Administración de Sistemas Linux

Juan Carlos Pérez Darias

CAPÍTULO 1
ADMINISTRACIÓN DE SISTEMAS LINUX

Juan Carlos Pérez
Universidad de La Laguna

ÍNDICE

1.. INTRODUCCIÓN AL SISTEMA OPERATIVO LINUX

1.1.. Antecedentes históricos

En 1991, Linus Benedict Torvalds cursaba el segundo curso de Informática en la Universidad de Helsinki. En esos momentos, dos sistemas operativos copaban el mercado informático. Por una parte, el sistema DOS, comprado por Bill Gates a un hacker de Seattle por 50 dólares, dominaba el mercado de los ordenadores personales gracias a su estrategia de marketing. El resto del mercado era absorbido por los sistemas UNIX que, desarrollado por los Laboratorios Bell para entornos de grandes máquinas de uso compartido, no era accesible para los usuarios de ordenadores personales debido a su alto precio. Con la aparición de MINIX, un sistema operativo escrito por el profesor holandés Andrew S. Tanembaum para enseñar a los alumnos el funcionamiento interno de un sistema real, parecía que una alternativa al DOS había surgido. A pesar de que no era un sistema completamente safisfactorio, su principal ventaja era la disponibilidad del código fuente. Esto permitió que, por primera vez, curiosos de las interioridades del sistema y hackers pudieran leer las cerca de 12000 líneas de código C y ensamblador del sistema operativo.

Uno de estos lectores fue Linus Torvalds que, partiendo de la idea de este sistema diseñado como una herramienta docente, se propuso añadirle las funcionalidades que los profesionales demandaban. Además, en esa época, los programadores alrededor del mundo estaban inspirados por el proyecto GNU de Richard Stallman, un movimiento que proclamaba el software libre. Con estos antecedentes, el 25 de Agosto de 1991, apareció el mensaje enviado por Linus al grupo de noticias de MINIX:

```
Hello everybody out there using minix - I'm doing a (free) operating system
(just a hobby, won't be big and professional like gnu) for 386(486) AT clones....
```

Este mensaje marcó el inicio de una nueva etapa en los sistemas informáticos pues, gracias a la inclusión del mismo en el proyecto GNU, cientos de miles de desarrolladores alrededor del mundo han contribuido a que este sistema operativo dejara de ser un juguete de hackers o una alternativa de bajo coste a UNIX para estudiantes, a un serio competidor en un mercado dominado por Microsoft.

1.2.. Sumario de distribuciones Linux

Normalmente la gente conoce a Linux como un paquete entero consistente en un conjunto de aplicaciones y herramientas tales como herramientas de desarrollo, editores, utilidades de red, aplicaciones multimedias, y cosas de este tipo. Sin embargo, si hablamos de una manera más formal, tales paquetes constituyen lo que se conoce como distribuciones y podemos considerarlas como "todo lo que necesita Linux".

Hablando con más propiedad, podríamos decir que Linux es el núcleo (también conocido como kernel) de un sistema operativo tipo UNIX de distribución libre. Este núcleo no es más que un programa muy complejo que actúa como jefe de operaciones del sistema, controlando todas las acciones vitales para el mismo. Así, es responsable del arranque y la parada de los otros programas, de controlar el acceso a los periféricos tales como discos o unidades de red, o de gestionar la memoria. Este programa, que es el mismo para todas las distribuciones, ofrece la mayoría de los beneficios de los entornos UNIX a un coste muy inferior a un sistema UNIX tradicional. Estas ventajas incluyen las siguientes:

- Multitarea: Linux es un sistema operativo multitarea que permite la ejecución de un gran número de aplicaciones simultáneas, lo que lo hace un sistema ideal para servidores corporativos o estaciones de trabajo de alta carga.
- Multiusuario: Linux permite a múltiples usuarios validarse en un mismo sistema, lo que da la posibilidad de que muchos usuarios puedan usar de forma simultánea el sistema.
- Estable: Los sistemas linux son famosos por su extremada estabilidad. Es usual encontrar sistemas que llevan ejecutándose durante meses sin la necesidad de re-arrancarlos.
- Sencillo de gestionar: Al igual que ocurre en UNIX, los sistemas Linux pueden ser controlados y administrados desde una línea de comandos sin necesidad de recurrir a herramientas gráficas que, por lo general, suelen ser poco eficientes. Además, un sistema Linux puede ser gestionado de forma remota.

Al ser el mismo núcleo, lo que diferencia una distribución de otra es el conjunto de herramientas de "valor añadido" que vienen con cada una. A continuación se presentan las características de las distribuciones más populares en la actualidad. Aunque, en general, todas tienen algo que ofrecer en el ámbito de los servidores, algunas de ellas están más enfocadas a su utlización en la empresa que otras. Para la elección de la distribución que mejor se adapte a nuestras necesidades, es necesario buscar aquella que proporcione un mejor balance entre utilidades, productividad y soporte.

- Debian GNU/Linux Esta distribución puede ser considerada como una de las más versátiles, pues a los más de 4000 paquetes software que complementan al núcleo, hay que añadir que soporta más plataformas hardware que ninguna otra distribución. Además, la instalación y desinstalación de aplicaciones se puede realizar de forma muy secilla con la herramienta dselect. Sin embargo, su falta de documentación y el proceso de instalación hace menos atractiva esta distribución para usuarios que se están incorporando al mundo Linux, mientras que los usuarios de organizaciones también pueden echar en falta un soporte formal para la resolución de problemas.
- Linux-Red Hat Esta distribución puede ser considerada como el standard de las plataformas Linux, ya que ocupa más de las dos terceras partes del mercado de las distribuciones Linux.

  El proceso de instalación, a pesar de no ser tan cómodo como en el caso de Linux-Mandrake, permite al usuario de una forma sencilla seleccionar el tipo de instalación que más se ajusta a sus necesidades: estación de trabajo, servidor, portátil ó a medida. Además nos da la posibilidad de instalar y configurar algunos servicios básicos de seguridad o de gestión de organizaciones como pueden ser los sistemas de seguridad firewalls o herramientas de centralización de cuentas de usuarios de gran utilidad para organizaciones que deseen minimizar el número de cuentas de usarios y passwords que los usuarios necesitan recordar. Además, incluye una herramienta para el manejo de aplicaciones, denominada RPM, que permite instalar, actualizar o desinstalar paquetes de aplicaciones de una forma muy sencilla.

  Podríamos decir que Red Hat es una solución Linux completa que incluye herramientas para la gestión del sistema y que facilitan la tarea del administrador, lo que lo hace muy atractivo para organizaciones de un tamaño considerable así como para pequeñas y medianas empresas.
- Linux-Mandrake Basada en Red Hat, podríamos considerar a esta distribución como el punto de inicio ideal para aquellos usuarios que se incorporan al mundo Linux, debido a su facilidad de instalación y a lo cómodo de su utilización. Así, la instalación se realiza de forma completamente gráfica y la selección de los paquetes a instalar se puede realizar de una forma intuitiva, abriéndonos la posibilidad de seleccionar grupos enteros de paquetes (paquetes de oficina, Internet, multimedia,...) o elegir paquetes individuales. Además, para los recién iniciados, nos ofrece la posibilidad de seleccionar instalaciones típicas de clientes o servidores, descargando así al usuario inexperto de la tarea de elegir los componentes de su sistema.

Después de seleccionar e instalar los paquetes, el programa de instalación permite configurar la red de una forma automática, detectando y configurando los parámetros necesarios para iniciar una conexión bien en una red de área local ó a través de un acceso telefónico.

Una vez instalado el sistema, la herramienta "Mandrake Control Center" permite configurar de una forma sencilla impresoras, monitores, tarjetas de sonido o video o tareas más complejas como la actualización de paquetes software o configuración avanzada de redes como el establecimiento de sistemas firewall. Todo esto con la posibilidad de un entorno gráfico tipo KDE o GNOME.

Además, Linux-Mandrake proporciona un sistema de soporte técnico. Una vez registrados, se obtienen 30 días de soporte técnico via MandrakeExpert, un fórum donde se pueden plantear cuestiones técnicas y recibir respuestas de expertos en Linux-Mandrake en un tiempo reducido.

- SuSE Linux Se podría decir que SuSE es la competencia europea a las distribuciones profesionales de Linux. Esta distribución, que está experimentando un gran auge en los últimos años, está orientada a su utilización en entornos de producción, con más de 2000 aplicaciones disponibles y aúna las características requeridas para este tipo de entornos: facilidad de instalación, seguridad, soporte multimedia y altas pestraciones, además de extensa documentación y un buen soporte técnico.
- Turbolinux A pesar de ser una de las distribuciones más antiguas, Turbolinx es el único de los sistemas linux comerciales que ofrece únicamente un sistema de instalación en modo texto. Además, para afrontar este proceso de instalación, es necesario que el usuario tenga alguna experiencia previa con Linux, con lo cual, a pesar de la buena documentación que la acompaña, esta distribución puede no ser la adecuada para aquellos usuarios iniciándose en este sistema operativo. En cuanto a la configuración y mantenimiento del sistema, Turbolinux no presenta una única herramienta tipo Linuxconf o Webmin presentes en las distribuciones anteriormente descritas que permita la gestión "centralizada" del sistema. Estos factores sugieren una vez más, que esta opción no es la recomendable para aquellos usuarios que se introducen a Linux.
- Caldera OpenLinux Esta distribución está orientada a la resolución de problemas concretos, como puede ser el caso de servidores de aplicaciones específicas. Así, OpenLinux eServer es una solución ideal para entornos empresariales, especialmente para organizaciones con recursos limitados, administradores de sistema con poca experiencia o para hardware antiguo. En estos casos, aunque la distribución incluye un elevado número de paquetes de aplicaciones, sólo instala aquéllas que sean imprescindibles para la correcta ejecución de los servicios requeridos.

1.3.. Instalación de Red Hat Linux

Una vez hemos presentado las características de las distribuciones Linux más utilizadas en la actualidad, a partir de ahora vamos a describir cómo se realizan las distintas tareas de administración para una distribución en concreto, Red Hat. La elección de esta distribución la hemos realizado atendiendo a la presencia masiva que la misma tiene en el mercado de los sistemas Linux. Sin embargo, aunque las herramientas puedan diferir de unas distribuciones a otras, la administración en los sistemas Linux se basa en la configuración de ficheros de texto que, en general, coinciden para todas las distribuciones. Por tanto, en la medida de lo posible, intentaremos indicar los ficheros implicados en las tareas administrativas que se vayan presentando así como el formato de los mismos, con el fin de evitar la dependencia hacia una distribución en concreto.

Antes de iniciar la fase de instalación es importante tomarse un poco de tiempo para evaluar algunos aspectos básicos:

1. El hardware del sistema donde se va a realizar la instalación
2. Las funciones que va a desempeñar el sistema: servidor, cliente, ...
3. Coexistencia de Linux con otros sistemas operativos

1.3.1.. Comprobación del hardware

Antes de iniciar la instalación de Linux o de cualquier otro sistema operativo, deberíamos determinar cuál es la configuración hardware del sistema sobre el que vamos a trabajar. Es aconsejable crear un listado lo más preciso posible con la siguiente información para los distintos componentes hardware:

- Compañía que suministra el componente
- Modelo y/o número
- Configuración de interrupciones para ese dispositivo (plug & play, IRQ's,...)
- Rango de direcciones de Entrada/Salida
- Memoria disponible
- Cualquier otra característica relevante del componente

Una vez finalizado el inventario "hardware" de nuestro sistema, es aconsejable asegurarse que todos los componentes incluidos en el listado anterior son soportados por la distribución de Linux que estamos a punto de instalar. Para ello, basta consultar la última versión de la lista de compatibilidad hardware HCL (Hardware Compatibility List) que publica en su página Web cada organización que suministra Linux. Por ejemplo, para el caso de Red Hat, esta lista se encuentra en la siguiente URL: http://www.redhat.com/hardware. De esta manera, cualquier componente hardware no soportado por la distribución, puede ser reemplazado antes de intentar la instalación. Una sugerencia general, que se puede aplicar a todos los sistemas operativos, es evitar las configuraciones de software y hardware de última generación, especialmente cuando estemos instalando un sistema que vaya a desempeñar tareas de servidor.

Por otra parte, es conveniente disponer de una fuente de ayuda en el caso de que se plantee algún problema importante durante la instalación. Esta ayuda la podemos obtener de otro administrador de sistemas que tenga experiencia en Linux, del soporte técnico que viene incluido con algunas de las distribuciones o la localización de foros de discusión relacionados. Esta última alternativa puede ser una solución muy adecuada porque, muy probablemente, algún otro usuario tenga los problemas que se nos están presentando en este momento. En este sentido, y haciendo referencia a la distribución Red Hat, en la dirección: http://www.redhat.com/support/docs/faqs/rhl_general_faq/faq.html se puede encontrar multitud de preguntas y respuestas relacionadas con la instalación y funcionamiento de un sistema Linux.

1.3.2.. Planificación de la Instalación

La siguiente decisión que debemos tomar no es técnica, sino administrativa. Antes de proceder a la instalación, es aconsejable tomarnos uno minutos para meditar sobre las funciones que ha de desempeñar nuestro sistema: cliente, servidor de algunas aplicaciones, servidor de dominio, conexión a la red, nivel de seguridad requerido, ... Dependiendo de estas necesidades, el tipo de instalación a efectuar varía significantemente. Esta decisión toma especial importancia cuando el análisis de nuestra organización sugiere la instalación de un equipo actuando como servidor. En este caso, la estabilidad, disponibilidad y rendimiento se convierten en un asunto prioritario. Normalmente, estos factores se mejoran con la adquisición de hardware más potente, aunque esto no siempre tiene que ser cierto, pues en multitud de ocasiones con el hardware disponible y una sintonización adecuada del sistema se consiguen estos objetivos. Por ejemplo en Linux, una decisión que ayuda a mejorar estos factores consiste en el diseño de un servidor que no sea amigable con los usuarios que accedan directamente desde este equipo. Esto implica no usar entorno de ventanas, herramientas multimedia o navegadores web pues estas aplicaciones cargan en exceso a un sistema cuyos objetivos son el proporcionar un servicio a la organización en su conjunto. Además, es aconsejable determinar cuáles son las funciones específicas del servidor y a continuación deshabiliar el resto de las funciones o, incluso, recompilar el kernel para que éste sólo contenga aquellas características que necesita realmente el sistema que se va a instalar.

Otro aspecto importante del diseño de un servidor es el entorno donde se va a ubicar el mismo. Como administrador de sistemas, se hace imprescindible asegurarse de la seguridad física

de los servidores, manteniéndolos, siempre que sea posible, en una habitación aislada bajo llaves con una temperatura adecuada y un sistema de alimentación ininterrumpido (SAI) que eviten la pérdida de archivos como consecuencia de cortes inesperados de corriente.

Por último, es necesario realizar una evaluación del sistema antes de proceder a una nueva instalación. En caso de tener otro sistema operativo instalado, es altamente aconsejable realizar una copia de seguridad antes de iniciar la instalación de Linux. Posteriormente hay que decidir si queremos que Linux coexista con el sistema previamente instalado, o sea el único sistema presente en nuestro equipo. En el primer caso, necesitaremos hacer modificaciones a las particiones de disco de modo que hagan hueco suficiente para la Linux, usando alguna herramienta como fips de MS-DOS.

1.3.3.. Inicio de la Instalación

En esta sección describiremos los pasos necesarios para instalar Red Hat Linux. Lo primero que necesitamos es la distribución que deseamos instalar. Aunque el caso más normal y sencillo es el de obtener dicha distribución en soporte de CD-ROM, también es posible obtener la misma a través de Internet, con lo cual podemos crearnos nuestro propio CD-ROM o realizar la instalación desde el servidor donde residen los ficheros de la distribución (ej.: http://www.redhat.com/download/howto_download.html). De esta forma, los modos que soporta Linux para realizar la instalación son:

1. CD-ROM: Original de la distribución o imagen creada a partir de los ficheros que la conforman
2. Disco duro local: La imagen de la distribución se ha copiado previamente a nuestro disco local
3. NFS: La distribución se encuentra en un servidor de red
4. FTP: La instalación se realiza directamente a partir del servidor FTP

A continuación se empieza con el proceso de arranque. Tenemos dos posibilidades para iniciar este proceso: utilización de un disquete de arranque o un CD-ROM. Si nuestro sistema soporta arrancar desde CD, ésta es la opción más rápida. Sin embargo, si nuestro sistema no soporta arranque de CD-ROM o el CD-ROM que disponemos no es arrancable, necesitaremos crear un disco de arranque. Para ello, necesitameros copiar el fichero boot.img [1] a un disco. Este fichero se encuentra en el directorio images de la distribución y no puede ser copiado usando los comandos normales de copia de los sistemas operativos, sino usando una de las siguientes opciones:

- Creación desde MS-DOS

  La copia del fichero se realiza utilizando el comando rawrite (que se encuentra en el directorio dosutils de la distribución), siendo el fichero fuente boot.img y como destino a:

- Creación desde UNIX/Linux

  Utilizar el comando dd para realizar la copia.

  ```
  dd if=boot.img of=/dev/fd0
  ```

Una vez que se dispone del soporte para el arranque de la instalación (CD-ROM ó disco), el siguiente paso es el arranque del sistema desde el medio elegido. Se presentará una pantalla indicando "boot:". Si no se pulsa ninguna tecla o bien se pulsa ENTER, el proceso de instalación se inicia de forma inmediata. También se puede utilizar esta pantalla para seleccionar un modo alternativo de instalación: en modo texto o en modo experto, en el cual no se detectan los dispositivos instalados en el sistema sino que el usuario especifica los módulos encargados de su gestión.

