

Derechos de Autor © 2003 Grupo de Usuarios de Linux de Ourense.

Permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU (FDL), Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation; con las Secciones Invariantes siendo PRÓLOGO, sin textos de Portada, y sin textos de la Cubierta Posterior. Una copia de la licencia es incluida en el Apéndice titulado "Licencia de Documentación Libre GNU".

ÍNDICE

Prólogo.....	7
1. Introducción e historia del sistema operativo Linux y el software libre. Juan A. Añel.....	11
1.1. El sistema operativo Linux.....	11
1.2. Historia del Proyecto GNU y la Free Software Foundation.....	17
2. Linux en la universidad y en la docencia. Una experiencia real. Leandro Rodríguez.....	19
2.1. Introducción.....	19
2.2. Mi experiencia personal.....	19
2.3. Ventajas y desventajas del uso de Linux.....	19
2.4. Linux en la investigación y en la docencia.....	21
2.5. Conclusiones.....	24
3. Aspectos generales de la arquitectura del núcleo Linux. David Olivieri.....	25
3.1. Introducción.....	25
3.2. Aspectos generales de los sistema Linux.....	25
3.3. Arquitectura del núcleo Linux.....	26
3.4. Conclusiones.....	30
4. Uso básico de Linux en formato live CD. David Fernández.....	31
4.1. El fenómeno live CD.....	31
4.2. Pasos previos al uso de una live CD.....	32
4.3. Comenzando con una live CD.....	32
4.4. Escritorio de una live CD.....	33
4.5. La consola.....	34
4.6. Persistencia de los datos en una live CD.....	34
4.7. Configuración de una live CD.....	35
4.8. Instalación de una live CD.....	36
4.9. Conclusiones.....	37
5. Instalación y configuración paso a paso de RED HAT 9.0. Santiago Vázquez.....	39
5.1. Introducción.....	39
5.2. Preparando espacio libre en el disco duro.....	40
5.3. Iniciando el instalador.....	40
5.4. Tipo de instalación.....	42
5.5. División del disco en particiones.....	43
5.6. Configuración del sistema.....	44
5.7. Selección de paquetes.....	47
5.8. Finalización de la instalación.....	48
6. Instalación y configuración paso a paso de Debian GNU/Linux. Manuel Pedreira.....	49
6.1. Introducción.....	49
6.2. ¿Qué es Debian GNU/Linux?.....	49
6.3. Instalación.....	50
6.4. Utilidad APT.....	62

7. Compatibilidad de Linux con otros sistemas operativos. Diego Pérez.....	67
7.1. Introducción.....	67
7.2. Accediendo a particiones Windows.....	67
7.3. Migración de paquetes y aplicaciones.....	68
7.4. SAMBA.....	70
7.5. Ejecutando aplicaciones de otros sistemas operativos.....	74
8. Programación básica en sistemas Linux. David Olivieri.....	77
8.1. Introducción.....	77
8.2. Programación para administración de sistemas.....	77
8.3. Conclusiones.....	88
9. Seguridad informática básica en sistemas GNU/Linux. Antonio Fernandes.....	89
9.1. Introducción.....	89
9.2. Passwords.....	89
9.3. Software del sistema.....	90
9.4. Ataques al software del sistema.....	91
9.5. Software de prevención y alerta.....	93
9.6. Seguridad en las comunicaciones.....	94
9.7. Recomendaciones para obtener un sistema seguro.....	95
10. Instalación de un servidor web. Instalación de un LAMP. Santiago Rodríguez.....	97
10.1. Introducción.....	97
10.2. Apache.....	97
10.3. MySQL.....	100
10.4. PHP.....	102
10.5. Instalación de una aplicación en PHP y MySQL: phpMyAdmin.....	104
10.6. Epílogo.....	104
11. Linux en ámbitos científicos: aplicaciones. Humberto Michinel.....	107
11.1. Introducción.....	107
11.2. SCIENTIFIC APPLICATIONS ON LINUX.....	107
11.3. Cálculo numérico: C/C++ y MATLAB.....	108
11.4. Gráficos: gnuplot, Xmgrace y MATLAB.....	110
11.5. Gráficos vectoriales: Xfig y Killustrator.....	114
11.6. Procesadores de texto.....	114
11.7. Tratamiento digital de imagen: Image-Magic y Gimp.....	115
11.8. Animaciones: mencoder y mplayer.....	115
Apéndice I: Estructura del árbol de directorios en Linux.....	121
Apéndice II: Manual básico de VIM.....	127
Apéndice III: El shell y comandos básicos en consola para Linux.....	131
Apéndice IV: Equivalencias Linux-Windows.....	135
Apéndice V: Acrónimos.....	153
Apéndice VI: Referencias bibliográficas.....	155
Apéndice VII: Licencia de Documentación Libre GNU.....	157

PRÓLOGO

El volumen que aquí se presenta es una realidad gracias a las contribuciones realizadas por los miembros del Grupo de Usuarios de Linux de Ourense (GULO), profesores de la Escuela Superior de Ingeniería Informática de Ourense, profesores de la Facultad de Ciencias de Ourense y la colaboración del Grupo de Programadores y Usuarios de Linux (GPUL) de la Universidad de la Coruña. Desde estas líneas quiero mostrar mi gratitud a todos ellos.

El fin de este manual es servir de introducción al sistema operativo Linux y la temática del Software Libre, en principio, para cualquier persona que guste de leerlo, pero principalmente para estudiantes universitarios, centrándose en los de la Universidad de Vigo.

Este manual no ambiciona ser el mejor o el más perfecto, ni siquiera carecer de errores, sólo trata de acercar el mundo de Linux y el Software Libre a cualquiera que muestre interés por aumentar sus conocimientos de informática con algo diferente a lo habitual.

El desarrollo del mismo se ha realizado basándose en los contenidos de las charlas y jornadas periódicamente organizadas por GULO sobre Linux y Software Libre, en temarios de asignaturas de la Escuela Superior de Ingeniería Informática de Ourense, teniendo en cuenta las necesidades informáticas para investigación de una Licenciatura en Física (por extensión totalmente válido para cualquier titulación de ciencias experimentales o ingeniería) y el software necesario para llevar todo esto a cabo.

Con este libro se entrega un disco compacto con “GPUL Live CD” que podría considerarse una versión de evaluación de un sistema Linux. Realizado por GPUL, contiene un sistema operativo Linux fácilmente utilizable simplemente con introducir el CD en la unidad lectora del ordenador y, aunque válido para trabajo diario, se trata más de una versión de pruebas para el lector, aunque no por ello ha de considerarse como un sistema incompleto, ya que este CD contiene todos los programas necesarios para estudiar la titulación de Ingeniería Informática en la Universidad de la Coruña.

Desde su aparición en Internet en torno a 1991, tras ser desarrollado por Linus Torvalds, el núcleo Linux ha ido mejorando y se ha convertido en un sistema mucho más potente que otros sistemas propietarios, los cuales son desarrollados con un único fin empresarial.

El negocio que ha surgido en torno al fenómeno Linux también ha sido notable. Muchas empresas han obtenido grandes beneficios simplemente regalando Linux y cobrando posteriormente por el servicio de apoyo técnico o vendiendo conjuntos de discos compactos con Linux y sus manuales, cuando hasta mediados de los años 90, en el mundo empresarial relacionado con la informática, el intentar hacer negocio con un programa o sistema dando gratis su código fuente y sin cobrar por su licencia era algo calificado como “locura absoluta e impensable”.

El hecho de tener la posibilidad de disponer de controlar totalmente el sistema, lo hace idóneo para temas relacionados con seguridad informática. Ésto lo están entendiendo en los últimos tiempos muchos países, los cuales han migrado toda su infraestructura informática a Linux, frente a la dependencia que mantenían anteriormente de, mayormente, el gigante del software propietario.

Otra cuestión que no se escapa a nadie es el tremendo ahorro en cuestión de licencias que supone el usarlo, ya no sólo para el usuario particular, sino que para la administración pública significa no tener que desembolsar millones de euros.

De total actualidad es el profundo debate que existe sobre la problemática de las patentes de software (ho hace mucho, el parlamento europeo ha votado la directiva sobre patentabilidad del software en Europa) y el concepto en sí de “Software Libre”. Ésto, que aparentemente es algo que no tiene mayor trascendencia, implica algo más allá del dinero que pueda haber en juego a la hora de utilizar un programa en concreto patentado por alguien o un programa que alguien nos vende, su fondo social es mucho más profundo. En una sociedad como la

nuestra, de comienzos del S.XXI y totalmente informatizada, el hecho de que tengamos una dependencia en este tema de unas pocas empresas que se mueven únicamente por intereses económicos, debería ser algo cuando menos preocupante.

La filosofía de Software Libre, cuyo fundador se puede considerar a Richard Mathew Stallman, nació en torno a 1984. Dicha filosofía gira en torno a la denominada licencia GPL (General Public License) y, a grandes rasgos, defiende el derecho del usuario final del programa informático a saber realmente qué es lo que está usando (es decir, que su programa no esté haciendo en su ordenador algo de lo que él no es consciente), a modificar el programa y a publicar nuevas versiones del mismo, siempre y cuando sea bajo las condiciones en que él accedió al original que modificó. De esta forma, uno no se encuentra con situaciones de dependencia, en las cuales, entre otras cosas, si a un usuario, empresa o estado, le hace falta una funcionalidad en su programa, no es necesario que espere a que la empresa que se lo ha vendido considere rentable económicamente proporcionar una nueva versión que la incluya, puede hacerla él mismo. De esta forma tampoco se está sujeto a la dictadura provocada porque una nueva versión de un programa no sea compatible con la anterior y tengamos que pagar de nuevo por él.

Este manual consta de once capítulos y siete apéndices. En los capítulos se muestra una introducción histórica, experiencias personales con Linux y razones por las cuales es una opción válida, la estructura y las entrañas de un sistema basado en Linux, cómo usar el CD que acompaña al manual de forma que sirva de introducción a todo este mundo, las instalaciones de dos sistemas basados en Linux muy extendidos, la compatibilidad de este sistema con otros existentes, fundamentos de programación en el mismo, y conceptos básicos de seguridad informática. Ya en un nivel más avanzado, se habla de la instalación de un servidor de páginas web y se pone de manifiesto que podemos hacer nuestro trabajo diario completo utilizando un sistema Linux. En los apéndices se presenta el árbol de directorios del sistema, se expone un mini-manual del uso de uno de los editores de texto por excelencia en Linux, una tabla de comandos que suele ser bastante útil cuando se comienza, un conjunto de equivalencias de programas con Windows, un listado de acrónimos, referencias bibliográficas y la licencia de documentación libre GNU.

De cara al lector podríamos quizás distinguir entre dos posibles modos de afrontar el manual: para los usuarios totalmente noveles lo recomendable sería comenzar con los capítulos 1, 2, 4, 5, 7 y los apéndices, probando así mismo la GPUL live CD en su ordenador; para el usuario más atrevido o con nociones, la lectura completa del manual es una buena opción.

A GULO le gustaría agradecer desde estas líneas el apoyo recibido por parte de la Universidad de Vigo, concretamente al Vicerrectorado de Extensión Universitaria proporcionándonos medios económicos para llevar a cabo nuestras actividades. De igual forma nos gustaría agradecer la colaboración ofrecida por la Facultad de Ciencias de Ourense y la Escuela Superior de Ingeniería Informática, quienes nos proporcionan los locales y los medios técnicos necesarios. Por último agradecer al Vicerrectorado de Planificación y Desarrollo del Campus de Ourense y la Diputación Provincial de Ourense las ayudas concedidas recientemente para la organización de un curso que trate temas similares a los de este manual en el Campus de Ourense.

Juan Antonio Añel Cabanelas

CAPÍTULO 1

Introducción e historia del sistema operativo Linux y el Software Libre

Juan Antonio Añel Cabanelas*

1.1. El sistema operativo Linux

1.1.1. Historia

A finales de la década de 1960, en Bell Laboratories de AT&T, en EE.UU., Ken Thompson y Dennis Ritchie desarrollaron un sistema operativo: UNIX. UNIX estaba escrito inicialmente en lenguaje ensamblador, pero tenía un problema, era específico para la máquina para la que había sido diseñado, es decir, no era portable a otros ordenadores distintos. Ésto a todas luces no era excesivamente práctico, así que Thompson y Ritchie se propusieron diseñar un lenguaje que minimizara el esfuerzo para portar el sistema operativo de una máquina a otra, naciendo de esta forma el lenguaje de programación C, lenguaje con el que fué reescrito UNIX, de forma que se convirtió en el primer sistema operativo portable.

En 1991, un estudiante de la Universidad de Helsinki, Linus Torvalds, se propuso escribir su propio sistema operativo. En principio trataba de escribir uno pequeño, originalmente concebido de una forma similar a Minix. Todo ello nació de una larga experiencia como programador de bajo nivel, ya que Linus ya había programado en ordenadores Commodore y lenguaje ensamblador anteriormente. En 1990, Linus asistió a un curso de UNIX en la Universidad de Helsinki, lugar donde estudiaba para obtener su título en “Ciencias de la Computación” (objetivo que logró en 1997). A lo largo de ese año, Linus se encontró con un gran problema ya que en su universidad sólo había 16 terminales, por lo cual tenía que hacer una larga cola para acceder a una. Fue entonces cuando se decidió por trabajar con un PC, algo a lo que en principio se resistía porque pensaba que no podría aprender nada con esa arquitectura y con MS-DOS.

En su asignatura de Sistemas Operativos se empleaba un libro de texto que contenía una guía para desarrollar Minix, una especie de UNIX muy básico que tenía por objetivo simplemente enseñar el diseño de un sistema operativo, pero que dos meses después de su puesta a disposición del público, en 1987, tenía un grupo de noticias de 40000 personas a nivel mundial y del que se solicitaban a cada rato nuevas capacidades.

A Linus en el grupo de noticias de Minix le propusieron que realizase una pequeña contribución trabajando para el proyecto, y poco a poco, cuando se dio cuenta, al cabo de aproximadamente un año, tenía lo que se podía considerar un núcleo de Linux útil (es decir, apto para su uso). Linux fue liberado desde los servidores de la Universidad de Helsinki bajo licencia GPL en torno a principios de 1992. Inicialmente, Linus no estaba muy seguro de utilizar esta licencia, pero finalmente lo convencieron de ello. Linux no fue el nombre elegido por Linus para

* 1. Grupo de Usuarios de Linux de Ourense.// 2. Área de Física de la Tierra . FT2DC. (Universidade de Vigo)

su sistema operativo; el nombre inicialmente elegido era Freax (free+freak+unix).



Logotipo de Linux

En enero de 1992 existían unos 100 usuarios de la versión 0.02 de Linux, que dieron información e hicieron críticas al sistema, sugiriendo soluciones para los errores encontrados. En marzo de 1994 se liberó la versión 1.0. A quien primero atrajo Linux fue a los hackers (entendamos por hacker su verdadera definición, que es la de una persona que muestra especial interés por aumentar sus conocimientos sobre algo, sea cual sea la disciplina, en este caso las ciencias de la computación), ya que Linux es distribuido gratuitamente a través de Internet y se ajusta a las necesidades de cada uno y al tipo de computadora. Por esta razón la Free Software Foundation (FSF) lo empleó para el proyecto GNU, del cual se hablará más adelante.

Las contribuciones a Linux llegaron enseguida, una de las más importantes fue la del proyecto Xfree86, que se encargó de desarrollar un entorno gráfico para Linux, ya que hasta el momento sólo funcionaba en modo texto a través de la línea de comandos.

En torno a 1994 fueron apareciendo los primeros proyectos organizados que permitían que usuarios menos experimentados pudieran utilizar Linux sin necesidad de tener grandes conocimientos de informática. Las primeras distribuciones que permitieron tener en un CD-ROM todas las herramientas necesarias para poder instalar el sistema operativo, sin necesidad de tener que acudir a Internet para conseguirlas o buscar ayuda de otros usuarios, fueron Slackware, Yggdrasil y Debian. Algunos de estos proyectos iniciales han prosperado, como por ejemplo Debian, y otros resisten, aún siendo algo minoritarios en cuanto a su uso, como Slackware.

Poco después comenzaron a aparecer los grupos locales, denominados LUGs (Linux Users Groups), con la función inicial de que sus miembros colaborasen entre sí. Posteriormente apareció la primera revista mundial dedicada explícitamente al fenómeno Linux, "Linux Journal". Esta revista proporciona además distribuciones trimestrales en CD, lo cual permite a los usuarios sin conexión a Internet mantener actualizado su sistema.

Linux había sido desarrollado inicialmente para plataformas Intel, pero a partir de este momento se comenzó a migrar el sistema a otras distintas, arquitecturas de 64 bits, siendo Digital la primera empresa que apoyó el proyecto Linux al coordinar Linux el migrado del sistema a su plataforma Alpha.

A partir de este momento Linux comenzó a ser conocido mundialmente, con repercusión en las publicaciones del sector y, como no, con la explotación comercial del fenómeno. Surgieron las empresas que venden su propio conjunto de utilidades y software para instalar Linux, pero con una diferencia sustancial respecto a otras empresas del sector de la comercialización de software, su negocio principal no se encuentra en la venta de costosas licencias, ya que el sistema básico de empleo suele ser regalado. La base principal de ingresos de estas empresas se establece en torno al soporte técnico; la más famosa de todas es RED HAT, la cual desde un primer momento hizo más sencillo lo que hasta el momento era lo más difícil, la instalación del sistema, logrando pronto una apariencia gráfica de instalación similar a la de otros sistemas ya establecidos en el mercado, la cual resulta más agradable e intuitiva para el usuario sin conocimientos previos.

En torno a 1996 Linux toma mayor consistencia, aumenta la implantación de LUGs y aparece algo muy importante en su difusión, como son los grupos de traducción. Una de las grandes

ventajas de Linux es que en algún rincón del mundo ha habido alguien que se ha encargado de traducir a una determinada lengua las aplicaciones, el sistema y la documentación existente para el mismo. De este modo se puede instalar en multitud de idiomas, muchos de los cuales no existen para sistemas operativos comerciales desarrollados por empresas que se dedican al negocio del software, ¿por qué?, porque no resulta rentable traducir todo un sistema a un idioma que sólo van emplear unas pocas personas. Lo más cercano a nosotros es el proyecto TRASNO, proyecto que aunque está en declive se encarga de traducir al gallego todas las aplicaciones para Linux. LUCAS (LinUx en CASTellano) se encarga de la traducción al castellano.

Actualmente Linux sigue siendo desarrollado por Linus, pero el mayor autor de todos es Internet, ejerciendo Linus como coordinador, al cual se le conoce como “el dictador benévolo”.

1.1.2. ¿Qué hardware se necesita para utilizar Linux?

Linux es un sistema con un aprovechamiento de recursos de lo más eficiente, posee requerimientos de hardware mínimos, calificables de irrisorios en comparación con otros sistemas. Una configuración mínima para emplearlo podría ser una computadora 386SX con 1 MB de RAM y una disquetera. A mayores se pueden añadir componentes como el teclado, una placa de vídeo cualquiera y un monitor. Con esto es suficiente para arrancar y entrar al sistema. Si quisiéramos lujos, necesitaríamos un disco duro de unos 10 MB, que sería suficiente para disponer de todos los comandos y un par de aplicaciones pequeñas. Para un sistema más completo se aconsejan 4 MB de memoria RAM, o 8 MB si se tiene pensado utilizar una interfaz gráfica. Si se van a tener muchos usuarios y procesos conjuntamente, sería recomendable 16 MB; 32 MB ya sería para el caso de procesos muy pesados. En cuanto al disco duro, varían las necesidades bastante según la cantidad de aplicaciones que se haya pensado en instalar, pero se puede decir que actualmente van desde los 10 MB hasta un máximo de 4'7 GB para el caso de una distribución Debian GNU/Linux instalada con todas sus aplicaciones (en torno a unos 9000 programas distintos). Resulta obvio decir que cuanto más rápido sea el procesador y cuanta más memoria tengamos mejor rendimiento obtendremos.

Todo ésto es lo que más llama la atención a usuarios de otros sistemas no basados en UNIX. Un simple 486-DX2 con 8 MB de RAM basta para tener nuestro propio servidor de páginas web, obviando decir que en este caso otros sistemas operativos con capacidad de servidor, ya no es que den un mal rendimiento, sino que son incapaces de instalarse en la computadora en cuestión.

Sin caer en el masoquismo una configuración realista para un servidor web con Linux, y en el que además vayamos a utilizar el modo gráfico, es la que a continuación se recomienda: Pentium 90, 16 MB de RAM y 500 MB de disco duro.

1.1.3. ¿Por qué Linux en vez de otro sistema operativo?

A continuación esgrimiremos varias de las razones que pueden llevarnos a decidarnos por Linux en vez de por otro sistema operativo:

- Linux tiene dos ventajas evidentes y que quizás son las que primero llaman la atención, podemos conseguirlo de manera gratuita, es decir, podemos conseguirlo sin pagar absolutamente nada (o pagando si lo que queremos es una distribución comercial con soporte técnico, etc.), siendo además libre, lo que quiere decir que podemos hacer con él lo que queramos (modificarlo y ajustarlo a nuestras necesidades, evaluar sus bondades y defectos, etc.).
- Si lo comparamos con otras versiones de UNIX para PC, podremos observar que su velocidad y fiabilidad son muy superiores.

- Multitarea real y multiusuario: a diferencia de otros sistemas operativos, Linux sí es capaz de realizar varias tareas a la vez sin que interfieran en el rendimiento obtenido y con múltiples usuarios trabajando al mismo tiempo.
- Linux aprovecha mejor los recursos de nuestra computadora. De hecho se puede decir que si llegamos a compilar nuestro propio kernel, tendremos el mejor sistema operativo posible para nuestra computadora, ya que estará hecho a la medida de nuestro ordenador y no para otro cualquiera. Decir que, hoy por hoy, con las últimas versiones del kernel, la compilación ya casi es innecesaria para alcanzar ese nivel óptimo de rendimiento del sistema.
- Sus requisitos de hardware son irrisorios en comparación con otros sistemas.
- Linux tiene una gran capacidad en cuanto a conectividad a redes.
- Es abierto, es decir, conocemos el código fuente y la forma en que se ha programado, por lo tanto lo controlamos, en vez de controlarnos él a nosotros, ya que no existen tareas ocultas que desconozcamos.
- Es infinitamente más seguro que otros sistemas, mucho menos vulnerable frente a ataques directos y menos propicio a su infección por parte de virus, troyanos, gusanos, etc.

También rebatiremos algunos de los argumentos empleados usualmente para no utilizar Linux aún viendo las enormes ventajas que se muestran en la lista anterior:

- Costes de migración desde otro sistema. Estos costes existen pero podríamos compararlo con los costes de actualización que debemos asumir cada vez que sale una aplicación no compatible para otro sistema o una versión nueva del mismo.
- No tengo los conocimientos suficientes. Esta situación ha sido rebatida de la mejor forma posible, demostrando que es falsa. La Junta de Extremadura desarrolló su propia versión de Linux, llamada LinEx, migrando todos sus sistemas a ella. Los funcionarios de la administración extremeña la han aceptado y trabajan a diario sin problemas a nivel usuario, de igual forma que lo hacían antes.
- Linux está bien para pruebas pero no para cosas serias: Linux es empleado por Google, Amazon, Oracle, Dell y muchas otras grandes empresas. Películas como Titanic, El Señor de los Anillos, Shrek o el Episodio II de La Guerra de las Galaxias han sido realizadas utilizando Linux.
- Si es gratis no puede ser tan bueno como otras alternativas que cuestan más dinero y tienen detrás a una gran empresa: en realidad el número de defectos por líneas de código del Software Libre es mucho menor que el del software propietario. En algunos estudios sobre la pila TCP/IP se afirma que el software comercial tiene 0'55 defectos por cada mil líneas de código frente a Linux que tiene 0'1.

1.1.4. Historia de algunas distribuciones destacadas en la actualidad

A continuación hablaremos de distribuciones de Linux. Nos vamos a centrar particularmente en Debian por dos cuestiones: 1) es un proyecto no comercial, colaborativo y totalmente gratuito y libre y 2) su estructura de desarrollo y actualización le confiere una fiabilidad y estabilidad de la cual, hoy por hoy, el resto de distribuciones se encuentra muy lejos, mientras que los sistemas comerciales están a años luz.

El proyecto Debian: el proyecto Debian lo forman voluntarios a lo largo de todo el mundo, los cuales se esfuerzan para conseguir una distribución que esté compuesta, única y exclusivamente, por Software Libre. El producto principal de este proyecto es la distribución

Debian GNU/Linux, que incluye el núcleo del sistema operativo Linux y miles de aplicaciones preempaquetadas. Da soporte para varios tipos de arquitecturas: Intel i386 y superiores, Alpha, ARM, Motorola 68k, MIPS, PowerPC, Sparc y UltraSparc, HP PA-RISC, IBM S/390 e Hitachi SuperH.

Este proyecto fue fundado oficialmente por Ian Murdock el 16 de Agosto de 1993. Por aquel entonces el concepto de una “distribución” de Linux era totalmente nuevo. La creación de Debian fue apoyada por el proyecto GNU de la Free Software Foundation durante un año (noviembre de 1994 a noviembre de 1995).

Debian es la única distribución que es completamente abierta para que todos los usuarios y desarrolladores contribuyan con su trabajo. A diferencia de otras, Debian no es una entidad comercial, se trata solamente de un gran proyecto de colaboración a través de Internet.

Debian divide sus ramas de desarrollo en tres partes: estable, testing e inestable. Estable es la rama “estable”, valga la redundancia, es decir, provee un sistema operativo y un conjunto de paquetes totalmente fiable y sin ningún tipo de fallo. Para ello cualquier paquete o “programa” candidato a la rama “estable” pasa por un período de pruebas en el cual se dice que está “frozen” (es decir, congelado). Durante este período no se hace ninguna modificación sobre el mismo, pero sigue siendo utilizado. Esta fase suele ser de varios meses, asegurándose que a lo largo de todo el mundo nadie informa, en varios meses de uso, de ningún problema de ningún tipo con ese programa. Si supera esta prueba, el paquete pasa a ser estable. La siguiente rama es “testing”, paquetes a prueba que se encuentran cerca de pasar a frozen o en pruebas muy cercanas, susceptibles de sufrir problemas, pero que no serán fallos críticos del sistema operativo. “Inestable” es una rama empleada principalmente por desarrolladores del sistema operativo. Un sistema inestable sólo suele ser instalado por gente que trabaja con él para encontrar fallos en paquetes y corregirlos, o avisar de ellos al proyecto. Más información en <http://www.debian.org>

Existen empresas que desarrollan su propia distribución basándose en Debian GNU/Linux, entre ellas se pueden tener en cuenta Libranet (<http://www.libranet.com>) y Esware (<http://www.esware.com>). Ambas disponen de una versión base que es gratuita y de una versión ampliada de pago.

RED HAT Linux: en octubre de 1994 Marc Ewing liberó la que, hoy por hoy, es posiblemente la distribución comercial Linux más famosa en todo el mundo: “Red Hat Linux”. En 1995 Bob Young, una persona que ya había tenido contacto con Linux y el Grupo de Usuarios de Linux de New York, compró el pequeño negocio de Ewing y lo fusionó con ACC Corporation (una empresa que se dedicaba a la publicación y a la venta de accesorios informáticos, revistas y libros relacionados con UNIX) y llamó a la empresa resultante Red Hat Software. En este año se liberó la versión 2.0 de Red Hat Linux. A partir de ahí, Red Hat Software ha ido expandiéndose de manera imparable y en distintas facetas del sector: da soporte técnico telefónico, y hasta no hace mucho regalaba su distribución Linux básica. Red Hat vende versiones más completas de su distribución, sirve a grandes compañías (entre sus principales clientes se encuentra la productora cinematográfica Dreamworks) y sobre todo realiza su mayor negocio dando soporte técnico completo para sus sistemas Linux. Red Hat y Bob Young han recibido gran cantidad de premios y reconocimientos y las grandes empresas del sector informático (IBM, Dell, HP, Oracle, BMC y VERITAS) dan soporte para Red Hat Linux.

Ya no como distribución Linux sino como modelo de empresa basado en Software Libre, decir como ejemplo que en junio de 2002 Red Hat Software cerró por primera vez en bolsa por encima del gigante del sector informático Sun Microsystems.

Recientemente Red Hat ha abandonado su distribución gratuita para destinar sus esfuerzos a su versión para empresas, pero sigue apoyando activamente una versión basada en su distribución mediante el denominado Proyecto Fedora.

Aurox es también una distribución basada en Red Hat que últimamente se encuentra en

auge.

Se puede obtener más información en:

<http://www.redhat.es>

<http://fedora.redhat.com>

<http://www.aurox.org>

SUSE LINUX: fue fundada en Alemania en 1992 por estudiantes universitarios con el nombre de Gesellschaft für Software- und System-Entwicklung mbH, creándose en 1998 el holding SUSE LINUX AG. Cuenta actualmente con sedes en Alemania, Gran Bretaña, Italia, República Checa y Estados Unidos. Entre sus inversores se cuentan Compaq Computer Corp., Intel Corp. y Silicon Graphics Inc. (SGI). SUSE se ha convertido en uno de los proveedores a escala mundial del sistema operativo Linux. Socios corporativos actuales de SUSE son IBM y AMD. De igual forma que RED HAT, tiene su mayor negocio en los entornos corporativos, encontrándonos en sus últimas versiones con que no cuenta con una versión libre de su sistema operativo más allá de su versión de evaluación sin restricciones temporales de un sólo cd “Suse Live CD evaluation”. Según sus propias estadísticas, SUSE cuenta con más de 15 millones de usuarios en todo el mundo actualmente, proporcionando soluciones informáticas a empresas punteras, cada una en su campo: Lufthansa, Babcock-BSH, Miele, el Centro de Investigación Langley de la NASA o el Banco Mercantil de Venezuela; en el sector público instituciones como el Ministerio de Defensa británico (RAF) y la policía central escocesa utilizan SUSE LINUX.

No hace mucho SUSE LINUX ha sido adquirida por la empresa norteamericana NOVELL. Más información en: <http://www.suse.de/es>

Mandrake Linux: Mandrake nació de la colaboración de varias personas que se conocieron en Internet en torno a 1998 y que decidieron crear MandrakeSoft. Desde entonces Mandrake ha sido una referencia a nivel mundial en el mundo Linux con su distribución Linux-Mandrake. Su éxito se basa en un desarrollo de Linux para todos los usuarios, desde el principiante al experto. Mandrake cogió para su distribución desde un principio el sistema de empaquetado empleado por RED HAT, pero añadió muchas cosas en su distribución. Una excelente forma de instalación gráfica, que la hacía más accesible para el usuario novel, vino a solucionar uno de los grandes dolores de cabeza hasta el momento en el mundo Linux, el soporte para hardware (hasta su aparición la configuración de dispositivos de hardware algunas veces era muy difícil). En definitiva hizo Linux más atractivo al usuario acostumbrado a simplemente usar el ordenador para jugar o hacer sus documentos en la oficina o en casa sin preocuparse de mucho más. A todas estas personas es a quién llegó Mandrake, pero sin olvidar otro tipo de clientes, lo cual quedó demostrado con el premio a la mejor versión Linux para servidores otorgada en la LinuxWorld Expo de 1999. En Abril de 2000, recibió el premio Platinum por parte de la revista británica PC Answers y en septiembre de ese mismo año el Premio de Elección del Editor, por distribución cliente por parte de la americana American « Linux Magazine ». En los últimos tiempos MandrakeSoft parece haber tenido algunos problemas económicos pero como distribución sigue funcionando sin problemas. Más información en: <http://www.mandrake.com>

LinEx: es una distribución basada en Debian GNU/Linux, desarrollada para la Junta de Extremadura con el fin de dotar a toda su administración y su sistema educativo de un sistema operativo y aplicaciones libres, pero con un fin fundamental, el ahorro de dinero de los contribuyentes para que éste pueda ser destinado a otros fines. LinEx es actualmente un referente mundial en implantación de Linux en la administración pública. Como características principales tendremos su fácil instalación y uso intuitivo para el usuario novel. Más información en <http://www.linex.org>

TurboLinux: TurboLinux es una distribución Linux poco extendida en Europa. Su núcleo principal de usuarios se encuentra en Asia. TurboLinux fue fundada en 1995 y es una empresa que se dedica, con un modelo muy similar a Suse o Red Hat, a proporcionar soluciones Linux en el mundo de la informática. Es la distribución por excelencia en Japón y, aunque China dispone de su propia distribución denominada Red Flag Linux, en los últimos tiempos se ha implantado con mucha fuerza en ese país. Recientemente ha anunciado la liberación de su versión TurboLinux 10. Más información puede encontrarse en: <http://www.turbolinux.com>

1.2. Historia del Proyecto GNU y la Free Software Foundation

1.2.1. Historia

El proyecto GNU (<http://www.gnu.org>) no se puede entender sin Richard Mathew Stallman (rms). Richard Mathew Stallman empezó a trabajar en el Laboratorio de Inteligencia Artificial del MIT (Massachussets Institute of Technology) en torno a 1971. Su trabajo se desarrollaba en un grupo que usaba exclusivamente Software Libre, ya que por esta época hasta las grandes compañías lo distribuían. Los programadores, como Stallman, tenían la libertad de cooperar entre ellos y modificar el software que le era suministrado, mejorándolo y adaptándolo a sus necesidades. Todo esto comenzó a cambiar y, hacia la década de los 80 gran parte del software suministrado se había vuelto propietario, es decir, la empresa que lo suministraba no dejaba al usuario el código fuente, con lo cual no podía ser adaptado y evitaba que fuese mejorado por un simple hecho de cooperación, obligando al usuario a, si quería ver su software mejorado, adquirir una nueva versión a la empresa.

Todo esto llevo en 1983 a Stallman a concebir la Free Software Foundation (FSF) (<http://www.fsfeurope.org/index.es.html>) y, a partir de la misma, el denominado proyecto GNU, todo ello con el fin de volver a la fase en la que el software podía ser diseñado, mejorado e intercambiado sin estar sujetos a la dictadura de licencias y barreras legales impuesta por las grandes compañías. Debemos tener en cuenta que el término inglés “free” significa en este caso “libre”, no “gratis”; aunque ambas cosas suelen ir asociadas, es decir, el Software Libre es gratuito, lo que realmente quiere decir “free” en este caso, es que tenemos la libertad de modificarlo y ajustarlo a nuestras necesidades.



Richard Stallman durante una conferencia en el MIT

El proyecto GNU consiste en el desarrollo de un sistema operativo y un conjunto de aplicaciones totalmente libre y compatible con UNIX, de forma que si aparece una aplicación que no es libre, será desarrollada una similar pero libre, de forma que una computadora pueda estar equipada al 100 % con Software Libre y cumpla cualquier función (ello incluye el sistema operativo

y resto de aplicaciones necesarias).

En 1990 se habían escrito la mayor parte de las aplicaciones del sistema operativo, pero faltaba lo más importante: el kernel o núcleo. Por entonces apareció Linux, que era libre, de forma que combinando Linux con el resto del sistema GNU se llegó a la meta de tener un sistema operativo libre, el GNU basado en Linux.

1.2.2. Libertades

“Software Libre” se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, cambiar, estudiar y mejorar el software. De una manera más precisa se puede decir que para que un software se considere Software Libre, deben cumplirse las siguientes libertades:

- libertad 0: libertad de usar el programa con cualquier propósito.
- libertad 1: libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades.
- libertad 2: libertad de distribución de copias.
- libertad 3: libertad de mejorar el programa y hacer públicas las mejoras, de tal modo que la comunidad se beneficie.

1.2.3. ¿Por qué usar Software Libre?

Las razones para utilizar Software Libre son múltiples y variadas. En los siete puntos que siguen a continuación se exponen de una manera clara los más importantes:

- Está comprobado que es software de mayor calidad que el propietario (dispone del mayor soporte técnico del mundo). Ésto es una verdad irrefutable. Existen estudios rigurosos al respecto que lo demuestran (errores por líneas de código respecto al software propietario, número de personas trabajando en él, tiempo de respuesta a fallos de seguridad,...).
- Daría lugar a una mayor competencia, por lo tanto a mayor calidad. Ésto es una consecuencia directa de la veracidad de la primera razón expuesta.
- En Software Libre no hay situaciones “lock in” o de dependencia. Al ser el código libre nunca se depende de una empresa que lo posea, pudiendo en cualquier momento rescindir un contrato con la misma sin miedo a problemas en la gestión de nuestra empresa o a situaciones similares, ya que contratando a un programador podremos adecuar el software a nuestras necesidades.
- Inexistencia del usuario esclavo. Ésta es una consecuencia directa de la tercera, casi redundante.
- Por seguridad informática (accesos no deseados, puertas traseras, funcionalidades no documentadas). Está demostrado que el Software Libre es infinitamente más seguro que el propietario, y eso sin tener en cuenta los problemas con virus, troyanos,... del segundo.
- Estándares abiertos, no dependientes de una empresa. Ésta puede ser una razón más general de la cual son consecuencia la tercera y la cuarta.
- Documentado en multitud de idiomas. El Software Libre cuenta con el mayor equipo de traducción de documentación existente. Las empresas no traducen su documentación a ciertos idiomas porque no es rentable (quizá sólo por cuestiones de publicidad e imagen), mientras, los manuales de Software Libre están traducidos a idiomas impensables en otros casos, ya que siempre hay una persona que utiliza un programa y que habla cierto idioma que acaba traduciendo un manual para él a su lengua natal (él sólo o en colaboración con otros usuarios). Un ejemplo es el Proyecto TRASNO, que se ocupa de la traducción de Software Libre al gallego (<http://www.trasno.net>).

CAPÍTULO 2

Linux en la universidad y la docencia. Una experiencia real.

Leandro Rodríguez Liñares*

2.1.Introducción

En este documento comento mis experiencias como profesor de Universidad y usuario de Linux. He intentado que el enfoque, más que técnico, sea personal. Es decir, tras mi experiencia como usuario de Linux, comento cuáles son las ventajas que éste aporta a mi trabajo y cuáles son sus principales inconvenientes.

Para ello he estructurado el documento en tres partes principales. En primer lugar, comento brevemente mi experiencia, es decir, las condiciones en las que empecé a usar Linux y las distribuciones que he utilizado. A continuación, el siguiente apartado incluye las ventajas y desventajas que la utilización de Linux implica en mi trabajo. Dichas ventajas y desventajas se refieren a la utilización del sistema en general, no a la de una aplicación en particular. La siguiente sección incluye reflexiones sobre la utilidad de Linux para las dos vertientes fundamentales del trabajo del profesor universitario: la investigación y la docencia. Finalmente, acabo este documento con unas breves conclusiones.

2.2. Mi experiencia personal

Mi experiencia con Linux comenzó hace unos cuatro años, cuando decidí abandonar el uso de Windows'98 para pasar a utilizar Linux. Los motivos para realizar este cambio fueron la seguridad, la estabilidad y la disponibilidad de aplicaciones que Linux proporciona. En aquel momento comencé a utilizar Debian GNU/Linux, que posteriormente cambié por Mandrake Linux por su facilidad de instalación. En estos momentos utilizo de manera prácticamente exclusiva Fedora.

He de destacar que debido a mi situación personal me desplazo a menudo, por lo que hace ya tiempo que trabajo habitualmente con un ordenador portátil. El hecho de que el hardware de los portátiles sea más específico y con menor disponibilidad de controladores hace que la necesidad de conocimientos de administración del sistema sea mayor que en el caso de la utilización de ordenadores de sobremesa.

2.3.Ventajas y desventajas del uso de Linux

Existen en Internet numerosos sitios web de promoción del uso de Linux a todos los niveles en los cuales se encuentran enumeraciones de las ventajas que dicho sistema operativo aporta a un usuario final. Más difícil es, en cambio, encontrar información sobre las desventajas de Linux frente a otros sistemas que no esté influenciada por intereses comerciales.

* Área de Ciencias da Computación e Intelixencia Artificial. (Universidade de Vigo)

El enfoque adoptado en esta sección ha sido el de comentar las ventajas y desventajas que encuentro en mi uso personal del sistema Linux tras varios años de experiencia. Como ya se ha dicho anteriormente, las ventajas y desventajas aquí comentadas se refieren a un uso general del sistema, no a una aplicación en particular.

2.3.1. Seguridad y estabilidad

El entorno de trabajo habitual en el ámbito universitario es del de un ordenador conectado a una red local en la cual hay un servidor (normalmente en Linux) que ejerce de servidor de discos, cortafuegos, servidor web y de impresoras. Además, normalmente el profesorado universitario hace un uso intensivo del correo electrónico, tanto mediante mensajes personales, como mediante su suscripción a listas de distribución.

Vemos, por tanto, que el entorno de trabajo implica unos requerimientos de seguridad altos, tanto frente a intrusiones como frente a virus y gusanos. Éste es precisamente uno de los puntos donde Linux destaca claramente ya que provee herramientas que reducen en gran medida el riesgo de intrusión, mientras que el problema de los virus es prácticamente despreciable.

A cambio podemos comentar la desventaja de que, para que Linux sea seguro, tiene que estar bien configurado, lo cual a menudo no está al alcance del usuario final.

2.3.2. Disponibilidad de software

A nivel de la disponibilidad de software, básicamente dos son las grandes desventajas de Linux:

- No hay Microsoft Office para Linux: es muy habitual para el profesorado universitario el tener que rellenar documentos y formularios que son facilitados en formato electrónico, normalmente en formato Word. Este hecho, de dudosa legalidad, es un auténtico quebradero de cabeza para los usuarios de Linux. Recientemente han aparecido suites ofimáticas y programas capaces de abrir y grabar documentos de Office tales como OpenOffice, Abiword, Gnumeric, KOffice, etc. Aunque la calidad de estas aplicaciones es normalmente muy alta, la compatibilidad sólo está totalmente asegurada cuando el documento a importar es sencillo.
- Es habitual encontrar software y controladores exclusivamente desarrollados para Windows: muchos de los equipos que se utilizan en investigación son conectados de una forma u otra a algún sistema informático, que a menudo debe ejecutar alguna versión de Windows.

Con estas dos salvedades, la ventaja de Linux en el campo de la disponibilidad de software es inmensa. Las distribuciones más usuales incorporan cientos de programas que abarcan la mayoría de las necesidades, mientras que los usuarios de software propietario ven mucho más restringida la disponibilidad de software. Además, el ritmo de actualización de las aplicaciones en Linux sería impensable en las comerciales.

2.3.3. Configuración e instalación

Uno de los inconvenientes que a menudo se le achaca a Linux es su dificultad de instalación. Sobre este punto se puede hablar mucho y se pueden encontrar testimonios para todos los gustos, desde el "intenté instalar Linux una vez y no fui capaz de hacerlo funcionar" hasta el "hoy en día Linux se instala con cuatro toques de ratón".

Para intentar aclarar algo el panorama puedo contribuir a este punto con mi opinión: las distribuciones recientes y orientadas a usuario final (Mandrake Linux, Suse Linux, Fedora) son

sencillas de instalar y es muy fácil que el funcionamiento básico no dé problemas; las dificultades empiezan cuando algún componente de hardware no está bien soportado o algún componente software requiere una configuración específica no realizada de forma automática.

