llamado 'dbserver' ejecutar como si fueran el usuario 'director' la aplicación llamada 'facturacion', además como usuarios 'root' pueden ver el contenido de los archivos que contenga el directorio /var/log.

Lo anterior introduce algo nuevo, que en la lista de comandos es posible indicar bajo que usuario se debe ejecutar el permiso. Por defecto es el usuario 'root', pero no siempre tener que así. Además la lista 'hereda' la primera definición de usuario que se indica entre paréntesis (), por eso si se tiene más de alguno hay que cambiar de usuario en el comando conveniente, el ejemplo anterior también sería válido de la siguiente manera:

```
%gerentes dbserver = /var/log/*, (director) /usr/facturacion
```

No es necesario indicar (root) ya que es el usuario bajo el cual se ejecutan los comandos por defecto. También es válido usar (ALL) para indicar bajo cualquier usuario. El ejemplo siguiente da permisos absolutos.

sergio ALL = (ALL) ALL

Se establece permiso para el usuario 'sergio' en cualquier host, ejecutar cualquier comando de cualquier usuario, por supuesto incluyendo los de root.

SUPERVISORES PRODUCCION = OPERACION

Una regla formada solo por alias. En el alias de usuario 'SUPERVISORES' los usuarios que estén indicados en ese alias, tendrán permiso en los equipos definidos en el alias de host 'PRODUCCION', de ejecutar los comandos definidos o listados en el alias de comandos 'OPERACION'.

En este último ejemplo se aprecia lo útil que pueden ser los alias, ya que una vez definida la regla, solo debemos agregar o eliminar elementos de las listas de alias definidos previamente. Es decir, se agrega un equipo más a la red, se añade al alias 'PRODUCCION', un usuario renuncia a la empresa, alteramos el alias 'SUPERVISORES' eliminándolo de la lista, etc.

checo ALL = /usr/bin/passwd *, !/usr/bin/passwd root

Este es un ejemplo muy interesante de la potencia y flexibilidad . Al usuario 'checo', desde cualquier equipo, tiene permiso de cambiar la contraseña de cualquier usuario (usando el comando 'passwd'), excepto '!' la contraseña del usuario 'root'. Lo anterior se logra mediante el uso de argumentos en los comandos. En el primer ejemplo '/usr/bin/passwd *' el asterisco indica una expansión de comodín (wildcard) que indica cualquier argumento, es decir, cualquier usuario. En el segundo caso '!/usr/bin/passwd root', si indica un argumento específico 'root', y la '!' como ya se sabe indica negación, negando entonces el permiso a cambiar la contraseña de root.

Cuando se indica el comando sin argumentos: /sbin/iptables lo interpreta como 'puede usar iptables con cualquiera de sus argumentos'.

marijose ALL = "/sbin/lsmod"

Al estar entre comillas dobles un comando, entonces lo interpreta como 'puede hacer uso del comando lsmod pero sin argumentos'. En este caso el usuario 'marijose' podrá ver la lista de módulos del kernel, pero solo eso.

Tags (etiquetas de comandos)

Cuando se definen reglas, en la lista de comandos, estos pueden tener cero (como en los ejemplos anteriores) o más tags. Existen 6 de estas etiquetas o tags,

NOPASSWD Y PASSWD

Por defecto requiere que cualquier usuario se identifique o auténtique con su contraseña.

Aprendimos en la sección de 'Opciones' o 'Defaults' que es posible indicar que un usuario o alias de usuario no requiera de autenticación. Pero el control granular propio de , permite ir aun más lejos al indicar a nivel de comandos, cuáles requieren contraseña para su uso y cuáles no.

gerardo webserver = NOPASSWD: /bin/kill, /usr/bin/lprm, /etc/httpd/conf/

Usuario 'gerardo' en el equipo 'webserver' no requerira contraseña para los comandos listados. El tag se hereda, es decir no solo el primer elemento de la lista de comandos, sino los subsiguientes. Suponiendo que el último '/etc/httpd/conf/' elemento, que permite modificar cualquier archivo contenido en el directorio, si deseamos que use contraseña, lo siguiente lo conseguirá:

gerardo webserver = NOPASSWD: /bin/kill, /usr/bin/lprm, PASSWD: /etc/httpd/conf/

Aunque ya que solicitar contraseña es el default o defecto preestablecido, lo anterior también funcionará de la siguiente manera:

gerardo webserver = /etc/httpd/conf/, NOPASSWD: /bin/kill, /usr/bin/lprm,

NOEXEC Y EXEC

Este es un tag muy importante a considerar cuando sobre se otorgan permisos sobre programas que permiten escapes a shell (shell escape), como en el editor 'vi' que mediante el uso de '!' es posible ejecutar un comando en el shell sin salir de 'vi'. Con el tag NOEXEC se logra que esto no suceda, aunque no hay que tomarlo como un hecho, ya que siempre existe la posibilidad de vulnerabilidades no conocidas en los múltiples programas que utilizan escapes a shell. Al igual que los tags anteriores, el tag se hereda y se deshabilita con su tag contrario (EXEC), en caso de que en la lista de comandos hubiera varios comandos.

valeria ALL = NOEXEC: /usr/bin/vi

SETENV Y NOSETENV

Una de las múltiples opciones que pueden establecerse en la sección 'Defaults' u 'opciones' es la opción booleana o de flag 'setenv' que por defecto y para todos los usuarios esta establecida en 'off'. Esta opción si se activa por usuario (Defaults:sergio setenv) permitirá al usuario indicado cambiar el entorno de variables del usuario del cual tiene permisos de ejecutar comandos, y como generalmente este es 'root' pues es obvio que resulta bastante peligrosa esta opción. A nivel de lista de comandos, es posible entonces especificar el tag 'SETENV' a un solo comando o a una pequeña lista de estos y solo cuando se ejecuten estos se podrán alterar su entorno de variables. Es decir, en vez de establecerlo por usuario, sería mas conveniente establecerlo por comando a ejecutarse solamente.

ADMIN ALL = SETENV: /bin/date, NOSETENV ALL

A los usuarios definidos en el alias de usuario 'ADMIN' en cualquier host, pueden alterar las variables de entorno cuando ejecuten el comando 'date' (que puede ser útil por ejemplo para cambiar variables del tipo LOCALE), y cualquier otro comando, no tendrá esta opción al habilitar el tag contrario 'NOSETENV'. Y ya que este es el default, también sería válido de la siguiente manera y harían lo mismo:

ADMIN ALL = ALL, SETENV: /bin/date

ARCHIVO /ETC/SUDOERS DE EJEMPLO

Para concluir este manual, veamos un pequeño ejemplo de un archivo /etc/sudoers:

```
# ****
# LinuxTotal.com.mx, ejemplo de un archivo sudoers
```

```

# sergio.gonzalez.duran@gmail.com
# ****
# ****
# DEFINICION DE ALIAS
# ****

# administradores con todos los privilegios
User_Alias ADMINS = sergio, ana

# administradores de red - network operators
User_Alias NETOPS = marcela, andrea

# webmasters -
User_Alias WEBMAS = cristina, juan

# supervisores de producción (todos los del grupo de sistema supervisores)
User_Alias SUPPRO = samuel, %supervisores

# usuarios que pueden conectarse desde Internet
User_Alias INETUS = NETOPS, ADMINS, samuel

# servidores web
Host_Alias WEBSERVERS = 10.0.1.100, 10.0.1.101

# servidores de aplicaciones
Host_Alias APPLICACIONES = WEBSERVERS, 10.0.1.102, 10.0.1.103, mailserver

# comandos de red permitidos
Cmnd_Alias REDCMDS = /sbin/ifconfig, /sbin/iptables

# comandos de apache
Cmnd_Alias APACHECMDS = /usr/sbin/apachectl, /sbin/service httpd *

# ****
# DEFINICION DE OPCIONES
# ****

# Los usuarios administradores, requieren autenticarse con la contraseña de
# 'root'
Defaults>ADMINS rootpw

# Para todos los usuarios, tienen hasta dos intentos para ingresar su contraseña
# y 3 minuto para que esta expire
Defaults passwd_tries = 4, passwd_timeout = 1

# Los usuarios que se conectan desde Internet, solo tienen una oportunidad y
# cero timeout lo que implica
# que cada comando que usen a través de sudo requerira siempre de
# autenticación.
Defaults:INETUS passwd_tries = 1, passwd_timeout = 0

# Máscara de directorios y archivos por default, para los que ejecuten sudo en
# los servidores web
Defaults@WEBSERVERS umask = 022

# ****
# DEFINICION DE REGLAS
# ****

```

```
# administradores todo se les permite en cualquier equipo (||||cuidado con esto
en la vida real!!!!!
ADMINS ALL = (ALL) ALL

# administradores de red, en todos los equipos, los comandos de red
NETOPS ALL = REDCMDS

# webmasters, en los servidores web con los comandos indicados en apachecmds y
además sin necesidad
# de contraseña acceder a las bitácoras de apache y reiniciar los servidores.
WEBMAS WEBVERS = APACHECMDS, NOPASSWD: /var/log/apache/, /sbin/reboot

# supervisores, pueden ejecutar los comandos indicados en los equipos indicados
en el alias
# aplicaciones y además son ejecutados bajo el usuario apps.
SUPPRO APPLICACIONES = NOEXEC: (apps) /usr/local/facturacion.exe,
/usr/local/ventas.exe, /usr/local/nomina.exe

# no definidos por alias previos, sino directamente

# regina es de recursos humanos y puede cambiar contraseñas de cualquier usuario
menos de root
regina ALL = /usr/bin/passwd *, !/usr/bin/passwd root

# david, puede apagar los equipos de aplicaciones
david APPLICACIONES = /sbin/shutdown, /sbin/halt

# El equipo firewall de la red puede ser reiniciado (no apagado) por fernanda
que es asistente de redes
fernanda firewall = /sbin/shutdown -r now
```

107.2. Automatizar y programar tareas administrativas del sistema.

Peso en el examen de certificación: 4 puntos.

Objetivo: Utilizar cron or anacron para ejecutar trabajos a intervalos de tiempo regulares y utilizar at para ejecutar trabajos en un momento determinado.

Conceptos y áreas de conocimiento:

- Mantenimiento de trabajos con at y cron.
- Configurar el acceso de usuario a los servicios cron y at.

Términos y utilidades:

- /etc/cron
- /etc/at.deny
- /etc/at.allow
- /etc/crontab
- /etc/cron.allow
- /etc/cron.deny
- /var/spool/cron/*
- crontab
- at
- atq
- atrm

107.2. Automatizar y programar tareas administrativas del sistema.

Existe una gran cantidad de tareas que un sistema Linux debe ejecutar regularmente, como por ejemplo la limpieza de los ficheros temporales, la administración de los ficheros de log, copias de seguridad, actualización de bases de datos, etc.

Todas estas tareas se automatizan en Linux usando los comandos **cron** y **at**. Ambos comandos pueden ejecutar tareas programadas regularmente.

El comando **cron** se utiliza para ejecutar las conocidas como tareas cron, que pueden ser ejecutadas de forma periódica, es decir, repitiéndose la misma tarea cada cierto periodo.

El comando **at** se utiliza para programar el momento en que queremos que se ejecuta una tarea una única vez.

107.2.1. Comando at.

Un usuario que quiera definir una tarea o trabajo para que se ejecute en un momento determinado puede utilizar el comando **at**. Este ofrece numerosas posibilidades.

Los comandos que componen el trabajo por defecto se toman de la entrada estándar, una vez invocado el comando, o de un fichero utilizando la opción **-f**, el camino a este fichero debe estar especificado a partir del directorio *home* del usuario.

```
$ at -f ordenes.sh
```

En su uso normal (sin la opción **-f**) basta con indicarle al comando el momento en que queremos que se ejecute la tarea. El sistema responderá con su propio prompt **at>** y podremos escribir los comandos que componen la tarea y pulsar **Control-D** para finalizar la introducción.

```
$ at 15:00  
at> cp /home/usuario/log.txt /copiaseguridad
```

El momento en que queremos ejecutar la tarea se puede pasar de varias formas:

- **Una hora:** Podemos especificar la hora como *HH:MM*, seguida opcionalmente de *AM* o *PM*.
- **Una palabra clave:** *noon* (mediodía), *midnight*(medianocche), *teatime* (16:00), *now* (ahora).
- **Un dia especificado:** Podemos especificar la fecha completa con el formato *MMDDAA*, *MM/DD/AA* o *DD.MM.AA*. También podemos indicar el nombre del mes o el nombre del día.
- **Un periodo futuro especificado:** Podemos usar el signo de la suma y un periodo temporal, como por ejemplo *now + 4* o *13:00 + 5 days*.

El comando **at** depende de la ejecución del demonio **atd**, que evidentemente tiene que estar en funcionamiento para que el comando **at** pueda funcionar.

Podemos utilizar con **at** varias herramientas, como **atq** que nos lista las tareas **at** pendientes cada una con un número identificativo, **atrm** que elimina una tarea **at** de la cola introduciendo su número identificativo, y **batch** que funciona de un modo muy similar a **at**, pero ejecuta las tareas cuando el nivel de carga del sistema baja del 0,8.

La salida estándar y de errores de estas tareas se envía a través del correo local al usuario correspondiente a menos que este las redireccione utilizando los operadores correspondientes. El administrador del sistema puede especificar qué usuarios pueden o no utilizar el comando **at** en los ficheros */etc/at.allow* y */etc/at.deny*. Si estos ficheros no existen, solo el usuario root podrá utilizar el comando **at**.

107.2.2. Comando cron.

El comando para realizar tareas periódicas es **cron**. Este comando es un demonio que se está ejecutando en segundo plano continuamente. Cada minuto, examina los ficheros de configuración en los directorios */var/spool/cron* y */etc/cron.d* además del fichero */etc/crontab* y ejecuta las tareas que se encuentre en ellos si la hora del sistema corresponde con la indicada en las mismas.

Nos podemos encontrar con tareas **cron** tanto de sistema como de usuario. Las de sistema se utilizan para tareas de mantenimiento y siempre se ejecutan como si fueran lanzadas por el usuario **root**. Los usuarios también pueden crear sus propias tareas **cron** que serán lanzadas con los privilegios del usuario que las crea.

Estas tareas **cron** no pueden interactuar con el usuario, por lo que no deben incluir comandos que dependan de la entrada de un usuario.

107.2.2.1. Tareas cron del sistema.

Las tareas **cron** del sistema se especifican en el fichero */etc/crontab*. Al comienzo de este fichero nos podemos encontrar con varias líneas que definen variables de entorno y posteriormente líneas que especifican las tareas periódicas a realizar.

Básicamente para cada tarea periódica se escribe una línea donde se especifican los momentos en

que se ejecutará el comando, el usuario bajo el que se ejecutará y el comando correspondiente. Las fechas se especifican utilizando cinco indicadores separados por espacios. A continuación se escribe el usuario y el comando a ejecutar:

- **Minutos:** Oscila entre 0 y 59
- **Hora:** Oscila entre 0 y 23
- **Día del mes:** Oscila entre 1 y 31
- **Mes:** Oscila entre 1 y 12 (se pueden poner también las tres primeras letras del nombre del mes en inglés)
- **Día de la semana:** Oscila entre 0 y 7 (0 y 7 corresponden al domingo, también se pueden usar las tres primeras letras del nombre del día en inglés).

Tras estos 5 campos que indican el intervalo de ejecución de la tarea, la línea continúa con:

- **Usuario:** Nombre de la cuenta de usuario a utilizar cuando se ejecute el programa.
- **Comando a ejecutar:** El comando que se ejecutará.

En cualquiera de los campos que indican la periodicidad de la tarea se pueden especificar algunos valores especiales:

- **Asterisco:** coincide con todos los valores posibles del campo.
- **Lista separada por comas:** Coincide con cualquier de los valores indicados.
- **Rango:** Dos valores separados por un guión, coincide con el rango entre dichos valores, ambos inclusive.
- **Barra:** Valor escalonado.

Algunos ejemplos:

```
10 4 * * 0 root copiaseg.sh
```

- Minutos: 10
- Hora: 4
- Dia del mes: Cualquiera
- Mes: Cualquiera
- Dia de la semana: 0

Es decir, todos los domingos a las 4 y 10 de la madrugada se ejecutará *copiaseg.sh* con los permisos de **root**.

```
0 5 1,15,30 * * root aviso.sh
```

Todos los días 1, 15 y 30 de cada mes a las 5 de la madrugada se ejecutará *aviso.sh* con los permisos de **root**.

```
*/10 * * May 3 usuario script.sh
```

Todos los miércoles de mayo se ejecutará *script23.sh* cada 10 minutos con los permisos de **usuario**

```
02 4 * * * root run-parts /etc/cron.daily
```

Todos los días a las 4 horas 2 minutos se ejecutará el comando **run-parts** con el parametro */etc/cron.daily*. Las entradas en */etc/crontab* por defecto utilizan, normalmente, **run-parts,crondloop** o una utilidad similar que inicia cualquier script ejecutable contenido en un directorio. Por consiguiente, el ejemplo anterior ejecuta todos los scripts de */etc/cron.daily* a las 4:02 A.M. de cada día.

Las tareas del sistema se suelen organizar en cuatro grupos: tareas que se ejecutan cada hora, tareas diarias, tareas semanales y tareas mensuales. A cada uno de estos grupos le corresponde un

directorio: `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` y `/etc/cron.monthly` respectivamente. Estos directorios cuentan con tareas periódicas como las vistas en el ejemplo anterior.

Si queremos añadir una tarea periódica al sistema, podemos crear un script que realice dicha tarea y copiar simplemente dicho script al directorio `/etc/cron` correspondiente.

Si queremos añadir una tarea periódica en un intervalo de tiempo que no sea uno de los indicados anteriormente, podemos crear una tarea **cron** del usuario. Si el usuario que crea dichas tareas es **root**, las tareas se ejecutan con las mismas propiedades que una tarea del sistema.

107.2.2. Tareas cron del usuario.

El sistema cuenta con el comando **crontab**, que nos permite crear tareas periódicas o tareas **cron** para los usuarios. Este comando se utiliza de la siguiente manera:

crontab [-u usuario] [-l | -e | -r] [fichero]

- **-u usuario**: crontab modifica la tarea cron asociada al usuario indicado. Si no se especifica, crontab modifica la tarea cron asociada al usuario actual. Estas tareas cron del usuario se suelen denominar crontabs.
- **-l** : Lista las tareas cron del usuario.
- **-e** : Edita las tareas del usuario.
- **-r** : Elimina las tareas del usuario.
- **fichero**: Utiliza dicho fichero como fichero de tareas para el usuario.

Cada tarea se especifica mediante una línea, exactamente igual que hemos visto con las tareas **cron** del sistema. La única diferencia es que en las tareas **cron** del usuario no se indica el usuario. Por lo tanto, las líneas para definir las tareas cron de los usuarios tienen los siguientes campos:

- **Minutos**: Oscila entre 0 y 59
- **Hora**: Oscila entre 0 y 23
- **Día del mes**: Oscila entre 1 y 31
- **Mes**: Oscila entre 1 y 12 (se pueden poner también las tres primeras letras del nombre del mes en inglés)
- **Día de la semana**: Oscila entre 0 y 7 (0 y 7 corresponden al domingo, también se pueden usar las tres primeras letras del nombre del día en inglés).
- **Comando a ejecutar**: El comando que se ejecutará.

Los ficheros de tareas **cron** se almacenan en los directorios `/var/spool/cron`, `/var/spool/cron/tabs` o `/var/spool/cron/crontabs`. Cada fichero de este directorio recibe el nombre del usuario con el que se ejecute. Estos ficheros no se deben modificar, es preferible utilizar **crontab** para gestionarlos.

Existen los ficheros `/etc/cron.allow` y `/etc/cron.deny` que nos sirven para indicar listas de usuarios a los que se permite utilizar tareas **cron** (allow) y a los que se deniega el uso de tareas **cron**(deny).

107.2.3. Comando anacron.

Las tareas **cron** están diseñadas para usarse en equipos servidores, equipos que no se suelen apagar nunca, y por lo tanto realizarán las tareas del sistema aunque éstas se especifiquen a altas horas de la madrugada. Sin embargo en un ordenador de escritorio estas tareas no se suelen llegar a ejecutar; ya que la máquina se apaga frecuentemente. Para solucionar esto podemos utilizar la utilidad **anacron**.

Esta utilidad crea un registro con todas las tareas periódicas del sistema, y comprueba las últimas ejecuciones de dichas tareas. Si comprueba que una tarea no se ha llevado a cabo en su momento, la ejecuta en cuanto que pueda, de modo que se asegura que todas las tareas se van ejecutando sin

importar si la máquina se apaga o no.

Al igual que **cron**, **anacron** se controla a través de un fichero de configuración */etc/anacrontab*. Este fichero puede contar con líneas de comentario (#), líneas para asignar valores a variables y líneas para las tareas **anacron**. Cada una de estas líneas de tarea consta de 4 campos:

- **periodo**: Frecuencia, en días, con la que se debe ejecutar el comando.
- **retardo**: Periodo, en minutos, que transcurren entre el inicio de anacron y la ejecución del comando.
- **identificador**: Cadena que identifica el comando.
- **comando**: La tarea que se desea ejecutar.

Por ejemplo, la siguiente línea **anacron**:

```
1 5 tareas.diarias run-parts /etc/cron.daily
```

Ejecutaría cada día, después de 5 minutos de espera, el comando **run-parts** sobre el directorio */etc/cron.daily*. Esta tarea se identifica como *tareas.diarias*.

Hay que tener en cuenta que **anacron** no es un demonio como **cron**. Esto quiere decir que no es un comando que se quede residente en memoria ejecutándose continuamente. Esto implica que tendremos que ejecutar el comando **anacron** para que realice su función. Esto se puede conseguir de varias formas:

- **Script de inicio**: Podemos crear un script de inicio que ejecute anacron, bien mediante SysV o bien colocando una línea en los scripts de inicio local, como */etc/rc.d/rc.local* o */etc/boot.d/boot.local*.
- **Creando una tarea cron**: Podemos crear una tarea cron que ejecute regularmente anacron.

107.3. Localización e Internacionalización.

Peso en el examen de certificación: 3 puntos.

Objetivo: Adaptar un sistema a las características locales de un lenguaje diferente al inglés.

Además, la comprensión de porqué LANG = C es útil cuando se realiza scripting

Conceptos y áreas de conocimiento:

- Inicialización de Locale.
- Inicialización de Timezone.

Términos y utilidades:

- /etc/timezone
- /etc/localtime
- /usr/share/zoneinfo
- Environment variables
- /usr/bin/locale
- tzselect
- tzconfig
- date
- iconv
- UTF-8
- ISO-8859
- ASCII
- Unicode

107.3. Localización e internacionalización

Es posible configurar el sistema GNU/Linux para que pueda ser usado en una gran cantidad de regiones, cada una de ellas con su propio conjunto de caracteres, formatos de visualización de fecha y hora, símbolos monetarios, etc. Estas características se suelen configurar en la instalación del sistema, pero también pueden ser modificadas posteriormente.

107.3.1. Modificar zona horaria

Internamente en GNU/Linux se utiliza UTC (Universal Time Coordinated) que es la hora de Greenwich sin aplicar horario de verano. Todas las marcas temporales de los ficheros se guardan en GNU/Linux usando este sistema, pero podemos indicar al sistema que cuando nos presente o pida una hora no utilice UTC, sino que nos la presente en otro formato escogido por nosotros.

La zona horaria que se debe utilizar por parte del sistema se establece en el fichero */etc/localtime*, que no es un fichero de texto puro, por lo que no debe ser editado directamente. Dependiendo de la distribución GNU/Linux, */etc/localtime* puede ser el fichero directamente, o bien ser un enlace simbólico a otro fichero. La mayoría de las distribuciones actuales utilizan el sistema del enlace simbólico.

Para comprobar la zona horaria actual del sistema, podemos utilizar el comando **date**.

```
$ date
```

jue ene 26 12:21:25 CET 2012

Vemos como después de la fecha y la hora, el sistema nos indica con 3 caracteres la zona horaria actual del sistema, en este ejemplo CET. Podemos consultar el significado de estos caracteres accediendo a <http://www.timeanddate.com/library/abbreviations/timezones/>. Vemos como CET por ejemplo indica Central European Time UTC+1.

Si queremos cambiar esta zona horaria, debemos enlazar el fichero */etc/localtime* a otro fichero. Si accedemos a */usr/share/zoneinfo/* comprobaremos cómo tenemos una gran cantidad de ficheros de zonas horarias para elegir. Una vez encontrada la nuestra, que puede que esté dentro de un subdirectorio, tenemos que borrar el fichero */etc/localtime* y crearlo de nuevo como un enlace simbólico al fichero deseado. Por ejemplo:

```
# rm /etc/localtime  
# ln -s /usr/share/zoneinfo/Europe/Madrid /etc/localtime
```

Además de */etc/localtime* algunas distribuciones utilizan otro fichero */etc/timezone* o */etc/sysconfig/clock*. Este fichero suele contener nuestra zona horaria como una línea de texto, como por ejemplo */America/Los_Angeles*. Este fichero también debe de ser modificado cuando cambiamos */etc/localtime* para evitar errores con algunas utilidades del sistema.

Normalmente las distribuciones incluyen sus propios programas que nos permiten cambiar la zona horaria sin tener que crear nosotros los enlaces manualmente ni tener que modificar posteriormente */etc/timezone*. Estos programas son ***tzsetup***, ***tzselect*** o ***tzconfig***, dependiendo de la distribución GNU/Linux con la que contemos. Es posible que estos cambios no sean permanentes, teniendo que añadir algunas líneas al fichero de configuración correspondiente de nuestro perfil para que los cambios se mantengan entre inicios de sesión.

Si nuestra distribución está basada en Debian, también podemos reconfigurar la zona horaria mediante el comando

```
# dpkg-reconfigure tzdata
```

107.3.2. Configuraciones locales en GNU/Linux.

En GNU/Linux, un ajuste o configuración local (locale) es un modo de especificar el idioma y el país con el que queremos configurar nuestro sistema. Esto nos permite cambiar muchos aspectos del sistema relacionados con la visualización internacional. Cada ajuste local o locale se presenta de la siguiente forma:

[idioma[_región]][.conjunto-de-códigos][@modificador]

Así por ejemplo nuestro sistema puede estar configurado como

`es_ES:UTF-8`

Estos nos indica que nuestro idioma es castellano (es), la región España (ES) y se usa el conjunto de caracteres UTF-8.

El conjunto de códigos indica la codificación que se utiliza para representar cada uno de los caracteres del idioma. Algunos conjuntos de códigos son:

- **ASCII:** (American Standard Code for Información Interchange) es el método de codificación más antiguo. Usa 7 bits para representar cada uno de los caracteres. Es adecuado para el idioma inglés, pero no tiene caracteres para el resto de idiomas.

- **ISO-8859:** Es una ampliación de ASCII, utilizando 8 bits para almacenar cada uno de los caracteres. Sí incluye caracteres especiales para otros idiomas distintos al inglés. Existen una gran cantidad de versiones de esta codificación, por ejemplo ISO-8859-1 o ISO-8859-5, cada una de ellas pensadas para un idioma en concreto.
- **UTF-8:** Es el método de codificación de caracteres más moderno, que permite utilizar de 8 hasta 32 bits por cada carácter, lo que permite que en un mismo método de codificación podamos encontrar todos los caracteres deseados de cualquier idioma.

El modificador nos indica opciones especiales del conjunto local, como puede ser una ordenación especial que haya que llevar a cabo.

Los conjuntos locales del sistema están asignados a unas variables del entorno. Para ver estas variables y sus valores usamos el comando **locale**. La mayoría de estas variables de ajuste local definen características obvias como LC_TIME,LC_NUMERIC o LC_PAPER. Hay que tener en cuenta que si colocamos un valor en la variable LC_ALL se utiliza para todas las demás variables, sin importar el valor que tengamos en las mismas.

Veamos algunas de las variables locale más utilizadas.

- **LC_COLLATE:** Nos permite indicar el tipo de ordenación que deseamos. Si indicamos un código local español conseguiremos que la letra ñ se ordene entre la n y la o.
- **LC_CTYPE:** Cambia el tipo de carácter, usado por ejemplo para las funciones que comprueban si un carácter está en mayúsculas o no.
- **LC_MONETARY:** Cambia la forma de representar cantidades, indicando por ejemplo si se usa punto o coma para los decimales.
- **LC_MESSAGES:** Cambia el lenguaje en que los mensajes se escriben en pantalla, y le indica al sistema como se debe representar una respuesta afirmativa o negativa.
- **LC_NUMERIC:** Cambia la forma de representar números.
- **LC_TIME:** Cambia la forma en la que el sistema representa las fechas. Por ejemplo en la mayoría de Europa se utiliza el reloj de 24 horas, mientras que en Estados Unidos se usa el de 12.
- **LC_ALL:** El valor que se indique aquí, se utilizará como valor por defecto para todas las variables anteriores. En caso de que una variable tenga su propio valor definido, no se tendrá en cuenta el valor de LC_ALL.

El comando **locale** permite utilizar el parámetro **-a** que nos indica todos los ajustes locales posibles que se pueden asignar en nuestro sistema. Vemos como nos aparecerán ajustes para varios idiomas, y un ajuste que es el utilizado por defecto en sistemas de tipo POSIX, que veremos representado como POSIX o C.

Otra variable de entorno relacionada es LANG, que definen los ajustes locales pero únicamente si las variables LC_ no se han definido.

107.3.2.1. Cambiar el ajuste local.

Para cambiar el ajuste local en primer lugar debemos comprobar si existe uno apropiado ejecutando el comando **locale -a**. Si no aparece el código deseado tendremos que instalar paquetes adicionales en el sistema, y el nombre de los mismos depende de cada distribución GNU/Linux, aunque normalmente los nombres de dichos paquetes contendrán las palabras *locale* o *language*. En el caso de Ubuntu por ejemplo estos paquetes se denominan *language-support-XX* donde XX es el código del idioma con dos caracteres.

Para cambiar temporalmente un ajuste local se puede modificar directamente la variable de entorno LC_ALL. Se recomienda modificar también la variable de entorno LANG.

```
$ export LANG=es_CO.UTF-8  
$ export LC_ALL=es_CO.UTF-8
```

En estas líneas hemos cambiado la configuración local del sistema a castellano del país Colombia y codificación de caracteres UTF-8. Este cambio es válido únicamente para la sesión actual en la consola actual, y afecta a cualquier programa que se ejecute en dicha consola. Programas que ya estuvieran ejecutándose o bien que se inicien en otra consola no se verán afectados por esta configuración.

Si queremos configurar el ajuste local del sistema permanentemente, podemos editar los ficheros de scripts de inicio (como `~/.bashrc` o `/etc/profile`) y añadir las líneas **export** anteriores en el mismo script, de forma que se ejecuten siempre que se abre sesión.

Si queremos configurar el ajuste local en el entorno gráfico para el uso del teclado, configuraremos en el fichero de configuración de `x.org` la opción `XkbLayout` en la sección `InputDevice` del teclado. El valor que se da a dicha opción es el mismo que se da a los ajustes locales, pero siempre en minúsculas. Tendremos que reiniciar el servidor gráfico para que estos cambios tengan efecto.

Hay un valor de parámetro que requiere una mención especial: `LANG=C`. Cuando definimos la variable `LANG` con el valor `C`, los programas que ven esta variable de entorno muestran una salida que no ha pasado por el filtro de las traducciones de los ajustes locales. Esto puede resultar útil en algunos casos en los que el ajuste local corrompe la salida de un programa; por ejemplo, cuando las conversiones a UTF-8 cambian los caracteres que se deberían preservar como entidades de 8 bits. Por consiguiente, si definimos `LANG=C` podemos evitarnos algunos problemas, en particular en las canalizaciones y scripts que se pasan los datos de un programa a otro en formato binario.

Hay que tener en cuenta también que, aunque configuremos el ajuste local del sistema, es responsabilidad del programador de cada programa que dichos ajustes sean tenidos en cuenta o no. Nos encontraremos, por tanto, con muchos programas donde estos cambios no tendrán efecto o donde tendremos que usar la propia configuración del programa para indicar el ajuste local que deseamos usar en el mismo.

107.3.2.2. Modificar los ajustes locales de ficheros de texto.

Nos podemos encontrar con ficheros de texto provenientes de sistemas GNU/Linux que utilizaban una configuración local distinta a la que usa nuestro sistema. Esto implica que veremos el contenido de dicho fichero con símbolos extraños, ya que la codificación que se empleó para almacenar los caracteres originales no coincide con la codificación que se emplea para mostrar dichos caracteres.

Para resolver este problema, podemos usar el comando **iconv**.

iconv -f `codificación-origen` [**-t** `codificación-destino`] [fichero]

iconv convierte el fichero indicado de la codificación origen a la codificación destino. Si omitimos la codificación de destino, **iconv** utilizará la codificación del sistema actual como tal. Este comando manda la información a la salida estándar, por lo que si queremos crear un fichero tendremos que reenviar la salida a dicho fichero.

```
$ iconv -f iso-8859-1 -t UTF-8 datos.txt > datos_utf.txt
```

Podemos obtener un listado de codificaciones posibles ejecutando el comando **iconv** con el parámetro **--list**.

108 SERVICIOS ESENCIALES.

- 108.1. Mantenimiento de la fecha y hora del sistema.
- 108.2. Configurar y recurrir a archivos de log.
- 108.3. Fundamentos de Mail Transfer Agent (MTA)
- 108.4. Configuración de impresoras e impresión.

108.1. Mantenimiento de la fecha y hora del sistema.

Peso en el examen de certificación: 3 puntos.

Objetivo: Mantener la hora y fecha del sistema de forma manual y/o sincronizada via NTP.

Conceptos y áreas de conocimiento:

- Inicializar la fecha y la hora del sistema.
- Inicializar el reloj hardware de forma correcta en UTC.
- Configurar correctamente la zona horaria.
- Configuración básica de NTP.
- Conocimiento en el uso del servicio pool.ntp.org.

Términos y utilidades

- /usr/share/zoneinfo
- /etc/timezone
- /etc/localtime
- /etc/ntp.conf
- date
- hwclock
- ntpd
- ntpdate
- pool.ntp.org

108.1.0. Introducción

El mantenimiento de la fecha y la hora en sistemas Linux es algo más importante de lo que puede parecer a primera vista. Podemos tomar como ejemplo las entradas en los logs del sistema que indican de forma precisa la fecha y la hora en la que se ha realizado cada acción y esta información puede ser de mucha utilidad para cualquier usuario del sistema (ya sea administrador o no).

Veremos diferentes formas de manejar la hora de un sistema Linux, ya sea de manera automática, mediante la sincronización con servidores NTP, o de manera manual.

108.1.1. Mantenimiento manual

El manejo de la fecha y hora en un sistema Linux disponemos de un variado conjunto de herramientas. Éstas nos van a permitir desde la visualización hasta la modificación de los valores de fecha y hora.

En cualquier sistema Linux debemos distinguir entre el reloj hardware y el reloj software, mantenido por el sistema. Para manipular este último, la herramienta por excelencia es el comando DATE. Sin embargo para trabajar con la hora hardware usaremos el comando HW CLOCK. Y si lo que queremos es visualizar el calendario usaremos el comando CAL. Vamos a ver estos tres comandos de una forma más detallada.

108.1.1.1. Comando date.

Este comando nos va a permitir trabajar con la hora software del sistema. Las opciones irán desde la mera visualización de la fecha hasta su modificación con diferentes formatos. Su sintaxis es la siguiente:

date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

Tabla 1

Opciones usadas con el comando Date

-d, --date=CADENA	Muestra la fecha descrita por CADENA
-f, c--file=FICHERO	Hace lo mismo que el parámetro anterior para cada línea de FICHERO
-r, --reference=FICHERO	Muestra la fecha de la última modificación de FICHERO
-R, --rfc-2822	Muestra la fecha y la hora en formato RFC 2822. Ejemplo: Wed, 28 Dec 2011 19:57:49 +0100
-s, --set=CADENA	Cambia la fecha a la descrita por CADENA
-u, --utc, --universal	Muestra o cambia la Fecha Universal Coordinada

La manera más sencilla de usar el comando **date** es sin parámetros, lo que nos permite ver la hora del sistema.

\$ date

Lo que nos daría un resultado como el siguiente: mié dic 28 20:07:10 CET 2011.

El comando Date, nos da la opción de mostrar la fecha seleccionando los campos que queremos visualizar, así como la forma de hacerlo. La sintaxis para realizar esto es:

date [opción...] [+FORMAT]

Tabla 2

Opciones de formato usadas con el comando Date

%a	Día de la semana abreviado
%A	Día de la semana completo
%b	Nombre del mes abreviado
%B	Nombre del mes completo
%m	Número del mes
%d	Día del mes
%H	Hora (en formato de 24h)
%M	Minutos
%S	Segundos

Por ejemplo, si lo que queremos es visualizar el día de la semana, día de mes y mes (completos), deberíamos utilizar la siguiente orden:

```
$ date +"%A %d %B"
```

Obteniendo el resultado:

Miércoles 28 diciembre

Pero si lo que deseamos es modificar la fecha y hora del sistema debemos usar los parámetros para ello. La manera más sencilla es usar la orden pasando una cadena de caracteres en la que indicamos la nueva fecha y hora. La cadena debe tener el siguiente formato:

```
MMDDhhmm[YYYY][.ss]
```

Tabla 3

Opciones usadas para modificar la fecha y la hora

MM	Representa las dos cifras del mes
DD	Representa las dos cifras del día del mes
hh	Representa las dos cifras de la hora (formato 24 horas)
mm	Representa las dos cifras para los minutos
[YYYY]	Representa el año con dos o cuatro cifras. Es opcional
[.ss]	Representa las dos cifras para los segundos. Es opcional

Por ejemplo si queremos modificar la fecha a 12/01/2010 a las 14:45h, la orden que deberíamos utilizar sería (como root):

```
#date 011214452010
```

Y si quisiésemos cambiarlo con precisión de segundos, el comando sufriría una pequeña modificación:

```
#date 011214452010.32
```

En este caso cambiaríamos la fecha y horas del sistema al mismo valor que en el comando anterior, pero añadiendo además que comenzamos a partir de segundo 32.

Por defecto, el cambio de la fecha se hace a nivel local. Pero también se puede modificar la hora universal UTC simplemente utilizando el parámetro -u. Por ejemplo podríamos usar la orden:

```
#date -u 011214452010
```

Nuestra hora local se ajustará en función de la diferencia horaria con el horario UTC (en el caso del horario peninsular es de 1 hora más de diferencia). Así, con el ejemplo indicado más arriba, la hora de nuestro sistema cambiaría a las 15:45h.

8.1.1.2. Comando hwclock.

La orden que debemos utilizar para gestionar el reloj hardware del ordenador desde nuestro sistema es hwclock. Evidentemente, esta es una tarea que se puede llevar a cabo desde los menús de configuración oportunos en la BIOS del sistema.

Con este comando podremos desde visionar el valor que tiene el reloj hardware del sistema, hasta modificarlo. Su sintaxis es la siguiente:

Tabla 4

Opciones usadas con el comando Hwclock

-r, --show	Muestra el valor de la fecha y hora hardware del sistema
--set --date=FECHA	Modifica el valor de la fecha hardware del sistema a partir de la fecha dada en el valor FECHA
--systohc	Sincroniza el reloj hardware del sistema con el reloj software
--hctosys	Sincroniza el reloj software del sistema con el reloj hardware

Lo primero que es necesario saber sobre este comando es que es necesario usarlo como administrador del sistema. La forma más sencilla de utilizarlo es escribiendo la orden sin ningún parámetro (o utilizando los parámetros -r o --show):

```
#hwclock  
#hwclock -r / #hwclock --show
```

El resultado que nos dará será algo parecido a lo siguiente:

```
dom 04 dic 2011 18:50:49 CET -0.876015 segundos
```

Si lo que deseamos es modificar el valor de la fecha del sistema tenemos varias opciones. Podemos modificar el reloj hardware de manera manual o de manera automática a partir del reloj software del sistema. Adicionalmente también tendremos la opción de modificar el reloj hardware a partir del reloj software. Vamos a ver como realizar cada uno de estos tres pasos.

La modificación manual del reloj hardware del sistema se hace utilizando los parámetros --set --date=FECHA, donde FECHA es el valor de la nueva fecha que queremos que tenga el reloj. Por ejemplo podríamos usar la orden siguiente:

```
#hwclock --set --date="02/01/2012 18:41:00"
```

Cambiaría la fecha del reloj hardware al 1 de Febrero de 2012 a las 18:41h. Para ver el resultado utilizamos el comando hwclock sin parámetros y nos daría la siguiente salida:

```
mié 01 feb 2012 18:41:00 CET -0.516535 segundos
```

Otra de las opciones que nos ofrece este comando es la posibilidad de sincronizarlo con el reloj software del sistema. Para ello el parámetro que usaremos será systohc:

```
#hwclock --systohc
```

El proceso inverso también podemos llevarlo a cabo con el comando hwclock. Podríamos sincronizar el reloj software con el reloj hardware. El parámetro que utilizaremos será hctosys:

```
#hwclock --hctosys
```

8.1.1.3. Comando cal.

El comando CAL es una orden Linux que nos va permitir visualizar el calendario de diferentes formas. La manera más sencilla de utilizar el comando es simplemente usando la orden sin parámetros:

```
$ cal
```

Esto nos daría una salida como la siguiente:

Enero 2012						
lu	ma	mi	ju	vi	sá	do
					1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

En la que, como podemos ver nos muestra el calendario del mes actual y remarca el día en el que nos encontramos. Pero el comando cal es bastante más potente que lo que acabamos de ver.

De hecho posee una serie bastante amplia de parámetros que permiten modificar su comportamiento. Los más importantes se muestran en la siguiente tabla:

Tabla 5

Opciones usadas con el comando cal

yyyy ó -y yyyy	Muestra el calendario del año completo indicado por sus cuatro cifras
-m xx[p/f]	Muestra el calendario del mes indicado por las dos cifras xx. Opcionalmente podemos añadir p ó f para indicar que queremos ver el último mes que fue como el indicado (p) o el siguiente (f)
-m xx yyyy	Muestra el calendario del mes indicado por las cifras xx y el año yyyy
-h	Hace que el día actual no aparezca remarcado
-3	Muestra el mes anterior, el actual y el posterior
-A n	Muestra los n meses posteriores al mes actual
-B n	Muestra los n meses anteriores al mes actual

Vamos a usar algunos de estos parámetros en ejemplos concretos. Por ejemplo si quisiésemos ver el calendario del año 1999, tendríamos que escribir la siguiente orden:

\$ cal 1999

ó bien

\$ cal -y 1999

Ambas mostrarían como resultado el calendario completo del año especificado, es decir los 12 meses.

1999											
Enero					Febrero					Marzo	
lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi
1	2	3	4	5	6	7	1	2	3	4	5
8	9	10	11	12	13	14	8	9	10	11	12
15	16	17	18	19	20	21	15	16	17	18	19
22	23	24	25	26	27	28	22	23	24	25	26
29	30	31					29	30	31		
Abril					Mayo					Junio	
lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi
1	2	3	4	5	6	7	1	2	3	4	5
10	11	12	13	14	15	16	10	11	12	13	14
17	18	19	20	21	22	23	17	18	19	20	21
24	25	26	27	28	29	30	24	25	26	27	28
30							31				
Julio					Agosto					Septiembre	
lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi
1	2	3	4	5	6	7	1	2	3	4	5
10	11	12	13	14	15	16	10	11	12	13	14
17	18	19	20	21	22	23	17	18	19	20	21
24	25	26	27	28	29	30	24	25	26	27	28
31							31				
Octubre					Noviembre					Diciembre	
lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi
1	2	3	4	5	6	7	1	2	3	4	5
8	9	10	11	12	13	14	8	9	10	11	12
15	16	17	18	19	20	21	15	16	17	18	19
22	23	24	25	26	27	28	22	23	24	25	26
29	30	31					29	30	31		

Si lo que queremos es concretar a nivel de mes tenemos varias opciones. Si queremos visualizar el calendario de un mes concreto del año actual usaremos el parámetro -m de la siguiente manera.

```
$ cal -m 04
```

Nos mostraría el calendario del mes de Abril del año actual.

La orden cal nos permite mostrar de una manera rápida y sencilla el último mes indicado que no es el del año actual (p) o el próximo mes indicado que no es el actual (f). Para añadiremos un modificador al final de la orden anterior. Así, para el el último mes igual al indicado escribiremos:

```
$ cal -m 04p
```

Y para próximo mes igual al escrito ejecutaremos la siguiente orden:

```
$ cal -m 04f
```

Sin embargo si lo que queremos es ver el calendario del mes de Abril del año 1931, lo que tendríamos que hacer escribir la siguiente orden:

```
$ cal -m 04 1931
```

Existen algunos parámetros más que podemos usar con el comando cal, como por ejemplo el parámetro -3, que nos va a mostrar el calendario del mes actual, del mes anterior y del posterior. Así, escribiendo la orden:

```
$ cal -3
```

8.1.2. Sincronización automática

El trabajo con la fecha y hora de sistema no se reduce sólo a un trabajo manual, que dependa

del usuario, sino que se puede establecer de una manera automática usando servidores remotos. Esto se hace a través del protocolo NTP (Network Time Protocol).

De una manera resumida, podemos decir que este protocolo sincroniza nuestro reloj de sistema con una serie de servidores de manera simultánea, lo que hace que la desviación que hay respecto a la hora sea mínimo (al nivel de los milisegundos). La sincronización se produce con la escala UTC (Coordinated Universal Time) que es un estándar de alta precisión de tiempo atómico.

Todo este proceso de sincronización se realiza a través de varias herramientas que nos permiten tanto configurar los servidores NTP como realizar el propio proceso de sincronización.

8.1.2.1. El fichero ntp.conf

En este fichero se guarda la configuración por defecto del protocolo NTP en nuestro sistema. En este fichero podremos encontrar una amplia serie de opciones para modificar el comportamiento del protocolo NTP, pero la configuración que más nos interesa se encuentra en las líneas del tipo:

```
server ntp.ubuntu.com  
server pool.ntp.org
```

Esa o esas líneas son las que indicarán los servidores ntp a los que nos vamos a conectar para sincronizar la hora del sistema.

Buscando en Internet podremos encontrar muchos servidores ntp fiables, como por ejemplos los siguientes:

- ntp.ubuntu.com
- pool.ntp.org (donde la hora española está en es.pool.ntp.org)
- hora.rediris.es (RedIris)
- hora.roa.es (Real Observatorio de la Armada)

8.1.2.2. El demonio ntpd

Configurar correctamente el fichero ntp.conf es sólo el paso inicial para poder sincronizar el reloj de nuestro sistema. Una vez realizado ese paso debemos utilizar el demonio ntpd para que la sincronización se haga efectiva.

En este podemos realizar varias tareas. Por un lado debemos asegurarnos de que el demonio ntpd se lanza al inicio del sistema. Esto podemos realizarlo a través del comando:

```
# chkconfig ntpd on
```

Si tenemos dudas sobre el estado del demonio (si está lanzado o parado), podemos comprobarlo a través de la orden:

```
# /etc/init.d/ntp status
```

Podremos iniciar el demonio con el comando siguiente:

```
# /etc/init.d/ntp start
```

Podremos reiniciar el demonio con el comando siguiente:

```
# /etc/init.d/ntp restart
```

Podremos parar el demonio con el comando siguiente:

```
# /etc/init.d/ntp stop
```

8.1.2.3. El comando ntpdate

La orden ntpdate nos va a permitir sincronizar la hora del sistema con el servidor o servidores de hora que queramos.

La sintaxis básica del comando es:

```
ntpdate [opciones] server [server...]
```

donde "server" es un servidor ntp que nos va a permitir sincronizar la hora. Las opciones más usuales con este comando son las siguientes:

Tabla 6

Opciones usadas con el comando ntpdate

-q	Permite realizar una consulta a los servidores sin cambiar la hora del sistema
-s	Permite que la salida del comando se almacene en syslog en lugar de mostrarlo por pantalla
-u	Se utiliza para realizar la sincronización a través de otro puerto que no sea el 123 (que es el que usa normalmente), ya que puede haber cortafuegos que no permitan el tráfico a través de esa vía.
-v	Devuelve el resultado de la operación con más detalles

Con esta información, la orden que deberíamos ejecutar para sincronizar manualmente los servidores sería:

```
# ntpdate pool.ntp.org ntp.ubuntu.com
```

Donde pool.ntp.org y ntp.ubuntu.com son dos servidores que hemos elegido para realizar el proceso. Debemos tener en cuenta que esta orden sólo puede ejecutarse con éxito si el demonio ntp está parado. Si no indica que el socket NTP está en uso. Para detenerlo debemos ejecutar la orden explicada anteriormente (aquella en la que pasábamos el comando "stop" al demonio ntpd).

8.1.3. Zona Horaria.

La zona horaria de nuestro sistema se encuentra almacenada en */etc/localtime* que puede ser un fichero o un enlace simbólico a algún elemento del directorio */usr/share/zoneinfo*.

La manera más sencilla de realizar esta configuración es crear un enlace simbólico al fichero de */usr/share/zoneinfo* que nos interese. Así, por ejemplo, lo normal para una configuración en nuestro país sería:

```
# ln -s /usr/share/zoneinfo/Europe/Madrid /etc/localtime
```

Como consejo, es adecuado realizar previamente a este paso una copia de seguridad del fichero */etc/localtime* por si lo necesitásemos en un futuro.

108.2. Configurar y recurrir a archivos de log.

Peso en el examen de certificación: 2 puntos.

Objetivo: Configurar el servicio de log del sistema (**syslog**), incluyendo la característica de enviar la información recogida por él a un servidor central o viceversa, es decir, hacer que reciba salidas de servicios de log remotos.

Conceptos y áreas de conocimiento:

- ficheros de configuración de syslog
- syslog
- facilidades estandar, prioridades y acciones.

Términos y utilidades

- syslog.conf
- syslogd
- klogd
- logger

108.2. Configurar y recurrir a archivos de log.

En un sistema Linux se producen muchos eventos que se deben registrar con fines administrativos. Para ello, el sistema mantiene las bitácoras (ficheros de log o ficheros de registro) que registran detalles claves sobre las operaciones que se realizan en él. El demonio encargado de gestionar los registros del sistema es **syslogd** (syslog daemon). Algunos servidores y otros programas llevan a cabo sus propios registros por lo que han de configurarse independientemente de syslogd.

Hay diferentes aplicaciones disponibles para Linux que implementan la funcionalidad de syslogd y ofrecen alguna característica adicional. Algunos ejemplos son rsyslogd (usada desde hace tiempo en distribuciones como Ubuntu o Fedora) y syslog-ng. Para el examen LPI nos centraremos en el servidor tradicional syslogd.

108.2.1. Funcionamiento de syslogd.

El encargado tradicional del registro del sistema es syslogd, que se suele instalar desde un paquete llamado **sysklogd**. El demonio syslogd gestiona los mensajes de los servidores y de aplicaciones de usuario. Se le suele comparar con el demonio **klogd** que gestiona el registro de los mensajes del kernel y que suele instalarse desde el mismo paquete. Ambos deben trabajar conjuntamente y de forma coordinada,

La idea fundamental de una herramienta de registro es proporcionar un medio unificado de gestionar los ficheros de registro. Syslogd se ejecuta en segundo plano y acepta los datos enviados desde los servidores y otros programas que estén configurados para usarlo. Después utilizará esa información para clasificar los mensajes y enviarlos a los archivos de registro adecuados.

La configuración de syslogd permite utilizar un grupo predefinido de archivos de log en donde se registrarán la mayoría de eventos del sistema. De esta manera diferentes servicios del sistema pueden usar un mismo fichero de registro. El resultado de esta configuración permite agrupar todos los mensajes de registro en unos pocos archivos de registro estándar, lo cual es más fácil que controlar decenas de ficheros de log provenientes de los distintos servidores del sistema.

108.2.2. Configuración de syslogd.

El archivo principal de configuración de syslogd es `/etc/syslog.conf` (`/etc/rsyslog.conf` para rsyslogd). El formato de las líneas de este fichero es:

recurso.prioridad acción

Recurso es una palabra clave que identifica el tipo de programa o herramienta que ha generado el mensaje de log. Puede tomar los siguientes valores:

- **Authpriv o auth o security:** Utilizado por las aplicaciones que gestionan las autorizaciones del sistema (PAM, login, su, sudo, etc.). Security es obsoleta, es decir, en futuras versiones se dejarán de utilizar y por tanto no es recomendable su uso.
- **Cron:** Utilizado para herramientas de gestión automática de tareas como cron o anacron.
- **Daemon:** Cajón de sastre. Los servidores que no entran en ningún otro recurso a menudo lo utilizan.
- **Kern:** Mensajes del núcleo.
- **Lpr:** Mensajes relacionados con la gestión de la impresión en el sistema.
- **Mail:** Servidores de correo y herramientas de procesamiento de correo.
- **Mark:** Reservado sólo para el uso interno de syslogd. No se puede usar para construir reglas.
- **News:** Servidores de noticias.
- **Syslog:** Mensajes generados por syslogd.
- **User:** Lo pueden utilizar las aplicaciones de usuario del sistema para mensajes a medida.
- **Uucp:** Subsistema UUCP (Unix-Unix CoPy).
- **Local0 a local7:** Reservado para usos específicos que le sean asignados.

Prioridad indica la importancia del mensaje. Valores posibles (ordenados de menor a mayor importancia):

- **Debug:** Muestra la máxima información posible y sólo se utiliza cuando se está probando una aplicación. No es recomendable activarlo en un sistema en explotación ya que puede afectar al rendimiento del sistema.
- **Info:** Mensajes de información.
- **Notice:** Son mensajes que no son necesariamente errores pero que se deberían tener en cuenta.
- **Warning o warn:** Avisos importantes que aunque no son errores pueden tener algún tipo de repercusión. Warn es obsoleto.
- **Err error:** Mensajes de error. Error es obsoleto.
- **Crit:** Mensajes de error importantes (como errores de hardware).
- **Alert:** Mensajes críticos de error que se deberían solucionar inmediatamente.
- **Emerg o panic:** Mensajes muy graves que normalmente implican que la máquina dejará de funcionar inmediatamente o ya ha dejado de funcionar. Panic es obsoleto.

Se pueden especificar múltiples recursos separándolos por comas (,). Un asterisco (*) referencia todas las recursos.

Cuando un programa envía un mensaje a syslogd, el mensaje incluye un código de prioridad. Syslogd sólo guarda el mensaje en los ficheros de log si la prioridad del mensaje enviado es superior o igual al umbral de prioridad mínimo con que configuramos syslogd. Es decir si indicamos una prioridad de error entonces se mostrarán los mensajes con prioridad:

err (o error)

crit

alert

emerg (o panic)

Se puede utilizar la exclamación (!) para invertir el significado normal. Así, si ponemos !crit estaremos indicando que se registren todos los mensajes con una prioridad inferior a crit.

Si utilizamos el símbolo igual (=) indicamos sólo mostrar los mensajes con esa prioridad (p. ej. = info sólo registrará mensajes de información).

Se puede hacer referencia a todas las prioridades por medio de un asterisco (*).

Se pueden indicar múltiples selectores (recurso.prioridad) para una misma acción separándolos por punto y coma (;).

Si en lugar de una de las prioridades indicadas anteriormente se usa la palabra none, no se tendrá en cuenta ninguna prioridad para el recurso en cuestión y por lo tanto no se registrará ningún mensaje que provenga de dicho recurso.

Acción representa el destino de mensajes que corresponden a un selector, es decir, al conjunto formado por un recurso más una prioridad (recurso.prioridad).

- La acción puede ser un nombre de archivo (incluyendo la ruta completa) del directorio */var/log* según indica el estándar FHS. Los archivos *messages*, *syslog* o *secure* de dicho directorio son tres archivos comunes e importantes, aunque hay distribuciones que no emplean los tres.
- También puede ser el nombre de una máquina remota (o su dirección IP) precedido por el símbolo arroba (@) para registrar los datos del sistema especificado. En la máquina remota debe arrancarse *syslogd* con la opción *-r*, así escuchará (por defecto usa el puerto 514 UDP) los mensajes entrantes que provienen de la red.
- Además, puede ser una lista con los nombres de los usuarios (separados por comas) que deberían ver los mensajes cuando estén conectados al sistema o un asterisco (*) para indicar que todos los usuarios conectados recibirán el mensaje en sus consolas.
- A veces se indican como archivos de log el nombre de ficheros de dispositivos de consola como */dev/console* o */dev/xconsole*. En este caso los mensajes se muestran en la pantalla.

Veamos algunos ejemplos:

mail . * /var/log/mail

Envía todos los mensajes relacionados con el correo electrónico y de todas las prioridades al archivo */var/log/mail*.

En cambio:

***. emerg ***

Envía todos los mensajes críticos a las consolas de todos los usuarios conectados al sistema.

El siguiente ejemplo registra los mensajes del kernel de distintas maneras:

<i>kern</i>	<i>.*</i>	<i>/var/log/kernel</i>
<i>kern.crit</i>	<i>@</i>	<i>IP_MAQUINA_REMOTA</i>
<i>kern.crit /dev/console</i>		
<i>kern.info;kern.!err /var/log/kernel-info</i>		

La primera regla envía todos los mensajes del kernel a */var/log/kernel*.

Con la segunda los mensajes críticos del núcleo se envían a una máquina remota (que debe configurarse para aceptar los mensajes).

La tercera hace que los mensajes críticos se muestren por las consolas de los usuarios.

Con la última se envían mensajes del kernel que tengan una prioridad igual o mayor que info y menor o igual que err al archivo */var/log/kernel-info*.