A partir de aquí, van apareciendo una serie de pantallas que permiten al usuario configurar diversos módulos:

1. Selección del idioma a utilizar

[1] Si el modo de instalación elegido es cualquiera de las variantes de red indicadas anteriormente, es decir, FTP o NFS, la imagen del kernel que deberemos copiar al disco es bootnet.img en lugar de la boot.img.

2. Selección del tipo de teclado

3. Selección del ratón

4. Tipo de instalación: Actualización o instalación eliminando cualquier instalación previa. Esta pantalla, figura 1.1, permite seleccionar distintos tipos de instalación dependiendo de las funciones que vaya a realizar nuestro sistema. Red Hat Linux ofrece 3 clases diferentes de instalación:

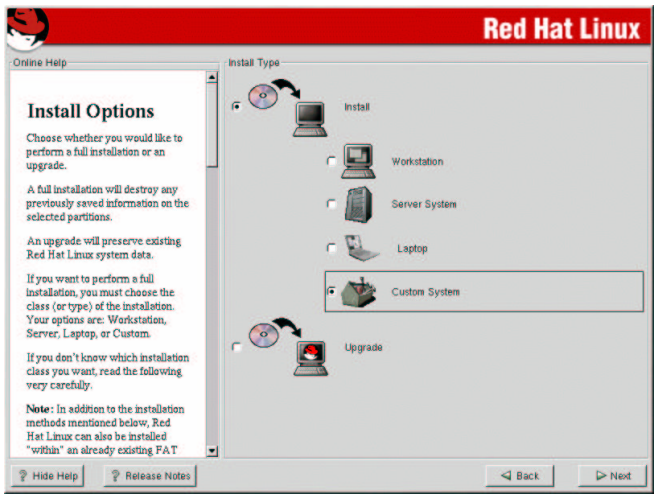

Figura 1.1: Pantalla de selección del tipo de instalación

- Workstation: Es la opción más apropiada para realizar nuestras primeras pruebas en el mundo Linux. Respondiendo a unas pocas preguntas durante la instalación, podemos tener ejecutando un sistema Linux en muy poco tiempo. Esta instalación configura los servicios básicos de un sistema cliente: entorno de ventanas, aplicaciones de oficina, multimedia,....
- Server: Esta opción permite la instalación rápida de un sistema servidor basado en linux, en la cual se configuran los servicios típicos de un servidor. Requiere de mucho espacio en disco (aproximadamente 1.7 GB) y por defecto no configura el sistema de ventanas X.
- Personalizada: Permite una instalación mucho más flexible que las otras modalidades. Ahora es el usuario el que debe tomar una serie de decisiones importantes durante la instalación: número de particiones a realizar y tamaño de las mismas, paquetes que se van a instalar en el sistema, utilización del cargador LILO,.... Es el tipo de instalación recomendada una vez el usuario se ha familiarizado con Linux. Requiere de un espacio mínimo de disco de aproximadamente 300MB aunque, dependiendo de la cantidad de paquetes seleccionados, pueden ser necesarios hasta 2GB de disco.

5. Creación de particiones para Linux: Durante el proceso de instalación de Linux es necesario determinar cómo particionar las unidades de disco de la mejor manera posible. Los sistemas Linux pueden ser instalados como sistema operativo único en el ordenador o en situación de arranque dual con Windows o cualquier otro sistema operativo.

   Para entender las diversas estrategias de particionado, es necesario conocer la forma en que Linux gestiona múltiples particiones. Como es sabido, en DOS y Windows, cada partición lleva asignada una letra de unidad independiente que se utiliza para el acceso a la citada partición. Cada una de estas unidades tiene un directorio raíz independiente y todos los demás directorios son subdirectorios en la estructura jerárquica de la unidad. Sin embargo, en Linux, todas las particiones son accesibles a partir de una estructura jerárquica única que

empieza en el directorio raíz "/" y continúa a través de un árbol de subdirectorios. De esta forma, cada partición se encuentra montada en una situación específica dentro del árbol con lo que puede ser accedida a partir de este "punto de montaje".

Estrategias para la creación de particiones

Para poner en funcionamiento un sistema Linux, la estrategia de particionado más sencilla consiste en la creación de dos particiones: una Linux montada como la partición raíz del sistema y una partición de intercambio o swap que es usada como una extensión a la memoria principal del equipo. Sin embargo, esta estrategia simplista es cada vez menos usada debido a que las actualizaciones del sistema o la gestión de espacio libre adicional se hacen más complejas, además de los problemas de eficiencia derivados del cuello de botella que se forma como consecuencia de que todo el tráfico de disco propio de las operaciones del sistema operativo, de las aplicaciones y de los datos de usuario ocurren en la misma partición.

Por tanto, es recomendable dividir el/los disco/s en distintos bloques lógicos. El número de ellos depende de la función a desarrollar por el sistema operativo aunque, para la mayoría de los usuarios, es aconsejable separar la instalación de Linux en dos o tres particiones aparte de la partición de swap imprescindible para un correcto funcionamiento. Un esquema típico de particionado para un sistema Linux concebido como servidor sería el siguiente:

- "/": Es donde se sitúa el directorio raíz del sistema de ficheros.
- "/usr": Es donde se sitúan normalmente todos los archivos de programas (El equivalente al directorio c:
  Archivos de Programas de Windows).
- "/home": En esta partición se sitúan los directorios de todos los usuarios. Manteniendo este directorio en una partición física separada, a medida que el tamaño de este directorio va creciendo, se pueden migrar estos directorios a particiones más grandes sin afectar el correcto funcionamiento del sistema operativo o de las propias aplicaciones instaladas. Además, si la partición home se encuentra situada en otro disco físico, la carga general del sistema disminuirá, mejorando por tanto la eficiencia general.
- "/boot": Esta partición, cuyo tamaño no necesita superar los 32MB, contiene el núcleo del sistema operativo así como algunos ficheros utilizados durante el arranque del sistema. Debido a las limitaciones de gran parte de las BIOS de los PC's, que no permiten acceder más allá del cilindro 1024, es recomendable reservar esta pequeña partición con este propósito.
- "/var": Este directorio es el destino de los archivos que se encargan de la auditoría del sistema. Debido a que estos archivos pueden verse afectados por usuarios externos, es recomendable situarlos en una partición para que, en caso de sufrir un ataque de tipo Denegación de Servicio que hace crecer de forma violenta los archivos de registro (archivos .log), este crecimiento no afecte a los datos situados en otras particiones.
- "swap": Éste es un sistema de archivos no accesible directamente para el usuario. Es usado como memoria virtual en el cual se almacenan las páginas de memoria física que han de ser liberadas a medida que la memoria se va llenando. Las siguientes consideraciones han de tenerse en cuenta a la hora de fijar el tamaño de esta partición:
  - No se pueden configurar más de 8 particiones de swap
  - Algunas versiones de Linux limitan el tamaño máximo de las particiones de swap a 128MB
  - En un servidor que soporte mucha carga, es recomendable situar esta partición en un disco separado para aumentar las prestaciones del sistema
  - Debido al coste actual de los discos, es preferible pecar de exceso que por defecto a la hora de designar el tamaño de esta partición, principalmente si observamos que la memoria principal de nuestro sistema no es suficiente para la ejecución de todos
  - Como posible estrategia a seguir, el espacio de swap puede ser configurado con un tamaño del doble de la memoria RAM disponible si ésta es menor de 128MB o del mismo tamaño de la RAM si es mayor de 128MB.

Creación de las particiones

Históricamente, la mayoría de las distribuciones Linux han usado el programa fdisk para realizar las particiones. Aunque algunas distribuciones han creado sus propias herramientas para esta función, fdisk sigue siendo la herramienta estándar y se encuentra disponible en todas las distribuciones. Además, entendiendo el funcionamiento de fdisk es fácil usar cualquiera de las otras aplicaciones.

fdisk permite crear, borrar y modificar particiones usando una interfaz de texto muy simple basada en menús. Las operaciones más comunes que se pueden realizar usando fdisk son:

- Creación de una partición primaria
- Creación de una partición lógica. En este caso, es necesario haber creado con anterioridad una partición extendida para que contenga una o varias particiones lógicas.
- Mostrar las particiones existentes
- Establecer el tipo de la partición: Cada partición debe tener un tipo asociado. Por defecto, todas las particiones son creadas como particiones Linux, pero si se pretende usar una partición como área de swap o una partición con el formato DOS-Windows, es necesario cambiar su identificador.
- Borrar una partición existente
- Actualizar los cambios realizados a disco.

Si se opta por la utilización de una herramienta de particiones, RedHat ha desarrollado Disk Druid como una forma fácil de crear particiones y asociarlas a los directorios donde se pretende realizar su punto de montaje. Cuando se arranca Disk Druid durante la instalación, se presenta una pantalla como la mostrada en la figura 1.2. En ella se muestran todas las particiones existentes en cada uno de los discos. Para cada una de estas particiones se presenta la siguiente información:

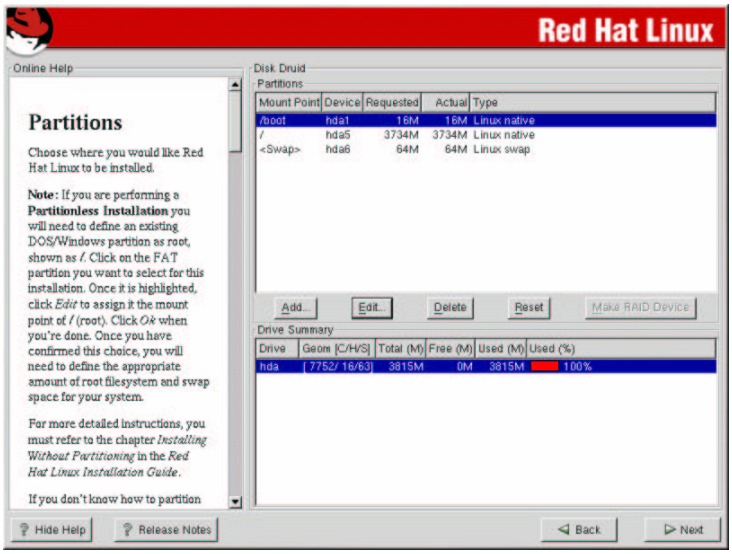

Figura 1.2: Pantalla de disk druid

- Punto de montaje: Indica la posición donde se monta la partición.
- Dispositivo: Unix, en general, asocia cada partición con un dispositivo. Estos dispositivos, que aparecen en el sistema de ficheros como "ficheros normales", se usan para acceder a los dispositivos correspondientes. Así, asumiendo que nuestro sistema está compuesto por varios discos IDE, la sintaxis de los dispositivos asociados sería de la forma

/dev/hdXY, donde X toma los valores a,b,c,d para indicar, respectivamente, el disco maestro de la controladora primaria, el disco esclavo de esta misma controladora, y los discos maestros y esclavos de la segunda controladora. Y indica el número de la partición de disco. Por ejemplo, /dev/hda1 hace referencia a la primera partición del disco maestro de la controladora primaria del sistema.

- Requerido: indica el tamaño original de la partición
- Actual: Muestra el espacio actualmente asignado a esta partición
- Tipo: Tipo de la partición (Nativo linux, FAT, VFAT, NTFS, ...

La segunda mitad de la pantalla muestra el sumario de la unidad. Cada línea presenta una unidad simple y sus características. La información mostrada consiste en:

- El nombre de la unidad (no mostrando el "/dev")
- La geometría del disco en formato Cilindros/Cabezas/Sectores
- Tamaño total del disco
- Cantidad de disco ocupado (particionado)
- Cantidad de disco disponible que todavía se puede particionar

Por último, en la parte central de la pantalla se encuentran los botones que permiten realizar las operaciones sobre las particiones: añadir, modificar o borrar.

6. Formateo de las particiones: Permite seleccionar aquellas particiones que requieren ser formateadas durante la instalación. Inicialmente es conveniente seleccionar la opción "buscar bloques defectuosos" pues permite al sistema no utilizar estos bloques defectuosos para el almacenamiento de información.

7. Instalación de LILO: LILO es el gestor de arranque de Linux. Este programa, que permite el arranque de distintos sistemas operativos puede ser ubicado en el MBR (Master Boot Record) que es lo primero que se lee cuando se terminan los testeos automáticos de hardware y se pasa el control al software, o en la partición root, lo que hace que se ejecute el gestor de arranque ubicado en el MBR y que, opcionalmente pase el control a LILO.

8. Configuración de la red A continuación RedHat permite configurar las interfaces de red así como los protocolos que se van a utilizar para la comunicación entre los distintos equipso en la red. Esta pantalla nos permite elegir una configuración "manual" o mediante "DHCP", un protocolo que permite la gestión de la red de un equipo de forma remota.

9. Creación de cuentas: Aparte de la cuenta de administrador conocida como root, la cual ha de ser protegida con una contraseña "fuerte", en esta fase de la instalación se nos permite generar nuevos usuarios de una forma muy cómoda.

10. Selección de los grupos de paquetes: En este punto es donde se realiza la selección de los paquetes que se van a instalar en el sistema. Esto se puede realizar eligiendo "grupos de paquetes" a instalar (figura 1.3) en los que la selección detallada de los paquetes la realiza la propia distribución o realizando una selección individual de paquetes (figura 1.4) en la que el usuario decide la instalación de cada paquete de forma individualizada.

11. Configuración de X-Windows: X-Windows es la interfaz de usuario gráfica de Linux. Los programas tales como KDE o GNOME usan X-Wiondows como un mecanismo estándar para comunicarse con el hardware de vídeo. Como se comentó previamente, lo que hace interesante a X-Windows es que no está unido de forma rígida con el sistema operativo, lo que supone que la caída del sistema de ventanas no implica la caída del sistema (figura 1.5).

    El proceso de instalación intentará detectar el tipo de tarjeta gráfica y de monitor que el usuario esté utilizando aunque también se le permite a éste introducir de forma manual los identificadores de estos dispositivos. También se le ofrece al usuario la posibilidad de configurar el sistema de tal modo que se cargue el sistema de ventanas al arrancar el sistema.

Figura 1.3: Pantalla de selección de grupos de paquetes

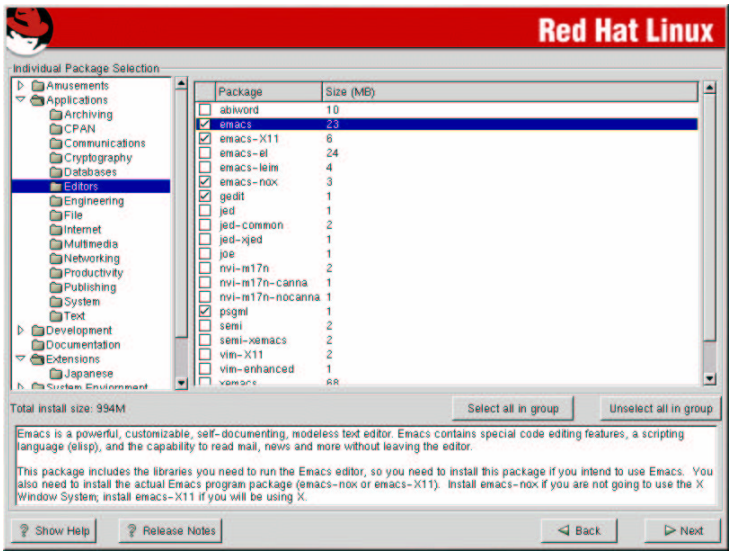

Figura 1.4: Selección de paquetes individuales

12. Creación de un disco de arranque: Es altamente recomendable la creación de un disco de arranque en esta fase de la instalación. Este disco permitirá arrancar el sistema en caso de fallo, pudiendo reconfigurar o reiniciar los componentes que den problemas.

A partir de aquí, el proceso de instalación dará formato a las particiones indicadas para ello y, a continuación, instalará los paquetes seleccionados. Si toda ha funcionado de forma correcta durante estos proceos, la máquina Linux pasará a ejecutarse normalmente tras su reinicialización.

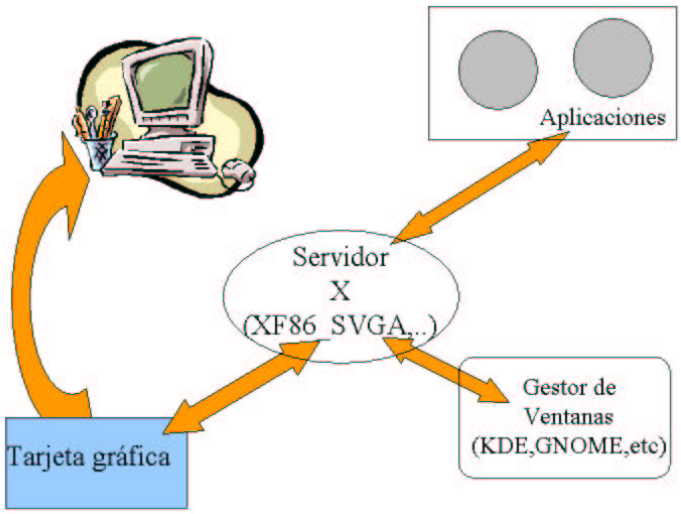

Figura 1.5: Relación de los distintos componentes que intervienen en el entorno gráfico

2.. ADMINISTRACIÓN BÁSICA DE UNIX/LINUX

2.1.. Gestión de usuarios y grupos

Los usuarios y los grupos de usuarios forman la base de la seguridad y del control de acceso en los sistemas UNIX. Para entrar en el sistema (login), cada usuario requiere de una cuenta que lo identifica de forma única. Cada una de estas cuentas lleva asociada una contraseña, un directorio de trabajo y un grupo principal. Desde el punto de vista del sistema operativo, estas cuentas de usuario están representadas por el número de identificación UID. Estos UID's son utilizados por el núcleo para determinar los derechos de acceso de cada usuario a los recursos del sistema.

