Linux Fácil

Jose Manuel Laveda [1]

24 de marzo de 2000

---

[1] a Fanny, Manu Soriano, Ismael Ripoll, Jose L. Sánchez, gente de Europa 3 BBS, Rafa Torres y a la comunidad LINUXen general

Índice General

Capítulo 1

Introducción

1.1 ¿Qué es LINUX?. Características.

LINUX es una implementación totalmente independiente del núcleo (kernel) del conocido sistema operativo UNIX con todas las características inherentes a éste, esto es: multitarea, multiusuario, multiplataforma, capacidad para la gestión de redes, soporte de varios sistemas de ficheros, ... además de una colección de programas y utilidades que lo acompañan: entorno gráfico X Window, procesadores de texto como el Vi y el Emacs, compiladores para varios lenguajes de programación (C, Lisp, Ada, Fortran, etc...) y otros aspectos que irán apareciendo en el desarrollo de los distintos capítulos que componen este trabajo.

LINUX cumple con las normas del estándar POSIX1 (IEEE) y es compatible con el System V y BSD, a nivel binario con el SCO SVR3 y SVR4 mediante ciertas técnicas de emulación.

Actualmente puede ser ejecutado en las siguientes plataformas:

- Intel 8086/80286/386/486/Pentium y compatibles con bus de tipo ISA, EISA, VLB y PCI. El núcleo destinado a los procesadores 8086 y 80286 se encuentran en desarrollo (versión alpha), se denomina ELKS (Embeddable Linux Kernel Subset)
- Procesadores de la familia Motorola 680x0.
- Procesadores Alpha de Digital Equipment Co.
- Sun Sparc.
- Actualmente está siendo llevado a la arquitectura PowerPC, MIPS y ARM.
- etc ...

Aunque el LINUX en sí mismo sólo es el núcleo del sistema, en adelante abarcaremos bajo esta denominación todos los comandos y programas que configuran su entorno de desarrollo y trabajo, especificando en su caso cuando se refiera concretamente al núcleo.

Con el término independiente se quiere señalar que LINUX ha sido totalmente escrito y concebido desde cero, sin ninguna implementación anterior de ningún otro sistema operativo similar a éste en la que basar su desarrollo (esto se puntualizará mejor en el siguiente apartado), con lo que no posee ningún código que este bajo licencia o propiedad de ninguna persona o entidad.

El núcleo del LINUX, al igual que la mayoría del software que lo complementa está amparado bajo los términos de la denominada GNU General Public License (GPL) de la Free Software Foundation2, fundación ubicada en Estados Unidos creada por Richard Stallman, autor del editor de textos Emacs, la palabra Free de su nombre debe ser traducida en su acepción de libertad (freedom) y no gratis (free), libertad para poder crear y compartir cualquier tipo de software.

Esta licencia permite básicamente al usuario final de un programa el poder usar, compartir, copiar y modificar con toda libertad el software que se acoge a esta licencia con ciertas restricciones. La propia licencia indica asimismo que no se responde sobre cualquier daño o perjuicio que la utilización del software pueda provocar.

Pero cuidado, este núcleo está bajo el copyright de su creador y la persona que actualmente se ocupa coordinar su actualización y puesta a punto: Linus Torvalds. Algunas partes de este núcleo y otros programas que lo acompañan son propiedad de sus autores. Con lo que LINUX no es shareware ni freeware como se podría llegar a pensar.

A pesar de ello se puede explotar para su uso comercial, esto es, se puede utilizar para el desarrollo de aplicaciones con carácter comercial al igual que extraer beneficios por su distribución. De hecho numerosas empresas ya han mostrado su interés y su confianza en este sistema operativo desarrollando software, comercializando bibliografía y creando paquetes que contienen distintas distribuciones del mismo acompañadas de numerosas utilidades de instalación y configuración al mismo tiempo que ayuda para llevar a buen término la explotación de sus recursos. Por lo que LINUX está soportado por las empresas que comercializan distribuciones y productos relacionados con éste.

Pero lo que realmente hace diferente a LINUX de otras implementaciones del sistema operativo UNIX es que junto a éste vamos a tener disponible el código fuente del núcleo, en lenguaje C y ciertas rutinas en ensamblador, para poder estudiar a fondo sus características y, porque no, poder realizar modificaciones o implementar nuevas posibilidades sin ningún limite.

Junto a este código fuente también tenemos a nuestra disposición el conjunto de utilidades necesarias para su configuración, que se puede ajustar totalmente tanto al hardware disponible como a las necesidades que se nos puedan plantear, para su posterior compilación y todo ello en el ordenador personal que podamos tener en nuestro domicilio o trabajo.

Otra ventaja consiste en su dinamismo, LINUX está en un constante proceso de mejora y actualización, para soportar cualquier novedad que pueda surgir en el terreno del hardware o en la mejora de cualquier parte de su código que pueda repercutir en una mayor estabilidad, rendimiento o productividad. Y todo ello gracias a una comunidad dentro de la red Internet que se encarga de esta tarea, sin ánimo de lucro alguno, con equipos perfectamente coordinados de gente que está trabajando continuamente en la mejora del núcleo, desarrollo de manejadores (drivers) para distintos tipos de periféricos, dando verdadero soporte técnico a los usuarios que día a día puedan surgir, escribiendo todo tipo de documentación técnica sobre cualquier tema relacionado con el mismo y desarrollando programas de diferente índole que hacen que LINUX deba tenerse en cuenta como una buena alternativa a los sistemas operativos existentes en el mercado, y todo ello, aunque pueda llegar a sonar reiterativo, con un coste nulo o en ocasiones mínimo, ya que la mayoría del software que se desarrolla es de dominio público o está acogido bajo la mencionada licencia GPL y se encuentra en múltiples lugares dentro de Internet y otras redes.

Buena prueba del "boom"que LINUX ha desencadenado es la cantidad de servidores de Internet y BBS en los que podemos encontrarnos cualquiera de sus distribuciones y programas para él, áreas temáticas dentro de distintas redes (Internet, Compuserve, Fidonet, y otras) dedicadas únicamente a cubrir todos los aspectos de este tema, desde el soporte técnico para cualquier duda a la discusión del desarrollo de nuevas posibilidades, foros en los que se libran verdaderas batallas (dialécticas, claro está, y en ocasiones con cierto toque de subjetividad) entre usuarios de este producto y de otros sistemas operativos, en fin, todo un esfuerzo por parte de sus usuarios para que su distribución sea lo más amplia posible.

1.2 Breve historia.

LINUX nació como un proyecto de Linus Torvalds, inspirado en el MINIX, el sistema operativo desarrollado por Andrew S. Tanenbaum en su obra "Sistemas Operativos: Diseño e Implementación", libro en el cual, tras un estudio general sobre los servicios que ha de proporcionar un sistema operativo y algunas formas de proporcionar éstos, introduce su propia implementación del UNIX en forma de código fuente en lenguaje C y ensamblador, además de las instrucciones necesarias para poder instalar y manejar el mismo.

La primera versión de LINUX, numerada como 0.01 contenía solo los rudimentos del núcleo y funcionaba sobre una máquina con el MINIX instalado, esto es, para compilar y jugar con LINUX era necesario tener instalado el MINIX de Tanembaum.

El 5 de Octubre de 1991, Linus anunció su primera versión 'oficial', la 0.02. Con esta versión ya se podía ejecutar el bash (GNU Bourne Shell) y el gcc (GNU C compiler).

Después de la versión 0.03, Linus cambió este número por 0.10 y tras las aportaciones de un grupo inicial de usuarios se incrementó de nuevo la denominación a 0.95, reflejando la clara voluntad de poder anunciar en breve una versión 'oficial' (con la denominación de 1.0).

En Diciembre de 1993 el núcleo estaba en su versión 0.99.pl14. En la actualidad la última versión estable es al 2.0.30 aunque existe ya la versión de desarrollo 2.1.

La numeración de las versiones de LINUX implica a tres números separados por puntos, el primero de ellos es la versión del sistema operativo es el que distingue unas versiones de otras cuando las diferencias son importantes, el segundo numero indica el nivel en que se encuentra dicha versión, si es un número impar quiere decir que es una versión de desarrollo con lo cual se nos avisa de que ciertos componentes del núcleo están en fase de prueba, si es par se considera una versión estable. El último número identifica el número de revisión para dicha versión del sistema operativo, suele ser debido a la corrección de pequeños problemas o al añadir algunos detalles que anteriormente no se contemplaban con lo cual no implican un cambio muy grande en el núcleo. Como ejemplo sirva la versión de LINUX con la que ha sido desarrollado este trabajo, la última estable hasta hace poco tiempo, su número es 1.2.13, esto es, la versión 1 en su nivel 2 (estable) y la revisión número 13 de la misma en este caso fue la última.

Hay que señalar que LINUX no sería lo que es sin la aportación de la Free Software Foundation y todo el software desarrollado bajo el soporte de esta asociación así como la contribución del UNIX de Berkley (BSD), tanto en programas transportados como en programas diseñados para éste que forman parte de algunas distribuciones del LINUX.

1.3 Como obtener LINUX. Distribuciones.

A pesar de que se haya hecho alguna referencia al modo de obtener LINUX en apartados anteriores detallaré este aspecto con el fin de concretar más.

Tanto si ya hemos tenido noticia de LINUX como si no necesitamos conocer el concepto de DISTRIBUCION porque de ello va a depender la forma que el sistema operativo adquirirá en nuestra máquina en cuanto a forma de instalarlo, ubicación de los ficheros del sistema y diferentes programas, software que tendremos disponible, utilidades para el mantenimiento y configuración del sistema operativo, etc.

Al querer obtener LINUX sea de Internet, en CD-ROM o disquetes (ya en desuso por el número de disquetes requeridos) veremos que de una distribución a otra pueden diferir algunas cosas o ser radicalmente distintas unas de otras.

Una distribución no es mas que una forma de organizar LINUX en el soporte físico de forma que tendremos generalmente el núcleo del sistema operativo además de software de distinta índole para poder explotar las capacidades del mismo. Además podemos disponer de utilidades (en modo gráfico o texto) que nos ayudarán a instalar el sistema operativo y los programas que acompañan a la distribución, configurar el entorno gráfico, gestión de usuarios, puesta a punto del entorno de red caso de estar disponible ...

Existen empresas que se dedican a la puesta a punto y comercialización de dichas distribuciones, algunas aprovechan las mismas como base para añadir software comercial propio creando una distribución diferente. Excepto la denominada Debian (y otras) que está mantenida por voluntarios y en la que todo se decide por consenso.

Para este trabajo me he basado en una de ellas, la denominada Slackware en su versión 3.03, quizá la más popular en nuestro país4 aunque otras están empezando ya a implantarse y a ganar adeptos.

Hay que indicar que aunque algunos aspectos varíen en su forma entre distribuciones los ficheros implicados en la configuración de LINUX son los mismos a pesar que la ubicación de estos pueda diferir sensiblemente. No ha de ser una limitación el que este trabajo se haya basado en Slackware a la hora de enfrentarse a cualquier máquina, los nombres de los ficheros y su sintaxis será la misma en cualquier caso.

La diferencia mencionada en el párrafo anterior estriba en el denominado FSSTND5, estándar que especifica donde deben estar los ficheros de configuración del sistema y da unas pautas generales sobre el resto. El problema con que nos enfrentamos es que no todas las distribuciones siguen esta norma, cosa que sería de gran ayuda para el usuario a la hora de utilizar distintas distribuciones y a la hora de la instalación de cualquier programa.

Otra de las principales diferencias estriba en la forma de instalar el sistema operativo y la forma en que están organizados los ficheros en la distribución. Cada una de ellas proporciona o disquetes que contienen una imagen del núcleo para arrancar la máquina, acompañada de utilidades para comenzar con la instalación o múltiples imágenes del núcleo en forma de ficheros además de las herramientas necesarias para grabarlas en disquete y realizar dicho proceso. Otras distribuciones incluso se pueden instalar directamente desde el CD-ROM sin necesidad de crear dichos disquetes o puede ejecutarse LINUX directamente desde este soporte de forma que podemos ver sus características sin haberlo instalado.

En cuanto a la forma de organizar todos los ficheros que configuran la distribución cada una utiliza un mecanismo diferente. Desde el empaquetado utilizando las utilidades UNIX tar y gzip de Slackware además de la organización de estos paquetes en las denominadas SERIES dependiendo de su cometido en aplicaciones, entorno gráfico, juegos, código fuente del núcleo, redes y comunicaciones, etc. a otros mecanismos de empaquetado más sofisticados como el '.deb' de Debian, '.rpm' de Red Hat (Red Hat Package Manager), el software para gestionar este tipo de paquetes está bajo la licencia GNU6, CRAFT (Component Replacament And Fabrication Technology) de Craftworks. La intención de estos formatos no es otra que la de poder tener un control absoluto sobre todos los programas que tenemos en el disco duro, además de facilitarnos la tarea de instalar nuevo software, actualizar o eliminar dichos componentes, una instalación sin mucho conocimiento de lo que estamos haciendo puede llenarnos la unidad de utilidades que no vamos a utilizar nunca o ni siquiera sepamos que están disponibles con el desperdicio de espacio que esto supone.

Con la llegada de la nueva versión estable del LINUX (la 2.0) así como nuevas distribuciones que incorporan dicho núcleo ya no tenemos que preocuparnos de otra limitación con que nos podíamos encontrar hasta el momento y que permanecía en ciertas distribuciones, la existencia de dos formatos binarios para los ficheros, el denominado 'a.out' que era el estándar hasta el momento y el nuevo formato ELF. La coexistencia en un sistema de ambos formatos

supone algunos problemas a la hora de ejecutar o desarrollar programas debido a que ambos formatos necesitan librerías propias que podían no estar en la distribución, con las consiguientes molestias para el usuario hasta averiguar porque cierto programa no funcionaba en su máquina o no compilaba correctamente. Todas las distribuciones están migrando al nuevo formato por lo que no hemos de tener problemas en este aspecto.

Con la lectura del apartado dedicado a la licencia de uso de LINUX podremos hacernos una idea mas concreta sobre el tema pero hay que recordar que aunque algunas distribuciones puedan llevar software comercial junto al sistema operativo podemos instalar el resto del software (siempre y cuando su licencia lo permita) en cuantas maquinas deseemos al igual que copiarlo, modificarlo y otras muchas posibilidades que la licencia nos permite.

Mi experiencia me ha llevado a manejar maquinas con Slackware, Debian y Red Hat y personalmente me decanto por las dos últimas en cuanto a su facilidad de configuración y control desde las múltiples utilidades para la gestión del sistema que acompañan a la distribución, si no deseamos modificar los distintos ficheros implicados desde el editor de textos. Pero este juicio personal no debe significar una pérdida de valor de la primera, es la que actualmente tengo instalada en mi máquina y con la que se ha desarrollado este proyecto. Cualquier distribución puede ser de utilidad y siempre encontraremos fervientes defensores o detractores de cualquiera de ellas, que pueden ayudarnos a escoger en función de nuestras necesidades o gustos.

Las tres distribuciones mencionadas (Slackware, Debian y Red Hat) se pueden obtener en formato de CD-ROM por lo que habrá que acudir a empresas que las distribuyan, en Internet vía el protocolo FTP en numerosos lugares de la red y en algunas BBS.

1.4 Sobre la Licencia de Uso.

A continuación intentaré explicar lo mas claramente posible los términos en que está redactada la mencionada GPL de la Free Software Foundation, mi intención es hacer un resumen de los puntos más importantes por lo que para un mayor detalle recomiendo acudir al propio texto de la licencia.

LINUX y la mayoría del software que se va a manejar están protegidos por la misma por lo que conocer sus principales características puede ser útil a la hora de saber que podemos hacer con el software que tenemos en nuestras manos y, como no, poder desarrollar programas y optar por ponerlos bajo esta licencia.

Existen otras licencias y formas de distribuir software de forma gratuita acompañando el código fuente, por ejemplo la Universidad de California y el Massachusetts Institute of Technology tienen sus propias licencias creadas para gran parte del software que distribuyen.

Las tres licencias, GPL, BSD y MIT tienen en común los siguientes puntos:

- Todo trabajo desarrollado bajo esta licencia7 puede ser copiado, modificado y distribuido sin cargo alguno siempre manteniendo la información referente al propietario del mismo (copyright).

- Los creadores no se responsabilizan de las consecuencias que puede acarrear el uso del software ni de la información incluida en cualquier documentación amparada bajo esta licencia, es por ello que utilizan la expresión "AS IS"que vendría a significar 'tal y como está', no existe una garantía del producto.

- La Universidad de California y el Massachusetts Institute of Technology además prohiben el uso de sus nombres o el de los desarrolladores para cualquier fin publicitario sin un permiso por escrito.

Las ideas principales de la licencia que nos ocupa son las siguientes, extraídas todas ellas de la GNU General Public License de la Free Software Foundation en su versión numero dos.

- Se puede copiar y distribuir copias del código fuente del software siempre y cuando se mantenga intacta la referencia a la licencia y al propietario del mismo, además se puede cobrar un importe por la distribución del software por cualquier medio.

- Cualquier modificación de un programa ha de llevar consigo los siguientes requisitos:
  - Los ficheros deben llevar en su caso indicación y fecha de dicha modificación.
  - Se debe mantener la información sobre el autor del programa y sobre el contenido de la licencia.
  - La nueva versión del programa ha de permanecer bajo la GPL como el originario a partir del cual se desarrolló.
- La copia y distribución de un programa en forma de código objeto o ejecutable debe implicar lo siguiente:
  - Acompañar al mismo del código fuente o un ofrecimiento por tres años de proporcionar el mismo sin cargo alguno excepto del derivado de la transmisión física del mismo o manteniendo cualquier información sobre este respecto.
  - Si la distribución del ejecutable o el código objeto está hecha con la posibilidad de obtener el código fuente de un lugar concreto pueden no distribuirse juntos.

El código fuente se refiere a todos los ficheros que forman el programa así como a todos los ficheros de configuración y de órdenes para la compilación. No se deben incluir los componentes que se consideran normales para la ejecución del programa tales como el compilador, núcleo del sistema operativo u otros.

- No se puede copiar, modificar, sub-licenciar o distribuir el programa excepto bajo los términos de esta licencia, cualquier intento en contra de este espíritu hará perder todos los derechos de la licencia.
- Para incluir fragmentos del programa en otro programa de dominio público pero con condiciones de distribución diferentes hay que dirigirse al autor.

Hay otros muchos puntos que concretan mucho más los aspectos explicados o indican el alcance legal de dichos términos. A fin de obtener el texto íntegro de la licencia podemos dirigirnos a la propia Fundación (ver Nota 1 del Apartado 1 de este mismo capítulo) o buscar el texto en alguno de los ficheros que acompañan a la distribución, todos los programas que están bajo esta licencia llevan el contenido de la misma en un fichero, por ejemplo en el propio directorio del código del sistema operativo (/usr/src/linux) en el fichero COPYING.

Pero la duda mas importante o más comentada en cuanto al uso de LINUX es si la GPL implica que no se puede utilizar el GNU gcc/g++ (compilador que acompaña a LINUX en todas sus distribuciones, propiedad de la FSF y bajo la misma licencia) con sus librerías para desarrollar software comercial para este sistema operativo y la respuesta es muy simple, SI.

El desarrollo de programas para LINUX no se considera un trabajo o programa derivado del mismo, y la inclusión de ficheros cabecera de LINUX en un programa se considera como la utilización de un ”interface público”.

En cuanto al compilador o compiladores que la FSF desarrolla bajo esta licencia su utilización no es tomada como un "trabajo basado en el programa"tal y como esta definido en la licencia, de hecho hay ya multitud de empresas utilizando dichas herramientas para el desarrollo de software comercial para LINUX sin limitación alguna.

En el caso de las librerías existe una versión de la GPL denominada LGPL (Library General Public License) cuyo contenido es muy similar, para la utilización de las mismas hay que cumplir uno de los siguientes requisitos:

- Compilar dinámicamente el programa.
- Distribuir el programa en forma de ficheros objeto que puedan ser compilados estáticamente por el usuario.
- Facilitar el código fuente del programa.

El objetivo de esta licencia es pues el que el software sea de dominio público además de proveer el código fuente en algunos casos para que cualquier persona pueda además de aprender el contribuir en forma de cambios que repercutan en un software de mejor calidad.

1.5 Fuentes de Información.

Si lo que deseamos es obtener cualquier dato referente al uso, configuración, implementación o en el ámbito del desarrollo de software para el LINUX hay varias formas de obtener dicha información, aunque como se irá viendo en el desarrollo de este apartado la 'Fuente por Excelencia' de información se puede considerar INTERNET, una vez más tendremos que acceder a ésta si queremos conocer cualquier detalle que no podamos encontrar en la documentación escrita a la que tengamos acceso, aunque existen ya foros de debate en otras redes que tratan específicamente sobre LINUX y cualquiera de los aspectos que lo rodean, además de poner a disposición de la persona que tenga acceso a éstas alguna distribución del sistema operativo y software para su disfrute. Valga como ejemplo FIDONET, en donde, a nivel nacional, existe un área dedicada exclusivamente al LINUX (R34.LINUX) y desde la que ya se han fraguado varios proyectos destinados a tareas que repercutan en la expansión de este sistema operativo, desde la mínima asistencia técnica (y moral) para los que están empezando hasta grupos que están trabajando en la traducción de la numerosa documentación que acompaña al LINUX y que por el idioma sigue siendo una barrera importante para algunos usuarios de informática interesados en este sistema.

Por otra parte el intentar indicar todos los recursos disponibles en Internet para Linux nos podría ocupar algo más de un apartado de un capítulo, basta con que el lector se conecte a cualquiera de los buscadores más conocidos (yahoo, altavista, donde?, olé, etc...) e introduzca la palabra LINUX para hacerse una idea de este aspecto.

En primer lugar hay que señalar que cualquier publicación o texto destinado a explicar el funcionamiento del sistema operativo UNIX en cualquiera de sus facetas nos puede ser útil para nuestro objetivo.

Existe numerosísima bibliografía escrita que abarca todo lo que podemos necesitar a la hora de hacer funcionar nuestro sistema. Desde la dedicada al uso básico del sistema en cuanto a comandos más usuales en cualquiera de sus shells, pasando por la configuración, uso de cualquiera de las herramientas que forman el conjunto del sistema (compiladores, editores de texto, entorno gráfico, etc...), hasta la más técnica en cuanto a la descripción de la implementación de un sistema operativo desde su inicio.

No obstante si lo que necesitamos es, como ya se ha indicado, cualquier otra información que no hayamos encontrado de esta forma, existen otras formas de obtenerla que a continuación se explican:

1.5.1 Documentos incluidos en ficheros.

Podemos obtener de numerosos lugares de Internet (ftp sites) ficheros en distintos formatos de texto (desde ascii hasta ficheros con extensión dvi y ps producidos por procesadores de texto del ámbito del UNIX) con información de todo tipo referente al LINUX. A su vez estos ficheros están empezando también a distribuirse por numerosas BBS a las cuales con una simple llamada telefónica podemos acceder y disponer de estos ficheros.

Otra forma de obtenerlos consiste en encontrar una de las numerosas distribuciones que existen en formato CD-ROM en las cuales suelen ir incluidos estos ficheros.

Los dos grupos de ficheros de ayuda principales que existen son los ficheros denominados FAQ (Frequently Asked Questions o Preguntas más frecuentes)) y HOWTO (de la expresión 'how to...' o 'cómo..' en español).

Estos ficheros tienen nombres que ayudan a su fácil identificación, tenemos por ejemplo ficheros denominados de la siguiente forma: Installation-HOWTO, Printing-HOWTO, Linux-FAQ, gcc-FAQ, etc....

Estos ficheros se suelen encontrar en el directorio /usr/doc/faq del sistema, una vez instalado éste. Existen asimismo libros escritos por el Linux Documentation Project (LDP), grupo destinado a la redacción de manuales para LINUX , estos se pueden encontrar en cualquier lugar con otros recursos para este sistema operativo y están disponibles en diferentes formatos de fichero para texto. Los libros disponibles son:

- "Linux Installation and Getting Started."de Matt Welsh. La primera guía del usuario dedicada total y exclusivamente a LINUX.
- "The Linux System Administrator's Guide."de Lars Wirzenius. Explica aspectos de la configuración y administración del sistema.
- "Linux Network Administrator's Guide."de Olaf Kirch. Abarca todos los aspectos de las redes y comunicaciones con LINUX.
- "The Linux Kernel Hacker's Guide."de Michael Johnson trata de cubrir la forma en que LINUX (el núcleo) está construido, dando continuos ejemplos extraídos del propio código fuente además de software añadido por el autor. Puede ser de utilidad como introducción al conocimiento a fondo del sistema operativo, aunque hay que indicar que está basado en una versión antigua8 por lo que muchos puntos han sufrido profundos cambios.

Así mismo en el directorio /usr/doc podremos encontrar directorios dedicados a contener ficheros con información sobre distintas aplicaciones que forman la distribución que hayamos adquirido, estas aplicaciones suelen encontrarse en los denominados paquetes, que forman grupos de programas reunidos por tener un objetivo común o estar destinados a tareas similares, aunque esto es muy dependiente de la forma en la que hayamos adquirido el LINUX.

Otra fuente en forma de ficheros de texto es la denominada 'Info', formato que permite acceder a la información sobre un determinado tema y podernos mover a través de las páginas que pueden conformar el mismo a base de comandos. Actualmente se esta abandonando este sistema en pro de otros mecanismos mas amenos y cómodos para el usuarios como formatos 'html' y otros.

Las páginas del manual (man) que acompañan a LINUX (a la distribución para ser mas exactos) son otra fuente inagotable de información sobre todos los programas que tenemos instalados, comandos y otros muchos aspectos, simplemente tecleando 'man ¡concepto¿' podemos obtener dicha información (si esta disponible). Para aprovechar las posibilidades de esta utilidad los ficheros de ayuda suelen estar en un formato especial, basta con teclear 'man man' para conocer la forma de funcionamiento de este mecanismo y los niveles en que está distribuida la información en función de si se refiere a comandos del shell, llamadas al sistema, utilidades, núcleo del sistema operativo, etc..

1.5.2 LINUX e *Internet*.

A través de este servicio que nos proporciona Internet también podemos averiguar cualquier cosa que necesitemos conocer. Podemos acceder a una página de información que nos servirá de entrada a otras relacionadas con cualquier utilidad de navegación con capacidad para la edición de estos documentos y obtener multitud de ficheros que se encuentran escritor en este formato de texto (html).

La página principal de información de la 'www' perteneciente al 'Linux Documentation Project (LDP)' se encuentra en la siguiente dirección:

http://sunsite.unc.edu/mdw/linux.html

Aunque también podemos acceder a cualquiera de los servidores que realizan la tarea de 'espejo' de esta máquina.

En castellano existen gran cantidad de paginas de información sobre LINUX y seria imposible el describirlas todas en este trabajo, sirva como ejemplo la pagina principal de los grupos SLUG9, INSFLUG y LUCAS que son los que actualmente aglutinan más usuarios e iniciativas: Listas de correos sobre Linux en castellano:

SLUG: l-linux@calvo.teleco.ulpgc.es Cotilleo: linux-qmd@calvo.teleco.ulpgc.es
Hades: linux@hades.udg.es

FAQ (Frequently Asked Questions) de la lista:

http://slug.ctv.es/ acastro/slugfaq/slugfaq.htm

ARCHIVOS COMPLETOS de la lista:

http://slug.ctv.es/ luis/l-linux/ http://calvo.teleco.ulpgc.es/listas/

BUSCADOR de los mensajes antiguos: http://gulic.ml.org/buscador/
http://gulic.eu.org/buscador/

ARCHIVOS de los mensajes de RESUMEN:

http://calvo.teleco.ulpgc.es/listas/

PROYECTO LuCAS (LinUx en CAStellano)

http://lucas.ctv.es/

MIRRORS del LDP:

http://calvo.teleco.ulpgc.es/LDP/

Linux Gazette es una publicación en este formato que puede ser leída en línea u obtenida en forma de fichero comprimido utilizando el protocolo FTP, para poder acceder a ésta la dirección es:

http://www.ssc.com/lg

Otro punto de interés es la página de la organización Linux International dedicada a la promoción de LINUX (http://www.ssc.com/linux-int).

Ayuda 'on-line'.

A través del servicio de Internet IRC (Internet Relay Chat) podemos encontrar canales dedicados exclusivamente a linux en varios idiomas, incluso en castellano. Basta con un cliente software y conocer la maquina a la que conectarnos y el nombre del canal, muchas veces bastante autodescriptivo en su título.

Como ejemplo de iniciativa en este apartado mencionar el grupo de voluntarios LISC (Linux Internet Support Cooperative) cuyo cometido es el de dar respuesta a todas las dudas que a un usuario de LINUX pueden surgirle, para conectar con este grupo hay que conectar con el servidor 'irc.linpeople.org' en el canal#LinPeople.

Listas de correo.

Si su acceso a internet únicamente está restringido al correo (e-mail) puede participar en varias de estas listas dedicadas a LINUX.

Este servicio consiste en que los mensajes que vd. mande a cierta dirección de correo serán enviados a todas las personas suscritas a dicha lista, al igual que usted recibirá en su buzón de correo los mensajes escritos por otras personas para esta lista.

En los grupos de noticias de Usenet y otras vías de información aparece periódicamente información de nuevas listas dedicadas a diferentes temas de este sistema operativo.

Una forma de obtener algunas listas dedicadas exclusivamente a LINUX es escribir un mensaje a la dirección majordomo(vger.rutgers.edu con lists en el cuerpo del mensaje (help si deseamos mas ayuda sobre el funcionamiento de este servicio).

En castellano existen tres listas13 en las que el mecanismo para suscribirse es muy sencillo, tras lo cual recibiremos un mensaje de bienvenida a la lista, información sobre el administrador e instrucciones de como eliminar dicha subscripción. A partir de este momento estaremos adscritos a la que hayamos elegido y entraremos en contacto con usuarios españoles en su mayoría (aunque también empiezan a ser conocidas en Sudamérica). Paso a describir la dirección de correo a la que hay que dirigir el mensaje y el contenido del cuerpo del mismo (se recomienda no poner nada en el campo Subject):

- majordomo@cic.teleco.ulpgc.es (Lista del SLUG).
- majordomo@hades.udg.es
- majordomo@tinet.fut.es

Foros en Úsenet

Prácticamente con el advenimiento del LINUX nacieron varios foros en esta red dedicados exclusivamente a los pormenores de este sistema operativo.