Ejemplo de */etc/syslog.conf*:

```
# Se registran todos los mensajes de todos los recursos
# excepto de mail y authpriv con prioridad info o superior
*.info;mail.none;authpriv.none /var/log/messages
# Los mensajes del recurso authpriv se guardan en un archivo
# llamado secure
authpriv.* /var/log/secure
# Los mensajes del recurso mail se guardan en el archivo maillog
mail.* /var/log/maillog
*.emerg
# Los mensajes del arranque del sistema se guardan en el archivo
boot.log
local7.* /var/log/boot.log
```

108.2.3. Registrar datos manualmente (logger).

La mayoría de mensajes de log son generados de forma automática por servicios y demonios del sistema. Con la herramienta logger se puede enviar un mensaje de forma manual. La sintaxis es:

`logger [-isd] [-f file] [-p pri] [-t tag] [-u socket] [message]`

Tabla 1

Opciones usadas con el comando logger

-i

Registra el PID del proceso logger en el fichero de log

-s	Permite que los mensajes se envíen al archivo de log y también a la salida de error estándar
-d	El mensaje se envía usando datagramas y no pasa por una conexión en streaming al socket del sistema de registro
-f	Permite enviar el contenido de un fichero a un archivo de log
-p	Permite indicar la prioridad del mensaje. Hay que indicarla con un selector (por defecto es user.notice)
-t	Permite indicar un etiqueta distinta a la que incluye por defecto logger, que es su propio nombre. Se suele utilizar para indicar para que programa se ha creado la entrada
-u	Se puede utilizar un socket de red para enviar los datos al fichero de log. Normalmente logger llama a las herramientas del sistema para esta tarea
message	El mensaje a guardar. Logger tomará como mensaje a guardar lo que se especifique desde la línea de comandos después de las opciones. Si no se indica el mensaje, logger aceptará como mensaje lo que se escriba en la entrada estándar en las líneas siguientes. Para finalizar la una introducción hay que pulsar Ctrl - D

Por ejemplo:

```
$ logger Apagando el sistema por tareas de mantenimiento
```

O lo que es lo mismo

```
$logger
Apagando el sistema por tareas de mantenimiento
<-- Ctrl-D
```

Produciría una entrada similar a esta en */var/log/messages*

Feb 16 19:45:58 antonio-desktop antonio: Apagando el sistema por tareas de mantenimiento

Podríamos querer tener registrados los mensajes generados por nuestros scripts en un archivo específico. Para ello deberíamos añadir una línea como la siguiente en el archivo de configuración */etc/syslog.conf*

local5.* /var/log/local5

Usaríamos logger para enviar los mensajes al recurso local5 desde nuestros scripts de la siguiente forma:

```
#logger -p local5.info "Script finalizado con normalidad"
```

108.2.4. Rotación de los archivos de registro.

Los archivos de registro pretenden retener información sobre las actividades del sistema durante un período razonable de tiempo, pero los demonios del sistema no proporcionan medios para controlar el tamaño de estos archivos. Si no se controla su tamaño podrían llegar a consumir todo el espacio disponible de la partición en la que residen. Para evitar este problema, los sistemas Linux emplean las herramientas de rotación. Estas herramientas permiten renombrar y, opcionalmente, comprimir los archivos de registro actuales, eliminar los antiguos, y forzar al sistema para comenzar a utilizar los nuevos archivos de registro.

La herramienta de registro de la rotación más común es un paquete llamado logrotate. Este programa se suele llamar de forma regular a través de cron. El programa logrotate consulta a un archivo de configuración llamado /etc/logrotate.conf que incluye varios ajustes por defecto. Este archivo suele hacer referencia a /etc/logrotate.d donde se pueden encontrar configuraciones a medida para manejar archivos específicos de registro que suelen ser controlados por los programas cuya actividad es registrada.

Ejemplo de fichero /etc/logrotate.conf

```
# Rotar los archivos semanalmente
weekly
# Conservar 4 copias de los archivos de registro antiguos
rotate 4
# Crear nuevos archivos de registro tras la rotación
create
# Comprimir los archivos de registro antiguos
compress
# Hacer referencia a los ficheros por programas individualizados
include /etc/logrotate.d
# Definición de opciones variadas
# No rotar el archivo si está vacío
notifempty
# No enviar notificación por correo cuando el archivo rote para ser eliminado
nomail
# Los archivos son rotados en el directorio donde residen
noolddir
# Opciones de rotación para el archivo wtmp, que no está controlado por ningún programa
específico
/var/log/wtmp {
monthly
create 0664 root utmp
```

```
rotate 1
```

```
}
```

Las últimas líneas del listado muestran el formato para la definición de un archivo de registro específico. Estas definiciones empiezan con el nombre del archivo, seguido por una llave de apertura ({). Terminan en una llave de clausura (}). Las líneas contenidas establecen opciones que pueden anular los valores predeterminados. Por ejemplo para el archivo `/var/log/wtmp` se establece una rotación mensual, haciendo caso omiso de la opción por defecto, que es la semanal que aparecía anteriormente. Estas definiciones son comunes en los archivos individuales situados en `/etc/logrotate.d`, que suelen ser propiedad de los programas cuya actividad es registrada.

Algunas de las características que a menudo se establecen en estas definiciones son:

- Nomenclatura de los archivos rotados: Normalmente a los archivos de registro rotados se les añaden números, como por ejemplo `messages.1` para la primera rotación del archivo de registro `messages`. Si se utiliza la opción **dateext**, al archivo de registro rotado se le añadirá en vez de un número un código de fecha, como `messages-20120105` para la rotación realizada el 5 de enero de 2012.
- Opciones de compresión: La opción **compress** se usa para comprimir los archivos de registro y así ahorrar espacio. Esto se hace usando gzip por defecto, pero se puede especificar otro programa con la palabra clave **compresscmd** (Ej: compresscmd bzip2). La palabra clave **compressoptions** permite pasar opciones al comando de compresión (por ejemplo, para mejorar la relación de compresión).
- Creación de nuevos archivos de registro: La opción **create** indica a logrotate la creación de un nuevo archivo de registro para el uso del sistema de registro o programa. Esta opción puede no funcionar bien con algunos programas, por ello la mayoría usarán la opción **copytruncate**, que le indica a logrotate que copie el antiguo archivo de registro con un nuevo nombre y que borre todos los datos del original.
- Opciones de tiempo: Las opciones **daily**, **weekly** y **monthly** indican al sistema que la rotación de los archivos de registro se haga con un intervalo temporal de un día, una semana y un mes respectivamente.
- Opciones de tamaño: A veces se prefiere establecer un umbral de tamaño en lugar de un umbral de tiempo para saber cuándo rotar los archivos de registro. La palabra clave **size** acompañada de un número establece el tamaño máximo (expresado en bytes) para un archivo de registro. La adición de k o M al número usado como argumento expresa el tamaño en kilobytes o megabytes, respectivamente. Por ejemplo, `size 100k` indica que hay que rotar el archivo cuando se llega a 100 KB.
- Opciones de rotación: La opción **rotate x** indica que se mantengan x copias de los archivos de registro antiguos. Por ejemplo, si se establece `rotate 2` para el archivo `/var/log/messages`, logrotate mantendrá `/var/log/messages.1` y `/var/log/messages.2` además del archivo activo `/var/log/messages`. Cuando se rote este archivo, `/var/log/messages.2` se elimina, `/var/log/messages.1` cambia el nombre a `/var/log/messages.2`, `/var/log/messages` se convierte en `/var/log/messages.1`, y se crea un nuevo archivo `/var/log/messages`.
- Las opciones de correo: si utiliza **mail dirección_correo**, logrotate enviará por correo un archivo de registro a la dirección especificada cuando éste rote para desaparecer. Si se usa **nomail** no se envirá dicho correo.
- Scripts: Las palabras clave **prerotate** y **postrotate** comienzan sendas series de líneas que se tratan como scripts a ejecutar justo antes o justo después de la rotación del archivo de

registro, respectivamente.

108.2.5. Consultando las bitácoras del sistema.

Los logs del sistema no sirven de mucho si sólo se van guardando y no se consultan nunca. El objetivo de los ficheros de log es identificar problemas durante la ejecución de programas y son una fuente de información importante para la resolución de problemas. Algunas procedimientos que nos pueden ayudar a examinar ficheros de registro son:

- Recorrer los archivos de registro usando paginadores como **more** o **less**.
- Buscar por palabras clave: Se usa grep para filtrar mensajes de ficheros de log o buscar texto en todos los ficheros de log. Por ejemplo **grep eth0 /var/log/*** localizará todas las líneas de todos los archivos del directorio **/var/log** que contengan eth0. Usaremos **zgrep** en el caso de que los archivos de log estén comprimidos.
- Mostrar el inicio y el final de los ficheros con comandos como **head** o **tail**. Este último puede monitorizar un fichero de registro en uso, mostrando por pantalla las líneas conforme se van añadiendo. Hay que usarlo con la opción **-f**, como en **tail -f /var/log/syslog**.

Las líneas contenidas en los archivos de registro poseen el siguiente formato:

Comienzan con una marca temporal y el nombre del equipo en el que ha tenido lugar la actividad. A continuación aparece un identificador del programa que ha registrado la actividad, incluyendo su número de PID. El resto de la entrada contiene los datos reales registrados. El ejemplo siguiente muestra las entradas de dos tareas periódicas de cron creadas por la herramienta munin.

```
Apr 24 13:25:01 BSFHPCasa CRON[15124]: (munin) CMD (if [ -x /usr/bin/munin-cron ]; then /usr/bin/munin-cron; fi)
Apr 24 13:25:01 BSFHPCasa CRON[15125]: (root) CMD (if [ -x /etc/munin/plugins/apt_all ]; then /etc/munin/plugins/apt_all update 7200 12 >/dev/null; elif [ -x /etc/munin /plugins/apt ]; then /etc/munin/plugins/apt update 7200 12 >/dev/null; fi)
```

Nos encontramos en primer lugar una marca temporal de cuando ha sucedido el evento (Apr 24 13:25:01).

Después el nombre de la maquina (BSFHPCasa) donde ha sucedido el evento (recuerde que es posible recibir mensajes de log de otras máquinas).

Seguidamente la aplicación que ha hecho la entrada y entre corchetes el PID del proceso que ha generado la entrada al registro del sistema y finalmente el contenido del mensaje.

108.3. Fundamentos de Mail Transfer Agent (MTA)

Peso en el examen de certificación: 3 puntos.

Objetivo: Conocer los programas más comunes para proveer a un sistema de la funcionalidad de MTA. Como cliente, realizar envíos y configurar alias. Otros aspectos de la configuración no están cubiertos.

Conceptos y áreas de conocimiento:

- Crear alias de direcciones de correo electrónico.
- Configurar entrega de correo electrónico (forwarding).
- Conocimiento de los programas MTA más comunes (postfix, sendmail, qmail, exim) pero no su configuración.

Términos y utilidades

- `~/.forward`
- comandos de emulación de sendmail.
- newaliases
- mail
- mailq
- postfix
- sendmail
- exim
- qmail

108.3.1. Introducción.

El correo electrónico es uno de los servicios de red más importantes. El envío y la entrega de correo electrónico es un servicio fundamental desde la creación de Internet. Es más, Linux se basa en el correo electrónico incluso en entornos completamente ajenos a las redes; algunos subsistemas, pueden utilizar el correo electrónico para informarle de sus actividades, utilizando el programa cron. Por este motivo la mayoría de las distribuciones Linux llevan instalado software de servidor de correo y están configuradas para sus actividades básicas, por lo que es importante conocer el funcionamiento básico de estos servidores para poder utilizarlos.

108.3.2. Fundamentos.

Antes de entrar en profundidad con este tema vamos a explicar una serie de conceptos básicos para entender el funcionamiento del correo electrónico.

Agente de transferencia de correo (MTA = Mail Transport Agent)

Es un programa que se ejecuta en el servidor de correo con el fin de transferir un conjunto de datos de una computadora a otras.

Cliente de correo electrónico (MUA= Mail User Agent)

Es un programa que permite leer y enviar mensajes de correo electrónico.

Agente de reparto de correo (MDA= Mail Delivery Agent)

Es un programa que acepta el correo entrante desde un MTA y se encarga de distribuirlo a los buzones de los destinatarios si estos se encuentran en la maquina local, si se encuentran en máquinas remotas, los reenviará a un servidor SMTP.

MX (Mail exchanger)

Un registro MX o Mail Exchange Record (registro de intercambio de correo) es un tipo de registro, un recurso DNS que especifica cómo debe ser encaminado un correo electrónico en Internet. Los registros MX apuntan a los servidores a los cuales envían un correo electrónico, y a cuál de ellos debería ser enviado en primer lugar, por prioridad.

1.Cuando un cliente (usuario) manda un mensaje, utiliza un **MUA**, como pueden ser Outlook Express, Kmail, Mutt, Evolution, Thunderbird, etc.

2.El **MUA** se encarga de enviar el mensaje al **MTA**. Este estudia la dirección electrónica para encontrar al usuario y dominio de destino. Después comprueba la información DNS de tipo **MX** para el dominio elegido, de esta manera sabe a qué servidor enviar el correo. Si ningún MTA está disponible, se coloca el mensaje a la fila de espera y se reenviara mas tarde (tardara más o menos dependiendo de la configuración que tenga el MTA).

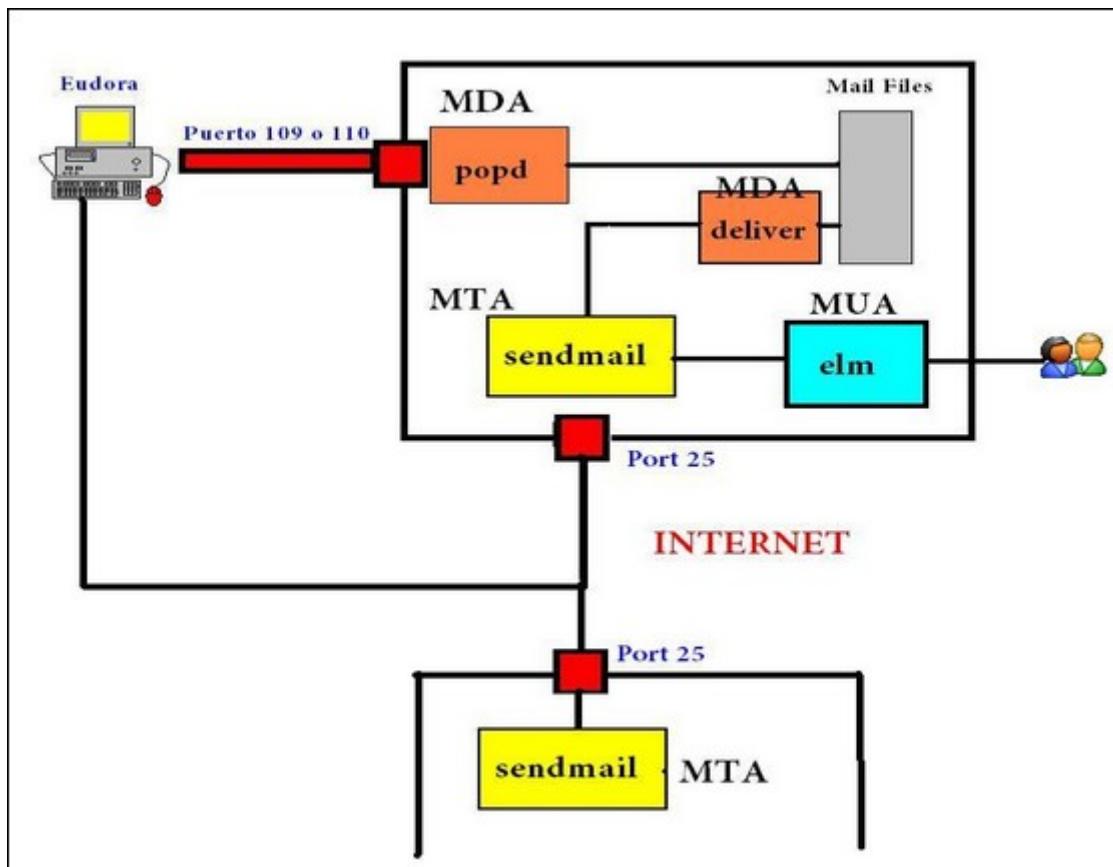
3.El MX puede ser un MTA, que se encargará de enrutador (si se trata de una redirección hacia un subdominio), o un MDA. Este último colocara el mensaje en un fichero temporal, puede filtrarlo, etc.

4.El destinatario recibe el mensaje o lo puede recuperar al leer directamente el fichero temporal o pasa por un protocolo de tipo *POP* o *IMAP*.

5.El protocolo de transporte de mensajes es el **SMTP** (Simple Mail Transfer Protocol) usando el puerto 25.

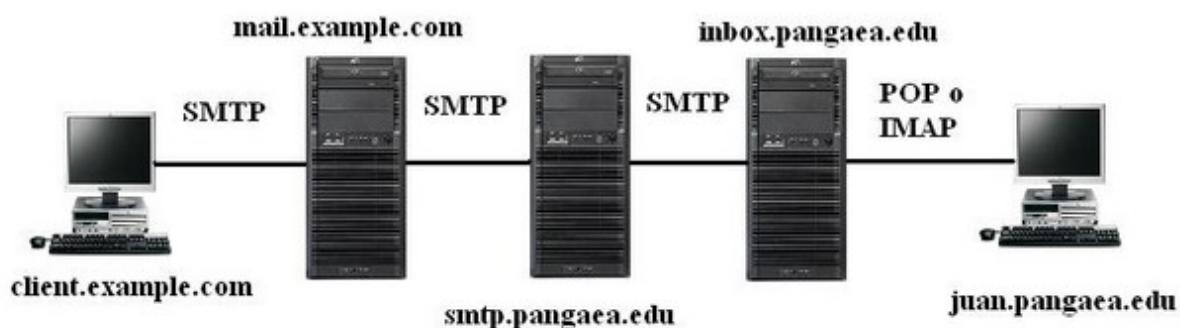
6.Los protocolos de recepción de mensajes son *POP* (**Post Office Protocol**) usando el puerto 110 (POP3), o *IMAP* (**Internet Message Access Protocol**).

En la imagen siguiente podemos ver como es la interacción de los diferentes agentes que intervienen en el proceso de funcionamiento de un servidor de correo.



Existen diferentes protocolos para administrar el correo electrónico. El más común de estos es **SMTP** (*Simple Transfer Protocol*, Protocolo simple de transferencia de correo), este permite enviar los correos electrónicos. Este protocolo se utiliza en la mayoría de los sistemas de reparto de correo. Sin embargo en su etapa final, se utiliza frecuentemente el protocolo **POP** (*Post Office Protocol*, **Protocolo de la oficina de correo**) o **IMAP** (*Internet Message Protocol*, **Protocolo de acceso a mensajes de Internet**). En estos protocolos, es el sistema receptor quien inicia la transferencia. Esto resulta útil cuando el sistema receptor es el equipo de un usuario, que puede no estar encendido en todo momento o ser capaz de recibir conexiones entrantes.

SMTP está concebido para permitir que un mensaje pase por un número arbitrario de ordenadores. Por ejemplo, un usuario final podría confeccionar un mensaje que se enviara al servidor SMTP local (MTA). Este servidor buscará un sistema receptor utilizando DNS y enviará el mensaje a dicho sistema. Este sistema puede utilizar su propia tabla de enrutamiento interna para redireccionar el mensaje a otro sistema local, desde el que se podrá leer el mensaje, bien directamente o a través de un servidor POP o IMAP. La imagen siguiente muestra esta composición.



El número de enlaces de esta cadena es variable y depende de cómo este configurado cada sistema. En el caso más sencillo, los correos locales permanecen en un solo sistema. En teoría, pueden verse implicados en el intercambio de correo un gran número de ordenadores, aunque en la práctica es raro encontrarse con correo que pase por más de media docena de sistemas.

El correo se altera en cada eslabón de la cadena de transmisión. Y lo que es más importante: cada servidor añade un encabezado al correo, que es una línea que proporciona información sobre el mensaje. Los servidores añaden encabezado **Received** (recibido) para documentar la ruta que ha tomado el correo. Esto le permite hacer una traza inversa del correo hasta llegar a su origen. Lamentablemente, los **spammers** y similares han aprendido a falsificar los encabezados del correo, lo que complica enormemente este análisis.

Un servidor SMTP puede hacer tanto de servidor (recibiendo correo de otro sistema) como de cliente (enviando correo a otros sistemas), deberá tratar con ambos lados de la ecuación de la configuración. Algunas veces un sistema no funciona con uno u otro rol, lo que puede simplificar las cosas, pero entonces deberá tener cuidado de no configurar el sistema incorrectamente por accidente. Se deben evitar las configuraciones **open relay** (de transmisión abierta), en las que un servidor de correo transmite el correo de cualquier usuario.

El servidor de correo contiene los mensajes entrantes de cada usuario, normalmente en el fichero `/var/spool/mail` (por ejemplo, `/var/spool/mail/juan` contiene el correo del usuario juan. Sin embargo, algunos servidores de correo nunca almacenan el correo entrante en subdirectorios de los directorios home de los usuarios. Este fichero o directorio de correo entrante se conoce como buzón de correo (*mail spool*) del usuario.

El correo electrónico se puede enviar y recibir. El método tradicional para enviar correos en Linux es hacer que los programas locales contacten con el servidor de correo local para el envío. El servidor de correo local contactará entonces con su servidor de correo saliente, como en la figura anterior. La mayoría de clientes de correo de Linux (conocidos por MUA), así como los programas similares de otras plataformas, ofrecen la opción de contactar directamente con un servidor SMTP remoto al enviar el correo. Este tipo de configuración simplifica la ruta del correo pero puede hacer que la operación sea poco fiable si se cae la conexión de la red local. Si su cliente de correo comunica con un servidor SMTP que se ejecuta localmente, el correo se podrá encolar para que lo distribuya el servidor SMTP aunque la red esté caída temporalmente.

108.3.3. Agente de transferencia de correo (MTA, Mail transfer Agent).

Las funciones básicas de cada MTA son el envío y recepción de mensajes en Internet. Los cuatro programas MTA más comunes disponibles en los sistemas Linux son sendmail, postfix, qmail y exim. Cada uno tiene sus propias características, diferenciándose fundamentalmente en la forma de configurar los ficheros.

Sendmail

(www.sendmail.org) fue uno de los primeros MTAs usados en los sistemas Unix. Surgió del programa "delivermail", con la versión BSD de Unix en 1979. Sendmail ha ido mejorando y evolucionando hacia un programa muy complejo, donde a menudo es bastante complicado configurarlo correctamente. Este hecho combinado con la gran cantidad de vulnerabilidades que ha tenido a lo largo de los años, ha provocado que su popularidad disminuyera. La configuración de sendmail es tan compleja que se inventó un lenguaje de macros llamado **m4** sólo para él. Así no se

edita (o muy poco) el fichero de configuración de sendmail, sino que se edita el fichero fuente de las macros y se vuelve a compilar. Aunque las más importantes distribuciones provienen de un paquete para Sendmail, ninguna de ellas viene en la actualidad con sendmail como MTA predeterminado.

Postfix

Fue diseñado a finales de 1990 como una alternativa más segura que sendmail. Es mucho mas sencillo de configurar que sendmail. Comparte muchas de las opciones de configuración de sendmail, pero no comparte ningún código. Postfix es muy popular en el mundo Linux y es el MTA predeterminado con las distribuciones de Linux más populares. Los objetivos de los desarrolladores de postfix son:

- La compatibilidad con sendmail
- La rapidez
- La sencillez
- La seguridad
- La modularidad

Postfix es muy sencillo de configurar, ya que con un solo comando y sin necesidad de editar un fichero se puede configurar.

Postfix

(www.postfix.org) utiliza varios programas, controlando cada cual su pequeña tarea específica. Este planteamiento mejora la seguridad, al menos en teoría.

Qmail

A mediados de los años 90 fue desarrollado Qmail (www.qmail.org) en respuesta al incremento del número de los incidentes relacionados con la seguridad de los MTAs. Qmail es pequeño, eficiente y seguro, por lo que es una opción popular para los sistemas con pocos recursos. Sin embargo no se ha desarrollado mucho desde 1997, y su falta de actualización para las opciones modernas, como Ipv6 ha limitado su utilidad. Qmail aunque sigue en vigencia, no suele aparecer en las nuevas distribuciones. Es un servidor modular con la seguridad como principal objetivo. NO es el servidor de correo estandar de ninguna distribución porque su licencia es un poco extraña y complica su distribución con Linux; sin embargo muchos administradores prefieren utilizarla en lugar de los servidores de correo estándar de sus distribuciones.

Exim

Exim (www.exim.org) es otro ejemplo de un MTA que se desarrolló como respuesta directa a los problemas de seguridad con sendmail. Fue diseñado para ser un cliente de correo de propósito general para sistemas Unix y es ampliamente utilizado en sistemas con una gran cantidad de requerimientos. Fue creado en 1995 y en la actualidad sigue teniendo un desarrollo activo. Exim es actualmente el MTA por defecto para la distribución Debian GNU/Linux.

El servidor de correo electrónico esta ejecutándose en el sistema de diferentes formas. La más fiable es utilizar ps para buscar los procesos en ejecución, o utilizar las herramientas de gestión de paquetes para ver cuál es el paquete es instalado. En cualquier caso, puede que tenga que comprobar cada uno de los programas a su vez. Por ejemplo, puede ver los resultados como estos.

Podemos averiguar cuál es el servidor de correo que ejecuta una distribución de varias maneras. Las dos más fiables son utilizar ps para buscar los procesos en ejecución o utilizar sus herramientas de administración de paquetes. Por ejemplo:

```
$ ps ax | grep send
```

```
31129 pts/2R+ 0:00 grep send
```

```
$ ps ax | grep post
```

```
7778 ? Ss 0:45 /usr/lib/postfix/master  
31132 pts/2S+ 0:00 grep post
```

En el primer ejemplo buscamos un proceso que contenga la cadena **send** (enviar) no ha funcionado, pero la búsqueda de **post** ha devuelto un proceso llamado */usr/lib/postfix/master*, por que en este sistema se esta ejecutando Postfix.

También podríamos buscar los nombres de los ficheros ejecutables de cada uno de los servidores de correo en */usr/bin* o */usr/sbin*, pero la mayoría de los servidores de correo de Linux incluyen un programa llamado **sendmail**. Esto se hace por compatibilidad con el programa sendmail original.

108.3.4. Configuración de Sendmail.

La configuración global de sendmail es bastante compleja y no es necesario para realizar el examen LPI 102, por lo que nos centraremos en el alias de dirección de correo electrónico, el envío de correo, los archivos del registro de seguimiento y la resolución de problemas básicos.

Sendmail es un conjunto de herramientas monolítico, con un simple manejo del envío y recepción de correo electrónico. Sendmail soporta muchos tipos de protocolos de retransmisión de correo, pero nosotros vamos a utilizar SMTP.

Por defecto, sendmail escuchara una conexión SMTP entrante. Cuando se recibe una conexión, sendmail inicia la conversación SMTP y acepta un email. Comprueba que las direcciones y dominios sean validos, pone un alias, reenvía correo, a agente local para el procesamiento local. Sendmail registra todas las actividades a través del servicio de syslog, que suele ser configurado para almacenar los registros relacionados con el correo en el archivo */var/log/maillog*. Aquí tenéis un ejemplo de la verificación de una instancia de sendmail y el envío de un correo de prueba.

```
# netstat -anlp --tcp | grep sendmail  
tcp        0      0 127.0.0.1:25            0.0.0.0:*          ESCUCHAR  
3138/sendmail: MTA:  
  
# ls -l /var/spool/mail/juan  
-rw-rw---- 1 juan mail 0 2012-01-15 12:40 /var/spool/mail/juan  
# echo "Esto es un mensaje de prueba" | mail juan  
# ls -l /var/spool/mail/juan  
-rw-rw---- 1 juan mail 689 2012-01-15 12:50 /var/spool/mail/juan  
# tail /var/log/maillog  
Jan 15 16:22:42 server sendmail[5387]: 017JMgbM005387:  
from=root, \  
size=32, class="0", nrcpts=1, msgid=<2201201151922.  
017JMgbM005387\  
@server>, relay=root@localhost  
Jan 15 16:22:42 server sendmail[5388]: 017JMgbM005388: \  
from=<root@server>, size=353, class="0", nrcpts=1, \  
msgid=<2201201151922.017JMgbM005387@server>, proto=ESMTP, \  
daemon=MTA, relay=server [127.0.0.1]  
Jan 15 16:22:42 server sendmail[5387]: 017JMgbM005387: to=juan, \  
ctladdr=root (0/0), delay=00:00:00. mailer=local, pri=30607, \  
pri=30032, relay = [127.0.0.1] [127.0.0.1]. dsn=2.0.0, stat=Sent \  
(017JMgbM005388 Message accepted for delivery)  
Jan 15 16:22:42 server sendmail[5389]: 017JMgbM005388: \  
to=<juan@server>, ctladdr=<root@server> (0/0), \  
delay=00:00:00, xdelay=00:00:00, mailer=local, pri=30607, \  
dsn=2.0.0, stat=Sent  
# cat /var/spool/mail/juan
```

```

From root@server Sat Jan 15 16:22:42 2012
Return-Path: <root@server>
Received: from server (server [127.0.0.1])
by server (8.4.2/8.4.2) with ESMTP id 017JMgbM005388
for <juan@server>; Sat Jan 15 16:22:42 2012 -0600
Received: (from root@localhost)
by server (8.4.2/8.4.2/Submit) id 017JMgbM005387
for juan; Sat Jun 15 16:22:42 2012 -0600
Date: Sat Jun 15 16:22:42 2012 -0600
From: root <root@server>
Message-Id: <2201201151922.017JMgbM005387@server>
To: juan@server
Esto es un mensaje de prueba

```

En este ejemplo, estamos verificando que escuchamos TCP en el puerto 25 y estamos usando los comandos estándar de correo de Linux para enviar un correo electrónico a través de sendmail. Sendmail graba por defecto el correo electrónico en /var/spool/mail/\$username , así nosotros vemos que el tamaño de /var/spool/mail/adamh se incrementa de 0 bytes a 689 bytes. Este fichero nos muestra la información del encabezado de correo que esta en formato **mbox**. Por ultimo, hemos visto lo que el registro aparece en el registro log al examinar el archivo /var/log/maillog.

El comando **mail** puede ser utilizado tanto enviar como para leer correo electrónico que se almacena en formato **mbox**. La forma más fácil de enviar el correo es por medio de tuberías con el comando mail, como se muestra en el ejemplo anterior. El comando **mail** tiene muchas otras opciones, y es un comando muy útil, a pesar de no ser un comando específico para sendmail. Está diseñado para trabajar con cualquier MTA compatible con los estándares.

Mail

Sintaxis

mail [-v] [-s asunto] [-c dir-cc] [-b dir-bcc] dir-para

mail [-v] [-f [nombre] | -u usuario]

La primera de estas líneas se utiliza para enviar correo y la segunda para leer correo (a diferencia de la mayoría de lectores de correo, **mail** sólo permite leer la cola de correo local, no se pueden leer los correos almacenados en servidores remotos que si se lean median **POP** o **IMAP**).

Descripción

Un sistema de procesamiento de correo que se pueden utilizar para enviar y leer correo de Internet. Posee la ventaja de que se puede utilizar desde un **script**, por lo que se puede escribir un script para controlar automáticamente algunas tareas del correo y puede que incluso ejecutar dicho script automáticamente.

Está pensado para utilizarse desde la línea de comandos para enviar o recibir mensajes.

Opciones

-v

Genera una salida más detallada. Esto puede servir para la depuración de programas.

-s asunto

Permite especificar una línea para el asunto.

-c dir-cc

Permite enviar una copia del mensaje a varias personas.

-b dir-cc

Igual que la opción anterior, con la diferencia que esta opción produce una copia oculta, lo que significa que la dirección del receptor no aparecerá en la lista de direcciones. Esto se utiliza cuando se desea enviar discretamente una copia de un correo a alguien, aunque puede que algunos filtros anti-spam eliminan este tipo de correos.

Especificificar la dirección del receptor: la dirección de correo del principal receptor finaliza la línea

de comandos de mail de un correo saliente.

-f nombre

Para leer el correo usamos -f seguido del nombre del fichero del buzón de correo.

-u usuario

Permite leer el correo del usuario especificado.

Existen mas opciones que se pueden consultar en la página **man** de **mail**, pero las mas habituales son las opciones comentadas anteriorment.

Ejemplo:

Iniciar correo en modo interactivo para leer su correo.

```
# mail
```

Send an email from the command line:

```
# mail -s "This is the subject" -c "root" juan
```

Hello

.

Cc: root

Este ejemplo muestra algunas opciones del comando **mail**, especialmente la posibilidad de indicar un sujeto y la lista Cc:. En este ejemplo, el cuerpo del mensaje fue dado de forma interactiva, terminando con un "." en una línea. También se podría usar **cat** en un archivo existente y redirigir STDOUT el programa **mail** para que no sea interactivo:

```
# echo "Message body" > /tmp/body.msg
```

```
# cat /tmp/body.msg | mail -s "This is the subject" -c " root"  
juan
```

Este correo fue entregado con éxito porque había una cuenta de usuario llamada **juan** en este sistema. ¿Qué pasa si queremos crear alias para este usuario para que pueda obtener por correo electrónico en la misma bandeja de entrada a través de una serie de direcciones de correo electrónico? Sendmail maneja alias con el archivo **/etc/aliases**:

```
# cat /etc/aliases  
#  
# Los alias en este fichero no se expandirán en la cabecera del  
# mail, pero serán visibles en las redes o desde /bin/mail.  
#  
# >>>>>>> Los programas newaliases deben ejecutarse  
# >> NOTA >> después de que este archivo se actualice, para  
# >>>>>>> que los cambios se muestren a través de sendmail  
#  
# Los alias basicos del sistema deben estar presentes.  
mailer-daemon:postmaster  
postmaster:root  
  
# Redirecciones generales para pseudo-cuentas.  
bin:root  
daemon:root  
adm:root  
lp:root  
# Usuarios que mantienen los alias  
juan:juanh  
juan.haeder:juanh  
haeder:juanh
```

Las líneas en este archivo tienen el formato de **alias: cuenta de usuario**. Después de las modificaciones realizados en este archivo, debe ejecutar el comando **newaliases** como root. El comando newaliases cogerá el archivo **/etc/aliases** y lo convertirá en archivo de BD hash de Berkeley. Este es un método estándar de configuración para sendmail, se realizan cambios en los archivos de configuración basados en texto, y luego se convierten en una DB Berkeley para los archivos más rápido de análisis. Los emails que son enviados a Juan, Juan.haeder o todos los haeder serán entregados al usuario Juan.

Y si desea reenviar todos los mensajes enviados a una cuenta de usuario específica a otra cuenta, ya sea en el mismo sistema o de una dirección de correo electrónico? La forma más sencilla de lograr esto es con el archivo **~/.forward**. Esto es simplemente un archivo de texto que se encuentra en el directorio home del usuario y contiene uno o más direcciones de correo electrónico para reenviar todo el correo. Estos pueden ser direcciones locales (nombres de usuario) o completas direcciones de correo de Internet (user@hostname.com). La ventaja del archivo **x** es que el usuario pueda mantenerse a sí mismo, mientras que el archivo **/ etc / alias** debe ser mantenida por el usuario root. La ventaja del archivo **~/.forward** es que el usuario puede mantenerse a sí mismo, mientras que el archivo **/ etc / alias** debe ser mantenida por el usuario root.

Ejemplo: Enviar un mensaje de correo a dos destinatarios para informarles de una cita.

```
mail -s "Recordar cita" -c juan@gmail.com fran@gmail.com  
Recordar la cita hoy a las 10  
Cc: juan@gmail.com
```

Después de escribir el comando **mail**, el programa esperará datos por la entrada estándar, pues no existe **prompt**. Para indicar el final del mensaje, pulse **Control-D**. Después de pulsar **Control-D**, el programa mostrará la línea **Cc:** para verificar esta opción. En este punto todavía podría cambiar la dirección, pero si no desea hacerlo, puede pulsar **Intro** para que el mensaje sea enviado.

Si utilizamos **mail** en un **script**, puede pasarselo a éste el contenido del fichero a enviar aplicando una redirección de la entrada:

```
mail -s "Alerta automática" < /tmp/alert.txt juan@gmail.com
```

Si la linea anterior se incluye en un script, enviará el contenido de **/tmp/alert.txt** a juan@gmail.com con el asunto especificado.

108.3.5. Colas de correo.

Un servidor de correo gestiona la cola de los mensajes de correo que debe distribuir. Esta cola es similar en algunos aspectos a la cola de tareas de impresión que controla el sistema de impresión de Linux, con la diferencia de que, en lugar de enviar tareas a una impresora, el servidor de correo envía mensajes de correo a otro ordenador o los almacena en los buzones de los usuarios locales.

Esta tarea parece sencilla, pero, a veces, resulta sorprendentemente compleja, ya que se le puede pedir al servidor que distribuya muchos mensajes en un periodo de tiempo muy corto, con lo que es posible que tenga que retrasar el envío de algunos mensajes mientras trabaja con otros. Es más, puede que aparezcan problemas que deriven en una incapacidad temporal o permanente para enviar mensajes. Cuando el problema parezca temporal, como un fallo del enrutamiento de la red, el servidor de correo deberá guardar el mensaje e intentar enviarlo de nuevo más adelante. Por tanto, la cola de correo de un ordenador Linux puede contener mensajes no enviados. Saber como identificar estos mensajes y gestionar la cola le será de utilidad para mantener la fluidez del funcionamiento del subsistema de correo de su ordenador.

Descripción

Sendmail es un agente de correo inteligente, y trata de entregar el correo aunque se produzcan fallos. Por ejemplo, si un usuario intenta enviar un correo electrónico a **user@hostname.com**, y el servidor de correo hostname.com se apaga, sendmail será incapaz de realizar una conexión al puerto TCP 25 en el servidor de correo para hostname.com y por consiguiente, no será capaz de entregar el correo. En lugar de darse por vencido, sendmail colocará el mensaje en una cola y intentara reenviarlo. La acción por defecto de sendmail es intentar el reenvío de correo en cola cada 4 horas durante 5 días antes de abandonar y el envío de una notificación "Delivery Failure (Falta de entrega)" al remitente original. Las colas de correo se almacena en el directorio /var/spool/mqueue y es administrado por el programa **mailq**.

mailq

Sintaxis

mailq [opciones]

Descripción

Muestra los correos que se encuentran en la cola y saber si ese correo ha sido enviado.

```
$ mailq
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-----
5B42F963F* 440    Sat Aug 23 13:58:19 juan@gmail.com
                  fran@gmail.com
-- 0 Kbytes in 1 Request.□
```

El ejemplo anterior, muestra el contenido de la cola de correo en todos los sistemas de correo.

El comando **mailq** equivale a escribir **sendmail -bp**.

Ejemplo: Intenta enviar un correo a los host que actualmente no estan aceptando correos electrónicos:

```
# echo "Failure Test" | mail user@unknown.com
View the mail queue
# mailq
01591AmX0056157182  Mon Jan 14 17:01 MAILER-DAEMON
8BITMIME(Deferred: Connection refused by unknown.com.)
<user@uknown.com>
```

Este correo se mantendrá en la cola durante un tiempo predeterminado de 5 días, con reintentos cada hora. Puede forzar a sendmail a un intento de volver a enviar de cada elemento de la cola de correo con el comando **sendmail -q -v**

.

Si la conexión de red se cae temporalmente o si un servidor de correo superior deja de estar operativo durante un tiempo, puede que los mensajes se amontonen en la cola. El servidor SMTP, intentará volver a distribuirlos más adelante; pero si su conexión vuelve a funcionar y desea vaciar la cola de inmediato, puede hacerlo. En la mayoría de los servidores SMTP bastará con escribir ***sendmail -q***; otros poseen comando equivalentes, como ***postqueue*** en Postfix o ***runq*** en Exim.

108.3.6. Configuración de Postfix.

Postfix fue creado para reemplazar a sendmail, y por lo tanto, mantiene un interfaz compatible con sendmail. En la mayoría de los casos, postfix puede reemplazar a sendmail, y los scripts que había llamando a sendmail directamente con varias opciones de línea de comandos continuarán trabajando. Postfix logra esto mediante la inclusión de la distribución de un programa llamado /usr/sbin/sendmail, que suele actuar como un puente entre las llamadas a sendmail y la utilidad de postfix. Debido a esto, muchos de los comandos que se utilizan en el sendmail trabajarán con postfix:

```
# which sendmail
/usr/sbin/sendmail
# for file in /usr/sbin/sendmail /usr/bin/mailq /usr/bin/newaliases; {echo -n
"$file: " && rpm -q --whatprovides ${file}; }
/usr/sbin/sendmail: postfix-2.3.2-32
/usr/bin/mailq: postfix-2.3.2-32
/usr/bin/newaliases: postfix-2.3.2-32
```

El sistema de postfix se compone de una serie de aplicaciones diferentes, en contraposición a la naturaleza monolítica de sendmail. El programa principal es /usr/lib/postfix/master, que es el demonio que escucha en el puerto TCP 25 para las conexiones SMTP y acepta correo. Otras aplicaciones se muestran en la tabla siguiente. Estas aplicaciones están en /usr/lib/postfix/ a menos que se indique lo contrario.

Programas de Postfix	
Nombre	Descripción
<i>anvil</i>	Mantiene estadísticas sobre las conexiones del cliente o el índice de peticiones del cliente. Esta información puede ser utilizada para controlar el número de correos enviados por unidad de tiempo. De esta forma presentamos una defensa contra clientes que realizan muchas conexiones o peticiones simultáneas al servidor en la unidad de tiempo especificada. Dirigido por el servidor maestro de Postfix.
<i>bounce</i>	Mantiene un fichero log donde se guarda la información de estado de cada mensaje de entregado. Dirigido por el servidor maestro de Postfix.
<i>cleanup</i>	Procesa el correo entrante, lo inserta en la cola de correo entrante, e informa al gestor de colas de su llegada.
<i>discard</i>	Procesa las solicitudes de entrega del gestor de colas que deben ser desechados.

error	Procesa las solicitudes de entrega del gestor de colas que deben ser registrados como errores.
flush	Mantiene un registro de los correos en espera por destinos.
lsmtp	Implementa los protocolos SMTP y LMTP de entrega de correo para entregar el correo.
local	Procesa las solicitudes de entrega del gestor de colas que deben ser suministrados localmente.
pickup	Mueve el correo desde el directorio maildrop para el proceso de limpieza.
pipe	Procesa peticiones procedentes del administrador de colas para entregar mensajes a comandos externos.
proxymap	Proporciona servicios de tablas de búsqueda de sólo lectura para los procesos de Postfix.
qmgr	Un demonio que espera la llegada de correo entrante y lo prepara para su entrega.
mqpd	Un demonio que recibe un mensaje por conexión, lo procesa a través del demonio cleanup y lo coloca en la cola de entrada.
scache	Mantiene una caché compartida de múltiples sesiones que pueden ser utilizados por los diferentes programas de postfix.
showq	Un demonio que informa sobre el estado de las colas de correo de Postfix.
smtp	Busca una lista de direcciones de intercambio de correo para el anfitrión de destino, ordena la lista por preferencias y se conecta a cada dirección listada hasta encontrar un servidor que responda.
spawn	Escucha en un puerto especificado en el fichero master.cf de Postfix y lanza un comando externo cada vez que se establece una conexión.
tlsmgr	Maneja el almacenamiento en caché de conexiones TLS.
trivial-rewrite	Un demonio que reescribe direcciones a un formato estándar.
verify	Mantiene un informe de qué direcciones de destinatarios se sabe que son entregables o no entregables.
virtual	Entrega correo a direcciones de correo de usuarios virtuales.

/usr/bin/newaliases	Compatible con el comando de sendmail newaliases . Convierte el fichero de texto <i>/etc/aliases</i> en un archivo binario que es analizado por postfix.
----------------------------	---

La configuración de **Postfix** se encuentra en */etc/postfix/main.cf*. Se modifican sus valores bien manualmente, o con la ayuda del comando **postconf**.

Postfix inicia primero un servicio maestro, **master**, encargado de los procesos secundarios **smtpd**, **pickup** y **nqmgr**.

Aplique una configuración básica con el comando **postconf**.

Dominio de origen de los mensajes

```
#postconf -e "myorigin = midominio.org"
```

De qué dominio recibir el correo

```
#postconf -e "mydestination = midominio.org"
```

De qué clientes transmitir el correo

```
#postconf -e "mynetworks = 192.168.1.0/24, 127.0.0.1"
```

En que interfaces escuchar

```
#postconf -e "inet_interface = all"
```

Inicie el servicio

```
#service postfix Stara
```

ó

```
#/etc/init.d/postfix start
```

--

Alias de usuarios

Es posible ubicar alias para los usuarios locales en el fichero */etc/aliases*. Por ejemplo, si los mensajes de webmaster, admin. Y root se deben redireccionar hacia Juan:

manuel: webmaster, admin, root

Probándolo

El registro se ubica en */var/log/maillog*. Pruebe el servidor de la manera siguiente:

```
mail -s `echo $USER` root@server1 < /etc/passwd
```

108.3.7. Configuración de Qmail.

Qmail es igual a Postfix en que fue diseñado para reemplazar a sendmail y es una colección de programas más pequeños en lugar de uno grande. El objetivo del diseño de Qmail es la seguridad, por lo que a menudo los programas más pequeños se ejecutarán con un menor número de usuarios privilegiados. En la tabla siguiente se muestran una lista de algunos de los programas más comunes de Qmail:

Programas Qmail	
Nombre	Descripción
tcpserver	Escucha las conexiones entrantes TCP y las maneja con el programa correspondiente. Similar a inetd o xinetd.

<i>qmail-smtpd</i>	Maneja el correo entrante.
<i>qmail-inject</i>	Inyecta el correo saliente en la cola de correo.
<i>qmail-send</i>	Envía mensajes de correo actualmente en la cola de correo.
<i>qmail-queue</i>	Formatea el correo correctamente y lo coloca en una cola para su entrega.
<i>qmail-lspawn</i>	Invoca qmail localmente para realizar la entrega.
<i>qmail-rspawn</i>	Invoca qmail remotamente para realizar la entrega a distancia.
<i>qmail-local</i>	Reparte el correo localmente.
<i>qmail-remote</i>	Reparte el correo remotamente.
<i>qmail-qmqpd</i>	Recibe el correo a través del protocolo Quick Mail Protocol y llama a <i>qmail-queue</i> para colocar el correo en la cola de salida.
<i>qmail-qstat</i>	Resumir el contenido actual de la cola de correo.
<i>qmail-qread</i>	Lista los mensajes y los destinatarios de los mensajes de correo en la cola de salida.
<i>qmail-tcpt0</i>	Listas los hosts que han agotado el tiempo para realizar la entrega de correo.
<i>qmail-tcpok</i>	Borrar de la lista los hosts que han expirado su tiempo.
<i>qmail-showctl</i>	Analiza los archivos de la configuración actuales de qmail y se explica la configuración.
<i>qmail-start</i>	Es un programa contenedor inicia <i>qmail-send</i> , <i>qmail-lspawn</i> , <i>qmail-repawn</i> y <i>qmail-clean</i> con los identificadores de usuario apropiados para que la entrega de correo pueda realizarse.

Qmail también trata de manera similar la compatibilidad con sendmail a Postfix. Qmail viene con el programa */var/qmail/bin/sendmail*, que está diseñado para utilizar las mismas opciones de línea de comandos que sendmail y los transforma a qmail. El programa **dot-forward** se utiliza para leer de un usuario. El archivo **forward** y el programa **fastforward** leerán el archivo de sendmail */etc/aliases*.

108.3.8. Configuración de Exim.

Exim, como Postfix y Qmail, fue diseñado para reemplazar a sendmail. Por lo tanto, tiene

programas de ayuda y apoyo a las opciones de línea de comandos para permitir una transición suave de sendmail. Exim es monolítico, como sendmail, y se encarga de la aceptación de correo electrónico sobre TCP en el puerto 25 y la entrega de correo. Hay una serie de programas de ayuda que vienen con exim. Algunos de los más comunes se encuentran en la tabla siguiente:

Programas de Exim	
Nombre del programa	Descripción
/usr/bin/mailq.exim	Sustituto para el comando de sendmail mailq.
/usr/bin/newaliases.exim	Sustituto para el comando de sendmail newaliases.
/usr/bin/runq.exim	Alias para /usr/sbin/exim. Ejecutar este comando es lo mismo que ejecutar exim -q. Esto obliga a ejecutar a las colas de correo, tratando de que se reenvíe a todos los elementos.
/usr/lib/sendmail.exim	Maneja las opciones de línea de comandos sendmail y los pasa a exim.
/usr/sbin/exim	Este proceso escucha sobre TCP en el puerto 25 para las conexiones SMTP entrantes y maneja el correo entrante para el agente de entrega local.
/usr/sbin/eximstats	Genera estadísticas de los ficheros de exim mainlog y syslog .
/usr/sbin/exiwhat	Describe lo que el proceso de exim que se están ejecutando actualmente.
/usr/sbin/exinext	Muestra información específica sobre los esfuerzos para reintentar una dirección de correo electrónico.
/usr/sbin/exipick	Muestra mensajes de la cola de correo de exim con diferentes criterios.
/usr/sbin/exigrep	Busca cadenas en la cola del correo.
/usr/sbin/exiqsumm	Resume el contenido actual de la cola de correo.

108.3.9. Redirigir el correo.

Los alias de correo permiten utilizar una dirección en lugar de otra. Por ejemplo: todos los servidores de correo deben mantener una cuenta llamada **postmaster** cuyo correo debería leer la persona responsable de mantener el sistema. Una manera de hacerlo es configurar un alias que enlazara el nombre **postmaster** con el nombre de una cuenta real. Eso se puede hacer editando el fichero **aliases**, que suele encontrarse en */etc* o, a veces, en */etc/mail*.

El formato del fichero **aliases** es bastante intuitivo. Las líneas de comentario comienzan con almohadillas (#) y el resto adoptan la siguiente forma:

nombre: dir1[,dir2[,...]]

El **nombre** que inicia la línea es un nombre local, como **postmaster**. Cada dirección (dir1, dir2, etc) puede ser un nombre de cuenta local al que reenviar los mensajes, el nombre de un fichero local en el que guardarlos (denotado por una barra inicial), un comando por el que se canalizarán los mensajes (denotado por una barra vertical), el nombre de un fichero cuyo contenido se tratará como una serie de direcciones (denotado por una cadena: **include:** inicial) o una dirección de correo completa (como juan@gmail.com).

Una configuración por defecto típica incluye unos cuantos alias útiles para cuentas como **postmaster**. La mayoría de estas configuraciones asocian a **root** con estos alias. Sin embargo, no se recomienda leer el correo como **root**, pues ello incrementaría las probabilidades de crear una fisura en la seguridad u otros problemas debidos a un error de escritura o un **bug** del lector de correo. Por tanto, es aconsejable definir una línea de alias como la siguiente:

root: sunombredeusuario

Esto redirecciona todo el correo de **root**, incluyendo el correo dirigido a **root** a través de otro alias, a **sunombredeusuario**, que puede adoptar cualquiera de las formas que acabamos de describir (lo más probable es que sea un nombre de usuario local o una dirección de correo remoto válida). Algunos servidores de correo, entre los que se incluyen sendmail, Postfix y qmail, requieren que se compile **/etc/aliases** en un fichero binario, que se pueda procesar más rápidamente. Para ello, utilice el comando **newaliases**:

newaliases

También se puede redirigir el correo es hacerlo a nivel de usuario. En particular, puede editar el fichero **~/.forward** del directorio home de un usuario para hacer que el correo de éste se envie a otra dirección. El fichero **~/.forward** debería contener específicamente la nueva dirección: bien un nombre de usuario del ordenador actual, bien una dirección de correo completa de otro ordenador. Este sistema posee la ventaja de que lo pueden utilizar los usuarios particulares para, por ejemplo, unificar el correo de varios sistemas en una cuenta sin molestar a los administradores de sistemas. La desventaja es que se puede utilizar para definir alias de cuentas no existentes o para cuentas que carecen de directorios home. El fichero **~/.forward** también lo puede cambiar o borrar el propietario de la cuenta, algo no deseable si quiere implementar una regla de redirección que el usuario no pueda invalidar.

108.3.10. Proteger el servidor de correo.

Un servidor de correo es un riesgo potencial para la seguridad. Por regla general, este riesgo adopta dos formas:

1. **Errores (bugs):** los **bugs** del servidor de correo pueden poner en peligro su sistema. En teoría un bug podría permitir a alguien obtener acceso a su sistema enviando un correo o conectándose al puerto SMTP 25 mediante un cliente Telnet y escribiendo comando SMTP para aprovecharse del error. Por este motivo, muchas distribuciones Linux actuales reducen el acceso al servidor de correo a sólo el ordenador local.
2. **Configuración defectuosa:** una configuración de un servidor de correo puede provocar problemas. Los servidores de correo no están diseñados para proporcionar acceso, por lo que es poco probable que se intente obtener un acceso completo. En vez de ello, el gran riesgo estriba en una configuración que haga de su sistema una amenaza para Internet. La habitual de estas configuraciones defectuosas es un **open relay**, que consiste en un ordenador que transmite el correo de un ordenador a otro. En el pasado, los **spammers** se aprovechaban con frecuencia de este sistema para poder ocultar sus identidades, pero los **spammers** actuales prefieren otras técnicas. Aun así, algunos aún abusan de los **open relay**.

Para protegerse de los **bugs**, debería asegurarse de que su servidor de correo está actualizado con la última versión. Las principales distribuciones de Linux configuran sus servidores de correo evitando los **open relays**; no obstante, una configuración defectuosa puede dejar abierto su servidor de correo. Hay varios sitios Web que permiten comprobar este tipo de configuraciones. Visite www.abuse.net/relay.html o www.saphelp.org/shopenrelay/ para asegurarse de que su sistema no es un **open relay**. Estos sitios ejecutan una serie de pruebas que intentan transmitir correos a través de su servidor. Si su servidor está adecuadamente configurado, la página le informará de que no ha podido conectarse o enviar el correo. Pero si el sitio de prueba ha podido transmitir el correo, deberá saber algo más para poder configurar adecuadamente su servidor. Lamentablemente, los pasos necesarios para proteger un **open relay** varían según el servidor de correo y requieren una configuración relativamente avanzada. Puede descubrir cómo cerrar configuraciones **open relay** en la documentación de su servidor de correo.

108.4. Configuración de impresoras e impresión.

Peso en el examen de certificación: 2 puntos.

Objetivo: Mantenimiento de las colas y los trabajos de impresión de los usuarios utilizando CUPS y el interface compatible con LPD.

Conceptos y áreas de conocimiento:

- Configuración básica de CUPS (para impresoras locales y remotas).
- Mantenimiento de las colas de impresión de los usuarios.
- Solucionar problemas generales de impresión.
- Insertar y eliminar trabajos en las colas de impresión.

Términos y utilidades:

- CUPS: ficheros de configuración, herramientas y utilidades.
- /etc/cups
- Interface tradicional lpd (lpr, lprm, lpq).

108.4.1. Una visión general de la impresión.

La impresión en Linux es un trabajo de cooperación entre varias herramientas. Las aplicaciones envían tareas de impresión como documentos PostScript. La mayoría de los sistemas Linux no están conectados directamente a impresoras PostScript reales, es por ello, por lo que se utiliza un programa llamado Ghostscript que convierte la tarea de impresión a una forma que la impresora del sistema pueda controlar de verdad. La cola de impresión gestionada por CUPS (Common Unix Printing System, Sistema de impresión común de Unix, que es el sistema de impresión más utilizado), envía después la tarea a la impresora. Los administradores pueden examinar el contenido de la cola de impresión en varias etapas y modificar ésta. Es necesario conocer las herramientas utilizadas para crear y gestionar las colas de impresión ya que nos ayudarán a administrar la impresión en Linux.

Todas las implementaciones de impresión disponibles para los sistemas Linux tienen una estructura básica en común. En cada sistema de impresión, un demonio central (o servicio) recibe trabajos de impresión, ya sea a través de un comando de usuario (como **lpr**) o la red. El trabajo de impresión se procesa a través de filtros de entrada si es necesario, y se envía a una impresora local o a otro demonio de impresión.

La impresión de documentos es un proceso lento y propenso a error. Las impresoras aceptan datos en pequeñas cantidades y son propensas a quedarse sin papel, atascos y quedarse fuera de línea por otras razones. Las impresoras también tienen que aceptar las peticiones de múltiples usuarios. Como resultado de su diseño, el usuario final no tiene porque conocer las funciones de impresión en la mayoría de los sistemas informáticos. El sistema de impresión utiliza una cola de impresión, que contiene las solicitudes de impresión hasta que la impresora está preparada. Un mismo ordenador puede admitir varias colas de impresión distintas, que suelen corresponder a impresoras físicas diferentes, aunque se pueden configurar varias colas para que impriman de maneras distintas en la misma impresora. Por ejemplo: podríamos utilizar una cola para imprimir por una sola cara y otra para imprimir a doble cara en una impresora que admite esta funcionalidad.

Los usuarios envían las tareas de impresión mediante un programa llamado *lpr*. Los usuarios pueden llamar a este programa directamente o dejar que otro realice esta llamada. En cualquier caso, *lpr* envía la tarea de impresión a una cola especificada. Esta cola responde a un directorio del disco duro, que suele estar en el directorio */var/spool/cups*. El demonio CUPS se ejecuta en segundo plano, esperando a que se le envíen las tareas de impresión. El sistema de impresión acepta tareas de impresión de *lpr* o de ordenadores remotos, colas de impresión de monitores y actúa como "agente de tráfico", dirigiendo las tareas de impresión de una manera ordenada de las colas de impresión a las impresoras.

El sistema de impresión también gestiona el orden en que los trabajos de impresión se procesan. La alimentación de los trabajos de impresión se llama *spooling*, y el programa que gestiona las colas de impresión se llama *spooler*. También puede ser llamado planificador.

108.4.2. POSTSCRIPT Y GHOSTSCRIPT.

En sistemas operativos como Windows, Mac OS, OS/2 y otros, cuando configuramos una impresora, se utilizan los drivers. El driver de impresión se encuentra entre la aplicación y la cola de impresión. En Linux, el driver de impresión es parte de ***Ghostscript*** (www.cs.wisc.edu/~ghost), que forma parte de la cola de impresión. ***Ghostscript*** hace de medio para traducir PostScript, un lenguaje común de las impresoras, a formatos inteligibles para muchas impresoras diferentes.

108.4.2.1 PostScript: El lenguaje de impresora de Linux

Con la aparición de las primeras impresoras láser en la década de los ochenta, aparecieron también el lenguaje de impresión PostScript, que estas incorporaban. Las impresoras PostScript se hicieron muy populares como accesorios para los sistemas Unix del momento. Las colas de impresión de Unix se diseñaron sin tener en cuenta los drivers de impresión de tipo Windows, por lo que los programas de Unix que sacaron partido de las características de impresión de las impresoras láser, por lo general, se desarrollaron para generar directamente una salida PostScript. Como consecuencia, PostScript llegó a ser el estándar de facto para la impresión en Unix, algo que heredó Linux.