A cambio también se puede argumentar que la instalación de Windows hoy en día tampoco es trivial, y que para la mayoría de los usuarios la reinstalación es un problema. Además, no debemos olvidar que es mucho más probable que surja la necesidad de una reinstalación completa cuando se usa un sistema Windows que cuando se trabaja con Linux.

2.3.4. Portabilidad

Los principales problemas a nivel de portabilidad que tenemos los usuarios de Linux en la Universidad son dos: el Microsoft Office (del que ya se habló anteriormente) y los binarios ejecutables. Para reducir la incompatibilidad a nivel de binarios, dos son las estrategias principales que se plantean:

- Programar en lenguajes que aseguren la portabilidad, como por ejemplo, Java o lenguajes de script (perl, python...).
- En caso de que haya que programar en algún otro lenguaje, hacerlo siempre teniendo en mente que los programas deberían poder ser compilados en otras plataformas.

Hechas estas dos salvedades, podemos decir que a nivel de portabilidad, Linux, al estar basado fundamentalmente en estándares abiertos, sobrepasa ampliamente a sus rivales propietarios en este sentido, ya que la compatibilidad está asegurada con sistemas como UNIX, Solaris, FreeBSD o incluso MacOS-X.

2.4. Linux en la investigación y en la docencia

En esta sección se comentan cuatro de los usos de un sistema informático que he considerado básicos en el ámbito universitario tecnológico: la escritura de documentos científico-técnicos, la docencia, la programación y el acceso a Internet. Como ya se ha puntualizado, la información aquí incluida es fruto básicamente de mi experiencia personal.

2.4.1. Edición de textos: LaTeX

Uno de los puntos fuertes de Linux es su capacidad para composición de textos mediante el uso de LaTeX. El concepto de creación de documentos con LaTeX es completamente distinto a la forma de trabajar con otros procesadores de textos gráficos del tipo de Word.

En LaTeX, el usuario escribe un fichero fuente, con cualquier editor de textos, en el cual incluye una serie de comandos que especifican cómo debe ser presentado ese documento. Posteriormente, se realiza una compilación (como si de un lenguaje de programación se tratara) para dar lugar a un documento en formato postscript o pdf. Por ejemplo, la frase "La fórmula de Einstein es $E=mc^2$ " en el documento original LaTeX se incluirá como:

*La fórmula de *Einstein* es $E=mc^2$*

donde “\underline” es para subrayar, “\em” es para resaltar texto y los caracteres “\$” limitan las fórmulas matemáticas.

Por tanto, podemos ver que la forma de trabajar es poco visual y que requiere ciertos conocimientos de la gramática de LaTeX. ¿Cuáles son entonces sus ventajas?. Mi experiencia

personal es que sus puntos fuertes son los siguientes:

- Con un procesador de textos convencional se pasa tanto tiempo escribiendo como dando formato, mientras que con LaTeX solamente se escribe, y el formato es dado por el procesador. En la cabecera de un documento LaTeX se especifica el formato a utilizar; en función del formato elegido (hay formatos para gran cantidad de tipos de documentos) los comandos tendrán un efecto final u otro. En el ejemplo anterior, el comando “\em” se traduce por cursiva, pero en otro formato podría traducirse por negrita, subrayado o un tipo de letra distinto. Podemos ver por tanto, que en LaTeX la persona que escribe se abstrae del resultado final del documento. Esta abstracción llega al punto de que es el procesador el que pagina el documento y coloca las figuras, tablas y demás elementos en el sitio adecuado, gestiona todas las referencias cruzadas, etc. y lo mejor de todo es que siempre lo hace bien. Cualquier persona que haya trabajado con Word creando documentos de tamaño y complejidad considerables entiende las ventajas que esto aporta.
- En LaTeX las figuras se incluyen como postscript encapsulado. Este hecho tiene la desventaja evidente de que las figuras a incluir deben estar en dicho formato (lo cual no siempre es posible) pero es a su vez uno de los puntos fuertes del trabajo con LaTeX: el escalado de las figuras postscript no ocasiona ningún problema. Cualquiera que haya intentado cambiar el tamaño de alguna figura en formato no vectorial (gif, jpg...) sabe que la figura se deteriora de forma notable al hacerlo.
- LaTeX hace un manejo perfecto de las referencias bibliográficas. Las referencias estarán en un documento aparte, y solamente las referenciadas en el texto serán incluidas en el documento final. El formato (incluyendo el orden de las referencias) es realizado de forma automática, simplemente especificando el formato deseado al principio del documento.
- Como hemos visto, los documentos LaTeX son documentos de texto, escritos con cualquier editor disponible, por tanto la portabilidad en este caso es total: un documento escrito en una máquina o sistema operativo podrá ser abierto por cualquier otro editor en otra máquina o sistema operativo, y compilado sin problemas (si LaTeX está instalado en el sistema de destino, por supuesto). LaTeX está disponible para todas las plataformas imaginables.
- Hay una alta disponibilidad de formatos para la composición de documentos. Esto es, supongamos que se desea presentar un artículo a una revista o un congreso científico. Lo habitual es que, a través de Internet, la organización ponga disponible el fichero con el formato que se va a utilizar en la revista o para las actas del congreso. Lo único que tenemos que hacer es especificar que se va a utilizar dicho formato en nuestro documento, con lo cual los comandos del fichero LaTeX serán traducidos de la forma adecuada. Además, hay repositorios públicos donde hay gran cantidad de formatos disponibles para cualquier necesidad que podamos imaginar (cartas, faxes, exámenes....).
- LaTeX está orientado a la creación de textos científicos. La capacidad de creación de fórmulas matemáticas en LaTeX no es igualada por la de ningún procesador de textos gráfico.
- El resultado final es de altísima calidad. El formato por defecto de LaTeX da una calidad que puede ser utilizada directamente en la impresión de cualquier libro. Con otros procesadores de texto como Word, la "puesta a punto" del documento antes de su impresión representa una carga de trabajo considerable.

Hasta aquí hemos visto las ventajas de LaTeX. ¿Cuáles son pues las desventajas de su uso?. Personalmente creo que hay dos fundamentales:

- La curva de aprendizaje es lenta, ya que de primera impresión es "poco amigable". De todos modos, con total seguridad el tiempo gastado en el aprendizaje de LaTeX es recuperado a la

hora de componer textos frente a la posibilidad de usar otros procesadores de textos de tipo gráfico.

- LaTeX es muy sencillo cuando se hace un uso básico del mismo, pero en otro caso las cosas se complican. Cuando se utiliza un formato predefinido (normalmente siempre hay alguno que nos vale) el funcionamiento es sencillo, pero la modificación o adaptación de formatos es una tarea muy complicada.

Se deja fuera de este apartado comentar otra de las grandes aplicaciones para la composición de documentos: docbook. El motivo es que docbook está más orientado que LaTeX a la composición de manuales y documentación técnica, mientras que éste ofrece mayores funcionalidades para la composición de textos científicos.

2.4.2. Herramientas para la docencia

En este apartado comento brevemente aquellas herramientas cuya finalidad es proveer de funcionalidades que permitan la utilización del ordenador como ayuda a la docencia. En concreto, me refiero a los programas para la creación de presentaciones para la conexión de un ordenador a un cañón de vídeo.

En este campo, la referencia al PowerPoint de Microsoft es inevitable. En mi opinión esto viene provocado por dos factores: las completas funcionalidades que este programa proporciona y el hecho de que ha sido durante mucho tiempo prácticamente la única posibilidad disponible para esta tarea.

El paquete OpenOffice incorpora una herramienta de creación de presentaciones llamada Impress cuyo objetivo es dar a los usuarios de Linux las posibilidades que ofrece PowerPoint a los usuarios de Windows. Debo reconocer que mi contacto con esta herramienta ha sido muy escaso, y mientras que versiones precedentes daban la impresión de que era un producto al cual le faltaba madurar, las versiones más recientes ya son una alternativa válida.

Otra posibilidad, con un enfoque totalmente distinto, viene una vez más de la mano de LaTeX. Existen para LaTeX paquetes que permiten la creación de transparencias al igual que otros tipos de documentos. Uno de los más conocidos es “prosper”, con el cual se generan transparencias en formato pdf que serán visualizadas por alguna herramienta como puede ser el Acrobat Reader de Adobe. Si bien la potencia y la comodidad de esta herramienta son notables, presenta, a mi modo de ver, el inconveniente de que su utilización requiere un conocimiento de LaTeX algo más profundo que el necesario para la creación de documentos de texto.

2.4.3. Programación

La disponibilidad de herramientas en este campo es absolutamente enorme: cualquier lenguaje de programación que podamos imaginar tendrá su compilador o intérprete correspondiente en Linux, y además de forma fácilmente accesible y gratuita.

La única particularidad que hay que tener en cuenta en este apartado es la de los lenguajes orientados a la creación de interfaces gráficos. Es evidente, que un lenguaje orientado a la creación de programas gráficos en entornos Windows, como por ejemplo Visual Basic, no tendrá su equivalente en Linux, pues el lenguaje será fuertemente dependiente del entorno gráfico de trabajo.

2.4.4. Utilidades de red

Como se ha comentado anteriormente, el uso que se hace de Internet en el ámbito universitario es intensivo, y probablemente mayor que en el caso de usuarios domésticos. Para este

tipo de tareas el ámbito de aplicaciones incluidas en todas las distribuciones de Linux es enorme, y de mucha mejor calidad que sus equivalentes en los sistemas Windows.

Por motivos de extensión, se comentan en esta sección las dos funcionalidades básicas en Internet: la navegación y el correo electrónico.

- Navegadores: mi experiencia personal es que los navegadores principales de Linux (mozilla, konqueror o epiphany) son de mucha mejor calidad que el Microsoft Explorer, ya que incorporan funcionalidades básicas como la posibilidad de trabajar con pestañas o el bloqueo de ventanas emergentes que mejoran en gran medida la navegación.
- Correo electrónico: la disponibilidad de gestores de correo es inmensa, pero opino que los dos de los principales entornos gráficos (kmail y evolution para KDE y Gnome, respectivamente) son claramente superiores a su equivalente en plataformas Windows (Outlook Express), por no comentar nada de las vulnerabilidades que aparecen de manera constante para el Outlook. Además, en Linux hay disponibles herramientas de filtrado de correo no deseado (spam) de gran potencia como “spamassassin”. Se dejan fuera de este documento otras funcionalidades de acceso a Internet como mensajería instantánea, chats, etc. El motivo de ésto es mi desconocimiento de las mismas, ya que no son herramientas que utilice habitualmente.

2.5. Conclusiones

La conclusión principal es muy clara: estoy muy satisfecho de haber optado por Linux y no pienso cambiar a otro sistema operativo. Linux cubre todas mis necesidades básicas (acceso a Internet, composición de textos, programación y docencia) de forma completamente satisfactoria. Hay además otro aspecto que no debemos olvidar: la Universidad es un organismo público, uno de cuyos objetivos debe ser hacer un uso eficaz de los recursos de que dispone, ya que al fin y al cabo es financiada por el conjunto de la sociedad. Además, estoy profundamente convencido de que la utilización de Software Libre debe ser, al igual que en el resto de las administraciones, promocionada fundamentalmente por dos motivos: el ahorro económico y la potenciación de un sistema cuyo desarrollo favorece al conjunto de la sociedad y no solamente a alguna empresa en particular.

CAPÍTULO 3

Aspectos generales de la arquitectura del núcleo Linux

David Olivieri*

3.1. Introducción

En este capítulo, se explica la organización general del núcleo de Linux. Estamos interesados en la arquitectura a nivel de los subsistemas del núcleo, pero no en detalles de la implementación o algoritmos, es decir, estamos interesados en cómo el núcleo de Linux está organizado y cuáles son las partes más importantes.

¿Qué hace el núcleo?. Antes de explicar el subsistema, tenemos que entender por qué existe en el núcleo y qué papel tiene. Quizás lo más importante que podemos decir del núcleo de Linux es que presenta una interfaz virtual o abstracta a los procesos del usuario. Los programas y comandos del usuario se ejecutan sin necesidad de especificar detalles del hardware físico que está instalado en una computadora. De este modo, el núcleo de Linux abstrae todo el hardware en un interfaz virtual constante. El núcleo de Linux soporta múltiples tareas, de manera que sean transparentes a los procesos del usuario. Cada proceso puede actuar como si fuera el único proceso en la computadora, con uso exclusivo de la memoria central y de otros recursos de hardware. El núcleo es el responsable de mediar en el acceso a los recursos de hardware, de modo que cada proceso tenga el acceso que requiere, mientras que se mantiene la seguridad entre procesos.

La estructura de la arquitectura que tiene Linux, es debida en gran parte a las necesidades como gran proyecto de software, con múltiples desarrolladores dispersados por todo el mundo. Así, el sistema Linux tiene subsistemas que son modulares. Los subsistemas modulares que destacamos en este capítulo son: el sistema de ficheros, el sistema de red y las interfaces al hardware.

3.2. Aspectos generales de los sistemas Linux

El núcleo de Linux es una parte de un sistema más grande, y es inútil aislado. El núcleo de Linux es el gestor encargado de manejar y hacer posible el correcto funcionamiento entre todas las partes del sistema. Podemos dividir un sistema Linux en cuatro partes, analizando el papel que tiene el núcleo:

- 1) Aplicaciones de software de usuario: un conjunto de aplicaciones.
- 2) Servicios del sistema operativo: los servicios se consideran típicamente parte del sistema operativo (un sistema de ventanas, una shell del comando, etc.); la interfaz de programación al núcleo (herramienta y biblioteca del compilador) se incluye en este subsistema.
- 3) Núcleo de Linux: el núcleo abstrae y media el acceso a los recursos de hardware, incluyendo la CPU.

* Escola Superior de Enxeñería Informática de Ourense. (Universidade de Vigo)

4) **Controladores del hardware**: este subsistema abarca todos los dispositivos físicos posibles en una instalación de Linux; por ejemplo, la CPU, el hardware de la memoria, los discos duros y el hardware de la red.

En este esquema cada capa del subsistema puede comunicarse solamente con las capas que están inmediatamente adyacentes a ella. Además, la dependencia entre los subsistemas es tal, que las capas más altas dependen de capas más bajas, pero los subsistemas más cerca del fondo no dependen de capas más altas.

Esta estructura es interesante porque, si estamos interesados en escribir software para aplicaciones de usuarios que requieren comunicación a través de red, podemos depender de servicios del sistema operativo, que presentan una abstracción respecto del hardware del sistema. Por otro lado, si tenemos que escribir los drivers para un dispositivo hardware, el núcleo es una interfaz abstracta que es independiente del hardware.

3.3. Arquitectura del núcleo Linux

El núcleo Linux se compone de cinco subsistemas principales:

- 1) El **planificador de procesos (PP)**: responsable de controlar el acceso de procesos a la CPU. El planificador hace cumplir una política que asegure que los procesos tengan un acceso justo a la CPU, mientras que se asegura que las acciones necesarias del hardware sean realizadas por el núcleo al mismo tiempo.
- 2) El **gestor de la memoria (GM)**: permite que el proceso múltiple comparta con seguridad el sistema de la memoria central de la máquina. Además, el encargado de la memoria mantiene la memoria virtual, lo cual permite que Linux mantenga los procesos que utilizan más memoria de la disponible en el sistema.
- 3) El **sistema de ficheros virtual (VFS)**: abstrae los detalles de la variedad de dispositivos de hardware presentando una interfaz del archivo común a todos los dispositivos. Además, el VFS mantiene varios formatos del sistema de ficheros que son compatibles con otros sistemas operativos.
- 4) La **interfaz de red (RED)**: proporciona el acceso a varios estándares del establecimiento de una red y a una gran variedad de hardware de la red.
- 5) El **subsistema de comunicación entre procesos (IPC)**: soporta varios mecanismos para la comunicación del proceso-a-proceso sobre un solo sistema Linux.

A continuación vamos a detallar algunas aspectos de cada uno de estos subsistemas que componen Linux. Además de explicar cada subsistema por separado, son importantes las dependencias que tienen entre sí. Una dependencia obvia es cuando se produce una interrupción, por ejemplo. En este caso un subsistema suspenderá generalmente un proceso que esté esperando una operación de hardware para terminar, y reanudará el proceso cuando se acabe la operación. Ésto tiene implicaciones en muchos de los subsistemas y no simplemente en uno. Existen muchas otras dependencias que son menos obvias pero igualmente importantes, y debido a ellas, Linux (u otros sistemas operativos) crece en complicación.

En las secciones que siguen exploramos cómo podemos dividir la complejidad de cada subsistema en partes aún más pequeñas. A continuación, estudiaremos cómo los distintos subsistemas están conectados entre sí.

3.3.1. Arquitectura del planificador de procesos (PP)

El planificador de procesos mantiene un bloque de datos para cada proceso que se encuentra activo; los bloques almacenados en una lista encadenada llaman a la lista de la tarea; el planificador de procesos mantiene siempre un indicador que muestra qué proceso está activo actualmente. Además tiene las responsabilidades siguientes: (i) permitir que los procesos creen nuevas copias de sí mismos, (ii) determinar qué proceso tendrá acceso a la CPU y efectuar la transferencia entre los procesos corrientes, (iii) recibir las interrupciones y encaminarlas al subsistema apropiado del núcleo, (iv) enviar las señales a los procesos del usuario, (v) manejar el hardware del contador de tiempo y (vi) liberar los recursos del proceso cuando éste termina de ejecutarse.

El planificador de proceso es el subsistema más importante del núcleo de Linux. Su propósito es controlar el acceso a la CPU. Ésto incluye no solamente el acceso de los procesos del usuario, sino también de otros subsistemas del núcleo. Se puede dividir el planificador de procesos en cuatro bloques funcionales: (1) la interfaz de llamadas del sistema, (2) el módulo de planificación de política, (3) módulos de arquitectura específicos y (4) módulo de arquitectura independiente. Una descripción breve de cada uno se muestra a continuación:

- 1) La interfaz de llamadas de sistema: es responsable de la comunicación con servicios del sistema operativo.
- 2) El módulo de planificación de política: es responsable de juzgar qué proceso tendrá acceso a la CPU.
- 3) Los módulos específicos de arquitectura: se diseñan con una interfaz común para abstraer los detalles de cualquier arquitectura particular de la computadora. Estos módulos son responsables de comunicarse con una CPU para suspender y para reasumir un proceso. Estas operaciones requieren conocimiento de cuáles son las informaciones de registros y estados que tienen que ser preservados para cada proceso. También es responsable de ejecutar el código necesario para suspender o iniciar una operación.
- 4) El módulo independiente de arquitectura: se comunica con el módulo de política para determinar qué proceso ejecutará a continuación, llamando posteriormente al módulo específico de la arquitectura para reasumir el proceso apropiado. Además, este módulo, llama al gestor de memoria para asegurarse de que el hardware de memoria se ha restaurado correctamente para el proceso en cuestión.

Por cada proceso activo en el sistema, se almacena información importante para el mismo. La estructura que mantiene toda la información de un proceso activo está en “task_struct”, que es parte del subsistema de planificación de tareas. Cuando se crea un proceso, se crea esta estructura y se almacena: su identificador, información sobre su esquema de memoria, indicadores de los i-nodos que representan todos los ficheros abiertos y asociados con este proceso, información sobre conexiones de red abiertas (con sockets), etc.

Pero, como hemos visto anteriormente, el subsistema de planificación de procesos tiene muchas dependencias de otros subsistemas. Algunos ejemplos de las dependencias son los siguientes:

- 1) El planificador de procesos requiere al gestor de memoria que invoque el esquema de memoria cuando un proceso está planificado. Además, el planificador de procesos depende del subsistema del IPC para los algoritmos de semáforo.
- 2) Conexión con el gestor de memoria: para cada proceso que se crea, el gestor de memoria almacena un esquema de direcciones virtuales, en una base del pre-proceso, y también almacena la información adicional sobre páginas del proceso. Toda esta información se almacena en una

estructura de esquema de memoria, que a su vez está almacenada en el “task_struct”.

3) Conexión con el sistema de ficheros virtual: los procesos pueden compartir ficheros si ambos tienen el mismo fichero abierto. La manera en que se hace es guardando índices asociados con cada fichero. Estos índices se llaman i-nodos (o índices-nodo) que el sistema de ficheros virtual utiliza para representar los ficheros lógicos. La estructura de datos de estos i-nodos contiene un esquema del número de bloques del archivo a las direcciones del dispositivo físico.

Estas dependencias se expresan con llamadas y el acceso de la función a la estructura de datos compartida de la lista de la tarea.

3.3.2. Arquitectura del Gestor de Memoria

El gestor de memoria es responsable de controlar el acceso de procesos a los recursos de la memoria del hardware. El mecanismo que utiliza este subsistema es disponer de un esquema entre referencias, de memoria de procesos a memoria física de la máquina. Como se dijo anteriormente, este esquema se realiza para cada proceso activo, para que dos procesos puedan acceder a la misma dirección de memoria virtual, pero utilizando direcciones de memoria física distintas. Además de este esquema, almacena datos adicionales en el bloque de los datos: por ejemplo cómo traer y almacenar las páginas, información sobre permisos de seguridad, etc.

El sistema tiene soporte para swapping, que es el mecanismo que mueve páginas de memoria a disco para dejar el soporte de memoria virtual.

Como para el subsistema planificador de procesos, dividimos el gestor de memoria en tres módulos funcionales: el módulo específico de arquitectura, el módulo independiente de arquitectura y un interfaz de llamadas de sistema:

- 1) El módulo específico de la arquitectura (architecture specific) presenta una interfaz virtual al hardware de gestión de la memoria.
- 2) El encargado independiente de la arquitectura (architecture independent) realiza todo el pre-proceso de trazado y el intercambio de la memoria virtual. Este módulo es responsable de determinar qué páginas de la memoria serán desahuciadas cuando haya un fallo de página -- no hay módulo separado de la política puesto que no se espera que esta política necesite cambiar.
- 3) Se proporciona una interfaz de llamada del sistema (system call interface) para permitir el acceso restringido a los procesos del usuario. Esta interfaz permite que los procesos del usuario asignen y liberen almacenamiento, y también que realicen el archivo de trazado de memoria I/O.

Ejemplos sencillos de algunas dependencias del gestor de memoria y el resto de subsistemas son:

1) Flujo de datos a hardware de memoria: consideremos el ejemplo de un fallo de página, que puede ocurrir si no se encuentra un bloque de memoria en el cache. El gestor de memoria recibe una notificación del hardware cuando ocurre un fallo de página, por lo tanto debe tener una conexión bidireccional con el hardware.

1) Flujo de datos con el sistema de ficheros y el planificador de procesos: en el caso de swapping (intercambio), el gestor de memoria tiene que hacer llamadas al sistema de ficheros para almacenar y coger páginas de memoria de disco. Debido a que tarda un tiempo finito para el intercambio de memoria con el sistema de ficheros, el gestor de memoria debe comunicarse con el subsistema de procesos para suspender un proceso hasta que el intercambio de memoria tenga lugar.

3.3.3. Arquitectura del Sistema de Ficheros Virtual

El sistema de ficheros virtual de Linux tiene las siguientes características: soporte para muchos dispositivos de hardware, soporte para muchos tipos de sistemas, soporte para muchos tipos de ejecutables, homogeneidad de interfaz, rendimiento con accesos de alta velocidad, mecanismos de protección, seguridad con permisos con sistemas de cuotas. El sistema de ficheros virtual abstrae los detalles del dispositivo físico y del sistema de ficheros lógico, permitiendo que los procesos del usuario tengan acceso a archivos usando una interfaz común, sin tener la necesidad de conocer en qué sistema físico o lógico reside el archivo.

Anteriormente hemos dicho que se representan todos los archivos usando i-nodos. Algunas las características de i-nodos son:

- 1) Cada estructura del i-nodo contiene la información de la localización para especificar en qué parte del dispositivo físico están los bloques del archivo.
- 2) El i-nodo almacena punteros a las rutinas en el módulo del sistema de ficheros lógicos, y en el driver de dispositivo que hace las operaciones de lectura/escritura.
- 3) La estructura de punteros es una manera para generalizar el uso de ficheros.

Podemos descomponer el VFS en cinco módulos funcionales:

- 1) El driver de dispositivo para cada controlador soportado del hardware: puesto que hay una gran cantidad de dispositivos de hardware incompatibles, hay una gran cantidad de drivers de dispositivo. La extensión más común de un sistema Linux es la adición de un driver de dispositivo nuevo.
- 2) La interfaz independiente del dispositivo proporciona una vista constante de todos los dispositivos.
- 3) El sistema lógico de ficheros para cada sistema de ficheros soportado.
- 4) La interfaz independiente del sistema presenta un hardware y una visión lógica archivo-sistema independiente de los recursos de hardware. Este módulo presenta todos los recursos usando una interfaz “block-oriented” o “character-oriented” del archivo.
- 5) La interfaz de la llamada de sistema proporciona el acceso controlado al sistema de ficheros para los procesos del usuario. El sistema de ficheros virtual exporta solamente funcionalidades específicas a los procesos del usuario.

Algunos ejemplos de cómo el VFS esta conectado con otros subsistemas:

- 1) Swapping: como se ha indicado, es una de las principales actividades del VFS. El gestor de memoria utiliza el sistema de ficheros virtual para intercambiar memoria con el disco (swapping) y para entrada/salida con esquema de memoria.
- 2) Interrupción: como en el caso del swapping o cualquier acceso a un dispositivo hardware de almacenamiento, el sistema de ficheros virtual utiliza el planificador de procesos para inhabilitar procesos mientras espera peticiones de finalización del hardware, también se ocupa de reasumirlas una vez que se haya terminado la petición.

3.3.4. Arquitectura de la interfaz de red.

El subsistema de la red permite que los sistemas Linux se conecten con otros sistemas sobre una red. Existe un número de dispositivos de hardware que hacen posible el soporte para que un número determinado de protocolos de red pueda ser utilizado. El subsistema de red abstrae estos

detalles de la puesta en práctica, de modo que los procesos del usuario y otros subsistemas del núcleo pueden tener acceso a la red sin necesariamente saber qué dispositivos o protocolos físicos se están utilizando.

Los bloques funcionales del subsistema de interfaz de red se pueden descomponer en cinco partes independientes:

- 1) Los drivers de dispositivo de la red se comunican con los dispositivos de hardware. Hay un módulo del driver de dispositivo para cada dispositivo de hardware posible.
- 2) El módulo de interfaz independiente del dispositivo proporciona una vista constante de todos los dispositivos de hardware, de modo que niveles más altos en el subsistema no necesitan un conocimiento específico del hardware en uso.
- 3) Los módulos del protocolo de red son responsables de poner en ejecución cada uno de los protocolos posibles del transporte de red.
- 4) El módulo de interfaz independiente del protocolo proporciona una interfaz que sea independiente de los dispositivos de hardware y del protocolo de red. Éste es el módulo de interfaz que es utilizado por otros subsistemas del núcleo para tener acceso a la red sin tener una dependencia de protocolos particulares o hardware.
- 5) Interfaz de llamadas de sistema

Cada objeto de red se encuentra representado por un “socket”. Los sockets están asociados con procesos de la misma manera que los i-nodos, ya que se pueden compartir los sockets entre sí, si están abiertos, de manera que las dos estructuras de “task-data” están apuntando a la misma estructura de “socket-data”. Éste es otro ejemplo de la transparencia de la arquitectura de Linux.

Como anteriormente, mostramos ejemplos de la relación de este subsistema con otros subsistemas.

- 1) Consideremos una interrupción creada con una interfaz física de red. El subsistema de red utiliza el planificador de procesos para suspender y restablecer procesos, mientras espera solicitudes de hardware para finalizar. Ésto demuestra una dependencia entre el subsistema de red y el planificador de procesos, y por lo tanto un flujo de control y datos entre ambos subsistemas.
- 2) Consideremos que estamos montando un sistema de ficheros remoto en red con NFS. El subsistema de red tiene una dependencia con el sistema de ficheros virtual. Ésto también indica un flujo de control y datos entre los dos subsistemas, en este caso con el VFS y el RED.

3.4. Conclusiones

Cuando crece el tamaño de los sistemas de software, la complejidad de la estructura de datos, las interconexiones, y la multitud de algoritmos en la implementación constituye una barrera para aprender su funcionamiento, representando demasiada complejidad para el experto. Si organizamos el sistema en una arquitectura de bloques funcionales, podemos comprender el funcionamiento del sistema mejor.

El núcleo Linux es un sistema complejo que consta de muchas partes interconectadas.

Estudiando la arquitectura funcional de alto nivel, podemos entender por qué se estructura de esta manera, lo cual nos ayudará en los detalles de la implementación.

CAPÍTULO 4

Uso básico de Linux en formato live CD

David Fernández Vaamonde*

4.1. El fenómeno live CD

Que el fenómeno Linux es una realidad en la sociedad de la información hoy en día es un hecho, pero también son un hecho los miedos que Linux todavía suscita en el gran público. Linux se ha creado grandes famas y grandes miedos: mucha gente ha dicho que la dificultad de instalación y uso del sistema es muy superior a la del uso de otros sistemas operativos propietarios, que las interfaces con el usuario son hoscas y austeras y que no contribuyen a la utilización del sistema.

Sin embargo, en los últimos años muchos de estos miedos se han desterrado fruto de dos pilares básicos.

Por una parte las mejoras en los entornos de escritorio que hacen que Linux presente una interfaz mucho más amigable de cara al usuario final, con interfaces de ventanas funcionales y bien diseñados, ejemplos claros de estas mejoras en escritorios son KDE y Gnome, estos dos entornos gráficos que proporcionan al usuario una visión simple y muy funcional del uso de un ordenador, han contribuido enormemente a que la visión de los usuarios de Linux escape a la austera línea de comandos a la que estaban acostumbrados.

Otro pilar fundamental en el acercamiento de Linux a los usuarios finales son las mejoras introducidas por las distintas distribuciones tanto en autodetección de hardware como en sencillez y facilidad de instalación.

Distribuciones como Red Hat o Mandrake, han contribuido decisivamente a facilitar a los usuarios el proceso de implantar en su ordenador un sistema Linux convirtiéndolo en una "workstation" completa.

Aun así muchos usuarios se encuentran confundidos a la hora de realizar una instalación o una configuración de su máquina Linux llegando en algunos casos a desistir, no por estar a disgusto con el sistema, sino por encontrarlo demasiado complejo para trabajar con él.

Ante esta problemática surge el fenómeno live CD. Las distribuciones en live CD son distribuciones que no han de ser instaladas para conseguir un funcionamiento integral y completo del sistema. Para trabajar con una "live" es necesario únicamente arrancar desde el lector de CDROM y el sistema se cargará en memoria ofreciéndonos un sistema Linux completo y configurado, que se puede usar desde el primer momento sin el trauma que puede suponer para el usuario una instalación, y sin realizar ninguna modificación en su configuración actual de disco duro.

La filosofía de las distribuciones live se resume en la siguiente frase: "pruébalo, úsalo y si te gusta quédatelo". En esta frase se condensa el propósito de una live CD, el de ofrecer el sistema al usuario antes de que tenga que instalarlo definitivamente en su PC, lo cual también se

* Presidente del Grupo de Programadores y Usuarios de Linux (GPUL).

realizará de la manera más sencilla posible.

Ejemplos de distribuciones live muy famosas son Knoppix, una live diseñada por Klaus Knopper y que se puede considerar la madre de todas las demás live por su detección de hardware y su autoconfiguración, Gnome Live CD, una live CD creada en España en el marco del proyecto metadistros, proyecto que pretende facilitar la creación de distribuciones live a medida de manera muy sencilla.

Como último ejemplo cabe destacar la GPUL live CD, distribución live creada por el Grupo de Programadores y Usuarios de Linux da Universidade da Coruña, la cual trata de cubrir el hueco que los estudiantes más noveles encuentran al tener que enfrentarse con entornos Linux en su primer año de carrera. Esta distribución ha buscado no sólo incluir todo el software necesario para que los alumnos puedan desarrollar sus prácticas sino que se ha intentado imbuir la cultura de la facultad dentro de la propia distribución.

4.2. Pasos previos al uso de una live CD

El primer paso para trabajar con una live CD es conseguir que ésta arranque en nuestro ordenador, para ello hemos de modificar la secuencia de arranque del mismo.

El ordenador intenta arrancar distintos dispositivos de manera secuencial (boot sequence) en un orden que viene dado por la BIOS (sistema que reside en nuestro ordenador y que se encarga, entre otras cosas, del arranque).

Para arrancar nuestra live necesitamos que la secuencia de dispositivos de arranque tenga el CDROM como primer dispositivo en el arranque, para ello hemos de entrar en nuestra BIOS y cambiar el parámetro que indica a la misma la secuencia de arranque.

Las BIOS son muy distintas y en ocasiones particulares a cada modelo de ordenador. Se suele acceder a ella pulsando la tecla “Supr” cuando el ordenador está arrancando. Aparecerá una pantalla en la mayoría de los casos azul y que en uno de sus menús (usualmente el “BIOS FEATURES SETUP”) pondrá “BOOT SEQUENCE”, pulsando generalmente la barra espaciadora del menú o la tecla “ENTER” con esa opción resaltada, el orden de las unidades de arranque irá cambiando. Nosotros necesitamos que ponga en primer lugar el CDROM, por ejemplo: CDROM,C,A. A continuación pulsamos “ESC” y elegimos la opción que se corresponda con salir guardando los cambios realizados (por ejemplo puede ser “SAVE & EXIT SETUP”).

Una vez hemos conseguido cambiar la secuencia de arranque, solo nos queda iniciar el ordenador con la live dentro de la unidad de CDROM y veremos el arranque de la distribución.

4.3. Comenzando con una live CD.

Para explicar el funcionamiento de una live CD tomaremos como patrón la GPUL live CD. Cada distribución live se comportará de distinta manera según como haya sido creada y el propósito para el que haya sido concebida. Sin embargo, en muchos casos, el arranque es similar.

Una vez que hayamos arrancado podremos ver como se inicia un entorno gráfico (aunque esto es completamente dependiente de la distribución) y podremos empezar a trabajar.

Debido a que el hardware es siempre muy variable y heterogéneo podemos encontrarnos problemas a la hora de arrancar la distribución. Aunque éstos no suelen ser habituales, es interesante tener algún modo de configurar el arranque para prever problemas.

Una facilidad que da GPUL live CD es la pantalla de ayuda a la que se accede mediante la pulsación de la tecla F1 en la pantalla de arranque y que muestra distintas formas de arranque para la distribución.

Una vez hayamos arrancado y se haya cargado el escritorio (en este caso particular KDE) estamos en disposición de usar la live CD como un escritorio común de forma muy similar al

de Windows.

4.4. Escritorio de una live CD

El escritorio de GPUL live CD presenta algunas novedades que es interesante que sean explicadas a la persona profana en la materia.

4.4.1. GPUL live CD y la identificación de particiones existentes.

Aunque la distribución no toca en ningún momento ni el disco duro ni ningún aspecto de la configuración de los sistemas existentes, sí detecta las particiones que tenemos en ese momento, de forma que podremos acceder en modo sólo lectura a cualquier dato que tengamos en el disco duro en ese momento, sin preocuparnos de que se puedan borrar o perder.

Llegados a este punto es interesante comentar la nomenclatura de los discos y particiones en sistemas Linux, de modo que el usuario pueda identificar qué particiones ha detectado la live y cómo se corresponde con nuestras particiones.

Linux identifica los discos o unidades de CDROM con hda, hdb, etc... si son discos IDE y sda, sdb, etc... si son dispositivos SCSI. Del mismo modo, las particiones se identifican con números sobre esta nomenclatura, la primera partición del primer disco duro IDE se nombraría como hda1, la segunda como hda2, etcétera y si fuesen unidades SCSI como sda1, sda2 y así consecutivamente.

Como ejemplo práctico: si en nuestro disco tuviésemos una única partición que Windows identificase como C:, probablemente Linux la reconocería como hda1.

Cada unidad detectada por la live se encuentra identificada con el icono de un disco duro en la parte izquierda de la pantalla y en su etiqueta podemos ver la nomenclatura que Linux ha dado a cada partición y que se corresponde con la explicada anteriormente, de este modo podremos acceder a esa partición en modo solo lectura, haciendo un click con el ratón sobre ese disco.

4.4.2. GPUL live CD y el entorno gráfico.

En el escritorio podemos identificar dos partes:

- El área de iconos: Donde podemos ver las particiones identificadas, el disquete y las unidades de CDROM y un manual de la distribución para que sea fácilmente accesible.
- El panel: El panel nos mostrará ciertos iconos para tener un acceso rápido a ellos (OpenOffice, el navegador Konqueror, etc...), un esquema de los distintos escritorios que tenemos y a los que se puede acceder pulsando sobre cada uno de los cuadros, y el menú general del sistema identificado con una letra "K" grande y que se puede considerar análogo al "Inicio" en el sistema operativo Windows.

Desde cada una de estas partes podemos acceder a distintas aplicaciones de una amplia variedad que suelen acompañar a distintas distribuciones. En concreto, en GPUL live CD, todas estas aplicaciones aparecen listadas en el manual y divididas en secciones para que cualquiera pueda acceder a cada una de ellas, bien por su nombre, bien por su ejecutable en una consola del sistema.

Tenemos ahora una distribución Linux totalmente funcional desde la que podemos acceder (como lectura) a particiones ya existentes y con una gama muy amplia de software, en muchos casos de propósito general, que nos permitirán desarrollar las funciones típicas de una

estación de trabajo desde nuestra distribución.

4.5. La consola

En muchos casos, el uso de un entorno gráfico manejable "a golpe de ratón" es suficiente para el usuario final de modo que éste pueda desempeñar todas las funciones que le son necesarias limitándose a trabajar sobre el entorno gráfico con ratón. Sin embargo, para usuarios algo más avanzados que precisan de entornos de programación, de compilación o simplemente quieren ir más allá en su manejo de la distribución, es preciso en algunos casos usar la línea de comandos para distintas funciones.

En GPUL live CD uno de los iconos que aparecen en el panel es el de un monitor con una concha encima, si pulsamos sobre él tendremos una ventana con una línea de comandos donde poder ejecutar ordenes para realizar distintas funciones en el sistema, ésto es lo que se conoce como "shell" y el programa que nos permite realizar ésto se denomina, en GPUL live CD, Konsole.

Esta shell nos va a permitir realizar tareas más avanzadas con la distribución, que irán desde su instalación a configuración de aspectos específicos del sistema o usos avanzados, como veremos más adelante.

4.6. Persistencia de los datos en una live CD

Cuando el usuario ha tomado ya contacto con la distribución, ha probado sus programas y decide ponerse a trabajar con la live, surge siempre una misma duda, ¿qué ocurre con los datos que puedo generar con las aplicaciones de la live?.

4.6.1. Persistencia de los ficheros creados.

Por el hecho de que la live corre en RAM y no tiene acceso de escritura en los discos duros, si guardamos en disco nuestro trabajo, una vez hayamos reiniciado se perderán todos los datos que hemos generado. Ésto no es operativo desde ningún punto de vista y es interesante saber las opciones que tenemos para que ésto no sea así.

Una de las soluciones más sencilla, pero también menos eficiente, que tenemos para guardar la persistencia de los datos generados con aplicaciones (ficheros con textos, programas realizados, imágenes, etc...) es la de crear estos siempre sobre un disquete. El disquete permite, por regla general en todas las live, acceso para escritura, de modo que si depositamos en la unidad de disquete los archivos generados, simplemente arrastrándolos a ella, estos archivos quedarán guardados en el disquete y será posible su utilización.

Pero existe un problema y es que en muchos casos los datos que deseamos generar ocuparán más de un disquete y no será posible guardarlos en él. Llegados a este punto es necesario "montar" las distintas particiones para lectura y escritura, de modo que podamos escribir ficheros en las mismas.

El concepto de "montar" es un término novedoso para la gente que llega de otros sistemas operativos no UNIX a Linux. El sistema de archivos en Linux se estructura en forma de árbol invertido, de modo que en la raíz del árbol tenemos la / (fichero raíz), y de ella cuelgan distintos directorios y subdirectorios que conforman el resto del árbol. Cada dispositivo tiene una /, por ejemplo un disquete tendrá una / de la que colgarán el resto de archivos y subdirectorios, la acción de montar un disquete por ejemplo sería asignar esa / del disquete a un directorio concreto, de modo que debajo de este directorio se encajará el árbol del disquete como un subárbol completo que pende del árbol general del sistema, pudiendo acceder de esta forma a sus ficheros.

Para realizar el "montado" de una partición debemos de entrar en el sistema como

administrador, para ello nos bastará ir al menú de aplicaciones y en el submenú “KNOPPIX” elegir la opción “KNOPPIX-ROOTSHELL”. Esta opción nos abrirá una consola con permisos de administrador desde la que podremos realizar algunas acciones que no nos estaban permitidas en otras, por ejemplo el montaje de una partición.

De esta manera, el montaje de una partición para lectura/escritura lo podemos hacer en línea de comandos de la siguiente manera:

```
# umount /mnt/(dispositivo)
# mount -o umask=000 /mnt(dispositivo)
```

siendo el dispositivo la nomenclatura de la partición que pretendamos montar (hda1, hda2, ...), e indicando el # que estamos accediendo como administrador del sistema (no se debe teclear).

Con este método podremos ahora arrastrar archivos a las particiones que hayamos montado de esa manera y permanecerán en esas particiones una vez se haya reiniciado el equipo.

4.6.2. Persistencia de la configuración de la distribución.

En el apartado anterior se han tratado los aspectos de persistencia que más interesan comúnmente a los usuarios, que son los de salvaguardar los datos generados en la distribución. Sin embargo existen usuarios que desearían poder guardar la configuración de su escritorio y sus aplicaciones, para poder recuperarla en su próxima sesión de trabajo con la distribución live.

Para ello, GPUL live CD se ha apoyado en Knoppix y en el menú “*KNOPPIX->Configuration Save*” conseguiremos salvar toda nuestra configuración a disquete.

Para recuperar esta configuración guardada previamente, debemos de escribir en el arranque, donde aparece la palabra boot: la siguiente instrucción:

```
knoppix floppyconfig.
```

Ésto hará que una vez arrancada la distribución, vaya al disquete a recuperar la configuración y restaurarla. De este modo conseguiremos recuperar la configuración que habíamos salvado.

4.7. Configuración de una live CD

Como ya hemos comentado en el apartado anterior, la configuración del escritorio (temas, ventanas, etc...) en una distribución live no es persistente por defecto, pero sí puede lograrse a través de utilidades como las que hemos comentado anteriormente.

En GPUL live CD tanto el aspecto del escritorio como distintos apartados de hardware pueden ser configurados, y su configuración guardada para sesiones posteriores.

En primer lugar podemos configurar el aspecto de escritorio con el icono del panel que presenta unas tarjetas integradas en su dibujo. Este icono permitirá cambiar el aspecto del tema, el color de las ventanas, el fondo de escritorio y ciertos parámetros que ayudarán a realizar una personalización del entorno de trabajo adaptándolo así a nuestros gustos.