El primer grupo o foro nacido por y para LINUX fue alt.os.linux como necesidad de desviar definitivamente toda la información intrínseca a este sistema de comp.os.minix, donde, como ya se ha explicado11, inicialmente se intercambiaron las primeras informaciones referentes al desarrollo de este sistema operativo. Posteriormente fueron naciendo grupos diferentes para cada uno de los temas que pueden surgir en el uso, desarrollo de software o simplemente para estar al día en las novedades que puedan surgir a este respecto.

Los principales grupos que se pueden mencionar son:

comp.os.linux.announce Dedicado a recoger cualquier novedad referente al sistema operativo. Podría decirse que es el grupo dedicado a las novedades. También recoge periódicamente artículos con ayuda para diferentes temas.

comp.os.linux.answers Para la publicación de los conocidos ficheros 'FAQ' y 'HOWTOS' y otros artículos técnicos.12

comp.os.linux.admin Cualquier duda sobre la instalación y configuración del LINUX, preferentemente en sistemas multiusuario, será resuelta en este lugar.

comp.os.linux.development.system Discusiones sobre los aspectos de desarrollo del sistema tales como el núcleo o manejadores de dispositivos.

comp.os.linux.development.apps Dedicado al desarrollo de aplicaciones de usuario para esta plataforma.

comp.os.linux.hardware Problemas con el hardware y su interacción con el LINUX.

comp.os.linux.networking LINUX y la gestión de redes, comunicaciones, todo lo relacionado con la capacidad de este sistema operativo para comunicar máquinas.

comp.os.linux.x Instalación, configuración y el trabajo con el X Windows (Xfree86), el entorno gráfico del que podemos disfrutar si lo deseamos.

comp.os.linux.misc Si su duda, o información no responde al contenido de los grupos anteriores, su respuesta puede estar en este área.

1.5.3 Revistas.

Linux Journal es una publicación norteamericana dedicada exclusivamente al mundo de LINUX. Básicamente está compuesta por artículos de distintos colaboradores (de los cuales se publica su foto, dedicación actual y dirección e-mail), sección de dudas de los lectores, además de servir de escaparate de todo tipo de productos relacionados con éste tales como programas, hardware, libros, paquetes en CD-ROM con alguna distribución o software destinado al sistema operativo. Su edición es mensual.

Para cualquier información dirigirse a:

> Linux Journal PO.Box. 85867 Seattle, WA 98145-1867 USA. FAX:
> 1-206-527-2806 e-mail:subs(ssc.com http://www.ssc.com

añadir Linux Actual y otras.....

1.5.4 Libros y Obras Publicadas.

Hasta el momento, la cantidad de obras publicadas referidas específicamente al LINUX es más bien escasa, ya que existen solo un número muy reducido de empresas que están empezando a dar soporte a la edición de libros y cualquier material referido a este sistema operativo, aunque existen una serie de libros que, por ser quizás los primeros en aparecer, son básicos si se quiere conocer a fondo este sistema.

Estas publicaciones suelen ser tanto los documentos que están al alcance de todos en Internet, pertenecientes al LDP, como otras publicaciones de personas ajenas a este grupo y que están empezando a escribir y documentar todos los aspectos del trabajo diario o conocimiento a fondo del sistema.

Algunas publicaciones en forma de revista especializada en el sector de la informática ya tiene secciones dedicadas a este sistema operativo o al UNIX con constantes alusiones al LINUX, incluso han acompañado sus publicaciones con algún CD-ROM con alguna de las distribuciones de este sistema operativo, pero hasta el momento esto es todo lo que podemos señalar en este sentido.

Capítulo 2

Instalación

2.1 Conceptos previos.

Antes de dar paso a la instalación del LINUX propiamente dicha, y que nos va a ocupar a lo largo de todo este capítulo, se dará un breve repaso a los conceptos básicos relacionados con la estructura organizativa de un disco duro de los que usualmente podemos encontrar instalados en cualquier ordenador personal compatible y que pueden servir de orientación a la hora de responder algunos de los interrogantes que se nos pueden presentar durante el proceso que nos ocupa y, como no, en otros momentos durante su uso y configuración.

También se detallará el problema de las BIOS que no tienen la capacidad de manejar discos duros con más de 16 cabezas, de gran proliferación, únicamente en lo que se refiere a la instalación de este sistema operativo y su uso.

Generalmente, en un PC con un solo sistema operativo instalado (que vamos a dar por entendido que se trata del DOS, en cualquiera de sus implementaciones1) no hemos tenido que enfrentarnos, para un uso normal, al concepto de partición, posibilidad que nos va a permitir el compartir el disco duro entre varios sistemas operativos sin que las características de uno interfiera en los restantes, cada uno contará con su formato y organización específica del sistema de ficheros sin problema alguno de solapamiento o errores en la configuración de estos si definimos bien el lugar que deben ocupar.

A la hora de instalar LINUX vamos a tener que decidir sobre el lugar donde instalarlo, atendiendo a nuestras necesidades, o simplemente a nuestros gustos. Un buen conocimiento de estas ideas puede ser clarificador.

2.1.1 Estructura lógica de un disco duro.

Para la gestión de toda la información que contiene un disco duro en cuanto a particiones se refiere disponemos de las estructuras:

Sector de Particiones FDISK crea este sector en el primer sector del disco duro (cabeza 0, cilindro 0, sector 0). Este es el sector que carga el BIOS después de un Reset o un arranque del

sistema a la posición de memoria 0000:7C00, siempre que no se encuentre disquete en la unidad A:. Si en los dos últimos Byte de los 512 de este sector se encuentra con una secuencia de código 55AA(hex), considera a

```
Direcc.  Contenido             Tipo
+000h.
Código de partición.
Código.
+1BEh.
1.Entrada en la tabla de particiones.
16 Byte.
+1CEh.
2.Entrada en la tabla de particiones.
16 Byte.
+1DEh.
3.Entrada en la tabla de particiones.
16 Byte.
+1EEh.
4.Entrada en la tabla de particiones.
16 Byte.
+1FEh.
Código de identificación (AA55) que
identifica al sector de particiones.

2 Byte.
```

Figura 2.1: Estructura del sector de particiones del disco duro

este sector como ejecutable, y comienza la ejecución de programa con el primer byte de sector. De lo contrario se produce un error.

El código que se encuentra en este sector es el que tiene como tarea el identificar cual es la partición activa y cargar su sector de arranque, lo cual implica el sistema operativo a ejecutar. Dónde se encuentra el sector de arranque y a que partición pertenece la rutina lo averigua en el contenido de la tabla de particiones, que se encuentra en la dirección 1BEh. en el sector de partición.

Tabla de particiones Cada una de las particiones que podemos tener tiene una entrada en esta tabla. Está limitada por cuestiones de espacio a 4 entradas, por lo que podemos tener solo este número de particiones primarias.

Existe otro tipo de partición, esta vez denominada extendida, de la cual el DOS solo permite una por dispositivo y que puede contener a su vez varias particiones lógicas, de esta forma nos podemos saltar el límite en el número de particiones que podemos tener.

Lo más usual es instalar LINUX en una partición primaria que va a ser la forma explicada para el proceso de instalación, pero esto no ha de ser así y podemos instalarlo en una partición lógica. Un punto a tener en cuenta a la hora de decidir la futura ubicación del sistema operativo es la forma en que podremos incializarlo una vez instalado (teniendo en cuenta las limitaciones que impone nuestra BIOS), sirva como adelanto que siempre podremos utilizar un disquete como método de arranque, método que en principio será el utilizado. Para más información recomiendo acudir al contenido del Apartado 2 del Capítulo III.

2.1.2 Discos duros IDE, BIOS y LINUX.

Ante las necesidades de espacio que pueden surgir debido al software que queremos utilizar y/o la instalación de varios sistemas operativos en nuestro ordenador puede que nos veamos obligados a adquirir un nuevo dispositivo de almacenamiento con más capacidad de la que disponemos. Esto ha propiciado la aparición de discos duros de gran capacidad (85O Mb, 1Gb., 2Gb, etc....) para el tipo de controladora que se ajusta al estándar ATA/IDE3.

El estándar ATA define un total de 28 bits para el direccionamiento de los sectores que componen un disco duro (8 para el sector, 10 para el cilindro, 4 para el número de cabezal y los 6 bits restantes indican el numero de sectores por cabeza), lo cual limita el número de cilindros a 1024, barrera que ya se ha superado en el hardware actual donde no es extraño encontrar discos duros con 1654 cilindros, 16 cabezas por cilindro y 63 sectores por cabeza, por ejemplo.

De esta forma tenemos limitado el tamaño de la unidad de almacenamiento a 504 Mb.4 sea cual sea su tamaño real debido a que esta información la gestiona la BIOS5 del ordenador.

El sistema operativo DOS utiliza los servicios de la BIOS para el acceso al disco duro por lo que esta limitación le afecta directamente.

Algunas BIOS ya han subsanado este problema, teniendo la posibilidad de acceder a los sectores mediante otros esquemas de direccionamiento tales como el LBA (de Logical Block Address) que establece un direccionamiento lineal de los sectores comenzando desde 0 separando de esta forma la dirección física de la lógica además de utilizar una geometría diferente para la misma unidad a fin de poder obtener más cilindros.

Con una BIOS que no tiene esta posibilidad existen varias opciones:

1. Si hemos definido la partición DOS al límite del espacio mencionado, hemos de tener en cuenta que tendremos que arrancar el LINUX con el disco de arranque habilitado a tal efecto o con el programa llamado Loadlin, no podremos utilizar el gestor de arranque Lilo que acompaña a la distribución.

2. Tener la parte correspondiente al arranque del sistema operativo (vmlinuz ó zimage) por debajo de este límite de 1024 cilindros, dentro del la partición correspondiente al DOS, o como partición dedicada únicamente al arranque, para que pueda funcionar el gestor de arranque Lilo. Si introducimos los ficheros necesarios en la partición DOS puede traer graves problemas si utilizamos un programa de defragmentacion del disco duro bajo DOS, puede ′mover′ esta imagen, confundir al Lilo, y no arrancar.

Lo que hay que dejar claro es que a LINUX no le afectan estas limitaciones porque no utiliza los servicios de la BIOS para acceder al disco duro durante su ejecución, sólo accede a los parámetros necesarios mientras arranca el sistema a fin de obtener su geometría. Al gestor de arranque Lilo le afecta dicho problema porque accede a la BIOS a la hora de obtener los parámetros de la unidad.

2.2 Crear los Discos para la Instalación.

Lo primero que hay que hacer para instalar el sistema operativo LINUX con esta distribución es crear dos discos que van a contener el software necesario para la instalación del mismo. Uno de ellos (boot) contendrá una imagen del núcleo a cargarse en memoria durante el arranque del sistema, esta imagen la escogeremos del directorio BOOTDISK, y el otro contendrá el resto de software indispensable (root).

Se da por supuesto que previamente hemos dejado espacio suficiente en nuestro disco duro para que pueda contenerlo, esto es, tendremos que haber ajustado el tamaño de las particiones que pudiéramos tener. Para este fin la distribución Slackware incorpora en uno de sus directorios un programa llamado FIPS6 que ayuda a reducir el tamaño de las particiones DOS, pudiendo evitar el tener que perder toda la información de la que disponemos en este lugar al tener que recurrir al FDISK. A pesar de estar convenientemente indicado en la documentación que lo acompaña conviene recordar que el trabajo con este programa no está garantizado en su éxito además requiere estar muy seguros del proceso que estamos realizando.

Sobre el espacio conveniente para la instalación de LINUX, Slackware no proporciona información sobre el tamaño que cada uno de los paquetes o series puede ocupar ni lo que diferentes tipos de instalación (mínima, normal, ...) puede necesitar. Para una instalación con los ficheros de arranque y configuración, algunas utilidades y el entorno gráfico podemos necesitar al menos 80 Mb., pero esto solo es un dato orientativo (otros documentos indican que el tamaño estimado es de 2 Mb. por paquete instalado).

Una vez formateados y verificados dos discos vamos a proceder a crear el disco de arranque (boot), para ello hemos de elegir entre todos los núcleos que se incluyen en el directorio BOOTDISK, pero antes unas notas.

Estos núcleos los podemos considerar 'genéricos', esto es, algunos de ellos están configurados para el soporte de lectores de CD-ROM específicos, tarjetas SCSI y otro tipo de hardware, pero esto no garantiza que vaya a reconocer este hardware ª la primera", nuestro objetivo es que el sistema reconozca el hardware básico de nuestro ordenador, para poder arrancar y posteriormente configuraremos o incluso compilaremos el núcleo para ajustarlo a nuestras necesidades.

En este directorio disponemos de varios ficheros con núcleos para diferentes configuraciones.

En el fichero which.one de este directorio tenemos una detallada descripción de cada uno de ellos y el tipo de hardware para el que está configurado, se optará por el más general, el fichero bare.gz puede servir para empezar, como se ha mencionado antes, ya tendremos ocasión de configurar el LINUX para que reconozca nuestro hardware más concretamente.

Es por esto que, si tenemos la distribución en formato CD-ROM, aconsejo el copiarla previamente en la unidad C:, o en disquetes si lo deseamos, ya que puede que ninguno de los ficheros que tenemos a nuestra disposición reconozca nuestro lector (de momento hemos optado por un fichero de arranque que no maneja esta unidad).

Podemos empezar por copiar a nuestro disco duro (o en disquetes) las series A, AP, D y K para la instalación y configuración. Esto se recomienda ya que hay numerosos problemas con los distintos tipos de lectores de disco óptico que existen en el mercado, todos preconizan el adaptarse a estándares como el IDE/ATAPI para este tipo de periféricos pero no obstante no estamos exentos de tener múltiples problemas a pesar de haber elegido ficheros como idecd.gz, abare.gz y otros destinados a esta tarea.

Para la copia del contenido de este fichero en el disquete utilizaremos dos programas que acompañan a la distribución que se denominan RAWRITE y GZIP, posteriormente volveremos a recurrir a éstos para la creación del disco root.

Los pasos necesarios serán:

1. Copiar el fichero elegido en nuestro disco duro (comando copy).
2. Descomprimir éste utilizando el programa GZIP con el comando:

   ```
   D:GZIP7 -d BARE.GZ
   ```

3. Invocar el programa RAWRITE para hacer la copia de este fichero al disco flexible.

   ```
   D:RAWRITE
   ```

Posteriormente se introduce un disco en blanco y se contesta a las dos preguntas que se nos hace con el nombre del fichero (BARE) y la unidad de destino (A:). RAWRITE hará una copia de este fichero a nuestro disco. Si apareciera algún mensaje de error durante este proceso es mejor cambiar el disco flexible por otro y repetir la copia.

La creación del disco root esencialmente consta de los mismos pasos, en este caso acudiremos al directorio RROTDISK en el que encontraremos varios ficheros de instalación (setup).

En este caso el fichero más común es color.gz, que está indicado para monitores en color, text.gz es para monitores en blanco y negro.

Tenemos que coger este fichero, y repetir los mismos pasos descritos para la creación del disco boot, excepto en lo que se refiere a su descompresión, este fichero no hay que descomprimirlo, hay que copiarlo (RAWRITE) a la unidad A: tal como está.

Llamaré la atención sobre uno de los ficheros contenido en este directorio, umsdos.gz que permite la instalación en una partición dedicada al sistema operativo DOS, es decir, no hemos de perder la información de nuestra partición en el proceso de redistribución del espacio, para el DOS el espacio que dediquemos al LINUX será un directorio en su estructura de ficheros, pero esto se comentará con más detalle a continuación.

2.3 Creación de las Particiones.

2.3.1 El área de intercambio con memoria principal.

Previamente a esta parte del proceso de instalación se describirá brevemente este concepto y su forma de plasmarlo bajo el punto de vista del LINUX.

Algunos sistemas operativos utilizan una zona de la memoria secundaria para almacenar temporalmente zonas de memoria principal que debe desocupar por necesidades de la carga del sistema que se está ejecutando ese momento. Todo esto definido de manera tan sucinta hay que señalar que depende de otros conceptos como la política de gestión de memoria, gestión de procesos, la capacidad RAM del sistema, el tipo de hardware disponible y la forma en que cada sistema operativo ha dado solución a estos problemas.

Esta posibilidad se debe ver desde el punto de vista práctico como el poder incrementar la memoria disponible en nuestro ordenador, sin necesidad de invertir en hardware adicional, el sistema operativo dispondrá de una cantidad de memoria concreta sin tener que ser toda física.

Con LINUX podemos habilitar esta zona como partición dedicada exclusivamente a esta tarea o como un fichero más de nuestro disco duro. Para configurar este área lo haremos dentro del proceso de instalación en el primer caso8 desde el programa setup y desde la línea de comandos para el segundo.

Para crear un fichero de, por ejemplo, ocho Megabytes de capacidad y destinarlo a este fin la secuencia de comandos será:

```
dd if=/dev/zero of=/swap bs=1024 count=8192
sync
mkswap -c /swap 81929
```

Con el primer comando se ha creado un fichero vacío del tamaño deseado, el segundo comando sirve para prevenir errores10 y el tercero es para que el sistema operativo reconozca este área. Una vez creado e inicializarlo hay que activarlo para hacerlo accesible al sistema:

```
swapon /swap
```

Si lo que deseamos es eliminar este área hay que, en primer lugar, desactivar este área:

```
swapoff /swap
```

para posteriormente eliminar el fichero con el comando rm swap, ¡Nunca hacerlo al contrario!, podría traer como efecto la pérdida de información de algunos de los programas que en ese momento estuviésemos ejecutando.

Si queremos que este fichero de intercambio lo reconozca nuestro sistema cada vez que arranque hemos de incluir la siguiente línea en el fichero /etc/fstab:

```
/swap     none     swap     defaults    1     1
```

y recordar el eliminarla si hemos borrado este fichero o queremos dejar su activación para cuando sea necesaria.

Hay que tener en cuenta a la hora de decidir el tipo de área que el acceso si está definida como una partición dedicada es mucho más rápido que si lo tenemos como un fichero, ya que de esta forma tiene todas las páginas de memoria salvadas de una forma contigua (= mayor velocidad), se evita la carga que supone el utilizar sistema de ficheros mediante el manejador conveniente, para acceder a su vez al fichero de intercambio, además que en el fichero la ubicación física de la información no tiene porque ser contigua.

Sobre el espacio que debemos dedicar a este área, decir que no hay una regla general a aplicar. Normalmente si queremos utilizar el entorno gráfico X Window y las aplicaciones que lo acompañan, cuatro Mb. de memoria RAM es poca memoria, se recomiendan como mínimo ocho. Si tenemos ocho Mb. podemos definir ocho más.

Alguna documentación indica como regla general el definir un área de intercambio con el doble de tamaño que la memoria física disponible.

LINUX utilizará todas las particiones y ficheros de intercambio que le definamos durante su utilización, podemos tener dos particiones y tres ficheros destinados al área de intercambio (aunque esta configuración no tiene mucho sentido), el límite para esta versión es de 8 dispositivos (ficheros o particiones) con un tamaño máximo por dispositivo de 128 Mb. aproximadamente.

2.3.2 Creación de las particiones.

Una vez creados los discos boot y root atendiendo a la configuración del sistema vamos a proceder a detallar lo mejor posible el proceso de instalación del software del sistema operativo.

Lo primero que hay que hacer es, una vez introducido en la disquetera el disco de arranque (boot), reiniciar el ordenador. Una vez hecho esto y tras una breve espera deberá aparecer en nuestra pantalla los siguientes mensajes:

Esta pantalla primero da la bienvenida y luego indica que ahora es el momento para la entrada de cualquier parámetro extra que necesitemos introducir a la hora de la iniciación del sistema. Indica que hay tres palabras autorizadas, root, mount y drive2 para este fin. Estos parámetros indicarán al sistema donde deseamos tener la partición principal (o de arranque), que por defecto es la /dev/hda1 si no se tratase de otra, si hay introducido el disco root en una segunda unidad de disco, etc...11

En condiciones normales no se tendrá que introducir ningún parámetro adicional de estas características, por lo que simplemente pulsaremos la tecla ENTER, tras lo cual se muestra el siguiente mensaje:

```
Loading ramdisk
```

seguido de una serie de puntos como indicación de el proceso de carga en memoria, este proceso puede durar unos segundos dependiendo de la velocidad de nuestra máquina.

Tras esto aparecerán por pantalla una serie de mensajes indicando todo el hardware de nuestro ordenador que está identificando el sistema durante el arranque al igual que mostrará alguna de las utilidades que ya están soportadas por el núcleo. Posteriormente se muestra un mensaje pidiendo que cambiemos el disco por el de root y pulsemos ENTER:

```
Insert RAMDISK floppy and press ENTER.
```

Tras la carga en memoria de este disco aparecerá de nuevo un mensaje de bienvenida propio de la distribución además de cuatro notas muy importantes a tener en cuenta antes de ejecutar el programa setup. Una de ellas se refiere a la creación de las particiones necesarias, tanto la principal como la de intercambio, la segunda a cómo activar este área de intercambio, la tercera invita a

```
 LILO.
 Welcome to the Slackware Linux 3.0.0 bootkernel disk!
 If you have any extra parameters to pass to the ker-
nel, enter them at the
 prompt belox after one of the valid configuration na-
mes (ramdisk,
 mount, drive2)

 Here are some examples (and more can be found in the BOO-
TING file):
 ramdisk hd=cyl,hds,secs...    (Whe-
re "cyl", "hds", and "secs" are
 the number of cylinders, sectors, and heads on the dri-
ve. Most
 machines won't need this.)
 In a pinch you can boot your system with a command like:
 mount root=/dev/hda1
 On machines with low memory, you can use mount root=/dev/fd1 or
 mount root=/dev/fd0 to install without a ram-
disk. See LOWMEM.TXT for
 details
 If you would rather load the root/install disk from your second
 floppy drive:
 drive 2  (or even this: ramdisk root=/dev/fd1)
 DON'T SWITCH ANY DISKS YET! This prompt is just for en-
tering extra
 parameters. If you don´t need to enter any parame-
ters, hit ENTER to
 continue.
 boot:
```

Figura 2.2: Mensaje de bienvenida de la distribución Slackware.

ejecutar el programa de instalación y, por último, se indica la forma de configurar el sistema si nuestro ordenador tiene el monitor en blanco y negro (puede retirar el disco si lo desea). El sistema mostrará como mensaje:

```
slackware login:
```

Aquí hemos de introducir la palabra root para poder acceder al sistema al igual que en cualquier máquina UNIX.

Como primer paso se empezará por definir las particiones que sena necesarias tanto para instalar el software como para el área de intercambio con memoria, teclearemos fdisk ¡unidad¿, donde unidad será la que vamos a destinar como destino de nuestra instalación, de no indicar nada, tomará la unidad por defecto.

Este mensaje avisa que las características del disco duro con el que vamos a

```
Using /dev/hda as default device!
The number of cylinders for this disk is set to 1654.
This is larger than 1024, and may cause problems with:
1) software that runs at boot time (e.g., LILO)
2) booting and partitioning software form other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):
```

Figura 2.3: Mensaje inicial de fdisk..

trabajar pueden afectar al tipo de programas que se mencionan como ejemplo, también muestra el mensaje a la espera de un comando.

En este caso si introducimos la letra sugerida 'm' se presentará el siguiente menú:

```
Command action
                a   toggle a bootable flag
                c   toggle the dos compatiblity flag
                d  delete a partition
                l    list known partition types
                m  print this menu
                n   add a new partition
                p   print the partition table
                q   quit without saving changes
                t    change a partition's system id
                u   change display/entry units
                v   verify the partition table
                w   write table to disk and exit
                x   extra functionality (experts only)

       Command (m for help):
```

Figura 2.4: Menú principal de fdisk.

De esta lista, las opciones más interesantes son:

- 'd' para borrar una partición.
- 'l' para listar el tipo de particiones reconocidas.
- 'm' para cambiar el tipo de partición.
- 'n' para crear una nueva partición.
- 'p' muestra el contenido de la tabla de particiones.
- 'q' salir sin grabar los datos.
- 't' cambiar el tipo de partición.

```
    Disk /dev/hda: 16 heads, 63 sectors, 1654 cylinders
    Units = cylinders of 1008 * 512 bytes

    Device Boot  Begin   Start     End  Blocks   Id  System
    /dev/hda1   *       1       1    1016  512032+   6  DOS 16-
bit >=32M

    Command (m for help):
```

Figura 2.5: Información mostrada sobre el disco duro.

- 'w' salir grabando los cambios realizados.

Si elegimos la opción 'p', obtendremos por pantalla algo similar a la12 información mostrada en la figura siguiente:

Mensaje que indica el nombre de la unidad, su geometría y los valores referidos a la única partición existente, en este caso dedicada al sistema operativo DOS y marcada como partición de arranque.

Empezaremos creando la partición que vamos a dedicar al sistema de ficheros, para esto hemos de tener muy en cuenta todo lo descrito sobre el problema de algunas BIOS con los discos duros de alta capacidad.

Elegiremos a este fin la opción 'n', tras lo cual aparecerá el mensaje:

```
Command (m for help):  n
Command action
e   extended
p   primary partition (1-4)
```

Figura 2.6:

Aquí podemos definir el tipo de partición que deseamos para la instalación, elegiremos la opción 'p' para instalarlo en una partición primaria, se mostrarán estos tres mensajes, uno tras otro, a los que deberemos responder, para el ejemplo, con los siguientes valores:

```
    Partition number (1-4): 2 <¿>
    First cylinder (1017-1654): 1017 <¿>
    Last cylinder or +size or +sizeM or +sizeK (1017-
1654): +289M <¿>
```

Figura 2.7:

Se ha definido la partición número dos, que comienza en el cilindro número 1017, el primero disponible después de la partición numero uno, con un tamaño de 289 Mbytes, el tamaño de la partición lo podemos especificar con el valor del último cilindro, tamaño en bytes, Megabytes o en Kilobytes como se puede apreciar en el formato de los parámetros que acepta el programa.

Ahora se procederá a definir la partición de intercambio (swap) en este caso con un tamaño de 24 Mbytes, para averiguar el valor a introducir como primer cilindro hay que recurrir a la opción 'p'. Para ello los primeros pasos son iguales a los que hemos visto anteriormente:

```
    Partition number (1-4): 3 <¿>
    First cylinder (1024-1654): 1024<¿>
    Last cylinder or +size or +sizeM or +sizeK (1024-
1654): +24M <¿>
```

Figura 2.8:

Una vez creadas las dos particiones echaremos una ojeada al estado de la tabla de particiones:

```
    Disk /dev/hda: 16 heads, 63 sectors, 1654 cylinders
    Units = cylinders of 1008 * 512 bytes

    Device Boot  Begin   Start     End  Blocks   Id  System
    /dev/hda1   *       1       1    1016  512032+   6  DOS 16-
bit >=32M
    /dev/hda2        1017    1017    1604  296352   83  Li-
nux native
    /dev/hda3        1024    1605    1653   24696   83  Li-
nux native
```

Figura 2.9:

Ya están creadas las dos particiones que LINUX va a utilizar, como último paso se definirá la partición '/dev/hda3' como destinada a swap, para ello se utiliza a la opción 't', tras ello hay que introducir los siguientes parámetros:

```
    Command  (m for help): t
    Partition  number   (1-4): 3
    Hex code (type L to list codes). 82
```

Figura 2.10:

Con ello ya se tiene la tabla de particiones configurada para el objetivo de este punto:

Disk /dev/hda: 16 heads, 63 sectors, 1654 cylinders Units = cylinders of 1008 * 512 bytes

Tras esto, si todo ha ido bien, abandonaremos del programa con la opción 'w' que grabará la tabla de particiones que hemos definido, conviene anotar

```
    Device Boot  Begin   Start     End  Blocks   Id  System
    /dev/hda1   *       1       1    1016  512032+   6  DOS 16-
bit >=32M
    /dev/hda2       1017    1017    1604  296352   83  Li-
nux native
    /dev/hda3       1024    1605    1653   24696   82  Li-
nux swap
```

Figura 2.11:

el número de bloques que tiene el área de intercambio si se desea activar este área a través de la línea de comandos, aunque el programa setup lo hace automáticamente en uno de los pasos de la instalación.

Si se necesita activar el área de intercambio, debido a la cantidad de memoria RAM disponible, introducir los siguientes comandos (una vez hemos abandonado el programa fdisk):

```
\small
 # mkswap -c /dev/hda3 24696
 # swapon /dev/hda3
```

Recordar el no acceder a esta opción dentro del programa de instalación si ya tenemos activada este área.

Por último decir que esto es lo que se podría considerar como una distribución del espacio del disco duro "típica", pero no tenemos porqué ceñirnos a ésta si lo deseamos, podemos definir primero el área de intercambio y después la destinada al sistema de ficheros, puede crearse más de una partición para los ficheros y otras combinaciones que la experiencia o nuestras necesidades puedan plantearnos13.

2.4 Instalación

añadir/buscar breve explicacion sobre instalacion de Red Hat, Suse y Debian...

Capítulo 3

Algo de Configuración.

3.1 Introducción.

Este capítulo tercero trata de ser un somero repaso a la configuración y uso del LINUX para adaptarlo a nuestras necesidades del día a día.

Comienza explicando los pasos necesarios para la compilación del núcleo del sistema operativo, conviene recordar que una de las características principales del LINUX es el llevar consigo el código fuente completo además de una serie de utilidades que nos pueden ayudar a su construcción, de esta forma tendremos un núcleo totalmente ajustado a nuestro hardware, además de permitirnos el poder modificar este código fuente para poder compilarlo y utilizarlo posteriormente (si todo ha ido bien). El código fuente viene en forma de ficheros en lenguaje C y ensamblador que serán procesados con la ayuda de la utilidad make que lo acompaña por el compilador (gcc) que viene con la distribución. Junto con este apartado también se detalla el mecanismo de los módulos dinámicos y las diferentes posibilidades que existen para el arranque del sistema.

