

MANUAL DE LINUX

Álvaro Alea Fdz.

Indice

INDICE..... 1

INTRODUCCIÓN E INSTALACIÓN.....4

 SOFTWARE LIBRE..... 4

Licencias..... 4

Software libre..... 4

 ORÍGENES DE LINUX 5

 CARACTERÍSTICAS DE LINUX 5

32 Bits..... 5

Multitarea.....5

Multiusuario.....6

POSIX..... 6

Compatibilidad..... 6

 Pero Nº 1: Ficheros.....6

 Pero Nº 2 : Programas..... 6

Estabilidad..... 6

Es libre..... 7

Soporte..... 7

Adaptación..... 7

 LAS DISTRIBUCIONES 7

Red Hat 7

Debian.....8

SUSE..... 8

Slackware 8

Mandrake..... 8

Gentoo..... 9

Y que distribución elijo? 9

 DÓNDE CONSEGUIR LINUX 9

Comprarla..... 9

Obtenerla "gratis"..... 9

 INSTALACIÓN DE FEDORA RED HAT..... 9

Inicio de la instalacion..... 10

Opciones de Instalacion..... 10

Instalacion de los paquetes de software..... 11

| | |
|--|-----------|
| <i>Primer Inicio</i> | 12 |
| MANEJO Y CONFIGURACIÓN | 13 |
| USUARIOS..... | 13 |
| SHELL Y COMANDOS..... | 14 |
| <i>Bash</i> | 14 |
| <i>Opciones de Comandos</i> | 14 |
| <i>Moviendose por tu Linux</i> | 15 |
| <i>Conseguir información</i> | 15 |
| <i>Comodines</i> | 16 |
| <i>Tuberías</i> | 16 |
| <i>Uso de unidades de disco</i> | 17 |
| <i>Organizacion de directorios</i> | 18 |
| <i>Otros comandos Interesantes</i> | 19 |
| PROCESOS Y NIVELES DE INICIO..... | 20 |
| <i>Control de Procesos</i> | 20 |
| <i>Servicios</i> | 20 |
| <i>Inetd</i> | 21 |
| ADMINISTRACIÓN DEL EQUIPO..... | 21 |
| <i>Permisos</i> | 21 |
| <i>Apagar el sistema</i> | 22 |
| <i>Instalacion de Software</i> | 23 |
| ENTORNO GRAFICO | 25 |
| EL SISTEMA X WINDOWS | 25 |
| KDE | 26 |
| GNOME..... | 26 |
| ADMINISTRACIÓN AVANZADA | 28 |
| RECOMPILAR EL KERNEL..... | 28 |
| CONFIGURACIÓN DE REDES..... | 29 |
| <i>Iptables</i> | 29 |
| SERVICIOS SAMBA..... | 29 |
| SERVICIOS DE DNS..... | 30 |
| <i>Configuración local</i> | 30 |
| <i>Pdns</i> | 30 |
| <i>Bind</i> | 30 |
| SERVICIOS DE WEB Y FTP..... | 30 |
| <i>Servicios Web</i> | 30 |
| <i>Servicios de FTP</i> | 31 |
| SERVICIOS DE ACCESO REMOTO..... | 31 |
| <i>Servidor X</i> | 31 |

| | |
|--|-----------|
| <i>VNC</i> | 32 |
| <i>Telnet</i> | 32 |
| <i>SSH</i> | 32 |
| APÉNDICE A – EDITOR DE TEXTOS VI..... | 33 |
| RÁPIDO TUTORIAL DE VI..... | 33 |
| <i>Ejecutar vi</i> | 33 |
| <i>Comandos de movimiento del cursor</i> | 33 |
| <i>Borrar texto</i> | 34 |
| <i>Salvar un fichero</i> | 34 |
| <i>¿Qué viene a continuación?</i> | 34 |
| TUTORIAL AVANZADO DE VI..... | 34 |
| <i>Movimiento</i> | 35 |
| <i>Modificación del texto</i> | 36 |
| <i>Copiar y mover bloques de texto</i> | 37 |
| <i>Búsqueda y cambio de texto</i> | 39 |
| APÉNDICE B - SITIOS DE INTERNET..... | 42 |
| APÉNDICE C -RECURSOS..... | 43 |
| MATERIAL EXTERNO..... | 43 |
| EL AUTOR..... | 43 |
| VERSIÓN..... | 43 |
| LICENCIA..... | 43 |

Introducción e Instalación

Software libre

Licencias

El software, como la mayor parte de las cosas que el hombre hace tiene un autor y unas leyes, de copyright, que protege sus derechos.

A diferencia de, por ejemplo, las obras de arte, donde no es posible su (fácil) copia, y la posesión de la misma se delega en alguien (un museo, una persona) que la gestiona. Con el software la facilidad de crear copias idénticas ha hecho que muchos autores creen unas restricciones sobre su creación, que suele aparecer como un contrato de licencia de uso, y es que cuando "compramos" un programa en realidad compramos una licencia, un permiso, para usarlo, normalmente una sola persona y en un solo ordenador. En estos casos el software sigue perteneciendo al autor (o la empresa que lo creó) de ahí la denominación de "software propietario".

Hay que aclarar que en informática la piratería no es, como en la música, la "copia de CDs", sino el uso sin licencia.