2.1.1.. Cuentas de usuarios. El fichero /etc/passwd

Las cuentas de usuarios se almacenan en forma de base de datos en el fichero /etc/passwd. Este fichero contiene una entrada por cada usuario en el sistema y en cada entrada, aparte del nombre y su UID, se almacena información que permite reconocer a ese usuario. Cada entrada se almacena en una línea separada y cada línea contiene una serie de campos separados por ":". La sintaxis de cada entrada es de la forma:

<nombre de usuario>:<contraseña>:<UID>:<GID>:<Nombre completo>:<directorio de trabajo>:<shell de arranque>

donde el significado de cada campo es el siguiente:

- nombre de usuario: El nombre que el usuario ha de introducir para entrar en el sistema. Es la identificación "amigable" de la cuenta de usuario. Estos nombres no pueden contener espacios en blanco y deben empezar por una letra o un número.
- contraseña: Este campo contiene una versión encriptada de la contraseña del usuario. Normalmente, estas contraseñas son generedas usando el comando passwd
- UID: Es el número de identificación de usuario usado por el sistema operativo para identificar la cuenta.
- GID: Cada usuario en el sistema debe pertenecer al menos a un grupo por defecto. Este grupo debe ser una entrada de grupo válida en el fichero /etc/group. Al igual que ocurre con el número de identificación de usuario, el sistema operativo identifica cada grupo con un número de identificación de grupo, siendo este valor el que se almacena en la entrada correspondiente del fichero /etc/passwd.

- Nombre completo: Para cada usuario se puede especificar un nombre completo. Este nombre puede ser usado por aplicaciones para identificar en "lenguaje humano" a un usuario. Por ejemplo, cuando se envía un e-mail, se puede incluir en el campo "desde" el nombre completo del usuario, no sólo su nombre de usuario.
- directorio de trabajo: Cada cuenta necesita un directorio de trabajo, que es el lugar por defecto donde el usuario puede almacenar sus ficheros y crear subdirectorios. La situación de estos directorios "home" es completamente arbitraria, aunque es función del administrador usar una política "consistente" para determinar su ubicación.
- shell de arranque: Cuando un usuario entra en el sistema, se le proporciona un programa conocido como intérprete de comandos o shell que permite al usuario ejecutar sus comandos. Normalmente, Linux usa la shell bash por defecto, pero usando esta entrada se puede especificar una opción diferente.

Aunque el fichero /etc/passwd es un fichero de texto plano, debido a la importancia que este fichero tiene en el sistema, no es recomendable editarlo directamente, sino utilizar cualquiera de las herramientas o comandos implementados para realizar esta función.

Selección de los UID's. Como se comentó anteriormente, el sistema identifica a los usuarios a partir del número de identificación o UID. Por tanto, es conveniente que el administrador de sistemas diseñe una política de asignación de UID's que garantice la unicidad del mismo para cada usuario y que permita mantener las cuentas de usuario organizadas de una forma lógica.

Los UID's son valores enteros que cubren el rango 0-65534. Estos identificadores no tienen que ser asignados de forma secuencial con lo que se puede establecer un esquema de organización interna para el uso de los mismos. De esta manera podríamos segmentar el rango total de identificadores de tal manera que refleje la estructura de la organización.

Generalmente, los UID's por debajo de 100 se reservan para cuentas del sistema tales como root o bin y para cuentas creadas por algunas aplicaciones tales como ftp y gdm.

Para el resto de las cuentas, algunas distribuciones adoptan un número base (p. ej. 500 en Red Hat) e incrementan el número en 1 para cada nueva cuenta. Sin embargo, si nuestra organización consta de un gran número de usuarios, resulta útil segmentar el rango disponible de UID's en grupos lógicos que de alguna manera se relacionen con los grupos dentro de la organización. Por ejemplo, podremos segmentar el espacio de identifcadores en grupos discretos de 100 o 1000 basados en la situación geográfica de los usuarios, de los departamentos, o incluso alfabéticamente por el apellido.

Situación de los directorios home. Una reflexión análoga a la distribución de los identificadores la podemos hacer para la ubicación de los directorios home de los usuarios. Generalmente, los sistemas Linux sitúan estos directorios bajo el subdirectorio /home. Así, los directorios de trabajo de cada usuario se nombran como */home/<nombre de usuario>*.

Al igual que los identificadores, este esquema puede ser el adecuado para organizaciones con un número limitado de usuarios. Sin embargo, cuando el número de usuarios empieza a ser importante, presenta algunas limitaciones. Por ejemplo, este esquema indica que todos los directorios se encuentren en una misma partición de un disco, lo que puede llevar a problemas de capacidad de almacenamiento, pues es necesario dividir el espacio de la partición entre todos los usuarios.

Por tanto, el administrador del sistema, en función del número de usuarios de su organización y del tamaño de las particiones dedicadas a los datos de los usuarios, ha de planificar una segmentación de los directorios home siguiendo una estructura lógica. Por ejemplo, para un departamento consistente en diversos grupos de investigación, cada uno de ellos con distintos requerimientos de espacio en disco, el administrador puede plantearse la siguiente estructura para ubicar los directorios home de los usuarios:

- /home/orgánica
- /home/inorgánica
- /home/molecular

Con esta división, el usuario pepito perteneciente al grupo de química orgánica tendría su directorio de trabajo en /home/orgánica/pepito, que puede encontrarse en una partición separada de la de los datos de los usuarios de molecular.

2.1.2.. Grupos de usuarios. Fichero /etc/group

Los usuarios del sistema pueden establecer relaciones con otros usuarios. Con este fin, en Unix se establece el concepto de grupos de usuarios. Estos grupos definen conjuntos de usuarios con similares privilegios en el sistema. Esto simplifica en gran medida la administración del sistema, permitiendo la aplicación de restricciones comunes de seguridad y control de acceso a los ficheros, directorios y servicios. Al igual que con los usuarios individuales, la gestión de estos grupos se realiza a través de un fichero de texto: /etc/group. Este fichero consiste de un conjunto de entradas de la forma:

<nombre de grupo>:<contraseña>:<GID>:<lista de usuarios>

siendo el significado de los distintos campos el siguiente:

- nombre de grupo: El nombre por el cual se identifica al grupo.
- contraseña:Aunque generalmente los grupos no requieren contraseñas, el "jefe" del grupo puede asignarle una para su administración.
- GID:Número de identificador de grupo. Es el número por el cual el sistema identifica al grupo.
- lista de usuarios: Lista de usuarios separados por "," que indican los usuarios que son miembros del grupo.

Selección de los GID's. Siguiendo la misma filosofía que los UID's, los GID's son valores enteros en el rango 0-65534. Cada grupo debe poseer un único número en este rango. La forma de asignar los valores a los grupos depende de la estrategia a seguir en la organización. Las distribuciones actuales de Linux adoptan dos filosofías para la asignación de grupos. Por una parte, algunas distribuciones como Red Hat, crean por defecto un grupo por cada usuario y este grupo pasa a ser el grupo principal de ese usuario. Por ejemplo, el usuario usu1 con el UID=550 tendrá como grupo por defecto el usu1 con GID=550. Esto proporciona un mecanismo robusto de seguridad puesto que no permitirá a otros usuarios en el sistema acceder a los recursos privados del usuario.

Por otra parte, otras distribuciones crean un único grupo usuarios y asignan automáticamente a todos los nuevos usuarios a este grupo. Esto hace posible asignar, de una forma sencilla, permisos para que todos los usuarios puedan accedar a ciertos ficheros del sistema.

Sin embargo, probablemente ninguna de las alternativas sea la adecuada para nuestra organización, en cuyo caso debemos planificar nuestra estrategia para la creación de grupos y la asignación de usuarios a los mismos. Por ejemplo, parece adecuado crear grupos por departamentos o unidades de investigación y asignar ese grupo a todos los usuarios de la unidad, puesto que estos usuarios necesitarán compartir datos o recursos entre sí, pero no con los usuarios de otros departamentos. Además, dependiendo de la complejidad de nuestra organización, puede ser adecuado asignar usuarios a varios grupos.

2.1.3.. Herramientas de administración de usuarios y grupos

Como hemos comentado en las dos secciones anteriores, tanto la base de datos de usuarios como la de grupos se encuentran en ficheros de texto con un formato bien definido. Esto permite que los administradores puedan desarrollar sus propias herramientas para la gestión de usuarios. Por ejemplo, un administrador puede crear una herramienta que permita integrar la gestión de las cuentas de usuarios con el resto de la infraestructrua de la organización. Por ejemplo, es posible actualizar, al dar de alta un usuario, otra información como la dirección, teléfonos corporativos, seguridad social,.... Por lo tanto, no es difícil encontrar en las distribuciones, o en la propia Internet, un conjunto de herramientas que facilitan la gestión de las cuentas de usuarios. Sin embargo, aunque muchas de estas herramientas presentan una interfaz gráfica muy amigable, no

son estándar en todas las distribuciones de Linux. Por tanto, en esta sección nos centraremos en la sintaxis de diversos comandos, presentes en todas las distribuciones, destinados a tal fin.

La creación de cuentas de usuarios es una tarea que ha de realizar el administrador usando la cuenta privilegiada de usario root. El proceso de creación de una cuenta de usuario consta de varios pasos:

- Seleccionar un UID
- Seleccionar uno o varios grupos para el usuario
- Crear la entrada en el fichero /etc/passwd
- Establecer la contraseña para el usuario
- Crear el directorio de trabajo
- Incluir en el directorio de trabajo la información necesaria para el usuario.

El comando useradd facilita esta labor permitiendo, en una sola operación, realizar todas las tareas anteriores. En su forma más simple, la sintaxis de este comando es:

```
#useradd usuario
```

En la mayoría de los sistema Linux, este comando creará la cuenta usuario de la siguiente forma:

- Crea la entrada en el fichero /etc/passwd dejando el campo de contraseña vacío
- Se le asigna automáticamente un UID siguiendo la política de la distribución
- Se incluye la cuenta en un grupo siguiendo la política de la distribución
- Se crea un directorio de trabajo para el usuario y se copian los contenidos del directorio /etc/skel en él.

Si se pretende modificar cualquiera de los parámetros se puede usar la opción correspondiente del comando useradd (ver el manual del comando: `#man useradd`).

También es posible cambiar la configuración por defecto para los nuevos usuarios. El comando `#useradd -D` nos permite ver la configuración que se utilizará en la creación de usuarios si no se especifica ningún parámetro al comando. Modificando estos valores de configuración en el fichero de texto /etc/default/useradd posibilita que los usuarios que se creen a partir de ese momento adopten estos nuevos parámetros (directorio de trabajo, shell de inicio, grupo principal,...)

Otros comandos útiles para la gestión de usuarios y grupos son los siguientes (para ver sintaxis, mirar la página del manual correspondiente):

- userdel: Permite eliminar un usuario del sistema. Esto incluye la eliminación del directorio de trabajo de este usuario.
- usermod: Permite modificar los atributos de la cuenta de un usuario
- groupadd: Crea una nueva cuenta de grupo
- groupdel: Elimina un grupo del fichero /etc/group
- groupmod: Modifica los atributos de un grupo.
- passwd: Cambia la contraseña de un usuario.

2.1.4.. Eliminación y desactivación de cuentas de usuarios

Si se decide en la organización que un usuario no debe tener acceso al sistema a partir de un momento determinado o debe ser suspendido temporalmente de usar el sistema, es necesario eliminar o desactivar la cuenta que identifica a ese usuario. La desactivación de una cuenta hace imposible que un usuario pueda entrar (login) en el sistema, aunque toda su información siga permaneciendo intacta. Sin embargo, el borrado de una cuenta elimina toda la información relacionada con el usuario. En este caso, si el usuario quisiera acceder al sistema posteriormente, sería necesario crear una nueva cuenta de usuario. Por tanto, a menos que sea segura la no utilización de una cuenta en el futuro, es preferible la desactivación frente al borrado.

Para desactivar una cuenta, es imprescindible que el usuario no pueda entrar en el sistema con ninguna contraseña, incluyendo la contraseña actual del usuario. Para conseguir esto, basta con incluir un "*" o un "!" en el campo de la contraseña del usuario del fichero /etc/passwd. Estos caracteres no pueden ser generados por el algoritmo de encriptación utilizado para crear las contraseñas, por lo que cualquier intento de entrar en el sistema será fallido. Para reactivar la cuenta basta con eliminar este carácter y el usuario puede usar su misma contraseña para darse de alta. Otra solución consiste en cambiarle temporalmente la shell de inicio de modo que la nueva "shell" sólo muestre un mensaje al usuario indicando el motivo de la desactivación y a continuación vuelva a salir sin darle la oportunidad de ejecutar ningún comando.

Para eliminar una cuenta de usuario, es conveniente realizar una copia de seguridad de los datos pertenecientes al mismo, ejecutar el comando userdel y asegurarse que no quedan ficheros cuyo propietario sea este usuario (comando find).

2.1.5.. Seguridad del sistema. La opción shadow

El modelo estándar de almacenamiento de las cuentas de usuarios en los sistemas Unix/Linux que acabamos de describir presenta dos grandes inconvenientes:

1. Las contraseñas almacenadas en el fichero /etc/passwd son codificadas usando el comando crypt disponible en todos los sistemas Unix. En la actualidad, con la potencia de cálculo de los ordenadores personales y la habitual debilidad de las contraseñas de usuarios, basadas en palabras de diccionario o en nombres propios, es cuestión de tiempo la decodificación de estas contraseñas por parte de algún usuario malintencionado.

2. El fichero /etc/passwd contiene, además de la contraseña encriptada, información tal como los UID's, GID's o los directorios de trabajo de los usuarios, que debe ser accesible por parte de todos los usuarios del sistema. Por tanto, aunque un usuario "normal" no pueda alterar directamente este fichero, si puede copiar los contenidos del mismo y proceder a la decodificación de las contraseñas sin más que aplicar un procedimiento de "fuerza bruta".

Este problema de seguridad ha llevado a los diseñadores de Linux a adoptar una serie de medidas que minimicen este problema. Estas medidas consisten en:

- Usar un esquema de encriptación más potente tal como el algoritmo "MD5" para la encriptación de las contraseñas de los usuarios.

- Utilización de la opción "Shadow Password" para mover el campo de las contraseñas desde el fichero /etc/passwd a un fichero para el cual los usuarios "regulares" no tienen permisos de lectura. Esto elimina el riesgo de copias maliciosas del fichero /etc/passwd para la posterior decodificación de las contraseñas. La implementación de esta opción se realiza separando la información originalmente almacenada en el fichero /etc/passwd en dos ficheros:

  - /etc/passwd: Contiene los mismos campos que el fichero tradicional excepto que las contraseñas encriptadas son sustituidas por el carácter "x". Este fichero sigue teniendo permiso de lectura para todos los usuarios.
  - /etc/shadow: Contiene el nombre de usuario, la contraseña codificada e información adicional relacionada con la cuenta del usuario. Este fichero sólo puede ser leído por el usuario "root".

La aparición de esta opción para incrementar la seguridad del sistema, fue aprovechada por los diseñadores del sistema operativo para añadir información que no fue incluida en el diseño original de Unix pero que es muy útil para la gestión de las cuentas de usuarios. Así, el formato que presenta el fichero /etc/passwd es el siguiente:

```
<nombre de usuario>:<contraseña>:<último cambio>:<permite cambio>:<requiere
        cambio>:<aviso>:<inactividad>:<días desactivada>:<reservado>
```

siendo el significado de los distintos campos:

- nombre de usuario y contraseña: Tienen el mismo significado que en el fichero /etc/passwd original.
- último cambio: Número de días (desde el 1 de Enero de 1970) desde la última modificación de la contraseña.
- permite cambio: Número de días que deben transcurrir para volver a cambiar la contraseña ("0" indica que es necesario cambiarla en el próximo inicio de sesión).
- requiere cambio: Número de días máximo que pueden transcurrir sin cambiar la contraseña.
- aviso: Número de días previos a la caducidad de la contraseña que se ha de avisar esta caducidad.
- inactividad: Número de días transcurrido el vencimiento de la contraseña que la cuenta permanecerá activada.
- días desactivada: Número de días (desde el 1 de Enero de 1970) que la cuenta ha permanecido deshabilitada.
- reservado: Campo reservado para uso futuro.

Para ayudar a la gestión de estos nuevos campos de las cuentas de usuarios, los comandos useradd y usermod han sido modificados de forma que pueden establecer los nuevos parámetros. Además se ha añadido un nuevo comando, chage que permite manipular toda la información relacionada con la caducidad de las contraseñas.