En el tercer apartado se describirán los diferentes tipos de sistemas de ficheros que LINUX puede manejar y la forma en que han de ser tratados para que el sistema operativo los reconozca.

Para la gestión de usuarios veremos los pasos necesarios para dar de alta a un usuario, eliminarlo, el programa sudo , que permite ejecutar ciertos comandos reservados al administrador a un usuario distinto además de conocer lo que es un grupo y la forma de crearlos, borrarlos y sus implicaciones.

Se detalla el mecanismo de arranque y parada del sistema dando un repaso a los ficheros implicados en este proceso. Estos ficheros vienen con un contenido determinado en la distribución pero pueden ser modificados si es necesario. Quizá el número de ficheros implicados en este mecanismo pueda parecer algo grande en un principio, pero una vez conocida su función seremos capaces de introducir cualquier modificación.

Quizá lo primero que hay que decir es que con LINUX el usuario se convierte al mismo tiempo en administrador, en algunas ocasiones tendrá la necesidad de acceder a la máquina mediante la cuenta principal (root) para realizar este tipo de tareas por lo que es necesario conocer muy bien aquello que se va a realizar, un error en cualquiera de los procesos descritos puede producir problemas por lo que el siguiente apartado ofrecerá un repaso a diferentes métodos destinados a poder recuperar el sistema ante cualquier imprevisto.

Los apartados siete y ocho tienen que ver con la capacidad que tiene LINUX para la gestión de redes y el soporte para distintos tipos de protocolos, los pasos básicos necesarios para la conexión a Internet y hacer uso del software disponible para usar en los servicios que esta red ofrece.

Insisto por último en la idea de que se evitarán muchos de estos imprevistos además de graves problemas si trabajamos al máximo desde un usuario distinto del administrador.

3.2 Compilar el núcleo.

Este es uno de los puntos principales que diferencian a LINUX del resto de sistemas operativos, la disponibilidad de todo el código fuente del núcleo en lenguaje C y ensamblador además de todas las herramientas necesarias para su compilación. Otras implementaciones del sistema operativo UNIX (bajo las condiciones de diferentes licencias de uso) también tienen esta posibilidad aunque en algunos casos disponemos de los fuentes en forma de ficheros precompilados (objeto).

Como se comentó en el capítulo I el código disponible es el del LINUX en sí, sólo recordar que el nombre atiende al núcleo del sistema operativo aunque se utiliza para denominar todo el conjunto.

Hasta el momento hemos estado utilizando el núcleo que en su día escogimos de los disponibles en la distribución: bare.gz, abare.gz, ... de esta forma teníamos varias versiones del núcleo del sistema operativo ya compiladas y listas para su utilización, lo único que hicimos fue copiarla en un disquete y posteriormente aprovecharla en la creación del disco de arranque desde el programa SETUP o en la configuración de LILO.

Para poder disponer de todo el material necesario para este apartado hemos de instalar la serie K de la distribución Slackware en la que tenemos dos versiones del núcleo del sistema operativo, la 1.2.13 como última versión estable y la 1.3.20 por si queremos estar 'a la última' en cuanto a versiones del núcleo.

Una vez instalada la serie, si se accede al directorio /usr/src/linux, contiene ficheros con información adicional y los directorios que contienen cada una de los componentes que van a formar el nuevo núcleo.

La necesidad de crear una nueva versión de LINUX viene determinada por ciertas situaciones que pueden surgir y que a continuación describo:

- La versión de la que disponemos no puede manejar cierto tipo de hardware que tenemos instalado en el ordenador, hace falta una actualización a una versión superior.

- LINUX sigue sin reconocer nuestro lector de CD-ROM, tarjeta de sonido o cualquier otro componente a pesar de haber escogido la imagen del núcleo correcta según las recomendaciones.

- Mejora del rendimiento del sistema,ajustando al máximo la configuración, un núcleo configurado con características que el ordenador no va a manejar será más grande en tamaño, consumirá más recursos y, por tanto, ralentizará el sistema.

- Se va a utilizar LINUX para comunicaciones (instalada la serie N), por lo que necesitamos soporte para redes, sistema de ficheros nfs, TCP/IP, SLIP ....
- El hardware del ordenador ha sufrido algún cambio.

El número de situaciones ante las que se puede plantear la necesidad de recompilar el núcleo puede ser muy amplio, no obstante, es necesario aclarar que no siempre este proceso va a ayudar a solucionar los problemas, a veces la mayoría de dificultades son debidas a una mala configuración del sistema.

Otra posibilidad que se ofrece es la de poder crear varias versiones del núcleo del sistema operativo, cada una con unas posibilidades diferentes de forma que podemos escoger durante el arranque cual se a utilizar. Tanto LILO como Loadlin1 tienen la posibilidad de escoger entre varias imágenes del núcleo o por el contrario se puede disponer de varios disquetes de arranque.

Como ejemplo podemos tener tres opciones en el arranque del sistema, un núcleo que esté ajustado al máximo para el manejo de nuestro ordenador y otro igual pero con soporte para redes destinado a si nuestra máquina está conectada a una red, acceso a Internet,....

Esta situación tiene un problema, si el sistema arranca con el núcleo sin soporte para redes porque se va a trabajar con una máquina de forma aislada generará numerosos mensajes de error debidos a ficheros cargados durante la inicialización del sistema (demonios) destinados a la automatización de tareas en la gestión de las comunicaciones ya que seguiremos teniendo los ficheros que intervienen en el arranque configurados para la carga de estos programas y el núcleo no estará configurado para su manejo. Hay que distinguir lo que está configurado en los ficheros de arranque y lo que el núcleo puede gestionar.

En el directorio que contiene el código fuente podemos ver algunos ficheros cuyo contenido paso a detallar:

- Makefile y Configure: Ficheros que utilizaremos para la creación del núcleo.
- README: Instrucciones de como realizar este proceso y que hacer ante algún problema.
- README.modules: Breve descripción de como compilar el núcleo para que pueda manejar módulos dinámicos.
- CREDITS: Datos sobre los autores de algunas partes del código del núcleo.
- System.map: Información sobre estructuras de datos y funciones que forman parte del núcleo y la posición en memoria que ocupan.

También podemos ver algunos directorios que contienen:

- arch: Contiene el código de los ficheros que configuran LINUX para ser ejecutado en distintas arquitectura en este caso disponemos de Alpha, Mips, Sparc e i386 para los procesadores de la familia Intel. En el resto de la estructura de ficheros tenemos los fuentes para el LINUX 'genérico' por decirlo de alguna forma, en este directorio tenemos los ficheros que marcan la diferencia para las distintas arquitecturas.

- drivers: Aquí disponemos de la parte del núcleo destinada al manejo de dispositivos.
- fs: Los distintos sistemas de ficheros disponibles (minix, msdos, ext2, ...).
- include: Ficheros cabecera organizados según la arquitectura a la que van destinados.
- init: Contiene el fichero principal del código fuente (main.c).
- ipc: Implementa los distintos mecanismos destinados a la comunicación entre procesos que LINUX soporta.
- kernel: Fuentes del núcleo común a todas las arquitecturas.
- lib: Ficheros con los que se construye la librería estática 'lib.a' que posteriormente se incorpora al núcleo.
- mm: Gestión de la memoria en LINUX.
- net: Encontraremos la implementación de los protocolos que el sistema operativo puede manejar en cuanto a redes y comunicaciones se refiere.

El primer paso que tenemos que dar para compilar la nueva versión es establecer la necesidades, esto se realiza mediante la contestación a una serie de preguntas que se presentarán al teclear (ubicados en el directorio /usr/src/linux y como usuario root):

```
# make config
```

Tras lo cual se presentarán una por una las cuestiones a responder, la opción especificada entre corchetes es la opción por defecto, podemos aceptarla pulsando la tecla Enter o modificarla. En función de la contestación a alguna de las preguntas se pueden mostrar otras, por lo que el ejemplo a continuación mostrado no tiene porque coincidir con sus respuestas:

En la primera sección de las preguntas hay que responder si se desea que el núcleo simule el coprocesador matemático, soporte de discos flexibles y si nuestro disco duro o lector de disco óptico es IDE, tras lo cual se invita a leer un fichero y se concreta en las siguientes tres preguntas el tipo de unidades que posee el sistema. También se pregunta si el disco duro es del tipo XT.

Como paso posterior se ofrecen una serie de marcas y modelos de lectores de discos ópticos por si el que tenemos coincide con alguno de ellos.

A continuación escogeremos los sistemas de ficheros que queremos que el núcleo soporte, si no tenemos idea de cual escoger podemos teclear el comando mount que nos mostrará información sobre los sistemas de ficheros disponibles, como mínimo optaremos por éstos3.

Posteriormente se especifica el tipo de impresora y ratón disponible además de si existe tarjeta de sonido y unidades de cinta (para copias de seguridad) del las indicadas.

La última línea permite el configurar el núcleo para poder, posteriormente con ayuda de ciertas herramientas el estudiar a fondo su funcionamiento, si no es nuestra intención no la tenemos que aceptar pues el núcleo irá más lento además de ser más inestable.

En otras distribuciones este proceso se puede realizar en un entorno gráfico (en modo texto o en X-Window) además de existir otros programas (independientes de la distribución) destinados a este fin cuyo cometido es facilitarnos la tarea.

Una vez hemos realizado la tarea de responder a las preguntas hay que teclear dos comandos antes de pasar a crear nuestro LINUX 'personalizado'. La primera secuencia a introducir es:

Lo cual prepara el sistema para ser compilado mediante la configuración de ciertos parámetros, este proceso crea un fichero (.depend) con información referente a los ficheros fuente que van a ser incluidos en el nuevo núcleo, ayuda a la utilidad make a conocer que ficheros van a ser invocados durante la compilación y la relación entre todos los ficheros del código fuente implicados en este proceso (dependencias).

Si todo ha ido bien se tecleará el siguiente comando:

```
# make clean
```

De esta forma se asegura que el proceso de creación del núcleo partirá de cero, borra todos los ficheros que hayan podido ser creados anteriormente, incluso cualquier imagen del núcleo existente (vmlinux) así como módulos creados anteriormente, existe otra posibilidad tecleando la palabra mrproper en vez de clean, de esta forma se eliminan muchos mas ficheros (el fichero con la configuración del núcleo (/usr/src/linux/.config, el fichero .depend, información sobre los módulos,...) por lo que tendríamos que empezar de nuevo.

Tras estos pasos introduciremos el comando:

```
# make zImage
```

Se dispondrá, al final, de una nueva imagen del núcleo en el directorio /usr/src/linux/arch/i386/boot disponible para ser pasada a un disquete para el arranque, utilizada por Lilo o Loadlin. Para crear un disco de arranque podemos teclear directamente 'make zdisk' introduciendo previamente un disquete en blanco en la unidad o tecleando al final del proceso de compilación:

```
# dd if=zImage of=/dev/fd0 bs=8192
```

Conviene introducir el siguiente comando si va a ser el disco el mecanismo utilizado como sistema de arranque en todo momento:

```
# rdev -R /dev/fd0 1
```

Lo cual indica al núcleo que el dispositivo desde el cual vamos a realizar la inicialización es la disquetera.

Caso de tener instalado el gestor de arranque (Lilo) teclearemos 'make zlilo' con lo cual se realizará la tarea de compilar el núcleo para pasar a realizar los pasos necesarios para que Lilo disponga de la información necesaria con el fin de poder arrancar el sistema. Es muy importante tener en cuenta este último caso ya que de no hacerlo correctamente nos encontraremos que estamos inicializando el sistema con una versión del núcleo del sistema operativo anterior a la que hemos creado.

Sobre el tiempo que ocupa la compilación del núcleo no hay una regla general, depende del tipo de máquina en la que estemos ejecutando este proceso y de las opciones que hayamos habilitado. De forma general se puede decir que puede tardar desde 5 a 10 minutos en las máquina mas rápidas (486 a 100/125 Mhz y Pentium) hasta varias horas en un 386 a 16 Mhz con 4Mb de memoria RAM (la persona que tenia esta configuración dejaba al sistema compilando toda la noche). En el ordenador utilizado para parte de este proyecto (un 386 de AMD con coprocesador numérico y 8 Mb de memoria RAM) tarda, en una compilación total unos 50 minutos aproximadamente, habilitando ciertas capacidades de redes. Para el caso de un Pentium a 120 Mhz, con 16 Megabytes de memoria principal y con una configuración similar para el núcleo el tiempo es de 7 minutos4. Todo ello depende de la configuración deseada del núcleo, la máquina en la que estemos ejecutando la compilación, el área de intercambio con memoria principal definida, etc.

3.2.1 Módulos.

Los módulos dinámicos constituyen un intento por parte de los autores de LINUX de separar diferentes partes del mismo del núcleo con la posibilidad de poder utilizarlos en cualquier momento cargándolos previamente. De esta forma tenemos la ventaja de que no hemos de preocuparnos por si nuestro núcleo soporta determinado dispositivo o sistema de ficheros, en el caso de que así lo hayamos configurado simplemente tendremos que cargar el módulo necesario y a partir de este momento LINUX estará disponible para la nueva necesidad planteada sin tener que recompilarlo de nuevo. Esto permite la flexibilidad de tener un núcleo configurado con las mínimas necesidades que nuestro hardware nos impone (ahorro de recursos) y el poder utilizar este mecanismo para cualquier eventualidad o durante el tiempo necesario.

Sirva como ejemplo la siguiente situación:

* Estamos trabajando en nuestra máquina con LINUX y necesitamos acceder a un sistema de ficheros que el núcleo que estamos utilizando no puede manejar, simplemente tendriamos que cargar el módulo que soporta dicho sistema de ficheros e instantáneamente podriamos montar dicho sistema de ficheros y acceder al mismo. Una vez obtenida la información necesaria podemos si lo deseamos, descargar dicho módulo de forma que el núcleo vuelve a quedar como al principio.

Para poder disfrutar de esta capacidad hemos de contestar afirmativamente durante el instante de la configuración a la siguiente pregunta: "Set version information on all symbols for modules (CONFIG$_{M}ODVERSIONS)[n($".

Tras compilar LINUX ejecutaremos los siguientes comandos:

```
# make modules
# make modules_install
```

Hallaremos en el directorio /usr/src/linux/modules todos los módulos creados disponibles para su utilización (después de arrancar la máquina con el nuevo núcleo) además de que quedarán instalados en el directorio /lib/modules/1.2.13.

Si se va a utilizar esta posibilidad conviene ejecutar la siguiente línea de comandos desde cualquiera de los ficheros de arranque (p.ej. /etc/rc.d/rc.S): "/sbin/depmod -a"que creará la información conocida como dependencias y que sirve para que si un módulo necesita de otro para poder funcionar no debemos preocuparnos de su carga previa, el sistema lo hará por nosotros (ver fichero modules.dep en el directorio /lib/modules/1.2.13).

Para cargar un modulo (de forma individual) hemos de teclear desde la línea de comandos:

```
# insmod <nombre_del_módulo>
```

También podemos incluir este comando en los ficheros de arranque del sistema para cargarlo al iniciarse el mismo (rmmod sirve para la eliminación de un módulo y lsmod nos mostrará información sobre los módulos cargados en memoria actualmente, el directorio /proc también contiene información al respecto).

Disponemos asimismo de una utilidad que nos permite el manejar módulos mas sencillamente, esta se denomina modprobe y se invocará desde la línea de comandos seguido del módulo a probar. Con este mecanismo nos aprovechamos plenamente de las dependencias mencionadas anteriormente por lo que la gestión de módulos es mucho más sencilla. Caso de no encontrar o no cargarse correctamente un módulo se borrarán todos los cargados anteriormente a fin de que el último pueda funcionar.

3.2.2 Lilo.

Un Gestor de Arranque consiste en una utilidad que permite el tener instalado en un disco duro varios sistemas operativos y poder escoger entre ellos durante el arranque del sistema. Hasta el momento hemos visto que podemos tener una partición dedicada al DOS y varias a LINUX, en cuyo caso podemos optar por tener varios discos para el arranque del sistema o varias imágenes del sistema operativo, a utilizar con Loadlin. Pero puede surgir la necesidad de tener más sistemas operativos (DOS, OS/2, Linux, FreeBSD...) instalados en la misma máquina, por lo que necesitamos una herramienta rápida y flexible que nos permita escoger entre los mismos.

Lilo se ha convertido en el estándar en cuanto a este tipo de programas se refiere debido a que casi todas las distribuciones de LINUX lo incorporan, aunque algunas tienen la posibilidad de instalar por defecto otros gestores de arranque propios de la distribución.

Lilo en realidad denomina el conjunto de ficheros que van a ayudar en la tarea de configurar el arranque de la máquina, aunque se ha aplicado este nombre en concreto al programa que controla todo este mecanismo.

Antes de dar paso a describir brevemente la forma de instalar Lilo he de advertir que su uso es PELIGROSO siempre y cuando no sepamos bien lo que vamos a hacer o tratemos de hacer pruebas con el mismo. La tarea de un gestor de arranque consiste en instalarse en lugares como el sector de arranque principal8 del disco duro, a fin de posibilitar el escoger entre varios sistemas operativos indicando cual va a ser en ese momento nuestra partición de arranque. lo cual implica cierto riesgo si no hemos configurado bien el programa o tenemos algún problema con nuestro disco duro. A pesar de todo esto hay que decir que Lilo chequea el estado de nuestro disco duro y realiza copias de zonas críticas del mismo antes de instalarse, deteniendo el proceso de instalación y mostrándonos un mensaje de error caso de encontrar alguna situación anómala. Como primera precaución hemos de crear un disco de arranque por si las cosas no fueran tan bien como es deseable. Si no se está todavía demasiado familiarizado con LINUX puede crearse un disco de este tipo desde DOS siempre y cuando contenga el programa FDISK además de software adicional de utilidad.

En nuestro caso si después de instalar Lilo no arrancara el sistema puede intentarse una de estas dos cosas:

- Arrancar con el disquete LINUX y desinstalar Lilo con la opción 'lilo -u' que debe sacarnos del problema ya que restaurará el formato anterior del sector de arranque.
- Arrancar con el disquete DOS y teclear posteriormente 'fdisk /mbr'.
- Existe la opción de crear un disco con el contenido del MBR del disco duro para poder restaurarlo posteriormente para el caso en que Lilo no pueda desinstalarse automáticamente, el proceso es el siguiente:
    - Crear una copia del sector de arranque principal en disquete. tt # dd if=/dev/hda of=/dev/fd0 bs=512 count=1
    - Para restaurar el contenido original desde el disquete creado teclear:

      ```
      # dd if=/dev/fd0 of=/dev/hda bs=446 count=1
      ```
- Restaurar el contenido del sector de arranque modificado de nuestro disco duro a partir de la copia que Lilo realiza automáticamente al instalarse:

  ```
  # dd if=/boot/boot.0300 of=/dev/hda bs=446 count=1
  ```

Hay de estar muy seguros de que el contenido del fichero boot.0300 se corresponde con el fichero creado para nuestro disco duro (lo podemos comprobar viendo la fecha de creación del fichero), no es otro instalado por la distribución (Slackware no tiene este problema) y por supuesto nunca obtenerlo de una máquina distinta.

En este apartado se tratará de exponer lo más detalladamente posible el proceso de instalación de este software en un sistema con una partición primaria dedicada a LINUX y otra a DOS que es la configuración más común, otras opciones serán revisadas. Las posibilidades de Lilo son mucho más extensas y recomiendo el acudir a su documentación para conocer más a fondo las mismas. Incluso después de haber leído este apartado debería completar sus conocimientos a fin de poder enfrentarse a cualquier otra situación.

La secuencia de arranque normal de un ordenador con arquitectura de tipo PC consiste, a grandes rasgos, en que la rutina correspondiente de la ROM-BIOS tras el proceso de chequeo del hardware acude a la dirección 07C00 (hex.)

en busca del sector de arranque ubicado en el sector de particiones a fin de ejecutarlo lo cual provoca que el mismo se traslade a la dirección 9A000 (hex.), habilite ciertas estructuras de datos y cargue el código correspondiente al segundo sector de arranque (correspondiente a la partición donde de encuentra el sistema operativo a ejecutar) en la dirección 9B000 (hex.) ejecutándolo posteriormente. Lilo puede funcionar como sector arranque principal ubicándose en el MBR de la unidad o hacerlo en el sector de arranque de una partición primaria o extendida.

Una vez instalado el paquete correspondiente al software necesario (ubicado en la serie A3 de Slackware) tendremos el grueso de los ficheros instalados en el directorio /usr/lib/lilo incluyendo un subdirectorio /doc con la documentación en formato Latex, el fichero README del directorio principal contiene el manual completo de LILO en formato ASCII. En el directorio /usr/doc/lilo se encuentra el mismo contenido que el del directorio /usr/lib/lilo/doc aunque en este caso el primer directorio no ha sido instalado al hacerlo Lilo.

Existe la posibilidad de configurar Lilo desde la utilidad SETUP, tecleando Quickinst9 desde el directorio /usr/lib/lilo/doc o ejecutando el comando liloconfig del directorio /sbin pero no voy a contemplarlas en el desarrollo de este apartado debido a que se desaconseja el utilizarlas para configurar el gestor de arranque, pueden generar un fichero de configuración incorrecto y causar problemas. Dichas utilidades lo único que hacen es ver el estado del disco duro y hacernos varias preguntas en cuanto a posibilidades en el arranque del sistema se refiere pero puede ocurrir que no conozcamos el funcionamiento de este mecanismo, no tengamos claro lo que tenemos que hacer o el programa genere un posible fichero que se puede considerar estándar en cuanto a su contenido, en teoría cualquier sistema puede funcionar con el mismo, pero esto no es siempre así, es mejor generar nuestro propio fichero de configuración a partir de la información disponible.

En el directorio /boot tendremos los siguientes ficheros:

El primer fichero contiene información que Lilo va a utilizar para poder instalarse como sector de arranque; any$_b.bpermiteintercambiarladesignaciónde lasdosunidadesflexibles(A: \hat{U}B:)aligualqueany_d.blohaceconlasunidadesdediscorígido(C: \hat{U}D:);\ os2_d.bseutilizaparapodercargarelsistemaoperat$

Lilo únicamente puede ser instalado en:

- Sector de arranque de un disco flexible.
- MBR del disco rígido primero, (unidad C: para DOS y /dev/hda para LINUX).
- Sector de arranque de una partición primaria que contiene a LINUX.
- Sector de arranque de una partición de tipo extendida en el primer disco rígido.

Como se ha mencionado la opción más común consiste en instalar Lilo atendiendo a la opción número tres, aunque también podemos optar por la opción numero dos aunque de esta forma modificamos el MBR del disco duro, lo cual será siempre algo más 'delicado' a la hora de salvar el disco duro ante cualquier error.

El fichero en el que se ha de definir la forma en que deseamos utilizar el software se denomina /etc/lilo.conf el cual acepta gran cantidad de parámetros, los más usuales son:

- boot=¡dispositivo¿: Indica el dispositivo (unidad o partición) que contiene el sector de arranque que por defecto se trata de la que está montada como raíz.
- compact: Reduce el tiempo de carga de Lilo, recomendable cuando hemos instalado Lilo en disquete.
- delay=¡valor¿: Décimas de segundo que el sistema esperará antes de arrancar.
- disk=¡nombre¿: Nos permitirá el detallar la geometría de ciertos discos duros que Lilo no puede manejar normalmente debido al tipo de BIOS que tenemos instalada. Por ejemplo:

```
disk=/dev/hda
bios=0x80
sectors=32
heads=64
cylinders=632
```

- image=¡fichero¿: La imagen de LINUX a utilizar puede estar ubicada en una partición (a la que Lilo pueda tener acceso) o en un disquete, normalmente utilizaremos la primera opción por defecto.
- install=¡fichero¿: Que fichero va a instalarse como sector de arranque, por defecto se instalará el fichero /boot/boot.b.
- label=¡nombre¿: Asigna un nombre a la opción en la que se incluye, esto permite junto con prompt y timeout el poder escoger entre varias opciones durante el arranque del ordenador.
- linear: Algunas BIOS tienen formas de transformar el par dirección/geometría en los discos que no es compatible con Linux, para estos casos se recomienda utilizar esta opción junto con la geometría de la unidad (opción disk).
- map=¡fichero¿: Especifica la ubicación del fichero map que por defecto será /boot/map, creado por Lilo para contener información crítica del sistema.
- other=¡dispositivo¿: Se utiliza junto con el parámetro table para que Lilo pueda arrancar otros sistemas operativos (DOS en nuestro caso).
- root=¡dispositivo¿: Partición que va a ser montada como raíz del sistema de ficheros. De no indicarse se entiende que es en la que se encuentra Lilo en el momento de la instalación (más exactamente la que se encuentra configurada en la imagen del núcleo a utilizar y que podemos modificar con el comando rdev10).
- table=¡dispositivo¿: Dónde se encuentra la tabla de particiones, información que Lilo no maneja y que es necesaria para otros sistemas operativos.

Con las principales opciones del fichero de configuración entraré de lleno a describir un ejemplo: Tenemos un disco duro con una partición primaria dedicada a DOS (dev/hda1) y otra dedicada a LINUX (/dev/hda2) sin problemas en cuanto a la BIOS y los límites en el número de cilindros.

Para este caso el fichero será:

Aunque algunas entradas del fichero pueden ser eliminadas siendo el resultado el mismo las he dejado indicadas para más claridad.

Una vez creado este fichero instalaremos en el directorio raíz (principal) el fichero /usr/src/linux/System.map y el fichero con la imagen del sistema operativo (lo podemos obtener tecleando 'cp /usr/src/linux/arch/i386/boot/zImage /vmlinuz').

Posteriormente es necesario entrar como usuario principal (root) y teclear 'lilo' tras lo cual tendremos el sector de arranque instalado si no ha habido ningún problema, en tal caso Lilo no se instalará.

Por último activaremos la partición número 2 (/dev/hda2) como partición de arranque, esto lo podemos hacer desde el fdisk de LINUX con la opción 'a' que permite marcar/desmarcar las particiones para este fin (también podemos hacerlo desde el DOS).

Con todo esto se tiene el sector de arranque instalado en la segunda partición primaria de forma que hemos evitado el instalarlo en el principal (MBR) y en caso de algún problema fatal podemos volver a activar la partición DOS como de arranque y realizar las maniobras necesarias arrancando LINUX desde disquete.

Al arrancar el sistema de nuevo y tras el chequeo del hardware por parte de la BIOS observaremos la siguiente línea:

```
boot:
```

Tras lo cual, si no hemos introducido ninguna expresión (en este caso podemos teclear 'dos' o 'linux') tras 5 segundos de espera arrancará con el sistema operativo LINUX. Si se quitan las líneas prompt y delay no habrá período de espera y siempre arrancaremos LINUX a no ser que tengamos pulsada la tecla 'Shift' con lo cual se nos permitirá introducir manualmente la opción deseada.

Si se instala Lilo en el sector de arranque de una partición extendida los cambios son mínimos (sólo hemos de cambiar las líneas boot y root) pero hay que tener en cuenta que solo el fdisk de LINUX permite marcar este tipo de particiones como de arranque. También se nos provee del programa activate que permite el activar cualquier partición del disco duro.

Para el caso de querer arrancar con sistemas operativos ubicados en una unidad distinta de la primera el único lugar donde Lilo puede ser instalado es el MBR del primero (o en el sector de arranque de una partición extendida). La primera opción también podría haber sido utilizada para el caso anterior pero, como ya se ha señalado, nos evitamos el modificar el sector de arranque principal del disco duro.

Para el caso en que sea necesario instalar Lilo en un disco flexible lo cual puede ser útil para realizar pruebas de este software o en el caso de que existan problemas con la BIOS el mecanismo es muy simple, basta con tener los ficheros /boot/boot.b y una imagen del núcleo en el disco flexible (necesitamos un disco con formato y montado) e invocar Lilo con la ubicación de los ficheros modificada convenientemente en el fichero de configuración.

Crearemos un sistema de ficheros en el disco flexible y lo montaremos de forma que pueda ser accesible desde LINUX copiando los ficheros necesarios en el mismo:

```
# mke2fs /dev/fd0
# mount /dev/fd0 /mnt
# cp /boot/boot.b /mnt
# cp /usr/src/linux/arch/i386/boot/zImage /mnt/vmlinuz
```

Tras lo cual creamos el fichero lilo.conf se invoca el programa de forma que se instalará en el disco flexible y desmontaremos la unidad (umount /mnt).

Lilo, al igual que Loadlin, acepta el paso de parámetros al núcleo y al programa init durante el arranque del sistema, los básicos son los mismos que para Loadlin, para saber los todos parámetros que se pueden utilizar acudir al fichero BootPrompt-HOWTO.gz, ejemplo:

```
# boot:  linux single root=/dev/hda2
```

En la invocación del programa Lilo también podemos utilizar algunos modificadores que pueden ayudar a configurar el programa desde la línea de comandos directamente.

La opción '-t' es de gran utilidad pues Lilo realizará todo el proceso normal pero en modo de prueba de forma que no modificará nada de nuestro disco duro y, por tanto, no se instala pero ayuda a saber si lo puede hacer correctamente.

Con '-f' se señala un fichero donde tendremos almacenada la geometría del disco duro tal y como se ha indicado con el parámetro disk.

El modificador '-r' puede ser útil para indicar que la partición que contiene el directorio principal es diferente a la que actualmente tenemos activa.

Para las variables image, label, other y table no existe equivalente en la línea de comandos.

Lilo debe ser invocado tras una nueva compilación del núcleo del sistema operativo (a no ser que realicemos un 'make zlilo') y que debemos desinstalarlo previamente si vamos a eliminar la partición dedicada a LINUX y lo tenemos instalado como MBR o si vamos a instalar una versión superior del mismo programa.