Sin embargo, tan importante como esta adaptación es la configuración de algunos aspectos de hardware que aunque normalmente son autodetectados por la distribución, en ocasiones es necesario correr algún script (o programa) para alguna configuración adicional. Este es el caso de la configuración de la red. En muchos casos y si la red tiene un servidor DHCP, lo cual deberíamos consultar con nuestro administrador de red, la live CD se configurará automáticamente y todos los parámetros se actualizarán en nuestra instalación para permitir la interconexión con otras redes. Sin

embargo en otros muchos casos no tendremos un servidor DHCP en nuestra red y deberemos configurar nuestras tarjetas de red con nuestra IP, nuestra máscara de red, y nuestra puerta de enlace. Esto lo podemos hacer en GPUL live CD con la opción del menú principal “*KNOPPIX->Network/Internet->Card Configuration*”, la cual nos permitirá introducir todos los parámetros necesarios.

Del mismo modo podemos realizar la configuración de la tarjeta de sonido con la utilidad del menú “*KNOPPIX->Configuration->Sound Card Configuration*”.

Estos menús nos ayudarán a hacer una configuración muy sencilla de estos dos elementos, cuya configuración podría resultar problemática para el usuario novel.

4.8. Instalación de una live CD.

Una vez hayamos trabajado durante un tiempo con nuestra live CD y estemos acostumbrados a su manejo, probablemente desearemos instalar todo este entorno en nuestro disco duro de manera que podamos trabajar directamente en nuestras particiones y con más comodidad que en la distribución live.

Comentaremos aquí el proceso de instalación de GPUL live CD paso a paso, proceso para el que se ha utilizado el instalador de Knoppix, de modo que servirá como referencia para cualquier distribución descendiente de este proyecto.

Pasos en la instalación:

1) Selección del disco de instalación

En esta fase y una vez se ha presentado el mensaje de bienvenida, la instalación nos permitirá elegir en qué disco instalaremos nuestra distribución (posteriormente tendremos la oportunidad de indicar en qué partición concreta lo haremos).

Los discos los nombra con la nomenclatura que vimos anteriormente, ésto es, el disco o dispositivo conectado en el IDE primario como maestro será hda, el colocado en el IDE primario con esclavo será hdb, el colocado en el IDE secundario será hdc y así consecutivamente.

2) Particionado

Una vez hemos elegido el disco en el que haremos la instalación, se nos presentará una herramienta que nos permitirá realizar una partición para nuestro sistema Linux.

Para la instalación hemos de hacer dos particiones, una que no podrá ser menor de 2 GB, y que será la que albergará nuestro sistema, y otra que será una partición de intercambio y que podría tener unos 300 MB.

Para crear estas particiones debemos de ir a una zona de espacio libre y pulsar en la opción "Nueva" y nos preguntará por el espacio que queremos para ella. Posteriormente y una vez creada pulsaremos en la opción "Tipo" para indicar el tipo de la partición: a la partición Linux (la de más de 2GB) le corresponde el tipo "Linux" (83) y a la partición de intercambio (la de 300MB) le corresponde el tipo "Linux Swap" (82). Una vez creadas estas particiones sólo nos restará escoger la opción "Escribir", con lo que se guardarán los datos.

3) Particiones y copia.

Ahora nos preguntará qué partición queremos para instalar la distribución y qué partición queremos para asignar al swap (partición de intercambio), ésto lo autodetecta y sólo nos mostrará para elegir aquellas que tengan tipo Linux y Linux Swap respectivamente.

Una vez realizada esta elección, la instalación pasará a copiar el contenido de la live al disco duro teniendo así en disco lo mismo que teníamos en la distribución en modo live.

4) Creación de usuarios

Una vez ha copiado el contenido de la live nos preguntará por la contraseña que queremos darle al usuario root y por el nombre y contraseña de un usuario común en el sistema. Es interesante hacer notar que para el trabajo común que desempeñamos a diario con Linux, no debemos entrar como el usuario root, ya que éste es administrador del sistema y puede provocar cambios irreversibles en él. Debemos siempre, en la medida que nos sea posible, trabajar con el otro usuario creado, salvo para tareas de administración.

5) Instalación del gestor de arranque.

Una vez hemos acabado estos pasos, nos preguntará si queremos hacer un disquete de arranque o meter dicho arranque en el sector de arranque del disco. Ante esto hemos de realizar varias consideraciones: por una parte siempre es interesante tener un disco de arranque por lo que pueda pasar, mientras por otra, si metemos el arranque en el disco, éste borrará el arranque de Windows intentando meter a su vez la partición Windows en el arranque; ante esta circunstancia siempre es interesante tener un disco de rescate de Windows para poder recuperar cualquier efecto no deseado. (Todo esto suponiendo que tengamos Windows instalado en nuestro ordenador).

6) Ajustes post-instalación

Si Lilo, el gestor de arranque de Linux, no ha conseguido meter nuestra partición Windows en la secuencia de arranque, podemos hacerlo nosotros a mano. Para ello basta editar el fichero “/etc/lilo.conf” y añadir las siguientes líneas:

```
other=/dev/(partición windows)
label="Windows"
```

y en una consola de administrador ejecutar el comando “lilo”. De este modo añadiremos Windows a nuestro arranque y podremos tener arranque dual.

4.9. Conclusiones

Una vez instalada la distribución tendremos un sistema Linux totalmente funcional, basado en Debian GNU/Linux y donde se pueden aplicar todos y cada uno de los comandos típicos de Linux y Debian. También se mantendrán todas las utilidades que hemos manejado en la live hasta ahora y que podremos manejar más cómodamente desde nuestro PC, pudiendo escribir a disco y sin tener que hacer maniobras adicionales para salvaguardar nuestros datos y nuestra configuración.

CAPÍTULO 5

Instalación y configuración paso a paso de RED HAT Linux 9.0

Santiago Vázquez Salgado*

5.1.Introducción

La instalación de RED HAT Linux es muy sencilla, su mayor dificultad está en conocer cómo ha de ser repartido el disco duro y cómo han de ser las particiones necesarias para preparar el sistema.

Antes de nada debemos asegurarnos de que nuestro equipo cumple los requisitos mínimos necesarios para instalar el sistema, esto no quiere decir que si nuestro equipo no está en la siguiente tabla no podamos instalar Linux, pero sí que si lo hacemos, no tendremos garantía de que el ordenador funcione con “soltura” si le exigimos ciertas cosas por encima de lo razonable.

Mínimo para modo texto: Procesador compatible con Pentium, recomendado 200 Mhz o superior. Memoria RAM: 64MB o más.
Mínimo para modo gráfico: Procesador recomendado Pentium II 400 Mhz (o compatible) o superior. Memoria RAM mínima: 128 MB, recomendada 192 MB o más.
Espacio libre en disco duro: Instalación mínima : 475 MB Servidor mínimo: 850 MB Escritorio personal: 2 GB Estación de trabajo: 2,4 GB Todos los paquetes: 6,4 GB Se requiere también espacio libre adicional para los datos del usuario.

Si queremos comprobar que nuestra máquina está totalmente soportada por RED HAT Linux, una solución es visitar la dirección <http://hardware.redhat.com/hcl/> , ahí nos encontraremos el listado “Red Hat Linux compatibility List”, que es el listado completo de compatibilidad de hardware soportado por Red Hat Linux.

* Grupo de Usuarios de Linux de Ourense

Una vez comprobado que nuestro equipo es compatible con RED HAT Linux (la gran mayoría lo son) vamos a comenzar la instalación, la cual vamos a dividir en los siguientes pasos:

- Preparar Espacio libre en el disco duro.
- Iniciar el instalador.
- Tipo de instalación.
- División del disco en particiones.
- Configuración del sistema.
- Selección de paquetes.
- Finalización de la instalación.

5.2. Preparando espacio libre en el disco duro

Aquí tenemos dos casos principales: el primero es el de un usuario cualquiera que dispone de todo el disco duro para instalar Linux, mientras que el segundo, y el más común, es el usuario que ha comprado la máquina con un sistema operativo ya instalado (generalmente Windows) y éste ocupa todo el espacio del disco duro. El primero no tiene problema ninguno y puede pasar directamente al siguiente paso (Iniciar el instalador), mientras que el segundo tiene que “encoger” la partición de Windows. El método es sencillo, bastará usar algún programa de gestión de disco (como por ejemplo Partition Magic para Windows o run_qtparted para Linux) para modificar la partición de Windows y así dejar espacio libre para el nuevo sistema operativo que vamos a instalar. Ambos instaladores son muy gráficos y nos permiten escoger el tamaño de las particiones que dejamos en nuestro disco duro, mostrando el que dedicamos a Linux, Linux Swap y a otros sistemas operativos.

5.3. Iniciar el instalador

Una vez tenemos hueco en el disco duro para nuestra instalación de Linux, necesitamos arrancar ésta, para ello configuraremos la BIOS de nuestro ordenador para que arranque desde el CDROM (ver sección 4.2 del manual). El siguiente paso es introducir el primer CD de la distribución en la unidad lectora y reiniciar el ordenador. Puede darse el caso de que nuestra unidad lectora sea muy antigua o simplemente no permita leer correctamente un disco arrancable, o también puede ser que la BIOS de nuestro ordenador no permita el arranque desde la unidad de CD, en cuyo caso necesitaremos crear un disquete de arranque. El primer CD de la distribución trae una utilidad llamada rawrite y una imagen bootdisk.img, la cual una vez grabada en un disquete nos permitirá arrancar la instalación desde la disquetera del ordenador. El comando para crear el disquete a partir de uno formateado es:

En DOS:

```
x:\dosutils\rawrite x:\images\bootdisk.img
```

En Windows:

```
x:\dosutils\rawritewin\rawritewin x:\images\bootdisk.img
```

En Linux:

```
dd if=/mnt/CDrom/images/bootdisk.img of=/dev/fd0
```


donde en lugar de x tenemos que poner la letra de nuestro lector de CDROM (generalmente “d”).

Una vez iniciada la instalación veremos la pantalla de bienvenida. Para la mayoría de los usuarios se pulsa la tecla “Enter” y arranca la instalación en modo gráfico 800x600 con una profundidad de color de 16 bits. Puede que no todo el mundo tenga un equipo que soporte esa resolución por lo que tenemos otras opciones:

- Text: Instalación en modo texto.
- Lowres: Instalación a baja resolución (640 x 480)
- Expert: modo experto sin autodetectar los dispositivos.
- Rescue: modo de rescate. También pulsando la tecla F2.

Para cargar cualquiera de las opciones anteriores escribiremos Linux en la línea de comandos seguido de la opción seleccionada (por ejemplo, *boot: Linux text*).

Si tenemos un ordenador más o menos nuevo y con dispositivos no demasiado raros o difíciles de configurar, deberíamos poder arrancar simplemente pulsando la tecla “Enter”. Una recomendación muy a tener en cuenta es tener conectados y encendidos, en el momento de la instalación, todos los dispositivos que queramos usar en el ordenador, para que así ésta los detecte y los configure casi sin tener que hacer nosotros nada.

Lo primero que nos encontramos al iniciar la instalación es una prueba de CD's; es recomendable hacerla una vez al menos por cada CD, ya que puede venir defectuoso alguno y terminar la instalación a medio camino, por lo que seleccionamos la opción “OK” y pulsamos la tecla “Enter”. A continuación se probará el CD (lo cual lleva un poco de tiempo) y si todo está correcto nos dirá que metamos otro o continuemos con la instalación. Una vez realizada la prueba de todos los CD's pulsamos continuar y arrancará, esta vez sí, el instalador de RED HAT con un entorno totalmente gráfico y amigable al usuario.

Aparece una pantalla donde nos da a elegir nuestro idioma; elegimos el nuestro y pulsamos la opción “siguiente”. A continuación cargará la siguiente pantalla en el idioma que hayamos elegido; en esta pantalla es donde seleccionamos el tipo de teclado que tenemos, pulsamos “Siguiente” y nos saldrá algo parecido pero para el ratón. Generalmente ya aparecerá seleccionado teclado o ratón que tengamos, debido a que la instalación los habrá detectado, pero si observáramos que no es correcto o que hay alguno que sea más preciso a la hora de describir el dispositivo que tenemos, lo seleccionaremos y pulsaremos “Siguiente”.





5.4. Tipo de instalación

En este paso se nos da a elegir entre 4 tipos de instalación:

- **Escritorio personal:** la más apropiada para los usuarios de portátiles; esta elección instalará un sistema operativo con un escritorio gráfico y algunos programas básicos.
- **Estación de trabajo:** sistema con más herramientas de administración y para programar.
- **Servidor:** Sistema con un servidor en Linux (compartir impresoras, servidor web, servidor de bases de datos,...).



La elección de cualquiera de esas opciones dividirá automáticamente el disco duro y nos mostrará los paquetes a instalar; nosotros seleccionaremos la última opción, “instalación personalizada” y a continuación pulsaremos “Siguiente”. Nos preguntará si queremos particionar el disco duro manualmente mediante la herramienta “disk druid” o si por el contrario preferimos que sea el instalador el que asigne el espacio a cada partición automáticamente. Esta última opción es la más recomendable para una primera instalación, es decir, para un usuario que nunca haya instalado Linux y no sepa muy bien cómo particionar el disco duro. Nosotros seleccionaremos la opción de partición manual mediante disk druid y pulsaremos “Siguiente”.

5.5. División del disco en particiones

Linux necesita, como todo sistema operativo que se precie, una partición o espacio en el disco duro en donde guardar sus archivos. Esta partición (que en Windows se conoce generalmente como c:) es la partición primaria o raíz y se representa por el símbolo de barra inclinada “/”. De ella cuelgan todos los archivos. El tamaño necesario para esta partición es relativo, una instalación máxima viene ocupando unos 5 GB, mientras que una mínima ocupa unos 450 MB. Ésta debe ser una decisión del usuario dependiendo de los programas que necesite instalar. Como recomendación decir que una instalación media, para alguien que quiera probar distintos programas incluidos en la distribución, suele ocupar entre 2 y 3 GB. Aparte, Linux necesita una partición de intercambio de memoria, llamada “swap”, que suele ser del doble de tamaño que la memoria RAM de que dispongamos. Linux usa esta partición como un cajón de sastre, es decir, todo lo que no use en ese momento pero vaya a necesitar, o lo que no quepa en la memoria RAM, por encontrarse esta al 100 % de su uso ya, lo archivará en esa memoria de intercambio, facilitando así el acceso a los datos.

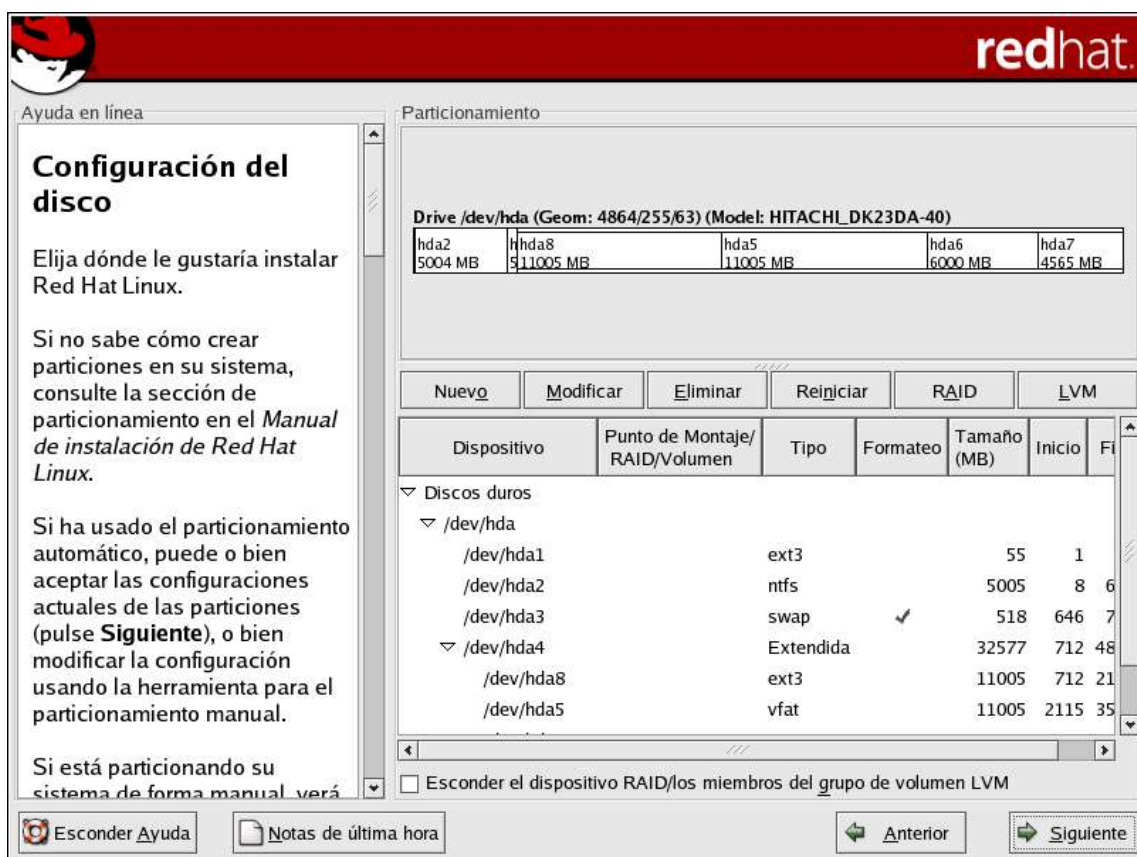
De igual modo es recomendable crear una partición llamada /home en la que irán nuestros archivos personales, configuraciones y demás. La creación de esta partición, si se dispone de espacio libre en el disco duro, es muy recomendable, ya que si decidimos por alguna razón reinstalar el sistema, o incluso cambiar de distribución Linux, nuestros datos y nuestras configuraciones de programas (escritorio e iconos, datos personales y archivos propios,...) se conservarán en esta partición.

Después de ver todo esto, nos queda la siguiente distribución del disco duro:

- “/” : entre 2-4 gigas (6 como máximo).
- “swap” : el doble de la memoria RAM.
- “/home”: a decisión del usuario (recomendado 500MB min).

Una vez sabemos cuántas particiones queremos y de qué tamaño, vamos a crearlas con la utilidad disk druid. La interfaz es muy sencilla: tiene un botón para crear particiones, uno para modificar una existente, uno para eliminar particiones y otro para restaurar la tabla de particiones a su estado anterior (deshacer todos los cambios hechos en este paso).

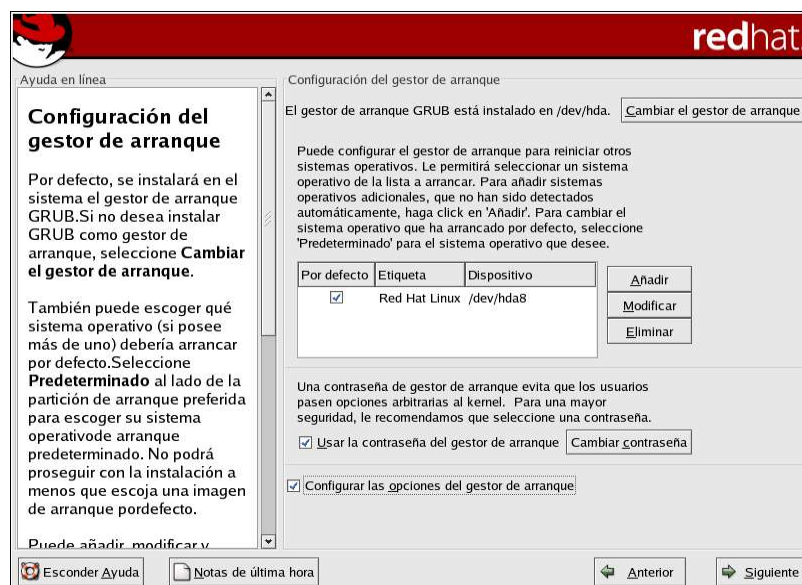
El botón crear nos abrirá un submenú en el que tendremos que indicarle el punto de montaje (el nombre de la partición, “/”, “/home”,...), el tipo de sistema de ficheros (ext3 para “/” y “/home”, y swap para la partición de intercambio) y el tamaño en MB. Además, ofrece opciones adicionales para el tamaño, como es completar el espacio libre restante en disco. El botón “Modificar” nos dejará modificar estos datos con un submenú parecido, si no igual al de “Crear”.



Una vez creadas las particiones en el disco duro pulsamos “Siguiente” y aparecerá un mensaje donde se nos dice que va a proceder a formatear las particiones marcadas para tal fin. Pulsamos “OK” y pasaremos al siguiente paso.

5.6. Configuración del sistema

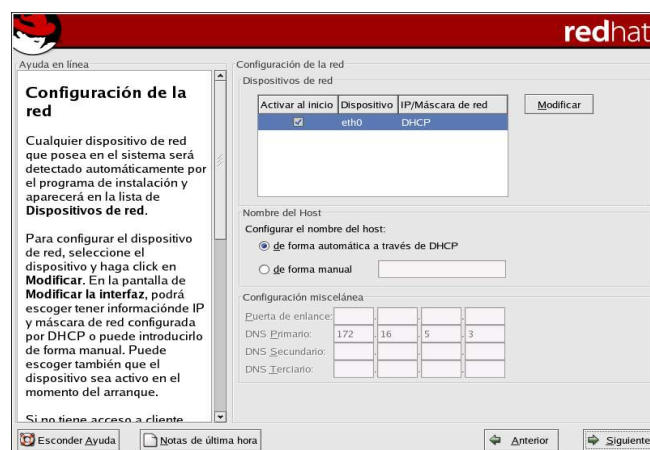
Una vez llegados a este punto habremos pasado los pasos más cruciales de la instalación, ya sólo nos quedará como paso difícil la instalación del gestor de arranque. El resto de la instalación no será más que una mera configuración del sistema (firewall, contraseña de root, grupos de paquetes a instalar, monitor...).



Un gestor de arranque es un programa que está instalado generalmente en el inicio del disco duro y que es cargado por la BIOS del ordenador para más tarde arrancar éste. El gestor es el que se encarga luego de decidir qué partición inicia, es decir, qué sistema operativo arranca. Los usuarios de Windows no conocerán este tipo de gestores de arranque ya que, según encienden el ordenador arranca ya directamente Windows sin dar a elegir otro sistema. RED HAT da la posibilidad de instalar dos gestores de arranque distintos: Grub o Lilo. Nosotros instalaremos Lilo (Linux Loader) por simplicidad. En el botón “Cambiar el gestor de arranque”, seleccionamos Lilo y después comprobamos si están en las etiquetas, que aparecen más abajo en la pantalla, todos los sistemas operativos que tenemos en el ordenador, seleccionando uno por defecto. Los que estén instalando Linux como único sistema operativo de su computadora no deberán ver nada más que la etiqueta Red Hat Linux, mientras que los que tengan más sistemas (como por ejemplo Windows) tendrán una etiqueta por cada sistema y el dispositivo (partición) en el que están instalados. Una vez comprobado que todas las etiquetas son correctas marcamos la casilla “Configurar opciones del gestor de arranque” y pulsamos “Siguiente”.

La siguiente ventana nos da opción a seleccionar dónde instalar el gestor de arranque. Es recomendable instalarlo en el MBR (master boot record) y así podremos despreocuparnos de cualquier otra configuración de la BIOS o de la existencia de otro gestor de arranque distinto.

El siguiente paso es la configuración de la red local (en caso de tener instalada una tarjeta de red). En la pantalla se nos muestra lo siguiente:



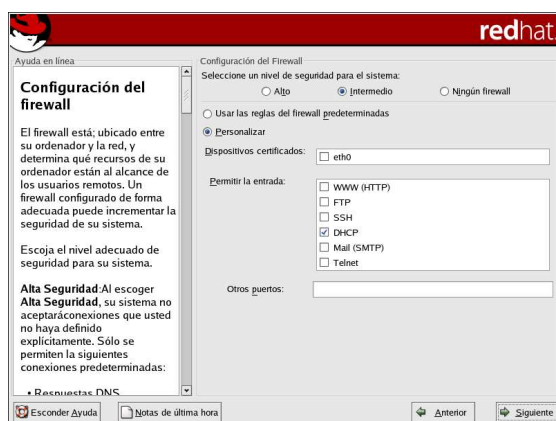
El primer listado es una lista de todas las tarjetas de red que tenemos en nuestro sistema. Seleccionando una y pulsando el botón “Modificar” entraremos en las opciones de red siguientes:



Como se ve a primera vista se puede activar DHCP de manera que sea el servidor el que nos de una IP y máscara de red, o el modo estático, escribiendo nosotros la IP del ordenador (estos datos deben ser suministrados por nuestro proveedor de Internet). Escribimos nuestra configuración de la red y pulsamos, una vez más, “Siguiente”.

Seguidamente nos permite configurar un firewall que trae por defecto ya instalado. Esta opción va dirigida a quienes tengan conexión a Internet y proporciona mayor seguridad a la hora de acceder a la red. Debemos fijarnos en qué puertos dejamos abiertos y cuáles no, ya que sin tener abiertos ciertos puertos, algunos programas no funcionarán. Si no vamos a poseer ninguna conexión a Internet, o si simplemente no queremos tener ningún firewall configurado, seleccionaremos la opción “Ningún Firewall”.

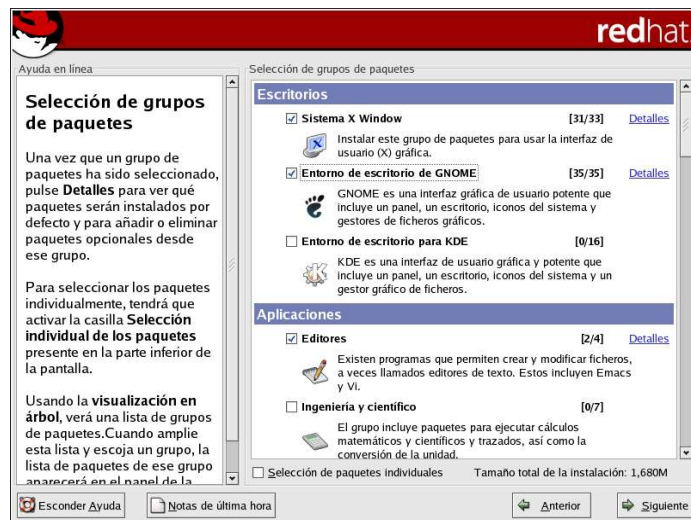
Si queremos un nivel de seguridad intermedia, seleccionaremos los puertos que queremos dejar abiertos; si no están en la lista los agregaremos en “Otros Puertos”. Para finalizar, si queremos tener un nivel de seguridad “paranoico”, seleccionaremos la opción “Alto” y no tendremos salida a ningún puerto que no sea indicado por el usuario.



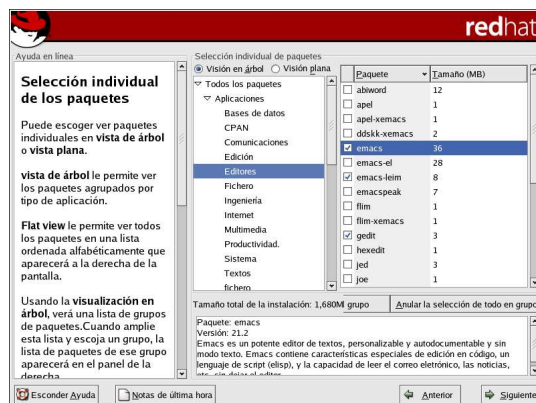
Una vez pulsamos “Siguiente” nos permitirá elegir en una lista el soporte a los idiomas que queramos, es decir, los idiomas que queremos que nuestro sistema soporte a la hora de visualizar, editar o trabajar con él. Seguidamente nos preguntará por el huso horario (seleccionaremos el lugar más cercano al que vivamos) y por la cuenta del “root”, es decir, la contraseña del “superusuario”, el que tendrá privilegios para configurar el sistema y administrarlo. Esta contraseña ha de ser mayor de 6 caracteres y tenemos que introducirla correctamente en los dos cuadros de texto. En caso de que las dos contraseñas sean iguales se habilitará el botón “Siguiente”, si no, mostrará un mensaje donde se nos dirá que son diferentes, obligándonos a reescribirlas.

5.7. Selección de paquetes

Aquí es donde seleccionaremos los paquetes a instalar. Tenemos desde paquetes de entorno gráfico o “suits” ofimáticas, hasta paquetes de configuración de servidores; todo esto es, más o menos, a elección del usuario, aunque hay un mínimo de paquetes que son recomendables, como son los paquetes de ofimática, los entornos gráficos (las X),... Si pulsamos “Detalles” se nos mostrará un listado con los paquetes que se instalarán y otro con los que no. Podemos modificar seleccionando cuales no (desmarcándolos) o cuales sí (marcándolos).



Si marcamos la opción “Selección de paquetes individuales” nos mostrará un listado de todos los paquetes disponibles, así como una casilla de verificación. Los que estén puestos para instalar aparecerán marcados y los que no sin marcar. Aquí podremos elegir paquetes en un listado por orden alfabético o, simplemente si seleccionamos el paquete, ver qué es, qué hace, o para qué sirve. Es recomendable leerse al menos una vez en la vida ésto, para así saber para que vale cada paquete, y tener claro en esta instalación y en las futuras qué necesitamos y qué no.



Una vez pulsamos “Siguiente” el sistema comprobará las dependencias y nos las mostrará, preguntándonos qué queremos hacer. Ésto se refiere a que cada paquete depende de otros, es decir, por ejemplo, no podemos instalar un navegador de Internet sin instalar un programa que nos conecte a la red. Tendremos tres opciones: dejar que instale los paquetes y sus dependencias, que ignore las dependencias o que no instale los paquetes que piden dependencias.

Cuando pulsamos “Siguiente” nos muestra una pantalla de fin de instalación. Aquí aún

podemos echarnos atrás, pero una vez que pulsemos el botón “Siguiente” comenzará el formateo de las particiones y la instalación del sistema operativo. Ésto, según el equipo de que dispongamos, tardará más o menos tiempo, pero podemos ver su progreso en la barra junto con el tiempo estimado.



5.8. Finalización de la instalación

Después de pasar algún tiempo observando el progreso de la instalación (pausado sólo por el intercambio de disco en la unidad de CD), es el momento de crear un disquete de arranque desde el que arrancar el sistema. Ésto se hace por si se diera el caso de que nuestro gestor de arranque no funcionara correctamente o se produjera cualquier otro error que nos impidiera iniciar correctamente el sistema. También puede darse el caso de que no instaláramos el gestor de arranque (Lilo o Grub) y queramos arrancar desde el disquete. Para crearlo metemos un disquete en la disquetera y pulsamos “Siguiente”.

A continuación comprobamos si nuestro sistema ha detectado correctamente nuestra tarjeta gráfica, si no es así, podemos cambiar manualmente todas las opciones o seleccionar una de la lista. Para los que posean una tarjeta gráfica con aceleración 3d y quieran aprovechar dicha aceleración por hardware, tendrán que acceder a la página web del fabricante de su tarjeta y comprobar si tiene drivers para Linux (por ejemplo, “nvidia” y “ATI” los tienen). Una vez que todo está correcto pulsamos “Siguiente”.

Otro paso es elegir el monitor adecuado. En caso de que el nuestro no aparezca en la lista podemos configurarlo manualmente. Debemos tener cuidado a la hora de introducir los datos de refresco (generalmente incluidos en el manual del monitor), ya que éste puede fallar. Otra opción es seleccionar un monitor genérico, con una resolución que sepamos soportada por el nuestro.

Configuramos en la siguiente pantalla la resolución y la profundidad de color (teniendo en cuenta lo soportado por nuestra tarjeta gráfica y nuestro monitor). A continuación sacamos los discos de las unidades y pulsamos “Salir”. El sistema se reiniciará con RED HAT Linux instalado y como opción en nuestro menú de Lilo donde podremos elegir con qué sistema operativo queremos trabajar en ese momento. Si elegimos RED HAT Linux éste se iniciará y aparecerá una pantalla donde nos permitirá introducir el usuario y password para acceder al sistema.

Paso siguiente: a trabajar.

CAPÍTULO 6

Instalación y configuración paso a paso de Debian GNU/Linux

Manuel Pedreira Antón*

6.1. Introducción

En este capítulo se tratará la instalación de Debian GNU/Linux (en adelante Debian). Debian es posiblemente una de las distribuciones más complicadas de instalar dentro del mundo de GNU/Linux y por ello menos aconsejable para todos aquellos que han decidido unirse al mundo Linux hace poco tiempo.

Aunque ya fueron comentados en el Capítulo 1 algunos detalles sobre esta distribución, a continuación se repasarán los mismos, ya que ante todo lo que se busca es que este capítulo pueda servir de referente como unidad individual sin depender excesivamente del resto del manual.

6.2. ¿Qué es Debian GNU/Linux?

Debian es un sistema operativo libre basado en el núcleo de Linux. En la actualidad, Debian está disponible para Intel x86, Power PC, Alpha y Sparc por citar algunas de las arquitecturas más importantes.

Aparte de su naturaleza voluntaria, el proyecto Debian tiene una característica que lo hace especialmente singular: el contrato social de Debian (Debian Social Contract). Este documento contiene no sólo los objetivos principales del ya citado proyecto sino también los medios que serán utilizados para llevarlos a cabo.

Debian también es conocida por tener una política de paquetes y de versionado muy estricta con el fin de obtener una mayor calidad del producto. En todo momento existen tres versiones diferentes de Debian: una versión estable, otra inestable y otra en pruebas.

Como su nombre indica, la versión estable es la versión destinada a sistemas y personas no aptas para sobresaltos. Su software ha de pasar determinado período de congelación en el cual únicamente se corrigen erratas. La norma es que la versión estable de Debian no debe contener ningún error crítico conocido. Por el contrario, la versión estable de Debian no suele tener las últimas versiones del software (lo más novedoso). Para los que deseen tener una versión con el software más actual existen otras dos versiones de Debian coetáneas con la estable.

La versión en pruebas incluye paquetes en vía de estabilización, mientras que la versión inestable, como su propio nombre indica, es la más proclive a fallar y contiene lo último de lo último en lo que a novedades en software se refiere.

A día de hoy, la versión estable de Debian es Debian 3.0rc2 (también conocida como

* 1. Grupo de Usuarios de Linux de Ourense // 2. Colaborador del Proyecto OpenMosix en Castellano

“Woody”), la inestable recibe el sobrenombre de SID (Sistema en Desarrollo) y la que se encuentra en pruebas es “Sarge”.

6.3. Instalación

6.3.1. ¿Qué se necesita para la instalación de Debian?

Aunque existen varios modos de realizar una instalación de Debian Woody, se ha elegido la instalación desde CD debido a que no todo el mundo tiene acceso a una conexión a Internet lo suficientemente rápida como para hacer una net-install (instalación por red). Será necesario:

- Un ordenador con lector de CDROM (se recomienda por lo menos un Intel Pentium o superior).
- 12 MB de memoria RAM.
- Al menos 250MB de disco (serían recomendables 1 o 2 GB).
- 7 CDs de Debian Woody (sólo es imprescindible el primer CD).
- No estaría de más disponer de conexión a Internet (para descargar actualizaciones de seguridad).

6.3.2. Configuración de la BIOS

Para la instalación desde CD es necesario que el equipo en el que se ha decidido realizar la instalación de Debian arranque desde el CDROM (en caso de que la BIOS no lo soporte se podrá hacer una instalación desde disquete).

Los pasos a seguir para hacer que la BIOS elija el arranque desde CD-ROM (o Diskette) son los siguientes: se debe acceder a la BIOS durante el arranque del ordenador (habitualmente se puede entrar presionando “Supr”). Una vez dentro de la BIOS, se debe ir a “boot device” (puede variar en cada BIOS), y modificar el orden de arranque para que sea el lector de CDROM (o el disquete) el primero en el orden de arranque.

Una explicación de este proceso de arranque desde CD también se puede ver en la sección 4.2 del manual.

Si se ha elegido la instalación desde disquete se deben descargar los archivos root.bin y rescue.bin de uno de los “mirrors” o espejos de Debian. La lista de los mismos se indica en <http://www.debian.org/distrib/ftplist.es.html>.

Una vez bajados los dos archivos de 1.44 MB cada uno, se deben copiar a un disquete con una utilidad como rawwrite¹ o con el dd de Linux ejecutando:

```
dd if=/ruta_del_archivo/rescue.bin of=/dev/fd0
dd if=/ruta_del_archivo/root.bin of=/dev/fd0
```

Acabados estos pasos previos ya se puede comenzar la instalación de Debian GNU/Linux Woody.

6.3.3. Selección del tipo de instalación

El primer paso para la instalación de Debian es, como en la mayoría de las instalaciones de sistemas operativos, el encender el ordenador con el CD de instalación dentro del CDROM (o el disquete de instalación en la disquetera si se va a instalar utilizando este soporte). Con esto lo que se consigue es que el ordenador en vez de intentar usar su propio disco duro intente

1 Utilidad para windows similar al DD disponible en: <http://uranus.it.swin.edu.au/~jn/Linux/rawwrite.htm>

leer primero el CD/disquete en busca de un sistema arrancable.

Si se ha realizado bien la configuración de la BIOS (ver 5.2.1) y el CD/disquete está bien grabado, debería aparecer en el ordenador una pantalla similar a la siguiente:

```

Welcome to Debian GNU/Linux 3.0!

This is a Debian CD-ROM. Keep it available once you have installed
your system, as you can boot from it to repair the system on your hard
disk if that ever becomes necessary (press <F3> for details).

For a "safe" installation with kernel 2.2.20, you can press <ENTER> to begin.
If you want additional features like modern hardware support, specify a
different boot flavor at the boot prompt (press <F3> to get an overview).
If you run into trouble or if you already have questions, press <F1>
for quick installation help.

WARNING: You should completely back up all of your hard disks before
proceeding. The installation procedure can completely and irreversibly
erase them! If you haven't made backups yet, remove the CD-ROM
from the drive and press <RESET> or <Control-Alt-Del> to get back to
your old system.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law. For copyright information, press <F10>.

Press <F1> for help, or <ENTER> to boot.

boot: _
```

Una vez en este punto, se puede decir que se ha empezado la instalación de Debian.

Llegado aquí lo primero que se debe hacer es elegir el tipo de instalación que se desea realizar. Para ello se puede obtener más información presionando 'F3' donde se indica que se pueden realizar las siguientes instalaciones:

- **vanilla** : es el paquete estándar del núcleo disponible en Debian GNU/Linux. Éste se caracteriza por tener casi todos los controladores soportados por Linux compilados como módulos.
- **compact**: parecido al anterior pero suprimiendo los módulos que menos se usan, esta instalación contiene los módulos de tarjetas de red, para poder instalar Debian GNU/Linux desde Internet.
- **Linux** o **idepci**: esta instalación lleva un núcleo que sólo soporta dispositivos IDE, PCI y algunos ISA. En Debian recomiendan usar esta imagen si con las otras se tienen problemas con dispositivos SCSI. Esta opción es la que se ejecuta por defecto si no se escribe nada en boot:
- **bf24**: esta instalación instala el núcleo 2.4.18, tiene soporte para sistema de ficheros EXT3, ReiserFS y soporte para mas hardware USB.

La instalación más recomendable es la última (bf24) ya que es la única que permite la instalación directa de sistemas con “Journaling” y es la única que soporta hardware USB. Se escribirá “bf24” después del boot, y se continuará la instalación.

Boot: bf24

6.3.4. Selección del idioma

El primer paso dentro de la instalación de Debian es la configuración del idioma.

Existen 19 idiomas en los que se puede hacer la instalación. Se deberá elegir el idioma que se quiera conservar durante la instalación del sistema. La selección de este idioma no implica el idioma del sistema, por tanto, es recomendable que se elija un idioma que se domine perfectamente (por ejemplo castellano o gallego).

6.3.5. Elección del teclado

Elegido el idioma, aparecerá un elemento que será visto en otras etapas de la instalación, el “Menú principal de Instalación de Debian GNU/Linux”.

Este menú guiará la instalación indicando en qué paso de la instalación se encuentra en cada momento y las alternativas que puede realizar.

El siguiente paso según el menú es instalar el teclado: se presionará “Enter” para elegir y se seleccionará “qwerty/es español”.

6.3.6. Particionar el disco duro

Llegados a este punto hay que decidir qué espacio es el que se destinará a Debian dentro del disco duro. Es necesaria la creación de dos particiones: una de tipo Linux Swap (82), que debe ser de dos veces el tamaño de la memoria RAM de la cual dispone el ordenador, y una de tipo Linux (83) que será donde se guarden los datos.

Para todo el proceso de particionado se utilizará la herramienta “cfdisk”. Lo primero que se debe hacer es elegir el disco duro donde se quiere realizar la instalación.

Hay que tener en cuenta que los discos duros en Linux se nombran hdX siendo X el lugar donde se encuentra del bus (hda es primario master, hdb es primario slave, hdc es secundario master y hdd es secundario slave). En caso de tener discos duros SCSI la nomenclatura será similar sólo que siendo 'sd' en vez de 'hd'.

Una vez seleccionado el disco se creará la partición Swap. Esta partición, llamada también de intercambio, es la encargada de guardar los datos que no caben en un momento dado en la RAM.

Para la creación de esta partición se debe elegir la opción “NEW” y “PRIMARY”; de tamaño debe ser el doble del tamaño de la RAM del ordenador e indicar que se sitúe en el principio del disco (BEGINING). Una vez creada se debe indicar que se trata de una partición Linux Swap, para ello pulsando “TYPE” saldrá un menú con los posibles tipos de particiones, seleccionando 82 (Linux Swap).

Creada la partición de Swap, se creará una de tipo Linux (83) con el tamaño que se quiera destinar a este sistema operativo (el tamaño puede variar y aunque no es necesario, para configurar un entorno de escritorio más agradable e intuitivo es recomendable 1Gb). Se seleccionará sobre la partición de Linux que sea 'arrancable'. El aspecto que debe tener al acabar será como el siguiente (es posible que haya más particiones, por ejemplo con Windows, que serán de tipo vfat o ntfs).

```

cfdisk 2.11n

Disk Drive: /dev/sda
Size: 4294967296 bytes
Heads: 255 Sectors per Track: 63 Cylinders: 522

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
sda1                      Primary     Linux swap    320.79
sda2      Boot      Primary     Linux          3972.62

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ]   [ Type ]  [ Units ] [ Write ]
Wrote partition table to disk

```

Una vez creadas las particiones se debe escribir la tabla de particiones en disco pulsando “WRITE” y escribiendo 'yes' para confirmar. Al acabar de escribir la tabla de particiones se presionará “QUIT” para continuar.

6.3.7. Dar formato y activar una partición de intercambio

Primero el instalador preguntará si se quieren comprobar los sectores erróneos; este proceso es bastante tedioso, sobre todo en discos grandes, y no es necesario llevarlo a cabo excepto que se haya tenido un fallo con el disco recientemente. Una vez elegido si se desea hacer la comprobación de sectores, Debian GNU/Linux reconocerá todas las particiones de tipo Linux Swap que hay en el sistema y le pedirá al usuario que elija una de ellas para ser formateada. Una vez confirmada la elección el sistema pasará al formateo de la misma.