Cuando se desarrollaban programas en sistemas Windows para que interactuaran con el driver de impresión de Windows, en Linux programas similares generaban PostScript y enviaban el resultado a la cola de impresión de Linux.

Algunos programas no siguen este estándar. Hay muchos programas que pueden generar salida como texto sin formato. Esta salida raramente supone un gran problema para las impresoras modernas, aunque ésta no sirve para algunos de los modelos que sólo admiten PostScript. Hay algunos otros programas que pueden producir una salida PostScript o **PCL (Printer Control Language)**, Lenguaje de control de impresoras) para las impresoras láser Hewlett-Packard o sus muchas imitadoras. Muy pocos programas pueden generar una salida que otros tipos de impresora puedan aceptar directamente.

El problema con PostScript como estándar es que es poco habitual en las impresoras de precio bajo o medio que se suelen asociar a Linux. Por ello, para imprimir en tales impresoras utilizando los programas tradicionales de Unix que generan salida PostScript, necesitará un traductor y un modo que adapte dicho traductor a la cola de impresión. Para se utiliza Ghostscript.

108.4.2.2. Ghostscript: El traductor de PostScript

Cuando se emplea una impresora PostScript tradicional, el ordenador envía un fichero PostScript directamente a esta. PostScript es un lenguaje de programación, a pesar de que esta orientado a

generar como salida un pagina impresa. Ghostscript es un intérprete de PostScript que se ejecuta en un ordenador. Toma la entrada de PostScript, la analiza y genera una salida en cualquiera de las varias decenas de formatos de mapa de bits diferentes, incluyendo formatos validos para muchas impresoras que no son PostScript. Esto hace de Ghostscript un medio para convertir muchas impresoras económicas en impresoras PostScript de bajo coste compatible con Linux. Ghostscript esta disponible como software de código abierto (GNU Ghostscript), con una variante mas avanzada (Aladdin Free Public Licence, o AFPL, Ghostscript), disponible gratuitamente. AFPL Ghostscript no se distribuye gratuitamente en ningún paquete comercial. Como todas las distribuciones de Linux que están disponibles en CD-ROM tienen un precio, vienen con el antiguo GNU Ghostscript, que es válido para la mayoría de los usuarios.

Una de las desventajas de Ghostscript es que genera largos ficheros de salida. Un fichero PostScript genera una página llena de texto que puede tener solo unos cuantos kilobytes de tamaño. Si esta página se imprime en una impresora de 600 puntos por pulgada (dpi) utilizando Ghostscript, el fichero de salida resultante tendría un tamaño de 4 MB, suponiendo que se tratase de blanco y negro, si la página incluyese color, el tamaño podría ser mucho mayor. Esto no es tan importante porque estos ficheros grandes sólo se almacenarán en su disco durante un corto espacio de tiempo. No obstante, aún tienen que llegar del ordenador a la impresora, un proceso que puede ser lento. Además, algunas impresoras (en particular las láser) pueden necesitar una ampliación de memoria para operar de manera fiable bajo Linux.

108.4.2.3. Integrar Ghostscript en la cola

Imprimir en Linux en una impresora que no sea PostScript exige integrar Ghostscript en la cola de impresión. Esto se suele hacer utilizando un filtro inteligente, que es un programa al que se llama como parte del proceso de impresión. El filtro inteligente examina el fichero que se va a imprimir, determina su tipo y lo pasa por uno o varios programas adicionales antes de que el software de impresión lo envíe a la impresora. El filtro inteligente se puede configurar para que llame a Ghostscript con los parámetros que sean pertinentes para generar la salida para la impresora de la cola.

CUPS viene con su propio conjunto de filtros inteligentes, a los que llama automáticamente cuando se le indica al sistema el modelo de la impresora que se utiliza. CUPS utiliza una herramienta de configuración Web. Estas herramientas GUI de configuración de la impresión específicas de la distribución pueden configurar una impresora CUPS de una manera bastante intuitiva.

El resultado final de una configuración de una impresora Linux es la posibilidad de tratar cualquier impresora soportada como si fuera una impresora PostScript. Las aplicaciones que pueden producir salida PostScript pueden imprimir directamente en esta cola. El filtro inteligente detecta que la salida es PostScript y la pasa por Ghostscript. También puede detectar otros tipos de fichero, como los de texto plano y varios ficheros gráficos, y puede enviarlos a los programas pertinentes en lugar de, o además de, Ghostscript con el fin de generar una impresión razonable.

Las impresoras que pueden procesar PostScript, probablemente esté implicado en un filtro inteligente, pero no pasara el PostScript por Ghostscript. En este caso, el filtro inteligente le pasara el PostScript directamente a la impresora, aunque enviará los demás tipos de fichero al procesamiento que sea preciso para convertirlos a PostScript.

108.4.3. Ejecutar un sistema de impresión

Los sistemas de impresión de Linux se ejecutan como demonios y deben iniciarse antes de ser utilizados. Esta tarea se suele gestionar automáticamente a través de los scripts de inicio de /etc/rc.d, /etc/init.d0/etc/rc?.d (donde ? es el número del modo de ejecución). Para saber que sistema

de impresión se está ejecutando tendríamos que buscar nombres de *scripts* de inicio que contengan la cadena **cups** (*0 lpd 0 lprng* en los sistemas antiguos). Pero si no estamos seguros de si un sistema de impresión está activo actualmente, podemos utilizar el comando **ps** para buscar los procesos por sus nombres, como en el siguiente ejemplo:

```
$ ps ax | grep cups
```

```
1896 ? Ss 0:01 cupsd
```

En este ejemplo se muestra que **cupsd** está en ejecución, el demonio de CUPS, por lo que el sistema está utilizando CUPS para la impresión. Si no encontramos ningún sistema de impresión en ejecución, tendríamos que consultar la documentación de la distribución que estemos utilizando, para comprobar que está instalado el paquete apropiado. Todas las distribuciones principales incluyen *scripts* de inicio que deberán iniciar el demonio de impresión pertinente cuando arranca el ordenador.

108.4.4. Interfaces de BSD y System V.

Históricamente ha habido dos implementaciones de sistemas de impresión en los sistemas Unix, uno inventado para BSD y otro para System V (SysV) Unix. Aunque las implementaciones son similares, tienen comandos completamente diferentes. BSD incluye los siguientes comandos de impresión: **lpr**, **lpd**, **lpq**, **lprm** and **lpc**. Los comandos de impresión que incluye System V son: **lp**, **enable**, **disable**, **cancel**, **lpstat**, **lpshut**, **accept**, **reject** y **lpadmin**. En los sistemas basados en System V, el comando **lpadmin** gestiona las colas de impresión. No existe un equivalente para los sistemas basados en BSD, que no sea editar */etc/printcap*. Aparte de **lpadmin**, hay una relación uno a uno entre los comando de los sistemas de impresión de BSD y System V. Sin embargo, los detalles internos, de como los archivos son utilizados, varían considerablemente.

Las distribuciones más antiguas de Linux suelen utilizar un puerto de la BSD de código **lpd** (y los comandos relacionados). Debido a diversos problemas de seguridad con el código de BSD (en su mayoría por el uso excesivo de comandos a través de la cuenta root usando ejecutables SUID), las distribuciones actuales han disminuido en gran medida el código de BSD en favor de CUPS.

108.4.5. LPRng.

LPRng es una completa revisión de las utilidades de BSD. Diseñada para ser portátil y segura. A diferencia de las utilidades de BSD, los programas clientes no necesitan ejecutarse como SUID. El servidor (también llamado **lpd**) es una implementación completa de la RFC 1179 protocolo Line Printer Daemon (o LPD).

Aunque **lpr** está completamente reescrito, la configuración en su mayor parte es la misma que las utilidades BSD. Todavía utiliza el archivo */etc/printcap* y también tiene dos archivos adicionales: */etc/lpd.conf*, que controla los detalles de LPRng y */etc/lpd.perms* que configura los controles de **lpd**.

Algunas veces es necesario integrar el servidor de impresión en una infraestructura heterogénea, para servir a los sistemas que utilizan el protocolo LPD legado. El paquete integrado cups-lpd es el CUPS Line Printer Daemon (LPD), mini servidores suportan a estos sistemas de cliente heredado. **cups-lpd** no actúa como un demonio de red independiente, sino que funciona en Internet con **inetd** o con el super servidor **xinetd**.

El servidor LPD escucha en los puertos especificados por defecto en el fichero */etc/service*:

También
podría aparecer lo siguiente:

```
Printer      515/tcp    spooler    # line printer spooler
Printer      515/udp    spooler    # line printer spooler
```

LPRng está disponible en <http://www.lprng.com>.

108.4.6. CUPS

CUPS (Common Unix Printing System) es un sistema de impresión, que fue inicialmente diseñado para soportar el protocolo de impresión de Internet (IPP), pero ha evolucionado hasta reemplazar los servicios de impresión de los sistemas BSD y System V.

A pesar de que mantiene la compatibilidad con sistemas de impresión más antiguos, los detalles internos de CUPS son significativamente diferentes.

Es un sistema de impresión orientado a red, con las siguientes características:

1. Basado en el protocolo **IPP** (*Internet Printing Protocol*).
2. Fácil de utilizar, gracias a un configuración y administración centralizada desde una interfaz HTTP, reglas de conversión basadas en los tipos MIME y ficheros de descripción de impresora estándares (**PPD**, *PostScript Printer Description*).
3. Reconoce los comandos de System V y BSD.
4. Es capaz de interactuar con los servidores de impresión LPG para guardar una compatibilidad ascendente.
5. Dispone de su propia API, que permite crear interfaces de usuarios que pueden integrarse en entornos gráficos o interfaces de administración.
6. La autenticación es posible por usuario, máquina o certificado numérico.

El componente de servidor **cupsd** controla las colas, e incluye un servidor Web para su configuración y gestión. Casi todo puede ser configurado a través de la interfaz web o con el comando **lpadmin**. Los archivos de configuración del directorio **/etc/cups** rara vez tienen que ser editados.

El servidor de impresión se llama **cupsd**.

```
$ps -ef | grep cupsd
root      1348      1  0 10:21 ?
          00:00:00 /usr/sbin/cupsd -C /etc/cups/cup
sd.conf
```

Este ejemplo muestra que cupsd, el demonio cups, se está ejecutando, por lo que CUPS es el sistema de impresión en uso.

CUPS utiliza varios archivos de configuración en el directorio **/etc/cups** y sus subdirectorios para gestionar su funcionamiento. Estos ficheros se pueden editar, y habrá que hacerlo, si queremos compartir impresoras o usar impresoras compartidas por CUPS de otros sistemas de impresión. La forma más sencilla de agregar impresoras en CUPS, sin embargo, es utilizar la utilidad de la herramienta de configuración basada en web.

Como hemos comentado antes, no es necesario utilizar herramientas gráficas para administrar un servidor CUPS. Sin embargo, para facilitar la administración de la impresión, la mayoría de las distribuciones cuentan con ellas. CUPS dispone de una interfaz de administración WEB directamente accesible desde el puerto 631 del servidor. La interfaz funciona con cualquier navegador, escribiendo: <http://localhost:631>.

Al escribir esto nos aparece la pantalla de administración de CUPS que se muestra en la imagen siguiente:

Printers - CUPS 1.2.2 - Firefox

File Edit View History Bookmarks Tools Help

SettingUpSambaPDC - Comm... Printers - CUPS 1.2.2

Printers

Home Administration Classes Documentation/Help Jobs Printers

Search in Printers: Search Clear

Showing 1 of 1 printer.

Sort Descending

laser (Default Printer)

Description: Greyscale laser printer
Location: Server local
Make and Model: Samsung ML-1510 Foomatic/gdi (recommended)
Printer State: idle, accepting jobs, published.
Device URI: usb://Samsung/ML-1520

Print Test Page Stop Printer Reject Jobs Move All Jobs Cancel All Jobs Unpublish Printer Modify Printer
Set Printer Options Delete Printer Set As Default Set Allowed Users

Sort Descending

The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of **Easy Software Products**. CUPS is copyright 1997-2006 by Easy Software Products, All Rights Reserved.

En esta página se muestran una lista de las tareas administrativas que puede realizar. Hacer clic en Printers (Impresoras) para abrir la página de administración de impresoras que vemos en la imagen superior.

Para añadir una impresora, hay que seguir los siguientes pasos:

1. Desde la pestaña Home (Inicio) o Administration (Administración), haga clic en **Add Printer** (Añadir impresora).

The screenshot shows the CUPS 1.3.9 administration interface running in Mozilla Firefox. The title bar reads "Administración - CUPS 1.3.9 - Mozilla Firefox". The menu bar includes "Archivo", "Editar", "Ver", "Historial", "Marcadores", "Dividir (P)", "Herramientas", and "Ayuda". The address bar shows the URL "http://localhost:631/admin". Below the address bar are links for "Más visitados" and "Marcadores rápidos", along with links to "Getting Started" and "Latest BBC Headli...". The main content area has a header "Administración" with a logo for "UNIX PRINTING SYSTEM". A navigation bar below the header includes "Inicio", "Administración" (which is selected), "Clases", "Documentación/Ayuda", "Trabajos", and "Impresoras".

Impresoras

- Añadir impresora
- Encontrar nuevas impresoras
- Administrador impresoras

Clases

- Añadir clase
- Administrador clases

Trabajos

- Administrador trabajos

Servidor

- Editar archivo configuración
- Ver archivo de registro de accesos
- Ver archivo de registro de errores
- Ver archivo de registro de páginas

Configuración básica del servidor:

- Mostrar impresoras compartidas por otros sistemas
- Compartir impresoras públicas conectadas a este sistema
 - Allow printing from the Internet
- Permitir administración remota
- Usar autentificación Kerberos ([FAQ](#))
- Permitir a los usuarios cancelar cualquier trabajo (no sólo los suyos propios)
- Guardar información de depuración para búsqueda de problemas

[Cambiar especificaciones](#)

Subscripciones

- Añadir subscripción RSS

2. El sistema muestra una página que pregunta por el nombre de la impresora, su ubicación y su descripción. Tenemos que introducir la información adecuada en los campos **Name** (Nombre), **Location** (Localización) y **Description** (Descripción). Estos campos son totalmente descriptivos, de modo que podemos introducir lo que quiera (no obstante, los usuarios utilizarán el valor de su campo **Name** para acceder a la impresora). Cuando hagamos clic en **Continuar** (Continuar), CUPS nos preguntará por el dispositivo de impresión.

Añadir impresora - CUPS 1.3.9 - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Dividir (P) Herramientas Ayuda

Más visitados ▾ Marcadores rápidos ▾ Getting Started Latest BBC Headli... ▾ G Google

Añadir impresora

Inicio Administración Clases Documentación/Ayuda Trabajos Impresoras

Añadir impresora nueva

Nombre: Lexmark_Printer
(Puede contener cualquier carácter imprimible excepto "/", "#", y espacio)

Ubicación: Local Printer
(Ubicación fácilmente leible tal como "Lab 1")

Descripción: Lexmark Lexmark Z600 Series
(Descripción fácilmente leible tal como "HP LaserJet de doble cara")

Siguiente

Common UNIX Printing System, CUPS, y el logo de CUPS son marcas registradas de **Apple, Inc.**. Los derechos de copia de CUPS 2007 son de Apple Inc. Todos los derechos reservados.

3. El dispositivo de impresión puede ser un puerto de hardware local (puerto paralelo o un puerto USB), una impresora LPD remota, una impresora SMB/CIFS (Samba) remota u otro dispositivo. Las opciones exactas disponibles varían de una distribución a otra. Selecciona la apropiada de la lista desplegable y haga clic en **Continúe**.



4. Si hemos introducido una impresora de red tenemos varias opciones. La mayoría de las impresoras conectadas a una red de tipo Ethernet o Wi-Fi cuentan con servicios de impresión LPD o Socket (impresión directa). Para el caso de impresión LPD, el resultado es una página que introducirá la ruta completa del dispositivo. La ruta será como la siguiente: *lpd://printserv/brother*, que imprime en la cola *brother* del ordenador *printserv*. Si usamos la opción *Socket* (impresión directa), la propia impresora gestiona las tareas de impresión. Si quisieramos instalar la impresora HP Laserjet 4M suponiendo que disponemos de una tarjeta de red y de un servidor de impresión jetDirect integrado. El URI insertado es *socket://162.168.1.10:9100*. La IP es de la impresora en la red: en este caso, usamos el puerto 9100, que es el estándar. Haga clic en **Continue** (siguiente) cuando haya terminado.

5. Si hemos introducido un dispositivo local en el paso 3 o cuando introduzcamos la ruta completa en el paso 4, verá una lista de fabricantes. Seleccione uno y haga clic en **Continue**. O bien, puede apuntar directamente a un fichero PPD, si tiene uno a mano. Si lo hace, sáltese el siguiente paso.

Añadir impresora

Marca/Fabricante de epson_calidad

Marca:

- Citoh
- Compaq
- DEC
- Dell
- Dymo
- Dymo-CoStar
- Epson**
- Fujifilm
- Fujitsu
- Generic

Siguiente

O proporcione un archivo PPD: Examinar... Añadir impresora

6. Ahora CUPS muestra una lista completa de modelos de impresoras del fabricante que eligió en el paso 5. Puede haber varios drivers para un modelo determinado. En este caso se suele indicar el driver aconsejado con recommended; tendremos que comprobar el driver realmente recomendado para nuestra impresora en el sitio siguiente: <http://www.linux-foundation.org/en/OpenPrinting>, que Para la impresora HP Laserjet 2100M, nos daría el siguiente mensaje:

For basic printing functionality use the Postscript PPD. For advanced functionality such as printer status and maintenance features, use the HPLIP driver (which includes HPIJS).

Por lo tanto, se debe elegir el driver **HP Laserjet 2100m Foomatic/hpijs**.

7. Seleccione un modelo apropiado y haga clic en **Continue**. Como alternativa, puede proporcionar un fichero PPD, si dispone de uno.

Añadir impresora - CUPS 1.2.4 - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Guadalinex Wikipedia

http://localhost:631/admin

Google

UNIX PRINTING SYSTEM

Añadir impresora

Inicio Administración Clases Documentación/Ayuda Trabajos Impresoras

Modelo/Controlador para epson_calidad

Modelo:

- Epson Stylus Color 600 - CUPS+Gutenprint v5.0.0 (en)
- Epson Stylus Color 600 Foomatic/stc600lh.upp (en)
- Epson Stylus Color 600 Foomatic/stcolor (en)
- Epson Stylus Color 640 - CUPS+Gutenprint v5.0.0 (en)**
- Epson Stylus Color 640 Foomatic/st640lh.upp (en)
- Epson Stylus Color 640 Foomatic/stc600lh.upp (en)
- Epson Stylus Color 640 Foomatic/stcolor (en)
- Epson Stylus Color 660 - CUPS+Gutenprint v5.0.0 (en)
- Epson Stylus Color 660 Foomatic/st640lh.upp (en)
- Epson Stylus Color 660 Foomatic/stc600lh.upp (en)

O proporcione un archivo PPD: Examinar...

8. Si no ha accedido aún para realizar otras tareas administrativas, CUPS le preguntará su nombre de usuario y su contraseña. Escriba root como nombre de usuario e introduzca su contraseña de root.

9. CUPS le informa de que se ha añadido la impresora.

10. Finalmente, si espera unos cuantos segundos, la notificación de que la impresora se ha añadido se sustituirá por una página en la que puede definir las opciones específicas de las impresoras, como la alimentación de las páginas y las resoluciones de la impresión. Ajuste las opciones que desee y seguidamente, haga clic en ***Set Printer Options*** (Definir las opciones de impresión).

Cambiar opciones impresora - CUPS 1.2.4 - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Guadalinex Wikipedia

Cambiar opciones impresora

Inicio Administración Clases Documentación/Ayuda Trabajos Impresoras

epson_calidad: General

Media Size: A4
Color Model: RGB Color
Color Precision: Best
Media Type: Inkjet Paper
Print Quality: High
Resolution: 1440 x 720 DPI Highest Quality
OutputOrder: Normal

epson_calidad: Printer Features Common

Ink Set: EPSON Standard Inks

Si hace clic en elemento Printers (Impresoras) de la parte superior de la página, debería regresar a la listas de impresoras (Imagen 1), la nueva impresora aparecería entre las impresoras instaladas. Puede imprimir una página de prueba si hace clic en Print Test Page (Imprimir página de prueba). Si la impresora se instaló correctamente saldrá una página de prueba. En caso contrario, tendríamos que revisar su configuración haciendo clic en **Modify Printer** (Modificar impresora).

Desde la lista de colas de impresoras también puede hacer clic en **Configure Printer** (Configurar impresora) para definir varias opciones de la impresora. Las opciones disponibles dependerán de la impresora, pero las habituales son la resolución, las opciones de mezcla de colores, el tamaño del papel, la activación de la impresión a doble cara y la presencia de la página de encabezado.

La imagen siguiente es una versión más moderna de CUPS:

Nombre de la cola	Descripción	Ubicación	Marca y modelo	Estado
Hewlett-Packard-HP-LaserJet-Professional-P1606dn-2	Hewlett-Packard HP LaserJet Professional P1606dn		HP LaserJet p1505n hpijs pcl3, 3.10.2, requires proprietary plugin	Inactiva
Ricoh_Aficio_MP_8000	Ricoh Aficio MP 8000	192.168.4.231	Ricoh Aficio MP 8000 PXL on c0.04000134.04.andared.cec.junta-andalucia.es	Inactiva
Samsung-ML-2510-Series	Samsung ML-2510 Series	pc186-7	Samsung ML-4500 Foomatic/gdi (recommended)	Inactiva

En el siguiente enlace <http://usuariodebian.blogspot.com/2010/03/cups-configurear-impresoras.html> podemos ver como se instala una impresora con una versión de CUPS más moderna.

Para poder solucionar problemas en un sistema de impresión, es muy importante comprender el flujo de datos y las medidas adoptadas tanto en el servidor como en el cliente. Estas serían las siguientes:

1. Quizás quisiste decir: Un trabajo de impresión es generado por una aplicación de forma local en el lado del cliente.
2. El trabajo de impresión se envía al servidor de impresión especificado por el protocolo seleccionado para esa cola.
3. En el servidor CUPS, el proceso de cola de impresión **cupsd** obtiene el flujo de datos y lo guarda en el directorio de cola de impresión. El directorio predeterminado es **/var/spool/cups**.
4. Si un filtro de entrada se especifica en la configuración, CUPs le pasara el trabajo. En todo caso, después de cualquier filtro, el trabajo se envía al backend. Muchos filtros crean formatos para las impresoras, como PostScript o archivos PDF.
5. El backend envía los datos específicos de la impresora a la impresora.
6. Una vez que haya terminado el trabajo, **cupsd** elimina los archivos respectivos del directorio de spool, dependiendo del tiempo de espera configurado.

CUPS está disponible en <http://www.cups.org>

108.4.6.1. Componentes de CUPS

En Linux, el sistema de impresión CUPS se compone de los siguientes elementos:

cupsd

Este demonio se inicia durante el arranque y se ejecuta de forma continua, escuchando las solicitudes de impresión dirigidos a varias impresoras.

/etc/cups/cupsd.conf

Este archivo configura el demonio **cupsd** que normalmente se encuentra en el directorio */etc/cups*.

Cada línea del archivo puede ser una directiva de configuración, una línea en blanco, o un comentario. Las directivas de configuración son intencionalmente similares a las utilizadas por el software de servidor web Apache.

/etc/cups/printers.conf

Este archivo define las impresoras locales disponibles. Es generado automáticamente por **cupsd** cuando las impresoras se agregan, eliminan o cambian. Este archivo no debe ser cambiado manualmente.

/etc/printcap

Este archivo está presente en el sistema para permitir que las aplicaciones de impresión más antiguas sigan funcionando. En CUPS, el archivo es generado automáticamente por **cupsd** desde el archivo */etc/cups/printers.conf*. Todos los cambios en el archivo se pierden si el servicio CUPS se reinicia.

Las nuevas definiciones de la impresora se añaden en */etc/printcap*.

Ejemplo del contenido de un archivo ***/etc/printcap***:

```
Lp|ljet:\n  :sd=/var/spool/lpd/lp:\
  :mx#0:\
  :sh:\
  :lp=/dev/lp0:\
  :if=/var/spool/lpd/lp/filter:
  :if=/var/spool/lpd/lp/log:
```

Las líneas de este ejemplo tienen los siguientes significados:

Lp|ljet:\n

Este parámetro define dos nombres alternativos para la impresora, lp o ljet.

:sd=/var/spool/lpd/lp:\n

Este parámetro especifica el directorio de spool en /var/spool/lpd.

mx=max_size

El tamaño máximo de un trabajo de impresión en bloques. Al definir este valor a 0 indica que no hay límite.

sh

Suprimir las páginas de encabezado. La colocación de este atributo en printcap supone, la eliminación de las cabeceras.

lp=printer_device

El dispositivo de impresión local, como el puerto paralelo.

if=input_filter

El filtro de entra a utilizar.

if=log_file

El archivo donde los mensajes de error se registran.

Lp

El ***lp*** (line print) envía los archivos y la información corriente a la entrada estándar de las colas de impresión.

Lpq

Este programa de consultas muestra el estado y el contenido de las colas de impresión.

Lprm

Este programa elimina los trabajos de impresión de las colas de impresión.

Lpadmin

Configura las colas de las clases e impresoras proporcionadas por CUPS. También se puede usar para poner la clase o impresora predeterminada del servidor.

lpc

Hoy en día, ***lpc*** proporciona un control limitado sobre las impresoras y las colas de clase proporcionado por CUPS. También se puede utilizar para consultar el estado de las colas. El comando ***lpadmin*** debería utilizarse en su lugar.

Directorio de Spool

El demonio ***cupsd*** usa `/var/spool/cups` para la cola de impresión de datos en espera de la impresión.

Trabajos de impresión

Cada solicitud de impresión enviada se almacena en una cola y se le asigna un número único. Los trabajos de impresión se pueden examinar y manipular según sea necesario.

Otros archivos también son utilizados por determinadas partes del sistema, tales como filtros de entrada.

108.4.6.2. Los backends de CUPS

Los ***backends*** es la manera en que la información es enviada a la impresora. Hay varios backends disponibles para CUPS: puertos como paralelo, serie y USB, como también a través de la red mediante los protocolos IPP, Jet Direct (AppSocket), Line Printer Protocol Daemon (LPD) y SMB.

Existen varios Backends para CUPS: paralelo, serie, SCSI, y USB, así como backends de red que operan en Internet utilizando protocolos como HTTP, HTTPS y IPP, protocolos como JetDirect (AppSocket Por 9100), Line Printer Daemon (LPD), CIFS (que antes se llamaba SMB), y muchos más. También tenemos a nuestra disposición generadores de PDF. Estos paquetes por lo general se encuentran en el directorio `/usr/lib/cups/backend`, y se puede compilar programas o scripts escritos en cualquier lenguaje de scripting, como Perl o Python.

El backend es siempre el último programa ejecutado para el procesamiento de un trabajo de impresión.

Backends utilizados por CUPS		
Backend	Sintaxis URI	Ejemplo URI
Parallel	parallel:/dev/lpnumber	parallel:/dev/lp0
USB	usb://make/model? serial=number	usb://vendor/printer% 201000?serial=A1B2C3
ipp	ipp://host/printers/ queue	ipp://host/printers/ printer1000
LPD	lpd://host/queue	lpd://host/printer
socket	socket://host:port	socket://ip:9100
CIFS (Common Internet Filesystem), el protocolo de Microsoft de sistemas basados en Windows, que se utilizan para la comunicación de archivos e impresoras en una red.	Ver la página de man referente a smbspool(8).	smb://user:password@ workgroup/host/share

108.4.6.3. Filtros de CUPS.

El núcleo del sistema de filtrado de CUPS se basa en Ghostscript, que forma parte del proyecto GNU. Este consulta a los archivos PPD, que son un estándar de la industria para la representación de las funciones de impresora (impresión a doble cara, de cuatro a una página, etc.).

Para las impresoras PostScript, el archivo PPD contiene las opciones específicas de la impresora, junto con los fragmentos de código PostScript que se deben enviar al intérprete de PostScript para activar una opción determinada.

Para las impresoras no PostScript, el archivo PPD contiene información adicional sobre qué programa controlador debe utilizar la impresora y las opciones disponibles para el driver. Si varios drivers se pueden utilizar para una impresora determinada, encontraremos varios archivos PPD.

Dependiendo de las opciones de impresión específicas establecidas para un trabajo de impresión determinado (por ejemplo: -o Page Size=A4), el sistema de filtros lee los fragmentos de código PostScript adecuado desde el archivo PPD y los inserta en el flujo de datos PostScript.

Los datos originales tienen un tipo MIME determinado por las opciones de configuración de /etc/cups/mime.types. Si el tipo no es de aplicación postscript, los datos se convierten en Postscript de acuerdo con el archivo de configuración /etc/cups/mime.convs. Por ejemplo, text/plain se convierte a PostScript con el programa /usr/lib/cups/filter/texttops.

Estos archivos de filtro se encuentran generalmente en el directorio /usr/lib/cups/filter y, como backend de CUPS, se puede compilar el código o ejecutar shell scripts.

108.4.6.4. Gestión de colas de impresión de CUPS.

Como administrador del sistema, es probable que se le pida gestionar y manipular la impresora con más frecuencia. En Linux, el lp, lpq, lpstat, lprm, y los comandos lpadmin son sus herramientas. Otras herramientas que se incluyen son lpoptions, accept, reject y cancel.

lp

Sintaxis

```
lp [ -E ] [ -U nombre de usuario ] [ -c ] [ -d destino[/instancia] ]
[ -h nombre de servidor[:puerto] ] [ -m ]
[ -n número de copias ] [ -o opcion[=valor] ] ] [ -q prioridad ] [ -s ] [ -t titulo ]
[ -H manejo ] [ -P lista-páginas ] [ -- ] [ fichero/s ]
```

Descripción

lp envía los archivos para su impresión o modifica un trabajo pendiente de impresión.

Ejemplo 1

Imprimir un documento de doble cara en una impresora llamada hp1

```
$ lp -d hp1 -o media:legal -o sides=two-sided-long-edge nombre del
fichero
```

Ejemplo2

Impresión de una imagen a través de cuatro página.

```
$ lp -d bar -o scaling=200 nombre del fichero
```

Ejemplo 3

Imprimir un archivo de texto con 12 caracteres por pulgada, 8 líneas por pulgada, y un margen izquierdo de 1 pulgada

```
$ lp -d bar -o cpi=12 -o lpi=8 -o page-left=72 nombre del fichero
```

Cancel

Sintaxis

```
cancel [ -E ] [ -U usuario ] [ -a ] [ -h nombre servidor[:puerto] ] [ -u usuario ] [ id ] ] [ destino-id ]
```

Descripción

Elimina los trabajos de impresión hechos por el comando **lp**.

Opciones

-a

Cancela todos los trabajos del destino especificado, o todos los trabajos en todos los destinos si el destino no ha sido especificado.

lpstat

Sintaxis

```
lpstat [ -E ] [ -U nombre usuario ][ -h servidor impresión[:puerto] ][ -l ] [ -W que-trabajos ]
[ -a [ destino(s) ] ] [ -c [ clase(s) ] ] [ -d ]
[ -o [ destino(s) ] ]
[ -p [ impresora(s) ] ] [ -r ] [ -R ] [ -s ] [ -t ] [ -u [usuario(s) ] ]
[ -v [impresora(s)] ]
```

Descripción

lpstat muestra información de estado sobre las clases actuales, trabajos, e impresoras. Cuando se ejecuta sin argumentos, muestra la cola de trabajos del usuario actual.

Opciones

-a [impresora(s)]

Muestra el estado de aceptación de las colas de impresión. Si no hay ninguna impresora especificada, muestra todas las impresoras.

-t

Muestra toda la información de estado. Esta opción es muy útil para solucionar problemas.

lpadmin

Sintaxis

```
lpadmin [ -E ] [ -U nombre de usuario ] [-h servidor[:puerto] ] -d destino  
lpadmin [ -E ] [ -U nombre de usuario ][-h servidor[:puerto]] -p printer opción(es)  
lpadmin [ -E ] [ -U nombre de usuario ] [-h servidor[:puerto] ] -x destino
```

Descripción

lpadmin configura las impresoras y las colas de clases proporcionados por CUPS. También se puede utilizar para establecer el valor predeterminado del servidor o la clase.

Opciones

-m modelo

Establece un guión (script) de interfaz estandar System V o archivo PPD desde el directorio modelo.

-v uri-dispositivo

Estable el atributo de uri-dispositivo de la cola de impresión. Si uri-dispositivo es un nombre de archivo, automáticamente es convertido a la forma file:/nombre/archivo.

-E

Activa la impresora y hace que acepte trabajos; es lo mismo que ejecutar los programas **accept** y **enable** en la impresora.

lpq

Sintaxis

```
lpq [opciones] [usuarios][trabajos]
```

Descripción

Muestra el estado actual de la cola de impresión de la impresora especificada. Se muestran los trabajos en cola en el destino predeterminado si no se especifica ni impresora ni clase de impresora en la línea de comandos.

Opciones

-l

Hace que se use un formato de listado más detallado.

-P name

Especifica el nombre de la cola de impresión. En ausencia de **-p** la impresora por defecto es consultada.

Ejemplo 1

Examinar los trabajos activos:

```
$ lpq
```

```
lp is ready and printing
Rank   Owner      Job    Files          Total     Size
active  root      193    filter        9443     bytes
1st    root      194    ejemplo.txt  12050     bytes
2nd    root      196    (standard input) 18550     bytes
```

Ejemplo 2

Examina esos mismos trabajos en formato largo:

```
$ lpq -l
```

```
lp is ready and printing
root: active                               [job 193AsJRzIt]
      filter                                9443 bytes
root: 1st                                  [job 194AMj9109]
      ejemplo.txt                           12050 bytes
root: 2nd                                  [job 196A6rUGu5]
      (standard input)                      18998 bytes
```

Ejemplo 3

Examina la cola, que resulta estar vacía:

```
$ lpq -Plp  
no entries
```

Ejemplo 4

Examina los trabajos de juan.

```
$ lpq juan
```

Rank	Owner	Job	Files	Total	Size
7th	juan	202	.bash_history	1263	bytes
9th	juan	204	.bash_profile	5676	bytes

Utilizando los números de trabajo mostrados por lpq, cualquier usuario puede eliminar sus propios trabajos de impresión de la cola. El superusuario puede eliminar cualquier trabajo.

Ejemplo 5

Examina esos mismos trabajos en formato largo:

```
$ lpq -Php4000
```

```
hp400 is ready and printing
```

Rank	Owner	Job	Files(s)	Total	Size
active	juan	1630	file:///	90112	bytes

Es útil conocer el número de la cola, en este ejemplo 1630. Se puede utilizar este número para borrar una tarea de la cola o reordenar éste para que imprima antes otras tareas.

lprm

Sintaxis

```
lprm [ -Pnombre ] [usuario] [ trabajos ]
```

```
lprm -ly
```

Descripción

Cancela los trabajos de impresión que han sido puestos en cola para ser impresos. La opción -P especifica la impresora o clase destino. Si no se especifican argumentos, se cancela el trabajo actual en el destino predeterminado. Puede especificar uno o más números ID de trabajos que sean cancelados, o usar la opción - para cancelar los trabajos.

Ejemplo 1

Como un usuario normal, quita todos los trabajos de impresión:

```
$ lprm -
```

Ejemplo 2

Como superusuario, borra todas las colas de ps:

```
$ lprm -Pps -
```

Algunas veces puede sorprender el ver una respuesta de que no hay entradas de lpq, a pesar de que la impresora este imprimiendo un documento. En tales casos, el buffer de impresión probablemente ha sido vaciado en la memoria de la impresora, y el trabajo ya no está bajo el control del sistema de impresión. Para matar estos trabajos, es necesario utilizar los controles de la impresora y eliminar los trabajos de la memoria.

lpr

Sintaxis

```
lpr [ opciones ] [ ficheros ]
```

Descripción

Realiza la impresión de archivos. Los archivos especificados en la línea de comandos se envían a la impresora especificada (o a la predeterminada si no se ha especificado destino). Si no se especifican archivos en la línea de comandos, entonces lpr lee el archivo a imprimir desde la entrada estándar. Una copia de la fuente de entrada se coloca en el directorio de spool en /var/spool/lpr hasta que el trabajo de impresión este completado.

El comando lpr está accesible tanto para usuarios ordinarios como para root, por lo que cualquiera puede imprimir utilizando. También recurren a éste muchos programas que necesitan imprimir

directamente, como los programas gráficos y los procesadores de texto.

A veces, deseará procesar un fichero de alguna manera antes de enviarlo a la impresora. El programa **mpage**, que lee ficheros en texto plano o PostScript y les vuelve a dar formato para que cada hoja impresa contenga varias páginas con un tamaño reducido respecto al documento original. Este puede ser un buen modo de ahorrar papel si no le importa reducir el tamaño del documento original. En el caso más sencillo, podríamos utilizar mpage igual que utilizariamos lpr:

```
$ mpage -Plexmark report.ps
```

Este comando imprime el fichero report.ps reducido a cuatro páginas por hoja. Podemos cambiar el número de páginas del original a ajustar en cada página impresa con las opciones -1,-2,-4 y -8, que especifican una, dos, cuatro y ocho páginas por cada página de salida, respectivamente. mpage tiene opciones adicionales para controlar características como el tamaño del papel, la fuente a utilizar para los ficheros de entrada de texto plano y el rango de páginas del fichero de entrada a imprimir.

Opciones de uso frecuente:

-#number

Establece el número de copias a imprimir.

-S

En lugar de copiar un archivo al área de spooling de impresión, hace un enlace simbólico al archivo, eliminando así el tiempo de transferencia y almacenamiento en /var/spool/lpr. Esto puede aliviar la carga sobre el sistema de demonios para archivos muy grandes.

-Pname

Especificar el nombre de la cola de impresión. En ausencia de P, la impresora por defecto se realiza una consulta.

-r

Normalmente lpr envía a la cola una copia del fichero a imprimir, dejando intacto el original. Al especificar la opción -r haremos que lpr elimine el fichero original tras imprimirlo.

-h

Suprime el encabezado para una única tarea de impresión. Las versiones anteriores de CUPS no incluyen esta opción, pero si las recientes.

-J nombredetarea

Las tareas de impresión tienen nombres que ayudan a identificarlas, tanto cuando están en la cola como cuando se imprimen (si la cola está configurada para imprimir páginas de encabezado). El nombre suele ser el nombre del primer fichero de la tarea de impresión, pero puede cambiarlo para que incluya, usando la opción -J nombredetarea, las opciones -C y -T son sinónimas.

-m nombredeusuario

Se envía un correo a nombredeusuario cuando la tarea finalice. Esta opción solo la incluyen las últimas opciones de CUPS.

Ejemplo 1

Imprimir el archivo /etc/fstab en la cola de impresión predeterminada

```
# lpr /etc/fstab
```

Ejemplo 2

Imprimir una página de man utilizando la entrada estándar de lpr.

```
# man -t 5 printcap | lpr
```

Ejemplo 3

Desactivar una cola de impresión

```
# lpr disable lp
```

Después, intentar imprimir 3 copias de un fichero en la cola desactivada como superusuario.

```
# lpr -#3 /etc/fstab
```

Lpr: Printer queue is disabled

Como era de esperar, los usuarios normales no pueden imprimir en una cola deshabilitada.

Ejemplo 4

Desactivar una cola de impresión

```
# lpr -Plexmark -m juan report.txt
```

Imprimir el fichero **report.txt** que imprimiremos en la impresora asociada a la cola **lexmark**. Esta cola suele estar ocupado, por lo que deseamos que el sistema envíe un correo electrónico cuando haya acabado a la cuenta **juan**, para informarle de que puede recoger el documento impreso.

108.4.6.5. Editar el fichero CUPS.

Podemos añadir o eliminar impresoras editando el archivo */etc/cups/printers.conf*, donde definimos las impresoras. Cada definición comienza con el nombre de una impresora, identificada por la cadena **DefaultPrinter** (de la impresora por defecto) o **Printer** (para una impresora predeterminado) entre los siguientes símbolos (<>), como en el siguiente ejemplo:

```
<DefaultPrinter okidata>
```

Esta línea marca el comienzo de una definición de una cola de impresión llamada **Okidata**. El final de esta definición es una línea que termina con **</Printer>**. En las líneas intermedias, se definen diversas opciones de la impresora, como las cadenas identificadoras, la ubicación de la impresora (su puerto hardware local o su lugar en red), su estado actual, etc.

Las opciones adicionales se almacenan en un archivo PostScript (PPD, **PostScript Printer Definition**, Definición de impresoras PostScript) que recibe el nombre de la cola y se almacenan en el subdirectorio */etc/cups/ppd*. Los archivos PPD siguen el formato estándar de la industria.

Para las impresoras PostScript, podemos obtener un fichero PPD del fabricante de la impresora, por lo general de una unidad de CD-ROM de drivers o desde el sitio web de los fabricantes. CUPS y sus paquetes de drivers complementarios también vienen con un gran número de ficheros PPD que se instalan automáticamente al utilizar las utilidades de configuración de tipo Web.

Como regla general, es mejor usar las herramientas de configuración Web de CUPS para agregar impresoras en lugar de agregar impresoras editando directamente los archivos de configuración. No obstante, si lo desea, puede estudiar los ficheros subyacentes y manipular las configuraciones utilizando un editor de texto para evitar tener que recurrir a la herramienta Web para realizar un cambio menor.

Una excepción a esta regla tiene que ver con la configuración de la propia herramienta de interfaz Web de CUPS y la capacidad de CUPS para hacer de interfaz entre los demás sistemas de CUPS. Una de las grandes ventajas de CUPS es que utiliza un nuevo protocolo de impresión en red, conocido como IPP (**Internet Printing Protocol**, Protocolo de impresión para Internet), además del antiguo protocolo LPD utilizado por BSD LPD y LPRng. IPP incluye una funcionalidad que denomina **browsing** (navegación), que permite a los ordenadores de una red intercambiar las listas de impresoras de manera automática. Esta funcionalidad simplifica enormemente la configuración de la impresión en red. Puede que necesite cambiar algunos ajustes del principal fichero de configuración de CUPS, */etc/cups/cupsd.conf*, para activar esta característica.

El archivo */etc/cups/cupsd.conf*, es estructuralmente similar a la del archivo de configuración del servidor Web Apache, contiene una serie de bloques de configuración que especifican que otros sistemas deben ser capaces de acceder a él. Cada bloque controla el acceso a un lugar determinado en el servidor. Estos bloques deben tener este aspecto:

Ejemplo 1

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
</Location>
```

Configura el acceso el acceso a *todo* CUPS. Cada directiva *Location* define la configuración para un directorio y sus hijos. Por lo tanto, la configuración del ejemplo 1 se aplica al directorio raíz y todos

sus hijos (que en la práctica es configurar el acceso a todo CUPS).

El significado de la directiva **Order** es el siguiente:

1. **Allow,Deny**: Se permite el acceso a todas las ip's excepto aquellas listadas en las directivas **Deny**
2. **Deny,Allow**: Solo se permite el acceso a las ip's listadas en directivas **Allow**

Para configurar el acceso a las impresoras se ha de definir la configuración para la carpeta **/printers/**:

Ejemplo 2

```
<Location /printers>
Order Deny, Allow
Deny from All
BrowseAllow from 127.0.0.1
BrowseAllow from 192.168.1.0/24
BrowseAllow from @LOCAL
Allow from 127.0.0.1
Allow from 192.168.1.0/24
Allow from @LOCAL
</Location>
```

Directiva order: La línea **Order Deny, Allow** le indica a CUPS el orden en que debería aplicar las directivas **allow** (permitir) y **deny** (denegar); en este caso, las directivas **allow** modifican las directivas **allow**.

Política por defecto: La línea **Deny from All** indica al sistema que rechace todas las conexiones, excepto las que se permitan explícitamente.

Líneas de control de navegación. Las líneas **BrowseAllow** le dicen a CUPS que otros sistemas deben aceptar las solicitudes de navegación. En el ejemplo anterior, acepta conexiones de sí mismo (127.0.0.1), de sistemas de la red 192.168.1.0/24, y de los sistemas conectados a una subred local (@ LOCAL).

Líneas de control de acceso. Las líneas con **Allow** permiten a los sistemas especificados imprimir en impresoras locales. En la mayoría de los casos, las líneas **Allow** son las mismas que las líneas **BrowseAllow**.

Ejemplo 3

```
<Location /printers/>
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 192.168.1. *
</Location>
```

La configuración del ejemplo 3 permite el acceso a las impresoras (imprimir, gestionar trabajos y gestionar impresora) a las ip's 127.0.0.1 (localhost) y toda la clase C 192.168.1.(1-254)

También puede crear una definición que utilice **Allow from All** y después cree líneas **BrowseDeny** y **Deny** para limitar el acceso. No obstante, como norma general, el método mostrado en el ejemplo 2

es más seguro. También puede ser importante lugares distintos a **/printers**. Por ejemplo, raíz (/) que especifica los permisos de acceso por defecto a todos los demás lugares, **/job** que permite la gestión de trabajos y un sitio **/admin** que controla el acceso a las funciones de administración de cups.

Antes de las definiciones de ubicación de cupsd.conf, hay unos cuantos parámetros que activan o desactivan la navegación y otras operaciones de red. Tenemos que buscar las siguientes opciones:

1. **Activar la navegación.** La directiva **Browsing** especifica si se buscan o no impresoras remotas. La directiva **Browsing** acepta los valores **On** y **Off**. Por defecto CUPS tiene la activo browsing (**Browsing On**), pero algunas distribuciones Linux tienen esta opción desactivada por defecto.
2. **Control de acceso de navegación.** La directiva **BrowseAddress** especifica la dirección de multidifusión a la que se deberá enviar la información de navegación. Por ejemplo, para difundir los datos de sus impresoras por la subred 192.168.1.0/24, habría que especificar **BrowseAddress 192.168.1.255**.

Una vez que ha configurado el servidor CUPS para dar acceso a otros sistemas a sus impresoras a través de las direcciones de los sitios pertinente y una vez que se han configurado los sistemas cliente para utilizar la navegación a través de **Browsing On**, todos los sistemas en la red deberían detectar automáticamente todas las impresoras de la red. No es necesario configurar la impresora en ningún equipo, excepto aquel al que está directamente conectado. CUPS propagará automáticamente todas las características de las impresoras, incluyendo su ubicación en la red y archivos PPD. Esta característica es más importante al configurar redes de gran tamaño con muchas impresoras o impresoras o redes en las que con frecuencia se agregan o eliminan impresoras.
Enlace a las directivas que reconoce [cupsd.conf](#).

108.4.6.6. Obtener las definiciones de impresora de CUPS.

La mayoría de las distribuciones de Linux vienen con un filtro inteligente de CUPS que admite varias impresoras. Si no se encuentra la suya, se pueden buscar definiciones de impresora adicionales. Estas definiciones pueden consistir en ficheros PPD, el "pegamento" apropiado que se aplica entre bastidores para indicarle a CUPS como utilizarlas, además de posibles ficheros de driver de Ghostscript.

Se pueden obtener estas definiciones de impresora de varias maneras:

1. **Su distribución Linux:** Muchas distribuciones vienen con definiciones de impresora adicionales bajo varios nombres, por lo que debería buscar en su distribución un paquete de este tipo. Muchas distribuciones incluyen algunos de los paquetes de drivers que se describen a continuación.
2. **Foomatic:** El sitio Web de impresión de Linux alberga un conjunto de utilidades y definiciones de impresora que en conjunto se conocen como Foomatic (www.linuxfoundation.org/en/OpenPrinting/Database/Foomatic), que proporcionan definiciones de impresora para CUPS adicionales (así como para otros sistemas de impresión).
3. **Gutenprint:** Los drivers de Gutenprint, originalmente conocidos como GIMP Print por el programa de manipulación de imágenes GIMP, admiten una amplia variedad de impresoras. Consulte <http://gimp-print.sourceforge.net> para obtener más información.
4. **CUPS DDK:** El kit de desarrollo de drivers de CUPS (DDK; www.cups.org/ddk/) es un conjunto de herramientas diseñado para simplificar el desarrollo de drivers de CUPS. Viene con un conjunto de drivers para impresoras Hewlett Packard y Epson.
5. **Los fabricantes de impresoras:** Algunos fabricantes de impresoras proporcionan drivers CUPS para sus impresoras. Puede que se trate de los drivers de Foomatic, Gutenprint y otros de código abierto; pero unos cuantos proporcionan drivers propietarios, algunos de los cuales incluyen funcionalidades de impresión que los drivers de código abierto no poseen.

Es bastante probable que su instalación estándar incluya a su impresora, en particular si su

distribución ha instalado los paquetes Foomatic o Gutenprint. De todos modos, si empieza a configurar impresoras y no localiza su modelo, debería buscar un conjunto adicional de definición de impresoras entre los indicados anteriormente.

108.4.7. Solución de problemas generales de impresión.

Los ficheros log son el primer y algunas veces la mejor guía para resolver problemas de impresión. Muchas personas todavía comenten el error de olvidar consultar los archivos de registro. Estos archivos se rotan, de modo que puedes encontrar los últimos sucesos en el archivo principal y los eventos más antiguos en formato gzip en archivos de copia de seguridad. Si necesitas más detalle, puedes cambiar el valor de la entrada **LogLevel** (esta especifica el nivel de registro: none (ninguno), warn (avisos), error (errores), info (información), debug (depuración), o debug2 (depuración2)) de la línea `/etc/cups/cupsd.conf`. Se puede extraer un montón de información adicional en los archivos de registro de las operaciones de impresión.

También debemos conocer el comando `cups-config`, que tiene algunas opciones que muestran información acerca del estado actual del sistema.

108.4.7.1. El fichero de registro error.

Los errores recientes y la información relacionada se puede encontrar en `/var/log/cups/error_log`. Este archivo muestra los mensajes del scheduler, que incluye tanto los errores como las advertencias. Tú puedes ver información detallada y en tiempo real acerca de la transferencia de datos, filtrado, mensajes, etc. Ejemplos de mensajes generados por un de trabajo típico son los siguientes:

```
I [16/Nov/2011:11:19:07 +0100] [Job 102] Adding start banner page "none".
I [16/Nov/2011:11:19:07 +0100] [Job 102] Adding end banner page "none".
I [16/Nov/2011:11:19:07 +0100] [Job 102] File of type application/postscript queued by "usrpru".
I [16/Nov/2011:11:19:07 +0100] [Job 102] Queued on "PDF" by "usrpru".
I [16/Nov/2011:11:19:07 +0100] [Job 102] Starter filter
/usr/libexec/cups/filter/pstops (PID 18223)
I [16/Nov/2011:11:19:07 +0100] [Job 102] Started backend
/usr/libexec/cups/backend/cups-pdf (PID 18224)
I [16/Nov/2011:11:19:07 +0100] [Job 102] Completed sucessfully.
I [16/Nov/2011:11:20:17 +0100] [Job ???] Request file type is application/postscript.
I [16/Nov/2011:11:20:17 +0100] [Job 103] Adding start banner page "none".
I [16/Nov/2011:11:20:17 +0100] [Job 103] Adding end banner page "none".
I [16/Nov/2011:11:20:17 +0100] [Job 103] File of type application/postscript queued by "usrpru".
I [16/Nov/2011:11:20:17 +0100] [Job 103] Queued on "PDF" by "usrpru".
I [16/Nov/2011:11:20:17 +0100] [Job 103] Started filter /usr/libexec/cups/filter/pstops (PID 18340)
I [16/Nov/2011:11:20:17 +0100] [Job 103] Started backend
/usr/libexec/cups/filter/pstops (PID 18341)
I [16/Nov/2011:11:20:17 +0100] [Job 103] Completed sucessfully.
```

La I con el que comienza cada línea representa la "información". En este caso, no se han generado ni errores ni advertencias (warnings).

```
E [10/Jan/2012:09:43:30 +0100] cupsdReadClient: 16 IPP Read Error!
E [10/Jan/2012:09:51:56 +0100] [cups-driverd] Bad driver information file
"/usr/share/cups/drvc/sample.drv"!
E [10/Jan/2012:09:51:56 +0100] [cups-driverd] Skipping "/usr/share/ppd/Epson": loop detected!
E [10/Jan/2012:09:52:28 +0100] [cups-driverd] Bad driver information file
"/usr/share/cups/drvc/sample.drv"!
E [10/Jan/2012:09:52:28 +0100] [cups-driverd] Skipping "/usr/share/ppd/Epson": loop detected!
E [10/Jan/2012:12:03:35 +0100] [Job 954] Unable to write print data: Broken pipe
D [10/Jan/2012:12:03:35 +0100] [Job 954] The following messages were recorded from 11:58:01 to
12:03:35
D [10/Jan/2012:12:03:35 +0100] [Job 954] Adding start banner page "none".
D [10/Jan/2012:12:03:35 +0100] [Job 954] Adding end banner page "none".
D [10/Jan/2012:12:03:35 +0100] [Job 954] File of type application/pdf queued by "usuario".
D [10/Jan/2012:12:03:35 +0100] [Job 954] hold_until=0
D [10/Jan/2012:12:03:35 +0100] [Job 954] Queued on "Hewlett-Packard-HP-LaserJet-Professional-
P1606dn-2" by "usuario".
D [10/Jan/2012:12:03:35 +0100] [Job 954] job-sheets=none,none
```

Otro ejemplo del contenido del fichero error_log.

108.4.7.2. EL fichero de registro page.

Este archivo se puede encontrar en /var/log/cups/page_log. Contiene la información de cada página que envíe a una impresora. Cada línea contiene la siguiente información:

**printer user job-id date-time page-number num-copies job-billing\\
job-originating-host-name jobname media sides**

Un ejemplo de lo que muestra el fichero:

```
Samsung_ML1710 usrpru 86 [15/Nov/2011:12:48:36 +0100] 1 1 - localhost  
Samsung_ML1710 usrpru 86 [15/Nov/2011:12:48:52 +0100] 2 1 - localhost  
HPLaserJet_5M usrpru 87 [16/Nov/2011:13:40:36 +0100] 1 1 - localhost  
HPLaserJet_5M usrpru 87 [16/Nov/2011:13:40:36 +0100] 2 1 - localhost  
HPLaserJet_5M usrpru 88 [17/Nov/2011:09:20:32 +0100] 1 1 - localhost  
PDF root 100 [18/Nov/2011:10:22:25 +0100] 1 1 - localhost  
PDF usrpru 100 [18/Nov/2011:10:28:12 +0100] 1 1 - localhost
```

108.4.7.3. El fichero de registro Access.

Este archivo se puede encontrar en /var/log/cups/access_log. En él se enumeran cada uno de los recursos HTTP accesibles por un navegador web o cliente. Cada línea está en una versión ampliada del formato de registro de llamadas común utilizado por muchos servidores web y herramientas de la web de información.

```
Localhost - - [16/Nov/2011:17:28:29 +0100] "POST / http/1.1" 200 138 \ CUPS-Get-Default  
successful-ok  
Localhost - - [16/Nov/2011:17:28:29 +0100] "POST / http/1.1" 200 552 \ CUPS-Get-Printers  
successful-ok  
Localhost - - [16/Nov/2011:17:28:29 +0100] "GET / printers http/1.1" \ 200 11258 --  
Localhost - root [16/Nov/2011:17:28:29 +0100] "GET \ /images/button-search.gif HTTP/1.1 200 332 --  
Localhost - root [16/Nov/2011:17:28:29 +0100] "GET \ /images/button-clear.gif HTTP/1.1 200 279 --
```

Otro ejemplo del contenido del fichero access_log.