Opciones en la línea de comandos

3.2.3 El Sistema de Ficheros.

LINUX es capaz de manejar múltiples dispositivos destinados al almacenamiento de información: discos duros, disquetes, discos ópticos, cintas... incluso existen manejadores para la última oleada de sistemas tales como discos magneto-ópticos, unidades ZIP, etc.. aunque para el manejo de estos dispositivos será necesaria una actualización del sistema operativo o software adicional a fin de poder manejarlos correctamente.

Al igual que existen varios tipos de dispositivos para el almacenamiento de información cada sistema operativo ha optado por almacenar esa información de forma distinta, creando su propio sistema de ficheros.

Cada uno de estos sistemas de ficheros tiene unas características que lo diferencia en cuanto a la forma de almacenar la información, que información guarda sobre cada unidad de almacenamiento (fichero), la forma de manejarla y el interfaz que pone a disposición del resto del sistema para su manejo.

Generalmente todos ofrecen una estructura lógica para organizar la información en forma de directorios y subdirectorios, capacidad de gestionar varios tipos de ficheros, información sobre la fecha y hora de creación de un fichero además de quien es su propietario, permisos de acceso, etc...

Antes de seguir con el desarrollo del apartado conviene hacer una aclaración previa, el uso de dispositivo, sistema de ficheros y unidad aunque a veces parezcan sinónimos o sean utilizadas indistintamente atiende a la flexibilidad en el lenguaje, con el riesgo que comporta de caer en la imprecisión. Un dispositivo es la parte hardware que puede almacenar información durante el tiempo necesario, hasta su borrado, unidad es la forma de denominar en algunos sistemas operativos a un dispositivo y en un dispositivo pueden coexistir varios sistemas de ficheros como en este caso que tenemos en nuestro disco duro 2 sistemas de ficheros (DOS, LINUX) y área de intercambio

Ante este panorama LINUX ofrece la mejor opción posible siendo capaz de manejar varios sistemas de ficheros al mismo tiempo. Los formatos que LINUX actualmente soporta son:

- ext2: El más utilizado para LINUX.
- ext: Versión anterior a ext2.
- minix: Sistema de ficheros del sistema operativo MINIX.
- xiafs: Parecido al ext2, pero poco usado.
- nfs: Permite el acceso a ficheros remotos en un entorno de red.
- umsdos: Dedicado a la instalación de LINUX en una partición dedicada al DOS.
- vfat/fat32: Sistema de ficheros propio del DOS/Windows.
- ntfs: Sistema de ficheros de Windows NT.
- hpfs: Utilizado por el sistema operativo OS/2 de IBM.
- iso9660: Formato usado por la mayoría de los discos ópticos.
- sysv: Acceso a ficheros del UNIX System V y variantes.
- xenix: Sistema de ficheros de una de las implementaciones del UNIX.
- coherent: Otra implementación del UNIX.

La versión del núcleo sobre la que se está desarrollando este trabajo (v.1.2.13) utiliza un mismo manejador para acceder a los sistemas de ficheros sysv, xenix y coherent.

En el directorio /usr/src/linux/fs podemos ver los subdirectorios que contienen el código fuente para los manejadores de los distintos sistemas de ficheros.

La gestión de los distintos dispositivos de almacenamiento requiere el introducir ciertos comandos como paso anterior y posterior a su uso, para entenderlo mejor no basta con introducir un disco flexible en la unidad correspondiente para poder acceder a la información que éste contiene.

Con LINUX, como con otras implementaciones del UNIX, tenemos que acostumbrarnos a conceptos como 'montar' y 'desmontar'13 dispositivos para poder utilizarlos.

El proceso de montado de un dispositivo, visto desde el punto de vista del usuario y, por tanto, muy genéricamente, consiste en que es 'asignado' a uno de los directorios que tenemos en nuestro disco duro o hemos creado a tal efecto, de esta forma para acceder a su contenido simplemente tenemos que movernos por el directorio escogido.

Las unidades a las que hasta el momento hemos tenido acceso al utilizar el sistema son montadas automáticamente por el proceso de arranque14 por lo que no ha hecho falta intervención alguna por nuestra parte, así mismo el sistema se encarga de desmontar estas unidades durante el proceso de parada.

Para poder realizar cualquiera de los dos procesos hay que acceder al sistema como administrador, cualquier otro usuario no puede realizar esta tarea excepto si es especificado en el fichero /etc/fstab. Este fichero contiene información sobre las distintas unidades disponibles en el sistema y es utilizado por los programas fsck (destinado al chequeo de los sistemas de ficheros para la detección de errores), mount y umount. Durante el arranque se accede al contenido de este fichero para montar las unidades especificadas, cada una con su sistema de ficheros propio.

El formato del fichero consiste en seis columnas separadas por carácteres de tabulación (no espacios). La primera de ellas especifica la partición a la que se hace referencia, observar que el directorio /proc no está asignado a ninguno (none) debido a sus especiales características; la segunda indica el punto (directorio) en el que vamos a montar el dispositivo; la tercera concreta el tipo de sistema de ficheros; la cuarta sirve para especificar opciones referidas al comando mount; el siguiente valor es utilizado por el comando dump (control de los buffers asignados a cada dispositivo y de la información que contienen). Por último tenemos un valor utilizado por el programa fsck como número de orden en el chequeo de los distintos sistemas de ficheros, los ubicados en el mismo dispositivo habrán de ser testados secuencialmente mientras que los que se encuentran en dispositivos distintos pueden ser chequeados en paralelo.

La palabra defaults colocada para todos los sistemas de ficheros indica que las opciones a utilizar por el comando mount son: rw, suid, dev, exec, auto, nouser y async, cuyo significado paso a describir:

- rw: Monta el sistema de ficheros para lectura y escritura.
- suid: Habilita el uso de los identificadores de usuario y de grupo para su uso cuando sean necesarios.
- dev: Interpreta correctamente los ficheros para el manejo de dispositivos de acceso por carácter y por bloque.
- exec: Permite la ejecución de los ficheros binarios.
- auto: Puede ser montado con la opción -a.

- nouser: Prohibe a un usuario distinto al administrador el montar el sistema de ficheros.
- async: Toda la entrada/salida al dispositivo se hará de forma asíncrona.

Otras opciones interesantes son:

- noexec: No permite la ejecución de ficheros.
- ro: Monta el sistema de ficheros para sólo lectura.
- user: Permite a cualquier usuario el montar/desmontar el dispositivo, implica los parámetros noexec, nosuid y nodev.

Como ejemplo supongamos que deseamos que la unidad lectora de discos ópticos pueda ser gestionada por cualquier usuario (al menos tenemos la tranquilidad de que no se puede perder información de este dispositivo) podemos añadir la línea siguiente a nuestro fichero fstab:

```
/dev/cdrom /mnt iso9660 ro,user
```

Recordar que el espacio entre las entradas debe introducirse con el tabulador.

Existen otras muchas opciones destinadas a cada uno de los sistemas de ficheros que LINUX puede soportar pero saldría del objetivo del capítulo el intentar explicarlas todas.

En el uso normal del sistema operativo podemos montar unidades no disponibles después del arranque del sistema utilizando mount desde la línea de comandos, su sintaxis es:

`# mount -t <sistema`$_{d}e_{f}icheros > dispositivopunto$

Para un disco flexible ubicado en la unidad conocida como A: utilizaremos el comando:

```
mount -t msdos /dev/fd0 /mnt
```

El disco que hemos montado que se supone con formato DOS, será accesible desde el directorio /mnt.

Para la unidad lectora de discos ópticos el comando será:

```
mount -t iso9660 /dev/cdrom /mnt
```

El punto (directorio) donde vamos a montar la unidad puede ser el que deseemos, podemos tener un directorio `/disco`$_{A}paraeldiscoflexibleyotro/cdrompara$

Si introducimos el comando sin parámetro alguno se nos mostrará información sobre todos los sistemas de ficheros accesibles en el momento.

Para desmontar una unidad hay que utilizar el comando umount que puede tener una de las formas siguientes:

`# umount /dev/fd0` o bien `# umount /mnt`

Cualquiera de las dos formas es válida, siempre y cuando se sepa bien lo que se ha indicado con cada una de ellas.

En el proceso de apagado o reinicialización del sistema se desmontan las unidades (especificadas en /etc/fstab) automáticamente.

¿Por que el montar/desmontar unidades?: Con el proceso de montado de una unidad la asignamos a uno de los ficheros ubicados en el directorio /dev.

En el uso del sistema de ficheros, si no ha sido especificado de otra forma, la lectura y la escritura se realizan de forma asíncrona, esto es, hay una zona de memoria (buffer) destinada a hacer de intermediaria entre el dispositivo y la parte del núcleo encargada del sistema de ficheros conteniendo información en todo momento proveniente de la interacción de los procesos con el manejador de ficheros, por lo que la E/S del sistema se realiza durante periodos de inactividad o baja actividad por parte de la CPU, de esta forma se evita una ralentización del sistema por un acceso a un dispositivo debido a la espera que esto puede suponer.

Al montar la unidad se le asigna una zona de memoria para estas tareas, si no desmontamos la unidad antes de su extracción de la unidad lectora o antes del apagado de la máquina podemos perder la información que ha quedado ubicada en este buffer y que todavía no ha sido actualizada en el dispositivo además de poder tener asignado un fichero /dev/xxx a un dispositivo que no existe (hemos extraído el disco), con lo que este dispositivo contiene información incorrecta. Todos estos factores pueden ser portadores de problemas.

Si lo que deseamos es utilizar un disco flexible para el soporte de ficheros podemos utilizar mkfs para la creación de un sistema de ficheros. Existe una 'versión' de mkfs (directorio /etc/fs/) para cada uno de los diferentes sistemas de ficheros que podemos necesitar, simplemente teclear:

```
# mkfs.msdos /dev/fd0
```

Tendremos creado un sistema de ficheros con formato DOS en el disco flexible de la unidad A:.

Previamente habremos dado formato a este disquete utilizaremos fdformat así evitamos posibles errores debidos a sectores defectuosos o cualquier otro problema físico del disquete. Para ello teclear:

```
# fdformat /dev/fd0
```

Para el uso de estos dos comandos la unidad A: no ha de estar montada, de otra forma ambas utilidades muestran un mensaje de error.

3.3 Gestión de Usuarios

Si se piensa que existe sólo un usuario del sistema siempre y cuando no esté conectada la máquina al exterior, teclear la siguiente orden:

```
# groups
```

La línea que mostrada a continuación contiene grupos de usuarios del sistema, el resultado de este comando será algo similar a:

```
root bin daemon sys adm disk wheel floppy ...
```

¿Grupos de usuarios? ¿Hay más usuarios utilizando el sistema? ¿Que significa este mensaje?

Si se ha seguido al pie de la letra los pasos descritos en el Capítulo II actualmente tiene creados dos cuentas de usuarios, el administrador (root) además de una con sus datos para tareas no administrativas además de haber tenido el primer contacto con el mecanismo para la creación de usuarios.

Una cuenta para un usuario no solo se refiere a dar de alta a éste para que pueda tener acceso al sistema, directa o indirectamente tiene las siguientes consecuencias:

- Controla el acceso al sistema, prohibiendo a usuarios no autorizados el acceso a la máquina o restringiendo su acceso al máximo.
- Permite el control de los usuarios al conocer la identidad de los que se hallan dados de alta como las actividades que pueden realizar.
- El usuario tiene la seguridad de que su trabajo e información no está al acceso de cualquier curioso (excepto el administrador).
- Cada uno dispone de un directorio de trabajo propio (/home/<usuario>).
- Los ficheros creados por un usuario podrán ser controlados por su propietario (y el administrador) asignándoles los permisos de lectura, escritura y modificación que crea convenientes.
- Los ficheros disponibles en el sistema de ficheros, además de los permisos de lectura, escritura y modificación propios de cualquier sistema operativo UNIX, pueden ser habilitados de forma que puedan ser accedidos por grupos de usuarios.

Aunque de momento no tengamos la máquina conectada de alguna forma con el exterior no por ello este capítulo deja de ser de interés debido a que con los mismos conocimientos puede gestionar a cualquier número de usuarios, no debe influir el número de éstos, conocerá el mecanismo de creación de cuentas de forma que puede entender y modificar a su gusto este proceso además de servir de base a la hora de configurar su ordenador el día que necesite conectarse a Internet, su BBS preferida, tenga que enfrentarse a la tarea de administrar una red.

No sólo existen usuarios en cuanto a personas se refiere sino que también existen usuarios que no tienen ninguna relación con éstas. Existen otros usuarios que el sistema operativo crea para la gestión del sistema de ficheros, conexiones con el exterior, control de la impresora y otros periféricos... Estos usuarios generalmente son los denominados demonios, programas que se inician en el arranque del sistema (o por ejecución manual), trabajan en segundo plano y se encargan de este tipo de tareas. Si hemos trabajado algo con LINUX quizá nos hayamos encontrado con ficheros y directorios cuyo propietario es algún usuario de este tipo.

3.3.1 Alta de Usuarios.

Crearemos un usuario como repaso al procedimiento general, mediante el comando adduser:

Tras la introducción de los campos que se piden, algunos como el GID y el UID conviene aceptarlos a partir del valor que se nos ofrece si no conocemos su significado, el programa creará el directorio para el usuario, además de introducir una serie de ficheros de configuración de ciertas utilidades. Estos ficheros se encuentran en el directorio /etc/skel (skel de skeleton=esqueleto) y pueden introducirse todos aquellos ficheros que necesiten compartir todos los usuarios creados.

El usuario puede posteriormente modificar su palabra clave si lo desea accediendo al sistema y utilizando el comando passwd.

Si cualquier usuario olvidase su palabra clave, se puede acceder al fichero /etc/passwd donde se hallan especificados todos los usuarios del sistema y la palabra clave que utilizan (encriptada), se puede borrar esta palabra de forma que el usuario pueda introducir una nueva. Para el caso de ser el administrador la tarea es algo más complicada pues debe arrancar desde un disquete (puede utilizar los disquetes creados para la instalación de LINUX), montar la unidad en donde se encuentre la partición principal y acceder a este fichero para realizar la misma maniobra.

3.3.2 Anulación de Usuarios.

Para anular la cuenta de un usuario se deben realizar los siguientes pasos:

1. Eliminarlo del fichero /etc/passwd y de ser necesario del fichero /etc/group.

2. Borrar su directorio de trabajo y cualquier fichero creado por este usuario que se encuentre en otro directorio.

Para anular temporalmente a un usuario sin borrar la información disponible sobre éste, lo podemos hacer introduciendo como primer carácter del campo clave en el fichero correspondiente un asterisco ('*'), por ejemplo:

```
jose:*DfK/OUxSRLi/2:501:100:Jose Manuel Laveda Molina:/home/jose:/bin/bash
```

3.3.3 Grupos.

La agrupación de todos los usuarios que tienen acceso a un sistema gestionado con LINUX permite organizarlos de forma que su control sea más sencillo.

Un grupo de usuarios puede trabajar compartiendo ficheros y directorios de forma que pueden desarrollar trabajos conjuntamente. Para en control de los grupos tenemos un fichero /etc/groups en el que podemos especificar los componentes de un grupo, además podemos introducir una clave para el grupo de forma que puedan acceder otros usuarios no pertenecientes al mismo.

El fichero /etc/groups tiene la siguiente sintaxis:

nombre$_{d}el_{g}rupo : clave : gid : lista_{m}iembros$

Los campos nombre$_{d}el_{g}rupoylista_{m}iembrossonbastanteautoexplicativos.$

El campo clave raramente es usado, si no esta especificado estará en blanco o contendrá un asterisco ('*') en su lugar. Si necesita introducir una clave en este campo para un grupo tendrá que realizar los siguientes pasos:

- Crear un usuario con la clave que desee introducir en el grupo, este usuario lo eliminará después pues solo es creado para este objetivo.
- Copiar el campo clave (ya encriptado) del fichero /etc/passwd en el lugar de la clave del fichero que estamos modificando.

Si un grupo tiene una clave definida, y nuestra cuenta no está incluida en él veremos el siguiente mensaje al intentar cambiar de grupo de trabajo:

```
# newgrp <grupo>
Password: password for group
# _
```

Un apunte sobre los permisos y propietarios en los ficheros, la precedencia es la siguiente: tipo de fichero (archivo normal, b=archivo especial por bloques, c=archivo especial por carácteres, d=directorio, l=enlace simbólico), permisos habilitados para el propietario del fichero, los del grupo y, por último, para el resto de usuarios del sistema, todos ellos indicados en grupos de tres carácteres (r=lectura, w=escritura y x=ejecución).

```
-rwxr-xr--  1  jose  proyecto   24856  Jun  2  15:30  prueba
```

Para el ejemplo tenemos un fichero normal llamado prueba cuyo propietario es el usuario 'jose' que pertenece al grupo 'proyecto'. El fichero tiene permisos de lectura, escritura y ejecución para su propietario, para el grupo tiene permisos de lectura y ejecución, sólo de lectura para el resto de usuarios. El comando utilizado para cambiar los permisos de un fichero es chmod, ejemplo:

```
# chmod a=rwx <archivo>
```

Concede permisos de lectura, escritura y ejecución para todos los usuarios (a). Los carácteres utilizados son: 'a' para indicar todos los usuarios del sistema, 'g' para el grupo, 'o' para el resto de usuarios y 'u' para el propietario del fichero.

Para cambiar el propietario de un fichero utilizaremos el comando chown con la siguiente sintaxis:

```
# chown <usuario>.<grupo> <archivo>
```

Dispone igualmente del comando chgrp para cambiar el sólo el grupo al que pertenecerá el fichero, introduciendo la línea siguiente podemos cambiar el grupo para el fichero prueba:

```
# chgrp otros prueba
```

Si observamos el contenido del fichero /etc/grups instalado en nuestro sistema veremos que la gran mayoría de grupos definidos en el fichero se refieren a los creados por programas dedicados al correo electrónico, news, gestión de impresora, etc... Sólo existen dos grupos para usuarios 'normales', users y root en este caso.

3.4 Arranque del Sistema. Init.

Una vez instalado el software que forma parte del LINUX y elegida la forma de inicializar el sistema (Loadlin, gestor de arranque ó disquete) cada vez que hemos realizado este proceso se nos ha presentado una pantalla muy parecida a esta, el contenido exacto de los mensajes variará dependiendo del tipo de hardware sobre el que tenemos instalado el sistema operativo:

La figura muestra la información extraída por el LINUX en cuanto al hardware en el que se va a desenvolver, toda esta información es buscada y mostrada por el núcleo del sistema durante la carga del mismo en memoria principal. El resto de mensajes que se puedan apreciar durante el arranque dependerá del contenido de los ficheros de configuración que intervienen en este paso.

Si se quiere ver esta información, ya que durante el arranque puede ser difícil de leer, o hay algún tipo de problema, hay acceder al fichero messages del directorio /var/log en el que están registrados la fecha de todas las veces que se ha inicializado el sistema y la información que se ha obtenido.

Aunque no se mencione explícitamente para cada uno de los ficheros y utilidades que se van a describir a continuación conviene dejar claro que la mayoría de ellos necesitan los privilegios de que dispone el usuario root para poder acceder a los mismos. De otra forma los ficheros descritos puede que sólo tengan el permiso de lectura para el resto de usuarios o ningún permiso para ellos, por lo que son inaccesibles si utilizamos LINUX con otro usuario distinto al administrador.

En la figura se puede ver la información mostrada por el núcleo para el ordenador con el que ha sido desarrollado parte de este trabajo.

Tras mostrar la configuración del tipo de resolución que va a utilizar para el trabajo en modo texto (80 columnas x 25 filas) pasa a calcular la velocidad de la UCP16 (Unidad Central de Proceso) del ordenador en BogoMips, unidad propia del LINUX que intenta ser una aproximación a los millones de instrucciones por segundo que la unidad puede calcular. La exactitud y fiabilidad de esta unidad no es muy buena como se describe en el fichero destinado a la explicación de este parámetro17.

Posteriormente se nos muestra información sobre los puertos disponibles y la interrupción a la que están asociados, además del modelo de controlador que los maneja (en este caso un chip 16450).

Sobre la impresora la información es similar, mostrándonos la dirección de memoria relacionada con el puerto al que esta conectada.

Las líneas siguientes muestran información sobre todas las unidades para el almacenamiento de información disponibles, en primer lugar el disco duro, después la unidad de CD-ROM y por último las dos unidades de disquete.

El resto de información se refiere a la memoria disponible, la detección del co-procesador numérico instalado, la versión del sistema operativo que estamos utilizando y cuando fue compilada, las particiones instaladas, el área de intercambio instalada y la detección del disco óptico instalado en la unidad lectora en el momento del arranque.

Una vez se ha completado esta etapa del arranque del sistema el núcleo ejecutará el programa init que será el encargado de configurar el resto de parámetros del sistema ayudado del fichero /etc/inittab y todos los que lo complementan.

3.4.1 El programa INIT.

El núcleo hace que, al invocarse este programa, pase a ser el padre de todos los procesos a ejecutarse durante la utilización del sistema operativo, este será el denominado en la bibliografía "proceso 0", todos los procesos creados a partir de este momento serán hijos de este proceso, creándose un árbol de procesos encabezado en todo momento por init.

Init, ahora como proceso, creará los sub-procesos necesarios para la espera de usuarios para el acceso a las terminales virtuales, fijará el nivel/es de ejecución de los distintos procesos, ejecutará las instrucciones contenidas en otros ficheros destinadas a montar las distintas unidades especificadas, chequear estas si es necesario....

Una vez que todos los procesos especificados en los ficheros han sido ejecutados init espera que sucedan tres cosas: que uno de los procesos hijo finalice, un fallo en el suministro eléctrico o que el nivel de ejecución de los procesos sea modificado mediante los programas init/telinit.

Ante la invocación de este programa y un cambio del nivel de ejecución por parte de éste init enviará la señal SIGTERM a todos los procesos que hay en ejecución y que no pertenecen a dicho nivel para que finalicen, tras veinte segundos de espera los que no han terminado su ejecución son eliminados mediante la señal SIGKILL. Más concretamente son eliminados los procesos que siguen en el mismo grupo que el proceso init, si algún proceso ha cambiado su grupo no es afectado por la terminación forzosa por parte de la señal enviada a los procesos.

Un concepto importante a la hora de describir la tarea que init realiza es el de nivel de ejecución (run level). Es una separación lógica de los procesos en función de las tareas a las que están orientados, los procesos se agrupan por niveles. Para el caso que nos ocupa se pueden manejar hasta 8 niveles (numerados del 0 al 6 mas uno especial denominado con la letra ’S’ o ’s’).

El nivel 0 está reservado para detener el sistema, el nivel 6 está reservado para re-inicializarlo y el nivel 1 está destinado para el trabajo en modo mono-usuario o simple (sin la posibilidad de disponer de varias terminales virtuales), destinado a los casos en los que es necesaria alguna tarea administrativa como por ejemplo ante un error grave en el sistema de ficheros, en el arranque del sistema o un fallo en el suministro eléctrico ante los cuales hay que intentar recuperar el sistema en las mejores condiciones posible.

La letra ’S’ sirve para indicar las tareas que se pueden realizar en el nivel 1.

En el nivel 1 o cuando tenemos el sistema en modo simple no disponemos de terminales virtuales, como se ha indicado, pero seguimos disponiendo de multitarea, es decir, podemos ejecutar procesos en segundo plano (background), conviene no confundir ambas ideas.

Initdefault es la primera entrada que init busca en el fichero inittab, ya que indica el nivel del que comenzarán a ejecutarse los procesos, en caso de no existir esta entrada el sistema pedirá que se introduzca manualmente.

Daremos un repaso al fichero inittab y a los ficheros que, junto a este, configuran el arranque y el estado en el que el sistema permanecerá. Como se ha mencionado anteriormente puede ser que no necesitemos el cambio de entrada alguna pero en cualquier caso conviene conocer muy bien el contenido y significado de los ficheros para no incurrir en un error que puede dar al traste con el proceso de arranque, quizá de los más críticos en el uso del sistema.

3.4.2 El fichero INITTAB.

Ubicado en el directorio /etc este es el fichero principal en el que init basará su trabajo. Contiene información sobre el nivel del que comenzarán a ejecutarse los procesos, ficheros a ejecutar ante ciertas situaciones, número de terminales virtuales a habilitar, etc...

La sintaxis de este fichero es la siguiente:

```
id:nivel:acción:proceso
```

id Identifica la línea, deben ser dos caracteres.

nivel En que nivel/es de ejecución puede ser ejecutada la acción especificada, puede ser un valor o una secuencia (1, 12, 123 ...).

acción Acción a realizar sobre el proceso posterior, las acciones válidas son:

- **respawn** El proceso será ejecutado de nuevo cuando finalice.
- **wait** El proceso se ejecutará cuando el sistema entre en el nivel de ejecución necesario e init esperará a su finalización.
- **once** El proceso solo se ejecutará una vez cuando esté activo el nivel necesario.
- **boot** Proceso ejecutado durante el arranque del sistema. El campo 'nivel' es ignorado.
- **bootwait** Al igual que el caso anterior pero init espera a su finalización. El campo 'nivel' es ignorado.
- **off** Esta entrada no hace nada.
- **ondemand** El proceso se realizará cuando dicho nivel de ejecución sea invocado bajo demanda. En la ejecución de este proceso no habrá implícito un cambio de nivel.
- **initdefault** Especifica el nivel a fijar después del arranque del sistema.
- **sysinit** Será ejecutado durante el arranque del sistema, antes que cualquier entrada boot ó bootwait.
- **powerwait** El proceso se ejecutará al recibirse la señal SIGPWR indicando que algo anda mal en el suministro de energía eléctrica, init esperará a que finalice dicho proceso.
- **powerfail** Igual a powerwait pero init no espera su ejecución total para continuar.

powerokwait Sigue con el control del fluido eléctrico, pero a condición de que exista un fichero (/etc/powerstatus) con la palabra OK, indicando que éste ha sido restablecido satisfactoriamente.

Ctlrlaltdel El proceso será ejecutado al recibir la señal SIGINT provocada por la pulsación simultánea de las teclas <Ctrl>+<Alt>+<Del>. Esto provocará una parada del sistema y vuelta a arrancar de una forma correcta, recuerde que con LINUX no podemos apagar la máquina ni hacer uso del botón RESET sin más.

kbrequest Para procesos a ejecutarse al recibir una señal del manejador del teclado como que ha sido utilizada una combinación especial de teclas.

Como veremos en su contenido, el fichero inittab se complementa con el contenido de otros ficheros, todos ellos ubicados en el directorio /etc/rc.d, cada uno de ellos contiene instrucciones destinadas a la realización de distintas tareas dependiendo de la situación en la que se encuentra init. Esta forma de agrupar los ficheros atiende a una forma de simplificar la configuración del sistema, imaginemos por un momento todos los ficheros reunidos en un solo inittab, además que permite el ejecutar cada uno de ellos con una acción determinada, lo que le da más flexibilidad.

El fichero puesto como ejemplo no es mas que el incluido en la distribución Slackware, con los comentarios en castellano para que sea lo suficientemente autoexplicativo, no obstante se procederá a explicar el contenido de cada una de las líneas y la tarea asignada.

Se fija el nivel de ejecución inicial a cinco como primera línea, aunque init la buscaría si no estuviera en esta posición como primer valor necesario.

El primer fichero a ejecutar es el denominado rc.S que contiene los comandos necesarios para activar el área de intercambio, montar los sistemas de ficheros configurados (fichero /etc/fstab) chequeándolos si es necesario, fija la hora,... mostrando información de su actividad en todo momento.

La siguiente línea indica que el fichero rc.K debe ser ejecutado al entrar en el nivel 1, para tareas administrativas, se 'matarán' todos los programas que se están ejecutando en segundo plano (demonios) aunque los sistemas de ficheros siguen montados.

A la entrada en modo multiusuario se ejecutará el fichero rc.M que se encarga de habilitar la función de protección de pantalla que se activará a los 15 minutos de inactividad además de activar la detección de la unidad de CD-ROM (mediante rc.cdrom), activa las tareas programadas mediante la utilidad cron e inicializa todos los demonios necesarios para la gestión de redes (si está configurado), la entrada en el sistema, manejo de la impresora... además de configurar el tipo de fuente (tipo de letra) y el teclado configurados durante la instalación del LINUX.

Los nombres rc.0 y rc.6 actualmente se refieren al mismo fichero, ejecutado en la parada y re-inicialización del sistema. Su actividad principal consiste en forzar la terminación de todos los procesos activos, desmontar las unidades activas, desactivar el área de intercambio (swap) y la parada del sistema o la vuelta a una nueva carga.

Las siguientes tres líneas se encargan de mantener el sistema en un estado lo más estable posible ante un fallo en el suministro eléctrico, la primera de ellas intenta una parada ante un corte en el suministro eléctrico, la segunda interrumpe esta parada si se ha vuelto a restaurar correctamente este suministro, la tercera finalmente se asegura de volver el sistema al modo multiusuario.

Por último tenemos las líneas encargadas de configurar el número de terminales virtuales de que dispondremos en el sistema a las que podremos acceder mediante la combinación de teclas <Alt>+[F1..F6], conectadas al puerto serie o por acceso remoto vía módem.

Sobre el número de terminales virtuales configuradas, tenemos por defecto 6, pero podemos configurar los que deseemos. Únicamente hay que tenerlo en consideración a la hora de ejecutar el entorno gráfico X Window ya que éste utilizará la primera terminal virtual disponible para su ejecución (la número 7 por defecto).

3.5 Recuperación del Sistema.

Debido a cualquier problema en la configuración o de tipo físico de la máquina podemos encontrarnos con la posibilidad de no poder ejecutar LINUX correctamente.

Si el error es debido al borrado de un fichero o la modificación de su contenido sin desearlo lamentablemente no hay solución, no existe ninguna herramienta de recuperación de ficheros borrados o sobrescritos para LINUX dadas las características del sistema de ficheros que utiliza (ext2 generalmente), excepto si el fichero lo tenemos en un disquete con formato msdos en cuyo caso podemos intentar utilizar cualquier utilidad disponible.

La única forma de evitar estos problemas es tener copias de seguridad de los ficheros y directorios que creemos críticos18, además de disponer de la distribución de LINUX en cualquier forma además de tenerla instalada, un cd-rom es el formato ideal por poder contener el sistema completo aunque recuerdo que podemos tener los ficheros instalados en la partición dedicada al sistema de ficheros DOS.