2.2.. Protección de recursos

En los sistemas Unix, la protección de recursos se realiza en base a dos conceptos: quién intenta acceder al recurso y cuáles son los permisos para ese recurso en concreto.

2.2.1.. Identificadores de procesos

Cada proceso que se ejecuta en Unix mantiene unos parámetros que identifican al usuario que lo está ejecutando. Estos atributos de identificación se le asignan al proceso en tiempo de creación y se heredan de procesos padres a hijos. Se componen de 3 tipos de información:

1. Identificadores del usuario propietario del proceso: permiten identificar al usuario "dueño" del proceso. Dos identificadores
   - `Identificador real de usuario: rUID`: Contiene el valor del UID del usuario que creó el proceso.
   - `Identificador efectivo de usuario: eUID`: Es el identificador de usuario que se utiliza en el mecanismo de protección

   Normalmente ambos identificadores coinciden salvo que se utilice el bit SETUID en el fichero ejecutable correspondiente.
2. Identificadores del grupo propietario del proceso
   - `Identificador real de grupo: rGID`: Contiene el valor GID del grupo primario al que pertenece el usuario.

- `Identificador efectivo de usuario: eGID`: Es el identificador de grupo que se utiliza en el mecanismo de protección

3. `Lista de grupos suplementarios`:Contiene la lista de los grupos a los cuales el usuario que creó el proceso pertenece.

2.2.2.. Protección de los datos

Como se ha visto en los capítulos previos, los recursos de datos (ficheros ó directorios) en Unix mantienen unos atributos que indican el permiso de acceso de los distintos usuarios a estos recursos. En función del alcance de cada uno de los atributos, estos se dividen en:

- OwnerUID: Contiene el identificador del usuario (UID) propietario del fichero. Puede ser modificado usando el comando chown
- OwnerGID: Contiene el identificador del grupo propietario del fichero. Se puede modificar con chgrp
- Bits de permisos: Es una estructura de 12 bits que fija los permisos de acceso para cada objeto del sistema. El significado de cada bit se observa en la tabla 1.1.

Bit	Significado
11	SETUID
10	SETGID
9	Sticky
8	Lectura para el propietario
7	Escritura para el propietario
6	Ejecución para el propietario
5	Lectura para el grupo propietario
4	Escritura para el grupo propietario
3	Ejecución para el grupo propietario
2	Lectura para el resto de usuarios
1	Escritura para el resto de usuarios
0	Ejecución para el resto de usuarios

Cuadro 1.1: Significado de los bits de permisos en Unix.

Como se desprende de la tabla 1.1 no existe ningún bit que controle explícitamente el borrado. Esto se consigue controlando el bit de escritura en el directorio correspondiente. Así, si un usuario tiene permiso de escritura sobre el directorio /home/prueba, podrá crear y borrar ficheros y/o subdirectorios a partir del directorio prueba.

2.2.3.. Reglas de protección básicas

Cuando un proceso notifica al sistema operativo que desea utilizar un determinado fichero, también notifica la operación que desea realizar: lectura, escritura o ejecución. El sistema operativo actuará de la siguiente forma:

- Si el identificador efectivo de usuario eUID es cero, se concede el permiso independientemente de los bits de protección del recurso, pues el superusuario tiene control total sobre todos los ficheros y directorios.
- Si el eUID del proceso que intenta el acceso coincide con el ownerUID del fichero, la operación se controla en función del valor de los bits 6 a 8.
- Si el eGID del proceso o alguno de los grupos suplementarios del proceso coincide con el ownerGID del fichero, se controla el acceso en función del valor de los bits 3 a 5.
- En otro caso, se controla la operación en función del valor de los bits 0 a 2.

Significado de los bits SETUID y SETGID en ficheros ejecutables

La utilización de estos bits especiales en ficheros ejecutables, permite que un programa se ejecute bajo los privilegios de un usuario distinto al que lo ejecuta. Un ejemplo típico de un fichero ejecutable de estas características es el comando passwd. Este programa, que modifica la contraseña de un usuario, necesita acceder de modo escritura al fichero /etc/passwd, cuyos permisos indican que sólo el usuario root puede acceder en modo escritura. Como el fichero ejecutable tiene el bit SETUID activo, el eUID del proceso que ejecuta se convertirá en el eUID del propietario del fichero.

Bit SETGID en directorios

El bit SETGID, que fue diseñado con el propósito indicado anteriormente, ha sido utilizado para facilitar el trabajo en grupo. De esta forma, si un directorio D tiene el bit SETGID activo, las siguientes situaciones se producen:

- Si se crea un fichero dentro de "D", el ownerGID del fichero va a coincidir con el ownerGID del directorio "D"
- Si se crea un subdirectorio dentro de "D", el ownerGID del directorio será el ownerGID de "D" y además este nuevo directorio también tendrá el bit SETGID activo.

De esta forma, aunque varios usuarios creen ficheros dentro de un directorio, todos estos ficheros pertenecerán al mismo grupo. Así, una utilización adecuada de los bits de permisos 3 a 5 permite controlar el acceso a la información de este directorio.

2.2.4.. Control del uso de disco con cuotas

La escasez de espacio en diso es uno de los problemas que se encuentra comunmente en todos los sistemas. Algunos sistemas Unix permiten limitar el espacio que cada usuario puede utilizar a través de la especificación de cuotas. De esta manera, cuando se habilita este servicio, el sistema mantiene información de la cantidad de disco que cada usuario está ocupando y del número total de ficheros que está empleando. Estos límites se pueden especificar de dos maneras:

- Límite duro: En este caso, el límite impuesto al usuario no puede ser sobrepasado bajo ninguna circunstancia. Cuando se alcanza este límite, el sistema muestra un mensaje al intentar sobrepasarlo y rechaza la operación
- Límite blando: Aquí el usuario puede sobrepasar el límite impuesto durante un periodo de tiempo limitado. La operación se realiza, pero el usuario recibe un mensaje de aviso cada vez que entra en el sistema hasta que, o bien el usuario reduce la utilización del disco por debajo de la cuota, o expira el tiempo de gracia con lo que el usuario es "forzado" a liberar espacio en disco.

Para la habilitación de las cuotas, es necesario indicar en el fichero /etc/fstab que el sistema de ficheros en cuestión tiene cuotas. Además, es necesaria la presencia de los ficheros quota.user y quota.group en el directorio raíz del sistema de ficheros.

Los comandos relacionados para la gestión de estos límites son:

- quota: Muestra la información de la cuota del usuario
- edquota: Permite la edición de las cuotas establecidas a los usuarios y/o grupos de usuarios
- repquota: Genera un informe de las diversas cuotas en un sistema de ficheros
- quotaon,quotaoff: Activan y desactivan el sistema de cuotas

3.. ARRANQUE Y PARADA DEL SISTEMA

A medida que los sistemas operativos han ido ganando en complejidad, el proceso de arranque y parada se ha convertido en mucho más que la activación o desactivación de un interruptor. El arranque y la parada del sistema ha de realizarse de forma correcta para conseguir que nuestro sistema funcione correctamente.

Durante el arranque, el sistema operativo ha de cargarse en memoria desde su localización persistente para a continuación, tomar el mando del equipo. Durante este proceso, el sistema pasa por una serie de fases de inicialización antes de que los usuarios puedan entrar en el sistema. Un administrador de sistemas debe entender qué pasa durante esta secuencia de arranque. Así, en caso de presentarse algún problema durante esta fase, un buen entendimiento del proceso puede ayudar a su resolución. Aunque el arranque es un proceso dependiente del hardware, los principios generales son aplicables a todos los sistemas Linux. Los pasos que se realizan durante el proceso de arranque son:

- Carga del núcleo de Linux en memoria
- Inicialización del sistema operativo y configuración hardware
- Arranque de algunos procesos especiales del núcleo
- Ejecución de los scripts de arranque
- Arranque en modo usuario único
- Arranque en modo multiusuario

3.1.. Carga del núcleo en memoria. Programa LILO

Como se comentó en la sección anterior, la primera tarea a realizarse durante el arranque, es la carga del sistema operativo a memoria y el inicio de su ejecución. En Linux, el sistema operativo o núcleo es un programa que se llama vmlinux o vmlinuz, siendo este último caso la versión comprimida que es la que se utiliza normalmente. Además, normalmente el nombre también especifica el número de versión (ej. vmlinuz-2.2.5-15). Dependiendo de la distribución que estemos utilizando, este fichero se puede encontrar en /vmlinuz o en /boot/vmlinuz.

La carga de este sistema operativo o de cualquier otro, se realiza en dos fases:

1. La BIOS del ordenador carga un pequeño programa de arranque, denominado cargador que permite seleccionar distintos sistemas operativos. El cargador más utilizado por las distribuciones linux es el llamado LILO (LInux LOader)
2. El cargador lee a memoria el sistema operativo elegido y le pasa el control

LILO

LILO es un gestor de arranque para los sistemas Linux en plataforma x86. Permite arrancar varios sistemas operativos tales como DOS,OS/2, Windows 9x, Windows NT, Windows 2k, proporcionando a cada sistema su propia partición. También permite cargar distintas imágenes de kernel de Linux.

LILO es una colección de programas, ficheros de datos y un fichero de configuración. El archivo de configuración /etc/lilo.conf especifica qué particiones son arrancables y, en el caso de ser linux, qué kernel de los múltiples posibles. De esta forma, cuando se ejecuta el programa /sbin/lilo, éste toma esta información y reescribe el correspondiente sector de arranque con el código necesario para presentar las opciones que se han especificado en el archivo de configuración. Cuando arranca el sistema, la BIOS carga LILO en memoria y éste toma el control de la máquina. Muestra un símbolo de sistema ("LILO:) y espera a que el usuario elija el sistema operativo que desea arrancar. Una vez hecha la selección, LILO carga el código necesario de la partición seleccionada y pasa el control a este código (el núcleo del sistema operativo elegido).

3.1.1.. Configuración de LILO

Como acabamos de mencionar, el programa /sbin/lilo lee los parámetros de entrada del fichero de configuración /etc/lilo.conf. Un fichero lilo.conf típico podría ser el siguiente:

```
boot=/dev/hda
propmt
timeout=50
default=linux
image=/boot/vmlinuz-2.2.5-15
        label=linux
        root=/dev/hda5
        read-only
image=/boot/vmlinuz-2.2.5-12
        label=linux.old
        root=/dev/hda5
        read-only
other=/dev/hda1
        label=windows
        table=/dev/hda
```

La primera línea, boot=/dev/hda, le dice a LILO dónde tiene que escribir el sector de arranque. Normalmente, esto se sitúa en el primer sector del disco de arranque (conocido como Master Boot Record o MBR). Este sector tiene un significado especial, pues la BIOS, en tiempo de arranque de la máquina y tras realizar las correspondientes comprobaciones hardware, accede a este sector y carga lo que allí se encuentre a memoria, cediéndole posteriormete el control.

Volviendo al archivo de configuración, el siguiente comando es prompt. Esta instrucción le dice a LILO que muestre el símbolo "LILO:" cuando arranque. Mientras se muestra este símbolo, el usuario puede escribir el nombre de la imagen de arranque o pulsar `TAB` para listar las opciones disponibles. El comando timeout=50 le dice a LILO que espere 5 segundos antes de seleccionar la imagen de arranque por defecto (indicada por el comando default) si el usuario no ha especificado ninguna.

Con la línea: *image=/boot/vmlinuz-2.2.5-15*
indicamos el inicio de una sección dentro del fichero, donde se describe una imagen de arranque específica. En este caso, la imagen a arrancar es el archivo /boot/vmlinuz-2.2.5-15, el cual se corresponde con un kernel de Linux. Dentro del bloque se indica la etiqueta (label=linux) con la que se le identfica esta imagen al usuario bajo el símbolo LILO:. También se indica dónde localizar la partición correspondiente al sistema de ficheros raíz donde reside la imagen que se quiere cargar. La última opción del bloque es read-only, la cual le dice a LILO que monte el sistema de ficheros raíz con permisos de sólo lectura cuando arranque el kernel. Esto es necesario para que se pueda comprobar la integridad de este sistema de ficheros durante el proceso de arranque. Una vez hecha esta comprobación, se vuelve a montar con acceso de lectura-escritura.

Esto termina el primer bloque. El siguiente bloque (que se inicia con la línea image=..., o con la línea other=...) muestra una imagen alternativa de kernel . En este caso, se trata de una imagen antigua que mantenemos para poder arrancar el sistema en caso de que ocurra algún problema con la nueva versión de kernel.

El último bloque, que se inicia con other=... indica la presencia de otro sistema operativo distinto a linux. En este caso, hay que indicar en qué partición se encuentra este sistema operativo (other=/dev/hda1), primera partición del disco en nuestro caso), la etiqueta con la que lo identificaremos en tiempo de arranque (label=windows) y dónde se encuentra la tabla de particiones para que LILO pueda acceder al sector de arranque de la partición donde reside ese sistema operativo (table=/dev/hda, indicando que ésta se encuentra en el sector de arranque del primer disco).

Además de estas opciones que hemos mostrado en el ejemplo, hay otras que se pueden especificar en el fichero /etc/lilo.conf. El significado de estas opciones así como la sintaxis de cada una de ellas pueden consultarse en la página del manual correspondiente: *man lilo.conf*.

Una vez se ha editado el fichero de configuración de LILO, el siguiente paso consiste en ejecutar el gestor de arranque para que escriba esta información en el sector de arranque apropiado. Para ello, basta con ejecutar /sbin/lilo.

3.2.. El proceso init

Una vez el kernel ha sido instalado en memoria RAM por el programa "cargador", empieza la ejecución de Linux. Éste comprueba el hardware y determina qué controladores se deben inicializar en este momento. A partir de aquí, el kernel "monta" sus sitema de archivos raíz e inicia la ejecución de un proceso llamado init.

El proceso init, es el primer proceso no de kernel que se ejecuta en Linux, y siempre tiene el número de identificación de procoeso (pid) "1". Este proceso, lee su archivo de configuración /etc/inittab y determina el nivel de ejecución en el que deberá iniciarse.

Niveles de ejecución

Los niveles de ejecución en Linux indican "estados" en los que se puede encontrar un sistema en ejecución. Cada nivel, designado por un número entero entre 0 y 6, sirve para un propósito específico. El significado de los distintos niveles son los siguientes:

0	Detiene la máquina (apaga)
1	Modo monousuario sin procesos de servidor
2	Modo multiusuario con procesos de servidor
3	Modo multiusuario completo
4	No usado
5	Igual que nivel 3, pero automáticamente entra en modo gráfico
6	Reinicio del sistema

Cuadro 1.2: Niveles de ejecución en Linux.

Las acciones que se realizan en cada nivel de ejecución son controladas por un fichero de configuración llamado /etc/inittab y que el proceso init lee. Así, cuando se entra en un determinado nivel, init ejecuta el script que se le indica en este fichero. El nivel al que se entra por defecto viene determinado por una entrada identificada por la etiqueta initdefault.

Scripts rc

Como se indicó en el párrafo anterior, el fichero /etc/inittab especifica lo que se debe ejecutar cuando se cambian los niveles de ejecución. Normalmete, estas acciones consisten en la ejecución de scripts, los cuales son los responsables de arrancar o parar los servicios particulares a ese nivel de ejecución.

Debido al número de servicios que se necesitan gestionar, se usan los scripts rc. Consiste en un programa principal /etc/rc.d/rc, que es el encargado de llamar a los scripts que arrancan o detienen los servicios particulares en el orden adecuado para cada nivel de ejecución. Así, para cada nivel, existe un subdirectorio en /etc/rc.d, de nombre rcX.d, donde X indica el nivel de ejecución. En estos directorios existen un conjunto de ficheros que no son más que enlaces simbólicos a scripts que se encuentran en el directorio /etc/rc.d/init.d. Estos enlaces simbólicos siguen una convención de nombres ordenada: el primer carácter es la letra S o la K, lo cual indica si el subsistema controlado por ese script debe arrancarse (S) o pararse (K) en ese nivel de ejecución. Luego vienen dos dígitos numéricos, seguidos por el nombre del script tal como aparece en el /etc/rc.d/init.d. Estos valores numéricos indican el orden de la invocación.

Cuando el nivel de ejecución cambia, init invocará al script /etc/rc.d/rc pasándole como parámetro el nivel al que se pretende entrar. Este script procesará inicialmente todos los archivos K del directorio y a continuación todos los S deteniendo y arrancando, respectivamente todos los servicios especificados para ese nivel. Dado que los scripts se ejecutan siguiendo el orden numérico indicado en su nombre, hay que prestar atención a la hora de asignarle número a estos scripts, pues algunos servicios necesitan de la presencia de otros para poderse ejecutar correctamente.

Para facilitar la gestión de los diversos servicios, existen herramientas que nos permite incluir o eliminar servicios en un determinado nivel sin la necesidad de manipular directamente los

directorios rcX.d. Programas como el chkconfig o la herramienta gráfica tksysv son de gran utilidad para la gestión de los niveles de ejecución.

3.3.. Apagado del sistema

Es importante que los sistemas Linux se apaguen correctamente. Si pulsamos directamente el interruptor de encendido, el sistema queda en un estado "impredecible", pues algunos "buffers" pueden no haberse escrito a disco, con la consiguiente pérdida de información. Por tanto, es necesario el uso de los comandos shutdown, reboot, halt o la combinación de las teclas Ctrl+Alt+Supr para detener de forma "limpia" nuestro sistema. De todos ellos, el comando shutdown es el aconsejable pues, además de apagar el sistema en forma segura, permite notificar a los usuarios conectados que el sistema va a detenerse. Esto permite que los usuarios guarden todos sus datos antes de la detención del sistema.