```
192.168.7.246 - - [10/Jan/2012:08:39:27 +0100] "POST /printers/Samsung-ML-2510-Series HTTP/1.1" 200 718714 Print-Job successful-ok  
localhost - - [10/Jan/2012:09:24:08 +0100] "POST /printers/Hewlett-Packard-HP-LaserJet-Professional-P1606dn HTTP/1.1" 200 320 Create-Job successful-ok  
localhost - - [10/Jan/2012:09:24:08 +0100] "POST /printers/Hewlett-Packard-HP-LaserJet-Professional-P1606dn HTTP/1.1" 200 132678 Send-Document successful-ok  
localhost - - [10/Jan/2012:09:24:40 +0100] "POST /admin/ HTTP/1.1" 401 217 Pause-Printer successful-ok  
localhost - root [10/Jan/2012:09:24:40 +0100] "POST /admin/ HTTP/1.1" 200 217 Pause-Printer successful-ok  
localhost - - [10/Jan/2012:09:25:00 +0100] "POST / HTTP/1.1" 401 234 CUPS-Get-Devices successful-ok  
localhost - root [10/Jan/2012:09:25:00 +0100] "POST / HTTP/1.1" 200 1322 CUPS-Get-Devices -  
localhost - - [10/Jan/2012:09:25:02 +0100] "POST /admin/ HTTP/1.1" 401 168 Resume-Printer successful-ok  
localhost - root [10/Jan/2012:09:25:02 +0100] "POST /admin/ HTTP/1.1" 200 168 Resume-Printer successful-ok  
localhost - - [10/Jan/2012:09:25:21 +0100] "POST /printers/Hewlett-Packard-HP-LaserJet-Professional-P1606dn HTTP/1.1" 200 326 Create-Job successful-ok  
localhost - - [10/Jan/2012:09:25:21 +0100] "POST /printers/Hewlett-Packard-HP-LaserJet-Professional-P1606dn HTTP/1.1" 200 215707 Send-Document successful-ok  
localhost - - [10/Jan/2012:09:25:57 +0100] "POST /printers/Hewlett-Packard-HP-LaserJet-Professional-P1606dn HTTP/1.1" 200 328 Create-Job successful-ok  
localhost - - [10/Jan/2012:09:25:57 +0100] "POST /printers/Hewlett-Packard-HP-LaserJet-Professional-P1606dn HTTP/1.1" 200 32777 Send-Document successful-ok  
localhost - - [10/Jan/2012:09:26:55 +0100] "POST /printers/Hewlett-Packard-HP-LaserJet-Professional-P1606dn HTTP/1.1" 200 12113397 Print-Job successful-ok  
localhost - - [10/Jan/2012:09:28:12 +0100] "POST /jobs/ HTTP/1.1" 200 158 Cancel-Job successful-ok
```

108.4.7.4. El uso de la utilidad cups-config para la depuración

La utilidad de configuración de cups-config tiene varios parámetros que pueden ser de utilidad para la solución de problemas. Estas opciones se describen en la tabla siguiente:

Opciones para cups-config

<i>Opción</i>	<i>Descripción</i>
--cflags	Muestra las opciones del compilador.
--datadir	Muestra el directorio de datos CUPS por defecto.
--help	Muestra la ayuda.
--ldflags	Muestra las opciones necesarias del enlazador.
--libs	Muestra las librerías necesarias para el enlazador.
--serverbin	Muestra el directorio binario CUPS por defecto, donde los filtros y backend son almacenados.
--serverroot	Muestra la configuración por defecto del directorio de archivos CUPS.

108.4 EXTRAS

108.4 EXTRAS Breve introducción a NFS

El **Network File System** (*Sistema de archivos de red*), o **NFS**, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales. El protocolo NFS está incluido por defecto en los Sistemas Operativos UNIX y la mayoría de distribuciones Linux. Originalmente fue desarrollado en 1984 por Sun Microsystems, con el objetivo de que sea independiente de la máquina, el sistema operativo y el protocolo de transporte, para ello el servicio NFS utiliza las llamadas a procedimientos remotos basadas en el protocolo RPC (del inglés, Remote Procedure Call) que permite desde un equipo (cliente) ejecutar código ubicado en otro equipo remoto (servidor) mediante el establecimiento de sockets (IP+puerto) entre ambas.

Aunque al servicio se le suele conocer con el nombre NFS, realmente NFS es un protocolo de nivel de Aplicación y por debajo, el protocolo subyacente que utiliza NFS son las Llamadas a Procedimientos Remotos (RPC) de nivel de Sesión, también utiliza TCP/UDP en el nivel Transporte e IP en el nivel de Red.

NFS es un protocolo sin memoria (state-less) en algunas de sus versiones. Es decir, el servidor no recuerda las solicitudes anteriores. Por tanto, cada llamada a un procedimiento contiene toda la información necesaria para su finalización. Si el servidor NFS falla, el sistema cliente repetirá las solicitudes de NFS hasta que obtenga una respuesta. Además, el servidor no realiza tareas de recuperación frente a fallos.

Las versiones de NFS mas importantes son NFSv2 (RFC 1094), NFSv3 (RFC 1813) y NFSv4 (RFC 3530). En general las versiones 2 y 3 de NFS permiten controlar la exportación y montaje de sistemas de archivos en función del equipo que hace la solicitud, pero no del usuario. Es decir no se contempla un control de acceso al sistema de archivos por usuario. Sólo para los equipos. Esto implica que si un sistema de archivos es exportado desde el servidor NFS, cualquier usuario de un equipo remoto cliente NFS podría acceder a él. Los únicos mecanismos de seguridad que quedan en este caso son los permisos de acceso (sólo lectura) o utilizar un usuario y grupo únicamente. En el caso de la versión 4 de NFS (<http://www.nfsv4.org>) estos problemas de seguridad desaparecen pero, a cambio, tiene unos requerimientos de configuración y servicios adicionales mucho mas importantes. Por ejemplo, en la versión 4 la utilización de mecanismos para la autenticación de los usuarios es obligatoria. Para ello y en función del tipo de seguridad seleccionada, se requiere la utilización del servicio Kerberos cuya misión será funcionar como servidor de entrega de tickets (KDC) y que debe estar configurado y funcionando correctamente antes de configurar el servidor NFSv4. Este requerimiento proporciona seguridad al servicio NFS a cambio de incluir mayor complejidad a su configuración y puesta a punto.

Los demonios imprescindibles del servicio NFS son los siguientes:

- **rpc.mountd:** demonio para el montaje remoto. Se ejecuta en el servidor. Recibe la petición de montaje desde un cliente NFS y comprueba en el archivo /var/lib/nfs/xtab si el sistema de archivos está exportado. Si el sistema de archivos está disponible, permite las solicitudes de acceso de NFS y después proporciona información sobre los sistemas de archivos mediante el comando showmount. Comprueba también que el cliente tenga permiso para solicitar acceso.
- **rpc.nfsd:** demonio para servir archivos. Gestiona las solicitudes del cliente una vez mountd ha dado el visto bueno al cliente. Se pueden arrancar varias copias de este demonio. Utiliza el puerto TCP/UDP 2049.
- **rpc.portmap:** es el encargado de decir a los clientes donde está localizado (número de puerto) el servicio real en el servidor. Como los servicios basados en RPC utilizan portmap para atender las peticiones de los clientes, este servicio debe estar disponible antes de cualquier otro servicio o demonio de NFS. No se utiliza en NFSv4. Utiliza el puerto TCP/UDP 111. Para comprobar que está activo ejecutar la orden:

```
$ sudo portmap status
```

- **rpc.lockd:** encargado de proporcionar el servicio de bloqueo de archivos para asegurar su consistencia ya que pueden ser accedidos de forma concurrente. Se ejecuta tanto en el servidor como en el cliente.
- **rpc.statd:** trabaja conjuntamente con lockd para permitir la recuperación en caída de sistemas. Mantiene información sobre los procesos en los clientes que poseen locks de archivos de determinado servidor. Cuando el servidor NFS se recupera statd informa a los otros procesos statd de los clientes, que el servidor se ha recuperado, y así ellos intentarán resolver los locks que tenían anteriormente. En los clientes statd se utiliza para avisar al servidor de que el cliente ha caído y así poder liberar los archivos que tuviera ese cliente bloqueados.

Los demonios estarán escuchando es sus puertos correspondientes. Podemos comprobarlo ejecutando la orden:

```
$ sudo netstat -tunpl

tcp 0 0 0.0.0.0:2049 0.0.0.0:*
ESCUCHAR -

tcp 0 0 0.0.0.0:111 0.0.0.0:*
ESCUCHAR -

udp 0 0 0.0.0.0:2049 0.0.0.0:*
-

udp 0 0 0.0.0.0:111 0.0.0.0:*
-

.....
```

Los archivos de configuración del servicio NFS son los siguientes:

- **/etc/fstab:** contiene los sistemas de archivos que pueden ser montados desde sistemas remotos en secuencia de arranque del equipo.
- **/etc(exports:** contiene una lista de los directorios del sistema local que se van a exportar a sistemas remotos utilizando NFS y los permisos de uso. La existencia de este archivo determina si el sistema local es un servidor de NFS. Este archivo contiene una línea por cada directorio a compartir.
- **/var/lib/nfs/etab:** contiene una lista de los sistemas de archivos actualmente exportados para el sistema local. Esta información es actualizada en este archivo cuando se ejecuta el comando exportfs que lee el archivo /etc/exports.
- **/etc/hosts.allow y /etc/hosts.deny:** NFS utiliza estos archivos para comprobar a qué máquinas se les acepta o deniega el uso de NFS. En general este sistema de comprobación se suele conocer con el nombre de wrappers TCP

Exportación de un directorio

La estructura de las líneas del archivo /etc/exports es la siguiente:

```
directorio
equipo1(opcion11,...)
equipo2(opcion21,...)
```

Donde:

directorio: es el **nombre del** directorio que se comparte.

EquipoX: son los clientes NFS que tendrán acceso al directorio compartido. Estos equipos se pueden identificar mediante su dirección IP o su nombre DNS (si se tiene disponible un servidor DNS). Se admite la utilización de los comodines '*' y '?', aunque su utilización puede ser algo peligrosa sino se conoce bien como se producirá su expansión.

optionXY: son las diferentes opciones que asignamos a este directorio para ese equipo en concreto y que determinarán los privilegios de acceso a él. De todas las opciones disponibles, las más significativas son:

- **ro|rw:** el directorio será compartido en solo lectura (*ro*) y es la opción por defecto. El directorio será compartido en lectura y escritura (*rw*).

- **sync|async**: *sync* comunica al usuario los cambios realizados sobre los archivos cuando realmente se han ejecutado y es la opción recomendada. La opción *async* mejora el rendimiento y agiliza el funcionamiento del servicio, pero puede generar archivos corruptos si se produce algún tipo de fallo en el servidor.
- **no_subtree_check**: permite que no se compruebe el camino hasta el directorio que se exporta, en el caso de que el usuario no tenga permisos sobre el directorio exportado.
- **root_squash | no_root_squash | all_squash**
 - root_squash indica que un usuario identificado como *root* tendrá acceso al directorio compartido sólo con privilegios de usuario anónimo. De esta forma se ha degradado al *root* al usuario local de privilegios más bajos protegiendo así los archivos en el servidor NFS. Esta opción se conoce también con el nombre de 'aplastamiento del *root*'. Para el resto de usuarios se intenta conservar su *UID* y *GID* en el servidor.
 - no_root_squash desactiva la opción anterior, es decir, los accesos realizados como *root* desde el cliente serán también de *root* en el servidor NFS.
 - all_squash indica que todos los clientes, incluido *root*, tendrán acceso al directorio con privilegios de un usuario anónimo. No se mantienen los *UID* y *GID* de ningún usuario.
 - Si se utiliza alguna de las opciones *squash* podemos indicar cuál es el *UID* y *GID* del usuario con el que se quiere que se acceda, en lugar del anónimo. En este caso hemos de indicar a continuación de la opción *squash* lo siguiente:

(rw,all_squash,anonuid=1002,anongid=1002)

Y significa que la conexión del cliente NFS se hará con los *UID* y *GID* 1002

Los ficheros /etc/hosts.allow y /etc/hosts.deny tienen la siguiente estructura:

**servicio: host [o red/mascara_subred],
host [o red/mascara_subred]**

Donde:

servicio : servicio permitido o denegado para algunos equipos (IP).

host [o red/mascara_subred] : dirección IP del host de un cliente.

Archivo /etc/hosts.deny

Incluimos todas las restricciones que harán mas seguro nuestro sistema.

En nuestro caso denegamos el acceso a portmap desde cualquier IP. De esta forma sólo tendrán acceso a portmap los equipos que incluyamos en /etc/hosts.allow.

El contenido de /etc/hosts.deny será:

portmap:ALL

Archivo /etc/hosts.allow

Incluimos a qué equipos permitimos el acceso al servicio de nfs y portmap. Podemos indicar hosts individuales o una red.

**portmap:192.168.0.0/255.255.255.0
nfs:192.168.0.0/255.255.255.0**

Después de configurar estos archivos hay que relanzar los servicios.

Instalación de NFS

Para poder disfrutar del servicio de compartir carpetas en la red mediante NFS, en el PC servidor es necesario instalar el paquete del **servidor NFS**. Lo normal es que todos los PCs dispongan del paquetes servidor de NFS ya que en cualquier momento puede existir la necesidad de tener que compartir una carpeta desde cualquier PC, aunque lo habitual es que el único que comparta sea el servidor. Que un PC de un usuario tenga instalado el paquete del servidor NFS, no significa que automáticamente esté compartiendo su sistema de archivos en la red. Para ello es necesario configurar y arrancar el servicio.

Si deseamos instalar la última versión disponible, podemos hacerlo con apt-get desde una consola de root:

```
// Instalación de NFS en Debian  
# apt-get install nfs-common nfs-kernel-server
```

//Instalación CentOS

```
#yum -y install nfs-utils
```

Configuración del servidor NFS

Antes de arrancar el servicio NFS, es necesario indicar qué carpetas deseamos compartir y si queremos que los usuarios accedan con **permisos de solo lectura o de lectura y escritura**. También existe la posibilidad de establecer desde qué PCs es posible conectarse. Estas opciones se configuran en el archivo /etc(exports

```
// Archivo de configuración del servidor NFS  
/etc(exports
```

En cada línea del archivo de configuración del servidor NFS /etc(exports, se puede especificar:

- La carpeta que se quiere compartir
- El modo en que se comparte (solo lectura 'ro' o lectura y escritura 'rw')
- Desde qué PC o PCs se permite el acceso (nombre o IP del PC o rango de IPs)

A continuación mostramos un sencillo archivo /etc(exports para configurar algunas carpetas compartidas

// Ejemplo de archivo /etc(exports de configuración del servidor NFS:

```
# Compartir la carpeta home del servidor  
# en modo lectura y escritura y accesible desde la red 192.168.0.0/24  
/home 192.168.0.0/255.255.255.0(rw)  
  
# Compartir carpeta tmp a todos como 'solo-lectura'  
/tmp *(ro)
```

```
# Compartir carpeta /var/log a un PC como 'solo-lectura'  
/var/log 192.168.0.211(ro)
```

Cuando se comparte por NFS, se recomienda restringir al máximo los permisos. Si los usuarios no tienen la necesidad de escribir, debemos compartir con permiso de 'solo lectura'. Si los usuarios solo se conectan desde nuestra red 192.168.0.0/24, debemos permitir el acceso solo desde dicha red.

Nota

Los permisos de compartición por NFS (y en realidad en todos los sistemas de ficheros de red) no excluyen a los permisos del sistema unix sino que **prevalecen los más restrictivos**. Si una carpeta está compartida con permiso NFS de lectura y escritura pero en los permisos del sistema solo disponemos de permiso de lectura, no podremos escribir. Si una carpeta está compartida con permisos NFS de lectura y disponemos de permisos de lectura y escritura en el sistema, tampoco podremos escribir. Para poder escribir necesitaremos disponer permiso de lectura y escritura tanto en los permisos del sistema como en los permisos de compartición NFS. De igual forma, si compartimos la carpeta /home con permisos de lectura y escritura pero el usuario pepe solo tiene acceso a la carpeta /home/pepe, no podrá acceder a ninguna otra carpeta dentro de /home ya que los permisos del sistema se lo impedirán.

109 FUNDAMENTOS DE RED

109.1. Fundamentos de los protocolos de Internet.

109.2 Configuración básica de red.

109.3 Soluciones para problemas simples de red.

109.4 Configuración del DNS cliente.

109.1. Fundamentos de los protocolos de Internet.

Peso en el examen de certificación: 4 puntos.

Objetivo: Demostrar un conocimiento apropiado de los fundamentos de las redes TCP/IP.

Conceptos y áreas de conocimiento:

- Máscaras de red.
- Diferencias entre el direccionamiento IP público y privado.
- Configuración de una ruta por defecto.
- Puertos comunes TCP y UDP (20, 21, 22, 23, 25, 53, 80, 110, 119, 139, 143, 161, 443, 465, 993, 995).
- Diferencias y principales características de UDP, TCP and ICMP.
- Principales diferencias entre IPv4 y IPV6.

Términos y utilidades

- etc/services
- ftp
- telnet
- host
- ping
- dig
- traceroute
- tracepath

ÍNDICE

109.1.1. La pila de protocolos TCP/IP

109.1.2. Arquitectura TCP/IP comparada con el modelo OSI

 109.1.2.1. El protocolo IP

 109.1.2.2. El protocolo ICMP

 109.1.2.3. El protocolo TCP

 109.1.2.4. El protocolo UDP

 109.1.2.5. El protocolo ARP

109.1.3. Direccionamiento en redes TCP/IP

 109.1.3.1. Direcciones públicas i privadas

 109.1.3.2. Clases de direcciones IP

 109.1.3.2.1. Clase A

 109.1.3.2.2. Clase B

 109.1.3.2.3. Clase C

 109.1.3.2.4. Clase D

 109.1.3.2.5. Clase E

 109.1.3.3. Máscaras de red

 109.1.3.4. Subredes

109.1.3.4.1 Cálculo de una subred

109.1.4. IP v6

109.1.5. Puertos

109.1.6. Comandos

 109.1.6.1. FTP

 109.1.6.2. Telnet

 109.1.6.3. Host

 109.1.6.4. Ping

 109.1.6.5. Traceroute

 109.1.6.6. Tracepath

 109.1.6.7. Dig

 109.1.6.8. Route

109.1.1 LA PILA DE PROTOCOLOS TCP/IP.

Los protocolos de la familia TCP / IP eran utilizados inicialmente en sistemas UNIX, con el crecimiento de las redes, se han convertido en el estándar de hecho de Internet.

El nombre de la arquitectura TCP / IP viene dado por el nombre de sus dos protocolos más representativos: el protocolo IP del nivel de red y el protocolo TCP del nivel de transporte.

Los protocolos más relevantes que forman esta familia son:

- Protocolo IP
- Protocolo ICMP
- Protocolo TCP
- Protocolo UDP
- Protocolo ARP

109.1.2.- Arquitectura TCP/IP comparada con el modelo OSI

A continuación, se comparan las capas que definen la arquitectura TCP / IP con el modelo de referencia:

El modelo OSI vs TCP/IP

Capa	Capa
Aplicación	Aplicación
Presentación	
Sesión	
Transporte	Transporte
Red	Internet
Enlace física	Interfaz de red

109.1.2.1- El protocolo IP

El protocolo **IP** (*Internet Protocol*) es el protocolo a nivel de red de ARPANET, el utilizado en los sistemas UNIX.

Es un protocolo pensado para que sus paquetes sean encaminados entre las diferentes redes de Internet. Proporciona el direccionamiento de red, es decir, sabe si una dirección IP de destino pertenece o no a una determinada red. También realiza funciones de enrutamiento que consiste en determinar por qué ruta debe enviar un paquete IP. Es el protocolo base para las transferencias de datos en Internet.

Características básicas:

- No orientado a conexión.
- No confiable: la confiabilidad es proporcionada por la capa de transporte cuando se utiliza el protocolo TCP.

Los paquetes IP tienen un tamaño máximo de 64KB (65536 bytes) y una cabecera de 20 bytes.

Es un protocolo pensado para que sus paquetes sean encaminados entre las diferentes redes de Internet. Proporciona el direccionamiento de red, es decir, sabe si una dirección IP de destino pertenece o no a una determinada red. También realiza funciones de enrutamiento que consiste en determinar por qué ruta debe enviar un paquete IP. Es el protocolo base para las transferencias de datos en Internet.

Características básicas:

- No orientado a conexión.
- No confiable: la confiabilidad es proporcionada por la capa de transporte cuando se utiliza el protocolo TCP.

Los paquetes IP tienen un tamaño máximo de 64KB (65536 bytes) y una cabecera de 20 bytes.

109.1.2.2- El protocolo ICMP

El protocolo **ICMP** (*Internet Control Message Protocol*) es el protocolo de supervisión de los sucesos de la red. Este protocolo es la base del comando ping.

Mensajes de supervisión más habituales:

- Destino inalcanzable: no podemos encontrar la subred de la dirección de destino.
- Tiempo excedido: el tiempo de vida del paquete IP se ha agotado.

109.1.2.3- El protocolo TCP

El protocolo **TCP** (*Transmission Control Protocol*) es un protocolo de la capa de transporte que interactúa con el protocolo de red IP.

Convierte los bloques de la capa superior, la capa de aplicación, en fragmentos de 64KB.

El protocolo TCP:

- Es orientado a conexión.
- Ordenar los segmentos al destino.
- Es confiable: proporciona control de errores y de flujo extremo a extremo.

- Control de errores:
 - Controla que los segmentos lleguen al destino.
 - Retransmisión de segmentos.
 - Control de flujo.

La confiabilidad es necesaria en aplicaciones del tipo cliente-servidor o de correo electrónico, entre otros. Esta propiedad que ofrece el TCP pero genera más tráfico en la red: cabeceras más grandes (20 bytes) y segmentos de confirmación (ACK), que no son necesarios en conexiones no confiables.

109.1.2.4- El protocolo UDP

El protocolo **UDP** (*User Datagram Protocol*) es el otro protocolo de la capa de transporte que se encapsula con el protocolo de red IP.

El protocolo UDP:

- Es no orientado a conexión.
- No ordena los datagramas al destino. Si la aplicación de destino necesita la información recibida ordenada deberá implementar la ordenación de los datagramas en el protocolo de la capa de aplicación.
- Es no confiable: no garantiza la entrega de los datagramas.

El protocolo UDP por ser no confiable ofrece más rendimiento que el protocolo TCP en detrimento de garantizar el envío de un datagrama.

El protocolo UDP tiene una cabecera de 8 bytes.

Algunos servicios que utilizan el protocolo UDP son TFTP, DHCP o el servicio de DNS.

109.1.2.5- El protocolo ARP

El protocolo **ARP** (*Address Resolution Protocol*) o protocolo de resolución de direcciones, sirve para obtener la dirección MAC o dirección física que se utiliza a nivel de enlace a partir de la dirección IP o dirección lógica.

Funcionamiento de una solicitud ARP:

1. ARP difunde la dirección IP de destino del datagrama que tiene que enviar por todos los nodos de la red. El procedimiento de enviar un paquete a todos los nodos de la red se conoce con el nombre de inundación o broadcast.
2. Cuando el nodo de destino recibe la petición, envía un paquete ARP con la dirección MAC al nodo difusor.
3. De esta manera el nodo emisor conoce la dirección física del nodo destino.

El protocolo **RARP** (*Reverse ARP*) es el protocolo inverso del ARP. Es decir, dada una dirección física se puede obtener la dirección lógica de un host.

109.1.3 DIRECCIONAMIENTO EN REDES TCP/IP.

Las direcciones IPv4 están formadas por 32 bits agrupados en 4 grupos de 8 bits. Se representan con

4 números decimales separados por puntos. El valor que puede tomar una dirección IP oscila entre 0.0.0.0 y 255.255.255.255. Ej: 68.44.26.156.

109.1.3.1- Direcciones IP públicas y privadas

Las direcciones públicas son direcciones IP asignadas por los proveedores de Internet (ISP). Permiten el direccionamiento de paquetes IP para toda la red de Internet. Ej: 176.83.109.98.

Las direcciones privadas son direcciones que sólo se pueden utilizar dentro de la red local. No son reconocidas fuera de la LAN. Ej.: 192.168.1.1.

109.1.3.2.- Clases de direcciones IP

Las direcciones IP se utilizan para interconectar los diferentes nodos que forman la red. Cada dirección IP codifica una red y un host, es decir, un identificador del dispositivo y un identificador de la red donde se encuentra este dispositivo.

Los bits de mayor peso identifican la red y los bits de menos peso identifican el host o dispositivo de red.

109.1.3.2.1.- Direcciones de clase A

- Se dedican 8 bits para identificar la red y 24 para identificar los hosts. El bit de mayor peso vale 0.
- Podemos tener 126 subredes diferentes y 16.777.214 hosts en cada una de ellas.
- Se omiten la red 0.0.0.0 (dirección por defecto) y la red 127.0.0.0 (reservada para localhost).
- Es un sistema utilizado en redes muy grandes.
- Rango de direcciones de la clase A: 1.0.0.0-127.255.255.255.
- Rango de direcciones privadas de clase A: 10.0.0.0-10.255.255.255

Representación de direcciones de clase A

Bits de red (7+1) Bits de host (24)

0rrrrrrr	hhhhhhh.hhhhhh.hhhhhh
----------	-----------------------

109.1.3.2.2. Direcciones de clase B

- Se dedican 16 bits para identificar la red y 16 para identificar los hosts. Los dos bits de más peso valen 10.
- Podemos tener 16.384 subredes diferentes y 65.534 hosts en cada una de ellas.
- Rango de direcciones de la clase B: 128.0.0.0-191.255.255.255.
- Rango de direcciones privadas de clase B: 172.16.0.0-172.31.255.255.

Representación de direcciones de clase B

Bits de red (14+2) Bits de host (16)

10rrrrrr.rrrrrrr	hhhhhhh.hhhhhh
------------------	----------------

109.1.3.2.3.- Direcciones de clase C

- Se dedican 24 bits para identificar la red y 8 para identificar los hosts. Los tres bits de más peso valen 110.
- Podemos tener 2.097.152 subredes diferentes y 254 hosts en cada una de ellas.
- Rango de direcciones de la clase C: 192.0.0.0-223.255.255.255.
- Rango de direcciones privadas de clase C: 192.168.0.0-192.168.255.255.

Representación de direcciones de clase C

Bits de red (21+3) Bits de host (8)
110rrrrr.rrrrrrr.rrrrrrr hhhhhh

109.1.3.2.4. Direcciones de clase D

- Los cuatro bits de más peso valen 1110.
- Los 28 bits restantes se utilizan para indicar una dirección multidifusión.
- Rango de direcciones de la clase D: 224.0.0.0-239.255.255.255.

Representación de direcciones de clase D

Dirección de multidifusión

1110xxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

109.1.3.2.5. Direcciones de clase E

- Los cuatro bits de más peso valen 1111.
- Reservadas por el IETF para investigación.
- Rango de direcciones de la clase E: 240.0.0.0-255.255.255.255.

Representación de direcciones de clase E

Dirección reservada

1111xxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

109.1.3.3.- Máscaras de red

Una máscara de red es una secuencia de 32 bits que se utiliza para distinguir qué parte codifica la subred y qué parte el host.

También se puede utilizar el término prefijo que es un número decimal que indica el número de bits de subred.

Los bits de la subred se ponen a 1 y los del host a 0.

Máscaras de red por defecto

Clase	Prefijo	Máscara decimal	Máscara en binario
A	8	255.0.0.0	11111111.00000000.00000000.00000000
B	16	255.255.0.0	11111111.11111111.00000000.00000000
C	24	255.255.255.0	11111111.11111111.11111111.00000000

Podemos utilizar cualquier combinación de unos y 0 siempre que sean cadenas continuas. Ej:
255255255240 => 11111111.11111111.11111111.11110000.

109.1.3.4.- Subredes

El direccionamiento IP por clases desaprovecha un gran número de hosts en las direcciones con muchos bits de hosts como en las clases A y B y tiene un elevado número de subredes en las direcciones de clase C. Actualmente, en vez de utilizar el direccionamiento por clases, se utiliza la técnica del **subnetting** o de división de redes.

La técnica de subnetting consiste en seleccionar el número de bits de host que sean necesarios según el número de hosts disponibles en la red que se quiera crear. Para aplicar esta técnica es necesario modificar el prefijo o máscara de red por defecto de cada una de las clases.

Problema 1: Calcula el número de subredes y el número de hosts que podemos obtener utilizando el prefijo 27 en una dirección de clase C.

1. Calcular el número de bits de host y de subred:

$$\text{bits subred} = \text{prefijo} - \text{prefijo por defecto} = 27-24 = 3 \text{ bits de subred}$$

$$\text{bits host} = \# \text{ bits de una dirección IP} - \text{prefijo} = 32-27 = 5 \text{ bits de host}$$

2. Calcular el número de hosts y de subred:

$$\text{Número de subredes} = 2^3 = 8$$

$$\text{Número de hosts} = 2^5 - 2 = 30$$

El número total de subredes de una dirección de clase C con prefijo 27 es de 8 subredes. El número de hosts de cada subred es de 30 hosts.

Problema 2: Dado el prefijo 27 calcula la máscara de red.

El prefijo indica el número de bits de red

Los bits de red de la máscara se ponen a 1 y los de host a 0

11111111.11111111.11111111.11100000

Convertimos a decimal:

255.255.255.224

109.1.3.4.1. Cálculo de una subred y la dirección broadcast

Para saber si un conjunto de direcciones IP forma parte de una misma red es necesario conocer la máscara de red y realizar los siguientes cálculos:

1. Convertir las direcciones IP y la máscara en binario.
2. Aplicar una AND lógica entre cada una de las direcciones y la máscara de subred.
3. Comprobar el resultado de las AND lógicas. Si coinciden, pertenecen a la misma red. Este resultado es la dirección de subred.

Si queremos calcular la dirección de broadcast de una determinada subred tendremos que cambiar

los ceros de la parte del host de la dirección de subred por unos.

Problema 1: Dadas las direcciones 192.168.1.1, 192.168.1.2 y 192.168.1.18 y la máscara 255.255.255.240 indica si las direcciones forman parte de la misma red.

1. Conversión de las direcciones IP y la máscara en binario:

IP1) 11000000.10101000.00000001.00000001

IP2) 11000000.10101000.00000001.00000010

IP3) 11000000.10101000.00000001.00010010

M) 11111111.11111111.11111111.11110000

2. Cálculo de subredes:

IP1 & M)

11000000.10101000.00000001.00000001

& 11111111.11111111.11111111.11110000

11000000.10101000.00000001.00000000

IP2 & M)

11000000.10101000.00000001.00000010

& 11111111.11111111.11111111.11110000

11000000.10101000.00000001.00000000

IP3 & M)

11000000.10101000.00000010.00010010

& 11111111.11111111.11111111.11110000

11000000.10101000.00000010.00010000

Las direcciones IP1 y IP2 pertenecen a la misma red:

11000000.10101000.00000001.00000000 (192.168.1.0/28).

La dirección IP3 pertenece a otra red:

11000000.10101000.00000010.00010000 (192.168.1.16/28).

Problema 2 :Dada la dirección 172.16.28.42/27 calcula la dirección de subred y la dirección de broadcast

1. Paso a binario de la IP, cálculo del prefijo y obtención de la subred:

IP= 10101100.00010000.00011100.00101010

M= 11111111.11111111.11111111.11100000

Subred= 10101100.00010000.00011100.00100000

2. Cálculo de la dirección de broadcast:

Subred= 10101100.00010000.00011100.001**00000**

Broadcast= 10101100.00010000.00011100.001**11111**

(en **negrita** se marcan los bits de host para el prefijo 27)

La dirección de broadcast para la subred 172.16.28.32/27 es la dirección 172.16.28.63

109.1.4.- IPv6

Debido al crecimiento que ha experimentado Internet en los últimos años, los organismos de estándares han desarrollado la **IP versión 6** para hacer frente a la demanda de direcciones IP públicas.

Las direcciones IPv4 tienen 32 bits que corresponden a un total de 4.294.967.296 que están a punto de agotarse. Las direcciones IPv6 tienen 128 bits que corresponden aproximadamente a $3,403 * 10^{38}$ direcciones. Con esta gran cantidad de direcciones se pretende solucionar la escasez de direcciones públicas durante los próximos años.

Las direcciones IPv6 se representan como una secuencia de X:X:X:X:X:X:X:X donde cada X equivale a cuatro dígitos hexadecimales. Ej: FF01:0000:0000:0A00:12DF:0000:0144:0001.

Actualmente conviven las dos versiones IP, aunque la mayoría de transferencias de Internet se siguen realizando a través de IPv4. Para garantizar la posibilidad de utilizar ambos protocolos, los sistemas operativos actuales implementan las dos pilas de protocolos, una para IPv4 y otra para IPv6. La utilización de túneles para usar IPv6 sobre IPv4 sirve para que los nodos IPv6 puedan utilizar la red de IPv4. En el futuro próximo, cuando los sistemas IPv6 sean mayoría, los nodos IPv4 podrán utilizar túneles IPv4 sobre IPv6 de esta forma, se conseguirá la migración de IPv4 a IPv6.

109.1.5. PUERTOS DE SERVICIOS TCP Y UDP

Los puntos de acceso a los servicios de la capa de transporte se llaman **sockets**. Cada servicio de red (HTTP, POP, etc.) Tiene asociado un puerto. Ej: el servicio HTTP está asociado al puerto 80.

En la siguiente tabla se muestra la relación del número de puerto y los servicios de red más utilizados:

Aplicación	Asignación de puertos	
	Puerto asociado al servicio	
FTP-datos	20	
FTP	21	
SSH	22	
Telnet	23	
SMTP	25	
DNS	53	
HTTP	80	
POP3	110	
NNTP	119	
NetBIOS	139	
IMAP	143	
SNMP	161	
HTTPS	443	

SMTSP (SMTP seguro) 465

IMAP seguro 993

POP3 seguro 995

En los sistemas Linux la asignación de puertos se almacena en el fichero /etc/services. A continuación mostramos parte de este fichero:

```
# Port Assignments:  
#  
# Keyword Decimal Description References  
# -----  
# 0/tcp Reserved  
# 0/udp Reserved  
# Jon Postel <postel@isi.edu>  
#spr-itunes 0/tcp # Shirt Pocket netTunes  
#spl-itunes 0/tcp # Shirt Pocket launchTunes  
# David Nanian <dnanian@shirt-pocket.com> 28  
September 2007  
tcpmux 1/tcp # TCP Port Service Multiplexer  
tcpmux 1/udp # TCP Port Service Multiplexer  
# Mark Lottor <MKL@nisc.sri.com>  
compressnet 2/tcp # Management Utility  
compressnet 2/udp # Management Utility  
compressnet 3/tcp # Compression Process  
compressnet 3/udp # Compression Process  
# Bernie Volz <volz@cisco.com>  
# 4/tcp Unassigned  
# 4/udp Unassigned  
rje 5/tcp # Remote Job Entry  
rje 5/udp # Remote Job Entry  
# Jon Postel <postel@isi.edu>  
# 6/tcp Unassigned  
# 6/udp Unassigned  
echo 7/tcp # Echo  
echo 7/udp # Echo  
# Jon Postel <postel@isi.edu>  
# 8/tcp Unassigned  
# 8/udp Unassigned  
discard 9/tcp # Discard  
discard 9/udp # Discard  
# Jon Postel <postel@isi.edu>  
discard 9/sctp # Discard  
# IETF TSVWG  
# Randall Stewart <rrs@cisco.com>  
# [RFC4960]  
discard 9/dccp # Discard SC:DISC  
# IETF dccp WG, Eddie Kohler <kohler@cs.ucla.edu>,  
[RFC4340]  
# 10/tcp Unassigned  
# 10/udp Unassigned  
systat 11/tcp # Active Users  
systat 11/udp # Active Users
```

```

# Jon Postel <postel@isi.edu>
# 12/tcp Unassigned
# 12/udp Unassigned
daytime 13/tcp # Daytime (RFC 867)
daytime 13/udp # Daytime (RFC 867)
# Jon Postel <postel@isi.edu>
# 14/tcp Unassigned
# 14/udp Unassigned
# 15/tcp Unassigned [was netstat]
# 15/udp Unassigned
# 16/tcp Unassigned
# 16/udp Unassigned
qotd 17/tcp # Quote of the Day
qotd 17/udp # Quote of the Day
# Jon Postel <postel@isi.edu>
msp 18/tcp # Message Send Protocol
msp 18/udp # Message Send Protocol
# Rina Nethaniel <---none--->
chargen 19/tcp # Character Generator
chargen 19/udp # Character Generator
ftp-data 20/tcp # File Transfer [Default Data]
ftp-data 20/udp # File Transfer [Default Data]
# Jon Postel <postel@isi.edu>
ftp-data 20/sctp # FTP
# IETF TSVWG
# Randall Stewart <rrs@cisco.com>
# [RFC4960]
ftp 21/tcp # File Transfer [Control]
ftp 21/udp # File Transfer [Control]
# Jon Postel <postel@isi.edu>
ftp 21/sctp # FTP
# IETF TSVWG
# Randall Stewart <rrs@cisco.com>
# [RFC4960]
ssh 22/tcp # SSH Remote Login Protocol
ssh 22/udp # SSH Remote Login Protocol
# Tatu Ylonen <ylo@cs.hut.fi>
ssh 22/sctp # SSH
# IETF TSVWG
# Randall Stewart <rrs@cisco.com>
# [RFC4960]
telnet 23/tcp # Telnet
telnet 23/udp # Telnet
# Jon Postel <postel@isi.edu>
# 24/tcp any private mail system
# 24/udp any private mail system
# Rick Adams <rick@UUNET.UU.NET>
smtp 25/tcp # Simple Mail Transfer
smtp 25/udp # Simple Mail Transfer
# Jon Postel <postel@isi.edu>
```

109.1.6. COMANDOS DE CONFIGURACIÓN Y VERIFICACIÓN DE REDES.

En el apartado anterior hemos visto algunos de los servicios de red más utilizados como son el correo electrónico, el acceso remoto, la transferencia de ficheros o la navegación web. En el presente apartado estudiaremos las aplicaciones más utilizadas relacionadas con estos servicios.

109.1.6.1- FTP

El comando **ftp** se utiliza para la transferencia de archivos entre dos máquinas conectadas en red. Para que el comando ftp funcione tiene que estar corriendo el servicio FTP en la máquina del servidor y los puertos 20 y 21 abiertos y en la máquina cliente hay que tener instalado un cliente FTP. Para conectarnos al servicio ejecutaremos el comando ftp seguido del nombre o IP del servidor: **\$ ftp hostname**.

En la tabla siguiente se describen los comandos más relevantes de ftp:

COMANDOS FTP

Comando FTP	Uso
cd	Cambia de directorio en la máquina remota
lcd	Cambia de directorio en la máquina local
bye, close o disconnect	Cierra la sesión FTP
ls	Lista el contenido de un directorio
get <i>fichero</i>	Descarga el fichero especificado de la máquina remota a la máquina local
put <i>fichero</i>	Sube el fichero especificado de la máquina local a la máquina remota
open <i>host</i>	Establece una conexión FTP con el host indicado
pwd	Muestra el directorio remoto actual
delete <i>fichero</i>	Suprime el fichero remoto especificado
quit	Fin de la conexión y cierre del comando FTP
rename	Renombra un fichero remoto
mkdir	Crea un directorio en la máquina remota
rmdir	Suprime un directorio de la máquina remota

Ejemplo de conexión ftp al servidor público ftp.suse.com con el usuario y contraseña 'anonymous' y listado del directorio 'pub':

```
$ ftp anonymous@ftp.suse.com
Trying 195.135.221.132...
Connected to ftp.suse.com.
220 "Welcome to ftp.suse.com"
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||
30182|)
```

```
150 Here comes the directory listing.  
drwxr-xr-x 6 ftp ftp 4096 Jul 01 2011 pub  
226 Directory send OK.  
ftp> cd pub  
250 Directory successfully changed.  
ftp> ls  
229 Entering Extended Passive Mode (|||  
30840|)  
150 Here comes the directory listing.  
-rw-r--r-- 1 ftp ftp 2862 Jul 01 2011  
README.mirror-policy  
-rw-r--r-- 1 ftp ftp 4046 Jan 14 2004  
README.txt  
-rw-r--r-- 1 ftp ftp 1030 Apr 25 2002  
README.upload  
-rw-r--r-- 1 ftp ftp 308655 Jul 01 2011  
find-ls.gz  
drwxr-xr-x 2 ftp ftp 123 Jan 01 00:01  
incoming  
-rw-r--r-- 1 ftp ftp 217757 Jul 01 2011  
ls-laR.gz  
drwxr-xr-x 83 ftp ftp 4096 Nov 30 09:52  
people  
drwxr-xr-x 35 ftp ftp 4096 Feb 19 2008  
projects  
drwxr-xr-x 5 ftp ftp 78 Mar 03 2010 suse  
226 Directory send OK.  
ftp> quit  
221 Goodbye.
```

109.1.6.2.- Telnet

El servicio **telnet** permite el acceso remoto a una máquina. Con el uso de telnet podemos acceder a una shell del host remoto.

Para acceder al equipo remoto se requiere un nombre de usuario y una contraseña. Los datos enviados usando el protocolo telnet viajan en claro, es decir, desde cualquier equipo de la red se pueden capturar los datos que se transmiten, incluso el usuario y la contraseña, por esta razón se desaconseja el uso de telnet. Por motivos de seguridad es preferible utilizar el protocolo SSH (*Secure Shell*) que también permite el acceso a un host remoto cifrando los datos transferidos, incluidos el usuario y la clave de acceso.

Ejemplo de conexión telnet a un router local:

```
$ telnet  
192.168.1.1  
Trying  
192.168.1.1...  
Connected to  
192.168.1.1.  
Escape character
```

```
is '^]'.
```

```
Password:
```

109.1.6.3.- Host

El comando host sirve para recabar información sobre un equipo de la red. Los resultados proporcionados pueden ser: dirección IP del equipo, nombre del equipo o direcciones IP de un dominio.

Ejemplos de uso del comando host con el buscador Google:

```
$ host www.google.com
```

```
www.google.com is an alias for www.l.google.com.  
www.l.google.com has address 173.194.34.48  
www.l.google.com has address 173.194.34.51  
www.l.google.com has address 173.194.34.52  
www.l.google.com has address 173.194.34.50  
www.l.google.com has address 173.194.34.49
```

```
$ host 173.194.34.48
```

```
48.34.194.173.in-addr.arpa domain name pointer par03s03-in-  
f16.1e100.net.
```

```
$ host google.com
```

```
google.com has address 74.125.230.210  
google.com has address 74.125.230.209  
google.com has address 74.125.230.212  
google.com has address 74.125.230.208  
google.com has address 74.125.230.211  
google.com mail is handled by 10 aspmx.l.google.com.  
google.com mail is handled by 40 alt3.aspmx.l.google.com.  
google.com mail is handled by 30 alt2.aspmx.l.google.com.  
google.com mail is handled by 20 alt1.aspmx.l.google.com.  
google.com mail is handled by 50 alt4.aspmx.l.google.com.
```

109.1.6.4.- Ping

La utilización del comando **ping** comprueba el correcto funcionamiento de TCP/IP. Se basa en el envío de paquetes ICMP a los hosts de la red. Sirve para verificar la conexión entre dos dispositivos de la red.

A continuación se muestra el resultado de ejecución de un ping limitando a 4 el número de paquetes a enviar:

```
$ ping -c 4 192.168.1.33
```

```
PING 192.168.1.33 (192.168.1.33) 56(84) bytes of data.  
64 bytes from 192.168.1.33: icmp_req=1 ttl=128 time=12.0 ms  
64 bytes from 192.168.1.33: icmp_req=2 ttl=128 time=0.000  
ms
```

```

64 bytes from 192.168.1.33: icmp_req=3 ttl=128 time=0.000
ms
64 bytes from 192.168.1.33: icmp_req=4 ttl=128 time=0.000
ms

--- 192.168.1.33 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
3008ms
rtt min/avg/max/mdev = 0.000/3.000/12.000/5.196 ms

```

En la siguiente tabla se presentan las opciones más utilizadas para este comando:

Opciones del comando ping

Opción	Función
-b	Permite la emisión de paquetes ICMP a una dirección broadcast
-c <i>número</i>	Con el parámetro <i>número</i> se determina el número de paquetes ICMP a enviar
-d	Habilita la opción de depuración
-f	Fuerza el envío de paquetes tan rápido como sea posible. Puede haber pérdida de datos.
-i <i>segundos</i>	Espera el tiempo en segundos indicado entre el envío de paquetes.
-I <i>iface</i>	Permite especificar la interfaz de red desde la cual se realiza el ping
-n	Muestra sólo direcciones IP, no resuelve nombres del dominio

109.1.6.5.- Traceroute

La utilidad **traceroute** también comprueba la conectividad entre hosts y, además, informa sobre las direcciones de los dispositivos de red que se encuentran entre el host origen y el host de destino. Este comando que utiliza ICMP y UDP permite comprobar en qué punto intermedio se pierde una conexión.

En el siguiente ejemplo se observa el número de salto entre nuestra máquina local y uno de los servidores del buscador Google:

```

# traceroute www.google.com
traceroute to www.google.com (74.125.230.210), 30 hops max, 40 byte packets
using UDP
1 192.168.1.1 (192.168.1.1) 0.000 ms 4.000 ms 0.000 ms
2 65.Red-217-126-16.staticIP.rima-tde.net (217.126.16.65) 48.000 ms 48.000 ms
48.000 ms
3 66.Red-81-46-65.staticIP.rima-tde.net (81.46.65.66) 92.000 ms 88.000 ms
84.000 ms
4 et7-0-0-1-GRTBCNES1.red.telefonica-wholesale.net.103.142.94.in-addr.arpa
(94.142.103.197) 48.000 ms 48.000 ms 48.000 ms
5 Xe11-0-0-0-grtpartv1.red.telefonica-wholesale.net (84.16.13.142) 64.000 ms
68.000 ms 68.000 ms
6 GOOGLE-xe-3-1-0-0-grtpartv1.red.telefonica-wholesale.net (84.16.6.98) 132.000
ms GOOGLE-xe-9-0-0-0-grtpartv1.red.telefonica-wholesale.net (84.16.6.106)
112.000 ms 108.000 ms
7 209.85.251.40 (209.85.251.40) 68.000 ms 72.000 ms 72.000 ms
8 209.85.242.49 (209.85.242.49) 68.000 ms 68.000 ms 68.000 ms
9 par08s09-in-f18.1e100.net (74.125.230.210) 68.000 ms 68.000 ms 68.000 ms

```

109.1.6.6.- Tracepath

La función de **tracepath** es similar a la de traceroute pero no requiere ser superusuario. En algunas distribuciones como Opensuse puede requerir la ejecución con path absoluto. Su homólogo para IPv6 es el comando **tracepath6**.

Ejemplo de ejecución de tracepath hasta que no encuentra respuesta:

```
$ /sbin/tracepath www.google.com
1: linux-1bxj.site 0.000ms pmtu 1500
1: 192.168.1.1 4.000ms asymm 2
1: 192.168.1.1 4.000ms asymm 2
2: 65.Red-217-126-16.staticIP.rima-tde.net 88.000ms
3: 241.Red-80-58-117.staticIP.rima-tde.net 88.000ms
4: et7-0-0-1-GRTBCNTB1.red.telefonica-wholesale.net.103.142.94.in-addr.arpa
96.000ms asymm 7
5: Xe-8-2-0-0-grtpartv1.red.telefonica-wholesale.net 116.000ms asymm 7
6: no reply
```

109.1.6.7.- Dig

La herramienta **dig** realiza consultas a servidores de nombres. En caso de no especificar ningún DNS se toman los servidores del fichero /etc/resolv.conf.

En el siguiente cuadro se muestran los principales parámetros de consulta del comando dig:

Parámetros para las consultas con dig

Parámetro	Función
a	Sólo la dirección
any	Toda la información del dominio
mx	Servidores de correo electrónico
ns	Servidores de nombres
soa segundos	Zona Start of Authority
hinfo iface	Información sobre el anfitrión
txt	Texto de descripción
ptr	Zona reversa del anfitrión
axfr	Lista de todos los anfitriones de la zona

Ejemplo de respuesta del comando dig:

```
$ dig telefonica.es

; <>> DiG 9.7.3 <>> telefonica.es
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id:
52202
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;telefonica.es. IN A
```

```
;; ANSWER SECTION:  
telefonica.es. 300 IN A 194.224.58.10  
  
;; Query time: 68 msec  
;; SERVER: 80.58.61.250#53(80.58.61.250)  
;; WHEN: Mon Jan 2 16:55:15 2012  
;; MSG SIZE rcvd: 47
```

```
$ dig -x 194.224.58.10  
  
; <>> DiG 9.7.3 <>> -x 194.224.58.10  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id:  
27419  
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,  
ADDITIONAL: 0  
  
;; QUESTION SECTION:  
;10.58.224.194.in-addr.arpa. IN PTR  
  
;; ANSWER SECTION:  
10.58.224.194.in-addr.arpa. 172800 IN PTR 194-224-58-  
010.rad.tsai.es.  
  
;; Query time: 68 msec  
;; SERVER: 80.58.61.250#53(80.58.61.250)  
;; WHEN: Mon Jan 2 16:55:51 2012  
;; MSG SIZE rcvd: 84
```

```
$ dig telefonica.es any  
  
; <>> DiG 9.7.3 <>> telefonica.es any  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57477  
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL:  
2  
  
;; QUESTION SECTION:  
;telefonica.es. IN ANY  
  
;; ANSWER SECTION:  
telefonica.es. 300 IN A 194.224.58.10  
telefonica.es. 300 IN NS ns3chos01.telefonica-data.com.  
telefonica.es. 300 IN NS nsjc8hos01.telefonica-data.com.  
telefonica.es. 300 IN SOA nsjc8hos01.telefonica-data.com.  
dnsadmin.tsai.es. 2011121501 86400 7200 2592000 300  
telefonica.es. 300 IN MX 20 mx3.correodeempresas.telefonica.es.
```

```

;; ADDITIONAL SECTION:
ns3chos01.telefonica-data.com. 101 IN A 213.4.194.5
nsjc8hos01.telefonica-data.com. 102 IN A 213.0.43.37

;; Query time: 64 msec
;; SERVER: 80.58.61.250#53(80.58.61.250)
;; WHEN: Mon Jan 2 16:56:49 2012
;; MSG SIZE rcvd: 234

```

109.1.6.8.- Route

La orden **route** se usa para mostrar, borrar o añadir una ruta de red en el host. Es obligatorio tener configurada la puerta de enlace del host para poder enviar paquetes fuera de nuestra red local.

Ejecución de route para obtener la tabla de rutas del host:

```

$ /sbin/route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use
Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
link-local * 255.255.0.0 U 0 0 0 eth0
loopback * 255.0.0.0 U 0 0 0 lo
default 192.168.1.1 0.0.0.0 UG 0 0 0 eth0

```

Ejemplo del uso de route para añadir una ruta por defecto:

```
# route add default gw 192.168.1.1
```

En los apartados de configuración básica de la red se estudiará el uso de este comando en mayor profundidad.

109.1 EXTRAS

109.1 EXTRAS analizadores de paquetes de red en gnu-linux

Los analizadores de paquetes de red, conocidos habitualmente como sniffers, son herramientas que permiten usar un dispositivo de acceso a la red para capturar tramas en el medio para posteriormente analizarlas. Para ello se debe actuar en “modo promiscuo”, esto es, capturar las tramas que viajen por el medio aunque no vayan destinado al host con el analizador.

Existen multitud de sniffers en GNU-linux pero únicamente se desarrollará el uso de Wireshark (uso en GUI) y tcpdump (herramienta de consola)

Wireshark

Este producto se distribuye como software Libre bajo licencia GPL en <http://www.wireshark.org/>. Algunas de sus características son:

- Disponible para UNIX y Windows.
- Captura de paquetes de datos en vivo de una interfaz de red.

- Muestra los paquetes con información de protocolo muy detallado.
- Abrir y guardar datos de paquetes capturados.
- Importar y exportar datos de paquetes desde y hacia muchos otros programas de captura.
- Filtrar paquetes en muchos criterios.
- Búsqueda de paquetes en muchos criterios.
- Colorear muestra de los paquetes en base a filtros.

Instalación en Debian

El procedimiento de instalación es sencillo:,

```
$ sudo apt-get install wireshark <-- con esto se instalara wireshark
$ sudo addgroup --quiet --system wireshark <-- creamos el grupo
wireshark
```

Nota: los atributos --quiet (muestra avisos y errores) y --system (crea un usuario del sistema o grupo)

más info: \$ man addgroup

Se modifica el grupo al que pertenecen los ficheros:

```
$ sudo chgrp wireshark /usr/bin/dumpcap
```

Se modifican los permisos de ficheros:

```
$ sudo chmod 750 /usr/bin/dumpcap
```

Ahora se indica que dumpcap, capture paquetes:

```
$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/bin/dumpcap
```

Agregar el usuario al grupo wireshark:

```
$ sudo usermod -a -G wireshark tu_usuario
```

Reconfigurar Wireshark para que los usuarios regulares puedan capturar paquetes:

```
$ sudo dpkg-reconfigure wireshark-common
```

Funcionamiento básico: captura y filtros

La potencia de esta herramienta (un fork de ethereal) va más allá de lo que se puede analizar en este manual, por ello únicamente se cubrirán los aspectos básicos de captura de paquetes y realización de filtros sencillos.

Para iniciar la captura primero se ha de seleccionar la interfaz de red que recogerá las tramas.

The screenshot shows the Wireshark Network Analyzer interface. At the top, the menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help. Below the menu is a toolbar with icons for opening files, capturing, filtering, and analyzing. A search bar with the placeholder "Expression..." is also present.

The main window has three main sections: "Capture" (containing "Interface List", "Start", "Capture Options", and "Capture Help"), "Files" (containing "Open" and "Sample Captures"), and "Online" (containing "Website", "User's Guide", and "Security").

At the bottom, there are status indicators: "Ready to load or capture" (radio button), "No Packets" (text), and a terminal-like interface showing "operador@instructor:~" and "Añadir/Quitar software". The profile is set to "Default".

A secondary window titled "Wireshark: Capture Interfaces" is displayed in the foreground. It lists network interfaces with their descriptions, IP addresses, and packet counts. The "eth0" interface is selected (indicated by a checked checkbox). Other listed interfaces include "bluetooth0", "nflog", "nfqueue", "usbmon1", "usbmon2", "any", and "lo".

Device	Description	IP	Packets	Packets/s
<input checked="" type="checkbox"/> eth0		172.16.1.16	6	0
<input type="checkbox"/> bluetooth0	Bluetooth adapter number 0	none	62	0
<input type="checkbox"/> nflog	Linux netfilter log (NFLOG) interface	none	0	0
<input type="checkbox"/> nfqueue	Linux netfilter queue (NFQUEUE) interface	none	0	0
<input type="checkbox"/> usbmon1	USB bus number 1	none	3	0
<input type="checkbox"/> usbmon2	USB bus number 2	none	8	0
<input type="checkbox"/> any	Pseudo-device that captures on all interfaces	none	6	0
<input type="checkbox"/> lo		127.0.0.1	0	0

Buttons at the bottom of the "Capture Interfaces" dialog are "Ayuda" (Help), "Start", "Stop", "Options", and "Cerrar" (Close).

Una vez que esta interfaz se ha seleccionado se puede pulsar start para capturar los paquetes. Esto solo es necesario en la primera ocasión en la que se selecciona la interfaz, posteriormente se puede acceder a capture > start. Durante el proceso de captura se verá información de las tramas que se capturan. Todo esto es modificable para optimizar el rendimiento.

Wireshark 1.8.10 (SVN Rev Unknown from unknown)

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Guardar

No.	Time	Source	Destination	Protocol	Length	Info
62	47.976402800	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8004	
63	49.975394000	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8006	
64	49.975412000	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8004	
65	51.967242000	40:00:40:11:8d:46	149 Ethernet II	45:00:00:95:00:00	0xac10	
66	51.967253000	172.16.1.2	255.255.255.255	DB-LSP-I	163	Dropbox LAN sync Discovery Protocol
67	51.967496000	40:00:40:11:df:36			45:00:00:95:00:00	0xac10 149 Ethernet II
68	51.967501000	172.16.1.2	172.16.1.255	DB-LSP-I	163	Dropbox LAN sync Discovery Protocol
69	51.974380000	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8006	
70	51.974388000	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8004	
71	53.973535000	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8006	
72	53.973553000	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8004	
73	55.972605000	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8006	
74	55.972647000	Comtrend_94:35:15	Spanning-tree-(for-br:STP	60	Conf. Root = 32768/0:00:1a:2b:19:5e:7e Cost = 0 Port = 0x8004	

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

IEEE 802.3 Ethernet
Logical-Link Control
Spanning Tree Protocol

0000 01 80 c2 00 00 00 00 00 1d 20 94 35 15 00 26 42 425..&BB
0010 03 00 00 00 00 80 00 00 1a 2b 19 5e 7e 00 00+.^~...
0020 00 00 80 00 00 1a 2b 19 5e 7e 80 06 00 00 14 00+. ~.....
0030 02 00 00 00 a5 a5 a5 a5 a5 a5 a5 a5 a5

eth0: <live capture in progress> File: Capturing from eth0 ... Profile: Default

Para detener la captura pulsar sobre



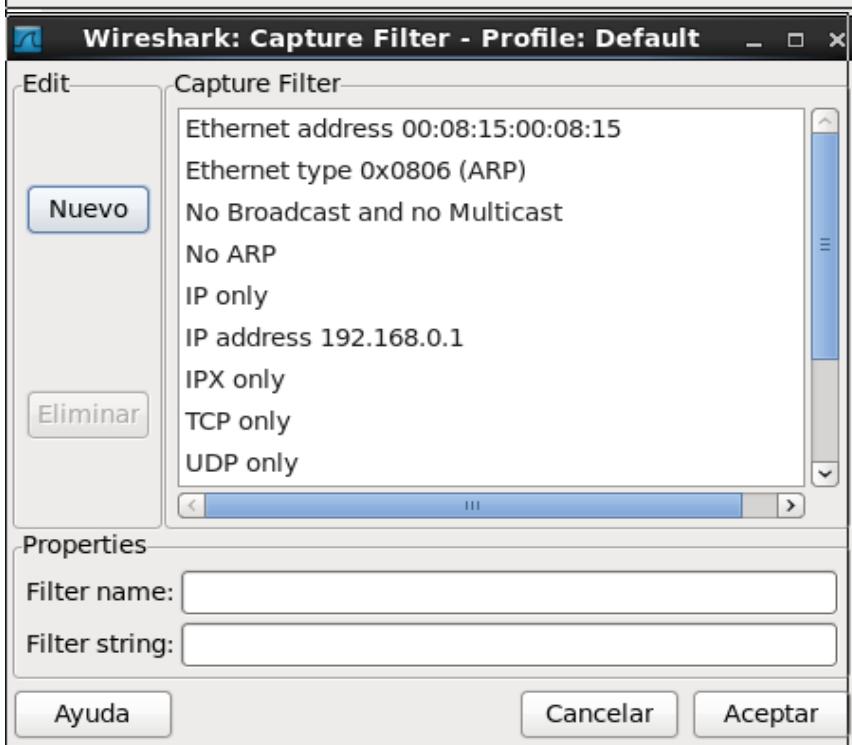
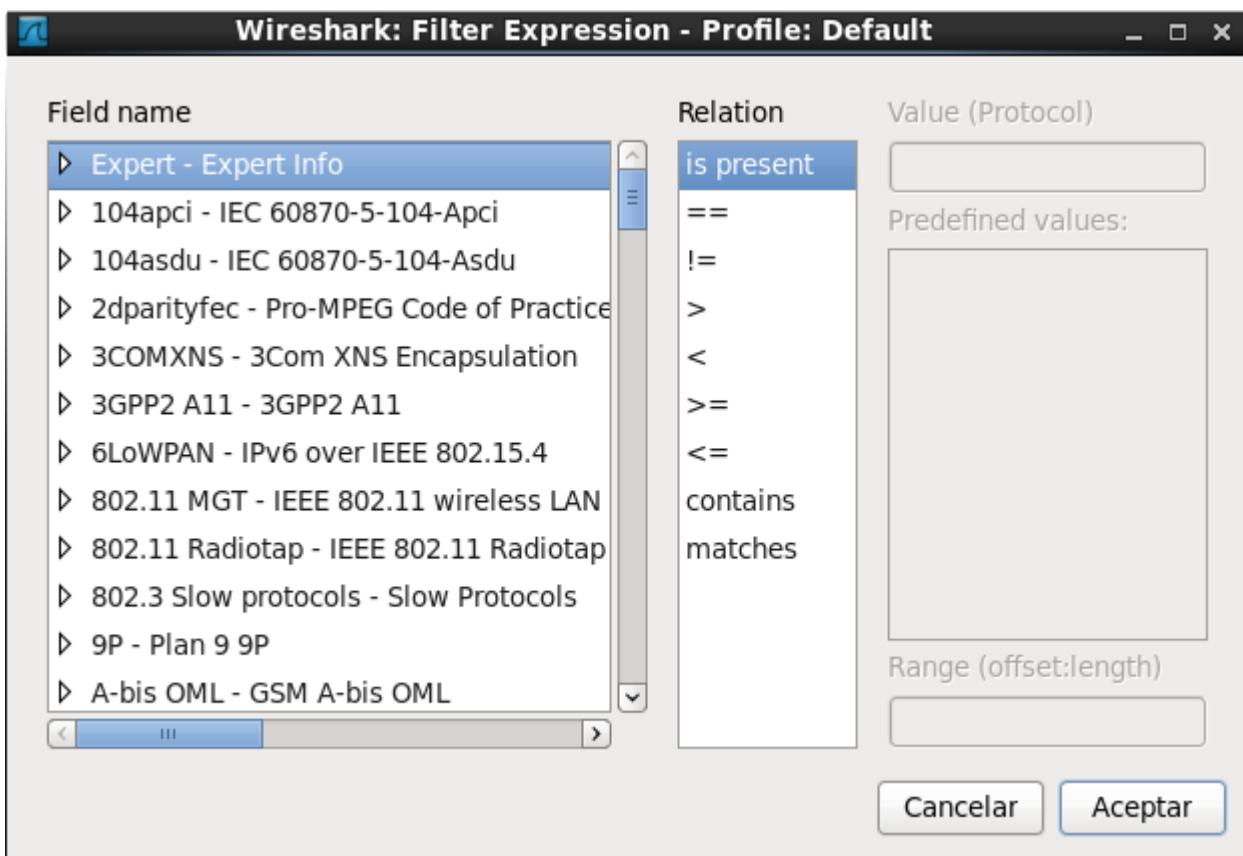
Wireshark hace uso de libpcap para la definición de filtros. Su sintaxis consta de una serie de expresiones conectadas por conjugaciones (and/or) con la opción de ser negada por el operador not:

[not]Expresion[and|or[not]expresión ...]

Por ejemplo, la siguiente expresión define un filtro para la captura de paquetes desde/hacia los host con dirección IP 172.17.250.1 y 172.17.1.81:

ip.addr==172.17.250.1 and ip.addr==172.17.1.81

En el sitio <http://wiki.wireshark.org/CaptureFilters> se puede obtener una serie de filtros que son usualmente aplicados por los administradores de red. Al diálogo de filtros de captura se accede desde la ventana de opciones de captura o desde el menú de Capturas para establecer un filtro de captación de mensajes. Esto evitará la captura de algunos mensajes: antes solo los excluimos de la presentación en pantalla mediante el filtrado de presentación. Si recorremos cada uno de los filtros predefinidos, veremos la expresión que los implemente en " Filter String ". Esta lista puede aumentarse definiendo nuestras propias expresiones y asignando un nombre para ellas. Una expresión debe evaluarse a "true".



TCPDUMP

Tcpdump es un a ojo sniffer de red y suele ir “de serie” en las distros de GNU-linux. Es una

herramienta de línea de comandos y por definición es menos intuitiva que wireshark desde el entorno GUI. Su instalación en debian:

```
root@instructor:~# apt-get install tcpdump
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  tcpdump
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 0 B/421 kB de archivos.
Se utilizarán 1.053 kB de espacio de disco adicional después de esta operación.
Seleccionando el paquete tcpdump previamente no seleccionado.
(Leyendo la base de datos ... 93140 ficheros o directorios instalados actualmente.)
Desempaquetando tcpdump (de .../tcpdump_4.3.0-1_amd64.deb) ...
Procesando disparadores para man-db ...
Configurando tcpdump (4.3.0-1) ...
```

Su ejecución más simple es:

```
root@instructor:~# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
^C22:17:07.790540 IP instructor.local.ssh > atlante.local.55479: Flags [P.], seq
1885455209:1885455321, ack 642264222, win 782, options [nop,nop,TS val 2377040 ecr 825279],
length 112

1 packet captured
24 packets received by filter
0 packets dropped by kernel
```

Habitualmente se usan operadores para que la información sea más precisa.

Algunos ejemplos rápidos de tcpdump:

```
root@instructor:~# tcpdump -i eth0 -c 10 -s 500
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 500 bytes
22:26:41.837560 IP instructor.local.ssh > atlante.local.33819: Flags [P.], seq
168106445:168106557, ack 1360227576, win 782, options [nop,nop,TS val 4294926320 ecr
968873], length 112
(...)
22:26:41.868316 IP 172.16.1.1.domain > instructor.local.51978: 25765 NXDomain 0/0/0 (41)
10 packets captured
60 packets received by filter
0 packets dropped by kernel
```

Mostrará por pantalla los primeros 10 paquetes (-c 100) que capture la interfaz eth0 (-i eth0) con un tamaño maximo de paquete de 500 bytes (-s 500)

```
root@instructor:~# tcpdump -qec 1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
22:28:43.119616 00:0c:29:f4:80:ed (oui Unknown) > 90:e6:ba:19:a4:ab (oui Unknown), IPv4,
length 178: instructor.local.ssh > atlante.local.33819: tcp 112
1 packet captured
23 packets received by filter
0 packets dropped by kernel
```

Existen varias formas de mostrar la información: -q (quiet o silencioso (poca información)) y -v y -vv que van de menos a mas información. La opción -e muestra las direcciones mac origen/destino y -c 1 captura solo el primer paquete. Las diferencias con los operadores -v o -vv son notables.