Es bueno tener los programas y ficheros que añadamos a la distribución del sistema salvados, además de guardar copias de seguridad de ficheros, por ejemplo el contenido del directorio /etc incluyendo sus subdirectorios, ya que generalmente se encuentra en éste numerosos ficheros de configuración del sistema además de ficheros específicos de otros programas, el directorio /home/<usuario> y otros ficheros que consideremos necesarios. También es conveniente salvar ficheros que veremos que son importantes en la determinación que como funcionará nuestro sistema X Window en el capítulo siguiente.

LINUX nos ofrece varias herramientas que podemos utilizar para esta tarea, las principales son cpio, tar y gzip (versión GNU de compresor con formato ZIP). Tar permite el crear un fichero que contiene todos los directorios y ficheros especificados, pudiendo conservar su estructura en el árbol de directorios de forma que al restablecer desde este fichero conservaremos la ubicación de cada uno de los ficheros. El fichero creado con tar puede ser comprimido para ahorrar espacio.

Los parámetros con que podemos ejecutar tar son muchos, incluso es posible la creación de volúmenes de un tamaño determinado, pero una posibilidad de uso general puede ser:

```
# tar cvfz /root/copia.tgz /etc /home/jose
```

De esta forma el contenido de los directorios /etc y /home/jose es archivado en un fichero denominado copia.tgz.

Para devolver el contenido de este fichero a su ubicación original teclearemos:

```
# tar zxfv copia.tgz
```

Si queremos hacerlo en disquetes tendremos que teclear:

```
# tar cfvM19 /dev/fd0 /etc
```

Con la salvedad que no hace falta realizar ninguna maniobra previa sobre los mismos (no es necesario formato, sistema de ficheros ni montar la unidad), el mismo programa se encarga de asignarles el formato adecuado.

Otra utilidad destinada a esta tarea es cpio que nos permite realizar copias de seguridad, al igual que la herramienta anterior, pero con una gama más extensa de opciones, acepta ficheros creados con tar. En este caso utiliza las características de E/S del intérprete de comandos (shell). Como ejemplo copiaremos un directorio completo (sin incluir subdirectorios) en un disquete para posteriormente restaurarlo otra vez:

```
# ls /home/jose | cpio -o > /dev/fd0

# cpio -i < /dev/fd0
```

Conviene tener siempre a mano un disquete de arranque con el la última versión del núcleo de que dispongamos, esto lo podemos hacer de tres formas:

- SETUP nos lo permite durante la instalación del software.
- Compilando el núcleo del sistema operativo con la instrucción:`# make zdisk`

  Que también podemos introducir tras haber creado el núcleo de alguna otra forma.
- Utilizando un disquete y utilizando el comando:

  ```
  # cp /usr/src/linux/arch/i386/boot/zImage /dev/fd020
  ```

  Para un disco formateado podemos introducir (sin montar la unidad):

  ```
  #dd if=/usr/src/linux/arch/i386/boot/zImage of=/dev/fd0
  bs=8192
  ```

De esta forma tenemos un disco a nuestra disposición para el caso en que tengamos un problema con el arranque desde el disco duro o deseemos utilizar el disquete como método de inicialización en todo momento, además hay que añadir el siguiente comando:

```
# rdev -R /dev/fd0 1
```

Que indica al núcleo que el arranque se va a realizar desde este dispositivo.

Si tenemos algún problema con el sistema de ficheros el programa a utilizar es e2fsck, que chequea el contenido del sistema de ficheros ayudándonos a recuperar ciertos errores, para el caso del sistema de ficheros ext2. Uno de los más comunes es un error en el superbloque, parte del sistema de ficheros que contiene información vital sobre éste, si está dañado la forma de repararlo es:

```
# e2fsck -f -b 8193 /dev/hda
```

Otra utilidad para chequear el estado de los sistemas de ficheros instalados es fsck. Hasta el momento puede que hayamos tenido noticia de la misma sólo en el proceso de arranque del sistema pero conviene conocer algunas nociones de su funcionamiento.

Lo primero a destacar es que el sistema a chequear no ha de estar montado porque podemos encontrarnos con el chequeo y posible reparación de ficheros en uso lo cual no es aconsejable. Es por esto que la revisión de la partición principal se hace durante el arranque del sistema.

La forma mas sencilla de invocación es, por ejemplo, 'fsck /dev/hda2' pero en este caso no hemos indicado el tipo de sistema de ficheros utilizados ni que hacer cuando se encuentra con un error. Algunos de los parámetros mas importantes son:

-A Realiza un testeo de todos los sistemas de ficheros descritos en el fichero /etc/fstab.

-V Muestra información mientras está realizando cualquier proceso.

-t ¡sistema$_{f}icheros > Tipodesistemadeficherosacomprobar.$

-a Repara automáticamente cualquier problema encontrado.

-r Pide confirmación antes de realizar cualquier reparación.

-s Fuerza el chequeo en serie de los sistemas de ficheros a procesar, la opción por defecto es el poder testearlos en paralelo si se encuentran en dispositivos diferentes.

Este programa también es ejecutado cada vez que el sistema arranca y son montadas las diferentes unidades especificadas en el fichero /etc/fstab ejecutándose de ser necesario, en este caso cuando se ha alcanzado un numero de veces determinado por un contador.

Otra parte importante de los discos duros es su sector de arranque y tabla de particiones, información que podemos grabar en un fichero con el comando:

```
# dd if=/dev/hda of=/root/arranque bs=1K count=1
```

En el fichero arranque tenemos toda la información necesaria, con el primer parámetro indicamos el origen de la información y con el segundo el destino, el parámetro bs indican el tamaño del bloque para el fichero (en este caso 1 K).

Si es necesario restaurar el sector de arranque y la tabla de particiones desde el fichero teclear21:

```
# dd if=/root/arranque of=/dev/hda bs=1K count=1
```

Pero podemos encontrarnos en el caso en que el problema es más grave, en cuyo caso necesitaremos algún tipo de herramienta para salvarnos de ’errores fatales’.

Existen para LINUX paquetes disponibles con software destinado a la creación de discos de arranque además de otros con numerosas utilidades que nos pueden ser de gran ayuda para esta labor.

Básicamente consiste en crear un disco de arranque (boot) y uno con las utilidades mínimas para poder recuperar el sistema, si es posible (root). La posibilidad de tener ambos grupos en un mismo disco (boot/root) también existe pero de esta forma limitamos mucho el número de programas que podemos introducir en el disquete.

3.6 Redes y Comunicaciones.

Si necesitamos que una máquina o grupo de máquinas sea capaz de relacionarse LINUX es la solución. Como implementación del sistema operativo UNIX está concebido para dar soporte de redes mediante la posibilidad de configurarlo para el manejo de diferentes tipos de protocolos y dispositivos. Una vez habilitado este soporte disponemos de una amplia oferta de software con el que podremos extraer el máximo rendimiento al sistema que vayamos a montar.

En este apartado sólo son tratados los pasos necesarios para que el sistema operativo pueda manejar distintos protocolos y dispositivos así como los principales ficheros implicados en el tema de las redes y comunicaciones. Se describirán algunas utilidades importantes que ayudarán a la gestión del entorno en estas condiciones. El protocolo PPP está mas extensamente explicado en el apartado siguiente debido a su uso para la conexión a Internet a través de Infovía, también se reserva para este apartado la explicación del programa named que nos permitirá la resolución de nombres en la máquina local mientras estamos conectados a la red. La posible repetición de algunos conceptos en los apartados siete y ocho responde a la intención de mantener los contenidos lo más separados posible de forma que se puede acudir al apartado ocho directamente si lo único que se necesita es el acceso a Internet.

El primer software para redes concebido por y para LINUX se denominó Net-1 (creado por Ross Biro) y apareció durante el otoño del año 1.992. Posteriormente otro autor (Fred van Kempen) se puso a trabajar en la mejora de este código y creó el denominado Net-2 que empezó a ser mantenido por un grupo más extenso de colaboradores destacando entre ellos Alan Cox. Tras una depuración profunda del código Net-2 sobrevino una versión denominada Net-3.

A partir de la versión 1.1.5 de LINUX este código forma parte íntegra del código del sistema operativo por lo que no necesitamos el obtener código adicional alguno si queremos adentrarnos en este terreno.

Actualmente el código Net-3 (lo que es igual a decir LINUX) soporta:

- La mayoría de las tarjetas de tipo Ethernet existentes.
- Protocolos de transmisión vía el puerto serie SLIP (Serial Line IP) y PPP (Point to Point Protocol).
- Compresión de encabezamiento de paquetes con el algoritmo Van Jacobson.
- PLIP para la conexión de ordenadores vía el puerto paralelo.
- EQL Load Balancing que permite el uso de más de una conexión simultánea con otra máquina, multiplicándose el ancho de banda disponible.

- NFS: los sistemas de ficheros remotos son accesibles vía este protocolo posibilitando el uso transparente para el usuario.
- IPX: protocolo utilizado por las redes de tipo Novel.
- NIS: (Network Information System) protocolo creado por la empresa SUN destinado a poner cierta información vital a todas las máquinas de una red lo cual tiene como objetivo el hacer más sencillo el manejo de la misma.
- ARCnet.
- Redes de tipo Token Ring (IBM).
- Appletalk: compartiremos ficheros y dispositivos con máquinas Machintosh.
- Redes sin cable: generalmente vía dispositivos infrarrojos.
- ISDN (ó RDSI 'Red Digital de Servicios Integrados').
- ATM: Protocolo de comunicaciones.
- IP firewalling: Para configurar la máquina LINUX como cortafuegos de una red.
- IP Accounting: Contabilidad del sistema.
- IP tunnelling: Permite el encapsulamiento de un protocolo dentro de otro ('disfrazar' un protocolo para que tenga apariencia de otro). También es utilizado para el caso de IP 'móviles', una máquina tiene una dirección IP fija pero no siempre está ubicada en un mismo lugar.

3.6.1 Configuración del núcleo

Cualquier intento de que LINUX pueda comunicarse22 ha de pasar necesariamente por su compilación23 durante la cual se nos presentan una serie de cuestiones a responder de las cuales una parte está dedicada a conocer el tipo de protocolos y manejadores de dispositivos necesarios.

La pregunta mostrada en la figura es la primera a responder afirmativamente si necesitamos que se nos muestren mas preguntas al respecto, de no hacerlo podemos encontrarnos con que no se nos presentan el resto de opciones y tengamos que abortar este proceso para comenzar de nuevo.

Una vez contestadas estas cuestiones pasaremos a describir el tipo de dispositivos que LINUX va a manejar. Sólo describiré aquellas cuestiones que puedan plantear cierta dificultad, el resto son bastante autoexplicativas.

El dispositivo dummy permite asignar una dirección IP 'falsa' a una máquina sin acceso a una red (no tiene asignada dirección IP 'legal' o le es asignada de modo dinámico), de forma que podemos enviar paquetes a esta dirección en vez de utilizar la dirección 127.0.0.1 (dirección por defecto de la máquina local). Esto puede ser útil para probar cierto tipo de software para redes y puede compararse con el dispositivo /dev/null. Una vez compilado el núcleo debemos hacer lo siguiente (también lo podemos introducir en el fichero /etc/rc.d/rc.inet1):

```
# ifconfig dummy 192.168.34.5624
# route add -net 192.168.34.0 device dummy
```

Ya podemos introducir en nuestro fichero /etc/hosts la siguiente información:

192.168.34.56 delfin.micasa.es delfin

Algunos autores recomiendan esta forma de proceder en vez de asignar el nombre de la máquina a la dirección 127.0.0.1, esta dirección se debe dejar como localhost.

Podemos utilizar el comando ifconfig para comprobar que todo ha ido bien:

El resto de preguntas se refiere a un listado de modelos de dispositivos que LINUX puede controlar, contestaremos afirmativamente si encontramos el que se corresponde con el instalado.

3.6.2 Ficheros implicados

Los ficheros que se van a detallar se pueden considerar como los mínimos necesarios a existir en el caso que necesitemos trabajar en un entorno de red. Todos ellos contienen una configuración por defecto que es la que la distribución instalará y en ocasiones no hay que cambiar el contenido de algunos de ellos.

Aquí se detalla su función y su contenido 'por defecto', hemos de tenerlos muy en cuenta a la hora de instalar el soporte de cualquier tipo de red o software dedicado a dar servicios a la misma. Todos ellos se encuentran en el directorio /etc.

HOSTNAME Destinado a contener el nombre de nuestra máquina.

hosts Consultado en primera instancia para la resolución de nombres en un entorno de red (obtener la dirección IP de una máquina a partir de su nombre completo25 o su alias). Debe estar presente como mínimo el par '127.0.0.1 localhost'. Podemos tener las direcciones más utilizadas por el usuario de forma que nos ahorramos el tener que acceder al protocolo de resolución de nombres para cada dirección necesitada.

hosts.conf Indica a LINUX la forma en que ha de proceder a la hora de resolver nombres de máquinas, la configuración más común es:

La orden a seguir como indica el fichero será buscar primero en el fichero hosts para acceder al protocolo bind26 en caso de no obtener resultados de la primera forma. La última línea permite tener más de una dirección IP para un misma máquina.

resolv.conf Contiene las direcciones IP de los servidores de nombres accesibles desde la red. También permite el indicar el dominio de la maquina en caso de tener uno asignado.

inetd.conf Inetd permite el tener una máquina a la espera de peticiones de conexión, puede considerarse como un gestor de todos los servidores que pueden haber instalados con la facilidad de que no es necesario que estén activos en todo momento. A la llegada de cualquier requerimiento de un servicio el programa iniciará el servidor adecuado.

El fichero de configuración contiene una lista de nombres de los servicios y el tipo de socket (mecanismo de comunicación entre procesos que puede utilizarse en un entorno de red o en una máquina aislada) de los cuales obtendrá con la ayuda del fichero /etc/services su número y protocolo a utilizar. De esta forma iniciará el programa conveniente.

services Listado del nombre, número y protocolo asignado a cada socket.

protocol Contiene todos los protocolos que podemos utilizar con LINUX y el número asignado a cada uno de ellos.

Los ficheros encargados de inicializar todos el software implicado en la gestión de redes así como contener todas las variables implícitas son rc.inet1 y rc.inet2. El primero de ellos contiene los primeros pasos de este proceso, asignando las direcciones IP de la máquina, las de red y difusión en la misma y la de la pasarela de existir. Seguidamente inicia los dispositivos instalados y carga las tablas con las rutas necesarias. El segundo fichero contiene unas líneas para cada uno de los programas que podemos ejecutar para gestionar la red.

En cuanto al software disponible para explotar las capacidades de red con LINUX es muy amplio, con la distribución disponemos de los programas básicos en cualquier sistema UNIX: ftp, telnet, ping, traceroute.... pero existen en múltiples lugares de la red y bajo distintos tipos de licencia numerosas utilidades que pueden hacer de este sistema operativo una alternativa seria en este aspecto. Sirva como ejemplo que existen numerosas empresas proveedoras de Internet que sólo utilizan LINUX (en España existen algunos que lo utilizan de momento para parte del trabajo).

3.7 Acceso a Internet con LINUX.

El nacimiento de la red Infovía ha llevado consigo la proliferación de numerosas empresas que, aprovechando las posibilidades de esta red, provee de acceso a Internet a cualquier usuario con un modem y el software necesario, a cambio de una cuota por el servicio además del coste de la llamada telefónica como urbana sea cual sea el punto desde el que llamemos.

Dependiendo de las características de dicha empresa proveedora (proveedor o ISP27 a partir de ahora) en cuanto a tipo de conexión y cuotas los servicios que se nos ofrecen pueden ser muy variados aunque en general todas ofrecen las siguientes posibilidades por una conexion que podemos denominar como normal:

- Dirección de correo y tamaño de buzón ilimitado.
- Posibilidad de albergar páginas web (hypertexto) propias con un tamaño máximo especificado.
- Servidores propios de correo, ficheros, IRC (para charla en la red), news (noticias en Usenet), etc.

Otras posibilidades que se suelen ofrecer son el poder disponer de otra dirección de correo, direcciones IP fijas, mayor tamaño en la página web, teléfono de contacto de coste reducido,....

Para hacer uso de todos los servicios que la red nos permite podemos utilizar LINUX ya que disponemos de todo lo necesario, incluso distintos programas para realizar la misma tarea.

En este apartado se desglosa paso a paso la configuración de LINUX para acceder a este tipo de proveedores así como la forma de poder manejar el correo y las news. Existen numerosas alternativas a los pasos aquí descritos en cuanto a ficheros de configuración y programas a utilizar, se he optado por reflejar la más común entre los usuarios de LINUX e Internet y centrada en el software disponible en la distribución Slackware.

Para cumplir nuestro objetivo debemos conocer previamente la siguiente información, que normalmente nos será dada al contratar un acceso con un proveedor de Internet cualquiera (reflejo entre paréntesis los valores que voy a utilizar a lo largo de todo el ejemplo).

- Denominación de nuestra cuenta de usuario que generalmente pasará a formar parte de nuestra dirección de correo. (jmlm(ibañez))
- Palabra clave de paso a utilizar. (mortadelo)
- Dirección del Servidor de Nombres a utilizar (DNS). (194.224.215.52)
- Nombre de la máquina o máquinas que se encargarán de la gestión del correo electrónico y de las áreas de noticias. (bacterio.ibañez.es y super.ibañez.es respectivamente).

Como se puede observar en momento alguno se hace referencia a una dirección IP propia, ya que generalmente se asigna una dirección dinámica a cada usuario en cada una de sus conexiones, lo cual implica que nuestra máquina no está permanentemente en Internet.

3.8 El protocolo PPP.

LINUX permite la utilización de dos protocolos de comunicación vía el puerto serie SLIP y PPP, para el tipo de conexión que necesitamos utilizaremos el segundo en su uso como software cliente aunque podemos utilizar ambos protocolos para habilitar servidores.

PPP no es exactamente un protocolo sino un conjunto de protocolos, cada uno de los cuales se encarga de la gestión de la conexión a un nivel determinado y está especificado en el RFC 1584.

El RFC28 es un tipo de documento que trata de especificar detalles técnicos sobre distintos aspectos del funcionamiento de Internet, pueden ser divididos en cinco grupos: requeridos, sugeridos, direccionales, información y obsoletos. La información clasificada como requerida debe estar implementada en cualquier máquina conectada a la red. Existen otro tipo de documentos redactados en un tono más divulgativo que se clasifican en la categoría denominada FYI de 'For Your Information' (Para su Información).

Todos estos documentos pueden ser obtenidos vía ftp de las direcciones nic.ddn.mil y ftp.uu.net o bien vía e-mail dirigiendo un mensaje a la dirección service(nic.ddn.mil con el la línea 'send RFC-xxxx.TXT' en el campo 'Subject:' del mensaje.

Los protocolos que configuran el PPP son el HDLC (High-level Data Link Control Protocol), en el nivel más bajo, define el aspecto que van a tener las tramas y provee de control de errores mediante un campo de control de 16 bits29. Las tramas PPP pueden contener paquetes de otros protocolos.

LCP (Link Control Protocol) por encima del anterior negocia el tamaño máximo de las unidades a transmitir (MRU) y el control asíncrono del mapa de carácteres (async map) que ayuda a identificar carácteres de control contenidos en los paquetes durante la comunicación, descrito en el RFC 1548.

IPCP (Internet Protocol Control Protocol) es el utilizado para la negociación de las direcciones IP de ambos extremos.

PPP permite dos formas de autentificación, el CHAP (Challenge Handsake Authentication Protocol) y el PAP (Password Authenticacion Potocol).

El PAP es el protocolo más extendido y el más sencillo (también el menos seguro) y básicamente se trata del intercambio del usuario y la palabra clave de acceso, similar al acceso normal a LINUX en un terminal.

CHAP es un protocolo mucho más completo y requiere la colaboración constante del cliente y el servidor. El mecanismo de funcionamiento consiste en que el servidor envía una tira al cliente junto con su nombre, el cliente usa el nombre del servidor para buscar la clave apropiada en su fichero de claves, la combina con la tira recibida, la encripta y la devuelve al emisor junto con el nombre del cliente. Esto no tiene porque realizarse en el momento de establecer la conexión, incluso CHAP permite el realizar este proceso a intervalos para asegurar que el cliente no es un intruso.

CHAT ayuda a PPP con un lenguaje propio destinado a tareas como el marcado y acceso al sistema remoto

El software que conforma el protocolo PPP en LINUX está escrito por Michael Callahan en su parte incluida en el núcleo (el nivel HDLC) y Al Longyear el programa pppd, demonio que va a realizar todas las tareas de manejo de la conexión. CHAT esta escrito por Karl Fox.

Primeramente se ha de compilar el núcleo para habilitar el soporte de los manejadores necesarios para la gestión de redes y el protocolo a utilizar, tendremos que responder afirmativamente a las siguientes cuestiones:

Si arrancamos con el nuevo núcleo podremos comprobar que todo ha ido bien comprobando que en la pantalla aparece el siguiente mensaje (lo podemos comprobar en el fichero /var/adm/messages):

De tener configurado el soporte para el directorio /proc podemos acudir al fichero /proc/net/dev en el que para nuestro ejemplo tiene el siguiente aspecto:

Para conocer si está instalado el software del programa pppd instalado (así como el CHAT y ficheros y utilidades de configuración) se puede acudir al directorio /var/adm/packages y comprobar si están creados los ficheros ppp y tcpip.

En la configuración del software tendremos que utilizar nuestra cuenta de administrador lo cual implica que hemos de tener cuidado en el transcurso de este proceso. Para la posterior utilización del mismo se podrá utilizar cualquier otro usuario.

A la hora de configurar el programa pppd puede optarse por un sencillo fichero ejecutable en el que tendremos la línea a utilizar al invocar el programa o podemos hacer uso de los ficheros a los que el demonio accede antes de interpretar la línea de comandos, estos son:

- /etc/ppp/options: Parámetros generales que van a ser interpretados antes que los introducidos en la línea de comandos por todas las conexiones establecidas. Puede definirse uno para cada línea disponible (options.ttyS1, options.ttyS2...) o uno por lugar a donde conectarnos (options.ISP, options.EUI, ...). Con el parámetro file indicamos a pppd que fichero utilizar para opciones.

- /.ppprc30: Opciones por defecto para un usuario determinado.

- /etc/ppp/ip-up: Fichero ejecutado cuando la conexión esta establecida, normalmente puede estar vacío pero lo podemos utilizar para, por ejemplo, incluir los comandos necesarios para enviar y recibir el correo automáticamente.

- /etc/ppp/ip-down: Ejecutado cuando la conexión ya se ha cortado, puede ser utilizado para restablecer el estado previo de la máquina a haber sido ejecutado el fichero anterior.

- /etc/ppp/pap-secrets: Contiene los datos necesarios para poder identificarnos utilizando el protocolo PAP. Contiene cuatro columnas con la denominación del cliente (nuestro nombre de usuario en este caso), identificador del servidor, la clave a utilizar y por último la dirección/es IP que el servidor puede para mantener la conexión con nosotros, en nuestro caso no haremos uso de esta entrada. Ejemplo:

  ```
  jmlm@ibañez bacterio mortadelo
  ```

  Otra forma de definir el nombre del cliente y la clave a utilizar consiste en definir un fichero e invocarlo con el parámetro '+ua' en la línea de comandos, el fichero se podría llamar /etc/claves y tendría esta forma:

  En la invocación del programa pppd se introduciría:

  ```
  # pppd ...  ...  +ua /etc/claves
  ```

  En nuestro caso como identificador del servidor (segunda columna del fichero pap-sectrets) podemos poner cualquier nombre, debido a que únicamente utilizamos esta entrada para conocer la clave a utilizar, combinándola con el uso del parámetro remotename en el pppd. También podemos colocar un asterisco ('*') si la única conexión que realizamos vía ppp es con el proveedor de internet, de esta forma para cualquier conexión establecida para el usuario utilizará la misma clave.

- /etc/ppp/chap-secrets: Su formato es igual al caso anterior, lo tendremos que crear en caso de que nuestro proveedor (máquina a la que queramos conectarnos) utilice este protocolo de autentificación.

  Otros ficheros implicados en el acceso a Internet y que han estar creados (los anteriores son todos optativos excepto el /etc/options que es obligado aunque se encuentre vacío):

- /etc/hosts: Debe contener como mínimo la dirección IP 127.0.0.1 que se considera una dirección local (para máquinas aisladas) con la que podemos probar programas que necesitan de una dirección IP para funcionar sin tener necesariamente que estar conectados. El fichero puede tener la siguiente forma:

  El formato del fichero consiste en tres valores; dirección IP, nombre completo de la máquina y nombre de pila (o alias). Podemos utilizas este fichero para almacenar las direcciones a las que solemos acceder evitándonos el tiempo que supone que cada invocación de una dirección tenga que ser procesada por el servidor de nombres.

- /etc/resolv.conf: Pondremos la dirección del servidor de nombres que el proveedor nos ha facilitado, permite colocar hasta tres valores en este fichero para distintos servidores de nombres que serán utilizados en el orden en que aparecen, el algoritmo utilizado comienza por consultar en el primer servidor, caso de no obtener la respuesta esperada o un tiempo de espera excesivo (timeout) se pasa a utilizar el siguiente. En nuestro caso tendrá este aspecto:

  De esta forma durante nuestro acceso a Internet cualquier llamada utilizando el nombre de una máquina será en primer lugar buscado en el fichero /etc/hosts a fin de obtener su dirección IP y después utilizando el protocolo especificado. Para asegurarnos este orden en la resolución de nombres hemos de tener el fichero /etc/hosts.conf con las siguientes líneas:

  La primera línea indica el orden en que han de ser utilizados los recursos disponibles en la resolución de nombres, la segunda permite la existencia de más de una dirección IP para una máquina en el fichero hosts.

Una vez introducidos todos los valores correspondientes en los ficheros indicados podemos dar paso a la invocación del programa pppd. Para el ejemplo introduciremos todos los parámetros en el fichero /etc/options y crearemos un fichero ejecutable con el nombre /usr/local/bin/conectar.

El fichero options tendrá el siguiente contenido:

La última línea no hará falta incluirla si hemos puesto el símbolo del asterisco como identificador del lugar al que queremos conectar en el fichero pap-secrets.

Los parámetros indican al programa pppd que:

- crtscts: Habilita el control del flujo por hardware utilizando las líneas RTS/CTS del modem.
- modem: Utilizar las líneas de control del módem.
- defaultroute: Añadir esta ruta a las tablas de enrutado del sistema.
- noipdefault : Indica que el extremo de la conexión ha de proveernos de una dirección IP, no hay que obtenerla a partir de la información contenida en el sistema.
- lock: Bloquea el acceso al puerto serie a otros programas.
- user: Identifica al usuario a efectos del protocolo de identificación PAP.

En el fichero conectar se invocará al programa pppd con el programa chat a fin de poder marcar el número de teléfono, puerto a utilizar así como la velocidad en baudios a utilizar para la conexión.

Una vez creado este fichero le daremos los permisos necesarios a fin que sea ejecutable:

```
# chmod 755 conectar
```

El único comentario a hacer al fichero descrito es que el programa CHAT se basa en una secuencia de cadenas recibidas/enviadas al módem, a base de comandos pertenecientes al estándar HAYES, por lo que si nuestro módem no responde correctamente o no es compatible con el estándar hemos de cambiar esta secuencia e introducir los parámetros adecuados.

El fichero /dev/módem no es más que un enlace al fichero que identifica al puerto en el que tenemos instalado el módem (/dev/cua0 para COM1, /dev/cua1 para COM2...) por lo que podemos poner directamente la denominación del puerto si lo deseamos.

Para verificar si nuestra conexión se establece correctamente y que ocurre durante este proceso introduciremos la siguiente línea en el fichero /etc/syslog.conf31:

```
local2.* /usr/adm/ppp.log
```

De esta forma tendremos todos los mensajes generados durante el establecimiento de la conexión en el fichero definido. Es posible aumentar el detalle los mismos (de cara a una posible depuración de errores) introduciendo en el fichero options el parámetro debug o el kdebug seguido del un valor que indica dicho nivel de detalle, este parámetro puede ser 1, 2 y 4 o la suma entre ellos (3, 5 y 6).

Si deseamos observar el desarrollo del proceso de conexión podemos teclear 'tail -f /usr/adm/ppp.log' en un terminal virtual una vez hemos introducido el comando conectar.

El contenido del fichero variará en función de los parámetros con los que hemos invocado tanto a pppd como a chat y si hemos introducido alguna opción para la depuración (debug) de la conexión.

Como nota final mencionaré que si lo que deseamos es simplemente el acceder a la red Infovía el mecanismo es idéntico al descrito, los únicos puntos que cambian son la dirección IP del servidor de nombres de Infovía (fichero /etc/resolv.conf) que en este caso es 10.0.0.1 y tanto el nombre de usuario como la clave serán la palabra 'Infovía'.

3.9 Correo Electrónico, 'News' y 'WWW'.

Una vez habilitada la posibilidad de conectarnos a Internet mediante la empresa proveedora escogida es necesario software que nos permita el explotar todas las posibilidades que la red nos ofrece, LINUX dispone de una amplia oferta que va desde gestores de correo típicos de entornos UNIX en modo texto como pine o elm hasta las ultimas versiones de navegadores como Mosaic y Netscape, desarrollados para trabajar bajo X-Window.

Si optamos por trabajar con Netscape la última versión es la 3.0 y podemos encontrarla en la dirección ftp.netscape.com en el directorio /pub/unsupported/gnu.