4.. COPIAS DE SEGURIDAD (BACKUPS)

La realización de copias de seguridad (backups) y la recuperación de datos salvados previamente, son tareas esenciales para un administrador de sistemas independientemente del sistema operativo que se use. Estas copias de los datos almacenados en los distintos sistemas de ficheros se realizan normalmente a unidades extraíbles, dentro de las cuales destacan las unidades de cinta. Los objetivos que se ha de fijar cualquier política de backup son los siguientes:

- Garantizar una recuperación del sistema ante un fallo de disco.
- Asegurar que se puede recuperar un archivo eliminido accidentalmente por un usuario.
- Asegurar que se puede recuperar el sistema ante un fallo en la actualización del mismo.

4.1.. Planificación de una estrategia de backups

El desarrollo de una solución óptima de backup no es una tarea trivial. Para empezar, es un proceso costoso en sí mismo. No sólo consume recursos del sistema sino tiempo del operador y por tanto no puede realizarse con la periodicidad que fuera deseable. Además requiere de un conocimiento, por parte del administrador, de las interacciones entre las distintas partes de la organización, especialmente cuando la operación de backup se centraliza. Cada organización tiene necesidades diferentes por lo que, a la hora de optar por una solución de copias de seguridad, hay que plantearse las siguientes cuestiones:

- Los datos que deben ser incluidos en el esquema de backup y la frecuencia con la que deben guardarse estos datos: Determinar la cantidad exacta de datos sobre los que se necesita hacer copia de seguridad y la frecuencia con la que estos datos cambian es la clave para estimar las necesidades de backups. Por ejemplo, los datos de usuarios generalmente cambian con mucha frecuencia. Por tanto, es recomendable que estos datos sean guardados diariamente. En contra, los ficheros de configuración del sistema normalmente son modificados con mucha menor periodicidad por lo que su copia una vez a la semana puede ser suficiente. En el contexto de Linux, los sistemas de ficheros "/" y "/usr" generalmente no cambian con demasiada frecuencia, por lo que son buenos candidatos a guardarse semanalmente. Por contra, los sistemas de ficheros "/home" y "/var" contienen los datos que cambian con mayor frecuencia y por tanto, idealmente, deberían ser guardados diariamente.
- El instante en el que se debe realizar la copia: Debido a los costes del proceso de backup en términos de ancho de bando de red consumido, utilización intensiva de discos, o tiempo de CPU requerido, es aconsejable que las copias de seguridad se realicen cuando el uso del sistema sea mínimo. Además, estas copias deben realizarse cuando sea menos probable que los datos que se están guardando se modifique. Normalmente, esto significa que, en la mayoría de los sistemas, los backups deberían ser programados para que se ejecuten durante la noche o a primeras horas de la mañana.

- El hardware que se utilizará para realizar la copia.: Como se comentó en la introducción, normalmente las copias se realizan sobre unidades de cintas. El tipo de unidad que se deberá elegir depende de la organización (cantidad de datos que se debe guardar, presupuesto, equipos a los que se conectará la unidad, ...). En general, las unidades SCSI presentan unas mejores prestaciones frente a las unidades IDE. En cuanto a la forma de acceder a estos dispositivos, al igual que con los restantes dispositivos del sistema, el acceso se realiza a través de un archivo especial. En este caso, el nombre dela archivo depende del tipo de unidad, del modo de operación y de cuántas unidades estén presentes en el sistema. Por ejemplo, las unidades de cintas SCSI usan el siguiente esquema de nombres:

  - /dev/stX: Dispositivo de cinta SCSI de rebobinado automático
  - /dev/nstX: Dispositivo SCSI sin rebobinado automático.

  donde X indica el número de la unidad (empezando por "0").

- La rapidez con la que se necesita recuperar los datos: En genreral, cuando se produce cualquier caída en un sistema informático, el usuario espera que ésta dure el menor tiempo posible. Por tanto, la política de backup debe incluir una previsión del tiempo que se tardará en cargar la copia de seguridad almacenada de un sistema de ficheros. A la hora de tomar la decisión, hay que llegar a una situación de compromiso entre el coste (tanto económico como en tiempo) de la realización de la copia y el tiempo de recuperación frente a desastre.

  Una estrategía que podría ser adecuada para un gran número de organizaciones sería la de realizar una copia completa una vez por semana (por ejemplo los fines de semana) y copias incrementales diariamente. Mientras la copia completa guarda el sistema de ficheros al completo, la copia incremental sólo guarda aquellos ficheros que han cambiado desde la última copia completa.

  Esta estrategia ahorra la cantidad de cintas necesarias para realizar el backup así como el tiempo necesario para completar el proceso. Sin embargo, aumenta la complejidad en la recuperación de los datos, pues se necesita de al menos dos cintas: Primero es necesario recuperar la última copia completa y a continuación es necesario recuperar las modificaciones almacenadas en los backups incrementales.

4.2.. Herramientas para la realización de backups

Las distribuciones de Linux normalmente incluyen algunas aplicaciones estándar que pueden ser utilizadas para la realización de copias de seguridad y recuperación desde las copias. Aunque no incluyen una interfaz gráfica para la administración, estas herramientas son simples de usar y muchos de los paquetes comerciales de copias usan internamente estas herramientas para realizar las copias.

4.2.1.. Copias con *tar*

El comando tar (tape archiver) es un programa de gran versatilidad que toma como entrada nombres de ficheros o directorios y los almacena en un único archivo, el cual puede ser escrito a una cinta o a un fichero en el disco duro. Este comando puede ser utilizado en tres modos que se describen por el primer argumento pasado en la línea de comandos:

- tar c: Modo de creación. Permite crear un "archivo" o añadir datos a un "archivo" ya existente.
- tar x: Modo de extracción. Permite extraer ficheros o directorios de un "archivo".
- tar t: Modo listado. Muestra la tabla de contenidos del "archivo".

Combinando este argumento con la amplia variedad de opciones disponibles para este comando, se puede tener un control fino de la forma en la que se realiza la copia o la recuperación de archivos.

Ejemplos A continuación se muestran algunos ejemplos de utilización del comando tar:

- *Escritura de un archivo* `tar a cinta`: Para copiar la estructura completa de un directorio a una cinta, se utiliza el siguiente comando: *tar cvf <nombre de dispositivo><directorio a copiar>*
- *Listado de los contenidos de un fichero* `tar en cinta`: Realizamos los siguientes pasos:
  1. Rebobinado de la cinta: *# mt rewind*
  2. Posicionamiento de la cabeza lectora al inicio del "archivo" deseado. Por ejemplo, para posicionar la cinta al inicio del quinto "archivo": *#mt fsf 5*
  3. Mostrar los contenidos del "archivo": *#tar tf /dev/tape |more*
- *Recupración de ficheros desde cinta.* Por ejemplo, si queremos recuperar la estructura de directorios /usr desde la unidad de cinta al directorio /recuperado, los pasos as seguir serían:
  1. Una vez colocados en el directorio destino (/recuperado), se rebobina la unidad de cinta: *# mt rewind*
  2. Se extraen los contenidos del "archivo": *#tar xf /dev/tape ./usr*

4.2.2.. Copias con *dump* y *restore*

El programa dump permite realizar copias completas o copias incrementales. Para soportar backups incrementales, el comando dump usa el concepto de niveles dump. Un nivel "0" significa una copia completa, mientras que cualquier nivel superior a "0" es un incremento relativo a la última vez que se realizó un dump con un nivel inferior. Por ejemplo, un nivel "1" cubre todos los cambios delsistema de ficheros desde el último nivel de dump "0", un nivel de "2" cubre todos los cambios desde el último nivel "1" y así sucesivamente (hasta llegar al nivel 9).

Por ejemplo, supongamos que tenemos tres dumps: el primero es de nivel "0", el segundo es de nivel "1" y el tercero también es de nivel "1". Mietras que el primero es una copia completa, el segundo contiene todos los cambios producidos desde el primer dump, y el tercero guarda también todos los cambios producidos desde el primer dump.

Este programa almacena toda la información sobre sus operaciones en "/etc/dumpdates". Aquí se mantiene un histórico de los dumps anteriores (cuándo se realizó y el nivel). Con esta información, se puede determinar qué cinta usar para restaurar un determinado archivo. Por ejemplo, si se realizó un dump de nivel "0" el lunes, uno de nivel 1 el martes y miércoles, y de nivel 2 el jueves y viernes, un fichero que se ha modificado el martes, pero que se borró accidentalmente el viernes, se puede recuperar de la copia de seguridad incremental del martes.

Uso de *dump*

El uso de dump para realizar un backup sigue la siguiente sintaxis: *dump <nivel dump>usf <longitud de la cinta><dispositivo><partición a copiar>* donde la opción "u" indica que debe actualizar el fichero "/etc/dumpdates", "s" indica que a continuación se especifica la longitud de la cinta de tal forma que se solicita al operador que se cambie la cinta una vez se ha alcanzado su final, y la opción "f" indica que la unidad donde ha de realizarse la copia es especificada en la línea de comandos. Ejemplo:

#dump 3usf 650000 /dev/tape /usr indica que la partición montada en /usr debe ser copiada a la unidad de cinta usando un backup incremental de nivel 3.

Uso de *restore*

El comando restore permite recuperar ficheros y directorios de backups creados con el comando dump. Este comando permite recuperar un sistema de ficheros entero a partir de un dump

de nivel 0 y subsiguientes dumps incrementales, o recuperar ficheros particulares de ellos. Por ejemplo, el comando:

#restore rf /dev/tape indica, con la opción "r" que se desea recuperar un sistema de ficheros completo almacenado en el dispositivo "/dev/tape".

Además, el comando restore puede ser usado de forma interactiva utilizando el parámetro "i". En este caso, se presenta una shell interactiva que permite "moverse" por los directorios contenidos en el backup, listar los contenidos de éstos o extraer ficheros de ellos.

5.. AUTOMATIZACIÓN DE TAREAS

Un sistema operativo multitarea como Linux ha sido diseñado para operar de forma ininterrumpida durante las 24 horas del día. Su misión es la de ejecutar concurrentemente un gran número de procesos, no sólo aquellos que se ejecutan de forma interactiva por el usuario, sino también aquellos que se ejecutan en segundo plano o background. Mientras que algunas de estas tareas cosisten en procesos de servidores tales como aquellos que se encargan de gestionar las peticiones de FTP, HTTP, ..., otras serán tareas que se ejecutan de forma periódica tales como las operaciones de mantenimiento rutinarias, realización de copias de seguridad,.... Sin embargo, sería aconsejable que estas tareas, que por lo general consumen buena parte de los recursos del sistema, interfieran en la menor medida de lo posible con aquellas tareas de usuarios. Una posible solución para conseguir esto sería la de programarlas para que se realicen cuando la carga del sistema sea mínima o cuando se prevea que el número de usuarios que están utilizando el sistema es mínimo.

En Linux, muchas de las tareas de administración pueden ser simplificadas gracias a la automatización. Dos características del sistema posibilitan esto: Por una parte, como se ha visto en los capítulos anteriores, casi todos los archivos de datos de configuración y de sistema utilizan texto para guardar su información, y gran parte del código de gestión del sistema está escrito como shell scripts, lo que permite a los administradores modificar estos archivos de forma sencilla o generar código "a medida" para la gestión de un sistema particular. Por otra parte, Linux incluye algunas herramientas que permiten la planificación de las tareas para su ejecución en ciertos instantes. Puesto que la descripción de los lenguajes de programación con scripts está fuera del objetivo de este curso, a continuación se describen las herramientas que permiten la automatización de tareas.

5.1.. El programa CRON

El programa cron permite a cualquier usuario de un sistema Unix programar aplicaciones para ejecutarlas en cualquier fecha, hora o día de la semana, con una resolución de minuto. Al igual que los otros servicios analizados previamente, esta aplicación consta de un proceso o demonio y de algunos ficheros de configuración.

5.1.1.. El demonio crond

Crond es el demonio responsable de ejecutar las tareas planificadas en el sistema. Su principio de funcionamiento es muy simple: lee los ficheros de configuración donde se han definido las tareas que se desean ejecutar, los instantes a los que se han de ejecutar estas tareas y el usuario que va a realizar la ejecución y "ejecuta" los comandos especificados cuando se alcanza el instante definido.

Al igual que ocurre con otros servicios, durante el arranque del sistema, crond se inicia desde un script en el directorio /etc/rc.d. Durante su inicio, crond lee el directorio /var/spool/cron (/var/spool/cron/crontab en algunos sistemas Linux) buscando archivos "crontab". Los ficheros crontab están organizados por nombres de usuarios y cada fichero contiene todas las tareas planificadas por ese usuario. Por ejemplo, si en nuestro sistema los usuarios "pepito" y "juanita" hubieran planificado alguna tarea con el programa cron, deberían existir dos ficheros de nombres "pepito" y "juanita" en el directorio /var/spool/cron.

Además, en Linux el demonio crond también mira el directorio /etc/cron.d donde normalmente se sitúan tareas del sistema. Todos los ficheros de configuración son cargados a memoria y el proceso crond se despierta cada minuto y comprueba si existe alguna tarea planificada para ese instante.

5.1.2.. El archivo de configuración *crontab*

La herramienta que permite editar las tareas que ejecuta cron se llama crontab. Básicamente, la función de este comando es la de verificar el permiso para modificar las configuraciones de cron para después invocar a un editor de texto para que el usuario pueda realizar sus cambios. Una vez finalizada la edición del fichero, crontab sitúa el archivo correspondiente en la posición correcta y finaliza la ejecución.

Para determinar si el usuario tiene los permisos adecuados para modificar su fichero de configuración, se utilizan los archivos /etc/cron.allow y /etc/cron.deny. Si cualquiera de estos archivos existe, el usuario debe estar explícitamente listado en ellos para que sus acciones tengan efecto. Por ejemplo, si existe el archivo /etc/cron.allow, el nombre del usuario debe estar en este fichero a fin de ser capaz de editar su entrada cron. Por otra parte, si el único archivo que existe es el /etc/cron.deny, a menos que el nombre de usuario se encuentre explícitamente en este archivo, se le permitirá al usuario editar su entrada.

Los archivos que contienen las tareas automatizadas se conocen como archivos crontab. El formato de estos ficheros consiste de una serie de entradas, cada una de ellas identificando la automatización de una tarea. El formato de cada entrada es el siguiente:

```
<minuto><hora><día><mes><día de semana><comando>
```

y el significado de cada campo se muestra en la tabla 1.3.

Campo	Descripción	Valores válidos
minuto	Minuto de arranque del programa	0-59
hora	Hora de inicio	0-23
día	Día del mes	0-31
Mes	Mes del año	0-12
Día de semana	Día particular de semana	0-7
Comando	Comando a ejecutar	–

Cuadro 1.3: Formato de los ficheros crontab.

Para la entrada mes, los valores 0 y 12 indican Enero, mientras que en el campo día de la semana los valores 0 y 7 hacen referencia al Domingo. Además en todos los campos, excepto el del comando, pueden usarse expresiones para indicar rangos o listas:

- Múltiples valores para una columna pueden ser separados por ",". Por ejemplo, para ejecutar una tarea los días 2 y 20 de cada mes, se puede especificar "2,20"en el tercer campo.
- Se pueden expresar rangos de números. Por ejemplo, para planificar una tarea a las 10,11 y 12 horas, se podría indicar "10-12" en el campo hora.
- Se puede usar un "*" para indicar el rango entero de posibles valores. Por ejemplo, para indicar que una tarea ha de ejecutarse todos los días de la semana, valdría con poner un "*" en la entrada correspondiente a día de semana.

Ejemplos:

- *0 4 * * *:* El trabajo se ejecutaría todos los días a las 4:00 am.
- *0 2,4,6 * * 1* La ejecución se producirá todos los lunes a las 2,4 y 6 de la mañana.

Edición del fichero crontab

Normalmete los usuarios no necesitan editar los ficheros "crontab" de forma manual. Como alternativa se puede usar el comando crontab. Con este comando se puede ver los contenidos de un fichero crontab (opción -l), editar el fichero (opción -e), o eliminar el fichero de configuración para el usuario que ejecuta el comando a menos que se especifique el usuario con la opción "-u".

5.2.. El programa AT

El demonio atd ejecuta trabajos planificados con el comando at. Al contrario que el demonio crond que era usado para ejecutar de forma repetida el mismo trabajo, el demonio atd normalmente se utiliza para planificar tareas que se ejecuten solamente una vez en el instante especificado. La sintaxis del comando, así como la descripción de las distintas opciones que implementa atd pueden ser consultadas en el manual del sistema.

6.. SISTEMA A MEDIDA: COMPILACIÓN DEL KERNEL

Como ya se comentó en la Introducción, una de las grandes ventajas de Linux es que su código fuente se encuentra disponible de una forma gratuita. Esto permite que el usuario pueda hacer sus propias "modificaciones" del sistema operativo. Para los usuarios procedentes de otros sistemas operativos, esta aproximación choca con la tradicional filosofía de los sistemas propietarios, en los que es necesario esperar que la compañía libere algún "parche" para la resolución de algún problema detectado. En el caso de Linux, se tiene la posibilidad de contactar directamente con el autor del correspodiente subsistema y contarle el problema con lo que, con mucha probabilidad, se disponga de un "parche" mucho tiempo antes de la liberación de la "versión oficial" del nuevo kernel.