6.3.8. Selección del tipo de sistema de ficheros

En este punto se seleccionará el sistema de archivos que tendrá Linux. Ninguno de ellos es claramente mejor que los demás, cada uno tiene sus ventajas y sus inconvenientes. Las posibles elecciones son:

- **EXT2:** sistema de ficheros clásico de Linux, es el que se usa en instalaciones como “vanilla” o idepci. Su principal inconveniente es que no dispone de journaling con lo que un apagón puede hacer que se pierdan datos en el sistema o que éste quede inutilizable.
- **EXT3:** básicamente se trata de EXT2+Journaling, es decir, se utiliza EXT2 pero hay un registro de sucesos del sistema donde se indican las operaciones realizadas, los cambios sobre archivos, etc. El principal problema de este sistema de archivos es que es un poco más lento.
- **REISERFS:** nuevo sistema de ficheros transaccional más rápido que EXT2 y EXT3 pero también menos evolucionado y menos probado.

Una vez vistos los tres tipos de sistemas de ficheros a tener en cuenta, hay que elegir cuál es el que más conviene a cada uno. Para la instalación de este sistema se ha elegido EXT3 pero esto no influye en el resto de la instalación.

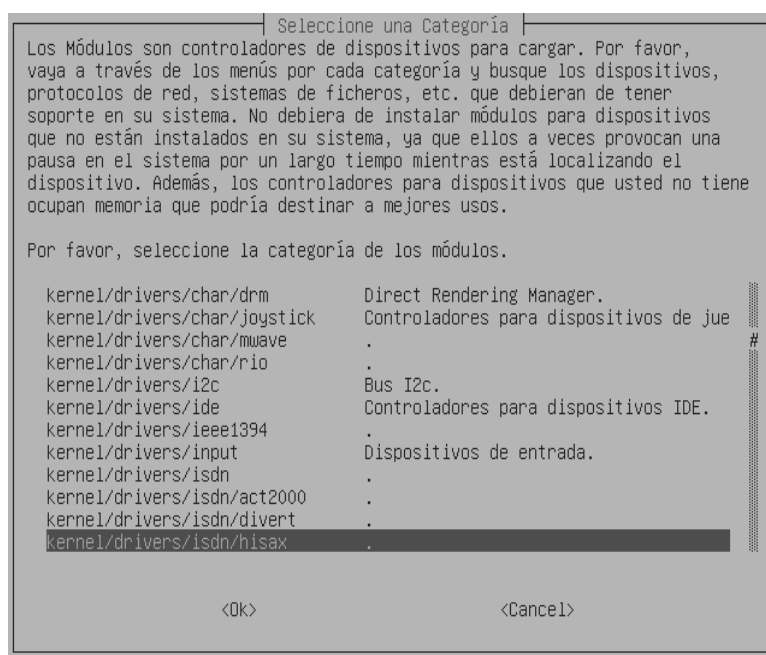
Acabada la elección, al igual que con la partición de Swap, preguntará si se quieren

revisar los bloques defectuosos y si se está seguro de formatear la partición de Linux. Acabado este formateo el sistema preguntará si se quiere que la partición que se ha creado se convierta en “/”² a lo que se responderá que si.

6.3.9. Elección de los módulos que se deben cargar en el sistema

Este paso es quizá el más importante para que el sistema que se está instalando funcione lo más correctamente posible.

Primero el sistema preguntará si para todo este proceso se quiere usar el CD de instalación (que es lo más común) o, si se está realizando una instalación desde disquetes, dará a elegir entre ftp, web, CD... Si se tiene una conexión rápida a Internet, puede ser interesante la instalación desde ftp o web, pero incluso en este caso lo más habitual es la instalación desde CD.



Llegados a este punto lo mejor para hacer una buena instalación es conocer bien el hardware. Cuanto mejor se conozca el ordenador en que se va a instalar Debian, se obtendrá un sistema más optimizado y mejor instalado. Podría ayudar mucho en este proceso el tener los manuales de la placa, las tarjetas de sonido, vídeo y red.

Los módulos se organizan en forma de árbol para una mejor comprensión y para que sea más fácil encontrarlos. Así, a partir de la raíz “kernel” salen cuatro ramas DRIVERS, FS, NET y ARCH.

- DRIVERS: donde se encuentran los controladores de los dispositivos.
- FS: donde se encuentran los sistemas de ficheros que puede reconocer el sistema (Jouliet, NTFS³, Fat32...).
- NET: donde se encuentran características de red como Irda (conexión por infrarrojos) o IPTABLES⁴.
- ARCH: donde se encuentran características de la arquitectura seleccionada (i386 en nuestro

² / es la raíz de la cual cuelgan todos los archivos y dispositivos de Linux

³ Sistema de ficheros usado en las versiones NT de Windows (NT4, 2000, XP etc.)

⁴ Cortafuegos a base de reglas muy utilizado en GNU/Linux

caso) como APM (Avanced Power Management) .

6.3.10. Módulos de uso habitual

Debido a la gran cantidad de módulos que existen en un kernel 2.4.18 sólo se explicarán los que se consideran más habituales.

- **/KERNEL/DRIVERS/CHART**

- AGP**

- AGPGART:** para ordenadores con AGP.

- DRM**

- Radeon:** permite activar el Direct Rendering Management en las tarjetas ATI Radeon.

- **/KERNEL/DRIVERS/IEEE1394**

En esta carpeta se encuentran todos los módulos de los chips de las tarjetas firewire que hay en la actualidad.

- **/KERNEL/DRIVERS/INPUT**

- MOUSEDEV:** permite al sistema tener ratón.

- JOYDEV:** da la posibilidad de usar Joystick.

- **/KERNEL/MEDIA/VIDEO**

BTTV: módulo de los chips de televisión bt848 y bt878. Este módulo es necesario para ver la televisión en Linux.

- **/KERNEL/DRIVERS/NET**

- PPP:** módulo necesario para la utilización del módem.

Tarjetas de red: en este apartado se encuentran todas las tarjetas que soporta el kernel 2.4.18 excepto las que ya están integradas en el sistema (como es el caso de la tan habitual RTL8139).

- **/KERNEL/DRIVERS/PARLPORT**

- PARLPORT:** permite usar el puerto paralelo.

- **/KERNEL/DRIVERS/SCSI**

- IDE-SCSI:** módulo necesario para grabar CD's en Debian.

TODOS LOS MODULOS SCSI: en esta rama del árbol están todos los drivers de los componentes SCSI soportados por Debian. Si en el sistema hay algún elemento SCSI debe buscarse aquí el módulo para que funcione.

- **/KERNEL/DRIVERS/SOUND**

TODAS LAS TARJETAS DE SONIDO: aquí se encuentran todos los módulos de las tarjetas de sonido soportadas. Son de uso bastante habitual los módulos btaudio (sonido para las tarjetas bttv), emu10k1 (soporte a Soundblaster Live), trident (tarjeta AC97 común en portátiles)...

- **/KERNEL/DRIVERS/USB**

PRINTER: soporte para impresoras USB.

SCANNER: soporte de scanners USB.

USB-OHCI y USB-UHCI: soporte para USB OHCI (que es el que montan Compaq y HP entre otros) y UHCI que es el más habitual.

USBCORE: módulo necesario para cualquier soporte de USB.

USBMOUSE: permite tener un ratón USB.

- **/KERNEL/DRIVERS/USB/STORAGE**

USBSTORAGE: módulo que permite tener almacenamiento USB y descargar fotografías de la mayor parte de cámaras digitales.

- **/KERNEL/DRIVERS/VIDEO**

RADEONFB: soporte de FrameBuffer para Ati Radeon.

- **/KERNEL/DRIVERS/VIDEO/RIVA**

RivaFB: soporte de FrameBuffer para tarjetas de Nvidia (Riva tnt, Gforce....).

- **/KERNEL/DRIVERS/FS**

BINFMT_A.OUT: soporte para archivos binarios A.out .

- **/KERNEL/DRIVERS/FS/NTFS**

Soporte de lectura para sistemas de ficheros NTFS.

- **/KERNEL/NET/NETFILTER**

En esta rama se encuentran todos los módulos de IPTABLES.

- **/KERNEL/ARCH/i386/KERNEL/**

APM: Advanced Power Management. Permite que el ordenador hiberne, se apague sólo,...

6.3.11. Configuración de la red

Llegado a este paso se va a proceder a la instalación de la red. El primer paso es elegir el nombre con el que se conocerá el equipo.

Una vez elegido el nombre del equipo, el instalador preguntará si estamos trabajando con un DHCP, en cuyo caso la instalación de la red habría acabado aquí. En caso contrario, el sistema preguntará primero la IP, la máscara de red, la pasarela (o gateway) si es que la hubiera, el nombre del dominio y por último de las DNS (estos datos son datos que debería proporcionarle el administrador de la red a la que esté conectado).

6.3.12. Instalación del sistema base

Acabados los pasos anteriores éste es el momento en que se procederá a ejecutar la instalación, con las opciones escogidas, en el disco duro. Para ello se elegirá el método de instalación más adecuado (si se dispone del CD se recomienda la instalación desde él).



6.3.13. Creación del disco de arranque

Acabada la instalación de paquetes básicos, el siguiente paso es la creación de un disco de arranque que podrá ser útil en caso de la instalación de otro sistema operativo como Windows.

6.3.14. Actualización del arranque

Acabada la primera fase de la instalación se debe escribir la lista de sistemas operativos arrancables. En este punto, como en muchos otros, lo mejor es hacer caso al instalador y escribir en el MBR el Lilo. Finaliza la primera parte de la instalación y es necesario reiniciar.

6.3.15. Configuración de la Zona Horaria

Tras haber reiniciado, lo primero que se debe hacer es configurar la zona horaria. España se encuentra en el GMT+01 por lo que a la primera pregunta de si se quiere que la hora del sistema sea GMT se debe decir que no. Después, se debe seleccionar Europa y luego Madrid.

6.3.16. Configuración de Passwords

Este punto es bastante importante en cuanto a seguridad del sistema se trata. Lo más seguro y recomendable es activar los passwords en md5 y el uso de “shadow”.

6.3.17. Elección del password de root y creación de una cuenta de usuario

En este momento se debe decidir qué clave poner al usuario root. Esta clave debe ser una clave larga, no ser de diccionario (no obtenerse a partir de cambios de palabras comunes) y ser conocida por el menor número de personas posible. Quizá las claves sencillas sean el principal problema de seguridad de los sistemas, si un usuario malintencionado averigua esta clave todos los sistemas de seguridad que se usen serán insuficientes.

Una vez cambiada la clave del administrador (o root), es recomendable la creación de un usuario para uso habitual. Cuanto más se use la cuenta de root, ésta estará más en peligro.

6.3.18. Configuración de APT

Lo primero que se debe hacer al hablar del APT es hacerle una digna presentación. APT es la aplicación más importante y más carismática de Debian. Con esta aplicación es posible la instalación de un paquete con un solo comando de consola.

Sobre la configuración, es recomendable no utilizar el ppp para ella (siempre habrá tiempo para usar ppp⁵). También es recomendable el añadir todos los CDs de los cuales se disponga con paquetes .deb y si se dispone de conexión directa a Internet (por una red corporativa o una ADSL) también es recomendable añadir algún source http o ftp (una buena elección puede ser rediris.es). Si en la actualidad no dispone de Internet o se va a conectar usando ppp, no se preocupe, la utilidad apt-setup permite modificar esta lista de sources y también es posible editarla a mano en '/etc/apt/sources.list'.

```
deb http://bulmalug.net/~daneel/debian/ ./
#deb http://people.debian.org/~ccheney/kde-other/ ./
#deb http://people.debian.org/~ccheney/kde-3.1.0-1/ ./
#deb http://people.debian.org/~bab/kde3.1/ ./
carmen@Framereley:~$ cat /etc/apt/sources.list
```

Acabados de añadir todos los sources que se crean convenientes, el sistema pregunta si se quiere añadir "security". Security es el lugar donde se ponen las revisiones de seguridad de los paquetes estables de Debian. Si hay conexión directa a Internet es altamente recomendable añadirlo; si no se dispone de conexión directa se dirá que no, pero cuando se disponga de Internet es muy recomendable modificar el sources y añadir "security".

6.3.19. TaskSell y Dselect

Estas dos aplicaciones son las encargadas de hacer más fácil la elección de paquetes de Debian.

TaskSell es más general que Dselect, mientras TaskSell elige grupos de paquetes como X window system, games, DNS server,... Dselect elige paquetes individuales como gcc, less, make,...

Como recomendación, un inexperto debería usar TaskSell frente a Dselect debido a que, si se hace mal con Dselect, se pueden olvidar paquetes necesarios y el sistema puede no funcionar bien, en cambio, con TaskSell, al elegirse grupos de paquetes en vez de paquetes individuales, ésto no pasará.

5 Point To Point- Conexión punto a punto, la más habitual es la conexión por MODEM



En TaskSell es recomendable activar X window system (para poder tener entorno gráfico), C y C++ para compilar sus programas, el kernel y spanish environment para tener manuales en castellano.

En Dselect es recomendable leer con calma los paquetes y seleccionar los que se vean necesarios.

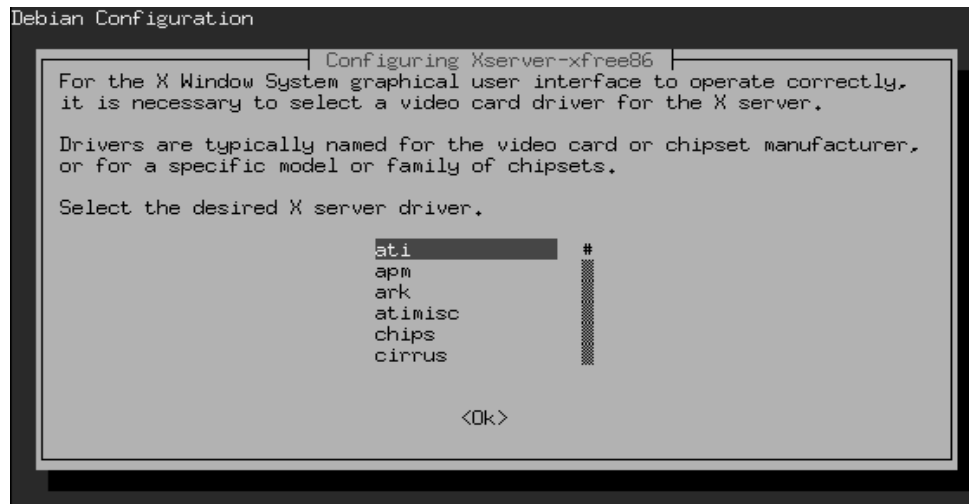
6.3.20. Configurar el Xfree86

- Configuración de la Tarjeta Gráfica

Al seleccionar el X window system saldrá un nuevo menú de configuración que permitirá la configuración del entorno gráfico.

El primer paso en esta configuración es la confirmación de si se quiere usar el debconf (utilidad de Debian para la configuración de los paquetes) para editar el archivo de configuración del entorno gráfico. Se debe contestar que "Sí".

El siguiente paso es la elección del driver para la tarjeta de vídeo. El configurador muestra una serie de drivers y se debe elegir el que sea más adecuado.



Los más utilizados son:

- **ati** – Utilizado por la mayoría de las tarjetas gráficas ATI.
- **nv** – Driver libre utilizado para la mayoría de tarjetas Nvidia.
- **Vesa** – Es la configuración que se debe escoger si no se sabe qué elegir. Esta configuración suele funcionar con todas las tarjetas gráficas.

Elegido el controlador más adecuado, el configurador pide un nombre para identificar esta tarjeta frente a otras en el caso de que usted disponga de más de una (caso poco habitual).

Seleccionado el driver y el nombre el sistema, pregunta datos sobre la tarjeta gráfica que, aunque no necesarios, sí que pueden mejorar el rendimiento del sistema.

La primera de esas preguntas es para saber dónde se encuentra la tarjeta gráfica dentro del bus del sistema. Ésto es algo que un usuario normal no conoce, por lo que es recomendable dejarlo en blanco y, una vez instalado el sistema, ejecutar el programa `lspci` (que describe todos los componentes del sistema) y posteriormente el programa `dpkg-reconfigure` (utilidad de Debian para reconfigurar los paquetes) y reconfigurar la utilidad.

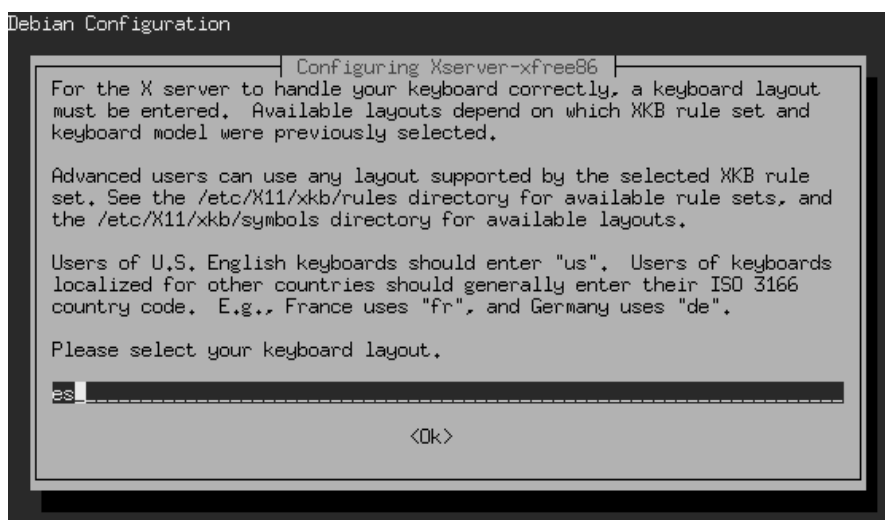
La siguiente pregunta es mucho más fácil de responder, se trata del número de Kb de que dispone la tarjeta gráfica.

La última pregunta acerca de la tarjeta gráfica es si se quiere usar Framebuffer o no. La respuesta variará según la tarjeta gráfica que se tenga, la mayoría de las tarjetas actuales disponen de Framebuffer, por lo que se debería decir que “Sí”. Si la tarjeta es antigua o no se sabe si soporta el framebuffer se debería responder que “No” para asegurar unas X que funcionen.

• Configuración del teclado

El primer paso es elegir, como ocurrió en el caso de la tarjeta gráfica, el driver del teclado. Si en la tarjeta gráfica había dudas, aquí no las hay: casi todos los teclados que tienen los ordenadores ix86 utilizan el driver “xfree86”.

Elegido el driver hay que elegir el número de teclas que tiene el teclado. Por lo habitual los teclados tienen entre 101 y 105 teclas (las teclas especiales no cuentan), con lo que se debe elegir `pcXXX`, siendo XXX el número de teclas.



Ahora lo que hay que seleccionar es el idioma del teclado; para español se debe seleccionar 'es', si por el contrario se tiene un teclado inglés se debe escoger 'us'.

El siguiente paso, y último en la configuración del teclado, es la elección de características adicionales al teclado. Por lo habitual se dejará en blanco.

- Configuración del ratón

En este punto se llega a la mitad de la instalación del entorno gráfico en Debian. La configuración del ratón empieza por la elección de dónde se encuentra el ratón. Por lo habitual, se debe elegir:

- **/dev/psaux** – si el ratón está conectado al PS2 (exteriormente es una circunferencia).
- **/dev/input/mice** – si el ratón está en el puerto USB.
- **/dev/ttyX** - si el ratón está conectado al puerto COM número X (exteriormente es similar a un óvalo).



Situado el ratón, se debe, como ya ocurrió con el teclado y la tarjeta gráfica, elegir el driver que éste utiliza. Por defecto se debe escoger PS/2 o ImPS/2 que son los protocolos más habituales.

Elegido el driver, el configurador pregunta dos características del funcionamiento del ratón en el sistema. La primera de estas preguntas es si se quiere activar el scroll en el sistema (si su

ratón dispone de scroll debería activarlo) y si se quiere emular el tercer botón (debe activarse en ratones que sólo tienen dos botones debido a la gran utilidad del tercer botón en Linux).

- Configuración del monitor

El primer paso en la configuración del monitor es la elección de un nombre para éste. Elegido ese nombre el sistema pregunta si se debe considerar que el sistema tiene una tarjeta LCD. Si se tiene una pantalla TFT o HFT se debe decir que sí.

Elegido el tipo de pantalla hay que configurar la resolución de la pantalla. Primero se seleccionará el modo de configuración de esta parte. Se puede elegir “simple”, “media” o “avanzada”. Por defecto, la elección más adecuada, a menos que se sepa mucho de la pantalla que se dispone, es “media”. Seleccionado este modo de configuración, se elegirá la resolución máxima del monitor.

El siguiente paso en la configuración es seleccionar las resoluciones a las que se quiere que el sistema funcione. Es conveniente que, a mayores de la que se quiere utilizar, se seleccione una resolución baja para tener siempre una configuración que no falle.

El último de los pasos de la configuración de la pantalla es la elección del número de colores que debe usar el sistema. Aquí se recomienda no usar más de 16 bits, ya que ello a veces ocasiona problemas.

- Pasos finales de la configuración

Acabadas las configuraciones de teclado, ratón, tarjeta gráfica y monitor se deben elegir los módulos que debe cargar el sistema. Aunque asuste un poco, la configuración por defecto de Debian es correcta por lo que no tiene porque modificarse.

Finalizada esta configuración, el sistema pide modificar los archivos del DRI y del Xfree86 a lo que, como es lógico, se debe decir que sí.

Si por alguna razón el entorno gráfico no funcionase a la primera, se puede modificar esta configuración ejecutando como super-usuario “dpkg-reconfigure xserver-xfree86”.

6.3.21. Configuración de EXIM

Excepto que se quiera montar un servidor de correo en el equipo (y en cuyo caso se recomendaría otros servidores de correo) es recomendable no configurar Exim (presionando 5-noconfiguration). Si en algún momento se quiere cambiar de idea siempre se podrá reconfigurar con 'dpkg-reconfigure exim'

Si se ha llegado a este punto, enhorabuena, ya ha instalado Debian.

6.4. Utilidad APT

6.4.1. Introducción

Para usar el APT de Debian, como mínimo se necesitan dos cosas: una dirección para el sourcelist y algo para descargar.

Lo primero que hay que aprender es que las sourcelist son direcciones que apuntan a servidores ftp, http o a algún otro sitio local como un disco duro o un lector de CDs; contienen paquetes deb los cuales pueden ser bajados de la forma que se verá en los apartados siguientes. Estas direcciones se guardan en el archivo /etc/apt/sources.list. Cada vez que se añada alguna dirección nueva a este archivo se debe ejecutar el comando 'apt-get update' como súper usuario (root). Este comando básicamente lo que hace es bajar la lista de paquetes disponibles en la nueva sourcelist que se ha añadido.

Existen multitud de aportaciones de toda la comunidad que crean ".deb" y los suben a sus propios servidores. En estos servidores hay programas en formato deb que por alguna razón no están oficialmente en los servidores de Debian. Hay que tener una cosa en cuenta, sólo las sourcelist de Debian son oficiales, estos otros son aportaciones de la comunidad y normalmente, aunque son totalmente fiables, no son responsabilidad de Debian.

La lista de mirrors o “espejos” oficiales de Debian se puede consultar en la dirección <http://www.debian.org/mirror/list> .

6.4.2. Las sourcelist

Las direcciones de las sourcelist se componen básicamente de una dirección de Internet, la versión de tu debian (estable, inestable o desarrollo), y los directorios que se quieren añadir a la lista de paquetes.

Ejemplo:

```
deb http://XX.debian.org/debian stable main contrib non-free
```

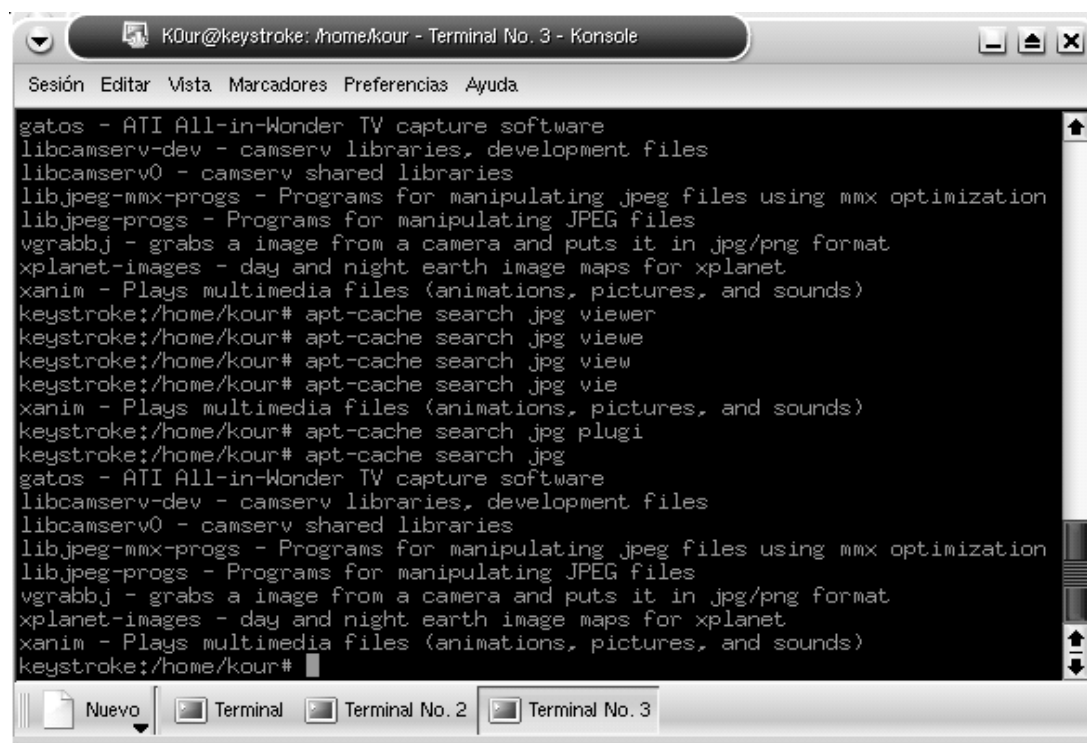
"XX" es el código del país, como por ejemplo "es" (España), "fr" (Francia) o "us" (Estados Unidos). La opción "stable" es la versión de tu debian, siendo "stable" la versión estable en el momento, "unstable" la versión inestable y "testing" la versión de desarrollo. “main contrib non-free” son los directorios los cuales se quiere que apt añada en su lista. Un último punto, la opción deb se refiere al tipo de paquetes, es decir, si se pone deb descargará binarios deb y si es sustituido por deb-src lo que descargará entonces serán los códigos fuente del paquete que le pidamos.

6.4.3. Usando APT

Aunque en esta introducción al apt no se pretende hacer un exhaustivo análisis al conjunto de aplicaciones apt, se tratarán de mostrar algunos de los muchos, y los más usuales, modos de uso de esta herramienta.

6.4.4. Buscando paquetes

Si lo que se necesita es buscar un paquete, la opción que se usa es "apt-cache search paquete-que-se-busca" si encuentra algo mostrara una lista como por ejemplo podría ser esta:



The screenshot shows a terminal window titled 'K0ur@keystroke: /home/kour - Terminal No. 3 - Konsole'. The window has a menu bar with 'Sesión', 'Editar', 'Vista', 'Marcadores', 'Preferencias', and 'Ayuda'. The terminal content shows a list of packages and their descriptions, followed by several 'apt-cache search' commands and their outputs. The packages listed include 'gatos', 'libcamserv-dev', 'libcamserv0', 'libjpeg-mmx-progs', 'libjpeg-progs', 'vgrabbbj', 'xplanet-images', and 'xanim'. The commands and their outputs are as follows:

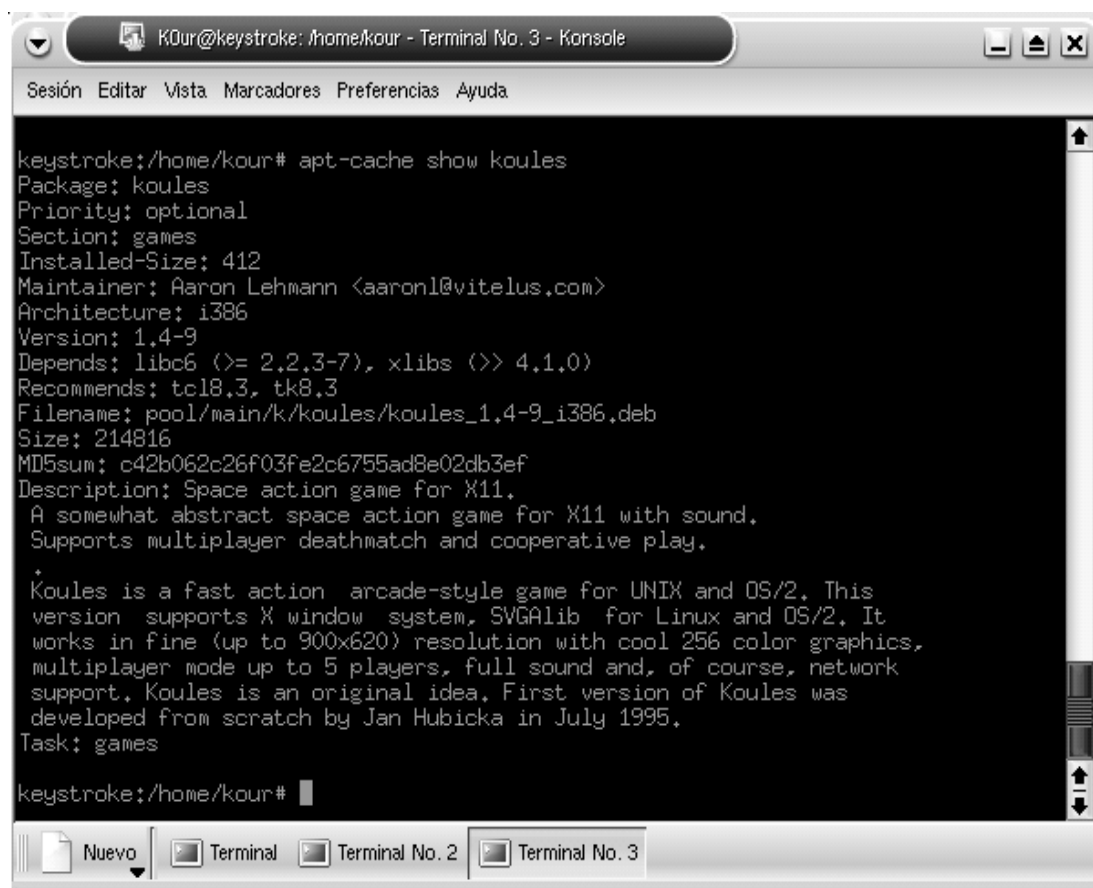
```
gatos - ATI All-in-Wonder TV capture software
libcamserv-dev - camserv libraries, development files
libcamserv0 - camserv shared libraries
libjpeg-mmx-progs - Programs for manipulating jpeg files using mmx optimization
libjpeg-progs - Programs for manipulating JPEG files
vgrabbbj - grabs a image from a camera and puts it in jpeg/png format
xplanet-images - day and night earth image maps for xplanet
xanim - Plays multimedia files (animations, pictures, and sounds)
keystroke:/home/kour# apt-cache search jpg viewer
keystroke:/home/kour# apt-cache search jpg viewe
keystroke:/home/kour# apt-cache search jpg view
keystroke:/home/kour# apt-cache search jpg vie
xanim - Plays multimedia files (animations, pictures, and sounds)
keystroke:/home/kour# apt-cache search jpg plugi
keystroke:/home/kour# apt-cache search jpg
gatos - ATI All-in-Wonder TV capture software
libcamserv-dev - camserv libraries, development files
libcamserv0 - camserv shared libraries
libjpeg-mmx-progs - Programs for manipulating jpeg files using mmx optimization
libjpeg-progs - Programs for manipulating JPEG files
vgrabbbj - grabs a image from a camera and puts it in jpeg/png format
xplanet-images - day and night earth image maps for xplanet
xanim - Plays multimedia files (animations, pictures, and sounds)
keystroke:/home/kour#
```

At the bottom of the terminal window, there is a tab bar with four tabs: 'Nuevo', 'Terminal', 'Terminal No. 2', and 'Terminal No. 3'. The 'Terminal' tab is currently selected.

En esta lista se muestra el paquete y a continuación una pequeña descripción. Si se escoge uno de esos paquetes para instalar se procederá al siguiente paso.

6.4.5. Mirar el contenido de un paquete

Una vez escogido un paquete interesante es posible que se quiera saber más acerca de él para ver si es lo que se está buscando o no. Para ello, lo que se debe hacer es ejecutar el comando 'apt-cache show paquete_interesante' con lo que te dará la información del paquete. Como ejemplo supongamos que se quiere ver si nos puede interesar la instalación del "koules":



```
KOur@keystroke: /home/kour - Terminal No. 3 - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda

keystroke:/home/kour# apt-cache show koules
Package: koules
Priority: optional
Section: games
Installed-Size: 412
Maintainer: Aaron Lehmann <aaronl@vitelus.com>
Architecture: i386
Version: 1.4-9
Depends: libc6 (>= 2.2.3-7), xlibs (> 4.1.0)
Recommends: tcl8.3, tk8.3
Filename: pool/main/k/koules/koules_1.4-9_i386.deb
Size: 214816
MD5sum: c42b062c26f03fe2c6755ad8e02db3ef
Description: Space action game for X11.
 A somewhat abstract space action game for X11 with sound.
 Supports multiplayer deathmatch and cooperative play.

 *
 Koules is a fast action arcade-style game for UNIX and OS/2. This
 version supports X window system, SWGAlib for Linux and OS/2. It
 works in fine (up to 900x620) resolution with cool 256 color graphics,
 multiplayer mode up to 5 players, full sound and, of course, network
 support. Koules is an original idea. First version of Koules was
 developed from scratch by Jan Hubicka in July 1995.
Task: games

keystroke:/home/kour#
```

6.4.6. Instalando paquetes

Si lo que se quiere es instalar el “xkoules” lo único que hay que hacer es ejecutar el comando “apt-get install koules” (como root). Éste proporcionará información sobre qué otros programas debe descargar para instalarlo (dependencias) y le preguntará si quiere que lo haga él automáticamente.

6.4.7. Desinstalar un paquete

Si una vez usado un paquete éste se quiere desinstalar, lo que se debe hacer se divide en dos pequeños pasos: el primero, que es averiguar el nombre del paquete, que bien se hará buscándolo con “apt-cache search XXX” o bien se hará confiando en la memoria del superusuario, y un segundo paso, propiamente el de desinstalación, que consiste en ejecutar “apt-get remove paquete_a_borrar”.

Cuando se desinstala un paquete lo que no lo hace son sus dependencias, es por ello que de vez en cuando es útil la ejecución del comando “deborphan” que busca todas aquellas librerías que no necesita ningún programa y permite desinstalarlas.

6.4.8. Actualizar los paquetes y los sources

Siempre que se añada una línea al sources.list es necesario indicarle al sistema que debe volver a buscar en los repositorios en busca de nuevas aplicaciones. Ésto se hace ejecutando como superusuario el comando “apt-get update”.

Además de cuando se añade un paquete, es recomendable el ejecutar este comando

periódicamente debido a que los paquetes en las ramas inestable y en pruebas se actualizan todos los días y en la estable, muy de vez en cuando, se añaden modificaciones de software por errores de seguridad.

Para que el sistema actualice las versiones de un paquete, o bien para que se actualice el sistema de versión en versión es necesario escribir 'apt-get update && apt-get upgrade'.

CAPÍTULO 7

Compatibilidad de Linux con otros sistemas operativos

Diego Pérez Montes¹

7.1. Introducción

Actualmente la mayoría de los PC's domésticos y estaciones de trabajo utilizados en empresas siguen usando Windows. En la convivencia entre Linux y Windows será en lo que se basará este capítulo, ya que ambos pueden vivir en perfecta armonía en la misma máquina/red.

Trataremos de rebatir una de las excusas más comunes que la mayoría de la gente plantea para no cambiarse a Linux: “es que si uso Linux me quedo aislado del resto de la gente que usa Windows”.

7.2. Accediendo a particiones Windows

Totalmente necesario para la coexistencia de los dos sistemas en la misma máquina es este apartado en el que se explicará el funcionamiento básico de un comando fundamental en cualquier sistema *NIX como es **mount**. Esta pequeña aplicación nos servirá para poder acceder a los ficheros de nuestra partición, sin importarnos el sistema de ficheros en el que se encuentre (ya que o será detectado o podremos pasárselo como parámetro).

- **Sintaxis elemental de mount:**

mount dispositivo(o partición) destino -t sistema_de_ficheros

Los sistemas de ficheros que necesitaremos conocer en este caso son:

- msdos: usado básicamente en Windows 9x/Me, incluye tanto FAT16 como FAT32.
- ntfs: usado por Windows NT/2000/XP.

De una forma más práctica para el caso que nos ocupa, mostraremos como ejemplo cómo montar una partición Windows 9x que se encuentra en la primera partición del disco duro, con sistema de ficheros FAT32, en el directorio */win* :

#mount /dev/hda1 /win -t msdos

¹ Grupo de Usuarios de Linux de Ourense

De esta manera podríamos operar en el directorio como en cualquier otro en nuestro sistema, aunque se debe advertir que los sistemas de ficheros NTFS pueden ser leídos, pero en cuanto a la escritura se refiere, aunque está permitida, puede causar daños irreversibles en nuestro sistema de ficheros (NTFS). Se está trabajando en un mejor soporte de escritura que se incluye en la versión 2.6 del kernel.

Si deseáramos ver los sistemas de ficheros mostrados, bastará con ejecutar *mount* sin ningún parámetro y nos mostrará una salida similar a la siguiente:

```
$ mount
```

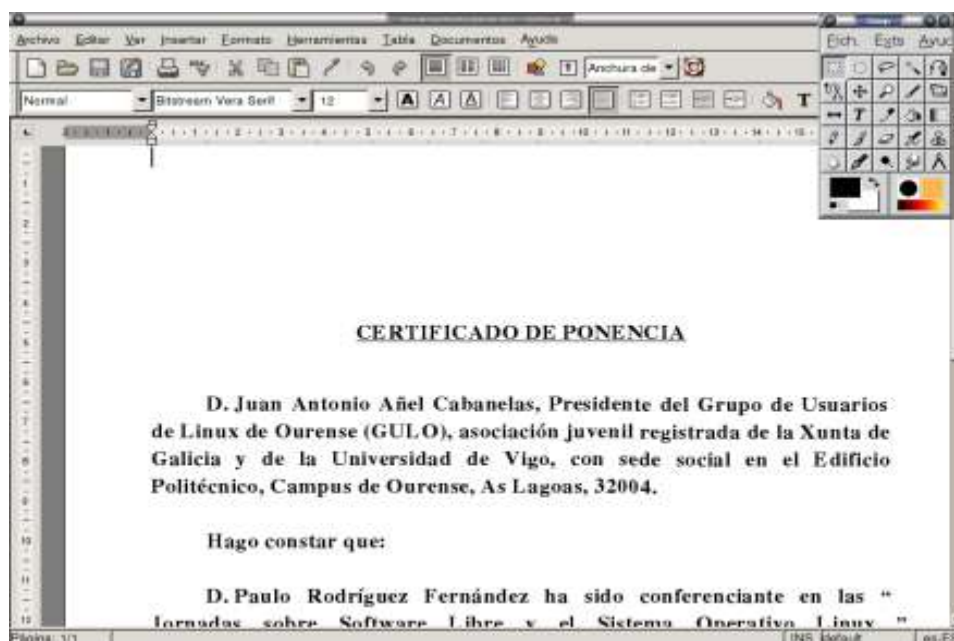
```
/dev/hda3 on / type ext3 (rw,errors=remount-ro)  
/dev/hda1 on /win type msdos (rw,errors=remount-ro)
```

```
...
```

En la segunda línea de la salida podemos ver nuestra partición (/dev/hda1) montada en el directorio /win con un sistema de ficheros msdos (FAT).

7.3. Migración de paquetes y aplicaciones

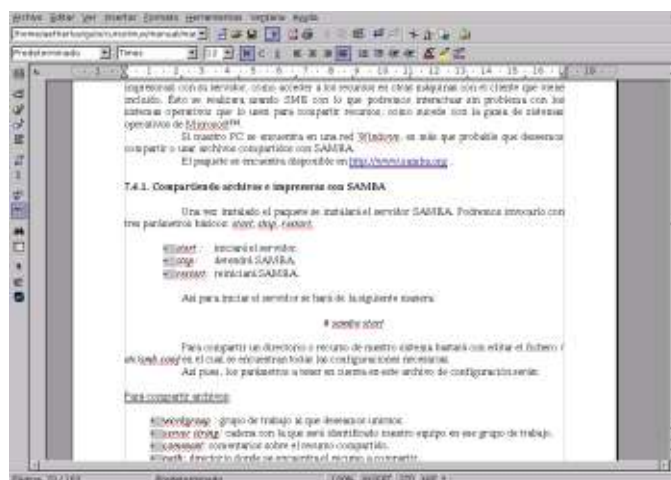
7.3.1. Abiword



Se trata de un procesador de texto What You See Is What You Get (WYSIWYG. Lo que ves es lo que consigues), es decir, que lo que se muestra en pantalla es lo que se logrará al realizar una salida por la impresora, de forma similar a Microsoft® Word; de hecho, no nos resultará difícil migrar todos nuestros documentos de la aplicación de Windows a AbiWord.

Se puede descargar e instalar fácilmente de <http://www.abisource.com/> y se encuentra disponible tanto para Linux como para Windows.

7.3.2. OpenOffice



OpenOffice es una suite de herramientas de ofimática basada en el código liberado por Sun de su conocida “hermana” StarOffice. El código de StarOffice fue liberado bajo licencia GPL, por lo que el código ha sido modificado y adaptado gracias a esto. OpenOffice se puede usar para cualquier fin, aun siendo comercial, sin tener que pagar por ella, mientras que por StarOffice se debe pagar. Todas las utilidades que componen el paquete traen útiles de migración que nos permitirán fácilmente reutilizar nuestros documentos escritos con la mayoría de los estándares de otras empresas (Microsoft, Apple,...) y sus paquetes de ofimática. Entre estas utilidades podemos destacar un procesador de texto (What You See Is What You Get), una sencilla hoja de cálculo, un paquete de creación de base de datos básicas y una aplicación para realizar presentaciones.

OpenOffice se puede conseguir en <http://www.openoffice.org> y StarOffice en <http://www.sun.com/staroffice/>. Ambas se encuentran disponibles igualmente para Linux y para Windows.

7.3.3. GQview

Nos encontramos ante una utilidad que nos permitirá ver imágenes y llevar a cabo una gran variedad de trabajos sobre ellas.

Este programa les resultará familiar y sencillo de usar a todos aquellos que en Windows han usado ACDSEE. Se puede descargar de <http://gqview.sourceforge.net>

7.3.4. Mplayer



Con mplayer podremos visualizar vídeos ya que soporta la mayoría de los codecs y formatos. Es además totalmente personalizable, y cuenta con una gran variedad de temas. La dirección de su web de descarga es <http://www.mplayerhq.hu>.

7.3.5. XMMS

X Multimedia System es un reproductor cuya imagen recuerda mucho a WinAMP. De hecho su uso más extendido es el de reproducir MP3, aunque puede reproducir una amplia gama de formatos de audio. Tiene así mismo una gran cantidad de temas para cambiar su aspecto y soporta temas de otros reproductores (incluyendo los del anteriormente mencionado WinAMP). Esta disponible en <http://www.xmms.org>.