```
root@instructor:~# tcpdump -vvec 3
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
22:31:46.426936 00:0c:29:f4:80:ed (oui Unknown) > 90:e6:ba:19:a4:ab (oui Unknown), ethertype
IPv4 (0x0800), length 114: (tos 0x10, ttl 64, id 16554, offset 0, flags [DF], proto TCP (6), length
100)
    instructor.local.ssh > atlante.local.33819: Flags [P.], cksum 0x5a87 (incorrect -> 0x1633), seq
168112909:168112957, ack 1360228984, win 782, options [nop,nop,TS val 35171 ecr 1045060],
length 48
22:31:46.427136 90:e6:ba:19:a4:ab (oui Unknown) > 00:0c:29:f4:80:ed (oui Unknown), ethertype
IPv4 (0x0800), length 66: (tos 0x0, ttl 64, id 54718, offset 0, flags [DF], proto TCP (6), length 52)
    atlante.local.33819 > instructor.local.ssh: Flags [.], cksum 0x1a89 (correct), seq 1, ack 48, win
330, options [nop,nop,TS val 1045063 ecr 35171], length 0
22:31:46.427252 00:0c:29:f4:80:ed (oui Unknown) > 90:e6:ba:19:a4:ab (oui Unknown), ethertype
IPv4 (0x0800), length 178: (tos 0x10, ttl 64, id 16555, offset 0, flags [DF], proto TCP (6), length
164)
    instructor.local.ssh > atlante.local.33819: Flags [P.], cksum 0x5ac7 (incorrect -> 0xc27d), seq
48:160, ack 1, win 782, options [nop,nop,TS val 35172 ecr 1045063], length 112
3 packets captured
34 packets received by filter
1 packet dropped by kernel
```

Es posible capturar y filtrar tráfico basado en:

- Direcciones
- Protocolos
- Puertos
- Características de paquetes
- Combinación de todos estos

Algunos ejemplos de filtros que son fácilmente comprensibles pensando en el destino y origen (dst,), protocolos, puertos, ... Se han obviado las salidas.

```
root@instructor:~# tcpdump dst 172.16.1.1
```

```
root@instructor:~# tcpdump ether src host 00:50:22:9a:d3:66
```

```
root@instructor:~# tcpdump udp
```

```
root@instructor:~# tcpdump port 22
```

```
root@instructor:~# tcpdump src 172.16.1.16 and port 22
```

```
root@instructor:~# tcpdump -vv not "(src 172.16.1.16 and port 22)"
```

La lectura del manual de tcpdump es imprescindible para capturas muy concretas y complejas. Una magnífica herramienta.

Por último, destacar otras herramientas muy utilizados y simples para “interactuar con la red”: dsniff, nmap, snort, inSSider, p0f, nemesis, ...

109.1 EXTRAS algo más sobre IPv6

El direccionamiento es el lugar donde se ven la mayoría de las diferencias entre IPv4 e IPv6, pero los cambios están en su mayoría en como son implementadas y usadas estas direcciones. El modelo general utilizado para la asignación de direcciones IP en IPv6 es más o menos el mismo que en IPv4, algunos aspectos no han cambiado en absoluto, mientras que otros han cambiado muy poco.

Aspectos sin cambios en el direccionamiento en IPv6.

Algunas de las características generales del modelo de direccionamiento IPv6 que son básicamente las mismas que en IPv4:

- **Funciones básicas de direccionamiento:** Las dos funciones principales del direccionamiento IPv6 son aun la identificación de la interfaz de red y el enrutamiento. Enrutamiento que es facilitado a través de la estructura de las direcciones en la red interna.
- **Direccionamiento de capa de red:** las direcciones IPv6 siguen estando relacionadas con la capa de red en redes TCP/IP, y son distintas de las direcciones de la capa de enlace de datos (también llamada a veces física).
- **Número de direcciones IP por dispositivo:** Las direcciones se siguen concediendo a las interfaces de red, por lo que un host regular, como una PC, por lo general tienen una dirección (unicast), y los routers tienen más de una, para cada una de las redes físicas a las que se conectan.
- **Interpretación de direcciones y representación de prefijo:** las direcciones IPv6 son similares a las direcciones IPv4 "classless" en que se interpretan en una parte que es el identificador de red y una parte el identificador de host, pero la delimitación no está codificada en la misma dirección. Un número de longitud de prefijo, similar a la notación CIDR, se utiliza para indicar la longitud del ID de red (longitud de prefijo).
- **Direcciones privadas y públicas:** Los dos tipos de direcciones existen en IPv6, a pesar de que se definen y utilizan un poco diferente.

Tipos de direcciones IPv6.

Un cambio importante en el modelo de direccionamiento en IPv6 son los tipos de direcciones soportados. IPv4 soporta tres tipos de direcciones: unicast, multicast y broadcast. De éstos, la gran mayoría del tráfico real es unicast. El soporte para direcciones IP multicast no se desplegó ampliamente hasta muchos años después de que la Internet fuera creada, y continúa siendo obstaculizado por diversas cuestiones. El uso del broadcast en IP tuvo que ser severamente restringido por razones de rendimiento (no queremos que ningún dispositivo sea capaz de transmitir a través de la Internet entera!)

IPv6 también es compatible con tres tipos de direcciones, pero con algunos cambios:

- **Direcciones unicast:** Estas son las direcciones de nivel unicast como en IPv4, una por cada interfaz de host.
- **Las direcciones de multidifusión:** Estas son direcciones que representan a varios grupos de dispositivos IP: un mensaje enviado a una dirección multicast llega a todos los dispositivos del grupo. IPv6 incluye muchas mejores características de multidifusión y muchas más direcciones multicast que IPv4. Dado que multicast en IPv4 se vio afectada en gran parte debido a la falta de soporte de la función en muchos dispositivos de hardware, el soporte para multidifusión es una parte necesaria, no opcional, de IPv6.
- **Direcciones anycast:** El direccionamiento anycast se utiliza cuando un mensaje debe ser enviado a cualquier miembro de un grupo, pero no es necesario que llegue a todos. Por lo general, al miembro del grupo que sea el más fácil de alcanzar se le enviará el mensaje. Un ejemplo común de cómo se puede emplear el direccionamiento anycast es en el intercambio de carga entre un grupo de routers en una organización.

Implicaciones de los cambios a los tipos de direcciones en IPv6.

El direccionamiento de difusión como un método distinto de direccionamiento se ha eliminado en IPv6. La funcionalidad de difusión se realiza utilizando direccionamiento multicast a los grupos de dispositivos. Un grupo multicast al que pertenecen todos los nodos pueden ser utilizados para la difusión en la red, por ejemplo.

Una implicación importante de la creación de anycast es la eliminación del requisito de estricta singularidad de las direcciones IP. Anycast se logra asignando la misma dirección IP a más de un dispositivo. A los dispositivos también se les debe especificar que están compartiendo una dirección anycast, pero las direcciones en sí son estructuralmente lo mismo que las direcciones unicast.

Notación hexadecimal de las direcciones IPv6.

Para hacer más cortas las direcciones, se tomó la decisión en IPv6 de cambiar el método principal de expresión de direcciones y utilizar notación hexadecimal en lugar de decimal. La ventaja de esto es que requiere menos caracteres para representar una dirección, y que la conversión de hexadecimal a binario y viceversa, es mucho más fácil que las conversiones entre decimales y binarios. La desventaja es que muchas personas, incluso personas con ciertos conocimientos sobre computadoras, encuentran los números hexadecimales difíciles de comprender y trabajar, sobre todo porque la noción de los 16 valores en cada dígito es un poco extraña.

La notación hexadecimal utilizada para las direcciones IPv6 es similar a el mismo método utilizado para las direcciones MAC IEEE 802, en tecnologías como Ethernet. Allí, los 48 bits están representados por seis octetos, cada octeto es un número hexadecimal de 0 a FF, separados por un guión o dos puntos, así:

0A-A7-94-07-CB-D0

Ya que las direcciones IPv6 son más grandes, en lugar de esto se agrupan en ocho palabras de 16 bits, separados por dos puntos, para crear lo que se llama a veces notación hexadecimal con dos puntos, una vez más se muestra en la Figura 95. Por lo tanto, la dirección IPv6 dada en el ejemplo anterior se puede expresar como:

805B:2D9D:DC28:0000:0000:FC57:D4C8:1fff

Para mantener el tamaño reducido, se pueden suprimir los ceros en la notación, por lo que de inmediato puede reducirse esto a:

805B:2D9D:DC28:0:0:FC57:D4C8:1fff

Además es posible realizar un proceso de reducción de las direcciones eliminando todos los valores 0 contiguos de la dirección, esto es:

805B:2D9D:DC28::FC57:D4C8:1fff

el ejemplo más evidente de esta reducción es la dirección de loopback ipv6 que son todo 0 exceptuando la activación de un único bit:

::1

El formato EUI-64 modificado IPv6.

El mapeo real de las direcciones de capa de enlace a los ID de interfaces IP depende de la tecnología en particular. Por supuesto es esencial que todos los dispositivos en la misma red utilicen la misma técnica de asignación. Con mucho, el tipo más común de direcciones de capa 2 en redes son las direcciones MAC IEEE 802, utilizadas por Ethernet y otras tecnologías de redes del proyecto IEEE 802. Como usted ya sabe, estas direcciones tienen 48 bits, organizados en dos bloques de 24. Los 24 bits "superiores" se organizan en un bloque llamado identificador único organizacional (OUI), con diferentes valores asignados a las distintas organizaciones, los 24 bits "inferiores" se utilizan entonces para un identificador para cada dispositivo específico.

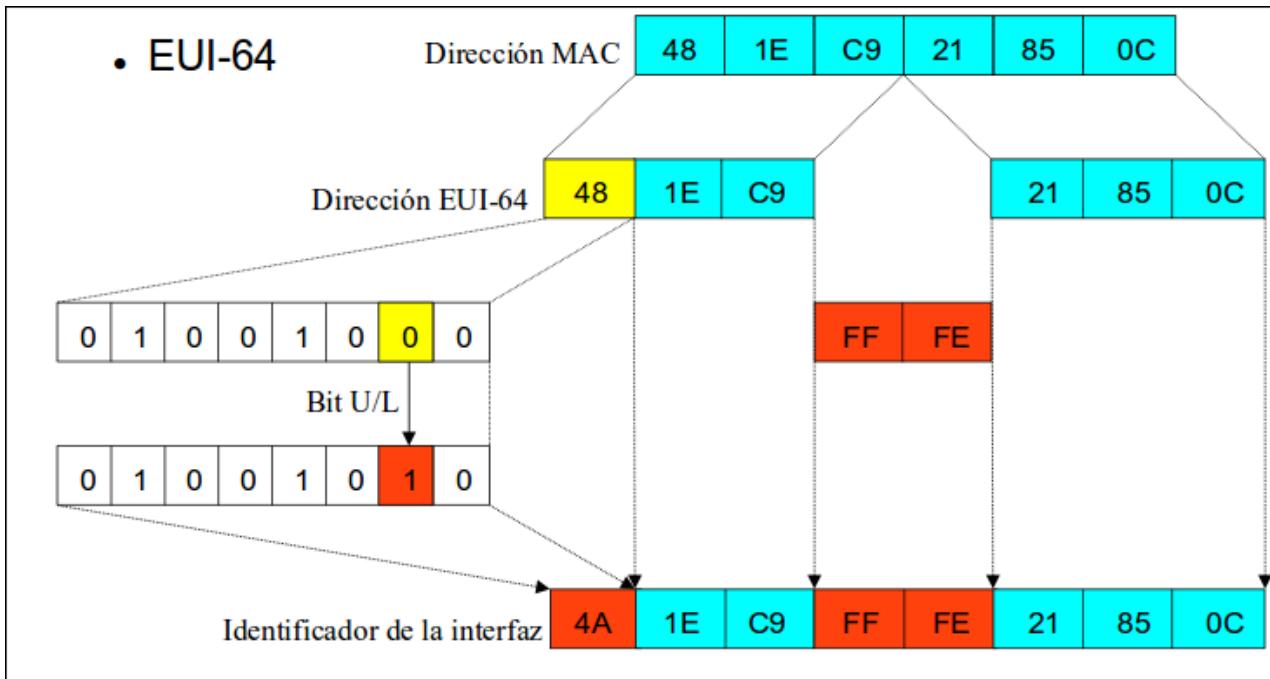
El IEEE ha definido un formato llamado identificador único extendido (Extended Unique Identifier) de 64-bit, abreviado EUI-64. Es similar al formato MAC de 48-bit, excepto que mientras que el OUI se mantiene en 24 bits, el identificador del dispositivo tiene 40 bits en lugar de 24. Esto le da a cada fabricante 65.536 veces más cantidad de direcciones de dispositivos dentro de su OUI.

Una forma de este formato, llamada EUI-64 modificado, ha sido adoptado para los identificadores de interfaz IPv6. Para obtener la identificación de la interfaz EUI-64 modificado de un dispositivo, basta con tomar la dirección EUI-64 y cambiar el bit séptimo desde la izquierda (el bit "universal/local" o "U/L") de cero a uno .

Conversión de direcciones MAC de 48 bits a identificadores IPv6 modificados EUI-64.

Por supuesto, la mayoría de los dispositivos siguen utilizando el viejo formato de 48 bits de direcciones MAC. Estos se pueden convertir en EUI-64 y luego a EUI-64 modificado para crear un identificador de interfaz IPv6. El proceso es el siguiente:

1. Tomamos la porción de 24-bit OUI, los 24 bits mas a la izquierda de la dirección Ethernet, y los ponemos en los correspondientes 24 bits mas a la izquierda de la ID de la interface. Tomamos la parte local de 24 bits (los 24 bits mas a la derecha de la dirección Ethernet) y lo ponemos en los 24 mas a la derecha de la ID de la interfaz.
2. En los restantes 16 bits en el medio del ID de interfaz ponemos el valor "11111111 11111110" ("FFFE" en hexadecimal).
3. La dirección está ahora en la forma EUI-64. Cambiamos el bit "universal/local" (bit 7 desde la izquierda) de un cero a uno. Esto nos da el ID de interfaz EUI-64 modificado.



De esta manera se puede identificar de manera única cada interfaz y este EUI se usará para la asignación de los diferentes tipos de direcciones que se pueden ligar a una interfaz.

PREFIJOS POR TIPO DE DIRECCIÓN

UNICAST: dirección para un único interfaz. Se pueden distinguir varios tipos. Ocupan un octavo de todo el espacio de direcciones IPv6.

Globales: son definidas por un prefijo

001	Global Routing Prefix (proveedor)				Subred ID (site)	Interface ID (host)
	TLA	RES	NLA	SLA		
↔ → 3	↔ 45 bits →	↔ 16 bits →	↔ 64 bits →			

TLA: Top-Level Aggregation Identifier (13 bits)

RES: Reservado para un uso futuro (8 bits)

NLA: Next-Level Aggregation Identifier (24 bits)

SLA: Site-Level Aggregation Identifier (16 bits)

El formato estándar es el que se indica arriba, pero cada organización puede elegir la forma de repartir su espacio, dependerá de su política de registro [RFC2374]

Site-local: equivalentes a las direcciones privadas en IPv4

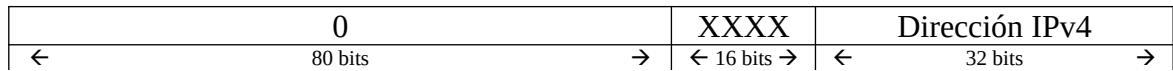
1111 1110 11	0	Subred	Interface ID
FEC0::/10	↔ 38 bits →	↔ 16 bits →	↔ 64 bits →

Link-local: utilizadas en el descubrimiento de los nodos vecinos y la autoconfiguración

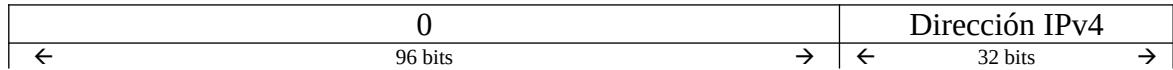
1111 1110 10	0	Interface ID
--------------	---	--------------



IPv4 mapped IPv6: utilizada para representar la dirección de un nodo IPv4 como una dirección IPv6



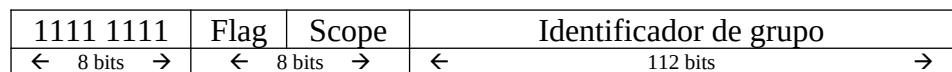
IPv4 compatible IPv6: obsoleta (CISCO y MSoft)



Interfaz ID se representa en el → formato EUI-64 (dirección MAC extendida)

ANYCAST: nuevo tipo de dirección asignada a un grupo de interfaces, típicamente pertenecientes a diferentes nodos, por lo tanto una dirección identifica múltiples interfaces. Un paquete enviado a una dirección anycast es enviado a la interfaz más cercana, según la especificación de los protocolos de routing. No se distinguen de las direcciones unicast globales.

MULTICAST: dirección que identifica un conjunto de interfaces que usualmente pertenecen a distintos nodos



FLAG (tiempo de vida)	
0	Permanente
1	Temporal

ÁMBITO	
1	Interface-local
2	Enlace-local
3	Subred-local
4	Admin-local
5	Site-local
8	Organización-local
E	Global

- Ejemplos:
- | | |
|---------|--------------------------------------|
| FF0::1 | - para el host |
| FF02::1 | - los nodos en el enlace local |
| FF01::2 | - routers dentro del nodo local |
| FF01::2 | - routers dentro del enlace local |
| FF05::2 | - todos los routers en el mismo site |

Resumiendo, en función de los primeros caracteres de la dirección se puede identificar el tipo de dirección IP que se nos presenta. La siguiente tabla es el resumen de los comentado:

Prefijo (hexadecimal)	Uso
00	Direcciones IPv4 y <i>compatibles con IPv4 sobre IPv6</i> . Son direcciones compatibles con IPv4. Un router adecuado tiene que convertir el paquete IPv6 a IPv4. Hay otras direcciones especiales (por ejemplo loopback device) que utilizan este prefijo.
Primera cifra 2 ó 3	(<i>Aggregatable Global Unicast Address</i>) Igual que ahora, también en el caso de IPv6 se puede recibir la asignación de subredes a través de un proveedor. En la actualidad existen los siguientes espacios de direcciones: 2001 :: /16 (<i>production quality address space</i>) y 2002 :: /16 (<i>6to4 address space</i>).
fe80::/10	(<i>link-local</i>) Las direcciones con este prefijo no pueden ser enrutadas y por tanto sólo se puede acceder a ellas en la misma subred.
fec0::/10	(<i>site-local</i>) Estas direcciones pueden ser enrutadas pero solamente dentro de una misma organización. Estas direcciones corresponden a las direcciones “privadas” actuales (por ejemplo 10.x.x.x).
ff	(<i>multicast</i>) Las direcciones IPv6 que comienzan por ff son direcciones multicast.

Deshabilitar IPv6

Si no se usa el direccionamiento IPv6 se puede deshabilitar usando el siguiente procedimiento:

1. Verificar que realmente tenemos el ipv6 soportado por el kernel

```
# dmesg | egrep IPv6
eth2: no IPv6 routers present

0:
#cat /proc/sys/net/ipv6/conf/all/disable_ipv6
0
(debe mostrar 1 si esta deshabilitado).
```

2. Modificar el file: */etc/sysctl.conf*

Añadir al final

```
#Deshabilitar ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

3. Reiniciar la red:

```
# /etc/init.d/network restart // CentOS
#/etc/init.d/networking restart //Debian
o reiniciando el host si fuera posible.
```

109.2 Configuración básica de red.

Peso en el examen de certificación: 4 puntos.

Objetivo: Visualizar, cambiar y verificar los parámetros de configuración en las máquinas cliente.

Conceptos y áreas de conocimiento:

- Configuración manual y automática de los interfaces de red.
- Configuración básica de TCP/IP en sistemas conectados a la red.

Términos y utilidades:

- /etc/hostname
- /etc/hosts
- /etc/resolv.conf
- /etc/nsswitch.conf
- ifconfig
- ifup
- ifdown
- route
- ping

ÍNDICE

- 109.2.1. Ifconfig
- 109.2.2. Ifup y ifdown
- 109.2.3. Route
- 109.2.4. Ping
- 109.2.5. Fichero /etc/hostname
- 109.2.6. Fichero /etc/hosts.
- 109.2.7. Fichero /etc/resolv.conf
- 109.2.8. Fichero /etc/nsswitch.conf

La configuración de la red se realiza mediante la orden *ifconfig* .

Para activar y desactivar una interfaz de red se utilizan las ordenes **ifup** y **ifdown**, respectivamente.

Para mostrar y modificar la tabla de enrutamiento se usa la orden **route** .

109.2.1 Ifconfig

Este comando permite fijar manualmente todos los parámetros de configuración básicos de TCP/IP, es decir, la dirección IP, la máscara y la dirección de broadcast.

Se puede activar la interfaz de red manualmente asociándole la dirección IP junto con la máscara de red y la dirección de broadcast:

Sintaxis básica:

#ifconfig <Interfaz> <dir_IP> netmask <máscara> broadcast <broadcast>

Ej.- ifconfig eth1 138.100.58.45 netmask 255.255.255.0 broadcast 138.100.58.255

Para mostrar la configuración actual:

\$ifconfig -a

Mostrar toda la información acerca de la configuración de TCP/IP de tu equipo, interfaces de red, IP, MAC Address, gateway, DNSs, etc.

```
$ ifconfig
eth0 Link encap:Ethernet direcciónHW 00:1d:92:f1:19:5e
inet dirección:192.168.0.6 Difusión:192.168.0.255
Máscara:255.255.255.0
dirección inet6: fe80::21d:92ff:fe1:195e/64 Alcance:Vínculo
ARRIBA DIFUSIÓN CORRIENDO MULTICAST MTU:1500 Métrica:1
RX packets:4097 errors:0 dropped:0 overruns:0 frame:0
TX packets:4684 errors:0 dropped:0 overruns:0 carrier:0
colisiones:0 txqueuelen:1000
RX bytes:2768916 (2.6 MB) TX bytes:1054771 (1.0 MB)
Interrupción:220 Dirección base: 0x6000
-
lo Link encap:Bucle local
inet dirección:127.0.0.1 Máscara:255.0.0.0
dirección inet6: ::1/128 Alcance:Anfitrión
ARRIBA LOOPBACK CORRIENDO MTU:16436 Métrica:1
RX packets:4564 errors:0 dropped:0 overruns:0 frame:0
TX packets:4564 errors:0 dropped:0 overruns:0 carrier:0
colisiones:0 txqueuelen:0
RX bytes:228200 (222.8 KB) TX bytes:228200 (222.8 KB)
```

Para activar o desactivar un interfaz de red:

```
#ifconfig eth1 up
#ifconfig eth1 down
```

Una misma interfaz de red eth0 puede tener varias IP o alias si definimos los archivos correspondientes.

Por ejemplo si definimos los ficheros:

ifcfg-eth0

ifcfg-eth0:0

La misma tarjeta Ethernet eth0 tendrá dos IP cada uno configurado en su archivo.

A continuación veremos algunos **ejemplos** del uso de **ifconfig**:

Ver la configuración de red de un adaptador de red

\$ifconfig

Invocado sin argumentos mostrará el detalle de todas las interfaces activas.

\$ifconfig eth0

Si como argumento pasamos el nombre de una interfaz, veremos los detalles específicos de una interfaz.

Ver un detalle de todas las interfaces (incluidas las deshabilitadas)

\$ifconfig -a

Deshabilitar la interfaz eth0

```
$ifconfig eth0 down
```

Habilitar la interfaz eth0

```
$ifconfig eth0 up
```

Asignar un dirección IP a la interfaz eth0

```
$ifconfig eth0 192.168.0.2
```

Cambiar la máscara de sub red:

```
$ifconfig eth0 netmask 255.255.255.0
```

Cambiar la dirección de broadcast:

```
$ifconfig eth0 broadcast 192.168.0.255
```

Asignar dirección IP, máscara y broadcast al mismo tiempo:

```
$ifconfig eth0 192.168.0.2 netmask 255.255.255.0 broadcast  
192.168.0.255
```

Cambiar el MTU (unidad máxima de transmisión)

```
$ ifconfig eth0 mtu XX
```

Para Ethernet, el tamaño máximo de un paquete a transmitir por transacción en TCP/IP por defecto es de 1500 bytes.

Poner la interfaz de red en modo promiscuo

```
$ifconfig eth0 promisc
```

Por defecto cuando una tarjeta de red recibe un paquete chequea si es para si mismo y si no lo es lo descarta. En modo promiscuo, la tarjeta no descarta ese paquete y acepta todos los paquetes así no sean para la misma.

El modo promiscuo se utiliza especialmente para capturar y analizar el tráfico de una red.

Poner la interfaz de red en modo normal

```
$ifconfig eth0 -promisc
```

109.2.2 ifup y ifdown

Las interfaces de red configuradas en el archivo

Para ello basta con indicar el nombre de la interfaz después del comando.

Por ejemplo:

```
$ifdown eth0
```

Desactivaría la primera interfaz Ethernet.

El uso de estos comandos es útil cuando queremos “cortar” temporalmente la conexión a red o bien queremos cambiar los parámetros de configuración.

Para ello, modificaremos en el archivo "/etc/network/interfaces" los parámetros pertinentes y luego ejecutaremos sucesivamente los comandos ifdown e ifup indicando el nombre del interfaz de red que hemos modificado.

```
$ifup eth0
```

Para volver a activar la interfaz de red

109.2.3 Route

El comando route muestra la tabla de enrutamiento que reside en el kernel y también se usa para

modificarla. La tabla que especifica cómo se enrutan los paquetes a un host se llama tabla de enrutamiento.

La sintaxis es:

\$route [opciones]

Opciones de configuración de las bases de datos NSS

Parámetro	Descripción
-n	Muestra la tabla de enrutamiento en formato numérico [dirección IP]
-e	Muestra la tabla de enrutamiento en formato hostname
add	Añade una nueva ruta a la tabla de enrutamiento
del	Elimina una ruta de la tabla de enrutamiento
Opciones usadas con add y del	
Parámetro	Descripción
-net	Indica que el objetivo es una red
-host	Indica que el objetivo es un host
gw	Especifica el puerta de enlace del host o red objetivo
netmask	Usado para especificar la máscara de subred del host o red de destino
dev	Especifica el dispositivo o interfaz donde se enviarán los paquetes
reject	Rechaza los paquetes enviados a una ruta o host particular

EJEMPLO:

1. Para mostrar la tabla de enrutamiento:

```
$route -n
```

El comando anterior mostrará:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.0.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	0	0	0	eth0
0.0.0.0	192.168.0.1	0.0.0.0	UG	0	0	0	eth0

En la tabla anterior:

Destination -Indica la dirección IP de la red o host de destino

Gateway -Indica el puerta de enlace desde el cual se alcanza el host o red de destino

Genmask -Indica el destino de la máscara de subred

Flags -Indica el estado actual de ruta

- U - La ruta está activa
- H - El objetivo es un host
- G - Utilizar puerta de enlace

Iface -Indica la interfaz

2. Para añadir ruta estática a una red en la tabla de enrutamiento:

```
$route add -net 192.168.1.0 netmask 255.255.255.0 gw  
192.168.1.1 dev eth0
```

En el comando anterior:

add -Indica que la ruta se añade a la tabla de enrutamiento.

-net -Indica que el destino es una red

192.168.0.1 -Indica la dirección IP de la red de destino

netmask -Indica la máscara de subred de la red de destino.

gw 192.168.1.1 -Indica la puerta de enlace de la red de destino.

dev eth0 -Indica que los paquetes se enrutan a través de la interfaz eth0.

3. Para eliminar una ruta de la tabla de enrutamiento:

```
$route del -net 192.168.1.0 netmask 255.255.255.0 gw  
192.168.1.1 dev eth0
```

El comando anterior eliminará la ruta a 192.168.1.0 de la tabla de enrutamiento.

109.2.4. Ping

El comando *ping* permite verificar si una máquina remota responde, nos dice si un ordenador está actualmente conectado a Internet y la calidad y velocidad de su conexión.

Utiliza el protocolo ICMP (*Internet Control Message Protocol*), un protocolo similar a UDP pero más simple, ya que no posee identificación de puertos, y que se utiliza para mensajes de control y error. Funciona enviando paquetes ICMP ECHO_REQUEST (pings).

Veamos cómo utilizar ping:

- **comprobar si una máquina remota responde:** podemos usar la dirección IP o el nombre del host. Por ejemplo:

```
$ ping pc350  
PING pc350.fransberns.com (192.168.0.5) 56(84) bytes of data.
```

```
64 bytes from pc350.fransberns.com: icmp_seq=1 ttl=64 time=0.792
ms
64 bytes from pc350.fransberns.com: icmp_seq=2 ttl=64 time=3.38 ms
64 bytes from pc350.fransberns.com: icmp_seq=3 ttl=64 time=0.752
ms
...
--- pc350.fransberns.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5013ms
rtt min/avg/max/mdev = 0.752/1.207/3.382/0.973 ms
```

Por defecto, el comando no se detiene y hay que apagarlo con <Ctrl+C>. Por eso es preferible especificar el número de pings que queremos hacer con la opción -c, por ejemplo 5:

```
$ ping -c 5 pc350
```

- **comprobar el sistema DNS:** ping permite comprobar el mapeo de nombres a IPs, y sirve tanto para la red local como para Internet, ya que utiliza /etc/hosts y /etc/resolv.conf

Probamos con localhost y lo encuentra en /etc/hosts:

```
$ ping localhost
PING localhost.localdomain (127.0.0.1) 56(84) bytes of data.
```

Probamos con la propia máquina, pc450, y la encuentra en /etc/hosts:

```
$ ping pc450
PING pc450.fransberns.com (192.168.0.2) 56(84) bytes of data.
```

Si probamos con una máquina de la red no la encuentra: en /etc/hosts no está listada, recurre a los DNS del ISP y allí no la conocen:

```
$ ping pc350
ping: unknown host pc350
```

Debemos añadir las máquinas locales a /etc/hosts o montar un servidor DNS local.

si probamos con telefonica.net consulta a los DNS del ISP y la encuentra:

```
$ ping telefonica.net
PING telefonica.net (213.4.130.95) 56(84) bytes of data.
```

109.2.5 Fichero /etc/hostname

Aquí se encuentra el nombre del ordenador, es decir, sólo el nombre del host sin el nombre de dominio. Hay distintos scripts que leen este archivo durante el arranque del ordenador.

¡No debe contener más que una sola línea con el nombre del ordenador!

109.2.6 Fichero /etc/hosts.

Este archivo tiene una tabla de asignación entre nombres de ordenadores y direcciones IP. En esta tabla deben aparecer todos los ordenadores con los que se quiere establecer una conexión IP cuando no se usa un servidor de nombres. Cada ordenador ocupa una línea en la tabla que contiene el número IP, el nombre completo de la máquina y el nombre (abreviado), por ejemplo tierra. La línea debe comenzar con la dirección IP y las demás indicaciones se separan con espacios o tabuladores. Los comentarios comienzan con #.

Ejemplo /etc/hosts:

```
127.0.0.1 localhost  
192.168.0.20 sol.example.com sol  
192.168.0.0 tierra.example.com tierra
```

109.2.7 Fichero /etc/resolv.conf

Al igual que el archivo /etc/host.conf, este también juega un papel en la resolución de nombres de ordenadores a través de la librería *resolver*.

En este archivo se indica el dominio al que pertenece el ordenador (palabra clave search) y la dirección del servidor de nombres (palabra clave nameserver) al que se debe dirigir. Se puede introducir más nombres de dominio. Al resolver nombres que no estén totalmente cualificados se intentará generar un nombre válido y cualificado añadiendo entradas únicas en search. Se puede dar a conocer otros servidores de nombres añadiendo más líneas que comiencen con nameserver. Se puede introducir comentarios con #.

Ejemplo /etc/resolv.conf

```
# Our domain  
search example.com  
#  
# We use sol (192.168.0.20) as nameserver  
nameserver 192.168.0.20
```

YaST escribe aquí el servidor de nombres especificado.

Algunos servicios, como pppd (wvdial), ipppd (isdn), dhcp (dhpcd y dhclient), pcmcia y hotplug pueden modificar los archivos /etc/resolv.conf mediante el script modify_resolvconf.

Al modificar el archivo /etc/resolv.conf con este script, aquel contendrá un comentario que da información sobre los servicios que se han modificado, el lugar donde se encuentra el archivo original y cómo se puede detener las modificaciones automáticas.

Si /etc/resolv.conf es modificado más veces, se volverá a limpiar este cúmulo de modificaciones cuando se recojan en otro orden; lo cual puede ocurrir con isdn, pcmcia y hotplug.

Si un servicio no ha finalizado “limpiamente”, se puede restaurar el estado original con ayuda del script modify_resolvconf. Al arrancar se probará si un resolv.conf se ha quedado modificado (por ejemplo debido a un cuelgue del sistema); en ese caso se volverá a restaurar el resolv.conf original (sin modificar).

Por medio de modify_resolvconf check, YaST averigua si resolv.conf fue modificado, tras lo cual avisa al usuario de que se han perdido sus cambios tras la restauración. En caso contrario, YaST no utiliza modify_resolvconf, lo que quiere decir que una modificación en el archivo resolv.conf mediante YaST equivale a una modificación manual. Ambas indican una modificación duradera

mientras que las realizadas por los servicios mencionados sólo son pasajeras.

109.2.8 Fichero /etc/nsswitch.conf

El archivo /etc/nsswitch.conf determina en qué orden se solicitan determinadas informaciones. El archivo etc/nsswitch.conf, muestra un ejemplo para nsswitch.conf en el cual las líneas de comentarios comienzan con #. Respecto a la “base de datos” hosts, el ejemplo siguiente indica que se envía una solicitud al servicio DNS después de consultar /etc/hosts (files).

Ejemplo /etc/nsswitch.conf

```
passwd: compat
group: compat

hosts: files dns
networks: files dns

services: db files
protocols: db files

netgroup: files
automount: files nis
```

Las “bases de datos” accesibles vía NSS se recogen en la tabla siguiente. Para el futuro se espera también la disponibilidad de automount, bootparams, netmasks y publickey.

Tabla Bases de datos accesibles a través de /etc/nsswitch.conf

Parámetro	Bases de datos accesibles /etc/nsswitch.conf
aliases	Descripción Alias de correo usado por sendmail, ver la página del manual man 5 aliases .
ethers	Direcciones de ethernet.
group	Usado por getrent para grupos de usuarios, ver la página del manual man 5 group .
hosts	Para nombres de host y direcciones IP, utilizados por funciones como gethostbyname o similares.
netgroup	Lista de hosts y de usuarios válida en la red para administrar los derechos de acceso; ver la página del manual man 5 netgroup .
networks	Nombres y direcciones de redes, lo usa getnetent.
passwd	Contraseñas de usuarios, utilizado por getpwent, ver la página del manual man 5 passwd .

protocols	Protocolos de red, información utilizada por getprotoent, ver la página del manual man 5 protocols.
rpc	Nombres y direcciones del tipo “Remote Procedure Call”; utilizado por getrpcbyname y funciones similares.
services	Servicios de red; lo usa getservent.
shadow	Las contraseñas “Shadow” de los usuarios, usado por getspnam, ver la página del manual man 5 shadow.
Las opciones de configuración de las “bases de datos” NSS se encuentran en tabla siguiente:	
Opciones de configuración de las bases de datos NSS	
Parámetro	Descripción
files	acceso directo a los archivos, por ejemplo a /etc/aliases.
db	acceso a través de una base de datos.

nis	NIS, ver apartado .
nisplus	
dns	Parámetro adicional, solo aplicable para hosts y networks.
compat	Parámetro adicional para passwd, shadow y group.
<i>además</i>	es posible conseguir diferentes resultados en caso de determinados eventos “Lookup”; puede encontrar información adicional en la página del manual man 5 nsswitch.conf .

109.2 EXTRAS

109.2 EXTRAS Configuración manual de la red mediante manipulación de ficheros

DEBIAN

Debian guarda la configuración de la red en el directorio donde se encuentran todas las configuraciones, `/etc`, en concreto en el **archivo interfaces**, que está dentro de la carpeta **network**. La ruta desde raíz sería esta:

/etc/network/interfaces

Para ver el contenido de este archivo recurriremos al comando `cat`, así que introduciríamos la siguiente linea en consola:

```
cat /etc/network/interfaces
```

Nos aparecerán las siguientes líneas, que serán la configuración de nuestra máquina:

```
lo auto
iface eth0 inet static
address 192.168.1.128
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

En este caso nos aparece que la máquina se encuentra en una red local. Puede ser que no aparezcan todas, que solamente aparezcan sólo las primeras, o de la siguiente forma:

```
auto lo
iface lo inet loopback
```

Nos vamos a centrar en la primera, por ser más descriptiva.

En primer lugar indica `lo auto`, será el loopback, aconsejable no tocarlo, hace referencia a la dirección `127.0.0.1`.

En segundo lugar nos dice `iface`, esto nos indica de qué interface se trata, en este caso es `eth0`, una ethernet, en caso de ser una wifi la llamará `wlan0`, o con otra numeración. Podremos saber qué interfaces está reconociendo debian con el comando `ifconfig`.

Después tenemos `static`, esto hace referencia a que esta interface está configurada de forma estática, es decir, que tiene una ip asignada, que no preguntará a ningún servidor dhcp que dirección coger para conectarse a la red, por el contrario, si aquí apareciera `dinamic`, el sistema esperaría que un servidor dhcp le asigne una ip para poder acceder a la red. En este caso, hemos configurado la red de forma estática, porque queremos enrutar puertos o tener identificada a la máquina en la red, o cualquier otro propósito.

Ya hemos definido en qué posición se encuentra la interface, será estática, así que le tendrémos que indicar qué valores tiene que tener, esos valoren serán `address`, dirección ip de la máquina, dentro de una red local con puerta de enlace `192.168.1.1`, tendremos disponible 254 direcciones para asignarle a la máquina, en este caso en `address` le hemos asignado `192.168.1.128`. La `netmask`, la máscara de red para esta clase de redes, será `255.255.255.0`, la propia red, `network`, será `192.168.1.0` y el `broadcast` estará en `192.168.1.255`, y por último la `gateway` o puerta de enlace, nuestro router, `192.168.1.1`, por defecto la mayoría de isps trabajan en rangos parecidos para sus router, `192.168.0.1`, `192.168.1.1` o `192.168.2.1`, para saberlo, si anteriormente lo teníamos en dinámico, simplemente con un `ifconfig`, podríamos ver en qué red estamos y cuál es la ruta de enlace.

CENTOS

Los principales archivos de configuración son

<code>/etc/sysconfig/network</code>	Este archivo de configuración es utilizado para definir las características de red deseadas
<code>/etc/sysconfig/network-scripts/ifcfg-<interfaz></code>	Estos archivos de configuración son utilizados para especificar la configuración de la tarjeta de red.

Configuración del archivo /etc/sysconfig/network

Los parámetros que utiliza este este archivo son:

NETWORKING	Los valores que admites son:\yes Permite la configuración de los servicio de red.\no No permite la configuración de los servicio de red.
FORWARD_IPV4	Habilita el reenvío de paquetes. Los valores que admite son yes o no.
HOSTNAME	Define el nombre del equipo, el cual debe de tener la forma del Fully Qualified Domain Name (FQDN). Por ejemplo: equipo.ejemplo.net
GATEWAY	Este parametro define la dirección IP del Gateway

Configuración de la interfaz de red

El directorio de configuración de la interfaz de red se encuentra en:

/etc/sysconfig/network-scripts/

Dentro de este directorio se encuentran los archivos de configuración de los dispositivos, dependiendo del numero de interfaces de red instaladas en el computadora será el numero de archivos de configuración, el nombre de estos archivos depende del tipo de dispositivo

Ethernet ifcfg-eth0, ifcfg-eth1, ..., ifcfg-ethN.

Wi-Fi ifcfg-wlan0, ifcfg-wlan1, ..., Ifcfg-wlanN.

Modem ifcfg-ppp0, ifcfg-ppp1, ..., ifcfg-pppN.

En donde N representa el numero de interfaz a configurar. Los parámetros que admiten los archivos de configuración de la interfaz de red Ethernet son los siguientes:

DEVICE	Define el nombre del dispositivo físico
BOOTPROTO	none No utiliza ningún protocolo de arranque.\static Se define de forma manual los parámetros de red.\dhcp Obtiene los parámetros de red por medio de un servidor de DHCPC
IPADDR	Define la dirección IP asignada a ese dispositivo.
NETMASK	Define la mascara de red.
NETWORK	Define el segmento de red
HWADDR	Define el dirección MAC del dispositivo de red. Se recomienda que modificar el valor de este parámetro.
GATEWAY	Define la Dirección IP del Gateway en la red
ONBOOT	Establece si el dispositivo debe activarse con los servicios de red
DNS1, DNS2	Define la direcciones de los servidores DNS primario y secundario a utilizar.
DHCP_HOSTNAME	Esta opción establece un nombre al equipo. Utilice esta opción si el servidor DHCP requiere que el cliente especifique el nombre de su equipo antes de recibir una dirección IP.

Ejemplo del archivo de configuración.

DEVICE=eth0

```
BOOTPROTO=static IPADDR=192.168.2.10 NETMASK=255.255.255.0
NETWORK=192.168.2.0 GATEWAY=192.168.2.254 DNS1=192.168.2.1 HWADDR=
00:1E:EC:6E:CD:51}}}
```

109.2 EXTRAS comando ip

El comando ip, que forma parte de la iproute2 suite , sustituye a los comandos ifconfig y route. Es funcional en la mayoría de distros.

Vamos a ver algunos ejemplos de uso del comando ip para comenzar a familiarizarnos con él y ver como realizaríamos las tareas más comunes que hacemos con ifconfig.

Ver interfaces de red y su configuración

El comando ip addr list vendría a ser lo mismo que ejecutar ifconfig. Se puede ejecutar con el mismo resultado como ip address show o ip address list:

```
root@cli:~# ip addr list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 00:0c:29:8f:cf:fc brd ff:ff:ff:ff:ff:ff
    inet 172.16.1.11/24 brd 172.16.1.255 scope global eth0
        inet6 fe80::20c:29ff:fe8f:cfc/64 scope link
            valid_lft forever preferred_lft forever
```

También podemos usar el comando ip link show para ver la información en capa 2 (data link layer) de las interfaces de red del sistema:

```
root@cli:~# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT qlen 1000
    link/ether 00:0c:29:8f:cf:fc brd ff:ff:ff:ff:ff:ff
```

Es interesante comentar que los comandos se pueden abreviar al estilo de sistemas operativos de red como ios o junos. Como ejemplo *ip address list* sería igual que *ip a l*

Activar/desactivar interfaces de red

Para habilitar o deshabilitar una interfaz de red seguiremos utilizando **ip link**:

```
~# ip link set eth2 down
~# ip addr list eth2
3: eth2: mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
    link/ether 00:1a:73:d1:67:45 brd ff:ff:ff:ff:ff:ff

~# ip link set eth2 up
~# ip addr list eth2
3: eth2: mtu 1500 qdisc pfifo_fast state DORMANT qlen 1000
    link/ether 00:1a:73:d1:67:45 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::21a:73ff:fed1:6745/64 scope link tentative
        valid_lft forever preferred_lft forever
```

Cambiar características de la interfaz

Con ip link también podemos realizar modificaciones en las características o flags de la interfaz, como por ejemplo configurar el **modo promiscuo**, **multicast**, **arp**, **dynamic** o **allmulti**. Se utilizan los valores **on|off**.

```
~# ip link set dev eth0 promisc on
~# ip addr list eth0
2: eth0: mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:1b:24:d5:18:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.128/24 brd 192.168.1.255 scope global eth0
        inet6 fe80::21b:24ff:fed5:1899/64 scope link
            valid_lft forever preferred_lft forever

~# ip link set dev eth0 promisc off
~# ip addr list eth0
2: eth0: mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:1b:24:d5:18:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.128/24 brd 192.168.1.255 scope global eth0
        inet6 fe80::21b:24ff:fed5:1899/64 scope link
            valid_lft forever preferred_lft forever
```

Configurar una IP para la interfaz

Con **ip addr add** podemos especificar la IP, máscara (también en formato CIDR como vemos a continuación) y la IP de broadcast:

```
~# ip addr add 10.0.0.100/24 broadcast 10.0.0.255 dev eth2
~# ip addr list eth2
3: eth2: mtu 1500 qdisc pfifo_fast state DORMANT qlen 1000
    link/ether 00:1a:73:d1:67:45 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.100/24 brd 10.0.0.255 scope global eth2
        inet6 fe80::21a:73ff:fed1:6745/64 scope link
            valid_lft forever preferred_lft forever
```

Y para eliminar la IP:

```
~# ip addr del 10.0.0.100/24 dev eth2
```

También podemos crear alias en la interfaz:

```
~# ip addr add 10.0.0.101/24 broadcast 10.0.0.255 dev eth2:1
```

Cambiar la MTU de la interfaz

```
~# ip link set dev eth2 mtu 9000
```

Ver la tabla de rutas

```
~# ip route show
default via 192.168.1.1 dev eth0 proto static
169.254.0.0/16 dev eth0 scope link metric 1000
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.128 metric 1
```

Ver la tabla de ARP Cache

De todas las interfaces:

```
~# ip neighbor show  
192.168.1.1 dev eth0 lladdr 64:68:0c:6b:5f:7e REACHABLE
```

De una interfaz concreta:

```
~# ip neighbor show dev eth0  
192.168.1.1 lladdr 64:68:0c:6b:5f:7e REACHABLE
```

Añadir un default gateway:

```
#ip route add default via 192.168.99.254
```

Añadir/quitar rutas:

```
#ip route add 192.168.55.0/24 via 192.168.1.254 dev eth1  
#ip route del 192.168.55.0/24 via 192.168.1.254 dev eth1
```

Prohibir una ruta (avisando con un destination unreachable):

```
#ip route add prohibit 209.10.26.51
```

Prohibir una ruta a un remitente en concreto:

```
#ip route add prohibit 209.10.26.51 from 192.168.99.35
```

109.3 Soluciones para problemas simples de red.

Peso en el examen de certificación: 4 puntos.

Objetivo: Solucionar problemas de redes en clientes.

Conceptos y áreas de conocimiento:

- Configuración manual y automática de interfaces de red y tablas de enrutamiento. Añadir, iniciar, parar, reiniciar, eliminar o reconfigurar interfaces de red.
- Cambiar, visualizar o configurar la tabla de enrutamiento corrigiendo valores impropios manualmente.
- Depurar problemas asociados con la configuración de la red.

Términos y utilidades

- ifconfig
- ifup
- ifdown
- route
- host
- hostname
- dig
- netstat
- ping
- traceroute

ÍNDICE

109.3.1. ifconfig

109.3.2. ifup

109.3.3. ifdown

109.3.4. route

109.3.5. host

109.3.6. hostname

109.3.7. dig

109.3.8. netstat

109.3.9. ping

109.3.10. traceroute

Una parte importante de la función del administrador es solucionar problemas de conectividad y localizar las fuentes de dichos problemas. Muchas de las herramientas introducidas en los capítulos anteriores también se pueden utilizar como herramientas de solución de problemas para ayudar en este proceso.

Este capítulo retoma algunos de estos comandos y expone como se pueden utilizar como recursos de diagnóstico, además de recursos de configuración. Es importante destacar que todos los cambios realizados con estos comandos no son cambios permanentes. Una vez se reinicia el ordenador se restablecen los parámetros establecidos en los ficheros de configuración, en el caso de Debian estos ficheros se encuentran en */etc/network/interfaces*.

109.3.1.- IFCONFIG

Esta herramienta permite visualizar todas las interfaces de red activadas. Se muestran en diferentes bloques y cada uno corresponde a un dispositivo de red ya sea físico o virtual.

\$ ifconfig

```
eth0 Vínculo encap:Ethernet HWaddr 00:XX:XX:XX:XX:XX
      inet adr:192.168.1.60 Bcast:192.168.1.255 Máscara:255.255.255.0
          adr inet6: fe80::21b:fcff:fec9:f81d/64 Scope:Vínculo
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
              RX packets:16522 errors:0 dropped:0 overruns:0 frame:0
              TX packets:13631 errors:0 dropped:0 overruns:0 carrier:2
              collisions:0 lg file transmission:1000
              RX bytes:17732221 (16.9 MB) TX bytes:1648879 (1.5 MB)

lo Vínculo encap:Local Loopback
      inet adr:127.0.0.1 Máscara:255.0.0.0
          adr inet6: ::1/128 Scope:Host
              UP LOOPBACK RUNNING MTU:16436 Metric:1
              RX packets:2051 errors:0 dropped:0 overruns:0 frame:0
              TX packets:2051 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 lg file transmission:0
              RX bytes:598941 (584.9 KiB) TX bytes:598941 (584.9 KiB)
```

Para la visualización de todas las interfaces de red, activadas o no, se utiliza la opción a.

\$ ifconfig -a

Para la visualización de una interfaz de red en concreto, se escribe el nombre de la interfaz.

\$ ifconfig eth0

```
eth0 Vínculo encap:Ethernet HWaddr 00:XX:XX:XX:XX:XX
      inet adr:192.168.1.60 Bcast:192.168.1.255 Máscara:255.255.255.0
          adr inet6: fe80::21b:fcff:fec9:f81d/64 Scope:Vínculo
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
              RX packets:16522 errors:0 dropped:0 overruns:0 frame:0
              TX packets:13631 errors:0 dropped:0 overruns:0 carrier:2
              collisions:0 lg file transmission:1000
              RX bytes:17732221 (16.9 MB) TX bytes:1648879 (1.5 MB)
```

Si queremos parar la interfaz de red eth0, utilizamos la opción down, en cambio para activarla utilizamos la opción up:

\$ ifconfig eth0 down

\$ ifconfig eth0 up

Cuando necesitamos activar una interfaz de red con una dirección IP distinta a la que tenemos actualmente, añadimos la IP al comando:

\$ ifconfig eth0 192.168.1.4 up

Para la configuración de la interfaz de red eth0 para la dirección de clase C 192.168.1.2, seguiremos el siguiente comando:

\$ ifconfig eth0 192.168.1.2

Si además queremos especificar la máscara hay que añadirle la opción inet y netmask, de la siguiente forma:

```
$ ifconfig eth0 inet 192.168.1.2 netmask 255.255.255.0
```

Para la activación del modo promiscuo, utilizamos la opción promisc. Esta opción se utiliza para ver qué paquetes atraviesan tu red.

```
$ ifconfig eth0 promisc
```

En cambio para parar el modo promiscuo, añadimos un guión delante la opción promisc.

```
$ ifconfig eth0 -promisc
```

Para la activación del protocolo ARP, utilizamos el parámetro arp, este es el responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP.

```
$ ifconfig eth0 arp
```

Y para detener el protocolo ARP, añadimos un guión delante la opción arp.

```
$ ifconfig eth0 -arp
```

109.3.2.- IFUP

Esta herramienta permite activar una interfaz de red. Es equivalente al comando anteriormente explicado:

```
$ ifconfig eth0 up
```

Su traducción sería esta:

```
$ ifup eth0
```

Es importante tener en cuenta que la interfaz que queremos activar este dentro de este fichero /etc/network/interfaces sino nos aparecerá un error:

```
$ ifup eth3
```

Ignoring unknown interface eth3

Y si en cambio, ya está configurada, nos da el siguiente aviso:

```
$ ifup eth3
```

ifup:interface eth3 already configured

Si queremos encender al mismo tiempo todas las interfaces del sistema, utilizamos la opción -a:

```
$ ifup -a
```

Cuando necesitamos consultar más información sobre los errores que da este comando, utilizamos la opción verbose:

```
$ ifup --verbose -a
```

109.3.3.- IFDOWN

Esta herramienta permite parar una interfaz de red. Cuando la interfaz se para, se eliminan todas las rutas asociadas en la tabla de enrutamiento. Es equivalente al comando anteriormente explicado:

```
$ ifconfig eth0 down
```

Su traducción sería esta:

```
$ ifdown eth0
```

Si queremos parar al mismo tiempo todas las interfaces del sistema, utilizamos la opción -a.

```
$ ifdown -a
```

Hay que vigilar no parar las interfaces de red en servidores remotos a los que nos conectamos en SSH, ya que esto nos impediría volver a acceder a la máquina.

109.3.4.- ROUTE

Esta herramienta nos permite visualizar o configurar las rutas IP actuales de la máquina.