Naturalmente, la desventaja que presenta esta filosofía es que el administrador del sistema necesita ser capaz de compilar su propio kernel en lugar de confiar en que alguien le suministre la versión compilada. Aunque en un sistema en explotación es muy poco frecuente la compilación de un nuevo kernel, es necesario conocer el procedimiento para realizar esta operación y la consiguiente instalación del recién generado sistema operativo.

6.1.. Visión global del kernel

Antes de pasar a analizar el proceso de compilación del kernel, vamos a describir qué es el kernel y el papel que juega en el sistema. Se conoce como "kernel" de un sistema operativo a la capa de software que se sitúa entre el "hardware" y los programas de aplicación. La figura 1.6 muestra un esquema simplificado de la estructura general del sistema operativo Linux y la interconexión entre los distintos subsistemas:

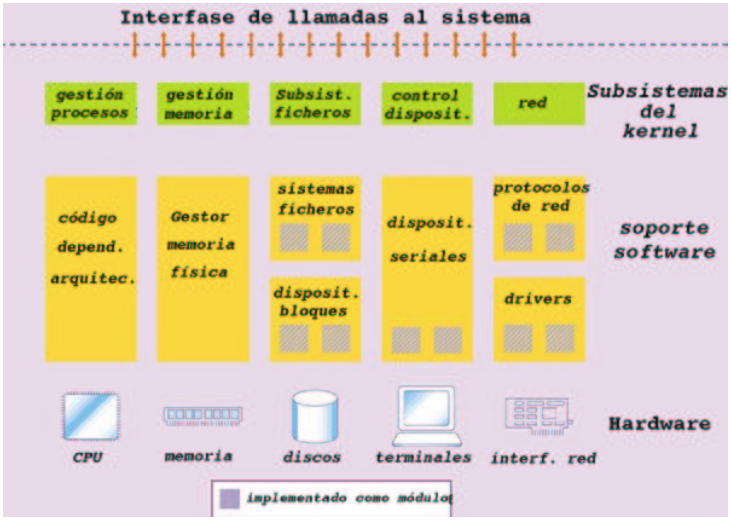

Figura 1.6: Estructura de un sistema operativo

El kernel es un programa "normal" con una funcionalidad especial, pues es el encargado de la correcta ejecución de los restantes procesos del sistema. A diferencia del resto de los programas

del sistema, este programa llamado vmlinux no se ejecuta como un proceso independiente y por lo tanto, no aparece cuando realizamos un listado de los procesos en ejecución (comando ps). En realidad, las distintas operaciones que realiza este programa las ejecuta "dentro" de los procesos de usuarios y sistema. Estas operaciones, o servicios, se pueden clasificar en las siguientes categorías:

- Control de procesos: Es el encargado de crear, finalizar, suspender y reanudar la ejecución de los procesos
- Planificación: Determina en cada instante qué proceso es el más adecuado para ejecutarse. La asignación la realiza en base a una política que intenta repartir de la forma más "justa" posible la CPU entre los distintos procesos.
- Gestión de memoria: Reparte la memoria física de la máquina entre los distintos procesos que, en cada instante, se encuentran en ejecución.
- Gestión de Entrada/Salida: Controla el acceso a los dispositivos periféricos y proporciona los mecanismos para las operaciones de lectura/escritura sobre los mismos.

6.2.. Cuándo se debe compilar un nuevo kernel

Aunque los "kernels" que vienen con las distintas distribuciones "Linux" son lo suficientemente genéricos para adaptarse a la mayoría de las configuraciones actuales, en ciertas situaciones es necesario reconfigurar y recompilar una nueva versión del sistema operativo. Algunas de estas situaciones podrían ser las siguientes:

- Puesto que los "kernels" genéricos son creados para soportar un amplio rango de dispositivos, normalmente incluyen la mayoría de las opciones activadas e incluyen un gran número de controladores de dispositivos. Esto hace que, aunque versátiles, estos núcleos son demasiado grandes, lo que a su vez repercute en la eficiencia del sistema. Si se recompila el kernel incluyendo sólo aquellos dispositivos y opciones necesarias para nuestro sistema, el programa generado se convierte en un kernel más rápido, eficiente y seguro.
- La aparición de nuevos dispositivos y sus correspondientes controladores hace necesario la modificación del kernel para soportarlos. Siempre que sea posible, estos controladores deberían incluirse en el kernel como "módulos que el kernel puede cargar y descargar de forma dinámica evitando de esta manera, tener que compilar el kernel por cada nuevo dispositivo que vaya apareciendo en el sistema.
- La aparición de nuevas "versiones" del kernel hace recomendable su compilación e instalación para mantener nuestro sistema "al día" con las últimas versiones estables libres de los fallos y problemas de seguridad de las versiones previas.
- Si el sistema se encuentra bajo una gran carga, a veces es necesario retocar los límites establecidos a los procesos para la utilización de los distintos recursos. Estos "retoques" llevan asociados una nueva compilación del kernel para que se vean reflejadas estas modificaciones.

Procedimiento de generación e instalación de un nuevo kernel

Cuando se presenta cualquiera de las situaciones descritas en el apartado anterior, deberemos generar un nuevo kernel que se ajuste a nuestras necesidades y hacer que nuestro sistema utilice esta versión "a medida" del sistema operativo. Este procedimiento consta de una serie de etapas que se describen en esta sección.

6.3.. Búsqueda del código fuente del kernel

El primer paso consiste en la adquisición del código del nuevo kernel que queremos instalar. Este código puede adquirirse usando ftp anónimo a ftp.kernel.org en el directorio /pub/linux/kernel o accediendo a cualquiera de los "mirrors" de esta dirección. Accediendo a estas direcciones, podremos ver que existen una serie de carpetas para las distintas versiones del kernel. Antes de proceder

a la adquisición de la última versión, es necesario entender el significado de la nomenclatura de las distintas "versiones". La convención seguida en Linux para el nombre de los distintos núcleos es linux-n.n.n representando n.n.n las versiones mayores, menores y el número de parche, respectivamente. Para distinguir aquellas versiones que están en fase experimental de aquellas que se consideran "estables" se utiliza el número menor: un número impar indica versión en fase experimental, mientras que un número par indica versión "estable". Por ejemplo, las versiones 2.3.x se consideran de desarrollo, mientras que las 2.4.x son versiones de producción.

6.4.. Desempaquetado del código fuente

Una vez se ha obtenido el código fuente de la versión de kernel a instalar, bien en formato tar o en algún formato de paquete, por ejemplo rpm, el siguiente paso consiste en extraer los ficheros fuentes de este "paquete". Sin embargo, antes de proceder a este "desempaquetado" es conveniente "guardar" el kernel actualmente en ejecución por si es necesario volver a él.

El código fuente de las versiones del kernel presentes en nuestro sistema se almacena bajo el directorio /usr/src. Además, normalmente el código fuente del kernel actualmente en uso se sitúa en /usr/src/linux. Suponiendo que se pretende actualizar la versión 2.4.18 a la versión 2.5.20 y que el 'paquete tar que contiene a esta última lo hemos colocado en /tmp los pasos necesarios para instalar el código fuente del nuevo kernel serían:

```
#cd /usr/src
#mv linux linux.2.4.18
#tar xzvf /tmp/linux-2.5.20.tar.gz
#mv linux linux-2.5.20
#ln -s linux.2.5.20 linux
```

6.5.. Configuración del kernel

Una vez se tiene el árbol de los fuentes de kernel desempaquetado, el siguiente paso consiste en la configuración del kernel. Para ello, se necesita conocer todos los detalles del hardware disponible en el sistema. Con esta información, se necesita configurar el kernel para su posterior compilación, un proceso que generará un fichero Makefile personalizado. Para realizar esta configuración se puede usar cualquiera de los 3 comandos siguientes:

- make config: Ejecuta un programa en modo texto a través del cual se pueden seleccionar las opciones del kernel y los controladores de dispositivos adecuados para nuestro sistema.
- make menuconfig: Ejecuta un programa basado en menús que permite hacer la selección de una manera más cómoda.
- make xconfig: Ejecuta un programa en entorno gráfico que permite seleccionar la configuración del sistema de una forma jerárquica. Por ejemplo, al ejecutar este comando se nos presenta una ventana como la mostrada en la figura 1.7. En esta ventana se muestran unos menús de configuración de alto nivel. Accediendo a cualquiera de estos botones, se puede abrir submenús que listan todas las características específicas que se pueden habilitar. Existe una configuración por defecto para la mayoría de las opciones, aunque es recomendable revisar todas las configuraciones para comprobar que están contempladas todas las necesidades.

Usando cualquiera de los tres métodos de configuración anteriores, cada opción puede ser habilitada o deshabilitada seleccionando "y" o "n" respectivamente. En el caso de los controladores se presenta una tercera opción "m" para indicar módulo (Ejemplo figura 1.8). Esta opción, permite al administrador incluir alguna característica del kernel durante el tiempo que este módulo permanece cargado en memoria. Con esto se consigue eliminar del kernel todo aquello que no se necesita de forma continuada, dejando memoria libre para las restantes aplicaciones. Un ejemplo de esto es el soporte a disquetes que normalmente se usa durante un período limitado. Durante este intervalo, se carga en memoria el módulo que contiene el controlador del dispositivo y una vez finalizado el acceso a la unidad, el módulo correspondiente es liberado de memoria.

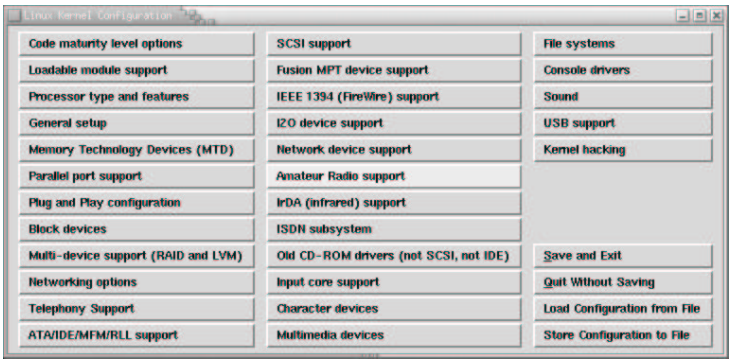

Figura 1.7: Herramienta de configuración del kernel en Linux

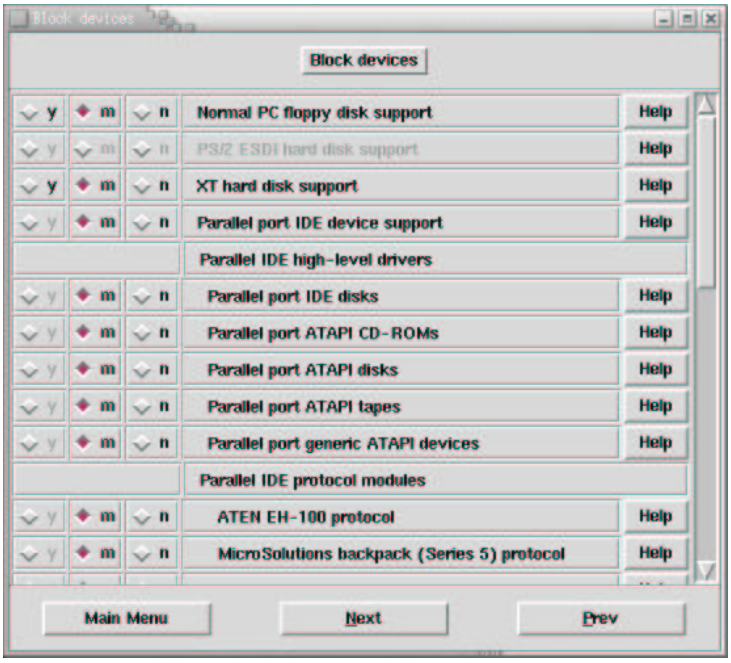

Figura 1.8: Ejemplo de soporte de módulos en la configuración de dispositivos

6.6.. Compilación e instalación del kernel

Una vez finalizada la fase de configuración del kernel, el fichero Makefile contiene las reglas necesarias para compilar el nuevo kernel. Esta parte es la más sencilla, pero también es la que consume más tiempo. La propia compilación está compuesta de tres fases. La primera crea el árbol de dependencias para asegurarse que todos los ficheros necesarios están disponibles. La segunda fase es la de limpieza en la que se eliminan todos los ficheros objetos antiguos de tal manera que se asegura que todas las opciones seleccionadas son realmente compiladas en el nuevo kernel. El último paso es la propia compilación; debido a la gran cantidad de código que se necesita compilar, esta fase puede durar desde un par de minutos hasta horas si se utilizan equipos antinguos. Por tanto, el comando para realizar este proceso completo de compilación sería:

```
#make dep; make clean; make zImage; make modules; make modules_install
```

donde el parámetro zImage indica al sistema que comprima el kernel final, y los parámetros make modules y make modules_install fuerzan la compilación de los módulos y la colocación de los correspondientes códigos objeto en /lib/modules.

Prueba del nuevo kernel

Cuando el programa de compilación finaliza su ejecución, un nuevo kernel ha sido creado. Antes de instalarlo de forma definitiva, es conveniente comprobar su funcionamiento. La manera más segura de probar el nuevo kernel es su instalación en un disquete:

```
#cd /usr/src/linux
#fdformat /dev/fd0H1440
#make bzdisk
```

donde fdformat formatea el disquete y make bzdisk copiará la imagen del kernel comprimida en el disquete. A continuación, se puede intentar arrancar el sistema con el nuevo kernel desde el disquete, y en el caso de que el nuevo kernel no sea capaz de arrancar el sistema, bastaría con arrancar desde el kernel antiguo.

Si por el contrario el nuevo kernel funciona de la manera esperada, es hora de instalarlo como kernel de arranque por defecto. Para ello, es conveniente realizar lo siguiete:

1. Guardar el kernel anterior: #cp /vmlinuz /vmlinuz.old
2. Instalar el nuevo kernel /vmlinuz: #cd /usr/src/linux; make install

Por último, es necesario configurar LILO para que ejecute el nuevo kernel por defecto (3.1.1.)

7.. CONFIGURACIÓN DE LA RED TCP/IP

Se conoce como TCP/IP a un paquete de protocolos de comunicaciones de datos que constituye el corazón de las redes de comunicaciones actuales y en particular, de la red de redes conocida como Internet.

El término TCP/IP se deriva de los nombres de los dos protocolos básicos que constituyen este paquete: El Protocolo de Control de Transmisión (Transmission Control Protocol) y el Protocolo de Internet (Internet Protocol). En esta combinación de protocolos, IP es el encargado de dirigir los paquetes TCP a y desde su destino, mientras que TCP es un protocolo utilizado por las aplicaciones de más alto nivel como FTP o HTTP para construir paquetes que tengan el formato adecuado para viajar por la red.

Utilizando un símil con otro protocolo más familiar, el que se sigue para el envío de un documento a otra persona por correo ordinario, se podría decir que el protocolo TCP se asemeja al protocolo que seguimos para "empaquetar" la información, es decir, introducirla en un sobre, escribir la dirección del destino y del remitente e introducir el sobre en el correspondiente buzón. A partir de este momento, entra en funcionamiento la capa IP, que en nuestro símil la implementa el servicio de correos, el cual se encarga de encaminar el "paquete" hasta la dirección destino. El camino que sigue dicho paquete lo fijan un conjunto de "reglas" que determinan el camino que ha de seguir el mismo hasta llegar a su destino, es decir, la combinación de aviones, barcos, trenes, camiones, motos,... que permitan optimizar el tiempo de entrega del mismo. Una vez introducido el "paquete" en el buzón del destinatario, éste ejecuta su porción del protocolo TCP, es decir, desempaqueta la información y la suministra a la aplicación de más alto nivel (en este caso, lectura del documento).

Aunque existen otros "paquetes" de protocolos para la comunicación de datos, las razones por las que TCP/IP se ha hecho tan popular se podrían resumir en:

- Es un protocolo abierto
- Es independiente de cualquier compañía, de cualquier tipo de arquitectura de ordenador e incluso de cualquier sistema operativo
- Es disponible de forma gratuita
- Es independiente del tipo de hardware de red. Las redes TCP/IP pueden ser implementadas usando tarjetas Ethernet, token-ring, X25, módems, fibra óptica, láser,...
- Proporciona un esquema de direcciones común globalmente, lo que permite identificar de manera unívoca a cualquier ordenador del mundo.

Direcciones IP

La identificación de cada máquina en el protocolo TCP/IP se realiza a través de lo que se conocen como direcciones IP. Éstas consisten en números de 32 bits que se asignan de forma exclusiva a cada máquina que utiliza este protocolo. Estas direcciones se interpretan como un número compuesto de dos partes. La primera parte identifica la red en la cual la máquina reside, mientras que la segunda identifica la máquina concreta dentro de esa red. Se denomina máscara de red a aquella dirección IP que tiene todos los bits a "1" en la parte correspondiente a la dirección de red y todos los bits a "0" en la parte correspondiente a la parte que identifica al equipo dentro de la red. Por ejemplo, para la dirección IP 192.145.102.144 correspondiente a una red Clase C, en ausencia de redimensionamientos de red (subnetting y supernetting), la máscara de red asociada sería 255.255.255.0

Aunque todas las direcciones IP (IPv4) consisten de 32 bits, el tamaño de las dos partes varía dependiendo del tipo de red o clase en la cual se se usa una dirección IP concreta. Existen principalmente tres clases de redes IP conocidas como Clase A, Clase B y Clase C. En la tabla 1.4 se muestran el número de bits que se utilizan para la red y para la máquina en cada una de estas clases así como el rango de direcciones asignadas para cada una de ellas.