Con este programa la configuración es muy sencilla, simplemente tendremos que indicarle el nombre de las máquinas que van a servirnos el correo (SMTP Server) y las news, el protocolo para obtener el correo del servidor (POP3) y otros datos de interés como nuestro nombre, dirección de correo, organización, dirección IP y puerto para el servidor 'proxy' (servicio que nos ofrece una caché con el contenido de todas las páginas web que hemos visitado, de forma que al volver a visitarla no es necesario volverla a obtener de nuevo lo podemos hacer desde este servidor ahorrando tiempo), etc... de forma que con un sólo programa tendremos todas nuestras necesidades cubiertas. Para una configuración mas detallada, a nivel de recursos, podemos extraer todo lo necesario del fichero /usr/local/lib/netscape30/Netscape.ad y modificarlo a nuestro gusto ubicándolo, por ejemplo, en el fichero .Xdefaults de nuestro directorio. Para la ordenación del correo en las diferentes carpetas que podemos tener abiertas haremos uso del fichero ./netscape/mailsort que tiene el formato:

Mosaic para LINUX se puede obtener en la dirección ftp.ncsa.uiuc.edu, en este caso sólo disponemos del navegador.

Existe otro navegador igualmente famoso entre los usuarios de este sistema operativo que se llama Lynx, se trata de un navegador sin capacidad para la visualización de gráficos (podríamos calificarlo como 'modo texto'), sonido y otros formatos de información. A pesar de su sencillez es muy útil para el caso en que queremos obtener de las páginas web sólo el texto de las mismas. Más información en www.wfbr.edu/dir/lynx o si se desea el programa también se puede obtener en sunsite.unc.edu/system/N
systems/www.

Si lo que necesitamos es un programa para la gestión de correo sin más existen multitud de opciones para LINUX, con la distribución tenemos por ejemplo pine, elm y mailx que trabajan en modo texto y xmh para X-Windows aunque debido al tipo de acceso del que disponemos necesitaremos un Agente de Transporte para este correo, es decir, un programa con el que poder enviar y recibir el mismo del servidor si queremos poder manejarlo en nuestra máquina local, Slackware nos ofrece dos posibilidades; sendmail y popclient.

Si utilizamos Netscape u otros programas para la gestión del correo quizá no necesitemos tener este tipo de software instalado ni configurado, pero para programas mas sencillos es necesario su uso si no queremos el estar conectados continuamente para tareas de este tipo.

Para el caso que nos ocupa se describen los pasos necesarios para configurar el programa sendmail para poder enviar el correo a nuestro proveedor y el popclient para poder obtenerlo del mismo y almacenarlo en un fichero para su posterior gestión.

Hemos de tener instalados los paquetes correspondientes al procesador de correo (pine.tgz) y los programas adicionales (sendmail.tgz, smailcfg.tgz).

Primeramente configuraremos el programa Pine con los parámetros que necesitemos introducir tales como nuestro nombre, dominio, ubicación de los ficheros que van a contener el correo, nombre de la máquina que nos provee de las news (super.ibañez.es), etc.., en cualquier momento el programa nos puede aportar ayuda sobre cualquier punto, lo único que hay que tener en cuenta es dejar en blanco la entrada en la que se nos pregunta por la máquina que gestiona el correo. Toda la configuración introducida se almacenará en el fichero .pinerc.

Con esto tenemos a Pine preparado para el manejo del correo ubicado en el directorio /mail del usuario.

Para la gestión de las news el mecanismo consiste en definir las áreas que queremos obtener, para una vez conectados poder obtener de las mismas el encabezado de los artículos y poder marcar los deseados, que serán grabados en nuestro disco duro, mecanismo muy parecido al utilizado por Netscape.

Sendmail es un potente programa diseñado para la gestión de correo en múltiples situaciones de ahí su complejidad a la hora de configurarlo, utilizaremos uno de los ficheros que se nos ofrecen como ayuda (contenidos en el directorio /usr/src/sendmail/cf/cf/) debido a que se ajusta plenamente nuestro objetivo y no tenemos que enfrentarnos con la tarea de hacerlo directamente sobre su fichero de configuración33. Para el procesamiento del fichero creado necesitamos tener instalada la utilidad 'm4'.

Primero desactivaremos cualquier referencia al programa sendmail en los ficheros de inicialización del sistema (contenidos en el directorio /etc/rc.d) que generalmente inicializan el programa para activarse en segundo plano (demonio) y procesar el correo a intervalos determinados, opción que no nos es útil para nuestro objetivo.

El fichero a utilizar se trata del clientproto.mc destinado simplemente al envío del correo a la máquina remota. Se hará una copia de este fichero e introduciremos las modificaciones pertinentes, suponiendo que llamamos a este fichero bacterio.mc lo dejaremos de la siguiente forma:

Para obtener el fichero que nos interesa (/etc/sendmail.cf) teclearemos:

```
# m4 bacterio.mc > /etc/sendmail.cf
```

De esta forma ya tenemos configurado el programa para que almacene temporalmente nuestro correo, de forma que podamos enviarlo cuando estemos conectados a nuestro proveedor.

La forma de utilizar esta posibilidad es muy sencilla, simplemente cuando termine de escribir el mensaje deseado (opción COMPOSE en pine) lo enviará con la combinación de teclas <Ctrl>+X. De esta forma tendremos en el directorio /var/spool/mqueue una serie de ficheros con el contenido de nuestros mensajes mas otros ficheros de control.

Una vez establecida la conexión con el proveedor será necesario teclear 'sendmail -q' si para procesar todo el correo almacenado y sea enviado a la dirección especificada (o incluir esta orden en el fichero /etc/ppp/ip-up).

El estado de la cola de mensajes se puede comprobar en cualquier momento, el comando 'mailq' nos mostrará información más detallada de la misma.

Utilizaremos el software 'popclient' a fin de obtener el correo de nuestro buzón en la máquina remota, en este caso los parámetros son:

```
# popclient -3 -v -u jmlm -o /home/jmlm/mail/inbox -p mortadelo bacterio.ibañez.es
```

Con esta línea se especifica la versión del cliente a utilizar (3), mostrar información mientras obtiene el correo (-v), el identificador como usuario, fichero en donde queremos almacenar el correo (a tener en cuenta a la hora de configurar el gestor de correo escogido) y el nombre de la máquina remota. Un parámetro útil en estos casos es '-k' que mantiene los mensajes en el servidor una vez recuperados en vez de borrarlos, destinado a poder hacer las pruebas necesarias.

Como software lector de news tenemos a nuestra disposición programas como tin, trn, nn, tas ... y como software para el transporte de las mismas a los programas cnews e inn, pero en este caso ambos programas están concebidos para el proceso de noticias como servidor o cliente en entornos de red por lo que para configurarlos y poder hacer algo similar al caso del correo no es tarea sencilla y en ocasiones podemos necesitar de programas que no vienen con la distribución, por lo que no voy a entrar en detalles en este caso, solo mencionar la existencia de programas (suck, leafnode ...) que están comenzando a simplificar este apartado para conexiones a Internet vía IP dinámica (nuestro caso para el ejemplo), por lo que el mecanismo consistirá en ir obteniendo aquellos artículos que han aparecido en el foro o foros escogidos desde nuestra conexion y

recuperarlos a fin de poder manejarlos una vez desconectados así como el envío de los artículos propuestos por nosotros.

3.10 Bind.

Hasta el momento se ha visto que es necesario un mecanismo que resuelva el problema de obtener la dirección IP a partir de un alias o su nombre completo debido a que lo que identifica a una máquina en Internet es dicho valor y es necesario conocerlo a fin de ejecutar cualquier intercambio de información con la misma.

Se han presentado dos aproximaciones; tener un fichero /etc/hosts con varias ternas dirección, nombre y alias, lo cual es un mecanismo muy limitado si nuestra intención es la de tener todas las direcciones IP a las que nos podemos conectar, o hacer uso de un servidor que realiza esta tarea por nosotros.

Con el tipo de conexión que se ha descrito esto tiene el problema de, además de manejar el tráfico normal debido a la tarea que estemos realizando, en caso de acceder a múltiples máquinas diferentes hay que resolver la dirección para cada una de ellas para cada conexión, por lo que tenemos cierta sobrecarga implícita debido a que todas estas resoluciones también son enviadas por el enlace. Hay que entender que este problema quizá no sea importante si no accedemos durante mucho tiempo a Internet, las de actividades que realizamos, servidor al que estamos conectado, etc ... depende de nosotros el valorar su impacto real sobre el rendimiento de nuestro acceso.

Con el ejemplo que voy a describir podremos habilitar un registro de todas las direcciones IP que hemos ido obteniendo del servidor remoto en nuestra máquina local, de forma que toda búsqueda de una dirección posterior y que ya tengamos almacenada se resolverá más rápido. Esto tiene una limitación a tener en cuenta, una vez apagamos la máquina o desactivamos el programa perdemos toda la información almacenada, la única solución posible es no apagar nunca el ordenador ni matar el proceso que realiza esta tarea.

BIND (Berkley Internet Name Domain Service) es el la denominación del protocolo implementado mediante el programa named. Slackware denomina con el nombre del protocolo al paquete que contiene el programa y todos los ficheros de configuración y documentación adicionales.

Crearemos los ficheros de configuración que el programa necesita para trabajar y modificaremos alguno de los creados para nuestra conexión al proveedor de Internet. Para el ejemplo configuraremos el software a fin de ejecutarse como cliente simplemente, almacenando todas las direcciones IP e información adicional necesaria para la resolución de nombres en la máquina local.

El primer fichero que necesitamos crear es el fichero /etc/named.boot en el que especificaremos el directorio de trabajo a utilizar, la forma en la que se va a trabajar, el dominio y el fichero que contiene la definición de los servidores de nombres a utilizar.

El parámetro ’cache’ es el que especifica a named la forma en que se va a trabajar habilitando un almacén local, como dominio especificamos el punto (.) que simboliza el dominio principal o el definido para la máquina (generalmente en variables de entorno o en el fichero /etc/hosts), el fichero /var/named/named.root34 es el que contiene información sobre los servidores de nombres a consultar.

El contenido del fichero named.root indicará al programa direcciones IP referidas a servidores de nombres que pueden ser consultados por nosotros durante el trabajo en la red.

En la primera entrada de las primeras líneas del fichero hemos definido el dominio en el que se va a trabajar, que debe coincidir con el definido en la entrada cache del fichero de configuración anterior y en este caso se especifica con el signo del punto, el valor que sigue no es mas que los segundos que el programa debe mantener la información obtenida del servidor antes de ser desechada, por lo que hemos definido el valor 99999999 como valor simbólico, en realidad para que nunca se elimine dato alguno36, por último indicamos con 'NS' que vamos a especificar un nombre para una máquina que en nuestro caso son nombres sin sentido alguno pues lo que realmente interesa serán sus direcciones IP definidas posteriormente.

En este fichero podemos ubicar tantas entradas como necesitemos, incluso podemos indicar solamente la primera de ellas que se refiere al servidor de nombres de nuestro proveedor de acceso a Internet, aunque de esta forma no explotamos todas las posibilidades que se nos ofrecen. Tampoco es necesario seguir el orden seguido, para más claridad si se desea se puede seguir este otro, siempre y cuando respetemos el orden de los pares de entradas para definir un servidor.

El único fichero que necesitamos modificar de cara a disponer de este nuevo servicio es el /etc/resolv.conf que pasará a tener el siguiente contenido:

De esta forma especificamos que el servidor de nombres va a ser nuestra máquina en primer lugar, para pasar a utilizar los especificados en el fichero named.root, esquema definido en el fichero /etc/host.conf.

Lo único que resta por hacer es arrancar el programa invocándolo como administrador. Si va a ser este el mecanismo utilizado por defecto para nuestra conexión a internet conviene cargar named durante el arranque de la máquina, opción que se especifica en el fichero /etc/rc.d/rc.S, simplemente hay que quitar los símbolos de comentario a las líneas implicadas.

Para conocer las direcciones IP de servidores de nombres que puedan sernos útiles podemos hacer dos cosa, escoger las dispuestas en los ficheros de ejemplo, poner dos o tres direcciones conocidas por nosotros o utilizar este pequeño 'truco'37:

Invocaremos, una vez conectados, el programa nslookup que sirve básicamente para la obtención de información de nuestro servidor de nombres sobre direcciones IP, dominios y sus parámetros.

```
# nslookup
Default Name Server: localhost
Address: 127.0.0.1
```

A partir de este momento entramos en lo que se denomina modo interactivo representado por el signo '>', teclearemos.

```
> set type=NS
> .
```

Se nos responderá con una lista de servidores de nombres y sus correspondientes direcciones IP que podremos aprovechar para nuestro fichero.

Como ya se ha dicho este es un ejemplo de como podemos aprovechar BIND para habilitar nuestra máquina como servidora (local) de nombres. Las posibilidades de este software son mucho mas extensas, así como la forma en que podemos combinar este mecanismo con la utilización de los ficheros de configuración de cara a obtener los mismos o incluso mejores resultados, por lo que dependiendo del tipo de instalación y el tipo de acceso a Internet disponible se puede profundizar en este tema o incluso obviarlo si se detecta que no va a aportar una mejora importante.

```
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh Configure arch/i386/config.in
*
* General setup
*
Kernel math emulation (CONFIG_MATH_EMULATION) [n]
Normal floppy disk support (CONFIG_BLK_DEV_FD) [y]
Normal (MFM/RLL) disk and IDE disk/cdrom support (CON-
FIG_ST506) [y]
*
* Please see drivers/block/README.ide for help/info on IDE drives
*
Use old disk-only driver for primary i/f (CON-
FIG_BLK_DEV_HD) [n]
Use new IDE driver for primary/secondary i/f (CON-
FIG_BLK_DEV_IDE) [y]
Include support for IDE/ATAPI CDROMs (CON-
FIG_BLK_DEV_IDECD) [y]
XT harddisk support (CONFIG_BLK_DEV_XD) [n]
Networking support (CONFIG_NET) [n]
Limit memory to low 16MB (CONFIG_MAX_16M) [y]
PCI bios support (CONFIG_PCI) [n]
System V IPC (CONFIG_SYSVIPC) [y]
Kernel support for ELF binaries (CONFIG_BINFMT_ELF) [y]
Use -m486 flag for 486-specific optimizations (CON-
FIG_M486) [y]
*
* Loadable module support*
Set version information on all symbols for modules (CON-
FIG_MODVERSIONS) [n]
*
* SCSI support
*
SCSI support? (CONFIG_SCSI) [n]
*
* Skipping SCSI configuration options...
*
*
* CD-ROM drivers (not for SCSI or IDE/ATAPI drives)
*
Sony CDU31A/CDU33A CDROM driver support (CON-
FIG_CDU31A) [n]
Mitsumi (not IDE/ATAPI) CDROM driver support (CON-
FIG_MCD) [n]
Matsushita/Panasonic CDROM driver support (CON-
FIG_SBPCD) [n]
Aztech/Orchid/Okano/Wearnes (non IDE) CDROM support (CON-
FIG_AZTCD) [n]
Sony CDU535 CDROM driver support (CONFIG_CDU535) [n]
```

Figura 3.1:

```
*
* Filesystems
*
Standard (minix) fs support (CONFIG_MINIX_FS) [n]
Extended fs support (CONFIG_EXT_FS) [n]
Second extended fs support (CONFIG_EXT2_FS) [y]
xiafs filesystem support (CONFIG_XIA_FS) [n]
msdos fs support (CONFIG_MSDOS_FS) [y]
umsdos: Unix like fs on top of std MSDOS FAT fs (CON-
FIG_UMSDOS_FS) [n]
/proc filesystem support (CONFIG_PROC_FS) [y]
ISO9660 cdrom filesystem support (CONFIG_ISO9660_FS) [y]
OS/2 HPFS filesystem support (read only) (CON-
FIG_HPFS_FS) [n]
System V and Coherent filesystem support (CON-
FIG_SYSV_FS) [n]
*
* character devices
*
Cyclades async mux support (CONFIG_CYCLADES) [n]
Parallel printer support (CONFIG_PRINTER) [y]
Logitech busmouse support (CONFIG_BUSMOUSE) [n]
PS/2 mouse (aka ªuxiliary device") support (CON-
FIG_PSMOUSE) [n]
Microsoft busmouse support (CONFIG_MS_BUSMOUSE) [y]
ATIXL busmouse support (CONFIG_ATIXL_BUSMOUSE) [n]
QIC-02 tape support (CONFIG_QIC02_TAPE) [n]
QIC-117 tape support (CONFIG_FTAPE) [n]
*
* Sound
*
Sound card support (CONFIG_SOUND) [n]
*
* Kernel hacking
*
Kernel profiling support (CONFIG_PROFILE) [n]
```

Figura 3.2: Ejemplo de configuración del núcleo.

```
make[1]: Entering directory '/usr/src/linuxelf-
1.2.13/arch/i386/boot'
make[1]: Nothing to be done for 'dep'.
make[1]: Leaving directory '/usr/src/linuxelf-
1.2.13/arch/i386/boot'
touch include/linux/version.h
for i in init/*.c;do echo -n "init/";gcc -D__KERNEL__ -
I/usr/src/linuxelf-1.2.13/include -E -D__ELF__ -
M $i;done > .tmpdepend
set -e; for i in kernel drivers mm fs net ipc lib arch/i386/kernel arch/i386/m
ke -C $i dep; done
make[1]: Entering directory '/usr/src/linuxelf-
1.2.13/kernel'
```

Figura 3.3: Fragmento de la salida por pantalla del comando.

```
boot.b
any_b.b
any_d.b
chain.b
os2_d.b
```

Figura 3.4: Contenido del directorio /boot.

```
boot=/dev/hda2
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/vmlinuz
label=linux
root=/dev/hda2
read-only
other=/dev/hda1
label=dos
table=/dev/hda
```

Figura 3.5: Ejemplo de fichero lilo.conf

```
 boot=/dev/hda11
 map=/boot/map
 install=/boot/boot.b
 image=/vmlinuz
 label=linux
    root=/dev/hdb1
    read-only
 other=/dev/hda1
    label=dos
    table=/dev/hda
```

Figura 3.6: Fichero lilo.conf para sistemas operativos en el segundo dico rígido.

```
 boot=/dev/fd0
 map=/mnt/map
 install=/mnt/boot.b
 image=/mnt/vmlinuz
 compact12
 label=linux
    root=/dev/hda2
    read-only
 other=/dev/hda1
    label=dos
    table=/dev/hda
```

Figura 3.7: Para instalar Lilo en un disquete

```
Variable
-b <dispositivo>
boot=<dispositivo>
-c
compact
-d <valor>
delay=<valor>
-i <dispositivo>
install=<dispositivo>
-l
linear
-m <fichero>
map=<fichero>
-r <dispositivo>
root=<dispositivo>
-t
--
-f <fichero>
disk=<dispositivo>
```

Figura 3.8: Tabla de equivalencias

```
/dev/hda3         swap    swap    defaults   1   1
/dev/hda2         /       ext2    defaults   1   1
/dev/hda1         /dosc   msdos   defaults   1   1
none              /proc   proc    defaults   1   1
```

Figura 3.9: Fichero fstab

```
 Adding a new user. The username should not ex-
ceed 8 characters
 in lenght, or may run into problems later.
 Enter login name for new user (\^C to quit): carlos

 Editing information for new user [carlos]

 Full Name: Carlos Laveda
 GID[100]:¿
 Group 'users', GID 100
 First unused UID is 502
 UID[502]:¿
 Home Directory [/home/carlos]:¿
 Shell [/bin/bash]:¿
 Passwword [carlos]: <clave>

 Information for new user [carlos]:
 Home directory [/home/carlos]: Shell: [/bin/bash]
 uid:[502]  gid:[100]

 Is this correct? [y/N]: y <¿>

 Adding login [carlos] and making direc-
tory [/home/carlos]
 Adding the files from the /etc/skel directory:
 ./.kermrc -> /home/carlos/./.kermrc
 ./.less     -> /home/carlos/./.less
 ./.lessrc  -> /home/carlos/./.lessrc
```

Figura 3.10: Creación de un usuario

```
root:9A4EYdkmSyFoZ:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
operator:*:11:0:operator:/root:/bin/bash
games:*:12:100:games:/usr/games:
man:*:13:15:man:/usr/man:
nobody:*:-1:100:nobody:/dev/null:
guest:*:405:100:guest:/dev/null:/dev/null
jose:DaK/OUkSRLi/2:501:100:Jose Manuel Lave-
da Molina:/home/jose:/bin/bash
carlos:EpxhzVZnChSso:502:100:Carlos Laveda:/home/carlos:/bin/bash
```

Figura 3.11: Aspecto del fichero /etc/passwd

```
Console: colour EGA+ 80x25, 1 virtual console (max 63)
Calibrating delay loop.. ok - 7.83 BogoMips
Serial driver version 4.11 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16450
tty01 at 0x02f8 (irq = 3) is a 16450
lp1 at 0x0378, using polling driver
hda: WDC AC2850F, 814MB w/64KB Cache, LBA, CHS=1654/16/63, MaxMult=16
hdb: GCD-R540, ATAPI, CDROM drive
ide0: primary interface on irq 14
Floppy drive(s): fd0 is 1.44M, fd1 is 1.2M
FDC 0 is a 8272A
Memory: 7168k/8192k available (452k kernel code, 384k re-
served, 188k data)
Swansea University Computer Society NET3.019
Checking 386/387 coupling... Ok, fpu using old IRQ13 error reporting
Checking 'hlt' instruction... Ok.
Linux version 1.2.13 (root@rita) (gcc ver-
sion 2.7.0)\#38 Mon Jun 24 20:58:43 MET DST1996
Partition check:
    hda: hda1 hda2 hda3
 VFS: Mounted root (ext2 filesystem) readonly.
Adding Swap: 24692k swap-space
hdb: media changed
hdb: status error: status=0x59
hdb: status error: error=0x60
ide0: do_ide_reset: success
VFS: Disk change detected on device 3/64
Max size:287861   Log zone size:2048
First datazone:20   Root inode number 40960
```

Figura 3.12: Información mostrada durante el arranque del sistema

```
# Nivel de ejecución por defecto, de no existir se pre-
guntará al arrancar.
id:5:initdefault:
# Ejecutar el ficheor rc.S al inicializar el siste-
ma (sysinit).
si:S:sysinit:/etc/rc.d/rc.S
# Ejecutar el fichero rc.K al entrar en modo simple (ni-
vel 1)
su:1S:wait:/etc/rc.d/rc.K
# Para pasar al modo multi-usuario ejecutar rc.M
rc:123456:wait:/etc/rc.d/rc.M
# Ejecutar el comando especificado al pul-
sar <Ctrl>+<Alt>+<Del>
ca::ctrlaltdel:/sbin/shutdown -t5 -rfn now
# Ejecutar estos ficheros al detener o arrancar el siste-
ma (actualmente es el
# mismo para ambos casos)
l0:0:wait:/etc/rc.d/rc.0
l6:6:wait:/etc/rc.d/rc.6
# Que hacer ante un fallo en el suministro eléctrico.
pf::powerfail:/sbin/shutdown -f +5 "¡¡ FALLO EN EL SUMI-
NISTRO ELECTRICO !!"
# El suministro eléctrico vuelve a ser normal.
pg:0123456:powerokwait:/sbin/shutdown -c "SUMINIS-
TRO CORRECTO"
# Volvemos al modo multi-usuario despues de volver to-
do a la normalidad.
ps:S:powerokwait:/sbin/init 5
# Aquí es donde especificamos el número de termina-
les virtuales que
# deseamos, el nivel 4 no se utiliza.
c1:1235:respawn:/sbin/agetty 38400 tty1 linux
c2:1235:respawn:/sbin/agetty 38400 tty2 linux
c3:5:respawn:/sbin/agetty 38400 tty3 linux
c4:5:respawn:/sbin/agetty 38400 tty4 linux
c5:5:respawn:/sbin/agetty 38400 tty5 linux
c6:45:respawn:/sbin/agetty 38400 tty6 linux
# Terminales enganchadas al puerto serie
#s1:45:respawn:/sbin/agetty 19200 ttyS0 vt100
#s2:45:respawn:/sbin/agetty 19200 ttyS1 vt100
# Terminales invocadas via modem
#d1:45:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS0 vt100
#d2:45:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS1 vt100
# El nivel 4 se reserva para la ejecución del entor-
no X Window.
x1:4:wait:/etc/rc.d/rc.4
```

Figura 3.13: Ejemplo de fichero inittab

```
 * General setup
 Networking support (CONFIG_NET) [y] y
```

Figura 3.14: Habilitar el soporte para redes en LINUX

```
  *
  * Networking options
  *
  TCP/IP networking (CONFIG_INET) [y]
  IP forwarding/gatewaying (CONFIG_IP_FORWARD) [n]
  IP multicasting (CONFIG_IP_MULTICAST) [n]
  IP firewalling (CONFIG_IP_FIREWALL) [n]
  IP accounting (CONFIG_IP_ACCT) [n]
  *
  * (it is safe to leave these untouched)
  *
  PC/TCP compatibility mode (CONFIG_INET_PCTCP) [n]
  Reverse ARP (CONFIG_INET_RARP) [n]
  Assume subnets are local (CONFIG_INET_SNARL) [y]
  Disable NAGLE algorithm (normally enabled) (CON-
FIG_TCP_NAGLE_OFF) [n]
  The IPX protocol (CONFIG_IPX) [n]
  *
```

Figura 3.15: Opciones generales para el soporte de redes.

```
 B.ROOT-SERVERS.NET      internet address = 128.9.0.107
 H.ROOT-SERVERS.NET      internet address = 128.63.2.53
 A.ROOT-SERVERS.NET      internet address = 198.41.0.4
 G.ROOT-SERVERS.NET      internet address = 192.112.36.4
 F.ROOT-SERVERS.NET      internet address = 192.5.5.241
 I.ROOT-SERVERS.NET      internet address = 192.36.148.17
```

Figura 3.41: Parte de la información mostrada por nslookup.

```
lo
          Link encap:Local Loopback
          inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUN-
NING  MTU:2000  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:26 errors:0 dropped:0 overruns:0

dummy

          Link encap:10Mbps Ethernet  HWaddr 00:00:00:00:00:00
          inet addr:192.168.34.56  Bcast:192.168.34.255  Mask:255.255.255.0
          UP BROADCAST RUNNING NOARP MULTI-
CAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:0 Base address:0x0
```

Figura 3.17: Apariencia del dispositivo dummy.

```
 order hosts,bind
 multi on
```

Figura 3.18: Fichero /etc/hosts.conf

Capítulo 4

X-WINDOW

4.1 Introducción.

X Window es un entorno gráfico desarrollado para el sistema operativo UNIX por el MIT1. Proporciona al sistema un interfaz completamente gráfico en base a ventanas en las que se ejecutarán las distintas aplicaciones que podemos ejecutar, menús desplegables,... todo ello con el uso del ratón en todo momento.

Si se está familiarizado con cualquier otro entorno similar, no será muy difícil el trabajar en el nuevo que se ofrece. De esta forma se cubre una de las deficiencias achacadas al UNIX en general en un momento en que cada vez es más importante el aspecto, y la sencillez de manejo para el usuario final.

Para ser más concretos en la definición hay que decir que X está desarrollado bajo el modelo cliente-servidor. La parte que realiza la tarea de servidor gestiona todo el acceso al hardware. El software cliente se comunica con el servidor mediante el envío de órdenes (mensajes) destinadas a la gestión del entorno y muestra de datos mediante éste. Ejemplo de clientes son xterm (simula un terminal en una ventana), xman, xview, Mosaic (existe una versión para LINUX), etc.. cada uno de lo que coloquialmente denominamos "programas para X "ó "programas para ejecutar bajo X"(aunque con xterm también podemos ejecutar cualquier otro software que no esté concebido para X). El servidor se encarga de algunas tareas como las descritas a continuación:

- Manipulación a bajo nivel de ventanas (operaciones de creación y destrucción de las mismas), sin confundirlo con el manejo que posteriormente podemos hacer mediante el gestor escogido.

- Servir como gestor de recursos del sistema para las aplicaciones a ejecutar.

- Procesar órdenes para el dibujo de gráficos.
- Notificar a las aplicaciones los eventos ocurridos (pulsado de una tecla, movimiento del ratón, errores...)
- Monitorizar y comunicar a las aplicaciones los errores que se pueden generar.
- Control de la pantalla.
- Comunicación entre los diferentes clientes (cortar y pegar...)

Incluso el gestor de ventanas (hay varios disponibles) es un programa cliente más de los que se pueden ejecutar, es más, será necesario ejecutar uno de ellos obligatoriamente para trabajar en este entorno, algunas de sus tareas son:

- Permitir redimensionar las ventanas.
- Mover las ventanas a cualquier posición de la pantalla.
- Convertir una ventana en un icono y viceversa.
- Decorar ventanas con barras de títulos y desplazamiento, bordes, botones, etc.
- Control de todas las ventanas abiertas en un determinado momento.
- Lanzamiento de aplicaciones.

X Window está concebido para entornos de red, clientes y servidor pueden ser ejecutados en máquinas distintas, el proceso de comunicación cliente-servidor (paso de mensajes) es totalmente transparente para el usuario.

En nuestro caso ejecutaremos el servidor y todos los clientes en la misma máquina (para LINUX también es una red) lo cual no supone problema alguno.

La versión actual del entorno X Window es la número once, en su revisión número 6 y pertenece al denominado "X Consortium", grupo formado en 1988 por el MIT junto a otras empresas como DEC, IBM,.. cuyo objetivo es la cooperación con la industria para la creación y evolución de un estándar gráfico basado en el X Window.

Existen varias categorías de miembros en esta organización, estructurado en forma de equipos multidisciplinares que participan en el desarrollo, implementación, revisión y prueba de cualquier idea que surja en el ámbito del desarrollo del entorno gráfico. Cualquier persona tiene cabida en esta entidad.

XFree86 es la implementación del X Window que tenemos a nuestra disposición para LINUX (y otras implementaciones del

UNIX) en su versión 3.1.22, basada en el servidor X386 desarrollado por el MIT destinado a plataformas de Intel 386/486 y Pentium además de la configuración hardware común a este tipo de procesadores. Para conocer más detalles del software instalado podemos ejecutar el comando ’xdpyinfo’ en una ventana de terminal el cual nos mostrará información detallada sobre la versión y características del entorno gráfico.

XFree86 está distribuido bajo una licencia que permite disponer libremente tanto de software que lo forma en forma de código fuente como en formato binario. Está desarrollado, mantenido y distribuido por grupo XFree86 Project Inc., cuyo fin es el desarrollar el servidor X Window para cualquier UNIX ejecutable en plataformas PC (http://www.xfree.org).