```
# route
```

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.2.0 * 255.255.255.0 U 0 0 0 eth0
link-local * 255.255.0.0 U 0 0 0 eth0
loopback * 255.0.0.0 U 0 0 0 lo
default 10.0.2.2 0.0.0.0 UG 0 0 0 eth0
```

La tabla de enrutamiento de una máquina se compone de una parte fija que se conserva al reiniciar el ordenador, y otra parte se guarda temporalmente. Con el siguiente comando vemos las dos partes:

```
# route -CFvee
```

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface MSS Window
irtt
10.0.2.0 * 255.255.255.0 U 0 0 0 eth0 0 0 0
link-local * 255.255.0.0 U 0 0 0 eth0 0 0 0
loopback * 255.0.0.0 U 0 0 0 lo 0 0 0
default 10.0.2.2 0.0.0.0 UG 0 0 0 eth0 0 0 0
```

Para crear la entrada de loopback:

```
# route add -net 127.0.0.0
```

Si queremos crear una ruta hacia la red 10.2.2.0, que pasa por eth0:

```
# route add -net 10.2.2.0 netmask 255.255.255.0 eth0
```

```
# route add -net 10.2.2.0/24 eth0
```

Si queremos crear una ruta hacia la red 10.2.2.0, que pasa por el enrutador 10.1.1.253:

```
# route add -net 10.2.2.0 netmask 255.255.255.0 gw 10.1.1.253
```

```
# route add -net 10.2.2.0/24 gw 10.1.1.253
```

El siguiente comando sirve para crear la pasarela por defecto hacia el enrutador:

```
# route add default gw 192.168.10.1
```

O para borrar la ruta hacia la red 172.16.1.2:

```
# route del -net 172.16.1.2 eth0
```

109.3.5.- HOST

Para poder obtener todos los hosts de un dominio es preciso estar dentro de este y también que se te permita usarlo.

En la siguiente tabla se presentan las opciones más utilizadas para este comando:

Opciones del comando host

Opción	Función
-l	Lista todo el dominio, además de todas las máquinas registradas en el servidor DNS (esto puede ser muy largo)
-v	Establece el modo detallado para ver la salida.

```
$ host -v google.com
```

```
Trying "google.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 461
;; flags: qr rd ra; QUERY: 1, ANSWER: 16, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com. IN A

;; ANSWER SECTION:
google.com. 250 IN A 173.194.34.6
google.com. 250 IN A 173.194.34.10
google.com. 250 IN A 173.194.34.11
google.com. 250 IN A 173.194.34.2
google.com. 250 IN A 173.194.34.4
google.com. 250 IN A 173.194.34.0
google.com. 250 IN A 173.194.34.14
google.com. 250 IN A 173.194.34.8
google.com. 250 IN A 173.194.34.9
google.com. 250 IN A 173.194.34.1
google.com. 250 IN A 173.194.34.15
google.com. 250 IN A 173.194.34.7
google.com. 250 IN A 173.194.34.12
google.com. 250 IN A 173.194.34.5
google.com. 250 IN A 173.194.34.3
google.com. 250 IN A 173.194.34.13
```

109.3.6.- HOSTNAME

Esta herramienta nos proporciona el nombre de la máquina.

```
$ hostname
```

También permite cambiar el nombre de una máquina.

```
# hostname nombre_maquina.dominio
```

Recuerda que las órdenes no acostumbran a generar cambios permanentes al sistema. Para que el cambio sea permanente es necesario modificar el fichero `/etc/hostname`.

Para reforzar el cambio de nombre de la máquina sin tener que reiniciarla utilizamos la opción start.

hostname nombre_maquina start

109.3.7.- DIG

Es la herramienta más completa para la búsqueda de resolución de nombres de DNS.

Su sintaxis es la siguiente:

\$ dig [@servidor]dominio[-c class][-t type][-x addr][+ options][-options]

\$ dig www.google.com

```
;;<>> DiG 9.7.3 <>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63278
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com. IN A

;; ANSWER SECTION:
www.google.com. 604733 IN CNAME www.l.google.com.
www.l.google.com. 233 IN A 173.194.34.17
www.l.google.com. 233 IN A 173.194.34.20
www.l.google.com. 233 IN A 173.194.34.18
www.l.google.com. 233 IN A 173.194.34.16
www.l.google.com. 233 IN A 173.194.34.19
```

```
;; Query time: 41 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Feb 15 12:27:46 2012
;; MSG SIZE rcvd: 132
```

A veces, es útil hacer la consulta con un nombre de servidor distinto al por defecto. Si queremos utilizar la dirección IP, en lugar del nombre del servidor utilizando la opción @.

\$ dig @192.168.10.10 www.google.es

Para reforzar el cambio de nombre de la máquina sin tener que reiniciarla utilizamos la opción start.

hostname nombre_maquina start

Si queremos obtener los servidores de correo de un dominio utilizamos la opción mx.

dig mx dominio

Y para obtener los servidores de DNS de un dominio, utlizamos la opción ns:

dig ns +norec +noques +nostats +nocmd @servidor dominio

También la podemos utilizar para comprobar la delegación de una zona:

```
# dig trace google.com
```

109.3.8.- NETSTAT

Dependiendo de las opciones, netstat muestra las conexiones de red, tablas de enrutamiento, las estadísticas de la interfaz, conexiones enmascaradas y las pertenencias de multidifusión.

Opciones del comando netstat

Opción	Función
-c	Obtiene un muestreo cada segundo hasta que Ctr-C lo interrumpe
-g	Visualiza los grupos multicast
-i	Visualiza la lista de interfaces.
-m	Visualiza las conexiones enmascaradas o NAT
-n	Visualiza las direcciones en modo numérico en lugar del host, el puerto y los nombres de usuario.
-p	Visualiza el proceso ID (PID) y el nombre del proceso
-r	Visualiza la tabla de enrutamiento en formato del proceso
-s	Visualiza estadísticas del uso de la red por protocolos
-v	Visualiza en modo detallado

Para visualizar todas las conexiones sin nombres de host ni de protocolos utilizamos:

```
# netstat -an -tcp
```

Para mostrar el PID y el nombre del proceso al que pertenece cada socket, utilizamos la opción -p. Así podríamos identificar la causa de algún problema en la red.

```
# netstat -p
```

Para mostrar la tabla de NAT, en sistemas donde el router esté activado utilizamos la opción -M:

```
# netstat -M
```

La visualización del netstat por defecto, suele ser bastante larga. En caso que solo nos interesen las conexiones establecidas por los protocolos TCP i UDP, utilizamos las opciones -t i -u.

```
# netstat -tu
```

Si además queremos consultar las conexiones que están abiertas añadiremos la opción -l.

```
# netstat -tul
```

Cuando queramos visualizar todos los puertos y procesos, utilizamos el siguiente comando:

```
# netstat -A inet -lnp
```

109.3.9.- PING

Este comando es una herramienta para el diagnóstico de la conectividad de la red, utiliza el protocolo ICMP. Su función es enviar un paquete ICMP ECHO_REQUEST y espera una respuesta ECHO_RESPONSE ICMP. Si obtenemos respuesta podremos decir que el host al que hemos hecho ping es accesible desde nuestra máquina.

Podemos hacer un ping a todas las máquinas de la red utilizando la dirección de broadcast:

\$ ping -b 192.168.1.255

También se puede hacer un ping a la dirección broadcast de multicast:

\$ ping 224.0.0.1

Este comando se puede utilizar como traceroute, con la opción -R.

\$ ping -nR www.google.com

109.3.10.- TRACEROUTE

Este comando se utiliza para determinar que ruta siguen los paquetes IP para llegar a una máquina en concreto, mostrando los gateways o routers intermedios.

Podéis comprobar que nunca siguen la misma ruta.

\$ traceroute www.google.es

```
traceroute: Warning: www.google.es has multiple addresses; using 216.239.59.99
traceroute to www.l.google.com (216.239.59.99), 30 hops max, 40 byte packets
1 192.168.1.1 (192.168.1.1) 0.469 ms 0.496 ms 0.429 ms
2 192.168.153.1 (192.168.153.1) 50.701 ms 47.989 ms 48.041 ms
3 97.Red-81-46-52.staticIP.rima-tde.net (81.46.52.97) 51.949 ms 52.153 ms 51.969 ms
4 33.Red-81-46-5.staticIP.rima-tde.net (81.46.5.33) 63.792 ms 63.746 ms 64.027 ms
5 84.16.8.125 (84.16.8.125) 62.078 ms 64.046 ms 64.030 ms
6 P12-0-grtlontl2.red.telefonica-wholesale.net (213.140.43.146) 98.030 ms 99.910 ms 98.018 ms
7 72.14.198.9 (72.14.198.9) 98.140 ms 97.968 ms 98.028 ms
8 66.249.95.107 (66.249.95.107) 109.928 ms 110.080 ms 109.853 ms
9 64.233.174.185 (64.233.174.185) 109.930 ms 72.14.232.241 (72.14.232.241) 110.133 ms 109.800 ms
10 216.239.49.114 (216.239.49.114) 114.104 ms 114.010 ms 109.848 ms
11 216.239.59.99 (216.239.59.99) 110.093 ms 110.094 ms 109.865 ms
```

Opciones del comando traceroute

Opción Función

-f Establece la prueba inicial para el valor ttl, en lugar de 1.

-g Visualiza de forma numérica la dirección en lugar de nombres.

-i Establece el modo detallado para ver la salida.

-m establece el tiempo de espera en segundos para devolver paquetes ICMP, en lugar de 5.

No hay una manera directa para obtener esta información, por lo que el traceroute utiliza el tiempo de vida (TTL) del campo encabezado IP, y sigue las respuestas de error de las pasarelas. Cuando el traceroute establece el TTL a 1 conocemos la primera máquina de la ruta, cuando establece el TTL a 2 conocemos la segunda máquina, y así sucesivamente hasta que el ping vuelve.

Sería equivalente a ejecutar el comando ping incrementando el valor de la opción -t hasta que responda.

\$ ping -t 1 www.google.es

\$ ping -t 2 www.google.es

109.4 Configuración del DNS cliente.

Peso en el examen de certificación: 2 puntos.

Objetivo: **Configuración del servicio DNS en la parte de cliente.**

Conceptos y áreas de conocimiento:

- Uso de DNS en una sistema local.
- Modificar el orden en el que se realiza la resolución de nombres.

Términos y utilidades:

- /etc/hosts
- /etc/resolv.conf
- /etc/nsswitch.conf

ÍNDICE

109.4.1. Componentes de un sistema DNS

109.4.2. Partes de un nombre de dominio

109.4.3. Cliente DNS

 109.4.3.1 Ficheros relacionados

 109.4.3.2 Ejemplos

Un sistema DNS es un sistema que nos permite usar nombres de dominio en lugar de direcciones IP. Su principal ventaja es que para nosotros es mucho más fácil recordar un nombre que una dirección IP.

La transformación de los nombres en direcciones IP es la resolución de nombre y la transformación de las direcciones IP en nombres de dominio es la resolución inversa.

109.4.1. Componentes de un sistema DNS

Para la operación práctica del sistema DNS se utilizan tres componentes principales:

- **Los Clientes DNS:** Un programa cliente DNS que se ejecuta en la computadora del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS (Por ejemplo: ¿Qué dirección IP corresponde a nombre.dominio?);
- **Los Servidores DNS:** Que contestan las peticiones de los clientes. Los servidores recursivos tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada.
- **Zonas de autoridad:** porciones del espacio de nombres de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad.

Una zona es una parte de un dominio gestionada por un servidor particular. Una zona puede gestionar uno o varios subdominios y se puede repartir un subdominio en varias zonas. Una zona representa una unidad de administración, es la forma en la que se distribuye la autoridad sobre un determinado dominio.

109.4.2 Partes de un nombre de dominio

Un nombre de dominio usualmente consiste en dos o más partes (técnicamente etiquetas), separadas por puntos. Ej: www.lpi.org, es.wikipedia.org

El nombre completo no abreviado se llama **FQDN** (Fully Qualified Domain Name), se lee de derecha a izquierda y cada nivel de la jerarquía está separado por un punto. El elemento que está más a la derecha se llama **TLD** (Top Level Domain) y el que está más a la izquierda representa el *anfitrión*.

La longitud máxima permitida para un FQDN es **255 caracteres**, con una restricción de **63 caracteres** para cada elemento.

El DNS consiste en un conjunto jerárquico de servidores DNS. Cada dominio o subdominio tiene una o más **zonas de autoridad** que publican la información acerca del dominio y los nombres de servicios de cualquier dominio incluido. La jerarquía de las zonas de autoridad coincide con la jerarquía de los dominios. Al inicio de esa jerarquía se encuentran los servidores raíz: los servidores que responden cuando se busca resolver un dominio de primer y segundo nivel.

109.4.3.Cliente DNS

Los usuarios generalmente no se comunican directamente con el servidor DNS: la resolución de nombres se hace de forma transparente por las aplicaciones del cliente (por ejemplo, navegadores, clientes de correo y otras aplicaciones que usan Internet).

Al realizar una petición que requiere una búsqueda de DNS, la petición se envía al servidor DNS local del sistema operativo. El sistema operativo, antes de establecer alguna comunicación, comprueba si la respuesta se encuentra en la memoria caché. En el caso de que no se encuentre, la petición se enviará a uno o más servidores DNS.

La mayoría de usuarios domésticos utilizan como servidor DNS el proporcionado por el proveedor de servicios de Internet. La dirección de estos servidores puede ser configurada de forma manual o automática mediante DHCP. En otros casos, los administradores de red tienen configurados sus propios servidores DNS.

Se utiliza el fichero **/etc/resolv.conf** para indicar al sistema qué servidores de nombres i qué dominios hay que consultar para que *resolver* (librería estandar de Linux) resuelva la peticiones DNS clientes. No hace falta añadir herramientas adicionales.

Un ejemplo del contenido de este fichero seria:

```
$cat /etc/resolv.conf
domain midominio.org
search midominio.org
nameserver 192.168.1.1
nameserver 192.168.1.2
```

- **domain:** nombre del dominio local. Las peticiones se suelen reducir a unos atajos relativos hacia el dominio local. Si no está creado, se debe determinar el nombre del dominio a partir del nombre completo del anfitrión: corresponde a la parte ubicada después del primer ‘.’
- **search:** lista de dominios de búsqueda. Por defecto, durante la utilización de atajos (nombres

de anfitriones cortos) el resolver inicia una búsqueda sobre el dominio definido por la línea domain, pero se puede especificar aquí una lista de dominios separados por espacios o comas.

- **nameserver**: dirección IP del servidor de nombres (el servidor DNS). Se puede colocar un máximo de tres. El resolver intenta utilizar el primero, En caso de fracaso (timeout), pasa al segundo, y así sucesivamente.
- **opciones**: se pueden especificar opciones. Por ejemplo, timeout:n, donde n (en segundos) indica el tiempo de espera de respuesta de un servidor de nombres antes de pasar al siguiente.

109.4.3.1. Ficheros relacionados.

- */etc/hosts* y */etc/networks* Sin siquiera utilizar un servidor de nombres, es posible establecer una correspondencia entre las direcciones IP y los nombres de las máquinas dentro del fichero /etc/hosts.

```
192.168.1.1 server www1 ftp
```

```
192.168.1.11 puesto1  
192.168.1.12 puesto2
```

Puede hacer lo mismo para nombrar las redes en el fichero */etc/networks*

```
loopnet 127.0.0.0  
localnet 192.168.1.0
```

- */etc/nsswitch.conf*

Este fichero permite determinar el orden en el cual el resolver recupera su información. Las dos líneas en negrita en el ejemplo indican que, durante una petición de resolución de nombre (o red), los ficheros son prioritarios. Primerose se lee el fichero /etc/hosts; luego, si el resolver no encuentra la información, busca mediante una resolución DNS.

```
passwd: compat  
group: compat  
hosts: files dns  
networks: files dns  
  
services: files  
protocols: files  
rpc: files  
ethers: files  
netmasks: files  
netgroup: files nis  
publickey: files  
  
bootparams: files  
automount: files nis  
aliases: files
```

109.4.3.2. Ejemplos

- **Ejemplo1**

El fichero *resolv.conf* contiene una lista de servidores, si está vacía el sistema considerará que el servidor está en la máquina local

- **Ejemplo2**

Se puede especificar un servidor dns local primario y otro externo como secundario. Se pueden especificar hasta 3 servidores de nombres.

```
#resolv.conf
domain midominio.es
search midominio.es
nameserver 192.168.0.10
nameserver 8.8.8.8
```

- **Ejemplo3**

La opción **domain**, nos permite usar nombres cortos (sin dominio) para máquinas que están en nuestro dominio.

Normalmente, para conectarnos a una máquina de la misma red, no queremos poner el dominio completo, sino su nombre. Por ejemplo, server1 en lugar de server1.midominio.es

```
#resolv.conf
domain midominio.es
nameserver 192.168.0.10
```

De esta forma *domain*, nos permite especificar un dominio predeterminado que se añade a las peticiones cuando su búsqueda inicial falla. Así pues, al buscar server1 y fallar el servidor de nombres buscándolo en internet, le añade automáticamente su dominio predeterminado y ya se puede resolver.

- **Ejemplo4**

Las opciones **domain** y **search** permiten usar nombres cortos (sin dominio) para máquinas que están en nuestro dominio.

Para conectarnos a una máquina de la misma red, pero de otro departamento no queremos poner el dominio completo, sino su nombre.

Usaremos la opción **search** y especifacaremos la lista de dominios donde resolver nombres cortos. Los elementos de la lista se separan mediante espacios o tabulaciones.

Por ejemplo, si estamos en el departamento de matemáticas y queremos acceder una máquina que está en el departamento de física. Con el siguiente fichero:

- Accederemos a la máquina gauss.mates.midominio.es tecleando gauss.
- Accederemos a la máquina quark.fisica.midominio.es tecleando quark.fisica.

```
#resolv.conf
domain midominio.es
search mates.midominio.es midominio.es nameserver 192.168.0.10
```

Las opciones **search** y **domain** son mútamente excluyentes y no deberían aparecer más de una vez.

Si se especifica más de un domain o search prevalece la última instancia.

Si ninguna de las dos se pone, el sistema intentará asignar a los nombres cortos el dominio de la máquina local. Si el nodo local no tiene dominio, se asumirá que el dominio predeterminado es el raíz.

- **Ejemplo5**

Asumamos que queremos conectarnos a la máquina foot.dominioempresa.com. Por un error tecleamos el nombre corto foo, que no existe.

```
#resolv.conf  
domain midominio.es  
nameserver 192.168.0.10
```

Cuando se trate de traducir el nombre foo, el sistema empezará por buscar directamente foo y si falla probará con foo.dominioempresa.com

El servidor de la empresa nos responderá que no existe el nodo.

109.4 EXTRAS

109.4 EXTRAS Instalación y configuración de un servidor DNS

Antes de analizar el proceso de instalación y configuración de un servidor de DNS, se va a tratar de mostrar el proceso de resolución de nombres de dominio. Se ha de tener muy en cuenta que DNS no es más que una base de datos distribuida en innumerables servidores por la internet. La estructura del sistema DNS se basa en una estructura de arbórea en donde se definen los dominios de nivel superior (llamados **TLD**, *Dominios de Nivel Superior*); esta estructura está conectada a un nodo raíz representado por un punto. Cada nodo del árbol se llama **nombre de dominio** y tiene una *etiqueta* con una longitud máxima de 63 caracteres. Por lo tanto, todos los nombres de dominio conforman una estructura arbórea inversa en donde cada nodo está separado del siguiente nodo por un punto (".").

El extremo de la bifurcación se denomina **host**, y corresponde a un equipo o entidad en la red. La palabra "**dominio**" corresponde formalmente al sufijo de un nombre de dominio, es decir, la recopilación de las etiquetas de nodo de la estructura arbórea, con excepción del ordenador.

El nombre absoluto está relacionado con todas las etiquetas de nodo de una estructura arbórea, separadas por puntos y que termina con un punto final que se denomina la **dirección FQDN** (*Nombre de Dominio totalmente calificado*). La profundidad máxima de una estructura arbórea es 127 niveles y la longitud máxima para un nombre FQDN es 255 caracteres. La dirección FQDN permite ubicar de manera única un equipo en la red de redes. Por lo tanto, *lpi.org.es.* es una dirección FQDN.

Los equipos llamados *servidores de nombres de dominio* permiten establecer la relación entre los nombres de dominio y las direcciones IP de los equipos de una red.

Cada dominio cuenta con un servidor de nombre de dominio, llamado *servidor de nombre de dominio principal*, así como también un *servidor de nombre de dominio secundario*, que puede encargarse del servidor de nombre de dominio principal en caso de falta de disponibilidad.

Cada servidor de nombre de dominio está especificado en el servidor de nombre de dominio en el nivel superior inmediato, lo que significa que la autoridad sobre los dominios puede delegarse implícitamente. El sistema de nombre es una arquitectura distribuida, en donde cada entidad es responsable de la administración de su nombre de dominio. Por lo tanto, no existe organización alguna que sea responsable de la administración de todos los nombres de dominio.

Los servidores relacionados con los dominios de nivel superior (TLD) se llaman "**servidores de**

dominio de nivel superior". Están distribuidos por todo el mundo y sus nombres van desde "a.root-servers.net" hasta "m.root-servers.net". Son estos:

```
; This file holds the information on root name servers needed to
; initialize cache of Internet domain name servers
; (e.g. reference this file in the "cache . <file>" configuration file of BIND domain name servers).
;
; This file is made available by InterNIC
; under anonymous FTP as
;   file          /domain/named.cache
;   on server    FTP.INTERNIC.NET
; -OR-
;   last update: Jan 3, 2013
;   related version of root zone: 2013010300
;
; formerly NS.INTERNIC.NET
;
.          3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000      A    198.41.0.4
A.ROOT-SERVERS.NET. 3600000      AAAA  2001:503:BA3E::2:30
;
; FORMERLY NS1.ISI.EDU
;
.          3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000      A    192.228.79.201
;
; FORMERLY C.PSI.NET
;
.          3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000      A    192.33.4.12
;
; FORMERLY TERP.UMD.EDU
;
.          3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000      A    199.7.91.13
D.ROOT-SERVERS.NET. 3600000      AAAA  2001:500:2D::D
;
; FORMERLY NS.NASA.GOV
;
.          3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET. 3600000      A    192.203.230.10
;
; FORMERLY NS.ISC.ORG
;
.          3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET. 3600000      A    192.5.5.241
F.ROOT-SERVERS.NET. 3600000      AAAA  2001:500:2F::F
;
; FORMERLY NS.NIC.DDN.MIL
;
.          3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET. 3600000      A    192.112.36.4
;
; FORMERLY AOS.ARL.ARMY.MIL
;
.          3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET. 3600000      A    128.63.2.53
H.ROOT-SERVERS.NET. 3600000      AAAA  2001:500:1::803F:235
;
```

```

; FORMERLY NIC.NORDU.NET
;
.          3600000      NS   I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET. 3600000      A    192.36.148.17
I.ROOT-SERVERS.NET. 3600000      AAAA 2001:7FE::53
;
; OPERATED BY VERISIGN, INC.
;
.          3600000      NS   J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET. 3600000      A    192.58.128.30
J.ROOT-SERVERS.NET. 3600000      AAAA 2001:503:C27::2:30
;
; OPERATED BY RIPE NCC
;
.          3600000      NS   K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET. 3600000      A    193.0.14.129
K.ROOT-SERVERS.NET. 3600000      AAAA 2001:7FD::1
;
; OPERATED BY ICANN
;
.          3600000      NS   L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET. 3600000      A    199.7.83.42
L.ROOT-SERVERS.NET. 3600000      AAAA 2001:500:3::42
;
; OPERATED BY WIDE
;
.          3600000      NS   M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET. 3600000      A    202.12.27.33
M.ROOT-SERVERS.NET. 3600000      AAAA 2001:DC3::35
; End of File

```

El servidor de nombre de dominio define una zona, es decir, una recopilación de dominios sobre la cual tiene autoridad. Si bien el sistema de *nombres de dominio* es transparente para el usuario, se deben tener en cuenta los siguientes puntos:

- Cada equipo debe configurarse con la dirección de un equipo que sea capaz de transformar cualquier nombre en una dirección IP. Este equipo se llama Servidor de nombres de dominio.
- También debe definirse la dirección IP de un segundo *Servidor de nombres de dominio* (Servidor de nombres de dominio secundario): el servidor de nombres de dominio secundario puede encargarse del servidor de nombres de dominio principal en caso de fallas en el sistema.

El servidor que se utiliza con más frecuencia se llama **BIND** (*Berkeley Internet Name Domain*). Es un software gratuito para sistemas UNIX, fue desarrollado inicialmente por la Universidad de Berkeley en California y en la actualidad está mantenido por *ISC* (*Internet Systems Consortium*).

Resolución de nombres de dominio

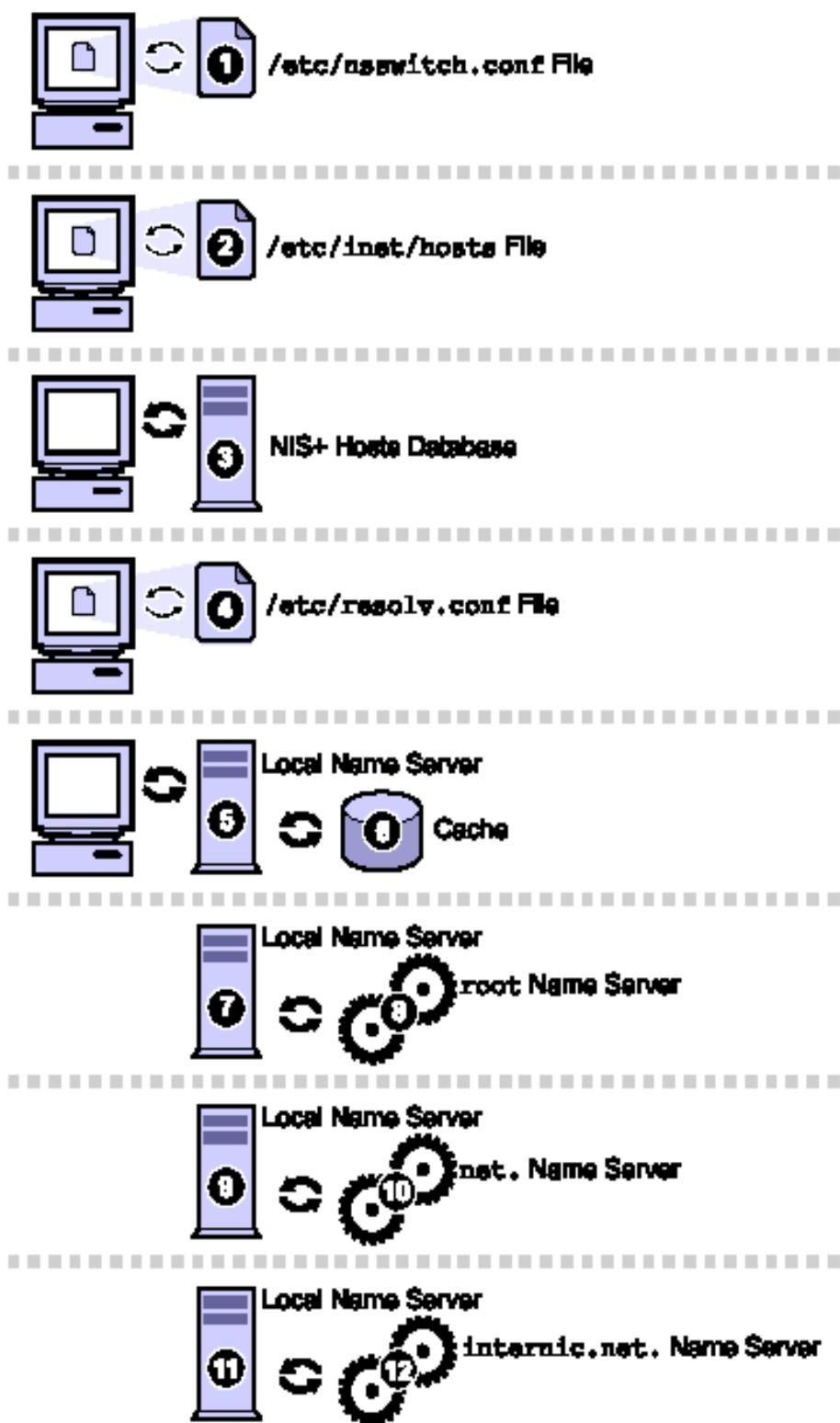
El mecanismo que consiste en encontrar la dirección IP relacionada al nombre de un ordenador se conoce como "**resolución del nombre de dominio**". La aplicación que permite realizar esta operación (por lo general, integrada en el sistema operativo) se llama "**resolución**".

Cuando una aplicación desea conectarse con un host conocido a través de su nombre de dominio

(por ejemplo, "lpi.org.es"), ésta interroga al servidor de nombre de dominio definido en la configuración de su red. De hecho, todos los equipos conectados a la red tienen en su configuración las direcciones IP de ambos servidores de nombre de dominio del proveedor de servicios.

Entonces se envía una solicitud al primer servidor de nombre de dominio (llamado el "servidor de nombre de dominio principal"). Si este servidor de nombre de dominio tiene el registro en su caché, lo envía a la aplicación; de lo contrario, interroga a un servidor de nivel superior (en nuestro caso un servidor relacionado con el TLD ".es"). El servidor de nombre de nivel superior envía una lista de servidores de nombres de dominio con autoridad sobre el dominio (en este caso, las direcciones IP de los servidores de nombres de dominio principal y secundario para *org.es*).

Entonces el servidor de nombres de dominio principal con autoridad sobre el dominio será interrogado y devolverá el registro correspondiente al dominio del servidor (en nuestro caso *lpi*).



Dominios de nivel superior

Existen dos categorías de **TLD** (*Dominios de Nivel Superior*):

- Los dominios que se conocen como "genéricos", llamados **gTLD** (*TLD genérico*). Los gTLD son nombres de dominio de nivel superior genéricos que ofrecen una clasificación de acuerdo con el sector de la actividad. Entonces cada gTLD tiene sus propias reglas de

acceso:

- gTLD historial:
 - **.arpa** relacionado con equipos pertenecientes a la red original;
 - **.com** inicialmente relacionado con empresas con fines comerciales. Sin embargo, este TLD se convirtió en el "TLD predeterminado" y hasta personas reales pueden adquirir dominios con esta extensión.
 - **.edu** relacionado con las organizaciones educativas;
 - **.gov** relacionado con las organizaciones gubernamentales;
 - **.int** relacionado con las organizaciones internacionales;
 - **.edu** relacionado con las organizaciones militares;
 - **.net** inicialmente relacionado con las organizaciones que administran redes. Con el transcurso de los años este TLD se ha convertido en un TLD común, y hasta personas reales pueden adquirir dominios con esta extensión.
 - **.org** está normalmente relacionado con organizaciones sin fines de lucro.
- nuevos gTLD presentado en noviembre de 2000 por ICANN:
 - **.aero** relacionado con la industria aeronáutica;
 - **.biz (negocios)** relacionado con empresas comerciales;
 - **.museum** relacionada con los museos;
 - **.name** relacionada con el nombre de personas reales o imaginarias;
 - **.info** relacionado con organizaciones que manejan información;
 - **.coop** relacionado con cooperativas;
 - **.pro** relacionado con profesiones liberales.
- gTLD especial:
 - **.arpa** relacionado con las infraestructuras para la administración de redes. El arpa gTLD también sirve para la resolución inversa de equipos en red y permite hallar el nombre relacionado con una dirección IP.
- Los dominios que se conocen como "nacionales", se llaman **ccTLD** (código de país TLD). El ccTLD está relacionado con los diferentes países y sus nombres refieren a las abreviaturas del nombre del país definidas en la norma ISO 3166. La tabla a continuación resume la lista de ccTLD.

Código	País
AC	Islas Ascensión.
AD	Andorra
AE	Emiratos Árabes Unidos
AF	Afganistán
AG	Antigua y Barbuda
AI	Anguila
AL	Albania
AM	Armenia
AN	Antillas Neerlandesas
AO	Angola
AQ	Antártida
AR	Argentina
AS	Samoa Americana
AT	Austria
AU	Australia

AW	Aruba
AZ	Azerbaiyán
BA	Bosnia y Herzegovina
BB	Barbados
BD	Bangladesh
BE	Bélgica
BF	Burkina Faso
BG	Bulgaria
BH	Bahrein
BI	Burundi
BJ	Benín
BM	Bermudas
BN	Brunei
BO	Bolivia
BR	Brasil
BS	Bahamas
BT	Bhután
BV	Isla Bouvet
BW	Botswana
BY	Bielorrusia
BZ	Belice
CA	Canadá
CC	Islas Cocos
CD	República Democrática del Congo
CCM	República Centroafricana
CG	Congo
CH	Suiza
CI	Costa de Marfil
CK	Islas Cook
CL	Chile
CM	Camerún
CN	China
CO	Colombia
COM	Organización comercial
CR	Costa Rica
CU	Cuba
CV	Cabo Verde
CX	Islas Christmas
CY	Chipre
CZ	República Checa
DE	Alemania
DJ	Djibouti

DK	Dinamarca
DM	Dominica
DO	República Dominicana
DZ	Argelia
EC	Ecuador
EDU	Organización con enlaces relacionados con la educación
EE	Estonia
EG	Egipto
EH	Sahara Occidental
ER	Eritrea
ES	España
ET	Etiopía
EU	Europa
FI	Finlandia
FJ	Fiji
FK	Islas Falkland (Malvinas)
FM	Micronesia
FO	Islas Feroe
FR	Francia
FX	Francia (Territorio EEEE europeo)
GA	Gabón
GB	Gran Bretaña
GD	Granada
GE	Georgia
GF	Guayana Francesa
GG	Guernsey
GH	Ghana
GI	Gibraltar
GL	Groenlandia
GM	Gambia
GN	Guinea
GOV	Organización gubernamental
GP	Guadalupe
GQ	Guinea Ecuatorial
GR	Grecia
GS	Georgia del Sur
GT	Guatemala
GU	Guam (USA)
GW	Guinea Bissau
GY	Guyana
HK	Hong Kong
HM	Islas Heard y McDonald

HN	Honduras
HR	Croacia
HT	Haití
HU	Hungría
ID	Indonesia
IE	Irlanda
IL	Israel
IM	Isla de Man
IN	India
IO	Territorio Británico del Océano Índico
IQ	Iraq
IR	Irán
IS	Islandia
IT	Italia
JM	Jamaica
JO	Jordania
JP	Japón
KE	Kenya
KG	Kirguistán
KH	Camboya
KI	Kiribati
KM	Comoras
KN	Saint Kitts y Nevis
KP	Corea del Norte
KR	Corea del Sur
KW	Kuwait
KY	Islas Caimán
KZ	Kazajstán
LA	Laos
LB	Líbano
LC	Santa Lucía
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lituania
LU	Luxemburgo
LV	Letonia
LY	Libia
MA	Marruecos
MC	Mónaco
MD	Moldova

MG	Madagascar
MH	Islas Marshall
MK	Macedonia
ML	Malí
MIL	Organización militar
MM	Myanmar
MN	Mongolia
MO	Macao
MP	Islas Marianas del Norte
MQ	Martinica
MR	Mauritania
MS	Montserrat
MU	Isla Mauricio
MV	Maldivas
MW	Malawi
MX	México
MY	Malasia
MZ	Mozambique
NA	Namibia
NC	Nueva Caledonia
NE	Níger
NET	Organización con enlaces relacionados con Internet
NF	Isla Norfolk
NG	Nigeria
NI	Nicaragua
NL	Países Bajos
NO	Noruega
NP	Nepal
NR	Nauru
NT	Zona Neutral
NU	Isla Niue
NZ	Nueva Zelanda
OM	Omán
ORG	Organización no específica
PA	Panamá
PE	Perú
PF	Polinesia Francesa
PG	Papua Nueva Guinea
PH	Filipinas
PK	Pakistán
PL	Polonia
PM	San Pedro y Miquelón

PN	Isla Pitcairn
PR	Puerto Rico (USA)
PS	Territorios Palestinos
PT	Portugal
PY	Paraguay
PW	Palau
QA	Qatar
RE	Reunión
RO	Rumania
RU	Federación de Rusia
RW	Rwanda
SA	Arabia Saudita
SB	Islas Solomón
SC	Seychelles
SD	Sudán
SE	Suecia
SG	Singapur
SH	Santa Elena
SI	Eslovenia
SJ	Islas Svalbard y Jan Mayen
SK	República Eslovaca
SL	Sierra Leona
SM	San Marino
SN	Senegal
SO	Somalia
SR	Suriname
ST	Santo Tomé y Príncipe
SU	Unión Soviética
SV	El Salvador
SY	Siria
SZ	Swazilandia
TC	Islas Turcas y Caicos
TD	Chad
TF	Territorios Australes Franceses
TG	Togo
TH	Tailandia
TJ	Tayikistán
TK	Tokelau
TM	Turkmenistán
TN	Túnez
TO	Tonga
TP	Timor Oriental

TR	Turquía
TT	Trinidad y Tobago
TV	Tuvalu
TW	Taiwán
TZ	Tanzania
UA	Ucrania
UG	Uganda
UK	Reino Unido
UM	Islas Periféricas Menores de los Estados Unidos
US	Estados Unidos
UY	Uruguay
UZ	Uzbekistán
VA	Ciudad del Vaticano
VC	San Vicente y las Granadinas
VE	Venezuela
VG	Islas Vírgenes Británicas
VI	Islas Vírgenes de los Estados Unidos
VN	Vietnam
VU	Vanuatu
WF	Islas Wallis y Futuna
WS	Samoa Occidental
YE	Yemen
YT	Mayotte
YU	Yugoslavia
ZA	Sudáfrica
ZM	Zambia
ZR	Zaire
ZW	Zimbabwe

Tipos de servidores

Cada servidor con autoridad sobre una zona, mantiene una porción de la base de datos distribuida DNS y es posible definir conceptualmente a estos en función de como responde las peticiones:

- **Primarios o maestros:** Guardan los datos de un espacio de nombres en sus ficheros
- **Secundarios o esclavos:** Obtienen los datos de los servidores primarios a través de una transferencia de zona.
- **Locales o caché:** No contienen la base de datos para la resolución de nombres. Cuando se les realiza una query, estos a su vez consultan a los servidores DNS correspondientes, almacenando la respuesta en su base de datos para agilizar la repetición de estas peticiones en el futuro continuo o libre.

Instalación de BIND en Debian y CentOS

```
root@server:~# aptitude install bind9 bind9-doc dnsutils
```

```
[root@server]# yum install -y bind bind-chroot bind-libs bind-utils caching-nameserver
```

Configuración de BIND

Fichero named.conf

Este software mantiene un fichero de configuración denominado *named.conf* que se puede localizar en diferentes directorios (/etc/, /etc/bind/, ...) en función del sistema operativo que de soporte a este servidor. El archivo *named.conf* es una colección de declaraciones usando opciones anidadas rodeadas por caracteres de llaves, { }. Los administradores deben tener mucho cuidado cuando estén modificando *named.conf* para evitar errores sintácticos puesto que hasta el error más pequeño puede impedir que el servicio *named* arranque. Existe una utilidad denominada *named-checkconf* que permite verificar la sintaxis.

Los tipos de declaraciones más habituales son:

ACL: La sentencia *acl* (o sentencia de control de acceso) define grupos de hosts a los que se les puede permitir o negar el acceso al servidor de nombres.

Una declaración *acl* tiene la siguiente forma:

```
acl <acl-name> {
    <match-element>;
    [<match-element>; ...]
};
```

En esta declaración, sustituya *<acl-name>* con el nombre de la lista de control de acceso y reemplace *<match-element>* con una lista de direcciones IP separada por puntos y comas. La mayoría de las veces, una dirección IP individual o notación de red IP (tal como 10.0.1.0/24) es usada para identificar las direcciones IP dentro de la declaración *acl*.

La siguiente lista de control de acceso ya están definidas como palabras claves para simplificar la configuración:

- *any* — Hace coincidir todas las direcciones IP.
- *localhost* — Hace coincidir cualquier dirección IP que se use el sistema local.
- *localnets* — Hace coincidir cualquier dirección IP en cualquier red en la que el sistema local está conectado.
- *none* — No concuerda ninguna dirección IP.

OPTIONS: La declaración *options* define opciones de configuración de servidor globales y configura otras declaraciones por defecto. Puede ser usado para especificar la ubicación del directorio de trabajo *named*, los tipos de consulta permitidos y mucho más.

La declaración *options* toma la forma siguiente:

```
options {
    <option>;
    [<option>; ...]
};
```

En esta declaración, las directivas *<option>* son reemplazadas con una opción válida.

Las siguientes son opciones usadas a menudo:

- allow-query — Especifica cuáles hosts tienen permitido consultar este servidor de nombres. Por defecto, todos los hosts tienen derecho a consultar. Una lista de control de acceso, o una colección de direcciones IP o redes se puede usar aquí para sólo permitir a hosts particulares hacer consultas al servidor de nombres.
- allow-recursion — Parecida a la opción allow-query, salvo que se aplica a las peticiones recursivas. Por defecto, todos los hosts están autorizados a presentar peticiones recursivas en un servidor de nombres.
- blackhole — Especifica cuáles hosts no tienen permitido consultar al servidor de nombres.
- directory — Especifica el directorio de trabajo named si es diferente del valor predeterminado /var/named.
- forward — Especifica el comportamiento de reenvío de una directiva forwarders.

Se aceptan las siguientes opciones:

- first — Indica que los servidores de nombres especificados en la directiva forwarders sean consultados antes de que named intente resolver el nombre él mismo.
- only — Especifica que named no intente la resolución de nombres él mismo en el evento de que fallen las consultas a los servidores de nombres especificados en la directriz forwarders.
- forwarders — Especifica una lista de direcciones IP válidas para los servidores de nombres donde las peticiones se pueden reenviar para ser resueltas.
- listen-on — Especifica la interfaz de red en la cual named escucha por solicitudes. Por defecto, todas las interfaces son usadas.

ZONE: Una declaración zone define las características de una zona tal como la ubicación de su archivo de configuración y opciones específicas de la zona. Esta declaración puede ser usada para ignorar las declaraciones globales options.

Una declaración zone tiene la forma siguiente:

```
zone <zone-name> <zone-class> {  
    <zone-options>;  
    [<zone-options>; ...]  
};
```

En esta declaración, *<zone-name>* es el nombre de la zona, *<zone-class>* es la clase opcional de la zona, y *<zone-options>* es una lista de opciones que caracterizan la zona.

El atributo *<zone-name>* para la declaración de zona es particularmente importante, pues es el valor por defecto asignado para la directriz \$ORIGIN usada dentro del archivo de zona correspondiente localizado en el directorio /var/named/. El demonio named anexa el nombre de la zona a cualquier nombre de dominio que no esté completamente cualificado listado en el archivo de zona.

Por ejemplo, si una declaración zone define el espacio de nombres para example.com, utilice example.com como el *<zone-name>* para que sea colocado al final de los nombres de hosts dentro del archivo de zona example.com.

Las opciones más comunes para la declaración zone incluyen lo siguiente:

- allow-query — Especifica los clientes que se autorizan para pedir información sobre una zona. Por defecto, todas las peticiones de información son autorizadas.
- allow-transfer — Especifica los servidores esclavos que están autorizados para pedir una transferencia de información de la zona. Por defecto, todas las peticiones se autorizan.
- allow-update — Especifica los hosts que están autorizados para actualizar dinámicamente la información en sus zonas. Por defecto, no se autoriza la actualización de la información dinámicamente.
- file — Especifica el nombre del archivo en el directorio de trabajo named que contiene los datos de configuración de zona.
- masters — Especifica las direcciones IP desde las cuales solicitar información autorizada. Solamente se usa si la zona está definida como type slave.
- notify — Controla si named notifica a los servidores esclavos cuando una zona es actualizada. Esta directiva sólo acepta las opciones siguientes:
 - yes — Notifica a los servidores esclavos.
 - no — No notifica a los servidores esclavos.
 - explicit — Solamente notifica a los servidores esclavos especificados en una lista de also-notify dentro de la declaración de una zona.
- type — Define el tipo de zona.

Abajo se muestra una lista de las opciones válidas:

- delegation-only — Refuerza el estado de delegación de las zonas de infraestructura tales como COM, NET u ORG. Cualquier respuesta recibida sin una delegación explícita o implícita es tratada como NXDOMAIN. Esta opción solamente es aplicable en TLDs o en archivos raíz de zona en implementaciones recursivas o de caché.
- forward — Dice al servidor de nombres que lleve a cabo todas las peticiones de información de la zona en cuestión hacia otros servidores de nombres.
- hint — Tipo especial de zona que se usa para orientar hacia los servidores de nombres root que sirven para resolver peticiones de una zona que no se conoce. No se requiere mayor configuración que la establecida por defecto con una zona hint.
- master — Designa el servidor de nombres actual como el que tiene la autoridad para esa zona. Una zona se puede configurar como tipo master si los archivos de configuración de la zona residen en el sistema.
- slave — Designa el servidor de nombres como un servidor esclavo para esa zona. También especifica la dirección IP del servidor de nombres maestro para la zona.

COMENTARIOS: Líneas no procesadas. La siguiente es una lista de las etiquetas de comentarios válidas usadas dentro de named.conf:

- // — Cuando se coloca al comienzo de una línea, esa línea es ignorada por named.
- # — Cuando se coloca al comienzo de una línea, esa línea es ignorada por named.
- /* y */ — Cuando el texto se coloca entre estas etiquetas, se ignora el bloque de texto por named.

Existen más que se pueden consultar en los manuales de bind.

```
/etc/named.conf
```

```
options {
    DIRECTORY "/var/named";
};

acl "nets"{
    {192.168.1.0/24;};
};

zone "." in {
    type hint;
    file "named.root";
};

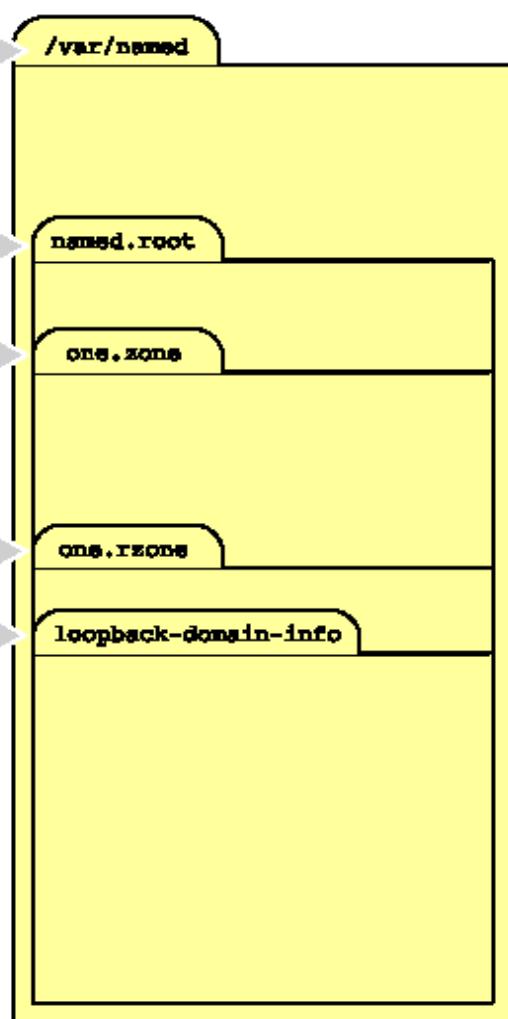
zone "one.edu" in {
    type master;
    file "one.zone";
};

allow-transfer {"nets";};

zone "1.168.192.in-addr.arpa" in {
    type master;
    file "one.rzone";
};

zone "127.in-addr.arpa" in {
    type master;
    file "loopback-domain-info";
};

/* This is a comment */
// This is a comment
# This is a comment
```



Fichero named.root

Este fichero especifica nombres y direcciones IP de los servidores raíz. La información de este fichero se describe como “hints” al proceso named. El proceso utilizará esta información cuando no sea capaz de resolver por sus propios medios.

Se puede descargar una actualización de este fichero desde la URL:

<ftp://ftp.rs.internic.net/domain/named.root>

Ficheros de zonas

Los *Archivos de zona* contienen información sobre un espacio de nombres particular y son almacenados en el directorio de trabajo named, por defecto /var/named/. Cada archivo de zona es nombrado de acuerdo a la opción file en la declaración zone, usualmente en una forma que relaciona al dominio en cuestión e identifica el archivo como contenido datos de zona, tal como example.com.zone.

Cada archivo de zona contiene *directivas* y *registros de recursos*. Las directivas le dicen al servidor de nombres que realice tareas o aplique configuraciones especiales a la zona. Los registros de recursos definen los parámetros de la zona y asignan identidades a hosts individuales. Las directivas son opcionales, pero los registros de recursos se requieren para proporcionar servicios de nombres a la zona.

Todas las directivas y registros de recursos deberían ir en sus propias líneas individuales.

Las directivas comienzan con el símbolo de dollar (\$) seguido del nombre de la directiva. Usualmente aparecen en la parte superior del archivo de zona.

Lo siguiente son directivas usadas a menudo:

- **\$INCLUDE** — Dice a named que incluya otro archivo de zona en el archivo de zona donde se usa la directiva. Así se pueden almacenar configuraciones de zona suplementarias aparte del archivo de zona principal.
- **\$ORIGIN** — Anexa el nombre del dominio a registros no cualificados, tales como aquellos con el nombre de host solamente.
- **\$TTL** — Ajusta el valor *Time to Live (TTL)* predeterminado para la zona. Este es el tiempo, en segundos, que un registro de recurso de zona es válido. Cada recurso puede contener su propio valor TTL, el cual ignora esta directiva.

Cuando se decide aumentar este valor, permite a los servidores de nombres remotos hacer caché a la información de zona para un período más largo de tiempo, reduciendo el número de consultas para la zona y alargando la cantidad de tiempo requerido para proliferar cambios de registros de recursos.

La base de datos (fichero de zona) contiene registros con información sobre los recursos del dominio que gestiona. Un registro contiene información de un dominio en particular. Cada registro tendrá una sintaxis especial. El formato común es:

[name] [ttl] class type data

Nombre (name): host o dominio, es decir, el “propietario” del registro.

Clase (class): identifica el protocolo o una instancia de protocolo. IN es la clase Internet system, o Sistema Internet en español.

Tipo (type): especifica el tipo de recurso, siendo los más comunes:

A: para asociar un nombre o host a una dirección IPv4. Del inglés Address (dirección).

AAAA: igual para direcciones IPv6.

MX: para identificar cuál host se encarga del servicio de correo. Del inglés Mail eXchange (intercambio de correo).

NS: para identificar los Servidores de Nombre autoritativos del dominio.

PTR: del inglés PoinTer Record, también llamado Reverse Record (registro DNS inverso), sirve para asociar una dirección IP a un nombre canónico, usando los cuatro octetos de la IP en sentido inverso seguido de “in-addr.arpa”.

TXT: registro en formato de texto, de uso libre y de carácter informativo.

SOA: identifica el comienzo de una zona autoritativa para el dominio. Este registro contiene información de la zona y se compone a su vez de una serie de parámetros. Del inglés Start of

Authority Record.

CNAME: identifica el nombre canónico de un alias, siendo un “nombre canónico” el nombre real de un host. Del inglés Canonical Name Record.

SRV: Permite indicar los servicios que ofrece el dominio. RFC 2782. Excepto Mx y Ns. Hay que incorporar el nombre del servicio, protocolo, dominio completo, prioridad del servicio, peso, puerto y el equipo completo.

Datos (data): contiene la respuesta o la definición del recurso (registro).

Time To Live: o TTL, es el tiempo de vida en segundos de un registro, es decir, el tiempo que se debe mantener en caché un registro dado antes de ser eliminado o renovado.

El siguiente fichero de Zona muestra algunos de los registros anteriores y el formato que deben tener:

```
$TTL 3D
@ IN SOA land-5.com. root.land-5.com. (
    199609206      ; serial, todays date + todays serial #
    8H             ; refresh, seconds
    2H             ; retry, seconds
    4W             ; expire, seconds
    1D )           ; minimum, seconds
    NS land-5.com.
    NS ns2.psi.net.
    MX 10 land-5.com. ; Primary Mail Exchanger
    TXT "LAND-5 Corporation"

localhost A 127.0.0.1
router A 206.6.177.1
land-5.com. A 206.6.177.2
ns A 206.6.177.3
www A 207.159.141.192

ftp CNAME land-5.com.
mail CNAME land-5.com.
news CNAME land-5.com.

funn A 206.6.177.2

;
;       Workstations
;
ws-177200 A 206.6.177.200
ws-177200 MX 10 land-5.com. ; Primary Mail Host
ws-177201 A 206.6.177.201
ws-177201 MX 10 land-5.com. ; Primary Mail Host
```

La sintaxis de los ficheros de zona se verifica con el comando *named-checkzone fichero [nombrezona]*

EJEMPLOS (ficheros named.conf)

Servidor de caché:

```
options {
    directory "/var/cache/bind";
```

```

forwarders {
    // OpenDNS servers
    208.67.222.222;
    208.67.220.220;
    // ADSL router
    192.168.1.1;
};

// Security options
listen-on port 53 { 127.0.0.1; 192.168.1.100; };
allow-query { 127.0.0.1; 192.168.1.0/24; };
allow-recursion { 127.0.0.1; 192.168.1.0/24; };
allow-transfer { none; };

};

```

Servidor maestro:

```

// Aquí van directivas de configuración como las analizadas anteriormente
//

zone "home.lan" {
    type master;
    file "/etc/bind/db.home.lan";
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.1.168.192";
};

```

Servidor esclavo:

```

options {
    DIRECTORY "/var/named";
};

zone "." in {
    type hint;
    file "named.root";
};

zone "127.in-addr.arpa" in {
    type master;
    file "loopback-domain-info";
};

zone "one.edu" in {
    type slave;
    file "one-backup";
    masters {
        192.168.1.1;
    };
};

zone "1.168.192.in-addr.arpa" in {
    type slave;
    file "one-rbackup";
    masters {
        192.168.1.1;
    };
};

```

110 SEGURIDAD

- 110.1 Tareas administrativas de seguridad.
- 110.2. Seguridad del host.
- 110.3 Protección de datos con encriptación.

110.1 Tareas administrativas de seguridad.

Peso en el examen de certificación: 3 puntos.

Objetivo: Revisar la configuración del sistema para garantizar la seguridad de conformidad con las políticas de seguridad establecidas.

Conceptos y áreas de conocimiento:

- Auditar el sistema buscando archivos con el suid/sgid bit activado.
- Establecer o cambiar las contraseñas de los usuarios así como la información de caducidad de las mismas.
- Utilizar nmap y netstat para descubrir puertos abiertos en un sistema.
- Establecer límites a los inicios de sesión de usuario, procesos y uso de memoria.
- Configuración de sudo básica y su uso.

Términos y utilidades:

- find
- passwd
- lsof
- nmap
- chage
- netstat
- sudo
- /etc/sudoers
- su
- usermod
- ulimit

Los candidatos deben saber cómo revisar la configuración del sistema para garantizar la seguridad de acogida de conformidad con las políticas de seguridad locales.

110.1.2. Auditar el sistema para encontrar archivos con el bit suid/sgid establecidos.

Habitualmente, los permisos de los archivos en Unix se corresponden con un número en octal que varía entre 000 y 777; sin embargo, existen unos permisos especiales que hacen variar ese número entre 0000 y 7777: se trata de los bits de permanencia (1000), SGID (2000) y SUID (4000).

El bit de SUID o setuid se activa sobre un fichero añadiéndole 4000 a la representación octal de los permisos del archivo y otorgándole además permiso de ejecución al propietario del mismo; al hacer esto, en lugar de la x en la primera terna de los permisos, aparecerá una s o una S si no hemos otorgado el permiso de ejecución correspondiente (en este caso el bit no tiene efecto):

```
# chmod 4777 /tmp/file1
# chmod 4444 /tmp/file2
# ls -l /tmp/file1
-rwsrwxrwx 1 root other 0 Feb 9 17:51 /tmp/file1*
# ls -l /tmp/file2
-r-Sr--r-- 1 root other 0 Feb 9 17:51 /tmp/file2*
#
```

El bit SUID activado sobre un fichero indica que todo aquél que ejecute el archivo va a tener durante la ejecución los mismos privilegios que quién lo creó; dicho de otra forma, si el administrador crea un fichero y lo setuida, todo aquel usuario que lo ejecute va a disponer, hasta que el programa finalice, de un nivel de privilegio total en el sistema. Podemos verlo con el siguiente ejemplo:

```
# cat testsuid.c

#include <stdio.h>
int main(void)
{ printf("UID: %d, EUID: %d\n", getuid(), geteuid()); }