7.4. SAMBA

SAMBA es un paquete que nos permitirá tanto compartir recursos (archivos o impresoras) con su servidor, como acceder a los recursos en otras máquinas con el cliente que viene incluido. Ésto se realizara usando SMB con lo que podremos interactuar sin problema con los sistemas operativos que lo usen para compartir recursos, como sucede con la gama de sistemas operativos de Microsoft™.

Si nuestro PC se encuentra en una red Windows, es más que probable que deseemos compartir o usar archivos compartidos con SAMBA.

El paquete se encuentra disponible en <http://www.samba.org>.

7.4.1. Compartiendo archivos e impresoras con SAMBA

Una vez instalado el paquete se instalará el servidor SAMBA. Podremos invocarlo con tres parámetros básicos: *start*, *stop*, *restart*.

- *start* : iniciará el servidor.
- *stop*: detendrá SAMBA.
- *restart*: reiniciará SAMBA.

Así para iniciar el servidor se hará de la siguiente manera:

```
# samba start
```

Para compartir un directorio o recurso de nuestro sistema bastará con editar el fichero */etc/smb.conf* en el cual se encuentran todas las configuraciones necesarias.

Así pues, los parámetros a tener en cuenta en este archivo de configuración serán:

Para compartir archivos:

- *workgroup* : grupo de trabajo al que deseamos unirnos.
- *server string*: cadena con la que será identificado nuestro equipo en ese grupo de trabajo.
- *comment*: comentarios sobre el recurso compartido.
- *path*: directorio donde se encuentra el recurso a compartir.
- *valid users* : usuarios con acceso.
- *browseable*: permiso de exploración del directorio.
- *writable*: permiso de escritura en el recurso.
- *public*: configuración del acceso anónimo.
- *create mode*: permisos globales (funcionan igual que los del sistema).

En cuanto a compartir impresoras se refiere:

- *print ok*: sitúa la impresora como lista.
- *printer name*: nombre con el que aparecerá compartida la impresora.
- *path*: directorio del spool.
- *valid users*: usuarios con acceso permitido.
- *browseable*: permiso de exploración de archivos en la cola de impresión.

El fichero de configuración instalado por defecto ya de por sí es totalmente funcional y trae suficientes ejemplos para su puesta en marcha, no obstante, nunca viene mal una pequeña ayuda y un ejemplo básico.

#Segmentos del fichero de configuración smb.conf

#Como compartir un archivo

[global]

#Grupo de trabajo al que unirn

workgroup=GULO # Podremos poner cualquier nombre de grupo

#Descripción del Equipo

server string=Equipo de Prueba # Cadena de texto con la que seremos
identificados dentro del grupo de trabajo elegido

...

...

[directorio] # Nombre que queremos dar al recurso compartido
comment=Compartido # Comentario del recurso compartido
path=/home/compartido # Ruta del directorio que queremos compartir
valid users=gulo # Usuarios permitidos a acceder al recurso compartido
browseable=yes # Aquí especificamos si es navegable o no el recurso
public=no # Especificamos si el usuario Invitado se permite
writeable=no # Permiso de escritura en nuestro recurso
create mode= 0700 # Permisos

...

De esta manera tendremos compartido cara a nuestra red el directorio “/home/compartido” como “Compartido”.

Compartir una impresora se podría hacer incluyendo en la configuración algo similar a lo siguiente:

#Como compartir Impresora

[impresora]

print ok=yes
printer name=nombreimpresora #Nombre de la Impresora
path= /home/gulo #Directorio
valid users=gulo #Usuarios permitidos
browseable=yes #Navegable

7.4.2. Utilidades de SAMBA

Ahora que nuestro equipo ya sirve ficheros con SAMBA, ¿por qué no ver qué tienen las máquinas de nuestra red compartido?. Para este fin existen varias utilidades, algunas contenidas en el paquete SAMBA y otras que no, pero fáciles de encontrar y utilizar.

7.4.2.1. smbclient

Herramienta básica contenida en el paquete SAMBA. De funcionamiento similar a un cliente de FTP, nos permitirá con una serie de operaciones básicas enviar/recibir ficheros usando SMB. Su sintaxis básica es:

```
$ smbclient recurso -U usuario
```

Si por ejemplo quisiéramos acceder al recurso compartido *compartido* en el equipo con ip 10.0.0.1 y el usuario popo, bastaría con ejecutar:

```
$ smbclient //10.0.0.1/compartido -U popo
Password:*****
smb: \>
```

Con esto tendríamos ya conectado nuestro cliente a su servidor de ficheros SMB en el que podríamos realizar diversas operaciones. Entre las cuales se encuentran las básicas:

<i>get fichero</i>	copia fichero del servidor
<i>put fichero</i>	envía fichero
<i>del fichero</i>	borra fichero deseado
<i>help</i>	muestra todos los comandos disponibles en el cliente
<i>quit</i>	sale del cliente

7.4.2.2. Utilidades del sistema de ficheros SMB

Estos útiles no se encuentran contenidos en el paquete básico SAMBA, pero sí en el paquete smbfs, fácilmente encontrable en <http://www.samba.org>.

- **smbmount** : esta herramienta tiene la misma función que *mount -t smbfs*, pero con muchas mas opciones orientadas al sistema de ficheros SMB. Esencialmente nos servirá para montar un recurso compartido en un directorio local y así manejarlo de manera más natural que con smbclient. Su sintaxis básica es:

```
# smbmount recurso_compartido punto_de_montaje -o username=usuario
Password:*****
```

Remitiéndonos al ejemplo usado con smbclient, es decir vamos a montar en nuestro directorio local /home/compartido el recurso compartido *compartido* de la máquina con ip 10.0.0.1, con usuario *popo* de la siguiente manera:


```
# smbmount //10.0.0.1/compartido /home/compartido -o username=popo  
Password:*****
```

Ahora ya tenemos montado el directorio y podremos operar en él como si se tratara de un directorio local a todos los efectos, con la salvedad de que estaremos limitados en cuanto a permisos a los asignados por el servidor.

- **smbumount:** después de que ya no nos sea necesario el uso del recurso compartido y montado con smbmount, es imprescindible para el buen funcionamiento de la red y de nuestro sistema deshacernos del directorio montado para evitar problemas. Así pues esta utilidad, opuesta de la anterior, nos permitirá desmontar el directorio local y liberar así la sesión en el servidor. Su sintaxis es bastante sencilla:

```
# smbmount punto_de_montaje
```

Así en el ejemplo usado bastaría con ejecutar:

```
# smbmount /home/compartido
```

7.4.3. En modo gráfico

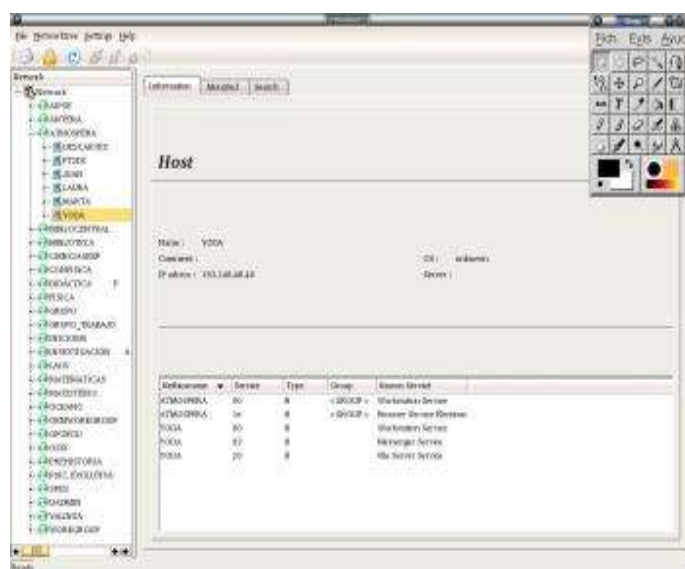
7.4.3.1. SWAT

SWAT es una utilidad que nos permitirá configurar SAMBA vía web. Su interfaz es totalmente intuitiva. Ejecutando *swat* en la línea de comandos se iniciará. El puerto donde se coloca el servicio por defecto es el 901, con lo que bastará introducir la dirección para acceder a él. Si quisiéramos acceder a nuestro equipo (local), abriríamos <http://127.0.0.1:901> en nuestro navegador y visualizaremos una hoja que nos guiará a través de la configuración básica.

Como la mayoría de las utilidades relacionadas con SAMBA, se puede encontrar en <http://www.samba.org>

7.4.3.2. Komba2

Aunque este “navegador de recursos compartidos” viene usualmente asociado al gestor KDE, se puede bajar de <http://komba.sourceforge.net> y usarlo sin necesidad de tener las KDE (eso sí, necesitaremos las librerías Qt+). Komba2 nos permitirá ver los recursos compartidos dentro del rango de IP's que le especifiquemos en las opciones. Una vez seleccionado un recurso lo montará automáticamente con **smbmount** en el directorio deseado.

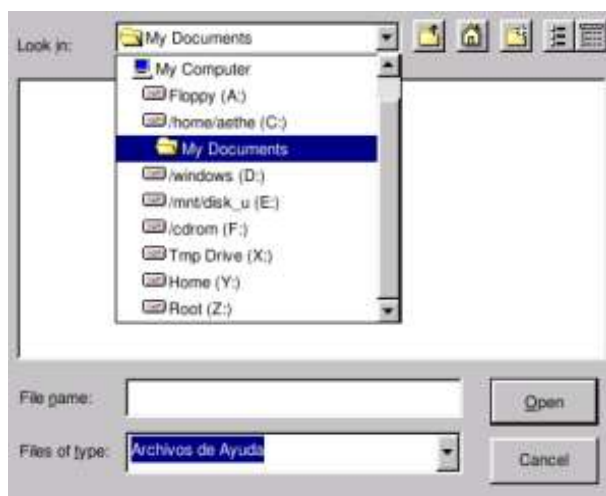


7.5. Ejecutando aplicaciones de otros sistemas operativos

7.5.1. WINE

WINE es una implementación de la API (una API no es más que una serie de servicios o funciones que el sistema operativo ofrece al programador, como por ejemplo, imprimir un carácter en pantalla, leer el teclado, escribir en un fichero de disco, etc.) de Windows, con lo que podremos ejecutar aplicaciones Windows con él en X. Se puede conseguir en <http://www.winehq.com/>

Con WINE podremos ejecutar prácticamente cualquier aplicación Windows, incluyendo juegos.



Su configuración es sencilla y bastará con modificar el fichero `/etc/wine/wine.conf`. La configuración por defecto funciona en su mayoría y bastará con retocar algunas líneas, en concreto:

```
[Drive C]
path=/win           #Directorio de Windows en el sistema de ficheros actual
Type=hd             #Tipo (hd,floppy,network)
Label=Windows       #Nombre
Filesystem=win95     #Sistema de ficheros (msdos,win95,unix)
```

Simplemente modificando estas líneas, WINE se encargará del resto. La mayoría de las utilidades están soportadas, aunque hay que decir que no se llevan demasiado bien con el sistema de ficheros NTFS, por lo que tendrá algunos problemas con Windows 2000 y XP.

7.5.2. VMWare

VMware es una solución propietaria que nos permitirá ejecutar una máquina virtual en nuestro sistema, con lo que podremos correr en ella cualquier sistema que deseemos, haciendo así más fácil la ejecución de utilidades Windows/Macintosh. Se puede bajar una versión trial en <http://www.vmware.com> así como comprar on-line una licencia.

7.5.3. WineX

Nos encontramos ante una versión “hermana” de WINE, pero con ciertos matices. El más importante es que se trata de software comercial, esto es, de pago, pero mucho más complejo y avanzado que WINE. Esta aplicación creada por Transgaming Technologies viene con las últimas DirectX implementadas en su código y optimizada para ejecutar juegos creados para plataformas Microsoft.

Se puede obtener información y comprar WineX en <http://www.transgaming.com>.

7.5.4. Emulando MacOS

Aunque no hay demasiados proyectos para la emulación de MacOS, anteriores al X, existen algunos que podrán facilitar la vida de los usuarios de Macintosh:

- Ejecutor 2.0 : emulador genérico de Macintosh. <http://www.ardi.com>
- Basilisk II: soporta aplicaciones hasta MacOS 8.1.
<http://www.uni-mainz.de/~bauec002/B2Main.html>
- Mace : gran emulador aunque aun esta en fase de desarrollo. <http://www.macehq.cx/>
- vMac: soporta hasta sistema 7.5.5. <http://www.vmac.org>

CAPÍTULO 8

Programación básica en sistemas Linux

David Olivieri*

8.1. Introducción

Para el programador los sistemas Linux ofrecen una plataforma y una amplia colección de herramientas consistentes que ayudan a construir y mantener proyectos grandes y complejos. Al mismo tiempo, el modelo Open Source ofrece una ventaja enorme, no sólo de coste, sino de disponibilidad y estandarización para la reutilización de software. El programador tiene acceso directamente con un sistema operativo de manera transparente. Sin embargo, esta flexibilidad requiere un gran conocimiento de la estructura de los sistemas Linux para que se pueda obtener el máximo rendimiento.

En este capítulo se explican algunos aspectos importantes para ser capaz de programar con eficacia en Linux. Debido a que es necesario manejar el sistema como administrador, se realizará programación directamente en el “shell”, con programación en bash y awk.

8.2. Programación para administración de sistemas

Imaginemos un paquete de software típico que se puede desarrollar en Linux. Quizás el software esté basado en un modelo cliente/servidor con demonios de servidor que escuchan, con sockets para conexiones de clientes y, al mismo tiempo, tenga un alto nivel de seguridad con SSL, o conecte con una base de datos y tenga que acceder a servicios de red, disco o dispositivos especiales. Debido a esta situación, que es habitual en la programación en Linux, es fundamental para el programador un conocimiento del propio sistema y el lenguaje que se proporciona, por lo cual éste tiene que entender el shell de Linux.

La interacción con el sistema operativo se hace a través del shell de Linux. El shell es el interfaz al mundo exterior de lo que está haciendo el núcleo en el fondo y es la forma mediante la cual se puede configurar, comprobar y modificar su funcionamiento interno. En esta sección se detallan algunos de los aspectos más importantes de programación en el shell.

8.2.1. Programación con Scripts

¿Qué es un script? La mayoría de servicios de configuración (si no todos) de un sistema Linux son iniciados, mantenidos, y apagados con scripts del shell. El shell proporciona un lenguaje completo para llevar a cabo todas las tareas necesarias, siendo los scripts programas en el lenguaje del shell. Es importante destacar que existen diferentes tipos de shells de las distintas versiones de UNIX, debido a muchos esfuerzos independientes en la historia para crearlos. Los distintos shells más conocidos son: bash (sh), el csh (c-shell) y ksh (korn) shell. A partir de aquí,

* Escola Superior de Enxeñería Informática de Ourense. (Universidade de Vigo)

sólo nos remitiremos al bash shell, aunque la mayoría de las estructuras funciona también en los otros.

Si tuviéramos que desarrollar un shell para Linux, ¿cuáles serían las características que debería tener para que pudiéramos tener un sistema completamente funcional?. Recordando que el shell es nuestro portal al sistema operativo, algunas de las características que podría necesitar serían: (a) una forma de guardar en un fichero grupos de comandos a ejecutar, (b) una forma de controlar la entrada/salida, (c) la posibilidad de definir variables y utilizarlas, (d) procesamiento de cadenas, (e) estructuras lógicas de control, (f) aritmética básica y (g) procedimientos o funciones. Cada una de estas características forma parte del shell de bash. A continuación vamos a estudiar estos aspectos.

8.2.1.1. Variables y variables del entorno

Se pueden asignar valores a variables dentro del shell sin necesidad de declarar su tipo o su existencia previamente. La asignación de valor es sencillamente `var=<valor>`. Fijémonos en que no hay espacio entre el nombre de la variable, el operador “=” o el valor que se puede asumir. También, si el valor es una cadena, únicamente si la cadena contiene un espacio en blanco, es necesario que se encierre con comillas. Para acceder al valor de la variable, se emplea el prefijo “\$”, de forma que si la asignación de nuestra variable es `xbar=5`, entonces se accede al valor de `xbar` utilizando `$xbar`.

Las variables del entorno son globales y están pensadas para guardar información, de manera global, para una sesión de Linux. Algunas de las variables que vamos utilizar más adelante son:

- `HOSTNAME`
- `LOGNAME`
- `SHELL`
- `USER`
- `PATH` lista de directorios en los que el shell puede buscar comandos
- `HOME` directorio “home” de usuario
- `CDPATH` lista de directorios usados por el comando “cd”
- `MAIL` lista de ficheros en los que el shell mira para ver nuevos mensajes de correo
- `PS1` cadena primaria de prompt
- `PS2` cadena secundaria de prompt

Como con las variables definidas por el usuario, al valor de un variable de entorno se accede añadiendo el prefijo \$ al nombre de la variable. Para listar todas las variables de entorno definidas en el sistema en cualquier momento, basta con ejecutar el comando “env”.

8.2.1.2. Operaciones Aritméticas y Lógicas

Para tener un lenguaje útil, tenemos que tener variables. Igualmente importante es ser capaz de operar con estas variables con operadores aritméticos o lógicos. Los operadores aritméticos disponibles en el shell de bash son: “+”, “-”, “*”, y “/”, para la suma, resta, multiplicación y división respectivamente. Las operaciones lógicas son: `-lt (<)`, `-gt (>)`, `-le (<=)`, `-ge (>=)`, `-eq (==)`, `-ne (!=)`.

Existen dos métodos para realizar operaciones aritméticas. Podemos utilizar la palabra “let”, o podemos utilizar “expr”. En el caso de “let”, por ejemplo, si queremos añadir dos números, la sentencia es

let a=1+2
usando “expr”

- suma de números: a=`expr 5 + 3`
- suma con variables: a=`expr \$a + 1`
- módulo: a=`expr 5 % 3`

Un ejemplo de utilización de operadores de comparación sería:

```
#!/bin/bash

a=animal
echo a is $a
if [ `expr $a = animal` ]
then
    echo "a is not "
fi
```

8.2.1.3. Scripts: el elemento básico en la programación de shell

Escribir un script es útil cuando existe una tarea del sistema para implementar repetitivamente. Muchos scripts existen en el sistema para poner en marcha servicios del mismo, por ejemplo la configuración de la red o una base de datos. Por lo tanto, un script consiste en un agrupamiento de comandos, con capacidad aritmética, bucles, y sentencias de control para implementar comandos del sistema. No obstante, la ejecución de un script es lenta comparada con un binario producido mediante la compilación de un programa en C, por ejemplo. Es decir, mientras un script puede ser muy fácil de implementar, no se debe usar un script que requiere velocidad, que accede a hardware del sistema, necesita utilizar E/S de sockets, o que requiere librerías.

Consideremos un script para hacer un “echo” de “date” y que liste los ficheros de hoy:

```
#!/bin/bash

echo date
find ~ -daystart -type f -exec ls -tl { } ';
```

Si hemos guardado en el fichero de texto con el nombre fhoy, entonces podemos ejecutar este script de dos maneras en la línea de comandos:

```
$ source fhoy
```

o modificarlo con el comando “chmod” para que sea ejecutable y a continuación ejecutarlo.

```
$ chmod +x fhoy
$ ./fhoy
```

Fijémonos en que la primera línea del script `#!/bin/bash` es una indicación al shell de qué programa vamos a utilizar para interpretar los comandos que siguen en el fichero. En este caso, se utiliza el propio shell `bash` para interpretar los comandos. Con el comando “source” (como se indica en la primera manera de ejecutar el script) no hace falta el `#!/bin/bash`, ya que se ejecuta con un comando del shell directamente, aceptando las líneas en el fichero como flujo. En el segundo ejemplo de ejecución del script, hemos tenido que cambiar los privilegios para que se trate al fichero como binario. En general no podemos ejecutar el nombre directamente si el directorio de trabajo no está en el `PATH` del usuario.

Una breve descripción del script: el símbolo “~” indica que el comando “find” va dentro del directorio y subdirectorios de “home” (indicado por `HOME`). Se utiliza el parámetro “daystart” para indicar que sólo imprime los ficheros regulares (indicado por “type”) que se han modificado hoy, mientras el parámetro “exec” es para procesar el resultado con el comando “ls”.

- **Estructuras de control con if-then-else-fi estructura**

Una de las estructuras de control en lenguajes de programación es la sentencia “if”. La forma general para sentencias de “if” en `bash` es:

```
if test-comandos then
    comandos;
[elif más-comandos; then
    comandos;]
[else alternativa-condición;]
fi
```

En la definición, los test-comandos se pueden especificar de dos maneras: utilizando la palabra `test`, o utilizando los corchetes `[]`.

Son dos formas completamente equivalentes y a continuación, exploramos distintas maneras para formar sentencias `if` con las dos formas.

Consideramos un ejemplo del uso de la estructura “if”. En este ejemplo comparamos dos variables “a” y “b”, y se imprime si “b” es mayor, menor o igual que el valor de la variable “a”.

```
#!/bin/sh
a=10
b=5
if test $a -gt $b ; then
    echo 'a es mayor que b'
elif test $a -lt $b ; then
    echo 'a es menor que b'
else
    echo 'a es igual a b'
fi
```

Notemos cómo a los valores de las variables se accede con `$a` y `$b`. También, nos

debemos fijar en otros detalles de este programa sencillo: la construcción del “echo” y cómo las variables se escriben entre comillas, así como para la asignación de las variables, no aparece ningún espacio entre el nombre, el operador y el valor.

Para scripts de administración de sistemas, siempre estamos interesados en cierto tipo de comprobaciones básicas: determinar si existen argumentos en la línea de comandos para un script, determinar si existe un fichero o determinar si un fichero es ejecutable. En los siguientes ejemplos, se comparan dos maneras de escribir la sentencia “if” para determinar si hay parámetros en la línea de comandos cuando se ejecuta el script.

```
#!/bin/bash

# script utilizando test
if test -z $1
then
    echo "No argumentos"
else
    echo "Primer argumento es $1."
fi
```

```
#!/bin/bash

# script utilizando [ ]
if [ -z $1 ]; then
    echo "No argumentos"
else
    echo "Primer argumento es $1."
fi
```

Si el fichero se llama “prueba”, entonces invocando el script en la línea de comandos con

\$/prueba arg1

producirá como resultado que el primer argumento es “arg1”. Fijémonos en otro detalle de bash: en el ejemplo de arriba se puede ver que en el “if [-z \$1]” escribimos espacios entre el “[“, el “-z” y el “]”; no es así para que sea más fácil de leer, es así porque si no estuvieran los espacios, bash daría un error.

Existen muchas opciones para pruebas con test (o con []). Algunas de ellas son: “-f” verdadero si el fichero existe, “-d” verdadero si existe y es un directorio, “-x” verdadero si el fichero existe y es un ejecutable, etc.

- **Estructura de case**

Bash tiene sentencias denominadas de “case”. Esta sentencia existe también en otros lenguajes de programación, como por ejemplo C. Su estructura es:

```
case palabra in
    [patron [| patron]...) comandos ;;]
esac
```

Ejecutan selectivamente los comandos basados en palabras que son iguales a “patrón”. La barra “|” se utiliza para separar patrones múltiples.

```
echo -n "Introduce el nombre de trabajador: "  
read trabajador  
echo -n "El $trabajador gana "  
case $trabajador in  
  abogado | medico) echo -n 'trenta-';;  
  profesor ) echo -n "dos-";;  
  *) echo -n "0";;  
esac  
echo "Mpts/año "
```

- **Estructuras de control con bucles**

La primera estructura de control con bucles que estudiamos es “for”. Su sintaxis es la siguiente:

```
for vars [in palabras ...]; do  
    comandos;  
done
```

donde “vars” es la variable que recorre todo lo que está dentro de “palabras”, que es una lista o secuencia. Los comandos están dentro de una estructura de “do - done”.

Un ejemplo estándar del bucle “for” es:

```
#!/bin/bash  
for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
do  
    echo $i  
done
```

```
#!/bin/bash  
  
for i in $(seq 15)  
do  
    echo $i  
done
```

Aunque el ejemplo de la izquierda funciona, es bastante limitado, especialmente si tenemos que hacer el bucle muchas veces. El ejemplo de la derecha hace lo mismo, pero con una implementación bastante mejor. No es necesario que la lista sean números, puede ser una serie de cadenas.

- **Estructura de control de until**

La estructura de control “until” se construye de la siguiente forma:

```
until test-comandos do  
    comandos;  
done
```

En este bucle se ejecutan comandos hasta que “test-comandos” es verdad. Este bucle tiene un estatus de salida distinto de cero. Una estructura de control muy parecida es la estructura “while”. Esta estructura tiene la siguiente forma:

```
while test-comandos; do
    comandos;
done
```

Consideremos, por ejemplo, que queremos hacer operaciones dentro de un bucle hasta que un contador, que disminuimos sucesivamente al completar el bucle, es igual a un número determinado. En los siguientes ejemplos implementamos el mismo algoritmo con dos tipos de bucles.

```
#!/bin/bash

contador=20
until [ $contador -lt 10 ]; do
    echo 'contador= $contador'
    let contador-=1
done
```

```
#!/bin/bash

contador=0
while [ $contador -lt 10 ]; do
    echo 'contador= $contador'
    let contador+=1
done
```

Un ejemplo más interesante y que utiliza las estructuras que estamos considerando hasta ahora, es el siguiente: consiste en guardar ficheros importantes en otros con nombres enumerados. Fijémonos en la manera en que se crea un nuevo nombre de fichero directamente con la asignación de variables.

```
#!/bin/sh
if [ `expr $1 = " "` ]; then
    echo 'Utilizacion: backup.sh <nombre-fichero>'
    exit
fi
for i in 0 1 2 3 4 5 6 7 8 9 ; do
    Nuevo_Fichero=$1.BAK-$i
    if test -e $Nuevo_Fichero ; then
        echo 'backup.sh : Aviso $Nuevo_Fichero'
        echo "“existe”"
    else
        cp $1 $Nuevo_Fichero
    fi
done
```

En el siguiente ejemplo se borra el contenido de un fichero haciendo una redirección de /dev/null al fichero. También se utiliza “touch” para crear el fichero vacío, “tail” para copiar las últimas líneas del fichero y enviarlas a un fichero diferente. Más adelante explicaremos la redirección en detalle.

```
#!/bin/bash
filename=sys.log
if [ ! -f $filename ]
then
    touch $filename; echo "Crear fichero"
else
    cat /dev/null > $filename; echo "Borrando contenidos del fichero."
fi
tail /var/log/messages > $filename
echo "$filename contiene las ultimas lineas de syslog."
exit 0

a=$(date +%T-%d_%m_%Y)
cp -i $1 ~/Backup/$subdir$1.$a
```

- **Funciones de Shell**

```
[`function`] NAME () { COMMAND-LIST; }
```

```
#!/bin/bash
function quit
{
    exit
}

hola ()
{
    echo Hola!
}
hola
quit
```

```
#!/bin/bash
function factorial ()
{
    N=$1
    A=$2
    while test $N -gt 0 ; do
        A = 'expr $A '+' $N'
        N = 'expr $N - 1'
    done
    echo $A
}
factorial 5 1
```

Estos dos ejemplos de funciones muestran algunas propiedades interesantes. El ejemplo de la izquierda muestra que las funciones no necesitan parámetros y que la palabra “function” es opcional. En el ejemplo de la derecha, la función factorial requiere unos parámetros, pero estos parámetros no se colocan entre paréntesis como en un programa de C, sino que se escriben indicando e invocando la función y listándolos sin comas.

- **Control de entrada/salida**

En nuestro ejemplo anterior utilizamos el comando “echo”, que imprime su resultado en pantalla. En realidad debemos hablar de “stdin”, “stdout”, y “stderr”. Linux/UNIX tiene mecanismos de redirección para convertir definiciones estándar. Por lo tanto, en vez de escribir nuestro resultado de “echo date” en pantalla, como es por defecto, podemos ejecutar:

```
echo date > ~/f-info.log
```

que es una forma de redireccionar la salida de “echo” a un fichero. Para entender la redirección en Linux, es preciso explicar tres descriptores de entrada/salida para ficheros: “stdin”, “stdout” y “stderr”, los cuales tienen asignada la siguiente numeración: 0,1,2, respectivamente. El “stdin” de un programa son los datos que necesito como entrada, mientras el “stdout” son los datos de salida. El “stderr” son los errores que produce el programa. Por defecto, Linux escribe el “stdout” y “stderr” en la pantalla del terminal. Cuando hacemos una redirección, estamos cambiando la dirección de “stdout”. Podemos cambiar también el “stderr” por defecto, para que se escriba en un fichero.

Las distintas maneras de direccionar “stdout” y “stderr” son:

- Escribir “stdout” a un fichero (ej. ./nombre-script > fichero.txt).
- Añadir el “stdout” al contenido de un fichero (ej. ./nombre-script >> fichero.txt).
- Escribir “stderr” a un fichero (ej. grep da * 2> grep-errors.txt).
- Escribir “stdout” al mismo descriptor de fichero que “stderr” (ej. grep da * 1>&2).
- Escribir “stderr” al mismo descriptor de fichero que “stdout”.

Consideremos el siguiente ejemplo, el cual redirecciona la salida del comando “stderr” al mismo lugar que esté direccionado “stdout”, que en este caso puede ser la pantalla.

```
grep * 2>&1  
rm -f $(find / -name core) &> /dev/null
```

El comando “rm” en el ejemplo de arriba busca ficheros de nombre “core” en todos los directorios de “/”, enviando cualquier mensaje que produzca a “/dev/null”.

Las “tuberías” son otra forma de enviar el “stdout” de un conjunto de comandos al “stdin” de otro conjunto de comandos. Las tuberías se representan con una barra. En los siguientes ejemplos se envía el stdout del comando “ls -l” al stdin de los comandos “sed” o “grep”.

```
ls -l | sed -e "s/[aeio]/u/g"  
ls -l | grep "\.txt$"
```

Se pueden utilizar múltiples tuberías como en el siguiente ejemplo: primero se lista el contenido de todos los ficheros que tienen extensión .lst, se pasa el listado al programa “sort” (que ordena según el primer campo), y después se envía el resultado a “uniq” (que elimina duplicados).

```
cat *.lst | sort | uniq
```

Podemos utilizar tuberías para simplificar, pasando de tener que escribir una serie de pasos a sólo uno. Consideremos el siguiente ejemplo de descompresión del núcleo con “bunzip2” y posterior desempaquetado con “tar”:

```
bunzip2 linux-2.2.15.tar.bz2 | tar xvf -
```

Existen caracteres especiales para acceder a valores del shell. Son especialmente importantes para programación con scripts. El primero de la lista, ya lo vimos anteriormente, es la manera de acceder a los parámetros en la línea de comandos cuando ejecutamos el script.

- \$0, \$1, \$2, etc. parámetros posicionales (dados en línea de comandos al script o a funciones)
- \$# número de argumentos en línea de comandos
- \$\$ identificador de proceso del script (utilizado para crear nombres de ficheros temp)
- \$? estatus de salida del comando, función, o el script.
- \$* todos los parámetros posicionales.
- @\$ lo mismo que \$*, pero cada parámetro es un cadena con comillas.
- \$- Flags pasadas al script.
- \$! PID de la último tarea ejecutada en background.

Consideremos el siguiente ejemplo para utilizar el valor que devuelve un comando.

```
#!/bin/bash
cd /dada &> /dev/null
echo rv: $?
cd $(pwd) &> /dev/null
echo rv: $?
```

8.2.2. Programación con awk

En la sección anterior, aprendimos que los scripts de bash proporcionan un lenguaje potente para programar tareas del sistema y recuperar información de comandos. El problema es que bash no está optimizado para tratar texto, aunque es perfectamente posible, requiere mucha práctica. Cuando estamos buscando información de sistemas, lo que siempre necesitamos hacer es procesar un flujo de texto o buscar y extraer palabras de cadenas. Existen programas o entornos de programación, que son interpretados como el lenguaje de bash, que están optimizados para procesar texto.

Awk es un lenguaje excelente para procesar texto. Es sencillo de programar, tiene todas las estructuras de bash, y es capaz de buscar y reconocer patrones dentro de un flujo de texto. Aunque muchos elementos del lenguaje de awk son parecidos a bash, su manera de funcionar es completamente distinta. Vamos a resumir brevemente cuál es el modelo básico de programación en awk.

La estructura de un programa awk consiste en el siguiente concepto: escanear líneas de texto, una tras otra y buscar patrones; si se encuentra el patrón, entonces actúa en él. Se puede ejecutar awk directamente en la línea de comandos en el terminal, o se puede construir un script. La forma básica del script de awk es la siguiente:

```
patrón { acción }
```

Cada línea del script consiste de un patrón y si se encuentra el patrón en el texto, se implementa la acción. Las acciones pueden consistir de múltiples líneas para el mismo patrón. Consideremos el siguiente ejemplo de un fichero denominado “miawk”

```
{ print $0 }
```

En este ejemplo, no existe el patrón, por lo cual se ejecuta la acción para cada línea de texto que awk va a procesar. En este ejemplo \$0 es toda la línea. Para invocar awk en este ejemplo:

```
awk -f miawk1 /etc/group
```

Un ejemplo de utilización de awk para buscar un patrón es:

```
/david/ { print $0 }
```

8.2.3. Programas awk con ficheros

Los siguientes scripts combinan lo que aprendimos de scripts de bash y unos conceptos avanzados de la utilización de awk.

Escribamos un script en bash que utilice el comando “find” para determinar los 10 usuarios que ocupan más espacio de disco duro en su sistema. Con estos usuarios, ubicados por ejemplo bajo /home, determinar los ficheros que son más grandes. El comando “find” puede ser conveniente (con la opción “-exec” junto con el programa awk). Las aplicaciones “sort” y “head” también pueden ser útiles.

```
#!/bin/sh
find / -type f -exec /bin/ls -ls {} ';' | awk '
{using[$4] += $1 }
END {for (name in using)
    { print name } }' | sort -nr +0 | sed 10q > tmp.txt
for i in $(exec cat tmp.txt)
do
    echo $i
    find /home/$i -type f -exec /bin/ls -ls {} ';' | awk '
    {print $4, $6, $10}' | sort -nr +1 | sed 4q
done
```

Escribamos un script de bash para determinar si un usuario está conectado al sistema. Si está en el sistema, queremos determinar el proceso que está ejecutando que utiliza más CPU.

```
#!/bin/sh
if
  who | grep $1 > /dev/null
then {
  echo "$1 esta logged"
  ps aux | gawk -v user=$1 '$1 == user {print
$0}' | sort -nr +2 | sed 1q
}
else
  echo "no esta"
fi
```

8.3. Conclusiones

En este capítulo, hemos visto algunos aspectos de programación con bash y también con awk. Para desarrollar programas en Linux es necesario un conocimiento de conceptos básicos de sus comandos y la forma de manejar el shell. Este capítulo puede constituir una base para continuar un estudio más profundo de programación con bash.

CAPÍTULO 9

Seguridad informática básica en sistemas GNU/Linux

Antonio José Fernandes Vázquez*

9.1. Introducción

La seguridad es un tema a tener en consideración en todos los sistemas operativos, ya que no queremos que cualquier persona no autorizada tenga acceso a nuestros datos confidenciales que puedan residir en una determinada máquina de nuestra posesión.

En este capítulo intentaremos abordar los pasos básicos para asegurar esa información y mantenernos a salvo, en lo posible, de los ataques que podamos sufrir de personas ajenas al sistema.

Podemos decir que la seguridad incluye dos fases. La primera fase sería la puesta a punto del sistema antes de entrar en producción, es decir, antes de conectar el equipo a cualquier tipo de red, mientras que la segunda es un proceso que se debe realizar continuamente, según utilicemos nuestro sistema.

9.2. Passwords

El uso de passwords sencillos es la forma más habitual en GNU/Linux de debilitar un sistema ante los ataques a la integridad del mismo. Hemos de tener en consideración algunos consejos básicos a la hora de elegir los passwords de nuestras cuentas de usuario.

- No utilizar palabras comunes, por ejemplo, cualquiera que aparezca en un diccionario. Independientemente del idioma que sea.
- Utilizar siempre que sea posible combinaciones de números, símbolos, mayúsculas y minúsculas, de forma aleatoria.
- No dejar escrito el password en ningún tipo de “post-it” o papel.
- No emplear para la creación del password ningún tipo de información personal, ya que aunque parezca increíble, prácticamente cualquier persona puede obtener información sobre nosotros de una manera relativamente sencilla.

Nos resultará útil saber cómo funciona el sistema de validación de passwords para entender por qué debemos seguir dentro de lo posible estos consejos.

El sistema capta el login y el password cuando nosotros los introducimos, cifra el password con el algoritmo de cifrado que emplee, compara el password ya cifrado con el existente en el sistema, y si éste coincide valida el acceso. Si no coincide, lo niega.

Los usuarios y passwords se almacenan habitualmente en el archivo de sistema localizado en: `/etc/passwd`.

* HispaLinux

El único problema que representa este sistema de autenticación es que este archivo en concreto ha de ser legible por todos los usuarios para permitir el funcionamiento correcto del sistema, por lo que si cualquier usuario se introduce en nuestro sistema podrá hacerse con una copia de este archivo maestro de usuarios y passwords, y proceder luego a un ataque contra el mismo para obtener toda la lista de usuarios con sus respectivos passwords.

Los ataques a estos archivos se suelen realizar a partir de uno o varios diccionarios de palabras en distintos idiomas, cifrando y comparando cada una de las palabras a los passwords que aparecen en el `/etc/passwd`.

El software que realiza este tipo de ataques también se encarga de añadir números y símbolos, así mismo, de cambiar minúsculas a mayúsculas, con el fin de probar todas las combinaciones posibles por cada palabra.

Un ataque mediante “fuerza bruta”, es decir, ir probando secuencialmente todas las combinaciones posibles entre letras (minúsculas/mayúsculas), números y símbolos, siempre ha sido considerado prácticamente imposible, siempre y cuando las contraseñas usadas sean largas.

El programa más empleado para este tipo de ataques, y el más conocido, se llama “John The Ripper”, y se puede encontrar disponible para su descarga desde (<http://www.openwall.com/john/>).

Afortunadamente existe una solución a este problema que es ampliamente utilizada por defecto en la mayoría de distribuciones de GNU/Linux en la actualidad, el sistema de ocultación de contraseñas, conocido popularmente como “shadow”. Este sistema crea un archivo (`/etc/shadow`) con permisos de lectura/escritura para root y de escritura para los usuarios pertenecientes al grupo *shadow*. En este archivo estarán los passwords en texto claro, sin embargo sólo el root podrá verlos, evitando así que cualquier usuario tenga acceso a las misma. En el archivo `/etc/passwd`, el campo en dónde aparecían los passwords cifrados aparecerá sustituido por una “x”.

9.3. Software del sistema

Instalar cualquier distribución de GNU/Linux por defecto no es nada recomendable desde el punto de vista de seguridad, ya que muchos de los paquetes de software que se instalan no los vamos a necesitar para absolutamente nada y el descubrimiento de un fallo de seguridad en cualquier paquete de software de nuestro sistema nos puede poner en un gran aprieto. Por todo esto es recomendable hacer una instalación personalizada del sistema, que contenga única y exclusivamente lo que vamos a necesitar para hacer el trabajo al que vamos a destinarlo.

Así pues, de nada nos sirve, por ejemplo, un software servidor de correo en una máquina cuya única función será la de servir páginas web. También deberíamos vigilar la configuración del superdemonio `inetd`, o del mas moderno `xinetd`; estos demonios se encargan de lanzar por defecto algunos servicios de red a la memoria del sistema (servicios como `telnet`, `talk`, `chargen`,...). Muchos de estos servicios no nos servirán absolutamente para nada, por lo que podremos prescindir de ellos sin mayor problema. En teoría estos demonios pueden gestionar casi cualquier software servidor que emplee la red, pero en la práctica es más recomendable que cada uno de los paquetes de software se lancen a memoria independientemente. Cualquier ataque o fallo de `inetd` o `xinetd` puede dejar fuera de combate a todos los servicios que dependan de ellos.

Hemos de tener en cuenta otro detalle del software instalado en nuestro sistema, los programas que cuenten con “SUID”. Un programa SUID es aquel que se ejecuta con los privilegios de su propietario, no con los privilegios del usuario que lo invoca. Un fallo en estos programas podría permitir la escritura/lectura en archivos en teoría restringidos, por lo que son programas con los que hemos de tener un cuidado especial.

Por último, en la puesta a punto del sistema, antes de entrar en producción, es recomendable que conozcamos (y usemos dentro de lo posible), el `CHROOT`. `CHROOT` es una llamada al sistema en UNIX que permite configurar como “raíz” del sistema de ficheros para un

procesos y todos sus hijos. Es decir, que nos permite configurar el sistema de tal forma que nos permita ejecutar procesos dentro de un directorio concreto. Para ellos, este directorio será “/” y cualquier directorio o archivo que esté fuera de ahí quedará inaccesible. Algo así como si encerráramos en jaulas a nuestros programas.

Quizás el empleo de CHROOT nos resulte a priori engorroso, pero es una forma adecuada de dotar a nuestro sistema de una forma más de evitar intrusiones no deseadas.

9.4. Ataques al software del sistema

Una vez hecha la puesta a punto del sistema ya podemos conectarlo a una red. En este capítulo constantemente se habla de conectarlo a una red, y es que la única forma de que un sistema esté seguro es no conectarlo a ninguna red, ya que una vez conectado puede ser objetivo de cualquier ataque. Estos ataques pueden surgir por varios motivos, desde que nuestro sistema le parezca interesante a cualquiera por su contenido, hasta simplemente por aburrimiento del atacante, por lo cual, mejor que estemos preparados aunque no creamos que podamos ser víctimas.

Hemos de comprender el funcionamiento y la importancia de los ataques que pueden realizarnos para intentar conseguir el acceso a nuestro sistema. Primero nos centraremos en los ataques al software que puedan permitir el acceso al sistema.

A veces por descuido, o bien por no emplear una programación segura, en ocasiones los programas que usamos tienen fallos de seguridad. Ésto no tendría mayor relevancia si todas las personas fuéramos éticamente correctas y simplemente nos dedicáramos a dar aviso de la existencia del fallo en concreto, y no hiciéramos nada más. Sin embargo ésto no es así, y muchas personas explotarán estos fallos de seguridad en los sistemas para poder hacerse con el control del mismo.

Hay muchos tipos de fallos de seguridad, “stack overflows”, “format strings”, “temp races”,... sin embargo aquí haremos especial hincapié en los stack overflows, para que el lector se haga una idea concreta de por qué el sistema falla y cómo lo emplean los potenciales atacantes.

Los “stack overflow” (desbordamiento de pila) son los mayores en número, así mismo, los que más crisis han podido producir en sistemas GNU/Linux. Para entender como funciona, debemos aclarar qué son los “stack buffers”, debemos entender primero cómo es el proceso de organización de la memoria. Los procesos se dividen en tres regiones: texto, datos y stack (pila). La pila se encarga de la abstracción a la hora de almacenar las variables locales de las funciones que manejan los programas y de saber a dónde retornan; es algo empleado en los lenguajes de programación de alto nivel. Cuando se escribe mas allá del último local de la pila, lo sobrepasas, y se puede sobrescribir la dirección de retorno. Podemos indicarle que ejecute de forma arbitraria una dirección de memoria, el problema aparece cuando en esa dirección de memoria hay un código malicioso. Así pues, el atacante puede aprovechar un “stack overflow” para ejecutar /bin/bash y tener una consola con acceso a nuestro sistema. Si aún por encima ataca a un programa que está siendo ejecutado con privilegios de root, el atacante tendrá un acceso al sistema como superusuario. Si se necesita mas información sobre cómo funcionan los “Stack Overflows” se puede consultar la siguiente web: <http://community.core-sdi.com/~juliano/>.