Las direcciones IP se escriben generalmente en un formato de cuatro decimales separados por punto. La dirección de 32 bits se divide en 4 números de 8 bits y cada uno de ellos se presenta

como un número decimal, haciendo la representación del número más fácil de entender. Por tanto, el rango de direcciones para cada uno de estos 4 números va desde 0 hasta 255. Ejemplos de direcciones válidas serían 192.168.120.3 ó 193.145.102.143.

Clase	Tamaño de parte de red	Tamaño de parte de ordenador	Número de ordenadores	rango de direcciones
A	8 bits	24 bits	16777214	0.0.0.0 - 127.255.255.255
B	16 bits	16 bits	65534	128.0.0.0 - 191.255.255.255
C	24 bits	8bits	254	192.0.0.0 - 223.255.255.255

Cuadro 1.4: Clases de direcciones IPv4

Las direcciones representadas por el primer byte mayor de 224 identifican redes reservadas tales como la red Clase D de multidifusión. Además de estas redes, otros tres bloques de direcciones están reservadas para uso interno en redes privadas que no están directamente conectadas an Internet. Estos 3 bloques son:

10.0.0.0 a 10.255.255.255
172.16.0.0 a 172.31.255.255
192.168.0.0 a 192.168.255.255

Las direcciones incluidas en estos 3 bloques no son encaminadas por Internet, y su radio de acción sólo alcanza la red privada. Por tanto, cualquier organización puede utilizar estos rangos de direcciones sin tener que registrarlas. Sin embargo, para usar cualquier otra dirección, es necesario registrarla con el fin de asegurar que la relación máquina - dirección sea unívoca.

7.1.. Configuración de la red en Linux

La configuración de la red para Linux implica generalmente dos pasos. En primer lugar, el kernel debe ser configurado con las opciones para red, y los controladores de los dispositivos que van a ser utilizados para el acceso a la red han de estar incluidos en el kernel, bien directamente o bien como módulo. El segundo paso consiste en modificar los ficheros de configuración necesarios para indicarle al sistema las características de la red donde está ubicado. Para abordar estas dos fases de configuración, el administrador puede optar por dos alternativas: usar herramientas proporcionadas por la distribución concreta de Linux que se está usando o modificar de forma manual los ficheros involucrados en la configuración de la red. En nuestro caso vamos a seguir el procedimiento "estándar" disponible en todas las distribuciones de Linux, es decir, la edición manual de los ficheros de configuración y utilización de los comandos disponibles en cualquier distribución. Herramientas gráficas que pueden utilizarse para estas labores pueden ser linuxconf o webmin.

Configuración manual de la red de Linux

Asumiendo que el módulo de kernel correspondiente a la interfaz de red se encuentra ya cargado y detectado como /dev/eth0 (suponiendo una interfaz tipo ethernet), la configuración de la red incluye 5 pasos básicos:

1. Configurar la interfaz de red
2. Especificar los servidores de nombres
3. Definir la traducción de nombres estática.
4. Definir las rutas de la red
5. Automatizar la configuración durante el arranque del sistema

7.1.1.. Configuración de la interfaz de red

Para la configuración y activación de la interfaz de red se utiliza el comando ifconfig. Vamos a considerar la siguiente situación:

Nuestro equipo, con la interfaz de red accesible a través de eth0 se encuentra conectado a la red privada de clase C 192.168.100.0, la cual tiene una máscara de red 255.255.255.0, siendo la dirección IP asignada a esta interfaz 192.168.100.15 Para configurar el equipo para conectarse a la red, se podría usar el siguiente comando:

```
#/sbin/ifconfig eth0 1923.168.100.15 netmask 255.255.255.0 up
```

Para comprobar que la configuración se ha realizado de forma correcta, se puede usar el mismo comando ifconfig sin argumentos, el cual muestra el estado de cada una de las interfaces de red activass. En la figura 1.9 se muestra la salida que se obtendría para nuestra interfaz.

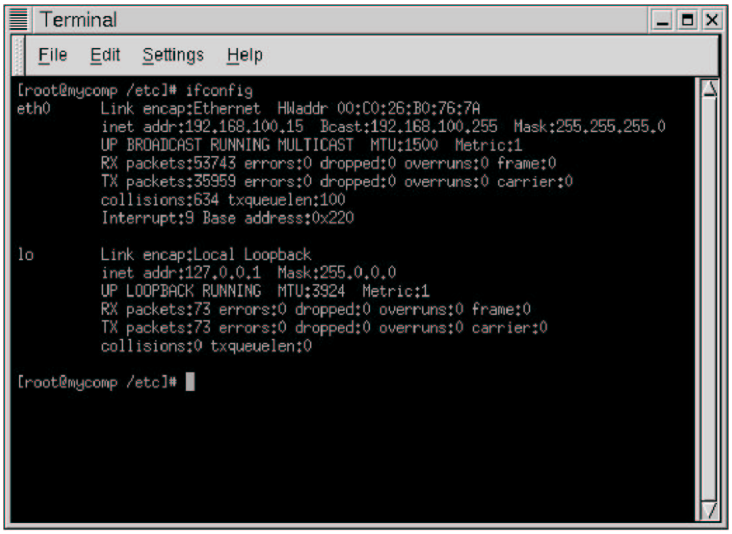

Figura 1.9: Configuración de la interfaz de red eth0

En la salida de ifconfig, mostrada en la figura 1.9, se comprueba que los valores de configuración asignados a nuestra interfaz eth0, son los adecuados para nuestra red. Además, se puede observar otra sección, identificada con la etiqueta "lo" que representa la interfaz virtual de lazo local (local loopback). Ésta es normalmente preconfigurada por Linux cuando se instala la distribución correspondiene. Este pseudodispositivo proporciona el mecanismo por el cual las aplicaciones de red pueden hablar con la máquina local sin tener que usar una interfaz de red externa. Es necesario su presencia pues muchas aplicaciones presuponen la existencia de una red lo cual se garantiza con la activación de este pseudo-dispositivo.

Una vez comprobado que la interfaz ha sido habilitada con los parámetros de configuración correctos, el siguiente paso consiste en comprobar que la misma es capaz de comunicarse con los demás miembros de la red. El comando ping es especialmente útil para realizar esta labor. Por ejemplo, si queremos comprobar que nuestro equipo es capaz de "hablar" con otro equipo de nuestra red, podríamos usar el siguiente comando:

```
#ping 192.168.100.25
```

Este comando intentará enviar de forma periódica un paquete de interrogación a la máquina especificada. Si la máquina remota lo recibe, le contestará confirmando la recepción del citado paquete. Además de confirmar la recepción de la respuesta, el comando nos mostrará los tiempos invertidos entre el envío y la llegada de la confirmación, lo cual nos permite además, conocer el estado de las líneas de comunicaciones entre los dos equipos.

7.1.2.. Resolución de nombres de Internet

Uno de los factores que han contribuido a la popularidad de Internet es su sistema de nombres para el acceso a una determinada máquina. Si en lugar de escribir http://www.yahoo.com tuviéramos que escribir su dirección IP http://204.71.200.68, la utilidad de Internet se vería drásticamente reducida. La necesidad de mapear las largas direcciones IP numéricas en un formato más amigable ha sido un tema prioritario desde la creación de TCP/IP en los años setenta. Aunque esta traducción no es obligatoria (siempre podremos acceder a una determinada máquina conociendo su dirección IP), hace que la red sea mucho más práctica y fácil de trabajar para la mayoría de los usuarios.

Inicialmente el mapeado de direcciones IP a nombres se hacía a través de un fichero conocido como "HOSTS.TXT" que se distribuía a todas la máquinas de Internet. Cuando el número de máquinas creció (a principio de los años 80), quedó claro que una persona gestionando un archivo para todas las máquinas no era una forma realista de realizar la traducción número IP - nombre de máquina. Para resolver el problema, se creó un sistema distribuido en el cual cada "dominio" mantenía la información de sus propias máquinas. A este sistema se le conoce en la actualidad como DNS o servicio de nombres de dominio. Usando este servicio, el procedimiento que se sigue para que una máquina A sea capaz de conocer la dirección IP de otra máquina B sería el siguiente:

1. La máquina A pregunta quién es la máquina autorizada de todos los nombres del dominio donde reside B.
2. La máquina A recibe la respuesta indicando la dirección IP del servidor de nombres del dominio donde recide B (NSB).
3. A pregunta a NSB la dirección IP de la máquina B
4. NSB contesta indicando esta dirección
5. La máquina A ya puede contactar directamente con B

En una red, todo este trabajo de resolución de nombres a IP lo realiza una máquina dedicada a ello conocida como DNS o servidor de nombres de dominio. Por tanto, para que la máquina que estamos configurando sea capaz de acceder a otras máquinas en Internet, debemos indicarle la dirección de la máquina que realiza este servicio. En el caso de Linux, la configuración se realiza a través del fichero de texto /etc/resolv.conf. El formato de este fichero consiste en dos tipos de entradas:

- domain: Esta entrada especifica el nombre del dominio de la red local donde se ubica nuestra máquina. Por ejemplo, si la máquina se encuentra en el dominio midept.ull.es, la primera línea del fichero resolv.conf sería:

  ```
  domain midept.ull.es
  ```

- nameserver: Indica la dirección de un servidor de nombres. Es necesario incluir una entrada por cada servidor DNS disponible en orden de prioridad. Así, el servidor primario debería ser la primera línea, el secundario la segunda y así sucesivamente. Si en nuestra red disponemos de dos servidores de nombres, el formato del fichero sería el siguiente:

  ```
  nameserver 192.168.100.2
  nameserver 192.168.100.3
  ```

Otra alternativa para la resolución de nombres de Internet la constituye el fichero local /etc/hosts. Este fichero permite establecer un mapa ¡nombre de máquiena¿- ¡dirección IP¿ que puede ser de utilidad en ciertas ocasiones:

- Acceso a determinadas direcciones muy frecuentadas: En este caso, incluyendo de forma estática la dirección IP de la máquina se evita todo el procedimiento de resolución de nombres.

- Red privada: En el caso de pertenecer a una red privada, puesto que las direcciones IP no traspasan el umbral de la propia red, cualquier servidor DNS situado en el exterior es incapaz de resolver nombres de máquinas pertenecientes a esta red. Por tanto, en estas situaciones se plantean dos soluciones: implantar un servidor DNS en el interior de la red, o editar manualmente el fichero /etc/hosts incluyendo la resolución estática de cada una de las máquinas

Es posible usar una solución híbrida, es decir, servidor DNS y /etc/hosts. Cuando se usa este fichero, es conveniente asegurarse que el orden de búsqueda está correctamente configurado. Por ejemplo, si queremos realizar la búsqueda de una determinada dirección primero en el fichero local y luego iniciar el procedimiento de resolución de nombres usando los servidores DNS, sería necesario configurar el fichero /etc/host.conf con la siguiente entrada:

```
order hosts,bind
```

Una vez definido el órden de búsqueda, el siguiente paso consite en la edición del fichero /etc/hosts para incluir en él tanto las direcciones de acceso frecuente como aquellas direcciones que no pueden ser resueltas por un servidor DNS. Este fichero consiste de una serie entradas, siendo el formato para cada una de ellas:

```
<dirección IP><nombre de máquina><alias><alias>...
```

Como mínimo, cada entrada necesita la dirección IP y el nombre de la máquina. Además se pueden especificar uno o más alias que representen nombrtes alternativos para la máquina concreta. Por ejemplo, un posible fichero de configuración podría ser:

```
127.0.0.1 localhost.localdomain localhost
192.168.100.20 contabilidad.miorg.com contabilidad
192.168.100.30 ventas.miorg.com ventas
```

En este caso, aparte de la primera entrada para la resolución del lazo local, se introduce la resolución de nombres de dos máquinas situadas en nuestra red virtual. El tercer campo que se ha incluido en estas dos entradas permite referenciar las citadas máquinas sin necesidad de incluir la dirección completa, siendo posible acceder a las mismas usando los nombres contabilidad y ventas, respectivamente.

7.1.3.. Definición de rutas

Otro aspecto que es necesario considerar cuando se configura una red es la definición de las rutas que han de seguir los paquetes en nuestra red local para alcanzar su destino. Como mínimo, es necesario definir dos rutas para nuesto sistema:

- Una ruta especificando que nuestra máquina local se accede a través de la interfaz de lazo local
- Una ruta especificando que todos los paquetes destinados a nuestra red local han de ser enviados a través de una interfaz de red concreta (eth0 en el caso más general)

Las rutas se definen usando el comando route. Para especificar una ruta hacia una máquina concreta, o hacia una red en general se usa, respectivamente, la siguiente sintaxis:

```
#route add -host <dirección IP o nombre de máquina><dispositivo>
#route add -net <dirección de red>netmask <máscara de red><dispositivo>
```

Por ejemplo, para nuestro sistema que se encuentra conectado en la red 192.168.100.0 con la máscara de red 255.255.255.0, la definición de una ruta para esta red a través de la interfaz eth0, se usa el siguiente comando:

```
#route add -net 192.168.100.0 netmask 255.255.255.0 eth0
```

Por último, si nuestra red está conectada a una red externa (por ejemlo Internet) a través de un router, es necesario definir una pasarela por defecto a esta red externa. Siguiendo con nuestro ejemplo, si la pasarela para nuestra red tiene el IP 192.168.100.1, el siguiente comando permitiría dirigir los paquetes que vayan destinados a la red externa:

```
#route add default gw -net 192.168.100.1 eth0
```

Una vez configuradas las rutas, el comando route sin argumentos permite mostrar la tabla con las rutas definidas:

```
#/sbin/route
```

Kernel IP routing table
en la que se puede ver que todos aquellos paquetes que no vayan dirigidos a la red local ni a la

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.100.0	*	255.255.255.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	192.168.100.1	0.0.0.0	UG	0	0	0	eth0

dirección de lazo local, serán enviados a la dirección 192.168.100.1.

El comando route cobra especial relevancia cuando nuestro equipo Linux posee más de una interfaz de red. En este caso, la máquina en cuestión puede desempeñar papeles de router y la tabla mostrada anteriormente debería incluir las reglas para el direccionamiento de los paquetes procedentes de las dos subredes.

7.1.4.. Automatización de la configuración de red

Los pasos indicados en las secciones anteriores permiten configurar el acceso a la red TCP/IP para una máquina que se encuentra en ejecución. Sin embargo, los parámetros de configuración asignados usando los comandos anteriores son válidos durante la sesión en curso. Por tanto, al reiniciar la máquina sería necesario volver a repetir estos pasos para mantener la configuración.

La automatización de esta configuración se puede realizar, una vez más, editando dos ficheros de sistema:

- /etc/sysconfig/network
- El script de configuración para cada interfaz de red

El fichero /etc/sysconfig/network contiene la información básica de la red. Un ejemplo típico sería el siguiente:

```
NETWORKING=yes
FORWARD_IPV4=yes
HOSTNAME=miord
GATEWAY=192.168.100.1
GATEWAYDEV=eth0
```

El significado de las entradas es el siguiente:

- NETWORKING: Indica si se habilita o no la red en el sistema
- HOSTNAME: Nombre de la máquina
- GATEWAY: Dirección IP o nombre de la máquina que se usará como pasarela por defecto.
- GATEWAYDEV: Interfaz que se usará para comunicarse con el Gateway

Aparte de este fichero general, es necesario indicar los parámetros de configuración específicos para cada una de los dispositivos de red presentes en el sistema. En el caso de la distribución Red Hat, existe un fichero por cada uno de ellos en el directorio /etc/sysconfig/network-scripts. El nombre de estos ficheros es de la forma ifcfg-¡dispositivo¿. Por ejemplo, para el dispositivo ethernet eth0, el fichero se llamará ifcfg-eth0 y su contenido, para una situación típica, podría ser algo similar a:

```
DEVICE="eth0"
IPADDR="192.168.100.23"
NETMASK="255.255.255.0"
ONBOOT="yes"
BOOTPROTO="none"
```

Las dos últimas entradas indican, respectivamente, si el dispositivo debe ser iniciado cuando se arranque el sistema y si es necesario ejecutar algún protocolo para configurar el dispositivo. Este último caso se utiliza, por ejemplo, cuando existe algún servidor DHCP encargado de gestionar las direcciones IP de la organización, encargándose este servidor de proporcionar la información necesaria para la configuración de la red en los clientes.

8.. DOMINIOS EN LINUX: HERRAMIENTAS NIS Y NFS

En Linux, Unix en general, no existe un concepto de dominio tan marcado como pueda existir en otros sistemas operativos de diseño más reciente. De hecho, en los capítulos anteriores se ha mostrado como la administración del sistema se realiza de forma local, es decir, a partir de unos ficheros de configuración que residen en el disco del equipo local.