# gcc -o testsuid testsuid.c
# chmod u+s testsuid
# ls -l testsuid
-rwsr-xr-x 1 root root 4305 Feb 10 02:34 testsuid

# su toni
$ id
uid=1000(toni) gid=100(users) groups=100(users)
$ ./testsuid
UID: 1000, EUID: 0
$
```

Podemos comprobar que el usuario toni, sin ningún privilegio especial en el sistema, cuando ejecuta nuestro programa setuidado de prueba está trabajando con un EUID (Effective UID) 0, lo que le otorga todo el poder del administrador (fíjémonos que éste último es el propietario del ejecutable); si en lugar de este código el ejecutable fuera una copia de un shell, el usuario toni tendría todos los privilegios del root mientras no finalice la ejecución, es decir, hasta que no se teclee exit en la línea de órdenes.

Todo lo comentado con respecto al bit setuid es aplicable al bit setgid pero a nivel de grupo del fichero en lugar de propietario: en lugar de trabajar con el EUID del propietario, todo usuario que ejecute un programa setgidado tendrá los privilegios del grupo al que pertenece el archivo. Para activar el bit de setgid sumaremos 2000 a la representación octal del permiso del fichero y además habremos de darle permiso de ejecución a la terna de grupo; si lo hacemos, la s o S aparecerá en lugar de la x en esta terna. Si el fichero es un directorio y no un archivo plano, el bit setgid afecta a los ficheros y subdirectorios que se creen en él: estos tendrán como grupo propietario al mismo que el directorio setgidado, siempre que el proceso que los cree pertenezca a dicho grupo.

Pero, ¿Cómo afecta todo esto a la seguridad del sistema? Muy sencillo: los bits de setuid y setgid dan a Unix una gran flexibilidad, pero constituyen al mismo tiempo la mayor fuente de ataques internos al sistema (entendiendo por ataques internos aquellos realizados por un usuario - autorizado o no - desde la propia máquina, generalmente con el objetivo de aumentar su nivel de privilegio en la misma). Cualquier sistema Unix tiene un cierto número de ejecutables setuidados y/o setgidados. Cada uno de ellos, se ejecuta con los privilegios de quien lo creó (generalmente el root u otro usuario con ciertos privilegios) lo que directamente implica que cualquier usuario tiene la capacidad de lanzar tareas que escapen total o parcialmente al control del sistema operativo: se ejecutan en modo privilegiado si es el administrador quien creó los ejecutables. Evidentemente, estas tareas han de estar controladas de una forma exhaustiva, ya que si una de ellas se comporta de forma anormal (un simple core dump) puede causar daños irreparables al sistema; tanto es así que hay innumerables documentos que definen, o lo intentan, pautas de programación considerada

`segura'. Si por cualquier motivo un programa setuidado falla se asume inmediatamente que presenta un problema de seguridad para la máquina, y se recomienda resetear el bit de setuid cuanto antes.

Está claro que asegurar completamente el comportamiento correcto de un programa es muy difícil, por no decir imposible; cada cierto tiempo suelen aparecer fallos (bugs) en ficheros setuidados de los diferentes clones de Unix que ponen en peligro la integridad del sistema. Entonces, ¿Por qué no se adopta una solución radical, como eliminar este tipo de archivos? Hay una sencilla razón: el riesgo que presentan no se corre inútilmente, para tentar al azar, sino que los archivos que se ejecutan con privilegios son estrictamente necesarios en Unix, al menos algunos de ellos. Veamos un ejemplo: un fichero setuidado clásico en cualquier clon es /bin/passwd, la orden para que los usuarios puedan cambiar su contraseña de entrada al sistema. No hace falta analizar con mucho detalle el funcionamiento de este programa para darse cuenta que una de sus funciones consiste en modificar el fichero de claves (/etc/passwd o /etc/shadow). Está claro que un usuario per se no tiene el nivel de privilegio necesario para hacer esto (incluso es posible que ni siquiera pueda leer el fichero de claves), por lo que frente a este problema tan simple existen varias soluciones: podemos asignar permiso de escritura para todo el mundo al fichero de contraseñas, podemos denegar a los usuarios el cambio de clave o podemos obligarles a pasar por la sala de operaciones cada vez que quieran cambiar su contraseña. Parece obvio que ninguna de ellas es apropiada para la seguridad del sistema (quizás la última lo sea, pero es impracticable en máquinas con un número de usuarios considerable). Por tanto, debemos asumir que el bit de setuid en /bin/passwd es imprescindible para un correcto funcionamiento del sistema. Sin embargo, esto no siempre sucede así: en un sistema Unix instalado out of the box el número de ficheros setuidados suele ser mayor de cincuenta; sin perjudicar al correcto funcionamiento de la máquina, este número se puede reducir a menos de cinco, lo que viene a indicar que una de las tareas de un administrador sobre un sistema recién instalado es minimizar el número de ficheros setuidados o setgidados. No obstante, tampoco es conveniente eliminarlos, sino simplemente resetear su bit de setuid mediante chmod:

```
# ls -l /bin/ping
-r-sr-xr-x 1 root bin 14064 May 10 1999 /bin/ping*
# chmod -s /bin/ping
# ls -l /bin/ping
-r-xr-xr-x 1 root bin 14064 May 10 1999 /bin/ping*
#
```

También hemos de estar atentos a nuevos ficheros de estas características que se localicen en la máquina; demasiadas aplicaciones de Unix se instalan por defecto con ejecutables setuidados cuando realmente este bit no es necesario, por lo que a la hora de instalar nuevo software o actualizar el existente hemos de acordarnos de resetear el bit de los ficheros que no lo necesiten. Especialmente grave es la aparición de archivos setuidados de los que el administrador no tenía constancia (ni son aplicaciones del sistema ni un aplicaciones añadidas), ya que esto casi en el 100% de los casos indica que nuestra máquina ha sido comprometida por un atacante. Para localizar los ficheros con alguno de estos bits activos, podemos ejecutar la siguiente orden:

```
# find / \(\ -perm -4000 -o -perm -2000 \) -type f -print
```

Por otra parte, el sticky bit o bit de permanencia se activa sumándole 1000 a la representación octal de los permisos de un determinado archivo y otorgándole además permiso de ejecución; si hacemos esto, veremos que en lugar de una x en la terna correspondiente al resto de usuarios aparece una t (si no le hemos dado permiso de ejecución al archivo, aparecerá una T):

```
# chmod 1777 /tmp/file1
```

```
# chmod 1774 /tmp/file2
# ls -l /tmp/file1
-rwxrwxrwt 1 root other 0 Feb 9 17:51 /tmp/file1*
# ls -l /tmp/file2
-rwxrwxr-T 1 root other 0 Feb 9 17:51 /tmp/file2*
#
```

Si el bit de permanencia de un fichero está activado (recordemos que si aparece una T no lo está) le estamos indicando al sistema operativo que se trata de un archivo muy utilizado, por lo que es conveniente que permanezca en memoria principal el mayor tiempo posible; esta opción se utilizaba en sistemas antiguos que disponían de muy poca RAM, pero hoy en día prácticamente no se utiliza. Lo que si sigue vigente es el efecto del sticky bit activado sobre un directorio: en este caso se indica al sistema operativo que, aunque los permisos 'normales' digan que cualquier usuario pueda crear y eliminar ficheros (por ejemplo, un 777 octal), sólo el propietario de cierto archivo y el administrador pueden borrar un archivo guardado en un directorio con estas características. Este bit, que sólo tiene efecto cuando es activado por el administrador (aunque cualquier usuario puede hacer que aparezca una t o una T en sus ficheros y directorios), se utiliza principalmente en directorios del sistema de ficheros en los que interesa que todos puedan escribir pero que no todos puedan borrar los datos escritos, como `/tmp/` o `/var/tmp/`: si el equivalente octal de los permisos de estos directorios fuera simplemente 777 en lugar de 1777, cualquier usuario podría borrar los ficheros del resto. Si pensamos que para evitar problemas podemos simplemente denegar la escritura en directorios como los anteriores también estamos equivocados: muchos programas - como compiladores, editores o gestores de correo - asumen que van a poder crear ficheros en `/tmp/` o `/var/tmp/`, de forma que si no se permite a los usuarios hacerlo no van a funcionar correctamente; por tanto, es muy recomendable para el buen funcionamiento del sistema que al menos el directorio `/tmp/` tenga el bit de permanencia activado.

Ya para finalizar, volvemos a lo que hemos comentado al principio de la sección: el equivalente octal de los permisos en Unix puede variar entre 0000 y 7777. Hemos visto que podíamos sumar 4000, 2000 o 1000 a los permisos 'normales' para activar respectivamente los bits setuid, setgid o sticky. Por supuesto, podemos activar varios de ellos a la vez simplemente sumando sus valores: en la situación poco probable de que necesitáramos todos los bits activos, sumaríamos 7000 a la terna octal 777. Si en lugar de especificar el valor octal de los permisos queremos utilizar la forma simbólica de chmod, utilizaremos +t para activar el bit de permanencia, g+s para activar el de setgid y u+s para hacer lo mismo con el de setuid; si queremos resetearlos, utilizamos un signo '-' en lugar de un '+' en la línea de órdenes.

110.1.3. Establecer o cambiar la contraseña de usuarios y la información de expiración de contraseñas.

El modelo de autenticación más básico consiste en decidir si un usuario es quien dice ser simplemente basándonos en una prueba de conocimiento que a priori sólo ese usuario puede superar; y desde Alí Babá y su 'Abrete, Sésamo' hasta los más modernos sistemas Unix, esa prueba de conocimiento no es más que una contraseña que en principio es secreta. Evidentemente, esta aproximación es la más vulnerable a todo tipo de ataques, pero también la más barata, por lo que se convierte en la técnica más utilizada en entornos que no precisan de una alta seguridad, como es el caso de los sistemas Unix en redes normales (y en general en todos los sistemas operativos en redes de seguridad media-baja); otros entornos en los que se suele aplicar este modelo de autenticación son las aplicaciones que requieren de alguna identificación de usuarios, como el software de cifrado PGP o el escáner de seguridad NESSUS. También se utiliza como complemento a otros mecanismos de autenticación, por ejemplo en el caso del Número de Identificación Personal (PIN) a

la hora de utilizar cajeros automáticos.

En todos los esquemas de autenticación basados en contraseñas se cumple el mismo protocolo: las entidades (generalmente dos) que participan en la autenticación acuerdan una clave, clave que han de mantener en secreto si desean que la autenticación sea fiable. Cuando una de las partes desea autenticarse ante otra se limita a mostrarle su conocimiento de esa clave común, y si ésta es correcta se otorga el acceso a un recurso. Lo habitual es que existan unos roles preestablecidos, con una entidad activa que desea autenticarse y otra pasiva que admite o rechaza a la anterior (en el modelo del acceso a sistemas Unix, tenemos al usuario y al sistema que le permite o niega la entrada).

Como hemos dicho, este esquema es muy frágil: basta con que una de las partes no mantenga la contraseña en secreto para que toda la seguridad del modelo se pierda; por ejemplo, si el usuario de una máquina Unix comparte su clave con un tercero, o si ese tercero consigue leerla y rompe su cifrado (por ejemplo, como veremos luego, mediante un ataque de diccionario), automáticamente esa persona puede autenticarse ante el sistema con éxito con la identidad de un usuario que no le corresponde.

En casi todas las implementaciones de Shadow Password actuales se suele incluir la implementación para otro mecanismo de protección de las claves denominado envejecimiento de contraseñas (Aging Password). La idea básica de este mecanismo es proteger los passwords de los usuarios dándoles un determinado periodo de vida: una contraseña sólo va a ser válida durante un cierto tiempo, pasado el cual expirará y el usuario deberá cambiarla.

Realmente, el envejecimiento previene más que problemas con las claves problemas con la transmisión de éstas por la red: cuando conectamos mediante mecanismos como telnet, ftp o rlogin a un sistema Unix, cualquier equipo entre el nuestro y el servidor puede leer los paquetes que enviamos por la red, incluyendo aquellos que contienen nuestro nombre de usuario y nuestra contraseña (hablaremos de esto más a fondo en los capítulos dedicados a la seguridad del sistema de red y a la criptografía); de esta forma, un atacante situado en un ordenador intermedio puede obtener muy fácilmente nuestro login y nuestro password. Si la clave capturada es válida indefinidamente, esa persona tiene un acceso asegurado al servidor en el momento que quiera; sin embargo, si la clave tiene un periodo de vida, el atacante sólo podrá utilizarla antes de que el sistema nos obligue a cambiarla.

A primera vista, puede parecer que la utilidad del envejecimiento de contraseñas no es muy grande; al fin y al cabo, la lectura de paquetes destinados a otros equipos (sniffing) no se hace por casualidad: el atacante que lea la red en busca de claves y nombres de usuario lo va a hacer porque quiere utilizar estos datos contra un sistema. Sin embargo, una práctica habitual es dejar programas escuchando durante días y grabando la información leída en ficheros; cada cierto tiempo el pirata consultará los resultados de tales programas, y si la clave leída ya ha expirado y su propietario la ha cambiado por otra, el haberla capturado no le servirá de nada a ese atacante.

Los periodos de espiración de las claves se suelen definir a la hora de crear a los usuarios con las herramientas que cada sistema ofrece para ello. Si queremos modificar alguno de estos períodos una vez establecidos, desde esas mismas herramientas de administración podremos hacerlo, y también desde línea de órdenes mediante órdenes como chage o usermod. Como antes hemos dicho, en el archivo `/etc/shadow` se almacena, junto a la clave cifrada de cada usuario, la información necesaria para implementar el envejecimiento de contraseñas; una entrada de este archivo es de la forma

```
toni:LEgPN8jqSCHCg:10322:0:99999:7:::
```

Tras el login y el password de cada usuario se guardan los campos siguientes:

- Días transcurridos desde el 1 de enero de 1970 hasta que la clave se cambió por última vez.

- Días que han de transcurrir antes de que el usuario pueda volver a cambiar su contraseña.
- Días tras los cuales se ha de cambiar la clave.
- Días durante los que el usuario será avisado de que su clave va a expirar antes de que ésta lo haga.
- Días que la cuenta estará habilitada tras la expiración de la clave.
- Días desde el 1 de enero de 1970 hasta que la cuenta se deshabilite.
- Campo reservado.

Como podemos ver, cuando un usuario cambia su clave el sistema le impide volverla a cambiar durante un periodo de tiempo; con esto se consigue que cuando el sistema obligue a cambiar la contraseña el usuario no restaure inmediatamente su clave antigua (en este caso el esquema no serviría de nada). Cuando este periodo finaliza, suele existir un intervalo de cambio voluntario: está permitido el cambio de contraseña, aunque no es obligatorio; al finalizar este nuevo periodo, el password ha expirado y ya es obligatorio cambiar la clave. Si el número máximo de días en los que el usuario no puede cambiar su contraseña es mayor que el número de días tras los cuales es obligatorio el cambio, el usuario no puede cambiar nunca su clave. Si tras el periodo de cambio obligatorio el password permanece inalterado, la cuenta se bloquea.

110.1.4. Ser capaz de usar nmap y netstat para descubrir puertos abiertos en el sistema.

Tras configurar los servicios de red, es importante prestarle atención a los puertos que actualmente están escuchando en las interfaces de red del sistema. Cualquier puerto abierto puede ser la evidencia de una intrusión.

Existen dos maneras fundamentales para listar los puertos que están abiertos en la red. La menos confiable consiste en consultar los paquetes en la red utilizando comandos como netstat -an o lsof -i. Este método es menos confiable debido a que estos programas no se conectan a la máquina desde la red, sino que verifican qué es lo que se está ejecutando en el sistema. Por esta razón, estas aplicaciones frecuentemente son reemplazadas por atacantes. Alguien que quiera ocultar el rastro que está dejando al ingresar, o al abrir sin autorización los puertos de un sistema, intentará reemplazar netstat y lsof, con sus versiones personales y modificadas.

Una forma más confiable de verificar los puertos que están escuchando en una red, es mediante la utilización de un escáner de puertos como nmap. El siguiente comando ejecutado desde una terminal, especifica los puertos que se encuentran abiertos a conexiones TCP desde la red:

```
#nmap -sT -O localhost
```

La salida de este comando es la siguiente:

```
Starting Nmap 4.68 ( http://nmap.org ) at 2009-03-06 12:08 EST
Interesting ports on localhost.localdomain (127.0.0.1):
Not shown: 1711 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
```

```
834/tcp open unknown
2601/tcp open zebra
32774/tcp open sometimes-rpc11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.24
Uptime: 4.122 days (since Mon Mar 2 09:12:31 2009)
Network Distance: 0 hops
OS detection performed. Please report any incorrect results at
http://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 1.420 seconds
```

Esta salida muestra que el sistema está ejecutando portmap debido a la presencia del servicio sunrpc. Sin embargo, existe además un servicio misterioso en el puerto 834. Para verificar si el puerto está asociado con la lista oficial de servicios conocidos, se puede escribir el siguiente comando:

```
#cat /etc/services | grep 834
```

Este comando no devuelve ninguna información. Lo que está indicando es que si bien el puerto se encuentra dentro del rango reservado (es decir, entre 0 y 1023), y que no necesita privilegios de usuario root para abrirse, sin embargo no está asociado con ningún servicio conocido.

A continuación, podemos verificar si existe información acerca del puerto utilizando netstat o lsof. Para verificar el puerto 834 utilizando netstat, se utiliza el siguiente comando:

```
#netstat -anp | grep 834
```

El comando devuelve la siguiente salida:

```
tcp 0 0 0.0.0.0:834 0.0.0.0:* LISTEN 653/ypbind
```

La presencia de un puerto abierto en netstat es un reaseguro, ya que si un atacante ha abierto un puerto en un sistema en el que no está autorizado a ingresar, seguramente no permitirá que sea detectada su presencia mediante este comando. Además, la opción [p] revela el proceso ID (PID) del servicio que ha abierto el puerto. En este caso, el puerto abierto pertenece a ypbnd (NIS), que es un servicio RPC administrado conjuntamente con el servicio portmap.

El comando lsof muestra información similar a netstat, ya que también es capaz de enlazar puertos con servicios:

```
#lsof -i | grep 834
```

La sección que nos interesa de la salida de este comando es la siguiente:

```
ypbind 653 0 7u IPv4 1319 TCP *:834 (LISTEN)
ypbind 655 0 7u IPv4 1319 TCP *:834 (LISTEN)
ypbind 656 0 7u IPv4 1319 TCP *:834 (LISTEN)
ypbind 657 0 7u IPv4 1319 TCP *:834 (LISTEN)
```

Estas herramientas nos dicen mucho acerca del estado en que se encuentran los servicios en ejecución de una máquina. Estas herramientas son flexibles y pueden ofrecer una importante cantidad de información acerca de los servicios de red y sus configuraciones. Para obtener más información, vea las páginas man de lsof, netstat, nmap, y services.

110.1.5. Establecer límites en el login de usuario, procesos y uso de memoria.

En un sistema Linux es posible controlar ciertos parámetros referentes al acceso de los usuarios a través de telnet o r- mediante el fichero `/etc/login.defs`. Como siempre, es necesario insistir en la necesidad de sustituir todos los protocolos en claro por equivalentes cifrados, con lo que ni telnet ni r- deberían existir como servicio en una maquina Unix, pero de cualquier forma vamos a comentar algunas directivas del fichero anterior que pueden resultar interesantes para nuestra seguridad, tanto si afectan a las conexiones remotas como si no:

- **FAIL_DELAY:** Retardo desde que se introduce un nombre de usuario o contraseña incorrectos hasta que se vuelve a solicitar el login de entrada al sistema.
- **LOGIN_RETRIES:** Número máximo de intentos antes de que se cierre la conexión.
- **LOGIN_TIMEOUT:** Tiempo máximo durante el que se permite la entrada antes de que se cierre la conexión.

Una vez iniciada la sesión, la posibilidad de que usuarios válidos del sistema puedan producir problemas por el consumo excesivo de recursos existe. Este problema puede ser accidental, pero si sucede de forma intencionada se denomina ataque de Denegación de Servicio (DoS). Por ejemplo, un usuario puede crear procesos que se repliquen a sí mismos y que nunca terminen. El sistema podría caerse ya que tendría que gestionar cientos de procesos que intentan clonarse a sí mismos. También podría darse el caso de un proceso que solicita memoria hasta que no hay más disponible.

Existe una herramienta interna a bash llamada ulimit con la que podemos especificar el límite máximo de cada uno de los recursos que el sistema pone a disposición de los usuarios. Para que estos límites se apliquen a todos los usuarios, hay que indicar los comandos ulimit en el fichero `/etc/profile`.

En los sistemas UNIX/LINUX existe la posibilidad de limitar recursos a los usuarios o grupos, por ejemplo, el máximo número de logins que puede realizar simultáneamente un usuario, el máximo tiempo de CPU, el máximo número de procesos etc. Estos límites se controlan en LINUX a través del fichero `/etc/security/limits.conf`. También es posible limitar los tiempos de acceso a los usuarios. Una de las formas de hacerlo es con el servicio **timeoutd**. Este servicio se instala a través de la distribución y, una vez instalado aparece un fichero de configuración `/etc/timeouts`.

110.1.6. Uso y configuración básica de sudo

El comando sudo (superuser do) permite a un administrador dar a usuarios regulares o grupos privilegios para ejecutar determinados comandos, guardando logs de lo ejecutado .

Para utilizar sudo se añade sudo al comando que se quiere ejecutar el sistema pregunta el password del usuario (no hay que saber el de root) , confirma que estás autorizado a ejecutar el comando, lo ejecuta y guarda log de lo hecho por defecto, el password se “cachea” un tiempo predeterminado durante el cual no es necesario volverlo a dar .

¿Quién está interesado en utilizar este comando?

- Todas aquellas personas que quieren minimizar el uso tanto del password de root como de un shell de root en una máquina en una sesión normal de trabajo
- Aquellas personas que deseen brindar acceso a comandos restringidos solo para root con el fin de realizar labores de mantenimiento a otros usuarios
- En ciertas configuraciones puede ser un remplazo efectivo del suid en los permisos de los binarios o scripts que vamos a ejecutar.

Sudo se configura mediante el fichero `/etc/sudoers` (existe la utilidad visudo para editar el fichero controlando la sintaxis) y existen tres tipos de entradas:

- Condiciones por defecto: variables para ajustar el comportamiento de sudo
- Aliases: variables para agrupar la información
- Especificaciones de usuario: indican quién puede ejecutar qué

Éste sería un ejemplo de fichero:

```
# Condiciones: ver man sudoers
Defaults env_reset, timestamp_timeout=0, insults, lecture=always
Defaults logfile=/var/log/sudolog
# Alias para hosts
Host_Alias SERVERS=servidor
Host_Alias CLIENTS=192.168.0.0/24, !SERVERS # Toda la red menos SERVERS
# Alias para usuarios
User_Alias AYUDANTES=tomas,pepe
User_Alias ADMINS=%wheel # Grupo wheel
# Alias para comandos
Cmnd_Alias SHUTDOWN=/sbin/shutdown -h now, /sbin/halt
Cmnd_Alias USERS=/usr/sbin/adduser, /usr/sbin/deluser
# Especificaciones de usuarios
root ALL=(ALL) ALL
# ADMINS pueden hacer todo como root
ADMINS ALL=ALL
# AYUDANTES pueden apagar los clientes sin contraseña
AYUDANTES ALL,!SERVERS=NOPASSWD: SHUTDOWN
# AYUDANTES pueden añadir usuarios en SERVERS
AYUDANTES SERVERS=USERS
# tomas puede usar dump y restore como usuario operator
tomas ALL=(operator) /sbin/dump, (operator) /sbin/restore
```

110.1.7. Archivos, términos y utilidades

110.1.7.1. find

El comando `find` recorre el árbol de directorio cuya raíz reside en cada nombre de fichero dado, evaluando de izquierda a derecha la expresión especificada, según las reglas de precedencia, hasta que se conoce el resultado (la parte izquierda es falsa para operaciones `and`, verdadera para `or`), en cuyo punto `find` se mueve al siguiente nombre de fichero.

OPCIONES

Aquí se muestran las opciones más utilizadas para la búsqueda de archivos con los bits `s` y `t` activados:

- perm modo: Los bits de permiso del fichero son exactamente modo (octal o simbólico). Los modos simbólicos utilizarán 0 como punto de partida.
- perm -modo: Todos los bits de permiso modo están activos para el fichero.
- perm +modo: Cualquiera de los bits de permiso de modo está activo para el fichero.
- type c: El fichero es de tipo c:

b especial de bloques
c especial de caracteres
d directorio
p tubería con nombre (FIFO)
f fichero regular
l enlace simbólico
s zócalo (socket)

-uid n El UID numérico del propietario del fichero es n.

-print: Imprime el nombre completo del fichero en la salida estándar, seguido por un salto de línea.

110.1.7.2. passwd

Cambia la contraseña del usuario

Uso

```
#passwd [-x max] [-n min] [-w warn] [-i inact] nombre  
#passwd {-l | -u} nombre
```

Descripción

Cambia la información de autenticación de cuentas de usuarios y grupos, incluyendo claves y detalles de caducidad de las claves, y puede ser utilizado para habilitar y deshabilitar cuentas. Solamente un usuario con los suficientes permisos puede cambiar la clave de otros usuarios o modificar la información de caducidad.

Opciones

-x max: Establece el número de días que una clave es válida.
-n min: Establece el número de días mínimos antes de poder cambiar una contraseña.
-w warn: Establece el número de días en los que se le avisa al usuario antes de que la contraseña caduque.
-i inactive: Deshabilita una cuenta después de que una cuenta lleve caducada el número de días indicados.
-l: Deshabilita una cuenta cambiando la contraseña a una valor que nunca puede ser obtenido una vez aplicado el algoritmo de cifrado.
-u: Rehabilita una cuenta cambiando la clave a su estado anterior.

110.1.7.3. lsof

lsof (Lista de archivos abiertos, en español) es una conocida herramienta de monitorización de sistemas operativos tipo Unix que se utiliza para mostrar todos los archivos de disco que mantiene abiertos un determinado proceso (PID), incluyendo los sockets de red abiertos, tuberías (pipes). lsof es software libre.

Ejemplo :

```
$ lsof -i -n -P | grep java  
java 11819 lyonn 34u IPv6 52669 TCP *:1095 (LISTEN)  
java 11819 lyonn 37u IPv6 52671 TCP *:53872 (LISTEN)  
java 11819 lyonn 39u IPv6 52732 TCP 10.225.183.218:41525->10.225.183.218:35812 (ESTABLISHED)  
java 11880 lyonn 34u IPv6 52743 TCP *:1096 (LISTEN)  
java 11880 lyonn 37u IPv6 52745 TCP *:33029 (LISTEN)
```

```
java 11880 lyonn 39u IPv6 52757 TCP 10.225.183.218:41530 -
>10.225.18
```

Los parámetros más utilizados son:

- i Lists IP sockets.
- n Do not resolve hostnames (no DNS).
- P Do not resolve port names (lista el número de puerto en lugar del nombre).

Si no se indica ninguna opción restrictiva o parámetro lsof enumera todos los archivos abiertos en ese momento, que normalmente suelen ser bastantes.

110.1.7.4. nmap

Nmap (“mapeador de redes”) es una herramienta de código abierto para exploración de red y auditoría de seguridad. Se diseñó para analizar rápidamente grandes redes, aunque funciona muy bien contra equipos individuales. Nmap determina qué equipos se encuentran disponibles en una red, qué servicios (nombre y versión de la aplicación) ofrecen, qué sistemas operativos (y sus versiones) ejecutan, qué tipo de filtros de paquetes o cortafuegos se están utilizando así como docenas de otras características. Aunque generalmente se utiliza Nmap en auditorías de seguridad, muchos administradores de redes y sistemas lo encuentran útil para realizar tareas rutinarias, como puede ser el inventariado de la red, la planificación de actualización de servicios y la monitorización del tiempo que los equipos o servicios se mantiene activos.

Uso:

```
#nmap [Tipo(s) de Análisis] [Opciones] {especificación de objetivos}
```

ESPECIFICACIÓN DE OBJETIVO:

Se pueden indicar nombres de sistema, direcciones IP, redes, etc. Ej: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

- iL <archivo_entrada>: Lee una lista de sistemas/redes del archivo.
- iR <número de sistemas>: Selecciona objetivos al azar
- exclude <sist1[,sist2][,sist3],...>: Excluye ciertos sistemas o redes
- excludefile <fichero_exclusión>: Excluye los sistemas indicados en el fichero

DESCUBRIMIENTO DE HOSTS:

- sL: Sondeo de lista - Simplemente lista los objetivos a analizar
 - sP: Sondeo Ping - Sólo determina si el objetivo está vivo
 - P0: Asume que todos los objetivos están vivos
 - PS/PA/PU [listadepuertos]: Análisis TCP SYN, ACK o UDP de los puertos indicados
 - PE/PP/PM: Solicita un análisis ICMP del tipo echo, marca de fecha y máscara de red
 - n/-R: No hacer resolución DNS / Siempre resolver [por omisión: a veces]
 - dns-servers <serv1[,serv2],...>: Especificar servidores DNS específicos
 - system-dns: Utilizar la resolución del sistema operativo

TÉCNICAS DE ANÁLISIS:

- sS/sT/sA/sW/sM: Análisis TCP SYN/Connect()/ACK/Window/Maimon
- sN/sF/sX: Análisis TCP Null, FIN, y Xmas

--scanflags <indicador>: Personalizar los indicadores TCP a utilizar
-sI <sistema zombi[:puerto_sonda]>: Análisis pasivo («Idle», N. del T.)
-sO: Análisis de protocolo IP
-b <servidor ftp rebote>: Análisis por rebote FTP

ESPECIFICACIÓN DE PUERTOS Y ORDEN DE ANÁLISIS:

-p <rango de puertos>: Sólo sondear los puertos indicados. Ej: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080
-F: Rápido - Analizar sólo los puertos listados en el archivo nmap-services
-r: Analizar los puertos secuencialmente, no al azar.

DETECCIÓN DE SERVICIO/VERSIÓN:

-sV: Sondear puertos abiertos, para obtener información de servicio/versión
--version-intensity <nivel>: Fijar de 0 (ligero) a 9 (probar todas las sondas)
--version-light: Limitar a las sondas más probables (intensidad 2)
--version-all: Utilizar todas las sondas (intensidad 9)
--version-trace: Presentar actividad detallada del análisis (para depurar)

DETECCIÓN DE SISTEMA OPERATIVO

-O: Activar la detección de sistema operativo (SO)
--osscan-limit: Limitar la detección de SO a objetivos prometedores
--osscan-guess: Adivinar el SO de la forma más agresiva

TEMPORIZADO Y RENDIMIENTO:

-T[0-5]: Seleccionar plantilla de temporizado (los números altos son más rápidos)
--min-hostgroup/max-hostgroup <tamaño>: Paralelizar los sondeos
--min-parallelism/max-parallelism <msegs>: Parallelización de sondeos
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <msegs>: Indica el tiempo de ida y vuelta de la sonda
--max-retries <reintentos>: Limita el número máximo de retransmisiones de las sondas de análisis de puertos
--host-timeout <msegs>: Abandonar un objetivo pasado este tiempo
--scan-delay/--max-scan-delay <msegs>: Ajusta el retraso entre sondas

EVASIÓN Y FALSIFICACIÓN PARA CORTAFUEGOS/IDS:

-f; --mtu <valor>: fragmentar paquetes (opc. con el MTU indicado)
-D <señuelo1,señuelo2,...>: Disimular el análisis con señuelos.
-S <Dirección_IP>: Falsificar la dirección IP origen
-e <interfaz>: Utilizar la interfaz indicada
-g/--source-port <numpuerto>: Utilizar el número de puerto dado
--data-length <num>: Agregar datos al azar a los paquetes enviados
--ttl <val>: Fijar el valor del campo time-to-live (TTL) de IP
--spoof-mac <dirección mac/prefijo/nombre de fabricante>: Falsificar la dirección MAC
--badsum: Enviar paquetes con una suma de comprobación TCP/UDP falsa

SALIDA:

-oN/-oX/-oS/-oG <file>: Guardar el sondeo en formato normal, XML, s|<rIpt kIddi3 (n3n3b4n4n4), y Grepeable (para usar con grep(1)), respectivamente, al archivo indicado.
-oA <nombre_base>: Guardar en los tres formatos principales al mismo tiempo
-v: Aumentar el nivel de mensajes detallados (-vv para aumentar el efecto)
-d[nivel]: Fijar o incrementar el nivel de depuración (Tiene sentido hasta 9)
--packet-trace: Mostrar todos los paquetes enviados y recibidos
--iflist: Mostrar interfaces y rutas (para depurar)
--append-output: Agregar, en vez de sobreescibir, a los archivos indicados con -o.
--resume <archivo>: Retomar un análisis abortado/detenido
--stylesheet <ruta/URL>: Convertir la salida XML a HTML según la hoja de estilo XSL indicada
--webxml: Referenciar a la hoja de estilo de Insecure.Org para tener un XML más portable
--no_stylesheet: No asociar la salida XML con ninguna hoja de estilos XSL

MISCELÁNEO:

-6: Habilitar análisis IPv6
-A: Habilita la detección de SO y de versión
--datadir <nombreDir>: Indicar la ubicación de los archivos de datos Nmap personalizados.
--send-eth/--send-ip: Enviar paquetes utilizando tramas Ethernet o paquetes IP "raw"
--privileged: Asumir que el usuario tiene todos los privilegios
-V: Muestra el número de versión
-h: Muestra esta página resumen de la ayuda.

EJEMPLOS:

```
nmap -v -A scanme.nmap.org
nmap -v -sP 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -P0 -p 80
```

110.1.5. chage

chage – cambia la información de caducidad de la contraseña

Sinopsis

#chage [opciones] [LOGIN]

Descripción

El comando chage comando el número de días entre cambios de contraseña y la fecha del último cambio de contraseña. Esta información es utilizada por el sistema para determinar cuando un usuario debe cambiar su contraseña.

Opciones

Las opciones que se aplican a la chage comando son:

-d , - lastday LAST_DAY. Establecer el número de días desde 1 de enero de 1970 que se cambió la contraseña anterior. La fecha también se puede expresar en el formato AAAA-MM-DD.
-E , - EXPIREDATE EXPIRE_DATE. Establecer la fecha o el número de días desde Enero 1 de 1970 en el que la cuenta del usuario ya no será accesible. La fecha también se puede expresar en el

formato AAAA-MM-DD. Un usuario cuya cuenta está bloqueada debe contactar con el administrador del sistema antes de poder utilizar el sistema de nuevo. Pasando el número -1 como el EXPIRE_DATE eliminará una fecha de caducidad de la cuenta.

-H , - help. Mostrar mensaje de ayuda y salir.

-I , - inactive INACTIVO. Establecer el número de días de inactividad después de una contraseña ha caducado antes de que la cuenta está bloqueada. La opción INACTIVO es el número de días de inactividad. Un usuario cuya cuenta está bloqueada debe contactar con el administrador del sistema antes de poder utilizar el sistema de nuevo. Pasando el número -1 como INACTIVO eliminará la inactividad de una cuenta.

-l , - list. Mostrar información de envejecimiento de la clave.

-m , - mindays MIN_DAYS. Establecer el número mínimo de días entre cambios de contraseña MIN_DAYS . Un valor de cero en este campo indica que el usuario puede cambiar su / su contraseña en cualquier momento.

-M , - maxdays MAX_DAYS. Establecer el número máximo de días que una contraseña es válida. Cuando MAX_DAYS más LAST_DAY es menor que el día de hoy, el usuario tendrá que cambiar su / su contraseña antes de poder usar su / su cuenta. Este hecho se puede planificar por adelantado por el uso de la W- opción, lo que proporciona al usuario una advertencia anticipada. Pasando el número -1 como MAX_DAYS eliminará la comprobación de validez de una contraseña.

-W , - warndays WARN_DAYS. Establecer el número de días de advertencia antes de un cambio de contraseña. La opción WARN_DAYS es el número de días antes de la expiración de la contraseña que el usuario será advertido de su / su contraseña está a punto de expirar.

Si ninguna de las opciones, chage funciona de manera interactiva, preguntándole al usuario todos los valores mostrándole su valor actual. El usuario introducirá el nuevo valor para cambiar el campo, o dejará en blanco la línea a utilizar el valor actual. El valor actual se muestra entre un par de corchetes.

110.1.7.6. netstat

Muestra conexiones de red, tablas de encaminamiento, estadísticas de interfaces, conexiones enmascaradas y mensajes del tipo netlink.

Opciones

(sin opciones): Se puede ver el estado de las conexiones de red al listar los conectores (sockets) abiertos.

-e: Se puede obtener información adicional (userid, identificador de usuario).

-v: Muestra las familias de direcciones conocidas no soportadas por el núcleo.

-o: Muestra información adicional sobre los temporizadores de red.

-a: Muestra todos los conectores, incluyendo los conectores a la escucha en el servidor.

-r, --route: Se obtienen las tablas de encaminamiento del núcleo en el mismo formato que usa route -e.

-i, --interface iface: Se mostrará una tabla de todos los interfaces (o del iface especificado).

-M, --masquerade: Se puede ver también una lista de todas las sesiones enmascaradas. Con la opción -e se puede incluir información referente a la numeración de secuencias.

-N, --netlink: Las versiones más recientes del núcleo disponen de un canal de comunicación entre el núcleo y el usuario llamado netlink. Se pueden obtener de /dev/route mensajes sobre la creación o destrucción de rutas o interfaces (36,0).

-n, --numeric: Muestra direcciones numéricas en vez de tratar de determinar un ordenador, puerto o nombre de usuario simbólicos.

-A, --af family: Usa un método diferente para establecer las familias de direcciones. family es una lista de palabras referentes a familias de direcciones separadas por comas (',') como inet, unix, ipx, ax25, netrom y ddp. Esto tiene el mismo efecto que usar las opciones largas --inet, --unix, --ipx, --ax25, --netrom y --ddp.

-c, --continuous: Esta opción hace que netstat muestre la tabla seleccionada en pantalla continuamente cada segundo hasta que el usuario lo interrumpa.

110.1.7.7. sudo

sudo (SUperuser DO) lo ejecuta un usuario normal, al que se supone tiene permisos para ejecutar cierto comando. Entonces, sudo requiere que los usuarios se autentifiquen a si mismos a través de su contraseña para permitirles la ejecución del comando. Veamos un ejemplo:

```
$ sudo /sbin/ifconfig  
Password:  
eth0 Link encap:Ethernet HWaddr 4C:00:10:60:5F:21  
inet addr:200.13.110.62 Bcast:200.13.110.255 Mask:255.255.255.0  
inet6 addr: fe80::4e00:10ff:fe60:5f21/64 Scope:Link
```

Como se podrá observar se usa el comando sudo seguido del comando (con toda su ruta si es que este no esta en el PATH del usuario) al que se tiene permiso. sudo pregunta por la contraseña del usuario que ejecuta el comando y listo.

Por defecto, después de hacer lo anterior se tendrán 5 minutos para volver a usar el mismo comando u otros a los que se tuviera derecho, sin necesidad de ingresar la contraseña de nuevo. Si se quiere extender el tiempo por otros 5 minutos se puede usar la opción sudo -v (validate). Por el contrario, si ya se terminó lo que se tenía que hacer, se puede usar sudo -k (kill) para terminar con el tiempo de gracia de validación.

Ahora bien, ¿Qué comandos son los que puedo utilizar?, pues la opción -l es la indicada para eso:

```
$ sudo -l  
User sergio may run the following commands on this host:  
(root) /sbin/ifconfig  
(root) /sbin/lspci
```

En el caso anterior se ejecutó un comando de root, pero no tiene que ser así, también es posible ejecutar comandos de otros usuarios del sistema indicando la opción -u:

```
$ sudo -u ana /comando/de/ana
```

Una de las opciones más interesantes es la que permite editar archivos de texto de root (claro, con el permiso otorgado en 'sudoers' como se verá más adelante), y esto se logra con la opción -e, esta opción está ligada a otro comando de sudo llamado sudoedit que invoca al editor por defecto del usuario, que generalmente es 'vi'.

```
$ sudo -e /etc/inittab
```

(Permitira modificar el archivo indicado como si se fuera root)

Cuando se configura sudo se tienen múltiples opciones que se pueden establecer, estas se consultan a través de la opción -L

```
$ sudo -L
Available options in a sudoers ``Defaults'' line:

syslog: Syslog facility if syslog is being used for logging
syslog_goodpri: Syslog priority to use when user authenticates
successfully
syslog_badpri: Syslog priority to use when user authenticates
unsuccessfully
long_otp_prompt: Put OTP prompt on its own line
ignore_dot: Ignore '.' in $PATH
mail_always: Always send mail when sudo is run
mail_badpass: Send mail if user authentication fails
mail_no_user: Send mail if the user is not in sudoers
mail_no_host: Send mail if the user is not in sudoers for this
host
mail_no_perms: Send mail if the user is not allowed to run a
command
tty_tickets: Use a separate timestamp for each user/tty combo
lecture: Lecture user the first time they run sudo
lecture_file: File containing the sudo lecture
authenticate: Require users to authenticate by default
root_sudo: Root may run sudo
```

Una de las opciones más importantes de consulta es -V, que permite listar las opciones (defaults) establecidas por defecto para sudo todos los usuarios, comandos, equipos, etc. Más adelante en este tutorial, aprenderemos como establecer opciones específicas para ciertos usuarios, comandos o equipos. NOTA: tienes que ser 'root' para usar esta opción.

```
# sudo -V
Sudo version 1.6.9p5

Sudoers path: /etc/sudoers
Authentication methods: 'pam'
Syslog facility if syslog is being used for logging: local2
Syslog priority to use when user authenticates successfully:
notice
Syslog priority to use when user authenticates unsuccessfully:
alert
Send mail if the user is not in sudoers
Lecture user the first time they run sudo
Require users to authenticate by default
Root may run sudo
Log the hostname in the (non-syslog) log file
Allow some information gathering to give useful error messages
Visudo will honor the EDITOR environment variable
Set the LOGNAME and USER environment variables
Reset the environment to a default set of variables
Length at which to wrap log file lines (0 for no wrap): 80
Authentication timestamp timeout: 5 minutes
Password prompt timeout: 5 minutes
Number of tries to enter a password: 3
```

```
Umask to use or 0777 to use user's: 022
Path to log file: /var/log/sudo.log
```

110.1.7.8. /etc/sudoers

El /etc/sudoers se divide en tres grandes secciones:

```
# Definiciones de alias
#
# Ajuste de opciones por defecto
#
# Reglas de acceso
#
```

Todas son opcionales. Obviamente, la más necesaria es la última ya que sin ésta el uso de sudo no tiene sentido. Como se puede observar, los comentarios se insertan igual que en los scripts del shell.

Definiciones de alias

Los alias son abreviaciones para cualquier tipo de elemento: comandos, usuarios, usuarios privilegiados y hosts. Éstos alias pueden ser utilizados en cualquier lugar donde se espere un comando, un usuario privilegiado o un host respectivamente. Insisto, cualquier lugar; inclúida la definición de un alias.

```
Tipo_Alias NOMBRE_ALIAS1 = elemento1, elemento2, elemento3
Tipo_Alias NOMBRE_ALIAS2 = elemento1, elemento5 : NOMBRE_ALIAS3 = elemento4
Tipo_Alias NOMBRE_ALIAS4 = elemento7, elemento2 :\
NOMBRE_ALIAS5 = elemento6, NOMBRE_ALIAS1
```

Tipo_Alias: Puede ser uno de los siguientes:

- Cmnd_Alias para comandos
- User_Alias para usuarios
- Runas_Alias para usuarios privilegiados
- Host_Alias para hosts

NOMBRE_ALIAS: Es el nombre del alias. Debe empezar por letra mayúscula y sólo se permiten letras mayúsculas y números. El resto son los elementos o listas de elementos por los cuales NOMBRE_ALIAS será expandido. Existe un alias especial, ALL, que se utiliza para englobar a todos los comandos, usuarios, usuarios privilegiados o hosts.

Ajuste de opciones

Como ya hemos dicho podemos definir opciones globalmente, por usuario, por usuario privilegiado y por host. La sintaxis es la siguiente:

```
Defaults lista_opciones
Defaults:usuario lista_opciones
Defaults>usuario_privilegiado lista_opciones
Defaults@host lista_opciones
```

La lista_opciones es una lista de opciones (como no) separadas por comas. Existen cuatro tipos de opciones:

- Booleanos: Que se activan con sólo escribir el nombre de la opción y se desactivan con el símbolo ! delante.
- Enteros: De la forma nombre_opcion = valor

- Strings: Igual que los enteros nombre_opcion = "valor"
- Listas: Que pueden ser de la forma nombre_opcion = "valor1 valor2". Éstas opciones también pueden utilizar += y -= en lugar de = para añadir elementos y quitar elementos respectivamente.

Reglas de acceso

Ahora toca definir los usuarios a los que permitimos utilizar sudo, los comandos que pueden ejecutar, bajo qué usuarios privilegiados ejecutarán los comandos y en qué hosts pueden hacerlo.
 usuario host = (usuario_privilegiado) comando

Bastante simple, ¿verdad? Hay que decir que cada elemento puede ser tanto un alias como una lista de elementos. La mención del usuario_privilegiado o la lista de ellos es opcional y por defecto se toma el root. En los ejemplos comentaremos algunos detalles aclaratorios de la sintaxis.

Existe una última posibilidad, y es poder eliminar la petición de contraseña para ejecutar uno o varios comandos. Se trata de las etiquetas NOPASSWD y PASSWD. Son opcionales y por defecto se asume PASSWD.

usuario host = (usuario_privilegiado) NOPASSWD: comando

110.1.7.9. su

Este comando nos permite ejecutar una shell como otro usuario en la sesión activa. Es decir, nos permite asumir la identidad de otro usuario (si conocemos su password claro):

```
$ whoami
pcm
$ su -
Password: *****
# whoami
root
# pwd
/root
#
```

Para terminar la sesión bastaría con presionar Ctrl + D (fin de fichero) o tecleando exit. En caso de no indicar ningún usuario con el comando su se supone que se está intentando asumir la identidad del root. La diferencia entre utilizar su - usuario y su usuario es que cuando se utiliza su - se hace login de la misma forma que si se logeará en la consola, cargando todos los ficheros de configuración de su perfil.

También es posible ejecutar comandos como si fueramos otro usuario utilizando su:

```
$ su lila -c 'rm -Rf /home/lila'
Password: *****
$
```

110.1.7.10. usermod

El comando usermod modifica los archivos donde se almacenan las configuraciones de las cuentas del sistema para reflejar los cambios que se especifican en la línea de comandos. Las opciones que se aplican a usermod son:

- a, - append
- c , - comment Comentario: Normalmente es modificado mediante *chfn*.
- d , - home HOME_DIR: Directorio de inicio de sesión.
- e , - EXPIREDATE EXPIRE_DATE: La fecha en la que será la cuenta de usuario deshabilitada. La fecha se especifica en el formato AAAA-MM-DD.
- f , - inactive INACTIVO: El número de días después que una contraseña ha caducado hasta que la cuenta es una discapacidad permanente. Un valor de 0 desactiva la cuenta tan pronto como la contraseña ha caducado, y un valor de -1 desactiva la función.
- g , - gid GRUPO: El nombre del grupo o el número de grupo de nuevo inicio de sesión inicial del usuario. El grupo debe existir.
- G , - groups GRUPO1 [, Grupo 2, ... [, GROUPN]]]: Una lista de grupos suplementarios que el usuario es miembro de. Cada grupo está separado del siguiente por una coma, sin espacios en blanco intermedios. Si el usuario es actualmente miembro de un grupo que no está en la lista, el usuario será eliminado del grupo.
- l , - login NEW_LOGIN: El nombre del usuario se cambia de LOGIN para NEW_LOGIN . Ninguna otra cosa ha cambiado. En particular, el nombre del usuario directorio de inicio probablemente se debe cambiar manualmente para reflejar el nuevo nombre de usuario.
- L , - lock: Bloqueo de la contraseña de un usuario. Esto pone un '!' frente a la contraseña encriptada.
- m , --move-home: Mover el contenido del directorio personal del usuario a la nueva ubicación.
- o , - no-unique: Cuando se utiliza con la -u opción, esta opción permite cambiar el ID de usuario a un valor no único.
- p , - password CONTRASEÑA: Esta opción no es recomendable ya que la contraseña será visible por los usuarios que consulten la lista de los procesos.
- s , - shell SHELL: El nombre de la shell del usuario nuevo. Si este campo en blanco hace que el sistema para seleccionar el shell por defecto.
- u , - uid UID: El nuevo valor numérico de identificación del usuario. Este valor debe ser único, a menos que la -o se utiliza la opción. El valor debe ser no negativo. Valores entre 0 y 999 se reservan normalmente para las cuentas del sistema. El buzón del usuario, y cualquier archivo que el usuario posee y que se encuentran en el directorio home del usuario tendrá el ID de usuario archivo modificado de forma automática. La propiedad de los archivos fuera del directorio home del usuario deben ser fijados manualmente.
- u , - unlock: Desbloquea la contraseña del usuario.

10.1.7.11. ulimit

Proporciona control sobre los recursos disponibles para el shell y para los procesos arrancados por él, en sistemas que permitan tal control. El valor de límite puede ser un número en la unidad especificada para el recurso, o el valor unlimited, o sea, ilimitado. Las opciones *-H* y *-S* especifican que el límite para el recurso dado va a ser duro o blando. Un límite duro es aquél que no puede ser aumentado una vez puesto; un límite blando puede incrementarse hasta el valor dado por el límite duro. Si no se especifican ni *-H* ni *-S*, se establecen ambos límites. Si límite se omite, se muestran los valores del límite blando del recurso, a menos que se dé la opción *-H*. Cuando se especifica más de un recurso, se imprime el nombre del límite y la unidad antes del valor. Otras opciones se interpretan como sigue:

- a Se informa de todos los límites actuales
- c El tamaño máximo de ficheros de volcados de memoria (core)
- d El tamaño máximo del segmento de datos de un proceso
- f El tamaño máximo de ficheros creados por el shell
- l El tamaño máximo que puede ser bloqueado en memoria

- m El tamaño del máximo conjunto residente (memoria)
- n El número máximo de descriptores de ficheros abiertos (la mayoría de sistemas no permiten establecer este valor)
- p El tamaño de una tubería en bloques de 512 B (esto puede no estar establecido)
- s El tamaño máximo de pila
- t La máxima cantidad de tiempo de CPU en segundos
- u El número máximo de procesos disponibles para un solo usuario
- v La máxima cantidad de memoria virtual disponible para el shell

Si se da límite, es el nuevo valor del recurso especificado (la opción -a es sólo para mostrar los valores). Si no se da ninguna opción, entonces se supone -f. Los valores están en incrementos de 1024 B, excepto para -t, que está en segundos, -p, que está en unidades de bloques de 512 B, y -n y -u, que son valores adimensionales. El estado de retorno es 0 a menos que se encuentre una opción inválida, se dé como límite un argumento no numérico distinto de unlimited, o bien ocurra un error mientras se establece un nuevo límite.

10.1.7.12. /etc/login.defs

Este archivo permite definir algunos valores por defecto para diferentes programas como useradd y espiración de contraseñas. Tiende a variar ligeramente entre distribuciones incluso entre versiones, pero suele estar bien comentado y tiende a contener los valores por defecto.

110.2.1 Contraseñas shadow

En un sistema Linux habitual cada usuario posee un nombre de entrada al sistema o login y una clave o password. El archivo */etc/passwd* contiene una línea por usuario donde se indica la información necesaria para que los usuarios puedan conectar al sistema y trabajar en él, separando los diferentes campos mediante ':'. Por ejemplo, en sistemas antiguos podemos encontrar entradas parecidas a la siguiente:

```
toni:LEgPN8jqSCHCg:1000:100:Antonio Villalon,,,:/export/home/toni:/bin/sh
```

En sistemas antiguos era habitual almacenar la clave cifrada en el propio fichero */etc/passwd* pero dado que a este archivo puede acceder cualquier usuario, no resultaba extraño que mediante programas de ataque por fuerza bruta, como John the Ripper, se pudiese obtener la clave de los usuarios.

En la actualidad se utiliza un método denominado Shadow Password u oscurecimiento de contraseñas para proteger las claves. La idea básica de este mecanismo es impedir que los usuarios sin privilegios puedan leer el fichero donde se almacenan las claves cifradas. Tal como se ha comentado anteriormente, el fichero */etc/passwd* tiene que tener permiso de lectura para todo el mundo si queremos que el sistema funcione correctamente. En equipos con oscurecimiento de contraseñas este fichero sigue siendo legible para todos los usuarios, pero a diferencia del mecanismo tradicional, las claves cifradas no se guardan en él, sino en el archivo */etc/shadow*, que sólo el root puede leer. En el campo correspondiente a la clave cifrada de */etc/passwd* no aparece ésta, sino un símbolo que indica a determinados programas (como */bin/login*) que han de buscar las claves en */etc/shadow*, generalmente una x:

```
toni:x:1000:100:Antonio Villalon,,,:/export/home/toni:/bin/sh
```

El aspecto de */etc/shadow* es en cierta forma similar al de */etc/passwd* que ya hemos comentado: existe una línea por cada usuario del sistema, en la que se almacena su login y su clave cifrada. Sin embargo, el resto de campos de este fichero son diferentes; corresponden a información que permite implementar otro mecanismo para proteger las claves de los usuarios, el envejecimiento de contraseñas o Aging Password:

```
toni:LEgPN8jqSCHCg:10322:0:99999:7:::
```

110.2.2 Servicios de Red

Podemos ver los diferentes servicios que un sistema Linux ofrece como potenciales puertas de entrada al mismo, o al menos como fuentes de ataques que ni siquiera tienen por qué proporcionar acceso a la máquina -como las negaciones de servicio-. De esta forma, si cada servicio ofrecido es un posible problema para nuestra seguridad, parece claro que lo ideal sería no ofrecer ninguno, poseer una máquina completamente aislada del resto; evidentemente, esto no suele ser posible hoy en día en la mayor parte de los sistemas. Por tanto, ya que es necesaria la conectividad entre equipos, hemos de ofrecer los mínimos servicios necesarios para que todo funcione correctamente; esto choca frontalmente con las políticas de la mayoría de fabricantes de sistemas Unix, que por defecto mantienen la mayoría de servicios abiertos al instalar un equipo nuevo: es responsabilidad del administrador preocuparse de cerrar los que no sean estrictamente necesarios.

Típicos ejemplos de servicios que suele ser necesario ofrecer son telnet o ftp; en estos casos no se puede aplicar el esquema todo o nada donde o bien ofrecíamos un servicio o lo denegábamos completamente: es necesaria una correcta configuración para que sólo sea posible acceder a ellos desde ciertas máquinas, como veremos al hablar de TCP Wrappers. También es una buena idea

sustituir estos servicios por equivalentes cifrados, como la familia de aplicaciones SSH, y concienciar a los usuarios para que utilicen estos equivalentes: hemos de recordar siempre - y recordar a los usuarios - que cualquier conexión en texto claro entre dos sistemas puede ser fácilmente capturada por cualquier persona situada en una máquina intermedia, con lo simplemente utilizando telnet estamos poniendo en juego la seguridad de sistemas y redes completas.

A parte de puertas de entrada, los servicios ofrecidos también son muy susceptibles de ataques de negación de servicio (DoS), por ejemplo por demasiadas conexiones abiertas simultáneamente en una misma máquina; incluso es posible que uno de estos ataques contra cierto servicio inutilice completamente a inetd, de forma que todos los ofrecidos desde él quedan bloqueados hasta que el demonio se reinicia. Este problema incluso puede ser muy grave: imaginemos que -por cualquier motivo- inetd deja de responder peticiones; si esto sucede es posible que ni siquiera podamos acceder a la máquina remotamente para solucionar el problema (por ejemplo telnet o incluso SSH si lo servimos desde inetd dejarían de funcionar).

110.2.3 TCP/Wrappers

TCP/Wrappers es un sistema de control de acceso a nivel de host, que permite controlar el acceso a los servicios de red, particularizando las acciones que se deben realizar para diferentes clientes.

- En función del sistema que envía la solicitud de conexión y del servicio solicitado se permite acceso o no.
- La gestión de acceso se realiza mediante dos ficheros: */etc/hosts.allow* y */etc/hosts.deny*.
- Para que los servicios TCP gestionados por inetd hagan uso de los wrappers hay que modificar el fichero */etc/inetd.conf*.

Ventajas del TCP Wrapper

- Transparencia para el cliente del host y el servicio de red
 - El cliente que se está conectando así como también el servicio de red wrapped no están al tanto de que están en uso los wrappers TCP.
 - Los usuarios legítimos son registrados y conectados al servicio solicitado mientras que las conexiones desde clientes prohibidos fallan.
- Administración centralizada de protocolos múltiples
 - Los wrappers TCP operan separadamente de los servicios de red que ellos protegen, permitiendo a muchas aplicaciones de servidor compartir un conjunto común de archivos de configuración para una administración más sencilla

Ficheros */etc/hosts.allow* y */etc/hosts.deny*

- Especifican pares servicio/cliente
- La forma de operar es:
 - Si se encuentra una coincidencia entre el par servicio/cliente en hosts.allow se da acceso.
 - En caso contrario si se encuentra una coincidencia en hosts.deny, se deniega el servicio.
 - En caso contrario, se da acceso
- Formato del fichero:

<lista_servicios> : <lista_clientes> [: órdenes]

- servicios: lista (delimitada por comas) de los servicios a los que se aplica la regla.
- clientes: lista de nombres o IPs de los clientes afectados por la regla.
- ordenes: (opcional) ordenes adicionales que se ejecutan cuando se cumple la regla.
- La configuración más segura es tener ALL: ALL por defecto en /etc/hosts.deny para después dar permiso específicamente a aquellos servicios y hosts que se deseé en /etc/hosts.allow.

Ejemplos

```
# /etc/host.allow
# Permitir correo a todo el mundo
in.smtpd : ALL
# ftp y finger sólo a host en mi dominio
# y una máquina específica
in.ftpd,in.fingerd : LOCAL, mihost.encasa.com
# Permite telnet para los host listados en un fichero
# y la red 192.168.
in.telnetd : /etc/telnet.hosts, 192.168.
# /etc/hosts.deny
# Desautorizo a todos los restantes
ALL : ALL
```

Palabras especiales

- ALL, que se corresponde con cualquier host.
- LOCAL se corresponde con cualquier nombre de host que no contenga un . o sea que esté en el mismo dominio que la máquina.
- PARANOID se corresponde con cualquier nombre que no se corresponda con su dirección IP (name spoofing).
- EXCEPT permite proporcionar una lista con excepciones.

Algunos patrones

- Nombre de host comenzando con un punto (.): todos los hosts compartiendo los componentes listados del nombre

ALL : .example.com

- Dirección IP que termina con un punto (.): todos los hosts compartiendo el grupo numérico inicial de una dirección IP

ALL : 192.168.

- Dirección IP/máscara de red

ALL : 192.168.0.0/255.255.254.0

- El asterisco (*): especifican grupos completos de nombres de host o direcciones IP

ALL : *.example.com

- La barra oblicua (/): especifica un nombre de archivo *in.telnetd : /etc/telnet.hosts*

Otras utilidades

- tcpdchk: chequea la configuración de *hosts.allow* y *hosts.deny*.
- tcpdmatch: predice como se comportarían las reglas ante una petición determinada.

110.2.4 Archivos y utilidades

110.2.4.1.- /etc/nologin

Evita que los usuarios no root entren al sistema. Si el fichero */etc/nologin* existe, **login** sólo permitirá acceder al usuario root. A cualquier otro usuario se le mostrará el contenido de este fichero y sus conexiones serán rechazadas.

110.2.4.2.- /etc/passwd

El archivo */etc/passwd* contiene toda la información relacionada con el usuario (registro, contraseña, etc.). Sólo el superusuario (root) puede cambiarla. Este archivo posee un formato especial que permite marcar a cada usuario y cada una de sus líneas tiene el siguiente formato:

nombre_de_cuenta : contraseña : numero_de_usuario : numero_de_grupo : comentario : directorio : programa_de_inicio

Se especifican siete campos separados por el carácter ":":

- El nombre de cuenta del usuario
- La contraseña del usuario (opcional pero si aparece codificada)
- El número entero que identifica al usuario para el sistema operativo (UID = ID del usuario, identificación del usuario)
- El número entero que identifica al grupo del usuario (GID = ID del grupo, identificación del grupo)
- El comentario en el que se puede encontrar la información sobre el usuario o simplemente su nombre real
- El directorio de conexión, que es el directorio que se abre cuando se conecta al sistema
- El comando es el que se ejecuta después de la conexión al sistema (con frecuencia éste es el intérprete de comandos)
- A continuación encontrará un ejemplo de un archivo *passwd*:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:daemon:/sbin:/bin/bash
news:x:9:13:News system:/etc/news:/bin/bash
uucp:x:10:14::/var/lib/uucp/taylor_config:/bin/bash
cquoi:x:500:100:Cool.....:/home/cquoi:/bin/bash
```

110.2.4.3.- /etc/shadow

/etc/shadow es un archivo que contiene la información de contraseñas para cuentas del sistema y la información opcional de duración de las mismas. Este archivo no debe ser legible por los usuarios normales si queremos mantener una mínima seguridad en las claves. Cada línea de este archivo contiene 9 campos, separados por dos puntos (":"), en el siguiente orden:

- **Cuenta:** Debe ser un nombre de cuenta válido, existente en el sistema.