Este tipo de fallo, al igual que el “format string”, tiene la peculiaridad de que se pueden emplear de forma local (con acceso al sistema) o de forma remota (desde cualquier puesto de una red o Internet).

El lector se planteará como va a saber él que existen estos fallos en los programas que él usa, pues bien, para eso la comunidad ha puesto a nuestra disposición listas de correo y páginas en Internet en donde se dan avisos de fallos de seguridad descubiertos en programas. Simplemente deberemos estar atentos a lo que ocurre a nuestro alrededor y actualizar los paquetes de software precisos cuando sea necesario. Algunas de estas listas son:

- SANS (System Administration Networking and Security) [<http://www.sans.org>] es

una organización de educación y difusión de noticias de seguridad. Produce una serie de resúmenes, libros y papeles físicos sobre estos temas.

- SecurityFocus [<http://www.securityfocus.com>] es un portal versado en seguridad informática recientemente adquirido por Symantec. En él se puede obtener información sobre las últimas vulnerabilidades descubiertas por la comunidad y algún que otro artículo interesante. Así mismo, desde él se puede suscribir a la lista de correo “bugtraq” en donde todos los días aparecen avisos sobre vulnerabilidades y se discute sobre ello.
- PacketStorm [<http://www.packetstormsecurity.com>] es un portal independiente en donde se comentan las últimas utilidades y versiones sobre programas de seguridad informática y también se publican pequeños códigos para demostrar que las vulnerabilidades realmente existen. En los últimos años este portal ha ido a menos, sin embargo de vez en cuando publican algo realmente interesante.
- CERT [<http://www.cert.org>] es un grupo veterano de seguridad en Internet fundado a finales de los años 90 por la DARPA (Defense Advanced Research Projects Agency). Aunque en un principio su función era la de centro de respuesta temprana a incidencias de seguridad en Internet, hoy en día gracias al “know-how” adquirido también se dedican a la investigación de las necesidades, en cuanto a seguridad, que precisará Internet con el paso del tiempo.

Aparte de estas listas en donde siempre dispondremos de una información útil a la hora de asegurar nuestro sistema, también tenemos que destacar un programa basado en software libre ideado para darnos información de los problemas de seguridad que tienen nuestras máquinas. Este programa es Nessus (<http://www.nessus.org>). Nessus se define como “escáner” de seguridad remota y básicamente audita cualquier máquina remota en busca de fallos de seguridad con el fin de indicar al administrador qué debería verificar. El equipo de programación de este software está constantemente actualizando la base de datos sobre vulnerabilidades para que los resultados obtenidos tengan la mayor fiabilidad posible.

Existe más software para auditar seguridad informática, y es de fácil búsqueda por Internet, sin embargo suelen ser paquetes comerciales bastante caros, sólo asequibles para grandes compañías y consultoras de seguridad informática.

Para acabar con este capítulo sobre la seguridad informática básica en el software de GNU/Linux, haremos un pequeño comentario sobre utilidades que se pueden añadir al kernel de Linux para aumentar considerablemente su seguridad.

Existen determinados parches no oficiales para el núcleo Linux que dan prestaciones especiales contra ataques como los de “stack overflow”. El empleo de estos parches está recomendado, siempre y cuando sus funcionalidades no interfieran en la estabilidad del software que se emplea en el sistema.

- OpenWall (<http://www.openwall.com/Linux/>): este parche para el kernel nos protege de ataque a la pila (“stack overflows”). Es seguro hasta cierto punto, se han descubierto formas de llegar a pasarlo.
- PaX (<http://pageexec.virtualave.net/>): este otro parche de kernel es categóricamente mejor que el OpenWall, sin embargo, su empleo ralentiza en cierta medida el funcionamiento del sistema.
- Grsecurity (<http://www.grsecurity.net/>): este paquete incluye varios parches, entre ellos el PaX, para dotar de una gran seguridad a los sistemas GNU/Linux.

En las respectivas webs se puede encontrar más información sobre estos parches, su

forma de instalación y funcionalidades. No se considera la profundización en estos temas algo adecuado para un manual de GNU/Linux básico.

Las denegaciones de servicio (DoS, Denial of Service), son otro tipo de ataque que pueden sufrir todos los sistemas, incluido GNU/Linux. Estos ataques no permiten el acceso a nuestro sistema por parte de los atacantes, pero sí que pueden “tumbar” alguno de los servicios que ofrecemos al exterior. Estos ataques suelen estar caracterizados por el envío masivo de paquetes de información “basura” a nuestro sistema. El más característico es el llamado “SYNFlood”, que consiste en el envío constante de paquetes SYN (inicio de conexión) a un determinado servicio. Este servicio envía la respuesta y espera a que le digan algo para establecer la conexión. Como nunca se llega a establecer dicha conexión, el sistema queda esperando, por lo que se consigue que las tablas de conexiones del kernel se llenen y no acepten más, quedando inutilizado. Sin embargo, para este problema ya se ha inventado la solución. El sistema GNU/Linux tiene una funcionalidad en el kernel que contesta a base de “SYN_cookies” a estos ataques y evita que la tabla de conexiones se colapse, pero lo que no evita es que se colapse el caudal que poseamos en Internet, por lo que si el tráfico que nos generan los paquete SYN, o cualquier otro tipo de paquete, hacia nuestra máquina es superior al límite del de la línea de conexión de la cual disponemos, nos quedaremos sin conectividad, o en el mejor caso, sin el canal de entrada a nuestro sistema.

Lamentablemente, este tipo de ataques de saturación de línea no se pueden evitar de ninguna forma con GNU/Linux, sería cuestión ya de nuestro proveedor de servicios de Internet.

Otra forma de provocarnos una denegación de servicio es mediante el anteriormente mencionado “stack overflow”. Si tenemos un fallo de seguridad e intentan acceder, pero hacen erróneamente la explotación del fallo, el intruso no entrará, aunque sin embargo pueden llegar a desactivar alguno de nuestros servicios.

Obviamente, si tenemos los parches de kernel que nos protegen de estos ataques, ni nos daremos cuenta, pero si vemos que un servicio se cae a menudo y no sabemos por qué es, deberíamos sospechar de algo así, no quedando más remedio que actualizar el software por uno que no debiera tener ese problema.

Aparte de las listas de seguridad que ya conocemos, tenemos que tener en cuenta los esfuerzos que hace la comunidad Debian y empresas como RED HAT para tener a sus usuarios siempre protegidos ante fallos de seguridad. Para ésto existen utilidades como apt-get, que permiten disponer de las últimas actualizaciones necesarias del software de nuestro sistema, y siempre dentro de lo posible, las más seguras.

9.5. Software de prevención y alerta

Siempre disponemos de la posibilidad de restringir el acceso a los servicios a determinadas direcciones IP (o direcciones de Internet), pudiendo incluso negar el acceso a todos los puertos de nuestro sistema a una concreta que se haya clasificado como un atacante.

Tradicionalmente se empleaba “TCP_Wrappers” (pseudo-cortafuegos), que se dedicaban única y exclusivamente a dejar acceder o no a determinadas IP's a los servicios existentes en el superdemonio inetd. Este tipo de cortafuegos es bastante débil para los sistemas actuales y, como se ha recomendado anteriormente, es mejor tener los servicios activados de forma autónoma y no dependiente de los superdemonios inetd o xinetd, por lo cual, lo más recomendable a la hora de crear un cortafuegos en GNU/Linux es el empleo de “cortafuegos de filtrado”.

Éstos vienen a ser:

- ipfwadm, en sistemas con kernel 2.0.x
- ipchains, en sistemas con kernel 2.2.x
- iptables, en sistemas con kernel 2.4.x

Este tipo de cortafuegos examina todo el tráfico de entrada y salida del sistema y lo coteja con las reglas existentes en su configuración, previamente establecidas.

Por defecto deja entrar y salir todo el tráfico sin distinción, pero con estos paquetes de software cortafuego podremos establecer reglas de filtrado por dirección, puerto, protocolo, destino, origen,..., prácticamente podemos definir cualquier política de acceso que se nos antoje útil para nuestro sistema. Este tipo de cortafuegos incluso nos permite mantener una lista de logs (registros) sobre todo lo que ha pasado a través de él, y lo que más nos importa, a quién ha denegado el acceso, cuándo y a qué puertos de nuestro sistema.

Aparte del control de acceso al sistema también podemos disponer de un software de detección de intrusos, más conocido como IDS (Intrusion Detection System). Este software también examina los paquetes que entran y salen a nuestro sistema y, mediante reglas preestablecidas en él, discierne sobre si una lista de paquetes es o no un posible ataque de intrusión o denegación. En el caso de que lo sea queda constancia en los logs del mismo de la hora, ip de origen, tipo de ataque que se ha intentado y contra qué servicio.

GNU/Linux trae un paquete de software libre de IDS, snort. (<http://www.snort.org/>). Mediante el uso de snort se crearán los ficheros de registro que son todo lo que ha pasado a través de nuestra red y los resultados después de pasar los filtros.

Cabe destacar que los resultados no siempre son reales, muchas veces aparecen falsos positivos. Hay que considerar que este tipo de análisis son muy difíciles ya que se hacen sobre la marcha.

La revisión exhaustiva de estos logs es normalmente tediosa, y pocos administradores lo hacen regularmente. Sin embargo, existe software que soluciona este problema. Una solución libre es, por ejemplo, ACID, que se puede descargar de la siguiente página web <http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html>; ACID nos da una información algo más visual que los archivos de texto a través de una web en PHP, permitiendo la búsqueda de patrones en los logs (previamente guardados en una base de datos SQL) a través de un sistema de búsqueda sencillo.

Otra solución existente es Demarc PureSecure (solución comercial) <http://www.demarc.com/>. Al igual que ACID emplea una base de datos SQL para guardar los logs. Desde una web en PHP, bastante más elegante e informativa que la solución anterior, nos muestra todas las incidencias de una forma clara.

Acabamos de ver como tanto el cortafuegos como el sistema de detección de intrusos guardan logs de todo lo que hacen. También el propio sistema GNU/Linux guarda constancia de todo lo que pasa en él, y gracias a ellos podemos tener una información constante de lo que pasa en nuestras máquinas y servicios. Así pues, en `/var/log/` encontraremos el archivo `syslog`, que depende del demonio `syslogd`, el cual podemos configurar desde `/etc/syslog.conf`. En él podemos definir qué queremos discernir de los logs y enviarlo a un archivo separado para tener una visión directa de los errores por servicios. Por ejemplo, podemos definir que sólo registre en el log mensajes determinados como pueden ser los de error, críticos, alerta o emergencia y saber si está pasando algo extraño en el sistema.

9.6. Seguridad en las comunicaciones

Por último, una vez que tenemos el sistema totalmente montado, prácticamente fortificado y con sistemas de aviso en caso de incidencias, de nada nos serviría si hacemos accesos no seguros al mismo.

Desde luego, el uso del telnet clásico debería estar totalmente prohibido. ¿Por qué corremos el riesgo de que algún infiltrado en la red local de nuestro sistema “espíe” la red con el software adecuado y nos robe nuestro usuario y password?.

Todas las comunicaciones hacia nuestro sistema han de ir cifradas, por ejemplo, el

acceso remoto al mismo ha de hacerse desde SSH (Secure Shell), un paquete libre disponible para esto puede ser OpenSSH, incluido en cualquier distribución de GNU/Linux actual.

Así mismo deberíamos usar cifrado siempre y cuando nos sea posible, sobre todo en servicios en dónde usuarios y passwords de autenticación viajan en texto plano por la red hasta nuestro sistema.

Ejemplos pueden ser el correo POP3 o IMAP o el envío de claves por web. Para lo primero existen soluciones con Software Libre que permiten el uso de cifrado, mientras para la web es tan fácil como usar el servidor seguro de Apache (el servidor web más extendido en Internet) para los menesteres que necesiten autenticación.

9.7. Recomendaciones para obtener un sistema seguro

En este punto lo más recomendable es descargarse de Internet la “Guía de Seguridad del Administrador de Linux”, lo cual se puede hacer buscando en <http://www.google.com> su nombre (está disponible en español, por supuesto) o descargándola desde las páginas de documentación en español de HispaLinux y el Proyecto LuCas.

Siguiendo paso a paso las recomendaciones de este manual de sencilla aplicación, se obtendrá un sistema Linux a prueba de casi cualquier tipo de intrusión imaginable.

CAPÍTULO 10

Instalación de un servidor web. Instalación de un LAMP.

Santiago Rodríguez Collazo¹

10.1. Introducción

El objetivo de este capítulo es servir como ayuda ante la instalación de un servidor web para una página web estática y para una instalación ya más avanzada de un servidor con soporte para páginas dinámicas utilizando Apache + php + mysql, y obviamente, sobre una plataforma Linux. Esta última combinación se denomina en ocasiones “plataforma LAMP”. Debido al creciente número de distribuciones Linux, y de otros sistemas operativos de código abierto (*BSD), trataremos de evitar referirnos a ninguna en especial, y para ello realizaremos la instalación desde los paquetes de código fuente ofrecidos en las páginas oficiales de cada aplicación.

Por supuesto todas las órdenes contenidas en este capítulo es conveniente ejecutarlas como superusuario (root).

10.2. Apache

10.2.1. ¿Qué es?

El servidor web Apache es un proyecto de desarrollo de un servidor web de código abierto para diversos sistemas operativos, entre los que se encuentran GNU/Linux, *BSD, o Windows. El objetivo del proyecto es proporcionar un servidor seguro, eficiente y ampliable, que proporcione los servicios HTTP de nuestros días, en conjunción con el estándar HTTP. El proyecto Apache desarrolla dos versiones de su servidor web. En este manual utilizaremos la versión 2.0.48 (la última disponible en el momento de escribir estas líneas). También se encuentra disponible la versión 1.3.29, recomendada para servidores con una carga elevada. Uno puede preguntarse por qué usar una versión que, aunque superior en número de orden, no es la recomendada para servidores de producción. Una de las principales diferencias con la 1.* es la reescritura total de la forma de comunicación interna (API). Las versiones 2.* serán algún día las únicas en desarrollo, y en estos momentos la plataforma sobre la que se planean las nuevas posibilidades de Apache.

El servidor web httpd, desarrollado por el proyecto Apache, es tremendamente modular, se le pueden añadir un número enorme de nuevas posibilidades simplemente añadiendo módulos. Algunos de estos módulos son obra de los desarrolladores de Apache, pero también se pueden encontrar otros desarrollados por personas totalmente ajenas al proyecto.

¹ Grupo de Usuarios de Linux de Ourense

10.2.2. Instalación de web estática

A la hora de instalar un servidor web las opciones para su contenido son tan variadas como los distintos tipos de páginas que podemos observar al navegar por Internet.

Si lo que buscamos es una web estática donde simplemente colgaremos una página que hemos diseñado nosotros en, por ejemplo lenguaje html, la cuestión es fácil de resolver. No hablaremos aquí de como hacer una página web, es decir, del contenido del servidor, ya que eso es una cuestión que concierne a otro tipo de libros y manuales de programación o programas orientados al diseño web.

La cuestión será tan fácil de resolver como instalar nuestro servidor web con sus respectivas dependencias.

Si estamos ante un sistema basado en paquetes .rpm, lo que tendremos que hacer es (suponiendo que instalamos desde CD) buscar entre nuestros discos el paquete Apache e instalarlo cubriendo todas sus dependencias (desde ciertos entornos gráficos puede llegar a ser tan sencillo como hacer un “doble click” con el ratón). El subdirectorio en el cual deberemos colocar nuestra web para que sea accesible a través de Internet, en las distribuciones basadas en .rpm varía ya no sólo entre distribución, sino que también muchas veces según la versión de la misma, con lo cual la mejor recomendación es leer la página del manual concerniente a la instalación de Apache, donde se nos dirá cual es el directorio en el que tenemos que sustituir la página de muestra de Apache por la nuestra.

En un sistema Debian GNU/Linux para algo sencillo, será tan fácil como instalar Apache:

```
# apt-get install apache
```

Con esta orden se instalarán el servidor Apache y los paquetes adicionales necesarios.

Debian Woody coloca como directorio para el servidor web “/var/www/”, lo único que se deberá hacer será colocar en este subdirectorio nuestra página web y reiniciar el servidor Apache (si no ha sido iniciado por defecto), para lo cual la orden será:

```
# /etc/init.d/apache start
```

Ya podemos disfrutar de nuestra propia página web estática. Si lo que se busca es una web más animada, en las secciones siguientes se expone como instalar y conseguir una web dinámica, aunque es más compleja de conseguir.

10.2.3. Instalación desde archivo de fuentes

Podemos obtener la ultima versión de Apache descargándola desde la página web <http://httpd.apache.org/download.cgi>. Aprovecharemos el tiempo de la descarga para crear un usuario y un grupo bajo los que funcionará el servidor web. Para ello escribimos en una consola con privilegios de superusuario lo siguiente:

```
groupadd Apache
useradd Apache -c “Servidor Web” -d /dev/null -s /bin/false
```

Una vez con el archivo de fuentes en nuestro poder, vamos a proceder a la instalación como si de cualquier otro programa se tratase, pero prestando una especial atención al script de configuración, ya que debemos activar algunos módulos en la compilación. Para ello ejecutaremos las siguientes instrucciones:

```
aklis@venus:~$ tar zxf httpd-2.0.XX.tar.gz
aklis@venus:~$ cd httpd-2.0.XX
aklis@venus:~/httpd-2.0.XX $ ./configure \
    --prefix=/usr/local/Apache \
    --enable-so\
    --enable-info=shared\
    --enable-cgi=shared\
    --enable-access=shared \
    --enable-rewrite=shared\
    --enable-speling=shared\
    --enable-usertrack=shared\
    --enable-deflate=shared \
    --enable-ssl=shared\
    --enable-mime-magic=shared
aklis@venus:~/httpd-2.0.XX $ make
aklis@venus:~/httpd-2.0.XX $ su -
aklis@venus: ~/httpd-2.0.XX # make install
```

El parámetro “prefix” indica dónde se va a instalar el httpd y todos sus archivos asociados, en nuestro caso, en /usr/local/Apache.

Prestémosle ahora interés a los módulos que hemos compilado, y cuál es su utilidad:

Nombre	Descripción
S.O	Habilita la carga de módulos dinámicos
Info	Proporciona información sobre el demonio Apache mientras se está ejecutando.
speling	Corrige los errores mas comunes al escribir una URL.
Deflate	Si el navegador lo soporta, habilita la compresión de datos para ahorrar ancho de banda y aumentar la velocidad
Usertrack	Gestión de sesiones por usuario.
Mime-magic	Detección automatiza de tipos mime.
ssl	Servidor seguro (http sobre ssl).
rewrite	Permite la búsqueda de expresiones regulares en url's.
Cgi	Permite el uso de cgi's.

En todos los casos construimos los módulos como “shared” (compartidos), para no engordar demasiado el binario httpd (el servidor web en sí), y por otro lado, para facilitar la configuración y permitirnos añadir nuevos módulos sin necesidad de tener que empezar desde el principio.

10.2.4. Configuración básica

Una vez finalizado el proceso de instalación, todos los archivos necesarios se encontrarán en /usr/local/Apache siguiendo una jerarquización parecida a la de “/” (existe bin/,

conf/, lib/,...). Copiamos el archivo con nombre `Apachectl` ubicado en el subdirectorio `bin/` a `/etc/init.d`, ya que en función del primer parámetro que le indiquemos nos servirá para arrancar (start), parar (stop), reiniciar (restart) o recargar la configuración del apache (graceful).

Aunque la configuración que acompaña al servidor no aprovecha al 100% las posibilidades de apache, con unos ligeros retoques podemos conseguir que funcione con un rendimiento más que aceptable.

Lo primero que haremos será cambiar el usuario que utiliza el `httpd` para poner el que nosotros hemos creado mientras bajábamos el código fuente. Las directivas de la configuración que tenemos que cambiar son las referentes a “User” y “Group” (aproximadamente en la línea 260).

Otro cambio que debemos realizar es en la directiva “ServerName”, donde tenemos que poner el nombre de nuestro servidor web. No es necesario que sea real, sólo es el nombre que mostrará el servidor en los mensajes de error.

Otro detalle a tener en cuenta son los logs. Por defecto se encuentran en el <directorio del Apache>/logs/. Es conveniente revisarlos para controlar el correcto funcionamiento del servidor y no está de más rotarlos cada cierto tiempo, bien usando el programa “logrotate”, o con algún script situado en el denominado “cron” del sistema.

El resto de la configuración de un servidor web, es una tarea difícil para la que no está pensada este manual. Al instalar Apache, la documentación incluida en el proyecto es accesible a través de <http://localhost/manual/> y es recomendable borrarla una vez que el sistema se encuentre en funcionamiento.

10.3. MySQL

10.3.1. ¿Qué es?

MySQL es un sistema de gestión de bases de datos rápido, multihilo y multiusuario. Consta de una API que facilita a otras aplicaciones, incluido PHP, su integración con bases de datos.

Al igual que Apache, es un proyecto de código abierto, y en caso de ser utilizado en aplicaciones que no se encuentren bajo licencia GPL, será necesario pagar por una licencia.

10.3.2. Instalación

Al contrario que Apache, los creadores de MySQL prefieren la distribución del producto en formato binario, por cuestiones de estabilidad y rendimiento, aunque también se encuentra disponible para descargar su código fuente. Podemos obtener una versión de MySQL para cualquiera de las arquitecturas soportadas, desde <http://www.mysql.com/downloads/mysql-4.0.html>.

Al tratarse de un programa distribuido en su forma binaria, es decir, ya compilada, no es necesario que lo hagamos nosotros. Para proceder a su instalación seguiremos estos sencillos pasos:

- Añadimos un usuario y un grupo con el nombre “mysql”:

```
addgroup mysql
addgroup -c “MySQL” -d /dev/null -s /bin/false
```

- Descomprimos el archivo obtenido anteriormente y lo copiamos al lugar deseado (en nuestro caso, a `/usr/local/mysql`).
- Nos situamos en el directorio creado en el paso anterior y ejecutamos “`./configure`”.

- Cambiamos el propietario y el grupo de los archivos que se encuentran en el directorio raíz de mysql/data (en nuestro caso, /usr/local/mysql/data):

```
chown mysql:mysql data/
```

- El script de inicio del servidor MySQL se encuentra en “supported-files/mysql.server” y es conveniente copiarlo a /etc/init.d/

10.3.3. Configuración

Una vez arrancado el servidor con la orden “/etc/init.d/mysql start”, procederemos a cambiar la contraseña de administración, eliminar todos los usuarios y bases de datos de ejemplo, y limitar el acceso al servidor para que sólo se pueda acceder al mismo desde la máquina en la que está instalado.

```
/usr/local/mysql/bin/mysql -u root  
mysql> SET PASSWORD FOR root@localhost=PASSWORD('nueva_pass');  
mysql> drop database test;  
mysql> use mysql;  
mysql> delete from db;  
mysql> delete from user where not (host="localhost" and user="root");  
mysql> flush privileges;
```

No está de más cambiar el nombre del usuario “root” por cualquier otro. Para ello, basta con escribir:

```
mysql> update user set user="nuevo_nombre" where user="root";  
mysql> flush privileges;
```

Cuando acabemos, debemos borrar el archivo de historial que se encuentra en “\$HOME/.mysql_history”, ya que en él se encuentran todos los comandos que escribamos en la shell de MySQL, incluidas las contraseñas.

El siguiente paso a realizar es copiar el archivo de configuración de MySQL que más se acerque a nuestras necesidades y equipo, desde su lugar en “<directorio de mysql>/support-files/*.cnf”, a “data/my.cnf”.

Elijamos el que elijamos, como medida de seguridad adicional, se puede desactivar totalmente el acceso por red de nuestro servidor MySQL. Para ello, editamos el archivo “data/my.cnf” con un editor cualquiera, por ejemplo Vim o gedit (el primero funciona en modo terminal y el segundo en modo gráfico), y descomentamos (se denomina así a borrar el símbolo “#” que está al principio de una línea) la instrucción “skip-networking”, eso sí, todas las aplicaciones que hagan uso del gestor de bases de datos tendrán que comunicarse a través del socket ubicado en “/tmp/mysql.sock”, lo que puede dar problemas con aplicaciones que no contemplen esta posibilidad.

Las opciones de crear nuevos usuarios y bases de datos, aunque pueden hacerse desde las herramientas proporcionadas por MySQL, es más sencillo hacerlas desde la herramienta phpMyAdmin, que se explica al final de este capítulo.

10.4. PHP

10.4.1. ¿Qué es?

PHP es un lenguaje de programación desarrollado por el proyecto Apache (los creadores del servidor httpd) y orientado al diseño e implementación de páginas web dinámicas. Está muy relacionado con las bases de datos, ya que proporcionan la posibilidad de incluir información de forma sencilla, cómoda y actualizada.

MySQL será la gasolina que moverá nuestro flamante bolido construido con PHP.

10.4.2. Instalación

Las fuentes desde las que instalar PHP las obtendremos desde <http://www.php.org>. Procederemos a instalarlo de forma normal, siguiendo el mismo método que con el servidor web, pero en este caso, ejecutaremos el script de configuración de la siguiente forma:

```
aklis@venus:~/php-XX $ ./configure \  
  --prefix=/usr/local/Apache/php  
  --with-apxs2=/usr/local/bin/Apache/bin/apxs \  
  --enable-mysql=/usr/local/mysql \  
  --with-config-file-path=/usr/local/Apache/php \
```

Una vez instalado pasaremos a la instalación del archivo de configuración, para ello sólo hay que copiar el archivo de muestra `php.ini-recommended` a `/usr/local/Apache/php/php.ini`

10.4.3. Configuración

Acabada la instalación, procederemos a configurar el servidor web para que interprete las páginas escritas en lenguaje php. Para hacer ésto debemos comprobar si existen las líneas mostradas abajo, en el archivo de configuración de Apache, situado en `"/usr/local/Apache/conf/httpd.conf"`. Para ello abrimos (de nuevo con el editor que prefiramos, VIM o gedit por ejemplo) el archivo y lo comprobamos:

```
LoadModule php4_module      modules/libphp4.so
```

```
AddType application/x-httpd-php php  
AddType application/x-httpd-php-source phps
```

Esta última opción, aunque innecesaria, es útil para ver el código fuente de PHP coloreado.

Aunque el archivo de configuración que se incluye en la distribución fuente de PHP se encuentra bastante bien documentado, la siguiente tabla muestra las opciones más importantes y su significado:

Parámetro	Descripción
safe_mode=ON	Con esta opción activada, las páginas en PHP sólo pueden acceder a ficheros cuyo propietario es el mismo que el de las páginas. Este es uno de los mecanismos de seguridad que proporciona PHP, impidiendo que los usuarios no autorizados accedan a archivos del sistema (p.ej: /etc/passwd)
safe_mode_gid=Off	Cuando la opción safe_mode está activada y la opción safe_mode_gid no, las páginas en PHP pueden acceder no solo a ficheros con el mismo propietario, sino también a aquellos pertenecientes al mismo grupo.
open_basedir = [directorio]	Con esta opción activada PHP sólo podrá acceder a ficheros que se encuentren en el directorio especificado (y en sus subdirectorios)
expose_php=Off	Poniendo esta opción en “off” PHP no mostrará información sobre sí mismo en las cabeceras HTTP enviadas a los clientes.
register_globals=Off	Esta opción hace que todos los parámetros (de entorno, GET, POST, cookies y servidor) sean registrados como variables globales. Éste es un importante problema de seguridad, así que se recomienda encarecidamente desactivar esta opción.
display_errors=Off	Como su propio nombre indica, con esta opción activada se mostrarán todos los errores y avisos de PHP. Esto puede revelar información importante sobre nuestro sistema, como los paths o las consultas SQL. Es muy recomendable poner esta variable a “off” en servidores abiertos hacia Internet, aunque resulta muy útil tenerla activada mientras se trabaja en páginas PHP.
Log_errors= on	Cuando esta opción está activada todos los errores son almacenados en el archivo especificado como parámetro de error_log. Si este archivo no se encuentra accesible, la información es almacenada por el servidor.
file_uploads = On	Permite o no subir archivos al servidor utilizando PHP.
max_file_size = 2M	Tamaño máximo de los archivos que se pueden subir al servidor, si la opción anterior está activada.
disable_functions =	Funciones “prohibidas”. Si en las aplicaciones PHP que utilizamos no existe ninguna llamada a passthru(), no tenemos porqué correr el riesgo de que en un despiste algún visitante malicioso pueda ejecutar comandos o obtener información relevante sobre el servidor, por lo que no está de más deshabilitarlas.

Para comprobar el correcto funcionamiento del intérprete de PHP, así como el correcto funcionamiento de su archivo de configuración, podemos utilizar una pequeña página que nosotros hayamos construido con extensión php y que contenga lo siguiente:

```
<?
phpinfo();
?>
```

10.5. Instalación de una aplicación en PHP y MySQL: phpMyAdmin

10.5.1. Historia

Existen multitud de aplicaciones construidas con PHP y que utilizan como base de datos MySQL. PHPMyAdmin es además un completo gestor de bases de datos en MySQL, y por otro lado, resulta muy útil a la hora de escribir sentencias SQL en PHP.

10.5.2. Instalación

Podemos obtener phpMyAdmin desde su pagina web: <http://www.phpmyadmin.net>, acudiendo a la sección downloads.

Una vez con el programa en nuestro poder, lo descomprimos en un directorio, lo abrimos con un editor (Vim, gedit,...), y añadimos las siguientes líneas en la configuración de Apache:

```
Alias /myadmin "/usr/local/apache/myadmin/"
<Directory "/usr/local/apache/myadmin">
Order deny,allow
Allow from localhost
Deny from all
</Directory>
```

Obviamente será necesario cambiarlo si lo hemos descomprimido en un directorio diferente a “/usr/local/apache/myadmin”.

Una vez descomprimido editamos (Vim, gedit,...) el archivo “config.inc.php” y cambiamos las variables referentes al host, usuario, contraseña y lugar dónde se encuentra el phpmyadmin. Podemos poner sin miedo el usuario “root” de MySQL, ya que esta aplicación no tendrá acceso externo.

Una vez modificado este archivo, abrimos con el navegador la página <http://localhost/myadmin/> y podremos empezar a disfrutar de este magnífico gestor de bases de datos.

Lo primero que haremos será crear un base de datos. Sólo hace falta dirigirse al lugar indicado, escribir el nombre que deseamos y pulsar “Crear”.

Para crear un usuario nos dirigiremos a “Privilegios”, y después a “Agregar nuevo usuario”. Recordemos que el servidor MySQL sólo admite conexiones originadas en la máquina local, por lo que debemos seleccionar en la pestaña “Servidor” la opción “Local”. También debemos ajustar los permisos para el usuario, y como ya vimos antes, evitar darle permisos administrativos. Es una buena política de uso el que cada usuario sólo tenga permisos sobre una base de datos, sobre todo si tenemos varias aplicaciones en php que utilizan distintas bases de datos.

10.6. Epílogo

Una vez que nuestro servidor empieza a funcionar, no podemos olvidarnos de que tiene que encenderse de forma automática en el arranque de la máquina. En sistemas GNU/Linux, si ya hemos copiado los archivos proporcionados por los servidores, tan solo tenemos que hacer un enlace simbólico en el nivel de arranque por defecto de nuestra máquina (en caso de que no se sepa cuál es, está definido en /etc/inittab) a los mismos.

En sistemas *bsd debemos añadirlos al script de arranque del sistema, de tal forma que

se inicien una vez arrancada la red.

También añadiremos que cualquier cambio en las configuraciones, tanto del servidor web como del gestor de bases de datos, no tendrán efecto mientras no se recargue la configuración de los mismos. Por lo demás, después de seguir los pasos de este manual, cualquier persona con una conexión 24h y un ancho de banda suficiente, puede construir su propio servidor web.

CAPÍTULO 11

Linux en ámbitos científicos: aplicaciones.

Humberto Michinel Álvarez*

11.1. Introducción

En este capítulo haremos un pequeño repaso de las aplicaciones que podemos utilizar un día cualquiera de labor investigadora en nuestro trabajo, laboratorio, centro de investigación, etc., con el fin de mostrar que no tenemos una dependencia total y absoluta de otros sistemas operativos.

La investigación científica en los últimos 50 años ha estado fuertemente ligada al desarrollo de las ciencias de la computación, principalmente al desarrollo de computadoras cada vez más potentes y el uso lo más eficiente posible de las mismas. Mucho antes de la aparición de Windows, las computadoras ya eran usadas en labores de investigación, trabajando sobre todo con sistemas UNIX, o basados en los mismos. Muchas de las aplicaciones que se utilizaban en la década de los 70 y principios de los 80 han mejorado con el paso del tiempo y otras muchas han aparecido, usándose en entornos Linux hoy en día debido a su mayor facilidad y rapidez, entre otras muchas ventajas.

Aún así no podemos negar la existencia del problema relativo a que determinado software científico para aplicaciones muy concretas, ya no sólo no tiene equivalente en Linux, sino que la misma empresa que lo desarrolla no provee su equivalente para este sistema operativo. Todo ello es debido a que son muy pocos los investigadores en todo el mundo que emplean esa aplicación en concreto, y el sistema operativo más extendido suele ser Windows (esto es discutible según el ámbito científico de que se trate).

11.2. Scientific Applications on Linux (SAL)

En primer lugar comentaremos brevemente una de las mejores páginas web de software científico en Linux: Scientific Applications on Linux (SAL), cuya página web es <http://sal.kachinatech.com/>. En esta web se halla clasificado por temas casi todo el software científico (libre y no libre) existente para Linux. El listado de aplicaciones es enorme y no vamos a reproducirlo aquí, pues tampoco es el fin de este capítulo, que sólo trata de dar una visión general de las posibilidades de Linux en ámbitos científicos.

El buscador de SAL permite hacer una búsqueda orientada por tipo de software, campo de aplicación, etc.

* Área de Óptica. Facultade de Ciencias de Ourense. (Universidade de Vigo)

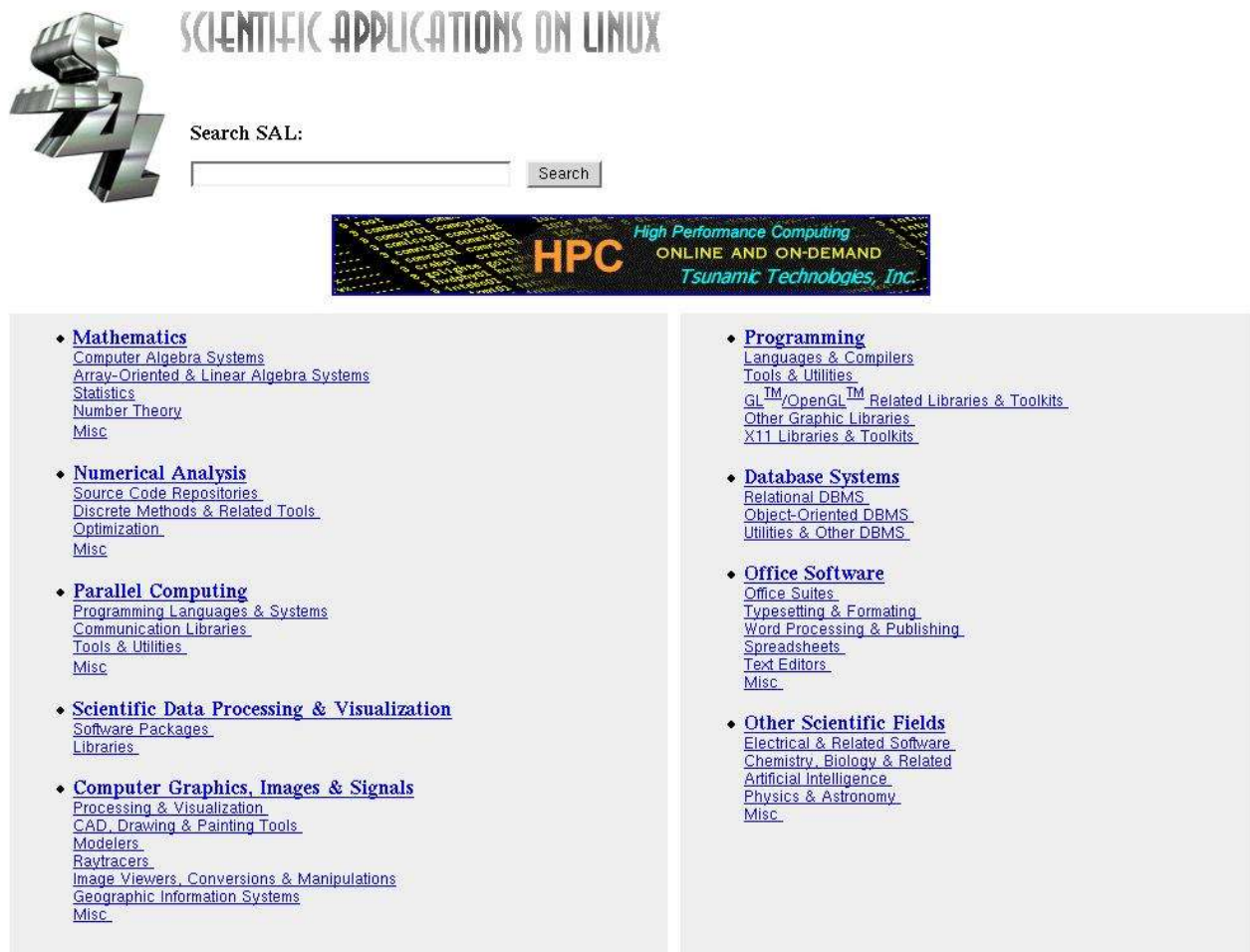


Ilustración 1 Scientific Applications on Linux

11.3. Cálculo Numérico: C/C++ y MATLAB

En nuestro caso, los programas de cálculo numérico son el código desarrollado en C/C++ o en matlab.

11.3.1. Código en C/C++

Para el desarrollo de código propio, dependiendo del problema a resolver, desde el punto de vista de Linux la aplicación de interés es el compilador. El que viene con el sistema operativo es gcc, habitualmente usado como parte de ficheros Makefile que compilan, enlazan y ejecutan el código resultante.

Ejemplo de makefile sencillo con gcc:

```
TASK = comp link rm
all: $(TASK)
comp: *.c
gcc -c *.c

link: *.o
gcc *.o -lm -lcurses

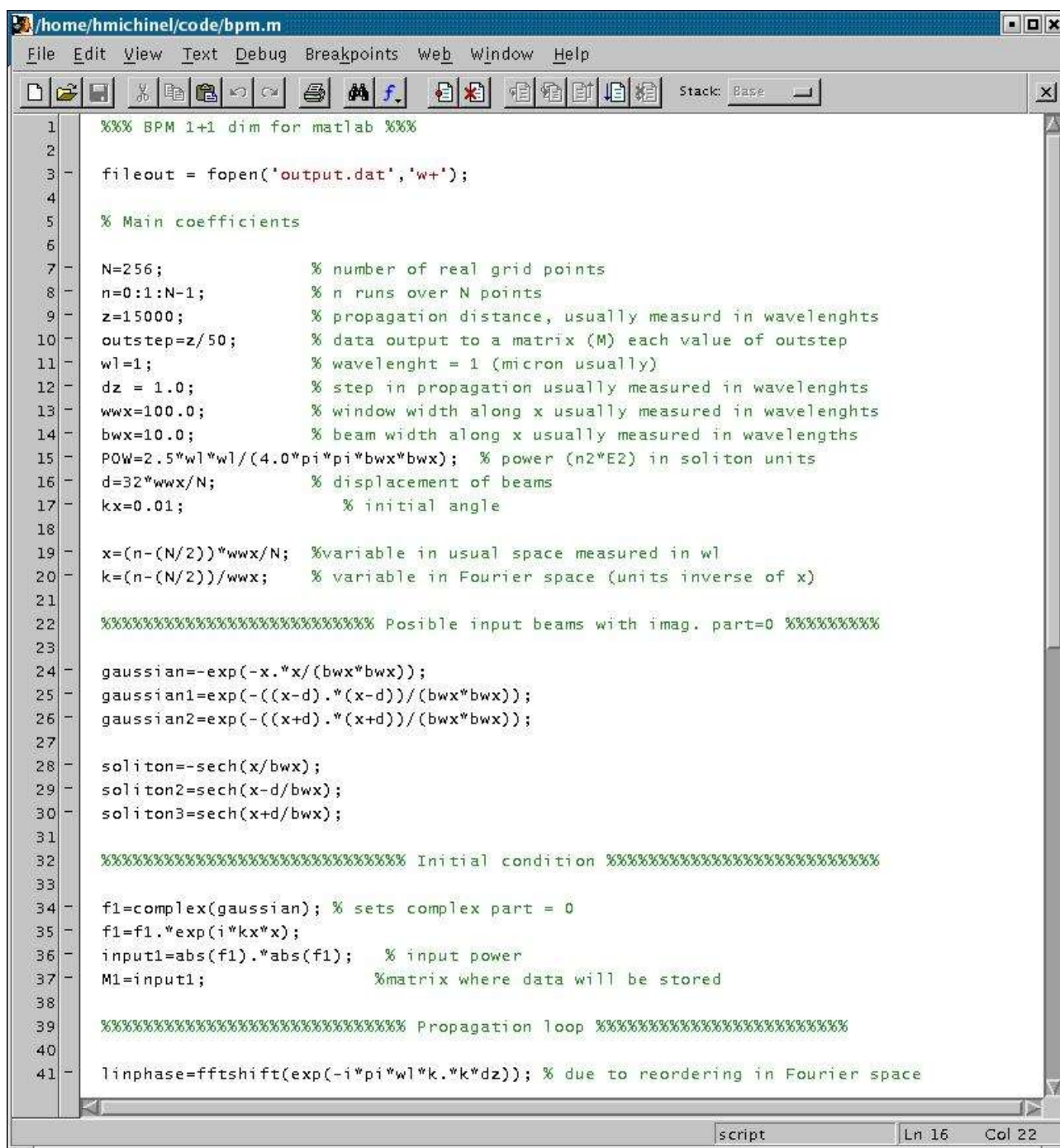
rm: *.o
rm *.o
```

El resultado es un ejecutable denominado a.out que puede ser utilizado en cualquier otro ordenador con Linux sin necesidad de ser compilado. En el caso de tener enlaces dinámicos con determinadas librerías será necesario asegurarse de que el árbol de directorios es el mismo que el del ordenador en el que se compiló el programa.

11.3.2. Matlab

A pesar de ser un programa de uso comercial, MATLAB se halla como software con licencia de campus en muchas universidades y es una herramienta especialmente útil en determinados casos, dependiendo del tipo de problema que se quiera resolver. Si no se necesitan unas herramientas gráficas sofisticadas, puede usarse “octave” (<http://www.octave.org/>) que es el clon libre y gratuito de Matlab para Linux.