En la distribución que hemos tomado como referencia (Slackware), las series que empiezan con la letra X (X, XAP, XD,....) son las relacionadas con este software. Para nuestros fines será necesario instalar las series X y XAP, con ello dispondremos del X Window y algunas aplicaciones que nos pueden ser de utilidad para el entorno.

Hay que avisar de algo muy importante, el software viene sin configurar, esto es, una vez instalado hay que realizar este proceso a fin de poder ejecutarlo. Intentar ejecutar este interfaz sin haberlo configurado previamente puede producir efectos no deseados, incluso en la documentación adicional se nos insiste en la posibilidad de dañar nuestro hardware por lo que se recomienda encarecidamente el documentarse bien antes de dar este paso.

En este capítulo se intentará explicar lo más detalladamente posible todos los pasos necesarios para poder configurar este software, o al menos, el poder ejecutar XFree86 de una forma que denominaré ’mínima’3 (el entorno funciona en sólo un modo) para, a partir de este punto, poder profundizar en el resto de parámetros que rodean a este proceso.

XFree86 no es sencillo de configurar, o más concretamente, no es tan sencillo como lo pueden ser otros entornos gráficos (algunos incluso tienen la capacidad de detectar el hardware y autoconfigurarse), pero esta dificultad es directamente proporcional a su flexibilidad, XFree86 puede ser ejecutado en un amplísimo conjunto del hardware disponible en el mercado destinado al uso de gráficos perteneciente a distintos fabricantes y , por tanto, con la ausencia de estándares bien definidos que faciliten la tarea. Muchos fabricantes anuncian sus productos como "compatibles con ...ªunque a la hora de la verdad nos podemos encontrar que esta compatibilidad no es total.

En el proceso que nos ocupa tendremos que enfrentarnos al manejo de parámetros con los que puede que no estemos familiarizados en absoluto, tales como ancho de banda, tasas de desplazamiento vertical y horizontal, pulsos de reloj... que son los que interesa conocer para nuestro propósito. Se va a tener siempre en cuenta lo que se puede denominar como 'caso peor', no se dispone de la información técnica necesaria sobre la tarjeta gráfica montada en el ordenador ni del monitor o esta no se ajusta a la contenida en alguno de los ficheros con información de los que acompañan al Xfree86.

Si nuestro hardware se ajusta exactamente a las especificaciones de alguno de estos ficheros tendremos el 90 hay que olvidar que la mayoría de los PC de los que hay en nuestro entorno carecen de los manuales con las especificaciones técnicas mínimas del tipo de componentes que lo forman.

Existen en el mercado implementaciones de este tipo de software dedicado a su explotación comercial cuya ventaja puede consistir en una mayor estabilidad, soporte técnico, facilidad de instalación y configuración, mayor rendimiento, etc... A pesar de poder existir diferencias todos son compatibles entre sí.

4.2 Información Disponible

A la hora de enfrentarnos a la configuración del XFree86 disponemos de la siguiente información:

- Ficheros FAQ y HOWTO:
  - /usr/doc/faq/faq/linux-faq.xxx4
  - /usr/doc/faq/howto/XFree86-HOWTO
  - Directorio /usr/doc/faq/faq/xfaq
- Otros ficheros:
  - Contenidos en el directorio /var/X11R6/lib/doc. Nos pueden ser de ayuda en el momento de la configuración con información para cada uno de los servidores de los que disponemos,sobre tarjetas gráficas, monitores, ...

Se recomienda leer los siguientes ficheros (como mínimo): README, README.Config, VideoModes.doc y README.Linux además de las páginas del manual (comando man) referentes al XFree86 y al servidor escogido.

4.3 Hardware necesario

XFree86 puede ser ejecutado en la mayoría del hardware existente en el mercado destinado a los gráficos. Dependiendo del servidor escogido podremos ejecutar el entorno X en diferentes tipos de hardware, cada servidor está concebido para soportar un grupo concreto de tarjetas gráficas, desde las monocromo, pasando por las VGA, SVGA, ... todo ello con buses del tipo ISA, VESA o PCI. Como ejemplo a continuación se detalla parte del hardware que puede manejar el servidor XF86$_SVGA, destinadoalma$

- Tseng ET3000, ET4000AX, ET4000/W32
- Western Digital/Paradise PVGA1
- Western Digital WD90C00, WD90C10, WD90C11, WD90C24, WD90C30, WD90C31, WD90C33
- Genoa GVGA
- Trident TVGA8800CS, TVGA8900B, TVGA8900C, TVGA9000i, TVGA9100B, TVGA9200CX, TVGA9320, TVGA9400CX, TVGA9420
- ATI 18800, 18800-1, 28800-2, 28800-4, 28800-5, 28800-6, 68800-3, 68800-6, 68800AX, 68800LX, 88800
- NCR 77C22, 77C22E, 77C22E+
- Cirrus Logic CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD6205, CLGD6215, CLGD6225, CLGD6235, CLGD6420
- Compaq AVGA
- OAK OTI067, OTI077
- Avance Logic AL2101
- MX MX68000, MX680010
- Video 7/Headland Technologies HT216-32

Dos definiciones previas:

Servidor : es la parte del software que va a encargarse del manejo del hardware tanto gráfico como el resto de componentes del ordenador, establecerá un interfaz con todas las aplicaciones cliente que correrán en el sistema y que simplemente han de enviar las órdenes precisas al servidor para obtener los resultados necesarios.

En la configuración del XFree86 no hablaremos de las tarjetas gráficas refiriéndonos a su nombre (marca y modelo) sino al chipset (juego de chips) que la integran, en la definición del hardware que soporta cada servidor nos vamos a encontrar con la especificación de los juegos que puede manejar, agrupados por sus fabricantes. Podemos conocer la tarjeta que tenemos pero nos hace falta saber la circuitería que integra.

El primer paso que tenemos que dar es escoger un servidor de los que tenemos disponibles, esto es imprescindible (la utilidad setup nos obligará a escoger uno) antes de iniciar nuestro cometido.

También hay que hacer una reseña en cuanto al resto del hardware necesario, XFree86 necesita gran cantidad de recursos para funcionar. En la documentación de que se dispone se indica que un procesador Intel 386 (o compatible) con la máxima velocidad posible y 4 Mb. de memoria RAM es la configuración mínima para poder ejecutar X, pero se recomienda al mismo tiempo el disponer de un procesador modelo 486 y de 8 Mb. de memoria RAM para poder ejecutar el entorno en unas condiciones optimas. En general se puede decir que, independientemente del procesador habremos de disponer de 16 Mb de memoria virtual como mínimo (8 Mb. de RAM y 8 de área de intercambio) para que el sistema funcione bien, a más memoria principal, mejor. Si no disponemos de área de intercambio teniendo 16 Mb. de memoria principal o más no será necesario el configurarla.

4.4 Instalación del Software

Una vez más podemos acudir al programa SETUP para realizar este proceso. Hemos de instalar las series X y XAP como mínimo para poder utilizar la versión del X Window que vamos a utilizar. Si posee un conocimiento exacto del software que necesita instalar y donde se encuentra puede utilizar PKGTOOL, pero conviene recordar que instalando este software 'a mano' habrá que introducir un comando a fin de dejar el sistema a punto para su configuración, todo esto se describe a continuación.

El primer paso a dar es escoger el servidor que se ajusta a nuestras necesidades, esto lo podemos hacer consultando los ficheros contenidos en el directorio /var/X11R6/lib/doc con nombre 'README.xxx' siendo xxx el nombre de cada uno de los servidores y además podemos recurrir al manual de cada uno de ellos5. Los servidores de los que disponemos son:

- XF86$_SVGA : Servidorgen\acute{e}ricoparatarjetasSVGA.Contienesoporteparajuegosdechipsacelerados6$
- XF86$_Mono : ParatarjetasdeltipoMonocromo, inclusolasH\acute{e}rcules.$
- XF86$_VGA16 : ParatarjetasVGAconcapacidadpara16colores.$
- XF86$_S3 : ServidoraceleradoparatarjetasconchipsS3.$
- XF86$_Mach32 : ServidoraceleradoparatarjetasconchipsATIMach32.$
- XF86$_Mach64 : ServidoraceleradoparatarjetasconchipsATIMach64.$
- XF86$_Mach8 : ServidoraceleradoparatarjetasconchipsATIMach8.$
- XF86$_8514 : Servidoraceleradoparatarjetasconchips8514/A.$
- XF86$_P9000 : ServidoraceleradoparatarjetasconchipsP9000.$
- XF86$_AGX : ServidoraceleradoparatarjetasconchipsAGX.$

- XF86$_W32 : Servidoraceleradoparatarjetasconchips ET4000/W32.$

Una vez escogido el servidor que mejor se adapta a nuestras necesidades daremos paso a la instalación del software. Si lo hacemos mediante el programa SETUP el mismo hará todas las tareas de instalación necesarias por nosotros, de hacerlo mediante PKGTOOL o con los comandos tar y gzip hemos de asegurarnos que el servidor esté correctamente instalado mediante el comando:

```
# ln -sf /usr/X11R6/bin/XF86_SVGA  /usr/X11R6/bin/X
```

Esto creará un enlace al fichero XF86$_SVGA(puestocomoejemplo)llamadoX, enestecasotendremosque$

```
# ls -all X
```

obtendremos por pantalla el siguiente resultado:
lrwxrwxrwx 1 root root 24 Mar 19 18:21 X->/usr/X11R6/bin/XF86$_SVGA$

Si la instalación se hace mediante SETUP es recomendable hacerla con la opción VERBOSE activada de forma que podemos ver y confirmar todo el software que se va a instalar, de esta forma nos podemos hacer una idea de todas las utilidades de que vamos a disponer para utilizar el entorno.

4.5 Configuración.

4.5.1 Detectando el Hardware.

En este momento y tras revisar todos los conceptos desarrollados en los capítulos anteriores nos podemos encontrar ante dos situaciones bien distintas:

1. Conocemos perfectamente todos los parámetros necesarios, ya que disponemos de información técnica suficiente tanto del monitor como de la tarjeta gráfica que tenemos instalada o, por otra parte, nuestro hardware coincide exactamente con la información disponible en alguno de los ficheros de ayuda que acompañan a Xfree86. En este caso el proceso será muy sencillo, ya que únicamente tendremos que trasladar esto parámetros al lugar correcto y, salvo pequeños ajustes, tendremos funcionando nuestro servidor X y alguna de las aplicaciones con las que contamos (clientes).

2. No disponemos de información alguna referente al ordenador en el que vamos a instalar XFree86 (cosa bastante frecuente en los denominados ordenadores "clónicos"). Aquí tendremos que recurrir igualmente a los ficheros mencionados, pero antes podemos ejecutar un comando que nos va a servir para obtener información muy útil sobre nuestro sistema:

```
# X -probeonly > salida.tmp1 2>&18
```

De esta forma en el fichero creado (salida.tmp1) tendremos información que nos puede ser de interés, la entrada

```
'2>&1'
```

permite que el fichero recoja la salida estándar y los errores.

En el último punto de este apartado se describe mayor información sobre todos los valores que se necesitan conocer y cómo extraerlos a partir de la información técnica disponible sobre la tarjeta gráfica y el monitor. La configuración detallada a continuación se basa en parámetros que pueden funcionar en nuestra máquina pero no por ello son los óptimos, se tratan en general de contribuciones de distintos autores, valores estándar para una familia de periféricos, etc...

```
XFree86 Version 3.1.2 / X Window System
(protocol Version 11, revision 0, vendor release 6000)
Operating System: Linux
Configured drivers:
  SVGA: server for 8-bit colour SVGA (Patchlevel 0):
      et4000, et4000w32, et4000w32i, et4000w32p, et3000, pvga1, wd90c00,
      wd90c10, wd90c30, wd90c24, wd90c31, wd90c33, gvga, vgawonder,
      tvga8800cs, tvga8900b, tvga8900c, tvga8900cl, tvga9000, clgd5420,
      clgd5422, clgd5424, clgd5426, clgd5428, clgd5429, clgd5430, clgd5434,
      clgd5436, clgd6205, clgd6215, clgd6225, clgd6235, ncr77c22, ncr77c22e,
      cpq_avga, oti067, oti077, oti087, mx, al2101, ali2228, ali2301,
      ali2302, ali2308, ali2401, cl6410, cl6412, cl6420, cl6440, video7,
      ct65520, ct65530, ct65540, ct65545, ark1000vl, ark1000pv, ark2000pv,
      realtek, generic
(using VT number 7)

XF86Config: /usr/X11R6/lib/X11/XF86Config
(**) stands for supplied, (--) stands for probed/default values
(**) Mouse: type: Microsoft, device: /dev/mouse, baudrate: 1200
(**) SVGA: Graphics device ID: "Generic SVGA"
(**) SVGA: Monitor ID: "Generic Monitor"
(**) FontPath set to "/usr/X11R6/lib/X11/fonts/misc/"
(--) Trident chipset version: 0x13 (TVGA8900C)
(--) SVGA: chipset:  tvga8900c
(--) SVGA: videoram: 1024k
(--) SVGA: clocks:  25.14  28.32  44.84  35.85  57.15  64.84  50.29  39.84
(--) SVGA: Maximum allowed dot-clock: 90.000 MHz
(**) SVGA: Mode "640x480": mode clock =  25.175, clock used =  25.140
(**) SVGA: Virtual resolution set to 800x600
(--) SVGA: SpeedUp code selection modified because virtualX != 1024
```

Figura 4.1: Aspecto del fichero salida.tmp1

Podemos ver que ya tenemos información sobre la tarjeta gráfica que tenemos instalada y una línea indicada como 'clocks' que puede ser de utilidad posteriormente.

Existe un programa que también ayuda a dar información sobre nuestra tarjeta gráfica que se indica como opcional si el proceso anterior no ha tenido el éxito esperado, este es el Superprobe. Tras su invocación tendremos una pantalla similar a la siguiente y la información que se indica:

```
SuperProbe Version 2.5
        (c) Copyright 1993,1994 by David Wexelblat <dwex@xfree86.org>
(...)
 6-bit First video: Super-VGA
        Chipset: Trident 8900C
```

```
        Memory:  1024 Kbytes
        RAMDAC:  Generic 8-bit pseudo-color DAC
                 (with 6-bit wide lookup tables (or mode))
```

Tenemos igualmente las características de nuestra tarjeta gráfica con la que podemos acceder a los ficheros Devices y ModeDB.txt para buscar los valores para este parámetro que coincidan.

Una vez hemos hecho este proceso previo vamos a crear el fichero principal de configuración para nuestro sistema, XF86Config, ubicándolo en el directorio /var/X11/lib. Tenemos uno con el nombre XF86Config.eg que nos va a servir como base para desarrollar nuestro propio fichero de configuración, si lo deseamos podemos crear una copia del mismo a partir de la cual trabajar:

```
# cp -a XF86Config.eg XF86Config
```

4.5.2 El Fichero XF86Config.

Está estructurado en siete secciones principales, cada una de las cuales contiene información referente a distintos parámetros que va a necesitar para funcionar el servidor gráfico, estos son:

Files Directorios donde va a buscar los ficheros necesarios para poder utilizar distintas fuentes y colores.

ServerFlags Algunas opciones a utilizar a la hora de ejecutar el servidor X.

Keyboard Configuración de algunas teclas especiales y la forma en que el servidor va a responder ante éstas.

Pointer Para la configuración del ratón.

Monitor Datos sobre el monitor.

Device Aquí es donde especificaremos todos los datos referentes a la tarjeta gráfica.

Screen Configuración del área de trabajo de la que podemos disponer en nuestra pantalla.

Todas las secciones tienen una sintaxis similar:

```
Section "Nombre de la Sección"
        Parámetros
                .
                .
EndSection
```

Cada una de las secciones será descrita con el fin de especificar la estructura de nuestro sistema, en general no hay que hacer muchos cambios en este fichero, exceptuando las secciones Monitor y Device.

El comando introducido para la creación del fichero salida.tmp1 extraerá ciertos valores del fichero (los marcados con el símbolo (**)) y los marcados con el símbolo (--) los calculará a partir del hardware disponible y ciertos algoritmos que el servidor contiene.

Las modificaciones a realizar en el fichero las podemos hacer con cualquiera de los procesadores de texto que acompañan al LINUX (vim, jed, jove, emacs...), con el que estemos más familiarizados.

Files

En esta sección generalmente no se ha de hacer cambio alguno, a no ser que hayamos instalado determinados tipos de fuentes de carácteres extra que deseamos que el servidor encuentre. Si durante la instalación del software hemos instalado las fuentes denominadas '75dpi' y '100dpi' incluiremos estas líneas en la sección (eliminando el signo '#' del comienzo de cada línea).

Serverflags

Tenemos tres parámetros que afectarán al comportamiento del servidor durante su funcionamiento.

- NoTrapSignals permite al servidor el interrumpirse a la llegada de cierto tipo de señales y crear un fichero que después podremos revisar con un programa depurador (debugger). La opción por defecto consiste en una salida 'limpia' del programa.
- DontZap Inhabilita el uso de la combinación de teclas <Ctrl>+<Alt>+<Del> para terminar la ejecución del servidor.

- DontZoom: Activada hará que no podamos cambiar la resolución (modos) del XFree86 a que podemos optar durante la ejecución del servidor mediante las teclas <Ctrl>+<Alt>+<TeclNum$_+ >$ $y < Ctrl > + < Alt > + < TeclNum_{-} > (signos + / - deltecladonum\acute{e}rico)$.

Sección Keyboard

En el apartado Protocol dejaremos definido el "Standard"ya que no es muy normal el tener que utilizar el modo "Xqueue"debido al hardware para el que está concebido.

- Autorepeat viene seguido de dos parámetros que van a permitirnos el afinar el comportamiento del teclado en cuanto a la capacidad de autorepetición cuando mantenemos una tecla pulsada durante un tiempo.

- ServerNumLock permite al servidor el utilizar la parte numérica del teclado.
- Xleds seguido de la serie de números del uno al tres (1, 1 2 ó 1 2 3) hará que el software cliente maneje los indicadores luminosos del teclado (NumLock, CapsLock y ScrollLock).
- LeftAlt, RightAlt, AltGr, ScrollLock y RightAlt seguido de uno de los parámetros siguientes: Meta, Compose, ModeShift, ModeLock, ScrollLock, Control nos va a definir el comportamiento de estas teclas que podíamos considerar como 'especiales'. Si no especificamos nada los valores por defecto serán:

  ```
  LeftAlt Meta.
  RightAlt Meta.
  RightCtrl Compose.
  ScrollLock ModeLock.
  ```

  Será necesario modificar el valor de la tecla RightAlt a ModeShift, de esta forma el servidor reconocerá la tecla marcada generalmente como "Alt Gr".

Sección Pointer

- Microsoft es el parámetro que utilizaremos en el apartado Procotol dado que la mayoría de los ratones existentes son compatibles con este protocolo, si disponemos de un ratón que no responde con este protocolo podemos probar con uno de los siguientes: BusMouse, Logitech, MMSeries, Mouseman, MouseSystems, PS/2, MMHitTab, OSMouse.

  El apartado Device contiene el fichero que manejará el ratón y que estará contenido en el directorio /dev, esta línea puede contener '/dev/mouse' como enlace al nombre del manejador del puerto en el que tengamos conectado el dispositivo. Es conveniente el comprobar si existe este enlace, si necesitamos crearlo nosotros mismos o, si lo deseamos, podemos poner el nombre directamente del manejador de la forma '/dev/ttyS0' (puerto al que lo tengamos conectado).
- BaudRate <valor> indicará al servidor la velocidad de comunicación con el ratón (en baudios). El valor estándar es 1200.
- Emulate3Buttons y Emulate3Timeout <valor> permiten al servidor el emular la existencia de un tercer botón en nuestro ratón en el supuesto que no lo tuviese o no lo reconoce existiendo éste, esto se hace pulsando los dos bo-

tones simultáneamente (en uno con tres botones no utilizaremos el botón central).

- ChordMidle reconoce el mencionado 3er. botón del ratón, aunque se especifica que es para ratones del tipo Logitech.
- SampleRate <valor> Fija el número de eventos que el ratón enviará al servidor por segundo. Válido solo para algunos ratones Logitech.
- ClerDTR 'limpia' la línea DTR del puerto serie usado por el ratón. Válido sólo para ratones que utilizan el protocolo MouseSystems.
- ClearRTS Tiene el mismo objetivo que el parámetro anterior pero en este caso se encarga de la línea RTS.

Sección Monitor.

En esta sección especificaremos el tipo de monitor disponible y los modos (resolución) con los que queremos trabajar, hay varios tipos disponibles dependiendo del servidor, monitor y tarjeta gráfica: 640x480, 800x600, 1024x768 y 1280x1024. Pero no sólo existen estos modos, en teoría se puede configurar cualquier modo que nuestro hardware pueda gestionar siempre que los valores introducidos indiquen al servidor como ha de comportarse.

En función del hardware podremos trabajar con una, dos, tres o todas las resoluciones disponibles sin más que pulsar la combinación de teclas <Ctr>+<Alt> y el signo '+' ó '-' del teclado numérico durante la ejecución de X-Window, generalmente los modos 640x480 y 800x600 son soportados por todo tipo de hardware.

En el fichero README.Config como apartado final tenemos una serie de modos disponibles y los parámetros que tendremos que introducir en nuestro fichero XF86Config.

En la figura podemos ver los modos agrupados por la resolución y con los parámetros a utilizar en el fichero de configuración, a éstos valores debemos de hacer constante referencia a la hora de configurar la sección que se describe.

A continuación se explica la forma en que hemos de completar esta sección únicamente de forma que tengamos disponible un par de modos o resoluciones. Si deseamos disponer de más modos el mecanismo es similar.

Dependiendo de la tarjeta gráfica y el monitor instalados optaremos por una de las formas en que podemos configurar los modos que se nos ofrece.

Los ítems que forman la sección son:

```
  #  Mode       Refresh  Hor. Sync  Dot-clock  Interla-
ced?  VESA?
  #  ----------------------------------------------------------
---
  #  640x480     60Hz      31.5k     25.175M       No         No
  #  640x480     60Hz      31.5k     25.175M       No         No
  #  640x480     63Hz      32.8k     28.322M       No         No
  #  640x480     70Hz      36.5k     31.5M         No         No
  #  640x480     72Hz      37.9k     31.5M         No        Yes
  #  800x600     56Hz      35.1k     36.0M         No        Yes
  #  800x600     56Hz      35.4k     36.0M         No         No
  #  800x600     60Hz      37.9k     40.0M         No        Yes
  #  800x600     60Hz      37.9k     40.0M         No         No
  #  800x600     72Hz      48.0k     50.0M         No        Yes
  #  1024x768i   43.5Hz    35.5k     44.9M        Yes         No
  #  1024x768    60Hz      48.4k     65.0M         No        Yes
  #  1024x768    60Hz      48.4k     62.0M         No         No
  #  1024x768    70Hz      56.5k     75.0M         No        Yes
  #  1024x768    70Hz      56.25k    72.0M         No         No
  #  1024x768    76Hz      62.5k     85.0M         No         No
  #  1280x1024i  44Hz      51kHz     80.0M        Yes         No
  #  1280x1024i  44Hz      47.6k     75.0M        Yes         No
```

Figura 4.2: Extracto de la tabla de modos disponibles

- Identifier «cadena>": Aquí hemos de introducir el nombre de nuestro monitor a fin de identificarlo. Este nombre deberá coincidir con el introducido posteriormente en la sección Screen. Si vamos a definir varias secciones Monitor cada una debe tener un identificador propio.
- VendorName «cadena>": Fabricante del Monitor.
- ModelName «cadena>": Modelo del Monitor.

  Estas dos últimas entradas son opcionales y pueden dejarse tal como están (al igual que la primera si sólo vamos a tener un monitor configurado).
- HorizSync <rango>: Especificaremos el valor/es del rango/s de frecuencia/s horizontal/es (Tasa de Desplazamiento Horizontal9) del monitor. Existen varias formas de especificar este valor, se puede ver en el fichero como ejemplo:
  - Valor constante: 31.510 para monitores con frecuencia fija.
  - Rango de Valores: 30-64 para monitores multifrecuencia.
  - Múltiples valores: 31.5, 35.2 para monitores con frecuencia fija y la necesidad de especificar varios valores.

```
 # 640x480@60Hz Non-Interlaced mode
 # Horizontal Sync = 31.5kHz
 # Timing: H=(0.95us, 3.81us, 1.59us), V=(0.35ms, 0.064ms, 1.02ms)
 #
 # name        clock   horizontal timing     vertical ti-
ming      flags
 "640x480"     25.175  640  664  760  800    480  491  493  525

 # Alternate 800x600@56Hz Non-Interlaced mode
 # Horizontal Sync = 35.4kHz
 # Timing: H=(0.89us, 4.00us, 1.11us) V=(0.11ms, 0.057ms, 0.79ms)
 #
 # name        clock   horizontal timing     vertical ti-
ming      flags
 "800x600"     36      800  832  976 1016    600  604  606  634
```

Figura 4.3: Algunos de los modos en su especificación concreta

  - Múltiples Rangos: 15-25, 30-50 para monitores multifrecuencia con varios rangos de valores.

  La frecuencia indica el número de líneas trazadas por segundo que el monitor puede soportar durante su trabajo.

  Esta forma de indicar los valores también nos será útil para el siguiente (VertRefresh).

  La unidad por defecto son los kiloherzios (kHz), aunque podemos especificar otros múltiplos sin más que indicarlo (Mhz para Megaherzios, Hz para Herzios).

- VertRefresh <rango>: Los valores que indiquemos en este apartado son para la Tasa de Desplazamiento Vertical del monitor, la unidad por defecto en este caso es el Herzio (podemos especificar también kiloherzios y megaherzios). En este caso damos a conocer el número de imágenes por segundo (entendiendo una imagen como el trazado de una pantalla completa) a que el monitor puede trabajar.

  Tanto la Tasa de Desplazamiento Horizontal como la Vertical ayudarán al servidor a conocer si los modos que vamos a especificar posteriormente son soportados por el monitor.

  Si desconocemos los valores para el monitor en el que estamos configurando el X Window tenemos tres opciones:

  1. Recurrir al fichero Monitors en el que tenemos una lista de monitores y los modos que se pueden configurar en estos. Está formada en base de contribuciones de usuarios.

```
#Date: Fri, 16 Sep 1994 23:16:32 -0700
#From: "Leonard N. Zubkoff" <lnz@dandelion.com>
Section "Monitor"
  Identifier "Apollo 1280x1024-68Hz"
  VendorName "Apollo"
  ModelName "010700-005"
  BandWidth 125
  HorizSync 73.702
  VertRefresh 68.24
  Mode "1280x1024"
    DotClock 124.996
    HTimings 1280 1312 1504 1696 VTimings 1024 1027 1030 1080
  EndMode
EndSection
```

2. En la distribución Slackware viene un programa llamado ConfigXF86 que ayuda en la configuración del servidor. Consta de una serie de menús en los que pregunta el tipo de hardware a configurar, algunos parámetros que debe conocer y éste creará un fichero (¡llamado Xconfig!) con los valores correctos. Seguimos teniendo el problema a la hora de no conocer exactamente el hardware disponible.
3. Escoger la entrada del fichero Monitors que más se parece al nuestro y probar a incluirlo en nuestro fichero, de esta forma haremos la prueba por 'ensayo y error', ya que al arrancar el Servidor éste nos indicará mediante un mensaje de error si los campos no están bien introducidos (podemos crear un fichero con la salida estándar para poder ver después el contenido de este en este caso). De esta forma seguimos corriendo el riesgo de dañar el monitor.

- Gamma <valor/es>: Valor opcional para introducir la corrección gamma de un monitor. Puede ser introducido como un valor o tres valores separados. No todos los servidores usan esta información.

- Mode «nombre>"Detallaremos a continuación los modos en que queremos que trabaje el monitor. Podremos definir tantos como el hardware pueda soportar. Es conveniente el intentarlo con un modo de los más simples (640x480 por ejemplo) hasta que funcione correctamente para, posteriormente, configurar los modos adicionales que deseemos.

  En este apartado existen igualmente dos formas de definir los campos que van a formar cada modo:

  ? ModeLine "640x480" 25.127 640 664 760 800 480 491 493 525
  ? Mode "640x480"

```
            DotClock        25.127
            HTimings        640 664 760 800
            VTimings        480 491 493 525
      EndMode
```

¿Que modos se pueden configurar y cómo se pueden conocer estos valores?

Para definir un modo hay que tener en cuenta que el valor DotClock del modo a definir coincide con uno de los valores que hemos obtenido de los DotClocks de la tarjeta gráfica y obtendremos el resto de parámetros del fichero README.Config.

Del fichero salida.tmp1 que hemos creado anteriormente observemos los valores obtenidos para la tarjeta gráfica:

```
(--) SVGA: clocks:  25.14 28.32 44.84 35.85 57.15 64.84
50.29 39.8411
```

Y que el valor marcado en 'negrita' coincide (aunque no exactamente) con el primer valor señalado para el modo escogido para el ejemplo:

```
? ModeLine "640x480" 25.127 640 664 760 800 480 491 493 525

? Mode "640x480"
        DotClock        25.127
        HTimings        640 664 760 800
        VTimings        480 491 493 525
  EndMode
```

Esta regla se tiene que aplicar para el resto de modos que existen en el fichero README.Config, el primer valor de la resolución escogida debe coincidir con uno de los valores de la línea clocks de la tarjeta gráfica.