- **Contraseña cifrada:** Si el campo de contraseña no contiene una cadena válida, por ejemplo una cadena que empiece por ! o *, el usuario no será capaz de utilizar una contraseña para iniciar sesión en el sistema (aunque el usuario podrá iniciar sesión en el sistema por otros medios). Este campo puede estar vacío, en cuyo caso no se requiere una contraseña para autenticar el nombre de usuario especificado. Sin embargo, algunas aplicaciones que leen el archivo /etc/shadow puede optar por no permitir la ejecución si el campo de la contraseña está vacía. Un campo de contraseña que se inicia con un signo de exclamación significa que la contraseña está bloqueada. Los caracteres restantes de la línea representan el campo de la contraseña antes de que la contraseña ha sido bloqueada.
- **Fecha de último cambio de contraseña:** La fecha de la último cambio de contraseña, expresado como el número de días desde el 1 de enero 1970. El valor 0 tiene un significado especial, y es que el usuario debe cambiar su password la próxima vez que se registrará en el sistema. Un campo vacío significa que las características de envejecimiento de contraseñas están desactivadas.
- **Duración mínima de la contraseña:** La duración mínima de la contraseña es el número de días que el usuario tendrá que esperar antes de que se le permitirá cambiar su contraseña nuevo. Un campo vacío y valor 0 significa que no hay duración mínima de la contraseña.
- **Duración máxima de la contraseña:** La duración máxima de la contraseña es el número de días después de que el usuario tenga que cambiar su contraseña. Después de este número de días transcurrido el tiempo, la contraseña puede ser todavía válida pero el usuario deberá cambiar su contraseña la próxima vez en que se iniciar sesión. Un campo vacío significa que no hay duración máxima de la contraseña, no hay período de contraseña de advertencia, y sin período de inactividad de la contraseña. Si la vigencia máxima es inferior a la duración mínima de la contraseña, el usuario no puede cambiar su contraseña.
- **Período de advertencia de la contraseña:** El número de días antes de una contraseña va a caducar durante el cual el usuario debe ser advertido de la caducidad de la misma. Un campo vacío y valor 0 significa que no hay período de advertencia de contraseña.
- **Período de inactividad de la contraseña:** El número de días después de que la contraseña ha expirado durante el cual la contraseña aún debe ser aceptada (y el usuario debe actualizar su contraseña durante el siguiente inicio de sesión). Después de la expiración de la contraseña y el período de caducidad ha transcurrido, no es posible registrarse con la contraseña del usuario actual. El usuario debe contactar con su administrador. Un campo vacío significa que no hay ejecución de un período de inactividad.
- **Fecha de caducidad de la cuenta:** La fecha de expiración de la cuenta, expresado como el número de días desde el 1 de enero 1970. Hay que tener en cuenta que la caducidad de la cuenta difiere de la caducidad de la contraseña. Un campo vacío significa que la cuenta nunca se expira. El valor 0 no debe utilizarse ya que se puede interpretar como una cuenta sin vencimiento, o como una caducidad en el día 1 de enero de 1970.
- **Campo reservado:**Este campo está reservado para uso futuro.

110.2.4.4.- /etc/xinetd.conf

El servicio xinetd es una versión extendida del servicio inetd y se configura a través del fichero /etc/xinetd.conf.. Este fichero contiene parámetros de configuración generales los cuales afectan

cada servicio bajo el control de xinetd. Se lee una vez cuando el servicio xinetd es iniciado, por esto, para que los cambios de la configuración tomen efecto, el administrador debe reiniciar el servicio xinetd. Abajo se muestra un ejemplo del archivo /etc/xinetd.conf:

```
defaults
{
instances = 60
log_type = SYSLOG authpriv
log_on_success = HOST PID
log_on_failure = HOST
cps = 25 30
}
includedir /etc/xinetd.d
```

Estas líneas controlan los siguientes aspectos de xinetd:

- instances — Configura el máximo número de peticiones que xinetd puede manejar simultáneamente.
- log_type — Configura xinetd para usar la facilidad de registro authpriv, el cual escribe las entradas de registro al archivo /var/log/secure. Al agregar una directiva tal como FILE /var/log/xinetdlog aquí, creará un archivo de registro personalizado llamado xinetdlog en el directorio /var/log/.
- log_on_success — Configura xinetd a registrar si la conexión es exitosa. Por defecto, la dirección IP del host remoto y el ID del proceso del servidor procesando la petición son grabados.
- log_on_failure — Configura xinetd para registrar si hay una falla de conexión o si la conexión no es permitida.
- cps — Configura xinetd para no permitir más de 25 conexiones por segundo a cualquier servicio dado. Si se alcanza este límite, el servicio es retirado por 30 segundos.
- includedir /etc/xinetd.d/ — Incluye las opciones declaradas en los archivos de configuración específicos del servicio localizados en el directorio /etc/xinetd.d/.

110.2.4.5.- /etc/xinet.d/*

El directorio /etc/xinetd.d/ contiene los archivos de configuración para cada servicio manejado por xinetd y los nombres de los archivos que se correlacionan con el servicio. Como sucede con xinetd.conf, este archivo sólo es leído cuando el servicio xinetd es arrancado. Para que los cambios tengan efecto, el administrador debe reiniciar el servicio xinetd.

El formato de los archivos en el directorio /etc/xinetd.d/ usan las mismas convenciones que /etc/xinetd.conf. La razón principal por la que la configuración para cada servicio es almacenada en un archivo separado es hacer más fácil la personalización y que sea menos probable afectar otros servicios.

Para tener una idea de cómo estos archivos están estructurados, considere el archivo /etc/xinetd.d/telnet:

```
service telnet
{
flags = REUSE
socket_type = stream
wait = no
```

```

user = root
server = /usr/sbin/in.telnetd
log_on_failure += USERID
disable = yes
}

```

Estas líneas controlan varios aspectos del servicio telnet:

- service — Define el nombre del servicio, usualmente uno listado en el archivo /etc/services.
- flags — Configura cualquier número de atributos para la conexión. REUSE instruye xinetd a reutilizar el socket para una conexión Telnet.
- socket_type — Configura el socket de red a escribir a stream.
- wait — Define si el servicio es de un sólo hilo (yes) o de múltiples hilos (no).
- user — Define bajo qué ID de usuario se ejecutará el proceso.
- server — Define el binario ejecutable a lanzar.
- log_on_failure — Define los parámetros de registro para log_on_failure además de aquellos ya definidos en xinetd.conf.
- disable — Define si el servicio está activo o no.

110.2.4.6.- /etc/inetd.conf

Inetd debería ejecutarse en el arranque. A partir de ese momento está a la escucha de conexiones en cierto conector (socket) de internet. Cuando encuentra una conexión en uno de sus conectores, decide a qué servicio de conexión corresponde, y llama a un programa para atender la solicitud. Cuando este programa termina, continúa a la escucha en el conector (salvo en algún caso que se describirá más adelante). Esencialmente, inetd permite ejecutar un demonio para llamar a otros muchos, reduciendo la carga del sistema.

Las opciones disponibles para inetd son:

-d: Activa la depuración.
-q longitudcola: Asigna el valor indicado al tamaño de la cola de escucha del conector. Por defecto es 128.

En ejecución, inetd lee su información de configuración de un fichero de configuración, que por defecto es /etc/inetd.conf. Tiene que haber una entrada para cada campo del fichero de configuración, con entradas para cada campo separadas por tab o espacios. Los comentarios se distinguen por un ``#'' al principio de la línea. Tiene que haber una entrada para cada campo. Los campos del fichero de configuración son de la siguiente forma:

```

nombre de servicio
tipo de conector
protocolo
wait/nowait[.max]
usuario[.grupo]
programa servidor
argumentos del programa servidor

```

Para especificar un servicio basado en Sun-RPC la entrada debería contener estos campos.

nombre servicio/versión
tipo de conector
rpc/protocolo
wait/nowait[.max]
usuario[.grupo]
programa servidor
argumentos del programa servidor

La entrada nombre de servicio es el nombre de un servicio válido del fichero /etc/services. Para servicio “internos” (discutidos después), el nombre de servicio tiene que ser el nombre oficial del servicio (esto es, la primera entrada de /etc/services). Cuando se usa para especificar un servicio basado en Sun-RPC, este campo es un nombre de servicio RPC válido del fichero /etc/rpc. la parte a la derecha de “/” es el número de versión RPC. Esto puede ser simplemente un argumento numérico o un rango de versiones. Un rango está acotado por las versiones menor y mayor - “rusers/1-3”.

El tipo de conector (tipo de socket) debería ser “stream”, “dgram”, “raw”, “rdm”, or “seqpacket”, dependiendo de si el conector es un flujo, datagrama, en bruto, mensaje entregado fiable o conector de paquetes secuenciados.

El protocolo tiene que ser un protocolo válido como los dados en /etc/protocols. Pueden ser ejemplos “tcp” o “udp”. Los servicios basado en Rpc se especifican con el tipo de servicio “rpc/tcp” o “rpc/udp”.

La entrada wait/nowait es aplicable a conectores de datagrama sólo (los otros conectores deberían tener una entrada “nowait” es este espacio). Si un servidor de datagrama conecta a su par, liberando el conector, así inetd puede recibir posteriores mensajes en el conector, esto se dice que es un servidor “multi-hilo” y debería usar la entrada “nowait” . Para los servidores de datagrama que procesa todos los datagramas entrantes por un conector y al fin y al cabo desconecta, el servidor se dice que es “hilo simple” y debería usar una entrada “wait”.

La entrada usuario debería contener el nombre de usuario bajo el que ejecutaría el servidor. Esto permite que a los servidores se les dé menos permisos que al root. Se puede especificar un nombre de grupo opcional añadiendo un punto al nombre de usuario seguido por el nombre de grupo. Esto permite a los servidores ejecutarse con un identificador de grupo (primario) diferente al especificado en el fichero /etc/passwd. Si se especifica un grupo y el usuario no es root, se asignan los grupos supplementarios asociados con ese usuario.

La entrada programa servidor debería contener la ruta completa del programa que se ejecutará por inetd cuando encuentre una solicitud en su conector. Si inetd proporciona este servicio internamente, esta entrada debería ser “internal”.

Los argumentos del programa servidor será como son normalmente los argumentos, empezando con argv[0], que es el nombre del programa Si proporciona este servicio internamente, la palabra “internal” debería estar en el lugar de esta entrada.

Inetd proporciona varios servicios “triviales” internamente usando rutinas con él mismo. “echo”, “discard”, “chargen” (generador de caracteres), “daytime” (fecha-hora en formato legible), y “time” (fecha-hora formato de máquina, en el formato del número de segundos desde medianoche de 1 de enero de 1900). Todos estos servicios están basadose en tcp. Para detalles de estos servicios, consulte el RFC adecuada del Network Information Center.

Inetd relee su fichero de configuración cuando recibe la señal de colgar SIGHUP. Se pueden añadir servicios, borrarlos o modificarlos cuando se lee el fichero de configuración. Inetd crea el fichero /var/run/inetd.pid que contiene su identificador de proceso.

110.2.4.7.- /etc/inet.d/*

El formato de los archivos en el directorio /etc/inetd.d/ usan las mismas convenciones que /etc/inetd.conf. La razón principal por la que la configuración para cada servicio es almacenada en un archivo separado es hacer más fácil la personalización y que sea menos probable afectar otros servicios.

110.2.4.8.- /etc/inittab

El fichero inittab describe qué procesos se inician en la carga y durante la operación normal (por ejemplo, /etc/init.d/boot,/etc/init.d/rc, gettys...). Init distingue múltiples niveles de ejecución, cada uno de los cuales puede tener su propio conjunto de procesos que se inician. Los niveles de ejecución válidos son 0-6 más

A, B y C para entradas bajo demanda. Una entrada del fichero inittab tiene el siguiente formato:

id:niveles_ejecución:acción:proceso

Las líneas que comienzan con `#' se ignoran

- **id:** Es una secuencia única de 1 a 4 caracteres que identifican una entrada de inittab (para las versiones de sysvinit compiladas con bibliotecas < 5.2.18 o bibliotecas a.out el límite es de 2 caracteres). Nota: Para gettys u otros procesos de presentación al sistema, el campo id debería de ser el sufijo tty de la correspondiente tty, por ejemplo, 1 para tty1. En otro caso, las contabilidades de conexiones puede que no funcionen correctamente.
- **niveles_ejecución:** Es la lista de niveles de ejecución para los cuales se llevarán a cabo las acciones especificadas.
- **acción:** Describe qué acción se debería llevar a cabo.
- **Proceso:** Especifica el proceso a ejecutar. Si el campo proceso comienza con un carácter '+', init no registrará utmp y wtmp para ese proceso. Esto es necesario para gettys que insisten en hacer sus propias labores de utmp/wtmp. Esto es también un fallo histórico.

El campo niveles_ejecución tiene que contener múltiples caracteres para diferentes niveles de ejecución. Por ejemplo, 123 especifica que el proceso se debería iniciar en los niveles de ejecución 1, 2 y 3. Las entradas de niveles de ejecución bajo demanda pueden contener una A, B, o C. Las entradas de campos de nivel_ejecución de sysinit, boot y bootwait se ignoran.

Cuando se cambia un nivel de ejecución, cualesquiera procesos en ejecución que no estén especificados en el nuevo nivel de ejecución se matan, primero con SIGTERM y después con SIGKILL.

Las acciones válidas para el campo acción son:

- **respawn:** El proceso se reiniciará cuando termine (v.g. getty).
- **wait:** El proceso se iniciará una vez cuando se entre en el nivel de ejecución específico e init esperará a su terminación.
- **once:** El proceso se ejecutará una vez cuando se entre en el nivel de ejecución

especificado.

- boot: El proceso se ejecutará durante el arranque del sistema. El campo The niveles_ejecución se ignora.
- Bootwait: El proceso se ejecutará durante el arranque del sistema, mientras init espera su terminación (v.g. /etc/rc). El campo niveles_ejecución se ignora.
- off: Esto no hace nada.
- Ondemand: Un proceso marcado con un nivel de ejecución ondemand se ejecutará cuando se llame al nivel de ejecución especificado ondemand. Sin embargo, no se produce cambio de nivel de ejecución (los niveles de ejecución ondemand son `a', `b', y `c').
- initdefault: Una entrada initdefault especifica el nivel de ejecución en el cual se entrará tras el arranque del sistema. Si no existe ninguno, init pedirá un nivel de ejecución en la consola. El campo proceso se ignora.
- Sysinit: El proceso se ejecutará durante el arranque del sistema. Se ejecutará antes de cualquier entrada boot o bootwait. El campo niveles_ejecución se ignora.
- Powerwait: El proceso se ejecutará cuando init reciba la señal SIGPWR, indicando que hay algún problema con la alimentación eléctrica. Init esperará que el proceso termine antes de continuar.
- Powerfail: Como en powerwait, excepto que init no espera que el proceso se complete.
- Powerokwait: El proceso se ejecutará cuando init reciba la señal SIGPWR, con la condición de que haya un fichero llamado /etc/powerstatus que contenga la palabra OK. Esto significa que la alimentación eléctrica ha vuelto.
- Ctrlaltdel: El proceso se ejecutará cuando init reciba la señal SIGINT. Esto significa que alguien en la consola del sistema ha pulsado la combinación de teclas CTRL-ALT-DEL . Normalmente uno quiere ejecutar algún tipo de shutdown bien para entrar en modo monousuario o reiniciar la máquina.
- Kbrequest: El proceso se ejecutará cuando init reciba una señal del gestor de teclado que se ha pulsado una combinación especial de teclas en el teclado de la consola.

Esto es un ejemplo de un inittab que reensambla el viejo inittab de Linux:

```
# inittab para linux
id:1:initdefault:
rc::bootwait:/etc/rc
1:1:respawn:/etc/getty 9600 tty1
2:1:respawn:/etc/getty 9600 tty2
3:1:respawn:/etc/getty 9600 tty3
4:1:respawn:/etc/getty 9600 tty4
```

Este fichero inittab ejecuta /etc/rc durante el arranque e inicia gettys en tty1-tty4. Un inittab más elaborado con diferentes niveles de ejecución (vea los comentarios interiores):

```
# Nivel para ejecutar
id:2:initdefault:

# Inicialización del sistema antes de cualquier otra cosa.
si::sysinit:/etc/rc.d/bcheckrc

# Nivel de Ejecución 0,6 es halt y reinicio, 1 modo mantenimiento.
```

```
l0:0:wait:/etc/rc.d/rc.halt
l1:1:wait:/etc/rc.d/rc.single
l2:2345:wait:/etc/rc.d/rc.multi
l6:6:wait:/etc/rc.d/rc.reboot

# Qué hacer ante el "saludo de 3 dedos".
ca::ctrlaltdel:/sbin/shutdown -t5 -rf now

# Nivel de ejecución 2&3: getty en consola, nivel 3 también getty
# en el puerto del módem.
1:23:respawn:/sbin/getty tty1 VC linux
2:23:respawn:/sbin/getty tty2 VC linux
3:23:respawn:/sbin/getty tty3 VC linux
4:23:respawn:/sbin/getty tty4 VC linux
S2:3:respawn:/sbin/uugetty ttyS2 M19200
```

110.2.4.9.- /etc/init.d/*

En este directorio se encuentran todos los "scripts" que facilitan el inicio y cierre de daemons/programas, estos "scripts" comúnmente toman los argumentos "stop" "start" "restart", estos argumentos generalmente provienen de lo que se especifica en los directorios /etc/rc.d/rc[0-6].d

110.3 Protección de datos con encriptación.

Peso en el examen de certificación: 3 puntos.

Objetivo: Utilizar técnicas de claves públicas para asegurar datos y comunicaciones.

Conceptos y áreas de conocimiento:

- Realizar la configuración básica del cliente OpenSSH 2 y su uso.
- Comprender el papel de OpenSSH
- Realizar la configuración básica y el uso de GnuPG.
- Entender los túneles SSH (incluyendo túneles X11)

Términos y utilidades:

- ssh
- ssh-keygen
- ssh-agent
- ssh-add
- `~/.ssh/id_rsa`
- `id_rsa.pub`
- `~/.ssh/id_dsa`
- `id_dsa.pub`
- `/etc/ssh/ssh_host_rsa_key`
- `ssh_host_rsa_key.pub`
- `/etc/ssh/ssh_host_dsa_key`
- `ssh_host_dsa_key.pub`
- `~/.ssh/authorized_keys`
- `/etc/ssh_known_hosts`
- gpg
- `~/.gnupg/*`

110.3.2 Uso y configuración básica del cliente OpenSSH 2

Casi todas las máquinas Linux (como casi todos los sistemas operativos) cuentan con un cliente de shell seguro (SSH). La versión más usada es OpenSSH, pero también son frecuentes una variedad de clientes SSH compatibles. Si bien el cliente SSH es esencial para efectuar la conexión con un host, los problemas de seguridad más importantes suelen relacionarse con la configuración correcta del servidor SSH.

Al iniciar una conexión con un servidor, el cliente decide confiar en el servidor de forma activa. El simple hecho de contar con un cliente SSH no permite acceso de ningún tipo al interior de la máquina; por consiguiente, contar con un cliente SSH no significa estar expuesto a vulnerabilidades.

La configuración de un servidor no es una tarea especialmente compleja ya que el daemon de servidor está diseñado para activar y hacer cumplir prácticas de seguridad beneficiosas. De todas formas, debe considerarse que se trata de un servidor que comparte recursos con los clientes cuyas solicitudes decide atender.

El protocolo SSH está disponible en dos versiones, versión 1 y versión 2. En los sistemas más modernos se recomienda usar la versión de protocolo 2, aunque ambos clientes y servidores suelen

mantener la compatibilidad con la versión 1 (a menos que esta capacidad se encuentre desactivada en las opciones de configuración) para poder conectarse con los sistemas que sólo admiten la versión 1, que cada vez son menos frecuentes.

Las versiones de protocolo 1 y 2 usan archivos de configuración algo diferentes. En la versión de protocolo 1, el cliente primero crea un par de claves RSA usando **ssh-keygen**. La clave privada se almacena en **\$HOME/.ssh/identity** y la pública en **\$HOME/.ssh/identity.pub**. Este mismo **identity.pub** debe agregarse a los archivos **\$HOME/.ssh/authorized_keys** remotos.

Evidentemente, aquí nos enfrentamos al problema del huevo o la gallina: ¿cómo es posible copiar un sistema remoto antes de tener acceso a él? Afortunadamente, SSH también soporta un método de autenticación alternativo que consiste en enviar contraseñas encriptadas en la línea que se evalúan mediante las pruebas usuales de inicio de sesión de sistemas remotos (por ejemplo: la cuenta de usuario debe existir y se debe proporcionar la contraseña correcta).

El protocolo 2 soporta tanto claves RSA como DSA. La autenticación RSA del protocolo 2 no es idéntica a la del protocolo 1, ya que presenta algunas mejoras. En el protocolo 2, las claves privadas se almacenan en **\$HOME/.ssh/id_rsa** y **\$HOME/.ssh/id_dsa**. El protocolo 2 también soporta una serie de algoritmos de confidencialidad e integridad adicionales: AES, 3DES, Blowfish, CAST128, HMAC-MD5, HMAC-SHA1, etc. El servidor puede configurarse en base a los algoritmos y el orden de alternativas deseados.

En las opciones de configuración generales, en lugar de almacenar información de claves, el cliente almacena sus claves en **/etc/ssh/ssh_config** (o, de estar disponible, en **/\$HOME/.ssh/config**). Las opciones del cliente también pueden configurarse con el switch **-o**; otro switch muy común es **-X** o **-x**, el cual activa o desactiva el reenvío X11. Si se encuentra activado, el puerto X11 se tuneliza a través de SSH para permitir el uso de conexiones X11 encriptadas.

Las herramientas como scp también usan un reenvío de puertos sobre SSH similar. Por ejemplo, en la pantalla local de la máquina local en la que me encuentro trabajando, puede iniciar una aplicación X11 que sólo existe remotamente (en la subred local, en este caso):

```
$ which gedit # not on local
system $ ssh -X dqm@192.168.2.2 Password: Linux averatec 2.6.10-5-386 #1 Mon Oct
10 11:15:41 UTC 2005 i686 GNU/Linux No mail. Last login: Thu Feb 23 03:51:15
2006 from 192.168.2.101 dqm@averatec:~$ gedit &
```

110.3.3 Claves del Servidor OpenSSH 2

El daemon sshd, específicamente en su versión OpenSSH, permite las comunicaciones encriptadas seguras entre dos hosts no confiables sobre una red no segura. El servidor sshd de base suele activarse durante la inicialización y escucha las conexiones de clientes bifurcando un nuevo daemon para cada conexión de cliente. Los daemons bifurcados administran el intercambio de claves, la encriptación, la autenticación, la ejecución de comandos y el intercambio de datos.

Al igual que con la herramienta de cliente, el servidor sshd acepta una variedad de opciones en la línea de comandos, pero suele estar configurado por el archivo **/etc/ssh/sshd_config**. También se usan otros archivos de configuración. Por ejemplo, se emplean los controles de acceso **/etc/hosts.allow** y **/etc/hosts.deny**. Las claves se almacenan de forma similar a como sucede en el extremo del cliente, en **/etc/ssh/ssh_host_key** (protocolo 1), **/etc/ssh/ssh_host_dsa_key**, **/etc/ssh/ssh_host_rsa_key**; y las claves públicas en **/etc/ssh/ssh_host_dsa_key.pub** y **friends**. También, al igual que con el cliente, se usa ssh-keygen para generar las claves en el comienzo.

Consulte las páginas man de sshd y ssh-keygen para obtener más información sobre la configuración de archivos y la copia de claves generadas en los archivos correspondientes.

Muchas opciones de configuración se encuentran en /etc/ssh/sshd_config y los valores predeterminados por lo general resultan adecuados (y adecuadamente seguros). Vale la pena destacar algunas opciones:

- *AllowTcpForwarding* activa o desactiva el reenvío de puertos (tunelización). Esta opción se encuentra predeterminadamente activada (en "YES").
- *Ciphers* controla la lista y el orden de los algoritmos de encriptación a utilizar.
- *AllowUsers* y *AllowGroups* aceptan patrones comodines y permiten establecer cuáles usuarios pueden siquiera intentar autenticarse.
- *DenyGroups* y *DenyUsers* actúan simétricamente, como podría esperarse.
- *PermitRootLogin* permite que el SSH del usuario root ingrese a la máquina.
- *Protocol* permite especificar si se aceptarán ambas versiones de protocolo (o cuál versión de protocolo se aceptará).
- *TCPKeepAlive* es la opción a revisar si está perdiendo conexiones SSH. Si esta opción está activada, envía un mensaje "keepalive" para verificar las conexiones y esto podría causar la desconexión si ocurren errores temporales en la ruta.

110.3.4 Configuraciones y usos básicos de GnuPG

GnuPG es una herramienta de seguridad en comunicaciones electrónicas. Entre las funciones de GnuPG se incluyen generar un par de claves, intercambiar y comprobar la autenticidad de claves, cifrar y descifrar documentos, y firmar documentos y verificar firmas digitales. GnuPG utiliza criptografía de clave pública para que los usuarios puedan comunicarse de un modo seguro. En un sistema de claves públicas cada usuario posee un par de claves, compuesto por una clave privada y una clave pública. Cada usuario debe mantener su clave privada secreta; no debe ser revelada nunca. La clave pública se puede entregar a cualquier persona con la que el usuario desee comunicarse. GnuPG implementa un esquema algo más sofisticado en el que un usuario tiene un par de claves primario, y ninguno o más de un par de claves adicionales subordinadas. Los pares de claves primarios y subordinados se encuentran agrupados para facilitar la gestión de claves, y el grupo puede ser considerado como un sólo par de claves.

Generar un nuevo par de claves

La opción de la línea de órdenes --gen-key se usa para generar un nuevo par de claves primario.

\$ gpg --gen-key

GnuPG es capaz de crear varios tipos diferentes de pares de claves, pero debe existir una clave primaria capaz de generar firmas. Por lo tanto, existen sólo tres opciones. La opción 1 genera dos pares de claves. Un par de claves DSA que es el par de claves primario que se usará sólo para firmar. Un par de claves subordinadas ElGamal que se usará para el cifrado. La opción 2 es parecida a la anterior, pero sólo genera un par de claves DSA. La opción 4[2] genera un único par de claves ElGamal, que se usará tanto para firmar como para cifrar. En todos los casos existe la posibilidad de añadir subclaves adicionales para cifrar y firmar «a posteriori». La mayoría de los usuarios tienen suficiente con la opción por definición.

También hay que escoger un tamaño para la clave. El tamaño de una clave DSA debe estar entre los 512 y 1024 bits, y una clave ElGamal puede ser de cualquier tamaño. Sin embargo, GnuPG requiere que las claves no sean menores de 768 bits. Por tanto, si se escogió la opción 1 y también un tamaño de claves mayor de 1024 bits, la clave ElGamal tendrá el tamaño deseado pero la DSA se limitará a

1024 bits.

Cuanto más larga sea la clave, más segura será contra ataques de «fuerza bruta», pero por lo demás el tamaño de la clave que se da por definición es el adecuado, ya que sería más barato circunvalar el cifrado que intentar entrar mediante ataques de fuerza. Además, el cifrado y descifrado de mensajes se ralentizaría a medida que se incrementara el tamaño de la clave, y un tamaño de clave más grande podría afectar a la longitud de la firma digital. Una vez seleccionado, el tamaño de una clave no se puede cambiar nunca.

Para terminar, hay que escoger una fecha de caducidad. Si se escogió anteriormente la opción 1, la fecha de caducidad se usará para sendos pares de claves, ElGamal y DSA.

Para la mayoría de los usuarios, una clave sin fecha de caducidad es la adecuada. Sin embargo, si se escoge con fecha de caducidad, el tiempo para ésta debe ser escogido con cuidado, ya que, aunque es posible cambiar la fecha de caducidad posteriormente a la generación de la clave, puede ser difícil comunicar un cambio a aquellos usuarios que posean esta clave pública.

Además de los parámetros de la clave, el usuario debe dar un identificador. El identificador de usuario se usa para asociar la clave que se está creando con una usuario real. Sólo se creará un identificador de usuario al generar una clave, pero es posible crear identificadores adicionales si se desea usar la clave en dos o más contextos, v.g., si se usa por una parte en la oficina como empleado y por otra parte en casa como activista político. Hay que tener cuidado al crear un identificador de usuario, ya que después éste no puede ser editado para introducir cambios.

GnuPG necesita una contraseña con el fin de proteger las claves privadas, primarias y secundarias, que posea el usuario. No hay límite para la longitud de una contraseña, y ésta debe ser escogida con sumo cuidado. Desde un punto de vista de seguridad, la contraseña que desbloquea la clave privada es uno de los puntos más débiles en GnuPG (y en otros sistemas de cifrado de clave pública), ya que es la única protección que tiene el usuario si alguien se apoderara de su clave privada. Para una contraseña lo ideal es que no se usen palabras de un diccionario, y que se mezclen mayúsculas y minúsculas, dígitos, y otros caracteres. Una buena contraseña es crucial para el uso seguro de GnuPG.

Generar un certificado de revocación

Después de haber generado un par de claves, el usuario debe, de forma inmediata, generar un certificado de revocación para la clave pública primaria, mediante el uso de la opción --gen-revoke. Si el usuario olvidara la contraseña, o si su clave privada estuviera en peligro o extraviada, este certificado de revocación podría ser hecho público para notificar a otros usuarios que la clave pública no debe ser usada nunca más. Una clave pública revocada puede ser usada para verificar firmas hechas por el usuario en el pasado, pero no puede ser usada para cifrar datos. Esto tampoco afecta a la capacidad de descifrar mensajes que hayan sido cifrados con la clave antes de su revocación, siempre y cuando el usuario todavía tenga acceso a la clave privada.

```
$ gpg --output D58711B7.asc --gen-revoke 0xD58711B7  
sec 1024D/D58711B7 1999-09-24 Javier (Paramo S.L.)  
<javier@casa.es>
```

El argumento miclave debe ser un especificador de clave, ya sea éste el identificador de clave ("key ID") del par primario del usuario, o ya sea cualquier otra parte de un identificador de usuario ("user ID") que identifique el par de claves del susodicho usuario. El certificado que se genere se encontrará en el fichero revoke.asc. Si se omite la opción --output, el resultado se pondrá en la salida típica. Dado que el certificado es corto, es posible que el usuario desee imprimir una copia en papel del certificado para guardarla en algún sitio seguro, como por ejemplo una caja fuerte de

seguridad. El certificado no debería ser guardado en lugares a los que otros puedan tener acceso, ya que cualquiera podría hacer público el certificado de revocación e inutilizar la correspondiente clave pública.

Intercambiar claves

Para poder comunicarse con otros, el usuario debe intercambiar las claves públicas. Para obtener una lista de las claves en el fichero («anillo») de claves públicas, se puede usar la opción de la línea de órdenes --list-keys.

```
$ gpg --list-keys
```

Exportar una clave pública

Para poder enviar una clave pública a un interlocutor, antes hay que exportarla. Para ello se usará la opción de la línea de órdenes --export. Es necesario un argumento adicional para poder identificar la clave pública que se va a exportar. Como en la opción anterior --gen-revoke, hay que usar el identificador de clave o cualquier parte del identificador de usuario para identificar la clave que se desea exportar.

```
javier:~$ gpg --output javi.gpg --export javier@casa.es
```

La clave se exporta en formato binario, y esto puede no ser conveniente cuando se envía la clave por correo electrónico o se publica en una página web. Por tanto, GnuPG ofrece una opción de la línea de órdenes --armor[5] que fuerza que la salida de la orden sea generada en formato armadura-ASCII, parecido a los documentos codificados con uuencode. Por regla general, cualquier salida de una orden de GnuPG, v.g.. claves, documentos cifrados y firmas, pueden ir en formato armadura-ASCII añadiendo a la orden la opción --armor.

```
javier:~$ gpg --armor --output javi.asc --export javier@casa.es
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v0.9.8 (GNU/Linux)
Comment: For info see http://www.gnupg.org

[...]
-----END PGP PUBLIC KEY BLOCK-----
```

Importar una clave pública

Se puede añadir una clave pública al anillo de claves públicas mediante la opción --import.

```
javier:~$ gpg --import arancha.gpg
gpg: key B63E132C: public key imported
gpg: Total number processed: 1
gpg: imported: 1
```

```
javier:~$ gpg --list-keys
```

```
/home/javier/.gnupg/pubring.gpg
```

```
-----  
pub 1024D/D58711B7 1999-09-24 Javier (Paramo S.L.)  
<javier@casa.es>  
sub 1024g/92F6C9E3 1999-09-24
```

```
pub 1024D/B63E132C 1999-09-24 Aranzazu (A.G.deZ.) <arancha@nav.es>  
sub 1024g/581A915F 1999-09-24
```

Una vez que la clave haya sido importada, es necesario validarla. GnuPG usa un potente y flexible modelo de confianza que no requiere que el usuario dé validez personalmente a cada clave que importe. Sin embargo, algunas claves pueden necesitar que el usuario les dé validez de forma personal. Una clave se valida verificando la huella digital de la clave, y firmando dicha clave para certificar su validez. La huella digital se puede ver con la opción de la línea de órdenes --fingerprint, pero para certificar la clave hay que editarla.

```
javier:~$ gpg --edit-key arancha@nav.es
```

```
pub 1024D/B63E132C created: 1999-09-24 expires: never trust: -/q  
sub 1024g/581A915F created: 1999-09-24 expires: never  
(1) Aranzazu (A.G.deZ.) <arancha@nav.es>
```

```
Command> fpr
```

```
pub 1024D/B63E132C 1999-09-24 Aranzazu (A.G.deZ.) <arancha@nav.es>  
Fingerprint: 4203 82E2 448C BD30 A36A 9644 0612 8A0F B63E 132C
```

La huella digital de una clave se verifica con el propietario de la clave. Esto puede hacerse en persona o por teléfono, o por medio de otras maneras, siempre y cuando el usuario pueda garantizar que la persona con la que se está comunicando sea el auténtico propietario de la clave. Si la huella digital que se obtiene por medio del propietario es la misma que la que se obtiene de la clave, entonces se puede estar seguro de que se está en posesión de una copia correcta de la clave.

Después de comprobar la huella digital ya se puede firmar la clave con el fin de validarla. Debido a que la verificación es un punto débil en criptografía de clave pública, es aconsejable ser cuidadoso en extremo y siempre comprobar la huella digital de una clave con la que nos dé el propietario antes de firmar dicha clave.

```
Command> sign
```

```
pub 1024D/B63E132C created: 1999-09-24 expires: never trust: -/q  
Fingerprint: 4203 82E2 448C BD30 A36A 9644 0612 8A0F B63E 132C
```

```
Aranzazu (A.G.deZ.) <arancha@nav.es>
```

```
Are you really sure that you want to sign this key  
with your key: "Javier (Paramo S.L.) <javier@casa.es>"
```

```
Really sign? y
```

```
You need a passphrase to unlock the secret key for  
user: "Javier (Paramo S.L.) <javier@casa.es>"  
1024-bit DSA key, ID D58711B7, created 1999-09-24
```

```
Enter passphrase:
```

Una vez firmada, el usuario puede comprobar la clave para obtener un listado de las firmas que lleva y para ver la firma que le acaba de añadir. Cada identificador de usuario tendrá una o más autofirmas, así como una firma por cada usuario que haya validado la clave en cuestión.

Command> check

```
uid Aranzazu (A.G.deZ.) <arancha@nav.es>
sig! B63E132C 1999-09-24 [self-signature]
sig! D58711B7 1999-09-24 Javier (Paramo S.L.) <javier@casa.es>
```

Command> quit

Cifrar y descifrar documentos

Cada clave pública y privada tiene un papel específico en el cifrado y descifrado de documentos. Se puede pensar en una clave pública como en una caja fuerte de seguridad. Cuando un remitente cifra un documento usando una clave pública, ese documento se pone en la caja fuerte, la caja se cierra, y el bloqueo de la combinación de ésta se gira varias veces. La parte correspondiente a la clave privada, esto es, el destinatario, es la combinación que puede volver a abrir la caja y retirar el documento. Dicho de otro modo, sólo la persona que posee la clave privada puede recuperar un documento cifrado usando la clave pública asociada al cifrado.

Con este modelo mental se ha mostrado el procedimiento de cifrar y descifrar documentos de un modo muy simple. Si el usuario quisiera cifrar un mensaje para Javier, lo haría usando la clave pública de Javier, y él lo descifraría con su propia clave privada. Si Javier quisiera enviar un mensaje al usuario, lo haría con la clave pública del usuario, y éste lo descifraría con su propia clave privada.

Para cifrar un documento se usa la opción --encrypt. El usuario debe tener las claves públicas de los pretendidos destinatarios. El programa espera recibir como entrada el nombre del documento que se desea cifrar o, si éste se omite, una entrada típica. El resultado cifrado se coloca en la salida típica o donde se haya especificado mediante la opción --output. El documento se comprime como medida adicional de seguridad, aparte de cifrarlo.

```
javier:~$ gpg --output doc.gpg --encrypt --recipient
arancha@nav.es doc
```

La opción --recipient se usa una vez para cada destinatario, y lleva un argumento extra que especifica la clave pública con la que será cifrado el documento. El documento cifrado sólo puede ser descifrado por alguien con una clave privada que complemente uno de las claves públicas de los destinatarios. El usuario, en este caso el remitente, no podrá descifrar un documento cifrado por sí mismo a menos que haya incluido su propia clave pública en la lista de destinatarios.

Para descifrar un mensaje se usa la opción --decrypt. Para ello es necesario poseer la clave privada para la que el mensaje ha sido cifrado. De igual modo que en el proceso de cifrado, el documento a descifrar es la entrada, y el resultado descifrado la salida.

```
arancha% gpg --output doc --decrypt doc.gpg
```

```
You need a passphrase to unlock the secret key for
user: "Aranzazu (A.G.deZ.) <arancha@nav.es>"  
1024-bit ELG-E key, ID 581A915F, created 1999-09-24 (main key ID  
B63E132C)
```

Enter passphrase:

También es posible cifrar documentos sin usar criptografía de clave pública. En su lugar, se puede usar sólo una clave de cifrado simétrico para cifrar el documento. La clave que se usa para el cifrado simétrico deriva de la contraseña dada en el momento de cifrar el documento, y por razones de seguridad, no debe ser la misma contraseña que se esté usando para proteger la clave privada. El cifrado simétrico es útil para asegurar documentos cuando no sea necesario dar la contraseña a otros. Un documento puede ser cifrado con una clave simétrica usando la opción --symmetric.

```
javier:~$ gpg --output doc.gpg --symmetric doc
```

Enter passphrase:

Firmar y verificar firmas

Una firma digital certifica un documento y le añade una marca de tiempo. Si posteriormente el documento fuera modificado en cualquier modo, el intento de verificar la firma fallaría. La utilidad de una firma digital es la misma que la de una firma escrita a mano, sólo que la digital tiene una resistencia a la falsificación. Por ejemplo, la distribución del código fuente de GnuPG viene firmada con el fin de que los usuarios puedan verificar que no ha habido ninguna manipulación o modificación al código fuente desde que fue archivado.

Para la creación y verificación de firmas, se utiliza el par público y privado de claves en una operación que es diferente a la de cifrado y descifrado. Se genera una firma con la clave privada del firmante. La firma se verifica por medio de la clave pública correspondiente. Por ejemplo, Javier haría uso de su propia clave privada para firmar digitalmente la entrega de su última ponencia a la Revista de Química Inorgánica. El editor asociado que la recibiera, usaría la clave pública de Javier para comprobar la firma, verificando de este modo que el envío proviene realmente de Javier, y que no ha sido modificado desde el momento en que Javier lo firmó. Una consecuencia directa del uso de firmas digitales es la dificultad en negar que fue el propio usuario quien puso la firma digital, ya que ello implicaría que su clave privada ha sido puesta en peligro.

La opción de línea de órdenes --sign se usa para generar una firma digital. El documento que se desea firmar es la entrada, y la salida es el documento firmado.

```
javier:~$ gpg --output doc.sig --sign doc
```

```
You need a passphrase to unlock the private key for
user: "Javier (Paramo S.L.) <javier@casa.es>"  
1024-bit DSA key, ID D58711B7, created 1999-09-24
```

Enter passphrase:

El documento se comprime antes de ser firmado, y la salida es en formato binario.

Con un documento con firma digital el usuario puede llevar a cabo dos acciones: comprobar sólo la firma o comprobar la firma y recuperar el documento original al mismo tiempo. Para comprobar la firma se usa la opción --verify. Para verificar la firma y extraer el documento se usa la opción --decrypt. El documento con la firma es la entrada, y el documento original recuperado es la salida.

```
arancha% gpg --output doc --decrypt doc.sig
gpg: Signature made Fri Sep 24 12:02:38 1999 CDT using DSA key ID
D58711B7
gpg: Good signature from "Javier (Paramo S.L.) <javier@casa.es>"
```

Documentos con firmas ASCII

Las firmas digitales suelen usarse a menudo para firmar mensajes de correo electrónicos o en los grupos de noticias. En estas situaciones no se debe comprimir el documento al firmarlo, ya que para aquellos que no dispongan de un sistema para procesarlo sería ininteligible.

```
javier:~$ gpg --clearsign doc
```

```
You need a passphrase to unlock the secret key for
user: "Javier (Paramo S.L.) <javier@casa.es>"
1024-bit DSA key, ID D58711B7, created 1999-09-24
```

```
-----BEGIN PGP SIGNED MESSAGE-----
```

```
Hash: SHA1
```

```
[...]
```

```
-----BEGIN PGP SIGNATURE-----
```

```
Version: GnuPG v0.9.8 (GNU/Linux)
```

```
Comment: For info see http://www.gnupg.org
```

```
iEYEAARECAAYFAjdYCQoACgkQJ9S6ULT1dqz6IwCfQ7wP6i/i8Hhbc0SKF4ELyQB1
oCoAo0uqpRqEzr4k0kQqHRLE/b8/Rw2k
=y6kj
```

```
-----END PGP SIGNATURE-----
```

Firmas acompañantes

Un documento firmado tiene una utilidad limitada. Los otros usuarios deben recuperar la versión original del documento de la versión firmada, y aun en el caso de los documentos firmados en ASCII, el documento firmado debe ser editado para poder recuperar el original. Por tanto, existe un tercer método para firmar un documento, que genera una firma acompañante. Para generar una firma acompañante se usa la opción --detach-sig.

```
javier:~$ gpg --output doc.sig --detach-sig doc
```

```
You need a passphrase to unlock the secret key for
user: "Javier (Paramo S.L.) <javier@casa.es>"
1024-bit DSA key, ID D58711B7, created 1999-09-24
```

Enter passphrase:

Tanto el documento como la firma acompañante son necesarios para poder verificar la firma. La opción --verify se usará para comprobar la firma.

```
arancha% gpg --verify doc.sig doc
gpg: Signature made Fri Sep 24 12:38:46 1999 CEST using DSA key ID
D58711B7
gpg: Good signature from "Javier (Paramo S.L.) <javier@casa.es>"
```

10.3.5 Túneles SSH

OpenSSH le permite crear un tunel para encapsular otro protocolo dentro de un canal SSH encriptado. Esta capacidad se encuentra predeterminadamente activada en el servidor sshd, pero podría haberse desactivado mediante opciones de la línea de comandos o del archivo de configuración. Con esta capacidad activada, los clientes pueden emular fácilmente cualquier puerto/protocolo que deseen usar para una conexión. En el siguiente ejemplo se crea un túnel para telnet:

```
% ssh -2 -N -f -L  
5023:localhost:23 user@foo.example.com % telnet localhost 5023
```

Por supuesto, este ejemplo no tiene ningún sentido, ya que un shell de comando SSH cumple con la misma función que el Shell telnet. Pero sí tendría sentido crear una conexión POP3, HTTP, SMTP, FTP, X11 u otra conexión de protocolo de manera análoga. El concepto de base consiste en que un puerto de host local específico actúa como si fuera el servicio remoto, logrando que paquetes de comunicación reales se transporten sobre la conexión SSH en forma encriptada.

Las opciones usadas en el ejemplo son las siguientes:

- 2(usuario protocolo 2),
- N(sin comandos/solo túnel),
- f(SSH en segundo plano) y
- L,(describir túnel como "localport:remotehost:remoteport").

También se especifica el servidor (con nombre de usuario).

10.3.6 Archivos y Utilidades

10.3.6.1 ssh

SSH es un programa que permite acceder a otro ordenador a través de la red, ejecutar comandos en la máquina remota y mover ficheros entre dos máquinas. Provee autenticación y comunicaciones seguras sobre canales inseguros. Es un reemplazo de rlogin, rsh y rcp.

Para iniciar una sesión en otra máquina usando ssh:

```
$ ssh usuario1@servidor.dominio.es  
The authenticity of host 'servidor.dominio.es (192.168.0.2)' can't  
be established.  
RSA key fingerprint is  
97:4f:66:f5:96:ba:6d:b2:ef:65:35:45:18:0d:cc:29.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'servidor.dominio.es' (RSA) to the list  
of known hosts.  
usuario1@servidor.dominio.es's password:
```

La primera vez que realizas la conexión debes aceptar la firma del otro host. De esta manera se establece una relación de confianza que se traduce en archivar la clave pública de este servidor en el

fichero \$HOME/.ssh/known_hosts.

La sintaxis básica del comando ssh es:

ssh user@hostname [command]

El comando es opcional. Si se especifica en lugar de obtener un shell se ejecuta el comando en la máquina remota. Por ejemplo podríamos hacer un ls en la máquina remota y observar su salida:

ssh usuario1@servidor.dominio.es ls

Una de las funcionalidades que le da mayor potencia al ssh es la redirección de las X. Si observas la variable de entorno DISPLAY observarás que tiene la forma localhost:n.n, esta permite que al abrir cualquier aplicación gráfica su salida se redirija al display del cliente.

```
$ ssh -X usuario1@servidor.dominio.es
$ echo $DISPLAY
localhost:11.0
$ xeyes&
```

10.3.6.2 ssh-keygen

Para poder crear nuevas llaves privadas y sus correspondientes llaves públicas, OpenSSH dispone de una herramienta llamada ssh-keygen, esta herramienta puede crear llaves RSA para el protocolo SSH versión 1, cuyo uso se desaconseja y por otro lado, también puede generar llaves RSA o DSA para el protocolo SSH versión 2. Para especificar que tipo de llave crear, se emplea el parámetro -t seguido del tipo de llave: "rsa" o "dsa". Por ejemplo, para crear una llave RSA podemos poner:

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hell/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hell/.ssh/id_rsa.
Your public key has been saved in /home/hell/.ssh/id_rsa.pub.
The key fingerprint is:
c0:40:50:27:e8:d9:b8:55:d6:a4:5f:af:e5:30:5d:9b hell@local
```

Por el contrario, para generar una llave DSA, simplemente:

```
$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/hell/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hell/.ssh/id_dsa.
Your public key has been saved in /home/hell/.ssh/id_dsa.pub.
The key fingerprint is:
f5:b2:f3:2d:43:1b:22:44:98:6c:fe:42:df:a3:15:09 hell@local
[hell@local] $
```

Los anteriores comandos piden los mismos datos, el primero, en donde salvar la llave privada, si simplemente pulsamos intro, se salvará en la ruta por defecto (la indicada entre paréntesis), a no ser que se empleen varias llaves privadas, la ruta por defecto es una buena opción, ya que el cliente la usará sin necesidad de que el usuario tenga que especificarla.

Las siguientes dos cosas que pide son la frase clave con la que encriptar la llave privada y una petición de que se repita la frase para cerciorarse de que no se han cometido errores al escribirla la primera vez. Para la frase con la que encriptar la llave privada se puede emplear cualquier carácter (letras, numeros, signos de puntuación, espacios), en principio el tamaño de la frase puede ser arbitrario, pero ssh-keygen se quejará si es menor de cuatro caracteres. En caso de no introducir ninguna frase, la llave privada quedará sin encriptar, lo cual podría ser útil para realizar algunas automatizaciones como podría ser el realizar copias de seguridad, pero, para uso habitual, se desaconseja dejar las llaves privadas sin encriptar por el peligro que puede representar que estas sean robadas.

A continuación el ssh-keygen muestra donde se ha salvado la llave privada (/home/hell/.ssh/id_dsa) y donde está la llave pública que le corresponde (/home/hell/.ssh/id_dsa.pub), que no es más que el mismo nombre de archivo con la extensión .pub añadida al final.

En la última línea imprime una huella dactilar que sirve para identificar la llave que acabamos de crear, seguida de un comentario que puede servir para identificar la llave pública.

Tamaño de las llaves públicas

Por defecto, ssh-keygen genera llaves de 2048 bits, cuanto más grande sea una llave, más segura será. En la actualidad, ssh-keygen admite que las llaves tengan un mínimo de 512 bits y aunque en la página del manual no se indique, el máximo son 32768 bits, pero este último valor podría cambiar a medida que la potencia de los equipos informáticos se incremente. No obstante, con la liberación de la versión 4.3 de OpenSSH, sus desarrolladores decidieron fijar el tamaño de las llaves DSA a 1024 bits, pudiendo alterarse tan sólo el tamaño para las llaves RSA.

Por lo general el número de bits para la llave que escoge ssh-keygen por defecto es suficiente, pero para quienes prefieran otros valores, se puede especificar el tamaño de la llave con el parámetro -b seguido del número de bits que se desea que tenga la llave. Un ejemplo para generar una llave RSA de 4096 bits podría ser el siguiente:

```
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hell/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hell/.ssh/id_rsa.
Your public key has been saved in /home/hell/.ssh/id_rsa.pub.
The key fingerprint is:
4b:29:23:e9:20:c1:e5:32:6e:fa:b4:91:9a:01:b5:10 hell@local
```

Comentarios de las llaves públicas

En el momento de su creación, ssh-keygen nos permite añadir un comentario a las claves públicas, por defecto el comentario que pone es del tipo usuario@máquina, pero podemos emplear el

parámetro -C seguido del comentario que queramos, para así diferenciar más fácilmente nuestra llave pública del resto de llaves. A modo de ejemplo, supongamos que queremos crear una clave de pruebas de tipo RSA con el comentario "Clave de pruebas":

```
$ ssh-keygen -t rsa -C "Clave de pruebas"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hell/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hell/.ssh/id_rsa.
Your public key has been saved in /home/hell/.ssh/id_rsa.pub.
The key fingerprint is:
97:47:90:2d:b6:d9:ab:6d:91:41:ed:ad:dc:fb:a0:64 Clave de pruebas
```

Cambiar la frase clave de una llave privada

En alguna ocasión nos veremos en la necesidad de querer cambiar la frase con la que una llave privada fue encriptada, o en el caso de que la llave privada no estubiese encriptada, querer encriptarla. Para conseguir este objetivo podemos invocar al programa ssh-keygen con el parámetro -p, veamos un ejemplo:

```
$ ssh-keygen -p
Enter file in which the key is (/home/hell/.ssh/id_rsa):
Enter old passphrase:
Key has comment '/home/hell/.ssh/id_rsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

10.3.6.3 ssh-agent

La función de un agente ssh es la de gestionar las llaves privadas en las conexiones SSH del lado del cliente. Esto nos permite mantener la llave desencriptada (sin el passphrase) en la memoria. Nos brinda comodidad porque el agente ssh se va a encargar de darle la llave privada (desencriptada) al cliente cuando éste la necesite y evitamos poner el passphrase cada vez que nos conectemos. Por otro lado nos brinda mayor seguridad, por tener la llave almacenada en memoria y encriptada en el disco.

Cuando invocamos ssh-agent éste da como salida comandos shell para guardar datos que necesita en las variables de ambiente de la sesión. Por eso en bash vamos a invocarlo de la siguiente manera:

```
$ssh-agent bash
```

De esa forma es posible que ssh-agent ejecute los comandos en el bash creando las variables de ambiente que necesitamos para poder utilizarlo.

10.3.6.4 ssh-add

Una vez que el ssh-agent esta corriendo, necesitamos decirle que tenemos una llave privada, y

donde está.

```
$ ssh-add ~/.ssh/id_dsa
Enter passphrase for /home/user/.ssh/id_dsa:
Identity added: /home/user/.ssh/id_dsa (/home/user/.ssh/id_dsa)
```

Nos preguntara por nuestra frase-de-paso, la ingresamos, y es todo. Ahora puede ingresar al servidor remoto sin tener que entrar su password.

La única desventaja es que cada nueva instancia de ssh-agent necesita ser ejecutada por cada consola (shell) que se abra, esto significa que se debe correr ssh-agent cada vez en cada consola. Hay una solución para esto, con un programa o mas bien un script llamado keychain.

10.3.6.5 ~/.ssh/id_rsa and id_rsa.pub

Las claves generadas por ssh-keygen se almacenan por defecto en ~/.ssh/ y los ficheros id_rsa e id_rsa.pub contienen respectivamente las claves privada y pública para RSA.

10.3.6.6 ~/.ssh/id_dsa and id_dsa.pub

Las claves generadas por ssh-keygen se almacenan por defecto en ~/.ssh/ y los ficheros id_dsa e id_dsa.pub contienen respectivamente las claves privada y pública para DSA.

10.3.6.7 /etc/ssh/ssh_host_rsa_key y ssh_host_rsa_key.pub

El demonio de ssh es el programa que espera conexiones de red de los clientes ssh, controla la autenticación y ejecuta el comando requerido. El puerto por defecto en el que escucha es el 22 y su fichero de configuración es /etc/ssh/sshd_config.

En esta configuración se indica también la ruta en la que encontrar las claves que identifican nuestro servidor. Estas son la base de la autenticación mediante clave publicaLas claves correspondientes al servidor. Esta claves se almacenan por defecto en /etc/ssh/ y los ficheros ssh_host_rsa_key e ssh_host_rsa_key.pub contienen respectivamente las claves privada y pública para RSA.

10.3.6.8 /etc/ssh/ssh_host_dsa_key y ssh_host_dsa_key.pub

El demonio de ssh es el programa que espera conexiones de red de los clientes ssh, controla la autenticación y ejecuta el comando requerido. El puerto por defecto en el que escucha es el 22 y su fichero de configuración es /etc/ssh/sshd_config.

En esta configuración se indica también la ruta en la que encontrar las claves que identifican nuestro servidor. Estas son la base de la autenticación mediante clave publicaLas claves correspondientes al servidor. Esta claves se almacenan por defecto en /etc/ssh/ y los ficheros ssh_host_dsa_key e ssh_host_dsa_key.pub contienen respectivamente las claves privada y pública para RSA.

10.3.6.9 ~/.ssh/authorized_keys

Cada línea del fichero authorized_keys (habitualmente en ~/.ssh/authorized_keys) contiene una clave pública. Tiene el siguiente formato: Cada entrada va en una sola línea. Los campos se separan por comas. Las líneas en blanco y las que empiezan por # se ignoran.

- Primero van las opciones, separadas por comas, por ejemplo:
 - from="pattern-list" El servidor sshd puede limitar que máquinas pueden conectarse a la máquina usando wrappers TCP, pero no a nivel de usuario. Utilizando esta opción

es posible limitar las conexiones para una identidad a un conjunto específico de máquinas. Los hosts se separan por comas y pueden contener comodines como * y ?. Se pueden rechazar hosts específicos prefijándolos con !. Por ejemplo:

```
from="!enemy.my_isp.net,*.my_isp.net,home.example.com"
```

- command="command"
- environment="NAME=value" Se usa si se quiere tener un entorno específico cuando se usa esta clave. Por ejemplo podemos poner un \$PATH restrictivo para que no pueda ejecutar ciertos comandos, cambiar \$HOME, etc.
- Si se tiene varias personas accediendo vía diferentes identidades a la misma cuenta - usando tal vez un comando forzado - puede ser útil establecer la variable \$LOGNAME de manera que identifique quién se conecta.
- nopty Este campo es opcional. Su presencia se detecta viendo si la línea comienza por un número o no
- La clave pública: bits, exponente y módulo
- Un comentario

10.3.6.10 /etc/ssh_known_hosts

Cuando un cliente se conecta con el server este le pasa sus claves públicas, el cliente compara la clave del host con una base de datos (un archivo de texto) para ver si es correcta. Las claves se guardan en /etc/ssh_known_hosts (global) y en ~/.ssh/known_hosts. Si la clave no coincide o jamás se ha establecido una conexión contra ese server el cliente pide confirmación al usuario antes de continuar.

10.3.6.11 ~/.gnupg/*

Las claves generadas por gnupg se almacenan por defecto en este directorio. También se crea un archivo de configuración llamado *~/.gnupg/gnupg.conf* donde se almacenan las preferencias del usuario.

~/.gnupg/secring.gpg

Anillo donde se guardan las claves privadas.

~/.gnupg/secring.gpg.lock

Archivo de bloqueo de las claves privadas

~/.gnupg/pubring.gpg

Anillo donde se guardan las claves públicas

~/.gnupg/pubring.gpg.lock

Archivo de bloqueo de las claves públicas.

~/.gnupg/trustdb.gpg

Base de datos de confianza

~/.gnupg/trustdb.gpg.lock

Archivo para la base de datos de confianza

~/.gnupg/random_seed

Fichero que permite mantener el estado del generador de datos aleatorios.