En la imagen inferior se ve el entrono de Matlab y el tipo de código, en muchos casos similar a C, que se emplea. Las posibilidades gráficas se comentan más adelante.



```

/home/hmichinel/code/bpm.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base

1  %%% BPM 1+1 dim for matlab %%%
2
3  fileout = fopen('output.dat','w+');
4
5  % Main coefficients
6
7  N=256;           % number of real grid points
8  n=0:1:N-1;      % n runs over N points
9  z=15000;        % propagation distance, usually measured in wavelengths
10 outstep=z/50;    % data output to a matrix (M) each value of outstep
11 wl=1;           % wavelength = 1 (micron usually)
12 dz = 1.0;       % step in propagation usually measured in wavelengths
13 ww=100.0;       % window width along x usually measured in wavelengths
14 bw=10.0;        % beam width along x usually measured in wavelengths
15 POW=2.5*wl*wl/(4.0*pi*pi*bw*bw); % power (n2*E2) in soliton units
16 d=32*ww/N;      % displacement of beams
17 kx=0.01;        % initial angle
18
19 x=(n-(N/2))*ww/N; % variable in usual space measured in wl
20 k=(n-(N/2))/ww;   % variable in Fourier space (units inverse of x)
21
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Possible input beams with imag. part=0 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23
24 gaussian=-exp(-x.*(x/(bw*bw)));
25 gaussian1=exp(-((x-d).*(x-d))/(bw*bw));
26 gaussian2=exp(-((x+d).*(x+d))/(bw*bw));
27
28 soliton=-sech(x/bw);
29 soliton2=sech(x-d/bw);
30 soliton3=sech(x+d/bw);
31
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Initial condition %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33
34 f1=complex(gaussian); % sets complex part = 0
35 f1=f1.*exp(i*kx*x);
36 input1=abs(f1).*abs(f1); % input power
37 M1=input1;             %matrix where data will be stored
38
39 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Propagation loop %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40
41 linphase=fftshift(exp(-i*pi*wl*k.*k*dz)); % due to reordering in Fourier space

```

Ilustración 2 Imagen del entorno MATLAB

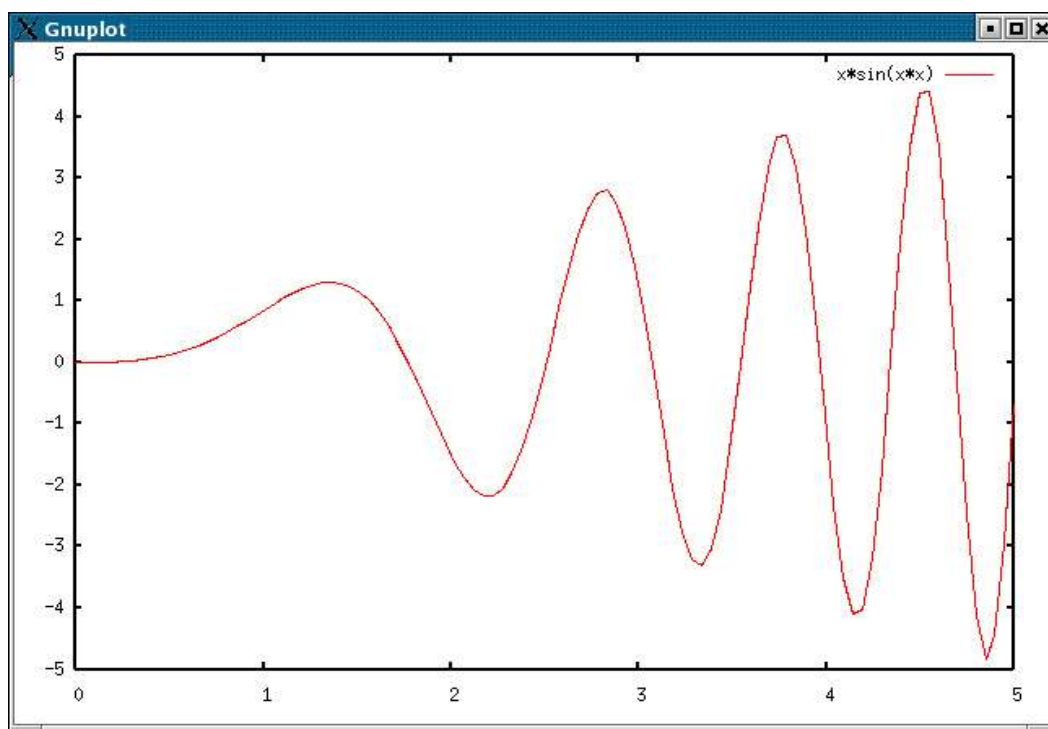
11.4. Gráficos: gnuplot, Xmgrace y MATLAB

11.4.1. gnuplot

gnuplot es un sencillo programa de gráficos que viene incorporado en la mayoría de las distribuciones de Linux. Se ejecuta en línea de comandos y también permite cargar scripts en los que el conjunto de instrucciones puede ser mas largo. Por ejemplo:

```
set nokey
set ticslevel 0
set xrange [0:512]
set yrange[0:25]
set hidden3d trianglepattern 2
splot 'ej1' matrix w l
pause -1
set term postscript solid
set output 'ejemplo1.eps'
splot 'ej1' matrix w l
set term postscript enhanced solid 'Times' 25
set nosurface
set view 0,0
set contour
set xlabel 'x-axis' 'Times,30'
set label 'prop. distance' at graph 1.2,.25 rotate font 'Times,30'
set output 'cejemplo1.eps'
splot 'ej1' matrix w l
```

Comandos sencillos como por ejemplo: **pl $x*\sin(x*x)$** dan como resultado:



11.4.2. Xmgrace

Éste es uno de los mejores programas para ajuste de datos experimentales. Xmgrace funciona en casi todo tipo de sistemas operativos basados en estilo UNIX, (por descontado también en Linux). Xmgrace tiene la ventaja de que, tras haber sido portado para sistemas Windows, su uso se ha difundido, por lo cual a las personas que usen esta aplicación en otros sistemas operativos no les costará cambiarse a Linux, ya que esta herramienta no es que se haya portado a Linux, sino que

ya existía antes que para Windows.

El entorno de Xmgrace en X11 es de muy fácil manejo y dispone de un manual muy completo. Entre sus características más reseñables están la posibilidad de exportar las gráficas obtenidas en varios formatos, así como soportar otros muchos distintos a los que puede generar. Xmgrace puede usarse para hacer gráficas bidimensionales muy complejas, como la de la figura:

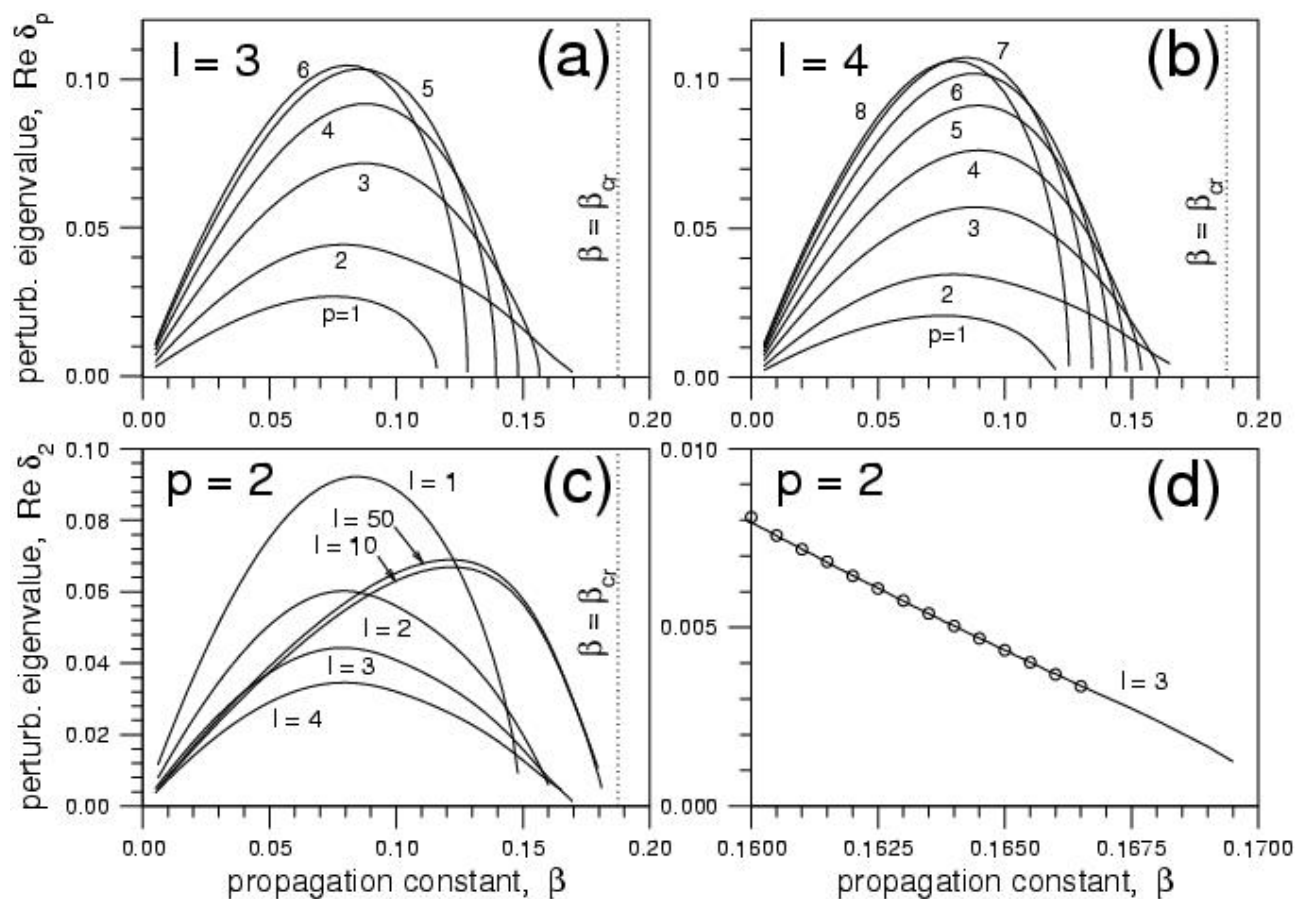
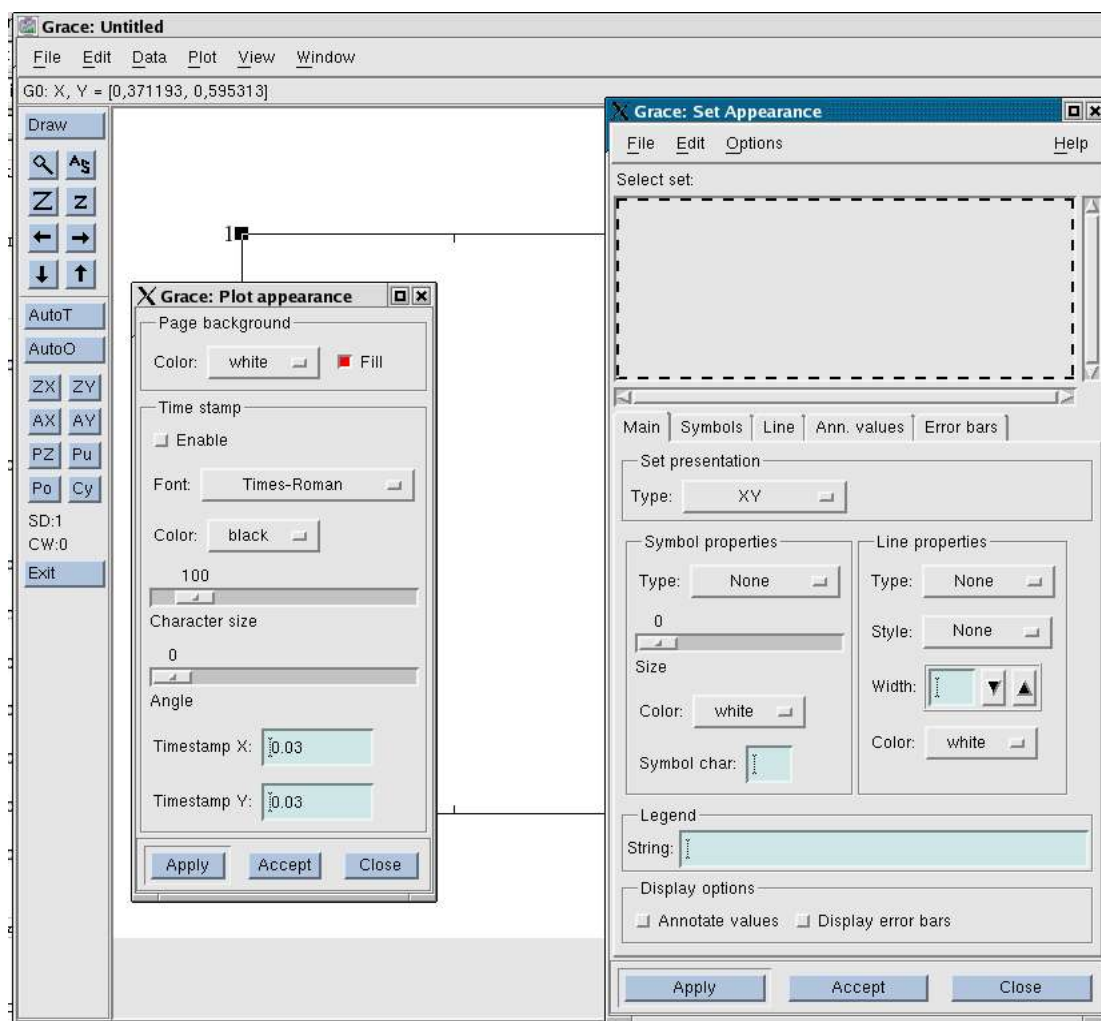


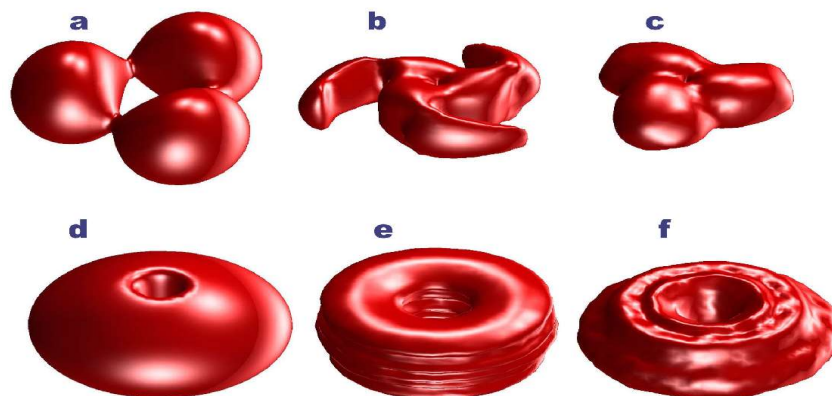
Ilustración 3 Ejemplo de gráfica realizada con Xmgrace

El programa también dispone de herramientas de análisis, permitiendo correlaciones, histogramas, etc.



11.4.3. Matlab

Las posibilidades que ofrece matlab para la realización de gráficas complejas son bien conocidas. Una de las más útiles son sus avanzadas rutinas de renderización, que permiten gráficos como los que a continuación se muestran:



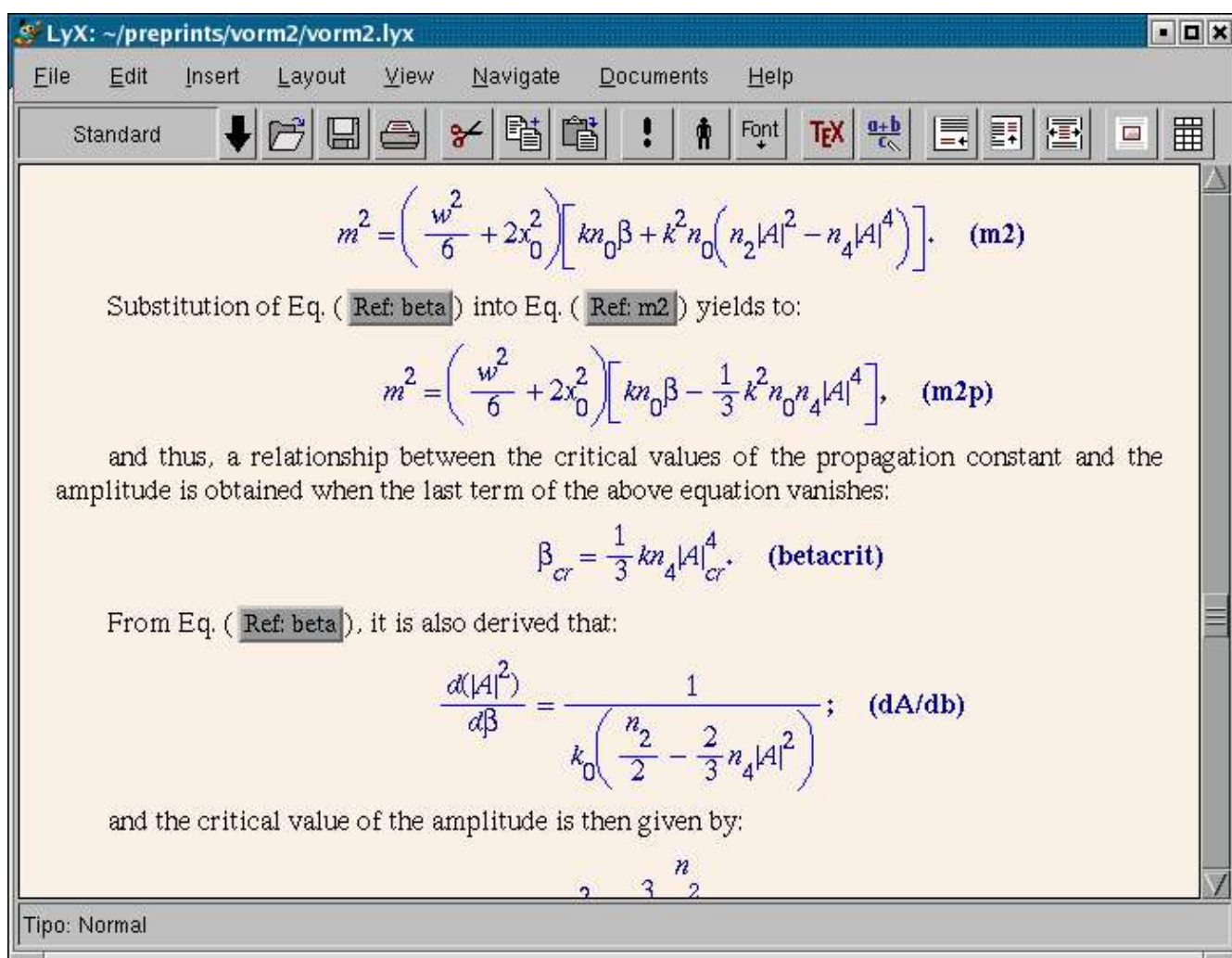
11.5. Gráficos vectoriales: Xfig y Killustrator

Dos de las aplicaciones más potentes para gráficos vectoriales son Xfig y Killustrator. El primero de ellos tiene un entorno más primitivo, aunque a largo plazo es una herramienta más potente si se tiene la suficiente paciencia como para aprender el manejo detallado.

11.6. Procesadores de texto

11.6.1. LaTeX/LyX

Desde el punto de vista científico, sin duda el procesador de texto más potente es TeX (o LaTeX). Para aquellos usuarios no muy acostumbrados a la compilación con LaTeX (incluido en la mayoría de las distribuciones de Linux) es muy útil disponer de un “front-end” que sea completamente compatible con LaTeX. Uno de los más adecuados y completamente gratuito es LyX, que permite ver las fórmulas matemáticas a medida que se van escribiendo. También resulta muy intuitivo Kile, que permite escribir código en LaTeX de una manera muy pedagógica.



11.6.2. OpenOffice y StarOffice

Las distribuciones más recientes de Linux suelen incluir paquetes de ofimática con procesadores de texto similares al Word de Microsoft. Los más habituales son OpenOffice y StarOffice. También hay otros como Kword y Abiword. Son especialmente útiles para editar desde Linux ficheros .doc sin tener que reiniciar a Windows, no obstante, dependiendo de la complejidad del fichero .doc algunos de estos programas no dan el resultado deseado, pues sus filtros algunas veces no funcionan adecuadamente.

11.7. Tratamiento digital de imagen: Image-Magick y Gimp

Existen dos programas muy interesantes para tratamiento de imágenes. Ambos incorporados a la mayoría de las distribuciones de Linux. ImageMagick es una suite especialmente útil para la conversión de formatos, la cual permite copiar imágenes de casi cualquier tipo de formato conocido a otro en línea da comandos. También se pueden realizar múltiples transformaciones.

Gimp hace más hincapié en el retoque de imágenes, mezcla, efectos tridimensionales,...

11.8. Animaciones: mencoder y mplayer

Recientemente el auge del DVD y los correspondientes “destripadores” (rippers), ha traído a Linux algunas de sus aplicaciones. Desde el punto de vista científico, una de las más notables es la posibilidad de realizar animaciones de gráficos generados por ordenador que se convierten en una pequeña película en formato avi.

Para la conversión se utiliza (en línea de comandos) el programa mencoder o su versión en entorno kde: kmencoder. La animación se puede llevar a cabo con distintos paquetes, como por ejemplo mplayer. Existen otros programas para ejecutar animaciones como “animate” (que es parte de ImageMagick) o “xanim”.

APÉNDICES

Apéndice I

ESTRUCTURA DEL ÁRBOL DE DIRECTORIOS EN LINUX

La estructura de un sistema UNIX/Linux típico es bastante distinta a la de otros sistemas operativos. En Linux todo parte de la idea de que “todo es un fichero”. Por razones no triviales cada archivo de configuración, ejecutable, dispositivo, etc., se encuentra en un lugar determinado del árbol de directorios.

Suele ser común una estructura con los siguientes directorios:

/ Directorio raíz. Donde comienza el árbol de directorios.

/bin Binarios esenciales en modo monousuario para reparación y arranque.

/boot Fichero de inicio (núcleo) y otros ficheros de carga.

/dev Ficheros de dispositivos.

/etc Ficheros de configuración de la máquina, en los que puede haber directorios dependiendo del programa que contenga los ficheros de configuración. Algunos programas guardarán sus ficheros de configuración en /etc o /usr/etc (que podrían ser enlaces).

/etc/skel
 Guardará el "esqueleto" que tendrá en común todo nuevo usuario que se cree.

/etc/X11
 Ficheros de configuración del sistema X11.

/home Directorio donde se guardarán los usuarios.

/lib Bibliotecas compartidas (shared) o librerías dinámicas, necesarias para el funcionamiento del sistema.

/mnt Directorio sobre el que se montarán las sistemas de ficheros.

/proc Información acerca del estado del núcleo.

`/sbin` Directorio que contiene comandos sólo ejecutables por el superusuario.

`/tmp` Este directorio contiene ficheros temporales que pueden borrarse sin previo aviso. o durante el arranque del sistema.

`/usr` Normalmente este directorio se monta desde una partición separada. Debería contener solamente datos compartibles de sólo lectura, de forma que pueda ser montado por varias máquinas que usen Linux.

`/usr/X11R6`
El sistema X-Window, versión 11 distribución 6.

`/usr/X11R6/bin`
Binarios pertenecientes al sistema X11R6 (X versión 11, revisión 6).

`/usr/X11R6/lib`
Bibliotecas de programas (librerías) asociadas a los binarios.

`/usr/X11R6/lib/X11`
Varios ficheros de distinta utilización para las X

`/usr/X11R6/include/X11`
Ficheros de cabecera, necesarios para la compilación de las mismas X, o para cualquier otro programa que se quiera compilar.

`/usr/bin`
Binarios para el funcionamiento del sistema. Deberán estar aquí todos aquellos programas que puedan usar los usuarios, excepto los programas que sean del administrador, que debieran estar en `/usr/sbin`.

`/usr/bin/X11`
Binarios de las X11, generalmente enlace a `/usr/X11R6`.

`/usr/dict`
Diccionarios de palabras para distintos correctores ortográficos.

`/usr/etc`
Directorio donde se guardan los ficheros de configuración de los distintos programas. Los programas que funcionen en `/usr/local/bin`, tendrán generalmente su configuración en `“/usr/local/etc”`.

`/usr/include`
Ficheros de cabecera para el compilador C.

`/usr/include/X11`
Ficheros de cabecera para el compilador C y el sistema X-Windows.

`/usr/include/asm`
Ficheros de cabecera que declaran algunas funciones.

/usr/include/linux

Información acerca de la versión del sistema Linux. Estas cabeceras son necesarias para la compilación de cualquier programa.

/usr/include/g++

Ficheros de cabecera para usar con el compilador GNU C++.

/usr/lib

Bibliotecas de programas (librerías) y ejecutables que son requeridos para el funcionamiento de algunos programas.

/usr/lib/X11

Librerías para las X.

/usr/lib/zoneinfo

Ficheros para la información de la zona horaria.

/usr/local

Aquí es donde van típicamente los programas que son locales a la máquina.

/usr/local/bin

Aquí van los binarios de los programas locales a la máquina.

/usr/local/doc

Documentación local.

/usr/local/etc

Ficheros de configuración instalados localmente.

/usr/local/lib

Aquí van los ficheros asociados a los programas instalados localmente.

/usr/local/man

Manuales

/usr/local/src

Código fuente para los programas instalados localmente.

/usr/man/<locale>/man[1-9]

Aquellos sistemas que den cabida a varios usuarios de distintas nacionalidades, podrán tener en la cadena <locale> el lenguaje al que pertenece cada ayuda.

/usr/sbin

Programas binarios para la administración del sistema.

/usr/src

Ficheros fuentes (incluido el núcleo).

/usr/src/linux

Núcleo en código fuente.

/usr/tmp

Directorio que contiene información temporal.

/var

Contenedor de información, como registros de último acceso, colas de impresión, peticiones, PIDs...

/var/lock

En este directorio se crean los ficheros de bloqueo. La convención para nombrar los ficheros de bloqueo es LCK.<device> donde <device> es el nombre del dispositivo en el sistema de ficheros. El formato utilizado es el de los ficheros de bloqueo HDU UUCP, ésto es, ficheros de bloqueo que contienen un PID como un número decimal ASCII de 10 bytes, seguido por un carácter de salto de línea.

/var/log

Ficheros "log" misceláneos.

/var/preserve

Copias de seguridad del editor VI.

/var/run

Ficheros de variables de ejecución, como los ficheros que contienen los identificadores de proceso (PIDs) y la información de los usuarios "logueados". (utmp). Los ficheros de este directorio se suelen borrar cuando se arranca el sistema.

/var/spool

Ficheros en cola para varios programas.

/var/spool/at

Trabajos en cola para at.

/var/spool/cron

Trabajos en cola para cron.

/var/spool/lpd

Trabajos en cola para su impresión.

/var/spool/mail

Buzones de los usuarios.

/var/spool/smmail

Ficheros en cola para el programa smmail de distribución de correo.

/var/spool/news

Directorio de encolado para el subsistema de noticias.

`/var/spool/uucp`

Ficheros en cola para uucp.

`/var/tmp`

Como `/tmp` este directorio contiene ficheros temporales, almacenados durante un tiempo no especificado.

Apéndice II

MANUAL BÁSICO DE VIM

VIM es un editor de texto visual, de pantalla completa, basado en el editor de línea ex. Es un editor poco intuitivo y con mala prensa entre los estudiantes que dan sus primeros pasos en UNIX/Linux, pero por otra parte es el editor favorito de los usuarios avanzados y de muchos programadores. Es además un editor que se puede encontrar en cualquier sistema UNIX, desde antiguas estaciones Sun Solaris o HP-UX hasta las más recientes distribuciones de GNU/Linux o FreeBSD, OpenBSD, etc. VI es un editor muy potente, que permite hacer complicadas operaciones en grandes ficheros con unos pocos comandos, por lo que su aprendizaje puede ahorrarnos mucho tiempo. Otra ventaja de VI es que al ser tan corriente suele encontrarse incluso en disquetes de rescate. Lógicamente poco se puede rescatar si no se sabe manejar el único editor disponible en un momento de emergencia. El manejo de VI es realmente incómodo si nos enfrentamos a la versión clásica. Por ejemplo, no podemos usar los cursores para movernos por el texto, debemos pasar al llamado "modo comando" y utilizar letras para movernos.

Aquí hablaremos de VIM. VIM significa "Vi IMproved" (VI Mejorado), y como su nombre indica es un clon mejorado del clásico editor VI. VIM es bastante más amigable que VI, ya que permite un uso más intuitivo (p.ej. los cursores y otras teclas para moverse).

Lo primero que debe aprenderse con VIM es la filosofía de los dos modos de trabajo: el modo comando y el modo edición. El modo comando se utiliza solamente para dar órdenes al editor, decirles que haga cosas como borrar una línea, buscar un patrón, ir a una determinada línea, guardar el fichero, salir, etc. El modo edición se utiliza solamente para escribir texto en el fichero. Es muy importante familiarizarse con esta filosofía de funcionamiento, ya que resulta imprescindible para cualquier operación que se quiera realizar con VIM.

Para ejecutar este editor el comando es:

\$ vi

Aunque conserva el nombre de VI estamos trabajando con VIM. Este comando admite varias opciones que se le pueden pasar como parámetros, por ejemplo el nombre del fichero que queremos editar:

\$ vi fichero

VIM comienza siempre en modo comando, preparado para realizar operaciones sobre el fichero. Una de estas operaciones es pasar al modo edición pulsando la tecla i (Insertar). Para pasar del modo edición al modo comando basta con pulsar la tecla de escape, que llamaremos

<ESC>. A continuación vamos a editar un pequeño fichero de prueba para familiarizarnos con sus comandos básicos.

Comenzamos invocando al editor desde la línea de comandos:

```
$ vi prueba.txt
```

Veremos que en la última línea de la consola aparece lo siguiente:

```
"prueba.txt" [New File]                                0,0-1 All
```

Esta línea es la barra de estado del editor. Es aquí donde teclearemos algunos comandos y donde aparecerá cierta información como el modo en el que estamos, la línea y columna en la que estamos, el porcentaje del documento en el que estamos, etc.

A continuación pulsamos la tecla i para pasar al modo edición. Observamos que la barra de estado se muestra diferente:

```
-- INSERT --                                           0,1 All
```

Tecleamos por ejemplo lo siguiente:

```
Lo pequeño es bello.
```

Cuando tenemos algo escrito, pulsamos <ESC> para pasar al modo comando. Entonces tecleamos la orden :w y pulsamos <Enter>. Veremos como la orden :w aparece en la barra de estado mientras la tecleamos, y luego al ejecutarla se muestra información sobre el resultado, en este caso información sobre el fichero que acabamos de guardar.

```
"prueba.txt" [New] 1L, 21C written                    1,20 All
```

Pasamos nuevamente al modo edición pulsando a (observamos la diferencia con pulsar i) y continuamos escribiendo:

```
---  
Lo pequeño es bello.  
---  
La medida de programar es programar sin medida.  
---  
Software is like sex, it's better when it's free.  
---
```

Observamos cómo podemos movernos libremente por el texto en modo edición utilizando los cursores, las teclas de Inicio, Fin, Av. Pag, Re. Pag., etc. Ésto no puede hacerse en el VI clásico. Volvemos a guardar el fichero con la orden :w.

Ahora pensemos que queremos eliminar esas líneas de tres guiones y cambiarlas por líneas de cinco asteriscos. Pasamos a modo comando, situamos el cursor en una de esas líneas y pulsamos dd. Veremos como la línea entera desaparece. Repetimos lo mismo con las otras líneas. Ahora nos situamos en la primera línea y pasamos a modo edición, escribimos cinco asteriscos y pulsamos <Enter>. Volvemos al modo comando, situamos el cursor en la nueva línea de asteriscos

y pulsamos de nuevo dd, vemos como la línea desaparece. Situamos el cursor en el principio de la línea y pulsamos P, vemos como la línea que habíamos borrado se inserta antes del cursor. Si situamos el cursor al principio de la tercera línea y pulsamos p vemos como la línea se inserta después del cursor. Añadimos las restantes líneas de asteriscos donde estaban las de guiones.

Para deshacer cualquier operación realizada pulsamos en modo comando la tecla u. Para salir del editor sin guardar el fichero se usa la orden :q!. Esta operación se suele realizar con mucha frecuencia al principio, cuando se comete algún error grave como teclear una palabra sin pasar al modo edición. En el modo comando cada tecla tiene su función, y es diferente además si está en mayúsculas que si está en minúsculas. Una regla de oro con VIM es: cuando no se sabe con qué comando se hace algo, no probar teclas al azar, y por supuesto cuando se está editando algo importante, guardar el fichero con frecuencia con la orden :w, sobretodo en las primeras semanas de uso.

Con estos pocos comandos es suficiente para la edición básica de textos y programas. En la tabla que se muestra a continuación se resumen éstos y otros comandos:

Comandos básicos de VI

Comando	Descripción
<ESC>	Mientras se teclea un comando lo cancela.
i	Inserta en la posición del cursor (pasa a modo comando).
a	Inserta tras la posición del cursor (pasa a modo comando).
I	Inserta al inicio de la línea (pasa a modo comando).
A	Inserta al final de la línea (pasa a modo comando).
x	Borra un carácter.
r	Reemplaza un carácter.
u	Deshace la última operación realizada (se puede repetir para deshacer varias operaciones).
U	Deshace los cambios efectuados sobre la línea actual.
:q	Salir del editor.
:x	Salir del editor guardando el fichero.
:q!	Salir del editor sin guardar el fichero.
:w	Guardar el fichero.
:w nombrefichero	Guardar el fichero con nombre “nombrefichero”.
nG	Ir a la línea n.
\$G	Ir al final del fichero.
v	Activa el modo de selección (utiliza los cursores para seleccionar texto).
y	Copia en memoria (buffer) el texto seleccionado.
d	Borra el texto seleccionado y lo copia en memoria (buffer).
p	Pega el texto copiado en memoria tras la posición del cursor.
P	Pega el texto copiado en memoria en la posición del cursor.
:syntax on	Activa el coloreado de sintaxis.
:syntax off	Desactiva el coloreado de sintaxis.
/palabra	Busca la cadena “palabra” hacia adelante.
?palabra	Busca la cadena “palabra” hacia atrás.
n	Muestra la siguiente coincidencia de la última búsqueda.

Apéndice III

EL SHELL Y COMANDOS BÁSICOS EN CONSOLA PARA LINUX

El intérprete de comandos es el programa que recibe lo que se escribe en la terminal y lo convierte en instrucciones para el sistema operativo, en otras palabras, el objetivo de cualquier intérprete de comandos es ejecutar los programas que el usuario teclea en el “prompt” del mismo. El “prompt” es una indicación que muestra el intérprete para anunciar que espera una orden del usuario. Cuando el usuario escribe una orden, el intérprete ejecuta dicha orden. En dicha orden, puede haber programas internos o externos: los programas internos son aquellos que vienen incorporados en el propio intérprete, mientras que los externos son programas separados (ej: aplicaciones de /usr/bin,...). En el mundo Linux/UNIX existen tres grandes familias de Shells, pero con la que es más habitual trabajar es con “bash”, un clon libre de la shell estándar AT&T Bourne shell. Éstas se diferencian entre sí básicamente en la sintaxis de sus comandos y en la interacción con el usuario.

Será muy útil disponer de una serie de comandos básicos junto con sus funciones a la hora de trabajar con un sistema Linux. Aquí se expone un pequeño listado.

Linux - comandos básicos

COMANDO	SINTAXIS	FINALIDAD
su	\$ su	transformarse en superusuario
pwd	\$ pwd	verificar cuál es el directorio actual
cd	\$ cd PATH	entrar en un directorio. Ejemplo: \$ cd /home/samu
ls	\$ ls OPCIÓN	listar el contenido de un directorio. Ejemplos: \$ ls \$ ls /home/samu \$ ls -a \$ ls -all
more	\$ more ARCHIVO	mostrar un archivo

less	\$ less ARCHIVO	mostrar un archivo (más recursos que more)
find	\$ find DIRECTORIO PARÁMETROS ACCIONES	comando para buscar archivos por su nombre. Ejemplo: \$ find -name index
grep	\$ grep TEXTO ARCHIVOS	comando para buscar archivos por su contenido. Ejemplo: \$ grep html *
tar	\$ tar cvf ARCHIVO.tar ARCHIVOS \$ tar xvf ARCHIVO.tar	comprimir archivos en un único archivo comprimir archivos
gzip	\$ gzip ARCHIVO	comprimir archivo
gunzip	\$ gunzip ARCHIVO	descomprimir archivo
man	\$ man COMANDO	muestra la página de manual del comando
cp	\$ cp ARCHIVO DIRECTORIO	copiar archivo. Ejemplos: \$ cp teste.txt /home/samu \$ cp teste.txt / home/samu/archivo1.txt \$ cp * /home/samu
rm	\$ rm ARCHIVO \$ rm -r DIRECTORIO \$ rm -rf DIRECTORIO	borrar archivo borrar directorios
mv	\$ mv ARCHIVO	mover y renombrar archivos. Ejemplos: \$ mv teste.txt archivo1.txt \$ mv teste.txt /home/samu \$ mv teste.txt / home/samu/archivo.txt
adduser	\$ adduser USUARIO	crear un nuevo usuario
cat	\$ cat ARCHIVO	concatenar y mostrar archivos

chmod	\$ chmod OPCIÓN ARCHIVO	cambiar permisos de un archivo. u => usuario g => grupo o => otros a => todos + => aumenta permisos - => retira permisos = => altera todos los permisos. Ejemplos: \$ chmod +x teste.txt \$ chmod go-w teste.txt
chown	\$ chown USUARIO ARCHIVO	cambia el propietario de un archivo
chgrp	\$ chgrp GRUPO ARCHIVO	cambia el grupo del archivo
clear	\$ clear	limpiar la pantalla
du	\$ du	informa del tamaño del directorio. Opciones: -s => informa del número de bloques -a => informa del tamaño de cada archivo
df	\$ df	muestra el espacio libre en el HD
echo	\$ echo TEXTO	muestra texto por pantalla
mkdir	\$ mkdir DIRECTORIO	crea un directorio nuevo
passwd	\$ passwd \$ passwd USUARIO	cambia la contraseña
reboot	\$ reboot	reinicia el ordenador
shutdown	\$ shutdown -r now \$ shutdown -r + TIEMPO "MSG"	apaga el ordenador
halt	\$halt	
startx	\$ startx	inicia el entorno gráfico o de ventanas
vi	\$ vi \$ vi ARCHIVO	abre el editor de textos VI
who	\$ who	informa de quién está usando el ordenador
fdformat	\$ fdformat /dev/fd0	formatea un disquete. /dev/fd0 equivale a la unidad a:
mke2fs	\$ mke2fs /dev/fd0	crea un sistema de archivos Linux (ext2) en el disquete. Este disquete sólo será válido en Linux

mkdosfs	\$ mkdosfs /dev/fd0	crear un sistema de archivos dos/windows (FAT). Este disquete será leído en Linux y en Windows
mount	\$ mount /dev/fd0/mnt/floppy	monta un disquete de sistema de archivos Linux. Para usar el disquete en Linux, tendrá que montarlo primero. En el caso del dispositivo /dev/fd0 (unidad a:) fue montado en el directorio “/mnt/floppy”.
umount	\$ umount /mnt/floppy	desmontar un disquete
mount -t vfat	\$ mount -t vfat /dev/fd0 /mnt/floppy	montar un disquete de sistema de archivos Windows

Apéndice IV

EQUIVALENCIAS LINUX-WINDOWS

Descripción del programa, tareas ejecutadas	Windows	Linux
Redes y Conectividad.		
<u>Navegadores Web</u>	Internet Explorer, Netscape / Mozilla, Opera, Phoenix, ...	1) Netscape / Mozilla. 2) Galeon. 3) Konqueror. 4) Opera. [Propietario] 5) Phoenix. 6) Nautilus. 7) Epiphany. 8) Links. 9) Dillo.
<u>Navegadores web para Consola</u>	Lynx para Windows	1) Links. 2) Lynx. 3) w3m. 4) Xemacs + w3.
<u>Clientes de correo electrónico</u>	Outlook Express, Mozilla for Windows, Eudora, Becky	1) Evolution. 2) Netscape / Mozilla messenger. 3) Sylpheed, Sylpheed-claws. 4) Kmail. 5) Gnus. 6) Balsa. 7) Bynari Insight GroupWare Suite. 8) Arrow. 9) Gnumail. 10) Althea. 11) Liamail. 12) Aethera.
<u>Clientes de correo electrónico al estilo MS Outlook style</u>	Outlook	1) Evolution. 2) Bynari Insight GroupWare Suite. 3) Aethera.
<u>Clientes de email al estilo The Bat</u>	The Bat	1) Sylpheed. 2) Sylpheed-claws. 3) Kmail. 4) Gnus. 5) Balsa.
<u>Cliente de email en Consola</u>	Mutt, Pine	1) Pine. 2) Mutt. 3) Gnus. 4) Elm. 5) Xemacs.

Introducción a Linux y al Software Libre

<u>Lectores de noticias</u>	Xnews, Outlook, Netscape / Mozilla	1) Knode. 2) Pan. 3) NewsReader. 4) Netscape / Mozilla. 5) Sylpheed / Sylpheed-claws. 6) Pine. 7) Mutt. 8) Gnus. 9) tin. 10) slrn. 11) Xemacs.
<u>Gestor de Descargas</u>	Flashget, Reget, Getright, Wget	1) Downloader for X. 2) Caitoo (former Kget). 3) Prozilla. 4) Wget (console, estándar). GUI: Kmagoo, QTget, Xget, ... 5) Aria. 6) Axel. 7) Download Accelerator Plus.
<u>Descargador de páginas Web</u>	Teleport Pro, ...	1) Htrack. 2) WWW Offline Explorer. 3) Wget. GUI: Kmagoo, QTget, Xget, ... 4) Downloader for X. 5) Pavuk. 6) Curl (console).
<u>Cientes FTP</u>	Bullet Proof FTP, CuteFTP, WSFTP, ...	1) Gftp. 2) Dpsftp. 3) KBear. 4) IglooFTP. 5) Nftp. 6) Wxftp. 7) axyFTP.
<u>Cientes FTP para Consola</u>	FTP in Far, ftp.exe	1) Ncftp. 2) Lftp. 3) mc. (cd ftp://...)
<u>Cientes IRC</u>	Mirc, Klient, VIRC, Xircon, Pirch, XChat	1) Xchat. 2) KVirC. 3) Irssi. 4) BitchX. 5) Ksirc. 6) Epic. 7) Sirc.
<u>Cientes Locales de Chat sin Servidor</u>	QuickChat	Echat.
<u>Mensajería local con máquinas windows</u>	WinPopUp	1) LinPopUp. 2) Kpopup.
<u>Cientes para Mesajería Instantánea</u>	ICQ, MSN, AIM, ... Trillian ICQ, Miranda	1) Licq (ICQ). 2) Centericq (ICQ, consola). 3) Alicq (ICQ). 4) Micq (ICQ). 5) GnomeICU (ICQ). 6) Gaim. 7) Kopete. 8) Everybuddy. 9) Simple Instant Messenger. 10) Imici Messenger. 11) Ickle (ICQ). 12) aMSN (MSN). 13) Kmerlin (MSN). 14) Kicq (ICQ). 15) YSM. (ICQ, consola). 16) kxicq.