Si tenemos dos formas de configurar un modo y en ambas el valor para DotClock coincide con alguno de los índices de la línea mencionada, podemos probar las dos formas y ver cual se ajusta más al hardware o funciona correctamente (siguiendo con el supuesto de no conocer el hardware disponible). Es aconsejable intentar aquellos marcados como VESA

```
 # Alternate 800x600@60Hz Non-Interlaced mode
 # Horizontal Sync = 37.9kHz
 (...)
# VESA 800x600@72Hz Non-Interlaced mode
# Horizontal Sync = 48kHz
# Timing: H=(1.12us, 2.40us, 1.28us) V=(0.77ms, 0.13ms, 0.48ms)
(...)
```

En este caso podemos escoger primero el segundo grupo de valores.

```
 # Alternate 800x600@60Hz Non-Interlaced mode
 # Horizontal Sync = 37.9kHz
 # Timing: H=(1.20us, 3.80us, 1.40us) V=(0.13ms, 0.053ms, 0.69ms)
 #
 # name        clock   horizontal timing          vertical ti-
ming          flags
 "800x600"     40      800 848 1000 1056     600  605  607  633

 # VESA 800x600@72Hz Non-Interlaced mode
 # Horizontal Sync = 48kHz
 # Timing: H=(1.12us, 2.40us, 1.28us) V=(0.77ms, 0.13ms, 0.48ms)
 #
 # name        clock   horizontal timing         vertical ti-
ming            flags
 "800x600"     50      800  856  976 1040    600  637  643  666  +hsync +vsync
```

Figura 4.4: Dos formas de definir el modo "800x600"

Los cuatro valores que se definen en la figura como horizontal timing y vertical timing indican al servidor cómo ha de hacer para visualizar el modo definido.

- Flags: Nos servirá para especificar parámetros opcionales en la definición. En este apartado podemos introducir los siguientes valores: "Interlace"para el modo entrelazado. "DoubleScan"cuando cada línea debe ser definida dos veces. "+HSync"y HSync"para ajustar la sincronía horizontal del monitor al igual que los parámetros "+VSync"y VSync"para la vertical. "Composite"para indicar que la sincronía es de este tipo en monitores que lo soporten. "+CSync"y CSync"sirven igualmente para concretar este apartado.

Sección Device.

Para la definición de la tarjeta gráfica instalada. En la sección anterior se ha tenido que hacer constante referencia a este elemento a pesar de no estar especificado hasta el momento. Esta sección es opcional.

- Identifier ≪cadena>": Al igual que en el caso anterior, el identificador asignado deberá coincidir con el colocado en la sección Screen posterior. Podemos tener varias tarjetas gráficas definidas.
- VendorName ≪cadena>": Nombre del fabricante (opcional)12.
- BoardName ≪cadena>": Nombre de la tarjeta gráfica (opcional).

- Chipset «tipo>": Especifica tipo de chip que configura la tarjeta. En muchos casos esta entrada no hace falta indicarla, pues el servidor puede probar el tipo de tarjeta instalada (opcional).
- Ramdac «tipo>": Tipo de RAMDAC usado por la tarjeta. Utilizado solo por algunos servidores y en otros casos no hace falta indicarlo (opcional).
- DacSpeed <valor>: Velocidad del RAMDAC, indicado en Mhz. Utilizado por un numero muy reducido de servidores.
- Clocks <valor1> <valor2> ...... : Quizá el campo más importante de esta sección, aquí especificaremos los valores para los pulsos de reloj que utiliza la tarjeta gráfica. Para el ejemplo los extraeremos del fichero creado al principio (salida.tmp1) y los colocaremos en esta línea sin alterar su orden ni intentar el redondeo de alguno de ellos, hemos de trasladar estos valores tal cual.

  De no colocar esta línea (también puede tomarse como opcional), será el servidor el que los calcule cada vez que arranque o basándose en la información de la entrada siguiente, utilizadas para tarjetas de tipo programable.
- Clockchip «tipo>": Especifica el tipo de chip que incorpora la tarjeta gráfica para las que lo tienen del tipo programable. Sólo algunos servidores soportan este tipo de tarjetas (opcional).
- ClockProg «comando>"[textclock]: Se ejecutará el comando señalado a fin de fijar el reloj de la tarjeta gráfica en lugar de utilizar el código incorporado al servidor. Este comando debe especificarse como una ruta en la estructura de ficheros disponible. La línea Clocks debe estar presente para especificar que valores estarán disponibles para el servidor.

  La palabra textclock indica al servidor que el comando debe ser ejecutado para restaurar el modo texto a la salida o en el cambio de terminales virtuales, debe coincidir con uno de los valores de la línea Clocks.

  El comando es ejecutado con el identificador real del usuario con la salida y la entrada estándar (stdin y stdout) hacia la salida gráfica. A este se le pasan dos argumentos, la frecuencia del reloj en Mhz y un índice a la entrada concreta en la entrada Clocks. La salida puede ser 0 (correcto) o un valor en el rango 1-254.

  El comando es ejecutado en la inicialización del primer modo configurado y en el cambio de resolución. Si el programa falla en su comienzo el servidor terminará. Si es durante un cambio de modo, este no es cambiado pero el servidor sigue ejecutándose.

  - Opción ≪cadena$_{d}e_{o}$*pciones*” : *Enelmanualdelservidortenemosinformacióndelasopcionessoportadasalahora*

- VideoRam <memoria>: Cantidad de RAM de vídeo disponible, medida en Kbytes. El servidor puede determinarla en el arranque (opcional).

- BIOSBase <dirección>: Nos sirve para especificar la dirección base de la parte de la BIOS encargada del manejo de la salida por pantalla con tarjetas VGA. Normalmente ésta es C0000 (hex.) (opcional).

- MemBase <dirección>: Indicará la dirección base de un buffer de la tarjeta gráfica destinado al trazado de líneas. Utilizado solo por algunos servidores (opcional).

- IOBase <dirección>: Especifica la dirección base de la BIOS para la entrada/salida. Utilizada solo por algunos servidores (opcional).

- DACBase <dirección>: La tarjeta gráfica VGA dispone de un DAC (Digital to Analogic Converter) como componente destinado a convertir la señal digital a una analógica entendible por en monitor. Con este campo indicamos la dirección de este componente (opcional).

- POSBase <dirección>: Para la dirección base del POS. Utilizado solo por algunos servidores (opcional).

- COPBase <dirección>: Dirección base del coprocesador numérico. Utilizado solo por algunos servidores (opcional).

- VGABase <dirección>: Dirección base de la memoria de la tarjeta gráfica VGA. Utilizada solo por algunos servidores (opcional).

- Instance <valor>: Indicaremos al servidor (el que lo utilice) si la tarjeta gráfica está integrada en la placa base o está colocada como una tarjeta de expansión más (opcional).

- Speedup ≪valor>": Detallaremos las velocidades habilitadas para el servidor. Utilizado solo por algunos servidores (opcional).

Los siguientes tres parámetros están indicados para el servidor XF86$_S$*3únicamente, hayquedirigirsealmanualdeésteservidorparamásdetalles* :

S3MNAdjust <M> <N>

S3MClk

S3RefClock

Sección Screen

Una vez introducidos el valor/es para los monitores y tarjetas gráficas que deseemos (recuerdo que se pueden definir varios en cada una de las secciones) es en esta sección en donde concretaremos que configuración ha de utilizar un servidor en particular. Podemos tener varias secciones de este tipo.

- Driver ≪cadena¿": Cada una de estas secciones debe comenzar con esta entrada y la ¡cadena¿ especificada debe ser única. El servidor a utilizar debe ser concretado en esta parte, podemos introducir: Accel, Mono, SVGA, VGA2 y VGA16.
- Device ≪cadena¿": Que dispositivo gráfico vamos a utilizar (¡debe estar definido anteriormente!).
- Monitor ≪cadena¿": Monitor a utilizar (también definido con anterioridad).
- ScreenNo ¡número¿: Para alterar el orden en que son consideradas las entradas Screen. Si no se concreta el orden es el de entrada de cada una de ellas (opcional).
- BlankTime ¡valor¿: Minutos que deben transcurrir el ordenador sin actividad para que se apague la pantalla. El valor por defecto es de 10 minutos. Se restablecerá de nuevo la pantalla si pulsamos cualquier tecla o movemos el ratón.
- SuspendTime ¡valor¿: Para monitores compatibles con el tipo VESA DPMS este valor (en minutos) especifica el tiempo de inactividad para pasar al estado 'suspendido'.
- OffTime ¡valor¿: Para los monitores anteriores, especifica el tiempo para pasar al estado 'off', el valor por defecto es de 30.

SuspendTime y OffTime son utilizadas sólo por algunos servidores.

Subsección Display

Nos encontramos ante una subsección incluida en la que estamos especificando para determinar ciertos valores, a pesar de la nomenclatura la sintaxis es igual a la de una sección (con las palabras Subsection Û EndSubsection).

Para algunos servidores y tarjetas gráficas esta subsección es obligatoria. Si tenemos varias secciones Display cada una debe identificarse por una entrada Depth única que después será utilizada en caso de definir, por ejemplo, más de un tipo de resolución en cuanto a colores disponibles (8, 16...).

- Depth ¡valor¿: Donde valor puede ser: 8, 15, 16 24 ó 32. De esta forma especificamos los 'bpp' (bits per pixel) que el servidor debe utilizar (si tiene esta capacidad) para la generación de colores y depende de la memoria disponible en la tarjeta gráfica. Por ejemplo para una resolución de "800x600"con una definición de 16 bits por pixel (65535 posibles colores) necesitamos $(800x600x16)/(8x1024) = 937.5$ Kb de memoria.

  En memoria dispondremos de esta forma de los bits especificados para referenciar un color concreto para cada pixel. El valor almacenado no es mas que un índice a un mapa de colores que indica al hardware como ha de dibujar cierto color. A mayor valor mayor número de colores podrán utilizar los clientes ejecutados.

  - Weight ¡valor¿: Para indicar el 'peso' de los tres colores básicos que forman la imagen (RBG) cuando el servidor trabaja en el modo 16bpp. Valores de referencia pueden ser 555 y 565 (opcional).
  - Virtual ¡valorx¿ ¡valory¿: Resolución de la pantalla virtual a usar. El primer valor debe ser múltiplo de 8 para servidores en color y de 32 para servidores monocromo. Si no existe este campo el servidor lo calculará de forma que se puedan representar todos los modos configurados. Algunos servidores no soportan esta capacidad (opcional).
  - ViewPort ¡x¿ ¡y¿: Determina la posición inicial de la esquina superior izquierda de la pantalla inicial. Sólo es necesario cuando la pantalla virtual es diferente de la resolución del modo inicial configurado. Si no se especifica se considerará la pantalla inicial centrada en el área virtual definida (opcional).
  - Modes ≪cadena¿"....: Obligatoria para algunos servidores. Indicaremos la lista de modos a utilizar concretados en la sección Monitor. Si algún modo no cumple ciertos requisitos en su especificación el servidor lo eliminará de los modos disponibles al ejecutarse.
  - InvertVCLK ≪cadena¿"[0||1]: Especifico del servidor XF86$_S3. Indicaelestadoinvertido/no-$ $invertidodelosmodos. Sielmodoes"*" seaplicaráatodoslosmodos. Ciertastarjetasgráficasnecesitanesteparámetr$

- EarlySC ≪cadena¿"[0||1]: Servidor XF86$_S3(opcional)$.
- BlankDelay ≪cadena¿≪valor1¿ ¡valor2¿: Para el servidor S3 (opcional).
- Visual ≪cadena¿": Esta entrada opcional indica el tipo por defecto de visualización para la pantalla principal (root). Para servidores de 8bpp tenemos los siguientes valores: StaticGray, GrayScale, StaticColor, TrueColor y DirectColor.

  Para servidores de 16bpp y 32bpp solo tenemos disponible el valor TrueColor.

  Para el servidor de 1bpp disponemos del valor StaticGray.

  Para servidores de 4bpp son: StaticGray, GrayScale, StaticColor, PseudoColor.
- Option ≪opciones¿": En el manual de cada uno de los servidores se especifican ciertos parámetros que pueden ser incluidos en esta línea (opcional).
- Black ¡red¿ ¡green¿ ¡blue¿: Permite especificar el color 'negro' en el manejador VGA2 del servidor XF86$_Mono.$
- White ¡red¿ ¡green¿ ¡blue¿: Para indicar el color 'blanco' como en caso anterior.

A fin de configurar ciertos aspectos disponemos además de dos programas que permiten configurar detalles como el gráfico a utilizar como puntero, si deseamos que se emitan ciertas señales sonoras durante nuestro trabajo, el aspecto de la pantalla principal (root), posibilidades de ahorro de energía, etc.. Dichos programas son xset y xsetroot.

Xset permite configurar el servidor desde la línea de comandos (o introduciendo los parámetros deseados en el fichero correspondiente), para conocer la configuración actual teclear 'xset q'.

Xsetroot lo utilizaremos si necesitamos configurar el aspecto del área de trabajo, permite escoger el tipo de ratón, el tipo de fondo a utilizar (podemos incluir el gráfico que deseemos), etc...

4.6 Cálculo de valores para la entrada ModeLine

A continuación se expondrá de forma detallada el mecanismo mediante el cual se obtiene la imagen que podemos ver en nuestro monitor durante una sesión de trabajo como base para entender todo el proceso descrito anteriormente para configurar XFree86, por último se describe paso a paso un ejemplo de cómo calcular los valores que van a describir una resolución a partir de las características técnicas de un monitor. El uso de la terminología anglosajona (siempre en cursiva) responde a un intento de claridad utilizando al máximo lo que puede encontrarse en cualquier documentación al respecto no obstante se intentará utilizar un término en castellano cuando sea posible.

Como en otras ocasiones formas de obtener un mismo resultado puede haber varias una vez conocida la relación entre los datos que se necesitan utilizar, puede basarse en este caso para calcular las entradas óptimas para su monitor. De no tener disponibles los datos necesarios para operaciones como las descritas en el ejemplo pueden utilizarse las frecuencias especificadas para el monitor a fin de escoger la entrada válida, siempre teniendo en cuenta si el monitor soporta los valores asignados por el modo escogido.

Puede describirse la forma de dibujar la pantalla del ordenador haciendo un símil con la escritura de un texto a mano o con un procesador de textos, la tarjeta gráfica emite una serie de señales que provocan que tres haces de electrones (uno por color: rojo, azul y verde) causen con su impacto en una zona sensible del tubo de imagen el que aparezca un punto luminoso, este fenómeno es repetido con una frecuencia muy alta, con el objetivo de mantener iluminado el máximo tiempo posible esta zona de la pantalla (pixel), de no ser así el punto se apagaría sin más.

Si aprovechamos este efecto para el trazado de líneas horizontales a en base a puntos contiguos ya tenemos el patrón mínimo que utilizaremos para cualquier dibujo. El trazado de las líneas se hace de izquierda a derecha de la pantalla, por lo que empezando por la esquina superior izquierda de la misma y siguiendo en la línea de abajo en sucesivas ocasiones tendremos como resultado el dibujo de una pantalla completa.

Debido a las limitaciones impuestas por los monitores el trazado de las líneas horizontales no comienza al principio de la pantalla ni finaliza en el extremo, se deja un margen a ambos lados debido a que el control de las señales en los extremos es difícil. Por el mismo motivo no son dibujadas todas las líneas para el trazado de una pantalla, de ahí que la imagen presente un marco alrededor.

Hacen falta por tanto una serie de señales para que este proceso esté perfectamente sincronizado que son generadas por la tarjeta gráfica mediante un circuito del tipo 6845 o similar, además dispone de memoria RAM y otros componentes que ayudan al trabajo en modo texto y gráfico. El 6845 genera los pulsos de reloj (dot clocks) que van a provocar el dibujado de los puntos en la pantalla, además la señal (horizontal sync) que provoca que se salte a la línea siguiente para trazar otra línea (horizontal retrace) y, una vez finalizado el trazado de una pantalla completa, (se ha llegado al extremo inferior derecho de la pantalla) se debe producir una (vertical sync) que provoque la vuelta al punto de partida de la pantalla (vertical retrace).

Existen asimismo una serie de tiempos de espera que hay que respetar antes y después de la generación de las señales descritas en el párrafo anterior, para desactivar y activar correctamente los haces de electrones durante estos intervalos, ya que no deseamos que se dibuje nada en la pantalla. Estos tiempos se denominan Front Porch para el instante antes de una señal y Back Porch para después.

Con todo lo expuesto resolveremos un modo para un monitor y una tarjeta gráfica imaginarios, suponemos un monitor con las siguientes características:

- Frecuencia Horizontal: 15-40 KHz.
- Frecuencia Vertical: 40-100 Hz.
- Ancho de Banda máximo: 80 MHz.
- Sincronismo Horizontal: 2 ms.
- Front Porch y Back Porch: 2 ms.
- Sincronismo Vertical: 0,08 ms.
- Front Porch: 0,02 ms.
- Back Porch: 0,6 ms.

Nuestro objetivo consiste en calcular la entrada ModeLine de tal forma que podamos trabajar en esta resolución, además de con una tasa de refresco (RR13) de la imagen lo más alta posible a fin de reducir drásticamente el parpadeo de la imagen en la pantalla que en ocasiones no es apreciable, pero puede perjudicarnos seriamente. El estándar VESA define una frecuencia de 72 Hz (72 imágenes/segundo) como valor ideal.

Con una Frecuencia Horizontal de 40 Khz. tenemos que se puede trazar una línea un tiempo de 25 ms $(1/40000)$ que con un tiempo total de espera de 6 ms. (Sincronismo Horizontal + Back Porch + Front Porch) nos deja con 19 ms. para el trazado de una línea.

Dividiendo los 19 ms. entre el número de puntos totales que queremos por línea, obtendremos el tiempo que se tarda en dibujar cada uno de ellos y de ahí la frecuencia a la que ha de trabajar la tarjeta gráfica, si quiere realizarlo correctamente (este será el valor que pondremos como dotclock).

$$\text{Tiempo por pixel} = \frac{19ms}{640puntos} = 29.6ns >> 30ns. \tag{4.1}$$

Por lo que necesitaremos que la tarjeta gráfica trabaje al menos a 33 Mhz. para poder obtener dicho resultado. Utilizaremos el valor más cercano posible si no disponemos de éste y siempre por debajo del ancho de banda máximo soportado por el monitor.

Todos los valores referidos al trazado horizontal (los cuatro primeros de la línea ModeLine sin contar la entrada dot clock) están especificados en puntos, por lo que necesitamos pasar los tiempos de espera a esta magnitud. También hay que señalar que todas estas entradas deben ser múltiplos de 8 debido al tipo de registros integrados en la tarjeta VGA y la forma en que el programa XFree86 los utiliza.

Con 2 ms. de tiempo para los valores de espera hemos de saber cuantos puede dibujar la tarjeta gráfica en ese tiempo:

$$Puntos = \frac{2ms}{33Mhz} = 66 \rightarrow 64 \text{(múltiplo de 8 más cercano)} \tag{4.2}$$

Ya podemos introducir en la línea los cinco primeros valores para la resolución deseada.

ModeLine ”640x480”33 640 (640+64) (640+64+64) (640+64+64+64)

Podemos ver que el primer valor se corresponde con el número de puntos que deseamos, el segundo con el punto de inicio del trazado de una línea, el tercero en punto en el que ha de finalizar y por último se refleja el número total de puntos dibujados (aunque no visibles, claro).

Los valores que definen el trazado vertical de la pantalla hemos de trasladarlos a líneas, en este caso no es necesario el redondeo a un valor múltiplo de 8. Para conseguir esto hemos de dividir los valores presentados entre el tiempo que se tarda en trazar una línea (25 ms) obtenido al comienzo del ejemplo, obtenemos:

- Sincronismo Vertical: 3 líneas.
- Front Porch: 0.8 líneas $>>$ 1 línea.
- Back Porch: 24 líneas.

Obtenemos así los siguientes valores verticales:

480 481 484 508

El primer valor indica las líneas que queremos, la segunda y la tercera en cual comienza y finaliza la señal de sincronismo y por último el número total de líneas trazadas.

Para los valores obtenidos podemos obtener la Frecuencia de Refresco de la pantalla a partir del siguiente cálculo:

$$RR = \frac{Dotclock}{Puntos \cdot Lineas(totales)} = \frac{33MHz}{832 \cdot 508} = 78Hz \tag{4.3}$$

Podemos ver que el valor conseguido está por encima del recomendado por el estándar mencionado.

A lo largo de toda la exposición se ha dado por entendido que no se desea utilizar ningún modo entrelazado, debido a que no se trata de un modo óptimo de trabajo con el entorno gráfico y se encuentra ya en desuso.

4.7 Disfrutando de Xfree86.

Una vez tenemos configurado el software se procederá a ejecutar XFree86, si se hace tecleando:

```
# X
```

Simplemente arranca el servidor, por lo que la pantalla se nos mostrará de color oscuro, aparecerá el símbolo del ratón y ... nada más. ¿Que se ha hecho mal?, la respuesta es NADA.

Acostumbrados a otros sistemas operativos se puede pensar que algo ha ido mal y que, por tanto, hay que volver a revisar el fichero XF86Config para ver que parámetro se ha introducido mal. Para atenernos a la terminología hay que decir que además del servidor hay que ejecutar algunos clientes básicos para podernos comunicar con el servidor correctamente, el principal en este caso es el gestor de ventanas.

Se ejecutará el entorno utilizando el comando 'startx', fichero que obtendrá los parámetros que deseamos pasar al servidor y a los clientes a ejecutar mediante el programa xinit que es el que realmente realiza la tarea de arrancar el software.

Startx busca la existencia de los ficheros .xinitrc y .xserverrc en el directorio del usuario que lo invoca, de no existir utilizará /usr/X11R6/lib/X11/xinit/xinitrc y /usr/X11R6/lib/X11/xinit/xserverrc respectivamente.

El fichero xserverrc generalmente no necesitaremos crearlo ya que por defecto se ejecuta el servidor denominado X (recuerde que con este valor ejecutaremos el escogido durante la instalación).

Caso de no existir ninguno de los ficheros xinitrc posibles xinit ejecutará:

```
# xterm -geometry +1+1 -n login -display :0
```

Para el caso en que deseemos especificar algunas opciones que el servidor se debe tener en cuenta a la hora de ejecutarse, se dispone de las siguientes (estas opciones son generales, para más información acudir al manual del servidor que estemos utilizando):

- vtXX: Con el número 'XX' concretaremos el terminal virtual (accesibles mediante las teclas F1...F12) que el servidor escogerá para su ejecución. Si no se concreta se escogerá el primero disponible, es decir, sin procesos pendientes a la espera de entrada de usuarios. Puede verse que terminal va a utilizarse en el fichero 'salida.tmp1' que creamos al comienzo, antes de crear el fichero XF86Config, en la línea '(using VT number 7)' .

- -probeonly: El servidor finalizará antes de pasar a ejecutarse. Accederá al fichero XF86Config para extraer cierta información, el resto la detectará. De ahí el comando descrito para la creación del fichero 'salida.tmp1'.

- -bpp ¡valor¿: Fija el valor para 'bits per pixel (bits utilizados para especificar cada punto de la imagen en pantalla). El valor por defecto es 8. No todos los servidores soportan el resto de valores posibles (15, 16, 24, 32).

- -weight ¡valor¿: Para el valor de los tres colores (RGB) en el modo 16bpp. Por defecto será 565.

- -gamma ¡valor¿: Corrección para el parámetro gamma, en el rango [0.1,10]. Por defecto es 1.0 para los valores R, G y B. No todos los servidores lo soportan.

- -rgamma ¡valor¿: Igual al caso anterior para el campo R (color rojo).
- -ggamma ¡valor¿: Igual al caso anterior para el campo G (color verde).
- -bgamma ¡valor¿: Para el campo B (color azul).
- -showconfig: Presenta una lista de 'drivers' gráficos configurados en el servidor.
- -verbose: Muestra la máxima información posible en el arranque.
- -xf86config ¡fichero¿: El fichero de configuración es distinto al XF86Config. Sólo se puede aprovechar esta posibilidad si se ejecuta el servidor como administrador (root).
- -keeptty: Útil para el testeo del servidor en su desarrollo.

En realidad existen varios ficheros xinitrc.¡nombre¿ en el directorio /usr/X11R6/lib/X11/xinit y un enlace al fichero xinitrc.fvwm (xinitrc). En el caso de ejemplo y que viene por defecto en la instalación se ejecuta el gestor de ventanas fvwm14.

Los ficheros que el fichero menciona inicialmente (.Xresources y .Xmodmap) son los que configuran el entorno a nivel de recursos15 y teclado. El programa que gestiona los recursos es el xrdb y para la redefinición del teclado es xmodmap.

En nuestro caso no existe inicialmente el fichero .Xresources por lo que el sistema tomará por defecto los valores especificados en el fichero .Xdefaults.

Podemos crear un fichero .xinitrc para cada usuario o modificar el xinitrc para que todas las sesiones que iniciemos de XFree86 sean afectadas por los mismos parámetros totalmente a nuestro gusto. Para ello veamos un ejemplo muy sencillo de fichero:

```
        #!/bin/sh
        # Un simple fichero .xinitrc
        # Iniciamos un terminal virtual (pueden ser más).
        color_xterm -geometry -20+10 -fg darkslategray -bg whi-
te &
        # Otros clientes
        oclock -geometry 70x70+5+5 &
        xload -geometry 85x60+85+5 &
        xsetroot -solid red &
        # Ejecutamos el gestor de ventanas
        exec fvwm
```

En este ejemplo solo se ha introducido cinco clientes: un terminal virtual, un sencillo reloj analógico, un monitor de la carga del sistema, para definir el color de la pantalla y por fin el gestor de ventanas. Señalar la importancia que tiene el que los clientes se ejecuten en segundo plano (indicado por el símbolo '&') o concurrentemente y el comando exec para el gestor de ventanas.

De no ejecutar los clientes en segundo plano nos veríamos en la situación de que el servidor los ejecutaría uno después de otro, en este caso el terminal virtual sería el primero en ejecutarse, al finalizar su ejecución daría paso al reloj y así sucesivamente.

Con el comando exec el gestor de ventanas sustituirá al proceso xinit que arrancó el servidor, de esta forma cuando salgamos del gestor de ventanas también finalizará éste.

Cada uno de los clientes tiene una serie de modificadores propios que nos ayudan a definir el tamaño de la ventana en el que va a ejecutarse, colores, posición... que hay que buscar en el manual propio a fin de poder configurarlo adecuadamente.

Por último mencionar la forma de finalizar la ejecución del servidor (y posteriormente del tándem servidor+gestor de ventanas), se puede hacer mediante las teclas: ¡Ctr¿+¡Alt¿+¡Backspace¿16 o con la opción adecuada del gestor de ventanas.

Como se puede observar hasta el momento tenemos varias posibilidades a la hora de configurar el comportamiento del entorno gráfico en general: el fichero XF86Config, ficheros xinitrc y xserverrc o directamente en la invocación del programa startx, por ejemplo:

```
# startx -- -bpp 16 -weight 555
```

En el apartado siguiente se verá como puede modificarse el aspecto del gestor de ventanas a utilizar además de una breve explicación del término recursos17 y como utilizar este mecanismo para cambiar el aspecto de los programas clientes que se vayan ejecutar y que permiten este sistema con lo que ya habremos repasado las principales formas de configurar el 'entorno gráfico' de LINUX.

4.8 Personalizar el entorno.

Hasta el momento hemos visto los valores que nos van a definir las condiciones en las que el servidor trabajará en relación con los clientes que podamos ejecutar. Se han configurado modos o resoluciones, directorios donde encontrar ciertos ficheros, frecuencias horizontales y verticales, pulsos de reloj de la tarjeta gráfica...

Todo esto es imprescindible para la ejecución del X Window en la máquina que vayamos a utilizar pero no hemos visto todavía la forma de aplicar nuestros gustos personales al trabajo que vayamos a realizar, la forma de especificar colores a utilizar, tipos de ventanas, forma del ratón, iconos, etc...

De esta tarea se encarga el gestor de ventanas que es un cliente al que hemos hecho constante referencia en apartados anteriores y vamos a concretar algo más sus características.

Existen muchos gestores implementados para el X Window, cada uno con unas características propias que lo distinguen totalmente de los demás en cuanto a apariencia del área de trabajo, uso de recursos del sistema, licencia de uso, forma de configurarlo,....

Una de las opciones que más llaman la atención sobre alguno de estos gestores de ventanas es el poder definir un área de trabajo virtual más grande que el espacio definido por nuestra pantalla, así podemos tener las ventanas fuera de los límites del área que puede verse a través de la pantalla repartidas por el área de trabajo a nuestro antojo.

Uno de los más extendidos, implementado específicamente para el sistema operativo LINUX es el fvwm. Este gestor de ventanas es el que viene definido por defecto en los ficheros de configuración, aunque la distribución Slackware contiene otros.

```
#!/bin/sh
# $XConsortium: startx.cpp,v 1.4 91/08/22 11:41:29 rws Exp $
# $XFree86: xc/programs/xinit/startx.cpp,v 3.0 1994/05/22 00:02:28 da-
wes Exp $
(...)
userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/usr/X11R6/lib/X11/xinit/xinitrc
sysserverrc=/usr/X11R6/lib/X11/xinit/xserverrc
clientargs=
serverargs=
if [ -f $userclientrc ]; then
    clientargs=$userclientrc
else if [ -f $sysclientrc ]; then
    clientargs=$sysclientrc
fi
fi if [ -f $userserverrc ]; then
    serverargs=$userserverrc
else if [ -f $sysserverrc ]; then
    serverargs=$sysserverrc
fi
fi
whoseargs="client"
while [ "x$1" != "x" ]; do
    case "$1" in
        /’’*|\.*)       if [ "$whoseargs" = "client" ]; then
                    clientargs="$1"
                else
                    serverargs="$1"
                fi ;;
        --)     whoseargs="server" ;;
        *)      if [ "$whoseargs" = "client" ]; then
                    clientargs="$clientargs $1"
                else
                    serverargs="$serverargs $1"
                fi ;;
    esac
    shift
done
xinit $clientargs -- $serverargs
```

Figura 4.5: Fichero startx

```
#!/bin/sh
# $XConsortium: xinitrc.cpp,v 1.4 91/08/22 11:41:34 rws Exp $

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap
# merge in defaults and keymaps
if [ -f $sysresources ]; then
    xrdb -merge $sysresources
fi

if [ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi

if [ -f $userresources ]; then
    xrdb -merge $userresources
fi

if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi
# start some nice programs
xsetroot -solid SteelBlue
fvwm
```

Figura 4.6: Fichero xinitrc.fvwm