Esta estrategia, que proporciona unaas buenas prestaciones para estaciones de trabajo individuales, presenta grandes inconvenientes de administración en organizaciones o grupos de trabajo donde se requiere trabajo cooperativo entre los distintos usuarios que componen dicha organización. Por ejemplo, supongamos un departamento con diversos servidores a los cuales es necesario proporcionarles acceso a todos los usuarios. Atendiendo al esquema de administración presentado en la sección 2.1., cada usuario del sistema debería aparecer en el fichero /etc/passwd de cada uno de los servidores al cual el usuario tiene acceso. Esto implica la duplicación de estas tablas y la consiguiente dificultad de gestión: cada alta de usuario, cada modificación de la contraseña, cada cambio de shell de inicio, ... implica la edición de las tablas de cada uno de los equipos del sistema. Además, cada usuario debería disponer de un directorio de trabajo en cada una de las máquinas lo que complica en exceso la sincronización de los datos, las políticas de copias de seguridad, ...

Afortunadamente, y a pesar de que estas funcionalidades no vienen incluidas en el propio diseño del sistema operativo, algunas herramientas se han desarrollado para que los sistemas Linux puedan usar una autenticación de usuarios centralizada y una distribución de la información de recursos para un entorno de trabajo en grupo. La herramienta más popular que proporciona estos servicios se conoce como Servicio de información de red o NIS (Network Information Service)

El propósito de NIS es situar la información básica de la red en una base de datos centralizada que pueda ser usada por cualquier máquina de la red. Aunque el diseño de la herramienta permite distribuir cualquier tipo de información, las bases de datos más comunmente distribuidas con NIS son las siguientes:

- Tabla de usuarios (fichero /etc/passwd)
- Tabla de grupos (fichero /etc/group)
- Tabla de resolución de nombres (fichero /etc/hosts)
- Tabla de servicios de red (fichero /etc/services)

Usando esta herramienta, se facilitan las tareas del administrador pues, por ejemplo, al cambiar la contraseña de un usuario en la base de datos NIS correspondiente, este cambio se refleja para todos los equipos de la red que se encuentren configurados como clientes NIS.

8.1.. Funcionamiento de NIS

NIS es, simplemente, una base de datos que los clientes pueden consultar. Esta base de datos consta de una serie de tablas independientes, siendo el formato de las entradas de estas tablas de la forma: “clave” - “valor”. NIS accede a estas tablas por nombre y permite realizar búsquedas de dos maneras:

- Listado de la tabla entera
- Búsqueda de una entrada a partir de una “clave” especificada

NIS opera usando una arquitectura cliente/servidor. La base de datos generada en el equipo actuando como servidor maestro puede ser replicadas a uno o más equipos que actúen como servidores secundarios. Mientras el servidor maestro contiene la base de datos original con permisos de lectura-escritura, los servidores esclavos contienen copias de sólo lectura de la base de datos original. La función de estos servidores esclavos es proporcionar un balanceo de carga de tal manera que las peticiones de los usuarios se reparten entre todos los servidores del sistema. Además, estos servidores secundarios también proporcionan un mecanismo de tolerancia a fallos: el servidor secundario puede continuar contestando peticiones incluso cuando el maestro no se encuentra disponible. Una configuración típica de un dominio NIS puede ser la que se muestra en la figura 1.10

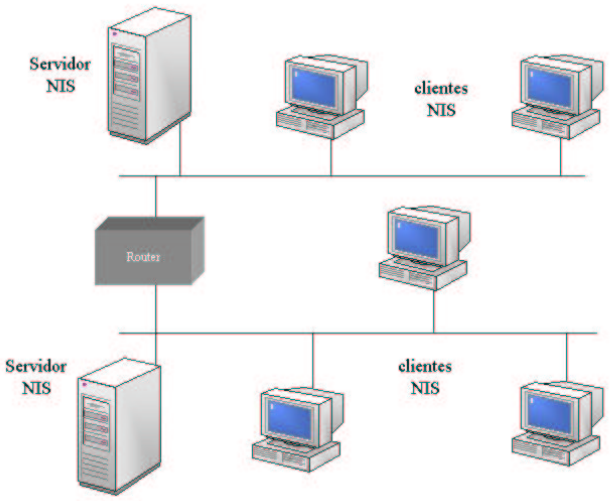

Figura 1.10: Configuración típica de un dominio NIS

En este caso, el dominio consiste en dos redes físicas interconectadas por un router. El servidor NIS maestro se encuentra en una de las redes, mientras que en la otra se ha configurado un servidor secundario. Esta topología es recomendable cuando se implementa un dominio NIS pues, por defecto, los clientes hacen sus peticiones a los servidores mediante paquetes de difusión (broadcast). Estos paquetes, generalmente son filtrados por los dispositivos de interconexión (routers), de tal manera que las solicitudes de los clientes no alcanzarían al servidor maestro situado al otro extremo del router. De esta manera, situando a un servidor secundario en cada red, las peticiones de los clientes situadas en ella, serán atendidas por el servidor secundario.

Los servidores NIS secundarios reciben actualizaciones si el servidor primario se actualiza, de tal manera que las respectivas bases de datos se encuentran sincronizadas.

8.2.. Configuración de un servidor maestro

Las distribuciones Linux normalmente vienen con NIS ya compilado e instalado. En caso contrario, es necesario instalar el paquete ypserv. La configuración de un servidor maestro consiste en cuatro etapas básicas:

1. Establecer el nombre de dominio NIS
2. Arrancar el demonio ypserv
3. Selección de los ficheros que van a formar los "mapas" NIS
4. Ejecución de ypinit

Asignación de un nombre de dominio NIS

Las redes NIS se identifican por un nombre de dominio. Así, un servidor NIS atiende un dominio específico y un cliente se configura para pertenecer a un dominio, de tal manera que sólo los servidores atendiendo a ese dominio gestionarán sus peticiones. Esta filosofía permite que distintos dominios NIS existan en la misma red. Cada dominio tendrá su propio servidor maestro y los clientes se configuran de manera que pertenezcan a ese dominio.

Generalmente, los nombres NIS no coinciden con los nombres DNS. Los nombres de dominio NIS se establecen con el comando domainname. Por ejemplo, si queremos establecer un dominio NIS de nombre midominio, el siguiente comando debe ser ejecutado:

```
#/bin/domainname midominio
```

A fin de que se establezca el nombre del dominio cada vez que se arranque el sistema, es necesario incluir este comando en los scripts de arranque:

- Si se usa Red Hat, basta con editar el fichero /etc/sysconfig/network añadiendo la línea:

  ```
  NIS_DOMAIN=midominio
  ```

- Si se usa otra distribución, basta con editar el script /etc/rc.d/init.d/ypserv incluyendo la siguiente línea al inicio del mismo:

  ```
  #/bin/domainname midominio
  ```

Inicio de NIS

El demonio ypserv es el encargado de atender las peticiones NIS. Por tanto, debe estar arrancado este servicio para el funcionamiento de NIS. Para ello, basta con ejecutar el script de arranque correspondiente:

```
#/etc/rc.d/init.d/ypserv start
```

Si fuera necesario parar el servidor NIS en cualquier momento, se puede hacer con:

```
#/etc/rc.d/init.d/ypserv stop
```

Selección de los "mapas" NIS

En el servidor maestro NIS, los "mapas" que se van a exportar se construyen a partir de los ficheros de configuración tradicionales de Unix situados en el directorio /etc: passwd, hosts, network, services, netgroup,.... Por tanto, la configuración de la red ha de realizarse en estos ficheros que posteriormente serán propagados a toda la red.

En el directorio /var/yp se encuentra un archivo Makefile que contiene las reglas para la generación de los mapas correspondientes. Editando este archivo, se puede configurar los archivos que se compartirán mediante NIS, así como algunos parámetros adicionales sobre cómo compartirlos.

Ejecución de ypinit

Una vez se tenga preparado el Makefile, se puede inicializar el servidor NIS usando el comando ypinit:

```
#/usr/lib/yp/ypinit -m
```

donde la opción "-m" indica a ypinit que configure el sistema como servidor maestro NIS. Durante la ejecución, se solicitarán los nombres de los servidores secundarios que se desean utilizar en estos dominios (los cuales se almacenarán en el fichero /var/yp/ypservers. Una vez hecho, ypinit ejecutará automáticamente el comando make, generando de esta manera los mapas y propagándolos hacia los servidores secundarios.

Una vez iniciado el servidor, cualquier modificación posterior de los ficheros de configuración (por ejemplo, al dar de alta nuevos usuarios), sólo implica la actualización de la base de datos, la cual se puede realizar ejecutando el comando make.

8.3.. Configuración de un cliente NIS

La configuración de los clientes NIS es muy sencilla. Para ello, basta con seguir los siguientes pasos:

1. Editar el fichero /etc/yp.conf
2. Arrancar el servicio ypbind

Edición del fichero /etc/yp.conf

El fichero /etc/yp.conf contiene la información necesaria para que el servicio cliente sea capaz de contactar con el servidor NIS. En este fichero se le indica cómo ha de encontrar al servidor:

- Mediante el uso de brodacast
- Especificando la dirección de la máquina servidora

La técnica de difusión (broadcast) es adecuada cuando se necesite mover el cliente entre distintas subredes y no se desee reconfigurar el cliente para el servidor NIS existente en esa subred. Como inconvenientes de esta técnica caben destacar las siguientes:

- Es necesario que exita un servidor por cada subred (como ya se comentó, los dispositivos de interconexión no propagan los broadcasts)
- Desde el punto de vista de seguridad, cualquier intruso podría suplantar la identidad del servidor y el cliente no sería capaz de detectarlo.

En este caso, en el fichero /etc/yp.conf debería aparecer una entrada como:

```
domain <nombre_dominio>broadcast
```

La otra técnica para contactar con el servidor consiste en especificar el nombre o la dirección IP del servidor. En este caso, ya no es necesaria la presencia de un servidor en cada subred y además, se reducen los problemas de seguridad puesto que en esta situación es más difícil suplantar al servidor del dominio.

La entrada que debe aparecer en el fichero de configuración será, en este caso:

```
domain <nombre_dominio>server <nombre_servidor>
```

donde ¡nombre_servidor¿es la dirección IP del servidor NIS o un nombre que pueda ser resuelto (bien a través de DNS o a través del fichero /etc/hosts

8.3.1.. Arranque del servicio

El cliente NIS se implementa con el demonio ypbind. Por tanto, para que sea accesible NIS después del arranque del sistema, será necesario configurar los scripts de arranque para que se ejecute este servicio. En el caso de Red Hat, la sintaxis de arranque de los distintos servicios tiene el mismo formato:

```
#/etc/rc.d/init.d/ypbind start
```

Podemos comprobar el correcto funcionamiento del dominio utilizando la utilidad ypcat. Así por ejemplo, ejecutando:

```
#ypcat passwd
```

nos debería listar el contenido del mapa Nis passwd o cualquier otro mapa que se esté compartiendo.

8.4.. Sistema de ficheros de red: NFS

El sistema de ficheros de red NFS (Network File System) es una utilidad desarrollada inicialmente por Sun Mycrosystems para los sistemas Unix, pero en la actualidad se ha convertido en el estándar para la compartición de ficheros y directorios entre los sistemas Unix y Linux utilizando el protocolo TCP/IP. Entre las ventajas que presentan la utilización de este protocolo, se incluyen las siguientes:

- Permite mantener de forma centralizada los directorios de trabajo de los usuarios
- Los datos administrativos se pueden mantener en una sóla máquina, evitando los problemas de consistencia de la información asociados al almacenamiento distribuido de la misma.

El mecanismo básico de funcionamiento consiste en una aplicación cliente/servidor típica. La máquina que actúa como servidora exporta un árbol en su estructura de directorios, la cual es "montada" por el equipo cliente dentro de su propia estructura de directorios. Por ejemplo, el comando:

```
#mount -t nfs server:/home/pepe /casa
```

"monta" el directorio /home/pepe situado en el equipo server, en su ruta local /casa. Una vez realizada esta operación, cualquier acceso local en el cliente al directorio /casa se transforma en accesos transparentes al directorio /home/pepe de la máquina que actúa como servidora.

Las siguietes situaciones son admitidas:

- Un servidor NFS puede exportar más de un directorio y atender simultáneamente a varios clientes
- Un cliente NFS puede "montar" directorios exportados por diversos servidores
- Cualquier máquina puede ser a la vez servidor y cliente.

8.4.1.. Configuración de NFS

Puesto que el protocolo NFS se apoya en otro protocolo de nivel inferior, conocido como llamadas a procedimientos remotos o RPC, es necesario que este protocolo se esté ejecutando en el sistema para soportar NFS. Concretamente, los programas que son requeridos para la ejecución de un servidor NFS incluyen los siguientes:

- portmap: Este programa es el encargado de registrar los distintos servicios que se ejecutan usando el protocolo RPC. Así, cuando se intenta acceder a cualquiera de estos servicios RPC, es necesario una conexión previa con el servicio portmanp para interrogarle por el puerto en el cual está escuchando el citado servicio.
- mountd Es el encargado de escuchar peticiones de "montaje" en el equipo servidor.

program	vers	proto	port	
100000	2	tcp	111	portmapper
100000	2	udp	111	portmapper
100005	1	tcp	747	mountd
100005	1	udp	745	mountd
100003	2	tcp	2049	nfs
100005	2	udp	2049	nfs

- nfsd: Resuelve las peticiones de acceso a un determinado directorio exportado por parte de un cliente.

La presencia de estos servicios se puede comprobar usando el comando rpcinfo -p. Si éstos se encuentran ejecutando, la salida del comando anterior podría ser: Una vez se están ejecutando los servicios, la configuración de un servidor NFS es un proceso de dos pasos:

1. Edición del fichero /etc/exports: Este archivo define los árboles de directorios que se comparten con el resto de la red y las reglas mediante las cuales se comparten. Estas reglas incluyen los equipos que están autorizados para acceder a los directorios exportados y los permisos que los distintos usuarios tendrán al acceder a ellos.

   El formato de este archivo consiste en una serie de entradas, cada una de ellas representando un directorio exportado y las restricciones de acceso a ese directorio:

   ```
   directorio_a_exportar cliente1 (permisos) cliente2 (permisos) ...
   ```

   cliente1, cliente2, ... son los nombres de las máquinas o direcciones IP de los clientes que tienen acceso al directorio exportado, mientras que permisos son los permisos que se le conceden a cada uno de los clientes. En la página del manual correspondiente a exports se puede encontrar el significado de las distintas opciones que se pueden especificar en el campo de permisos. De entre ellas, caben destacar las siguientes:

   - ro: Permiso de sólo lectura al directorio, independientemente de los permisos particulares de los ficheros.
   - noaccess: En este caso, al cliente se le denegará el acceso a todos los directorios que se encuentren por debajo del directorio indicado.
   - no_root_squash: Con esta opción se consigue que el servidor ignore las peticiones hechas por el usuario root sobre una partición montada por NFS. El control de permisos se realiza asumiendo que el usuario que intenta acceder es el usuario "anónimo".

   En la figura 1.11 se observa un fichero de configuración típico donde se puede observar como, por medidas de seguridad, el acceso como root se encuentra restringido para la mayoría de los directorios exportados.

2. Ejecutar el comando exportfs -ra para que el servidor lea el archivo /etc/exports y actualice las entradas presentes en el mismo.

8.4.2.. Configuración de los clientes NFS

Los clientes NFS, implementados con el comando mount, son muy fáciles de configurar en Linux, debido a que no requiere ningún software adicional para cargarse. El único requerimiento es que el kernel sea compatible con el soporte NFS, aunque por defecto, todas las distribuciones vienen con esta característica activada.

En la página del manual del comando mount se muestran las distintas opciones que se le pueden especificar a este comando.

Para finalizar, una recomendación: NFS tiene algunos problemas de seguridad y debe ser usado con mucho cuidado y sólo cuando sea necesario. Por norma general, es recomendable usar

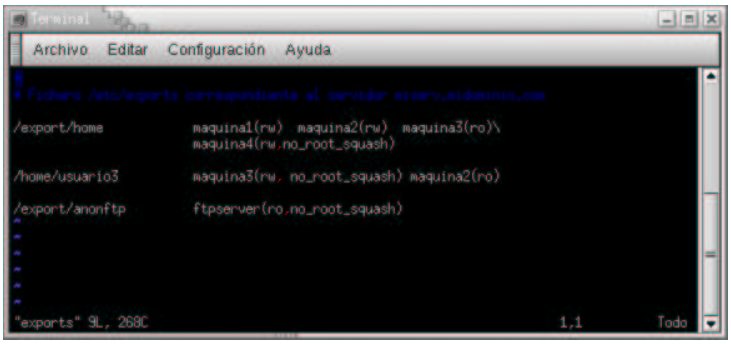

Figura 1.11: Configuración típica del fichero /etc/exports

NFS en aquellas redes que se encuentran protegidas del exterior por algún sistema de cortafuegos o firewall.

BIBLIOGRAFÍA

Linux Installation and Getting Started. Disponible en http://www.tldp.org/LDP/gs/

Unix System Administration Independent Learning. Disponible en http://www.ussg.iu.edu/usail

Curso de Linux. Disponible en www.cybercursos.net/linux2.htm

FRISCH, A. Essential System Administration. O'Reilly & Associates, 1995.

SHAH, S. Manual de Administración Linux. McGraw-Hill, 2001.

CARLING, M., DEGLER, S. Y DENNIS, J. Administración de Sistemas Linux. Prentice-Hall, 2000.