Introducción a Linux y al Software Libre

<u>Cientes Jabber IM</u>	JAJC, Tkabber, Psi, Exodus, WinJab, myJabber, RhymBox, Rival, Skabber, TipicIM, Vista, Yabber	1) Tkabber. 2) Gabber. 3) Psi. 4) Gaim.
<u>Monitorización de páginas web o bandejas de correo</u>	WatzNew	1) Web Secretary. 2) Knewsticker & korn. 3) Mozilla. 4) watch -n seconds lynx -dump
<u>Conferencias en línea con Video/audio</u>	NetMeeting	GnomeMeeting.
<u>Comunicación por voz</u>	Speak Freely	1) Speak Freely for UNIX. 2) TeamSpeak.
<u>Firewall (paquetes de filtración)</u>	BlackICE, ATGuard, ZoneAlarm, Agnitum Outpost Firewall, WinRoute Pro, ...	“iptables” o “ipchains” 1) Kmyfirewall. 2) Easy Firewall Generator. 3) Firewall Builder. 4) Shorewall. 5) Guarddog. 6) FireStarter. 7) Smoothwall. 8) IPCop.
<u>IDS (Detección de Intrusos en el Sistema)</u>	BlackICE, Agnitum Outpost Firewall	1) Snort. 2) Port Sentry / Hostsentry / LogSentry.
<u>Visual route</u>	VisualRoute	1) Xtracroute. 2) VisualRoute for Linux. 3) Mtr. 4) Geotrace.
<u>Filtrando contenidos</u>	Proxomitron, ATGuard, Agnitum Outpost Firewall, Privoxy, MS ISA server, ...	1) Squid. 2) DansGuardian. 3) Squidguard. 4) Privoxy. 5) JunkBuster. 6) Fork. 7) Mozilla.
<u>Restricción de tráfico</u>	???	1) IP Relay.
<u>Contabilidad de tráfico</u>	Tmeter, ...	1) Tcp4me. 2) Getstatd. 3) Ipacct. 4) Ipac-ng. 5) Ipaudit. 6) Lanbilling. 7) SARG. 8) TaLinux. 9) NetUP UserTrafManager. 10) MTRG. 11) NetTop.
<u>Compartiendo archivos clientes/servidor (redes punto a punto)</u>	Morpheus (Gnutella), WinMX, Napster, KaZaA (Fastrack), eDonkey / eMule, TheCircle, Bittorrent	1) LimeWire (Gnutella) 2) Lopster. (OpenNAP) 3) Gnapster. (OpenNAP) 4) Mldonkey. (eDonkey) 5) eDonkey(eDonkey) 6) cDonkey. (eDonkey) 7) Gift client / server (Fastrack) 8) ed2k_gui. 9) Gtk-Gnutella. (Gnutella) 10) Qtella. (Gnutella) 11) Mutella. (Gnutella, consola) 12) TheCircle. 13) Freenet. 14) GNUnet. 15) Lmule. (eDonkey) 16) Bittorrent.

Introducción a Linux y al Software Libre

<u>Programas para trabajar sonido con módem y muchas más funciones -</u>	Venta Fax, PrimaFax	1) Gfax. 2) PrimaFax 3) mgetty + voice + sendfax. 4) vgetty.
<u>Trabajando con Fax</u>	WinFax	1) HylaFax. 2) Fax2Send. 3) Efax.
<u>Dialup</u>	Vdialer, etc.	1) Kppp. 2) X-isp. 3) wvdial. (Front ends: X-wvdial, kvdial, gtkdial). 4) Gppp. 5) Kinternet. 6) Rp3.
<u>Administración remota</u>	VNC, Administrador Remoto, Asistencia Remota, Symantec pcAnywhere, Windows Terminal Server, Rdesktop, Radmin	1) VNC. 2) ssh / openssh. 3) Rdesktop Client. 4) freeS/WAN.
<u>Transmisión de archivos al módem</u>	HyperTerminal, Terminate, etc.	1) Minicom + lrzsz + Kermit. 2) Msterm. 3) Xtel.
<u>Trabajando con ssh</u>	Putty, Irlex, cygwin + ssh	1) Kssh. 2) ssh / openssh. 3) GTelnet. (Telnet, SSH, Rlogin)
<u>Herramientas de monitorización de redes</u>	Dumeter, Netmedic	1) Gkrellm. 2) Big Brother. 3) Etherape. 4) Nagios. 5) Tkined. 6) MRTG. 7) Rrdtool. 8) PIKT. 9) Autostatus. 10) bcnu. 11) mon. 12) Sysmon. 13) Spong. 14) SNIPS. 15) iptraf.
<u>Herramientas para mantenimiento de Redes</u>	HP OpenView, MS SMS, Tivoli	1) HP OpenView agents. 2) Big Brother. 3) Cheops. 4) Tkined.
<u>Analizando protocolos, sniffing</u>	Sniffer, EtherPeek (TokenPeek, AiroPeek), Windump, Ethereal, MS Network Monitor, Iris	1) Ethereal. 2) Tcpdump. 3) Etherape. 4) ipxdump.
<u>Escaneo de redes</u>	ShadowScan, ISS, Retina, Internet Scanner	1) Nessus. 2) Nmap. 3) Bastille. 4) Ettercap.
<u>Routing</u>	MS RRAS	1) Iproute2. 2) iptables o ipchains.
<u>Utilidades y librerías para el testeo Ethernet/IP</u>	lcrzo	lcrzo
<u>Telefonía IP</u>	Buddyphone [Propietario]	1) GNU Bayonne. 2) Openh323.
<u>Compartiendo Datos/archivos</u>	Windows shares	1) SAMBA. 2) SAMBA-TNG.

Introducción a Linux y al Software Libre

<u>Visualizando redes-windows</u>	Network neighborhood	1) SAMBA. 2) KDE Lan Browser, lisa 3) LinNeighborhood. 4) xSMBrowser.
<u>ADSL</u>	-	rp-pppoe.
<u>Trabajando con archivos.</u>		
<u>Administrador de archivos al estilo FAR y NC</u>	FAR, Norton Commander, etc	1) Midnight Commander. 2) X Northern Captain. 3) Deco. 4) Portos Commander. 5) Konqueror.
<u>Administrador de archivos al estilo Windows Commander</u>	Total Commander	1) Midnight Commander. 2) Krusader. 3) Kcommander. 4) FileRunner. 5) Linux Commander. 6) LinCommander. 7) Rox. 8) Emelfm.
<u>Administrador de archivos al estilo Windows</u>	Windows Explorer	1) Konqueror. 2) Gnome-Commander. 3) Nautilus. 4) Endeavour Mark II. 5) XWC.
<u>Inspección rápida de documentos de HTML locales</u>	Internet Explorer	1) Dillo. 2) Konqueror. 3) Nautilus. 4) Lynx / Links.
<u>Visualizador de todas las operaciones con archivos</u>	Filemon	Filemon.
<u>Trabajando con UDF</u>	Roxio UDF Reader, Roxio Direct CD	Linux-UDF.
<u>Sistemas de Software para Escritorio</u>		
<u>Editor de Texto</u>	Notepad, WordPad, TextPad, ...	1) Kedit (KDE). 2) Gedit (Gnome). 3) Gnotepad. 4) Kate (KDE). 5) KWrite (KDE). 6) Nedit. 7) Vim. 8) Xemacs. 9) Xcoral. 10) Nvi.
<u>Editor de Consola para Texto</u>	FAR Editor, ...	1) Vim. 2) Emacs. 3) pico. 4) joe. 5) Fte. 6) Jed.
<u>Editor de texto y código multifinalidad</u>	SciTE, UltraEdit, MultiEdit, ...	1) Kate (KDE). 2) Nedit. 3) CodeCommander. 4) SciTE. 5) Quanta Plus. 6) Vim. 7) Xemacs. 8) Mcedit. 9) Jed.

Introducción a Linux y al Software Libre

<u>Trabajando con compresores de archivos</u>	WinZip, WinRar	<ol style="list-style-type: none"> 1) FileRoller. 2) Gnozip. 3) LinZip. 4) Ark. 5) KArchiveur. 6) Gnochive. 7) RAR. 8) CAB Extract.
<u>Visualizador de PostScript</u>	RoPS	<ol style="list-style-type: none"> 1) GhostView. 2) Kghostview.
<u>Visualizador de PDF</u>	Adobe Acrobat Reader	<ol style="list-style-type: none"> 1) Acrobat Reader para Linux. 2) Xpdf. 3) GV.
<u>Creador de PDF</u>	Adobe Acrobat Distiller	<ol style="list-style-type: none"> 1) Muchos programas Linux WYSIWYG m -> imprimir el archivo -> ps2pdf. 2) Adobe Acrobat Distiller. 3) PStill. 4) PDFLatex. 5) Xfig. 6) Ghostscript. 7) Tex2Pdf. 8) Reportlab. 9) GV.
<u>Criptografía y Cifrado</u>	PGP, GnuPG + Windows Privacy Tools	<ol style="list-style-type: none"> 1) GnuPG + GPA, Kpgp. 2) PGP.
<u>Encriptación de volúmenes de disco</u>	EFS (estándar), PGP-Disk, BestCrypt, Private Disk Light	<ol style="list-style-type: none"> 1) Loop-aes. 2) CFS. 3) TCFS. 4) BestCrypt. 5) CryptFS.
<u>Task scheduler</u>	mstask, nnCron	cron.
<u>Virtual CD</u>	VirtualDrive, VirtualCD, Daemon Tools, ...	<ol style="list-style-type: none"> 1) Virtual CD Kernel Module. 2) "cp /dev/cdrom /mycd" + "mount -o loop /mycd /mnt/cdrom/".
<u>Reconocimiento de Texto (OCR)</u>	Recognita, FineReader	<ol style="list-style-type: none"> 1) ClaraOcr. 2) Gocr.
<u>Trabajando con el escáner</u>	Programas típicos para cada escáner.	<ol style="list-style-type: none"> 1) Xsane. 2) Kooka. 3) Xvscan.
<u>Antivirus</u>	AVG AntiVirus, NAV, Dr. Web, TrendMicro, F-Prot, Kaspersky, ...	<ol style="list-style-type: none"> 1) Dr. Web. 2) Trend ServerProtect. 3) RAV Antivirus. 4) OpenAntivirus + AMaViS / VirusHammer. 5) F-Prot. [Propietario] 6) Sophie / Trophie. 7) Kaspersky for Linux.
<u>Configuración del sistema</u>	Msconfig	<ol style="list-style-type: none"> 1) En la instalación de Red Hat Linux. 2) redhat-config-<característica>. El configurador gráfico de RED HAT 8.0. 3) Linuxconf. 4) Drakeconf. 5) Webmin. 6) yast y yast2 en SuSE. 7) sysinstall en FreeBSD. 8) /dev/hands :).

Introducción a Linux y al Software Libre

<u>Gestor de arranque</u>	System Commander, PowerQuest Boot Magic, ...	1) Grub. 2) Lilo. 3) ASPLoader. 4) Acronis OS Selector. 5) Ranish Partition Manager. 6) osbs. 7) Symon. 8) Smart Boot Manager.
<u>Administrador de las particiones del disco duro</u>	PowerQuest Partition Magic	1) PartGUI. 2) GNU Parted. (GUI - QTParted). 3) Partition Image. 4) fips. 5) Diskdrake (Mandrake). 6) Paragon Partition Manager .
<u>LVM + soft-RAID + parted + ...</u>	-	EVMS.
<u>Software para backup</u>	ntbackup (estándar), Legato Networker	1) Legato Networker. 2) Lonetar. 3) Disk Archive. 4) Bacula. 5) Taper. 6) Amanda. 7) Mondo Rescue.
<u>Hacer imágenes de las particiones de disco</u>	DriveImage, Ghostpe	1) PartitionImage (GUI - PartGUI). 2) dd. 3) Mondo Rescue.
<u>Administrador de tareas</u>	TaskMan (estándar), TaskInfo, ProcessExplorer NT.	1) top. 2) Gtop, Ktop. 3) kSysGuard.
<u>Reconocimiento de texto por voz</u>	MS text to speech	1) KDE Voice Plugins. 2) Festival. 3) Emacspeak.
<u>Programas para comparar archivos</u>	Araxis Merge, Minimalist GNU	1) WinMerge. 2) Mgdif. 3) diff, patch. 4) Xemacs.
<u>Programas para meta-búsquedas</u>	Integrado en el sistema	1) find. 2) Gsearchtool. 3) Kfind.
<u>Monitorización de Sistemas</u>	Integrado en el sistema	1) Gkrellm.
<u>Programa para visualizar logs</u>	???	1) Xlogmaster. 2) Analog. 3) Fwlogview.
<u>Herramientas para recuperación de datos</u>	R-Studio	1) e2undel. 2) myrescue. 3) TestDisk. 4) unrm. 5) Channel 16.
<u>Multimedia (audio / CD)</u>		
<u>Reproductores de música / mp3 / ogg's</u>	Winamp	1) XMMS. 2) Noatun. 3) Zinf. 4) Winamp. 5) Xamp. 6) GQmpeg. 7) SnackAmp.
<u>Consola de música / mp3 / ogg players</u>	mpg123 para Windows, dosamp	1) Cplayer. 2) mpg123 / ogg123. 3) mpg321. 4) Orpheus.

<u>Programas para grabar CD con GUI</u>	Nero, Roxio Easy CD Creator, ...	1) K3b. (KDE) 2) XCDRoast. 3) KOnCd. 4) Eclipt Roaster. 5) Gnome Toaster. 6) CD Bake Oven. 7) KreateCD. 8) SimpleCDR-X. 9) GCombust. 10) WebCDWriter.
<u>Reproductores de CD</u>	CD player	1) KsCD. 2) Orpheus. 3) Sadp. 4) WorkMan. 5) XmcD. 6) Grip. 7) XPlayCD. 8) ccd / cccd. (consola)
<u>Bajando la velocidad de rotación del CD</u>	Slowcd, Cdslow	1) hdparm -E <speed> 2) eject -x <speed> 3) cdspeed.
<u>CD ripping / grabación</u>	Windows Media Player, AudioGrabber, Nero, VirtualDrive, VirtualCD, ...	1) Grip. 2) Audacity. 3) RipperX. 4) tkcOggRipper. 5) cdda2wav. 6) Gnome pToaster. 7) Cdparanoia. 8) Cd2mp3. 9) Dagrab. 10) SimpleCDR-X. 11) RatRip. 12) AutoRip.
<u>Decodificadores Mp3</u>	Lame	1) Lame. 2) Bladeenc. 3) NotLame. 4) L3enc. 5) gogo.
<u>Trabajando con protocolos en tiempo real</u>	RealPlayer	1) RealPlayer. 2) Mplayer + librerías.
<u>Programas para recepción de estaciones de radio</u>	VC Radio, FMRadio, Digband Radio	1) xradio. 2) cRadio. 3) Xmradio. 4) RDJ. 5) RadioActive. 6) XMMS-FMRadio. 7) Gqradio.
<u>Editores de audio</u>	SoundForge, Cooledit, ...	1) Sweep. 2) WaveForge. 3) Sox. 4) Audacity. 5) GNUSound. 6) Ecasound.
<u>Procesadores de audio Multitrack</u>	Cubase	1) Ecasound.
<u>Sound tracker</u>	Fasttracker, ImpulseTracker	1) Soundtracker. 2) Insotracker. 3) CheeseTracker.
<u>Mezcladores de sonidos</u>	sndvol32	1) Opmixer. 2) aumix. 3) mix2000. 4) Mixer_app.

Introducción a Linux y al Software Libre

<u>Editor de notas musicales</u>	Finale, Sibelius, SmartScore	1) LilyPond. 2) Noteedit. 3) MuX2d.
<u>Secuenciador de Midi</u>	Cakewalk	1) RoseGarden. 2) Brahms. 3) Anthem. 4) Melys. 5) MuSE. Más información: Midi-Howto.
<u>Creadores de música</u>	Cakewalk, FruityLoops	1) RoseGarden. 2) Ardour.
<u>Sintetizadores</u>	Virtual waves, Csound	1) Csound. 2) Arts Builder.
<u>Multimedia (Gráficos)</u>		
<u>Visualizador de archivos gráficos</u>	ACDSee, IrfanView	1) Xnview. 2) GQview. 3) Qiv. 4) CompuPic. 5) Kuickshow. 6) GTKSee. 7) xv. 8) pornview. 9) imgv. 10) Gwenview. 11) Gliv. 12) Showimg. 13) Fbi. 14) Gthumb.
<u>Visualizador de archivos gráficos en consola</u>	QPEG	zgv.
<u>Editores simples de gráficos</u>	Paint	1) Kpaint. 2) Tuxpaint. 3) Xpaint. 4) Gpaint. 5) Killustrator.
<u>Editores potentes de gráficos tipo PhotoShop</u>	Adobe Photoshop, Gimp, Paint Shop Pro	1) Gimp, FilmGimp. 2) ImageMagick.
<u>Programas para trabajos con gráficos vectoriales</u>	Adobe Illustrator, Corel Draw 14, Freehand, AutoSketch	1) Sodipodi. 2) xfig. 3) Sketch. 4) Karbon14 y Kontour. 5) OpenOffice Draw. 6) Dia. 7) Tgif. 8) Gestalter. 9) ImPress.
	Corel Draw 9	Corel Draw 9
	Corel PhotoPaint 9	Corel PhotoPaint 9
<u>Programa para decorar texto</u>	Wordart	OpenOffice Draw.
<u>Programas para dibujar con código ASCII</u>	ANSI DRAW, Mazaika	1) CanvASCII. 2) Jave. 3) ANSI Draw.
<u>Convertir los archivos gráficos a pseudográficos</u>	???	aalib.
<u>Visualizadores de Flash</u>	Flash Player	1) SWF Macromedia Flash Player. 2) Flash Player. 3) Plugin de Netscape/Mozilla.
<u>Creación de Flash</u>	Macromedia Flash	1) DrawSWF. 2) Ming.

Introducción a Linux y al Software Libre

<u>Creación de gráficos Web</u>	Macromedia Fireworks	Gimp.
<u>Gráficos 3D</u>	3D Studio MAX, Maya, Povray, ...	1) Blender. 2) Maya. 3) KPovModeler. 4) K3Studio. 5) Moonlight. 6) GIG3DGO. 7) Povray. 8) K3D. 9) Wings 3D. 10) Softimage XSI.
<u>Editores de iconos</u>	Microangelo	1) Gnome-iconedit. 2) Kiconedit.
<u>Pequeños programas de captura de pantallas</u>	Integrado con el sistema (PrintScreen), Snag it, ...	1) Ksnapshot. 2) Xwpick. 3) Xwd, xgrabsc. 4) Motv (xawtv) 5) Streamer (video)
<u>Dibujo de Diagramas y Estructuras de Bases de Datos</u>	Access	1) Dia. 2) Toolkit for Conceptual Modelling.
<u>Multimedia (vídeo y otros)</u>		
<u>Reproductores de vídeo / mpeg4</u>	BSplayer, Zoomplayer, Windows Media Player, VideoLAN	1) Mplayer. 2) Xine. 3) Sinek. 4) VideoLAN. 5) Aviplay.
<u>Reproductores en consola para vídeo / mpeg4</u>	-----	Mplayer.
<u>Reproductores de DVD</u>	PowerDVD, WinDVD, MicroDVD, Windows Media Player, VideoLAN	1) Ogle. 2) Mplayer. 3) Xine. 4) Aviplay. 5) VideoLAN.
<u>Decodificadores de DVD rippers</u>	Gordian Knot	1) Drip. 2) Mencoder.
<u>Creación y edición sencilla de vídeo</u>	Windows Movie Maker	1) iMira Editing. 2) MainActor. 3) Broadcast 2000.
<u>Creación y edición profesional de vídeo</u>	Adobe Premiere, Media Studio Pro	1) iMira Editing. 2) Cinelerra. 3) MainActor. 4) Broadcast 2000. 5) Lives.
<u>Cortando vídeo</u>	Virtual Dub	1) Avidemux. 2) Kino.
<u>Convertidores de vídeo</u>	Virtual Dub	1) Transcode. 2) Mencoder.
<u>Trabajando con sintonizadores de TV</u>	AVerTV, PowerVCR 3.0, CinePlayer DVR, ...	1) Kwintv. 2) Xawtv. 3) Zapping. 4) GnomeTV. 5) Mplayer.
<u>Trabajando con formatos QuickTime</u>	QuickTime Player	1) QuickTime Player. 2) Mplayer + codec Sorenson.
<u>Programas para creación de efectos 2D y 3D, animación y composición</u>	Adobe After Effects	1) Shake. 2) Plugins para Gimp.
<u>generación Landscape / terrain</u>	Bryce	Terraform.

Introducción a Linux y al Software Libre

<u>Framework para desarrollo de aplicaciones en video</u>	???	Gstreamer.
<u>Ofimática/Negocios</u>		
<u>Paquetes de oficina</u>	MS Office, StarOffice / OpenOffice, 602Software	1) Openoffice. 2) Koffice. 3) Gnome Office. 4) Siag Office. 5) TeX, LaTeX, ...
<u>Procesador de Word</u>	Word, StarOffice / OpenOffice Writer, 602Text	1) Abiword. 2) WordPerfect. 3) Ted. 4) StarOffice / OpenOffice Writer. 5) Kword. 6) LyX. 7) Kile.
<u>Hojas de cálculo</u>	Excel, StarOffice / OpenOffice Calc, 602Tab	1) Gnumeric. 2) Abacus. 3) StarOffice / OpenOffice Calc. 4) Kspread.
<u>Dibujo de gráficos</u>	Excel, ...	1) Kivio. 2) Dia. 3) KChart. 4) xfig. 5) Gnuplot. 6) GtkGraph.
<u>Creación de presentaciones</u>	MS PowerPoint, StarOffice Presentation, OpenOffice Impress	1) StarOffice Presentation. 2) OpenOffice Impress. 3) Kpresenter. 4) MagicPoint. 5) Quickshow & gimp .
<u>Bases de datos locales</u>	Access	1) KNoda. 2) Gnome DB Manager. 3) OpenOffice + MySQL. 4) InterBase. 5) BDB. 6) Rekall. 7) StarOffice Adabase.
<u>Paquetes de contabilidad financiera (global)</u>	???	1) Hansa Business Solutions.
<u>Administrador personal de finanzas</u>	MS Money, Quicken	1) GNUcash. 2) GnoFin. 3) Kmymoney. 4) Grisbi.
<u>Administrador de proyectos</u>	MS Project, Project Expert 7	Mr Project.
<u>Software para e-comercio y web business</u>	Weblogic, Websphere, iPlanet	1) Weblogic. 2) JBoss. 3) IBM WebSphere Application Server.
<u>Juegos</u>		
<u>Lista de Juegos en Linux</u>	-	http://www.icculus.org/igfaq/gamelist.php
<u>Programación y Desarrollo</u>		
<u>IDE</u>	Microsoft VisualStudio .net	1) CodeForge. 2) Kdevelop + Qt3 Designer. 3) Eclipse. 4) Glade + Motor or + Xwpe or + any text editor. 5) Emacs, XEmacs.

Introducción a Linux y al Software Libre

<u>Visual C++ IDE</u>	Borland C++ Builder, MS Visual C	1) Anjuta + Glade + Devhelp. 2) KDE Studio Gold. 3) Dev-C++. 4) Kylix. 5) vtkBuilder. 6) foxBuilder. 7) wxDesigner. 8) Arriba. 9) Code Crusader. 10) CodeWarrior. 11) Gbuilder. 12) Source Navigator. 13) TimeStorm. [Propietario] 14) Understand for C++. 15) SlickEdit. 16) Vide.
<u>C++ IDE</u>	Borland Turbo C++ 3.0 para DOS, , Minimalist GNU para Windows (mingw32-gcc.exe)	1) GCC (+ Motor or + Xwpe). 2) LinEdit. 3) Rhide. 4) Wxstudio. 5) Eclipse.
<u>Objetos en Pascal IDE</u>	Delphi	1) Kylix. 2) Lazarus + FPC.
<u>Pascal</u>	Pascal, BP	1) Freepascal. 2) GNU Pascal. 3) Rshell.
<u>Basic</u>	Basic	1) Hbasic. 2) X-basic. 3) Yabasic. 4) SmallBASIC.
<u>Prolog</u>	VisualProlog, Mercury	1) GNU Prolog. 2) Mercury. 3) SWI-Prolog.
<u>Assembler</u>	TASM, MASM, NASM	1) NASM. 2) FLAT Assembler.
<u>Disassembler, ingeniería inversa</u>	SoftIce	ldasm.
<u>Debugger</u>	Minimalist GNU For Windows (gdb.exe)	1) gdb. 2) ddd.
<u>Editor Html WYSIWYG</u>	Macromedia Dreamweaver, Frontpage	1) Netscape / Mozilla Composer. 2) Openoffice HTML editor. 3) Amaya. 4) GINF. 5) IBM WebSphere Homepage Builder.
<u>Editores HTML</u>	HomeSite	1) Quanta Plus. 2) Bluefish. 3) WebMaker. 4) Screem. 5) Toppage. 6) WebDesigner. 7) ScriptEditor. 8) August.
<u>Editor HTML</u>	Arachnofilia	Arachnofilia.
<u>Perl/Python/Tcl IDE</u>	-	Komodo.
<u>IDE de Java</u>	JBuilder	Link: Java Tools. 1) Jbuilder. 2) NetBeans. 3) Eclipse. 4) Sun ONE Studio. 5) Vide.

Introducción a Linux y al Software Libre

<u>IDE para desarrollo de bases de datos Oracle</u>	T.O.A.D., SQL Navigator, PL/SQL Developer	Tora.
<u>CASE-facility para UML</u>	ArgoUML, Together ControlCenter [Propietario]	1) Umbrello UML Modeller. 2) Dia+Dia2Code. 3) PoceidonCE. 4) ArgoUML. 5) Together ControlCenter
<u>Elevados sistemas de CASE</u>	Rational Rose.	Rational Rose for Linux.
<u>Editores HEX</u>	Hiew	1) Biew. 2) KHexEdit.
<u>Compiladores y preprocesadores Clipper</u>	CA-Clipper, The Harbour Project	1) Clip. 2) The Harbour Project. 3) xHarbour.
<u>Plataformas al estilo dot-net</u>	M\$.Net	1) Mono. 2) DotGNU/Portable.NET
<u>Trabajando con CVS</u>	WinCVS, TortoiseCVS, BitKeeper [Propietario]	1) cvs (consola). 2) Cervisia (KDE). 3) Lincvs. (Front-end para CVS) 4) BitKeeper. [Propietario] 5) SubVersion. (enhanced CVSi-like platform + WebDAV -> SCM)
<u>Visual Basic</u>	Visual Basic	Phoenix.
<u>Software para Servidores</u>		
<u>Servidores Web</u>	Internet Information Server, Apache, roxen / roxen2, wn, cern-httpd, dhttpd, caudium, aolserver, boa, ...	1) Apache. 2) Xitami. 3) Thhttp. 4) Tux. 5) PublicFile.
<u>Servidores FTP</u>	Internet Information Server, ServU, War FTP, BulletProof FTP server	1) pure-ftpd. 2) vsftpd. 3) wu-ftpd. 4) proftpd. 5) gl-ftpd. 6) ftp. 7) PublicFile.
<u>Lenguajes para desarrollo web</u>	PHP	PHP.
	Perl	Perl.
	ASP	ASP módulos para Apache.
<u>Motores de bases de datos</u>	MS SQL, MySQL	1) Sybase Adaptive Server Enterprise. 2) PostgreSQL. 3) MySQL. 4) mSQL. 5) SAP DB.
	IBM DB2	IBM DB2
	Oracle	1) Oracle. 2) PostgreSQL.
	Informix	Informix.
	Borland Interbase, FireBird	FireBird.
<u>Servidores Email</u>	MDaemon, Hamster	1) Sendmail. 2) Qmail. 3) Postfix. 4) Exim.

Introducción a Linux y al Software Libre

<u>Email + Groupware</u>	Microsoft Exchange	1) CommuniGate Pro. 2) Bynari's Insight GroupWare Suite. 3) Samsung Contact. 4) Teamware Office para Linux. 5) Novell Netmail. 6) Amphora. 7) Tutos / Apache / PHP / Mysql / Sendmail. 8) Kroupware. 9) SuSe Linux Openexchange Server. 10) phpGroupWare. 11) SCOoffice Mail Server.
<u>Filtro Mail / Eliminador de spam</u>	cygwin+Exim	1) SpamAssassin. 2) Procmil. 3) Mailfilter. 4) Exim.
<u>Descargador de Mail</u>	MDaemon	Fetchmail.
<u>Server / router en un diskette.</u>	-----	Tantos como usted quiera. 1) muLinux. 2) Dachstein (firewall / dhcp). 3) Serverdisk (http / ftp). 4) Fli4l.
<u>Servidores Proxy</u>	MS Proxy Server, WinGate	1) Squid. 2) Paco. 3) Privoxy. 4) Wwwoffle.
<u>Servidores para soportar Java Servlets y JSP, puede trabajar con Apache</u>	Tomcat	Tomcat.
<u>Programas científicos y especiales</u>		
<u>Sistemas Matemáticos al estilo MathCad</u>	Mathcad	Gap.
<u>Sistemas Matemáticos al estilo Matlab</u>	Matlab	1) Matlab. [FTP] 2) Octave. (+ Gnuplot) 3) Scilab. 4) R. 5) Yorick. 6) rlab. 7) Euler.
<u>Sistemas Matemáticos al estilo Mathematica</u>	Mathematica	1) Mathematica. 2) Maxima. 3) MuPad.
<u>Sistemas Matemáticos al estilo Maple</u>	Maple	1) Maple. 2) Maxima. 3) MuPad.
<u>Editor de ecuaciones matemáticas</u>	Mathtype, MS Equation Editor	1) OpenOffice Math. 2) MathMLed. 3) Kformula (Koffice). 4) LyX.
<u>Programas para modelado tridimensional</u>	SolidWorks, ...	ProEngineer Linux.
	CATIA	CATIA.
	SolidEdge	SolidEdge.
<u>Ingeniería</u>	ANSYS	ANSYS.

Introducción a Linux y al Software Libre

<u>CAD/CAM/CAE</u>	Autocad, Microstation	1) Varkon. 2) Linuxcad. 3) Varicad. 4) Cymas. 5) Tomcad. 6) Thancad. 7) Fandango. 8) Lignumcad. 9) Giram. 10) Jcad. 11) QSCad. 12) FreeEngineer. 13) Ocadis. 14) Microform. 15) PythonCAD.
<u>CAD/CAM/CAE, simplificado</u>	ArchiCAD	Qcad.
<u>Sistemas de Publicación asistida por ordenador</u>	MS Publisher	1) Scribus - Desktop Publishing 2) KWord.
<u>Diagramación y Diseño Cartográfico</u>	Microsoft Visio	1) Kivio (Koffice). 2) Dia. 3) KChart. 4) xfig. 5) Tgif + dotty. 6) Tulip.
<u>Software de procesamiento de imágenes geográficas</u>	Erdas Imagine, ER Mapper, ENVI	ENVI.
<u>GIS (Sistema de información Geográfica)</u>	ArcView	1) Grass. 2) Quantum GIS. 3) PostGIS.
<u>Software CNC, controlador de herramientas de máquinas</u>	OpenCNC [Propietario]	EMC.
<u>Sistema de tratamiento de texto avanzado en estilo TeX</u>	MikTex, emTeX (DOS)	1) TeX. 2) TeTeX / LaTeX 3) LyX (WYSIWYM). 4) Kile.
<u>Lenguajes integrados de ambiente estadísticos</u>	S-PLUS	R.
<u>Software econométrico</u>	Eviews, Gretl	1) Gretl.
<u>Emuladores de circuitos</u>	Electronic Workbench	1) Geda. 2) Oregano. 3) Xcircuit. 4) Gnome Assisted Electronics. 5) SPICE. 6) SPICE OPUS. 7) NG-SPICE.
<u>Programas de dibujo de estructuras químicas</u>	Chemdraw, Isidraw	Xdrawchem.
<u>Diseño de esquemas electrónicos</u>	PCAD	1) Eagle. 2) Geda.
<u>Emulador de osciloscopio</u>	Winoscillo	Xoscope.
<u>Medida de la temperatura y voltajes sobre placa base</u>	MBMonitor, PCAlert	KHealthCare (KDE).
<u>S.M.A.R.T-atributos y temperatura del disco Duro</u>	-----	1) smartctl. 2) Hddtemp-0.3. 3) IDEload-0.2. 4) Smartsuite-2.1. 5) Smartmontools. 6) Ide-smart. 7) Smartsuite.

Introducción a Linux y al Software Libre

<u>Test de memoria</u>	SiSoft SANDRA	Memtest86.
<u>Programas para observar temperaturas, fanspeeds, etc</u>	SiSoft SANDRA, SiSoft SAMANTHA	1) Ksensors. 2) Lm_sensors.
<u>Para testeo y pruebas HDD</u>	SiSoft SANDRA, SiSoft SAMANTHA	1) hdparm. 2) Bonnie++. 3) IOzone. 4) Dbench. 5) Bonnie. 6) IO Bench. 7) Nhfsstone. 8) SPEC SFS.
<u>Para testeo y pruebas de vídeo</u>	Final Reality	1) X11perf. 2) Viewperf.
<u>Control de tiempo real</u>	SHA Sybera Hardware Access	DIAPM RTAI - Realtime Application Interface.
<u>Simulador de redes</u>	???	1) NS.
<u>Simulador de redes neuronales biológicas</u>	???	1) Xnbc.
<u>"Sensor para LCD"</u>	???	1) Sensors-lcd.
<u>Visualizador para electrocardiogramas</u>	???	1) ecg2png.
Otros		
<u>Sistemas operativos funcionando desde un CD sin instalarse (Live CD)</u>	-----Imposible-----	Knoppix. GPUL-live CD. Gentoo (live CD). SuSE live-eval CD. Gnome-live.
<u>Herramientas de rescate</u>	Disco de arranque de Windows	1) Disco arranque de Linux. 2) Tomsrptb. 3) BanShee Linux.
<u>Software instalador y desinstalador</u>	InstallShield, WISE, GhostInstaller, Microsoft Installer - análogo del rpm	1) Rpm & frontends. 2) Urpmi. 3) Apt & frontends (synaptic, ...). 4) Apt-rpm.
<u>Software instalador y desinstalador desde archivos fuentes</u>	Minimalist GNU para Windows	1) make install, make uninstall 2) CheckInstall. 3) Sinstall.
<u>Sistema de actualización</u>	Windows Update	1) Ximian Red Carpet. 2) Red Hat Network. 3) MandrakeOnline. 4) SuSE YaST Online Update. 5) Caldera Volution Online. 6) Apt. 7) Gentoo ebuids. 8) Debian GNU/Linux package search. 9) Yum.
<u>Emulador de Spectrum ZX</u>	X128, Speccyal, SpecX, SpecEmu, UnrealSpeccy, ...	1) Xzx. 2) Glukalka. 3) Fuse. 4) ZXSP-X. 5) FBZX.
<u>Máquinas virtuales</u>	VMWare	1) VMWare. 2) Win4Lin. 3) Bochs. 4) Plex86.
<u>Trabajando con Palm</u>	Escritorios Palm	1) Kpilot. 2) Jpilot.
<u>Mover archivos .html a un Palm</u>	iSilo	1) iSilo. 2) Plucker.

Introducción a Linux y al Software Libre

<u>Trabajando con salvapantallas</u>	Desktop properties	1) xset. 2) xlockmore. 3) xscreensaver. 4) kscreensaver.
<u>Chequeando el disco duro</u>	Scandisk	fsck -check o reiserfsck -check.
<u>Intrépretes de Comándos</u>	command.com :), cmd.exe, Windows Scripting Host, 4dos/4NT, Minimalist GNU	1) Bash. 2) Csh. 3) Zsh. 4) Ash. 5) Tcsh.
<u>Emulador de Windows</u>	-	1) Wine. 2) Transgaming WineX. 3) Crossover Office.

Apéndice V

ACRÓNIMOS

AGP (Accelerated Graphics Port): puerto para gráficos con conexión directa a la memoria principal.

API (Application Programming Interface): conjunto de funciones de un sistema definidas para su uso desde un programa.

APM (Advanced Power Management): gestión avanzada de alimentación.

AWK (Aho Weinberger Kernighan): utilidad de programación desarrollada en 1977 por Aho, Weinberger y Kernighan.

BIOS (Basic Input/Output System): sistema básico de entrada/salida.

CPU (Central Processing Unit): unidad central de procesos.

DEB: sistema de empaquetamiento empleado originariamente por Debian GNU/Linux.

DHCP (Dynamic Host Control Protocol): protocolo de control dinámico de host.

DRM:

DoS (Denial of Service): denegación de servicio.

FDL (Free Documentation License): licencia de documentación libre.

FSF (Free Software Foundation): Fundación para el Software Libre.

GM: gestor de memoria.

GNU: proyecto de la Free Software Foundation que pretende proporcionar un sistema operativo completo con todas sus aplicaciones necesarias, incluyendo el código fuente.

GPL (General Public License): licencia de distribución que asegura que cualquier modificación que se realice sobre el programa o software se distribuirá de manera libre con su código fuente y bajo la misma licencia.

GPUL: Grupo de Programadores y Usuarios de Linux.

GRUB (GRand Unified Bootloader): programa de administración de arranque.

HD (Hard Disk): disco duro.

IDE (Integrated Drive Electronics): sistema empleado en unidades de disco duro que incluye la electrónica del controlador junto con el disco.

IDS (Intrusion Detection System): sistema de detección de intrusos.

I/O (Input/Output): acrónimo de entrada/salida.

IP (Internet Protocol): protocolo de internet.

IPC (Interprocess Communication): comunicación entre procesos.

ISA (Integrated Software Architecture): bus de 16 bits para la conexión de placas en los PC.

KDE: Kool Desktop Environment.

LAMP (Linux, Apache, MySQL, PHP): conjunto de servicios de internet que incluye estas cuatro tecnologías.

LaTeX: sistema de preparación de documentos.

LiLo (Linux Loader): programada de administración de arranque.

LuCas (Linux en Castellano): proyecto de traducción de documentación sobre Linux al castellano.

MBR (Master Boot Record): sector de arranque del disco duro.

MySQL: Servidor de base de datos SQL multihilo y multiusuario.

NFS (Network File System): sistema de ficheros en red.

PC (Personal Computer): ordenador personal.

PCI (Peripheral Connect Interface): interfaz de conexión de periféricos.

PID (Process Identifier): identificador de proceso.

PHP: lenguaje de script de alto nivel embebido en páginas HTML y ejecutado en el servidor.

PP: planificador de procesos.

PPP (Point to Point Protocol): protocolo empleado para enviar tráfico TCP/IP a través de un enlace de transmisión serie.

RAM (Random Access Memory): memoria de acceso aleatorio.

RPM (Red Hat Package Manager): sistema de empaquetamiento empleado originariamente por Red Hat Linux.

SAL (Scientific Applications on Linux): aplicaciones científicas para Linux.

SCSI: Interfaz que permite conectar hasta siete dispositivos a un adaptador. Usa un conector paralelo de 8 bits que permite una velocidad de transferencia de 5 MB/s.

SMB (Server Message Block): protocolo de transmisión de datos.

TCP/IP: protocolo común de conexión entre ordenadores en red.

USB (Universal Serial Bus): dispositivo de conexión de periféricos que reduce la necesidad de utilizar tarjetas específicas.

VFS (Virtual File System): sistema de ficheros virtual.

WYSIWYG (What you see is what you get): lo que ves es lo que consigues.

Apéndice VI

BIBLIOGRAFÍA

Más allá de la consultada para escribir este manual, la bibliografía que se muestra a continuación expone algunos de los temas tratados en el libro en mayor profundidad o ampliando los contenidos ya mostrados en él. Esperamos que sea de ayuda para todos los lectores.

Ann Brashares, **Linus Torvalds: Software Rebel (Techies)**, 21st Century Books, 2001. ISBN: 0761319603.

Linus Torvalds y David Diamond, **Just for Fun: The Story of an Accidental Revolutionary**, HarperBusiness, 2002. ISBN: 0066620732.

Richard M. Stallman y Lawrence Lessig, **Free Software, Free Society: Selected Essays of Richard M. Stallman**, Free Software Foundation, 2002, ISBN: 1882114981.

Kernighan, B. W. y Ritchie D. M., **The C Programming Language**, Prentice Hall PTR, 2ª edición, 1988. ISBN: 0131103628.

Eric S. Raymond, **The New Hacker's Dictionary**, MIT Press, 3ª edición, 1996. ISBN: 0262680920.

Bernardo Cascales, Pascual Lucas, José Manuel Mira, Antonio Pallarés y Salvador Sánchez-Pedreño. **El libro de LaTeX**, Prentice Hall, Madrid, 2003. ISBN: 84-205-3779-9.

Bernardo Cascales, Pascual Lucas, José Manuel Mira, Antonio Pallarés y Salvador Sánchez-Pedreño. **LaTeX, una imprenta en sus manos**, ADI, Madrid, 2000. ISBN: 84-920381-9-5.

Daniel P. Bovet, Marco Cesati, **Understanding the Linux Kernel**, O'Reilly Press, 2000. ISBN: 0-596-00002-2.

Christopher Negus, **Red Hat Linux 9 Bible**, John Willey and Sons, 2003. ISBN: 0764539388.

Robert Eckstein, David Collier-Brown, Peter Kelly, **Using Samba**, O'Reilly Press, 1999. ISBN: 1-56592-449-5.

Mendel Cooper, **Advanced Bash Scripting Guide**, Linux Documentation Project.
<http://www.tldp.org/LDP/abs/html/index.html>

Bash Programming - Introduction HOW-TO, Linux Documentation Project,
<http://www.ibiblio.org/pub/Linux/docs/HOWTO/Bash-Prog-Intro-HOWTO>

Arnold Robbins, Dale Dougherty, **Sed & Awk 2nd Edition**, O'Reilly Press, 1997.
ISBN: 1-56592-225-5.

Gerhard Mourani, **Securing and Optimizing Linux: The Ultimate Solution**, Open Network Architecture Inc., 2001. ISBN: 0-9688793-0-6.

Ben Laurie, Peter Laurie, **Apache: The Definitive Guide**, O'Reilly Press, 3ª edición, 2002. ISBN: 0-596-00203-3.

George Reese, Randy Jay Yarger, Tim King, **Managing & Using MySQL, 2nd Edition, Open Source SQL Databases for Managing Information & Web Sites**, O'Reilly Press, 2002. ISBN: 0-596-00211-4.

David Sklar, Adam Trachtenberg, **PHP Cookbook**, O'Reilly Press, 2002. ISBN: 1-56592-681-1.

Apéndice VII

LICENCIA DE DOCUMENTACIÓN LIBRE GNU

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member

of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent.

An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here

XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated

version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this

License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but

may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.