Compara esto con lo que pasa con un libro o un CD de música, ¿te imaginas un libro que cuando lo compras te comprometes a que solo tu lo leas, o que solo tú estés permitido leerlo en una determinada habitación, sin que puedas, por ejemplo, ir a leerlo a un parque? , esto es lo que pasa con el software propietario.

Software libre

Existe un grupo de gente que considera que estas limitaciones son excesivas, y que el software debe ser de libre uso.

Pese a lo que pueda parecer esta idea no es contraria al concepto de negocio y sigue siendo posible "cobrar" por su creación o soporte.

La Free Software Foundation (<http://www.fsf.com>) reunió estas ideas en torno a una licencia la GPL que garantiza los derechos del autor y protege al software de que otros lo intenten usar en su beneficio.

Hay que destacar que esta idea de como debe ser el software tiene una componente histórica, pues antes del boom comercial del software todo el software que se producía era libre, y se distribuía con el código fuente, el listado, y aun ahora se pone especial hincapié no solo en la libertad de uso sino en la disponibilidad del código fuente.

Basándose en la GPL la FSF está creando un sistema operativo llamado GNU, del cual están disponibles desde hace mucho tiempo toda una serie de utilidades y desde hace poco tiempo en S.O. Completo denominado GNU/Hurd.

Orígenes de Linux

Entre los sistemas operativos que había hace una década estaba Minix, un sistema operativo tipo Unix, de fuentes públicas, que se había escrito a modo didáctico para los estudiantes de ingeniería informática. Funcionaba en un 8086, por lo que era un poco limitado.

Linus Torvalds un estudiante finlandés de informática que investigando y profundizando en los entresijos de los microprocesadores 386 decidió hacer, partiendo de cero, un sistema operativo, basado en Minix, pero que aprovechara toda la potencia del 386, memoria virtual, multitarea y otras cosas.

Así que empezó a crearlo, las primeras versiones eran poco atractivas, apenas ejecutabas el GCC (un compilador de C creado según el estilo GNU) el bash (el equivalente al command.com).

Pero Linus, lo publicó en Internet, con sus fuentes, y un montón de gente se interesó en él, modificándolo, mejorándolo y añadiéndole cosas, a la vez que Linus lo mejoraba y coordinaba todo el trabajo que hacían el resto de la gente. Y así sigue siendo hoy, cientos de versiones después hasta convertirse en lo que tenemos delante.

La gente de GNU creó un montón de programas para su sistema operativo que gracias a que son software libre son también usados en Linux y por eso a Linux se le llama muchas veces GNU/Linux.

El núcleo de Linux, el kernel, se distribuye bajo la licencia GPL, es un tipo de licencia, dentro de lo que podríamos llamar el Open Source Software, básicamente dice que cojas el programa, lo uses, aprendas, lo mejores y compartas esas mejoras con el resto del mundo. Además la licencia GNU fija una serie de derechos a programador que le protegen, pero en resumidas cuentas:

Un Programa con Licencia GPL puede ser vendido, alquilado, prestado modificado, pero:

- No se puede limitar el número de usuarios, copias o tiempo de uso.
- No se puede cobrar por usar el programa (pero sí por distribuirlo).
- No se puede impedir que otros lo vendan o distribuyan.
- Tienes que dar las fuentes del programa de una manera pública.
- Puedes modificar el programa, o aprovechar parte del código, pero el resultado tiene que seguir la misma filosofía.

Es básicamente lo que llamaríamos un programa Freeware, o Gratis. Linux sigue esta licencia. Por eso te encontraras a revistas que lo regalan a gente que cobra por él, a gente que lo coge de Internet, y lo encontraras en distribuciones, pero de esto hablaremos más adelante.

Características de Linux

32 Bits

Gracias a los 32 bits el sistema es rápido eficaz, seguro y fiable, sin que una aplicación pueda causar problemas a las otras, al no tener que guardar compatibilidad con los sistemas anteriores de 16 bits.

Multitarea

El ordenador puede estar haciendo varias cosas a la vez, y que no tendrás que esperar a que acabe una para hacer otra, la multitarea está controlada por el S.O. no por las aplicaciones, por lo que a diferencia de otros S.O. nunca se te quedará parado por culpa de una mala aplicación que consuma todos los recursos del ordenador. Aquí sí podrás bajar correo de Internet, formatear un disco, imprimir 100 hojas y jugar al Quake a la vez, y sin problemas.

Multiusuario

Si has manejado antes un ordenador, seguramente usarías MAC OS o Windows, en estos S.O. tú eres el único que lo usas, en Linux, puede haber varias personas usando el ordenador, compartiendo el microprocesador, así puedes ponerle un par de pantallas y teclados y estar otra persona navegando por Internet, escribiendo una carta, jugando en su pantalla, mientras tu estas en otra haciendo otra cosa completamente diferente, y estaréis ambos en el mismo ordenador.

Ademas proporciona los elementos necesarios para garantizar la seguridad y la privacidad de los datos entre usuarios.

POSIX

Aunque para los usuarios normales esto importa poco, POSIX es un estándar de la industria, que asegura una calidad mínima en ciertas partes del S.O. y asegura su compatibilidad, a nivel de código, es decir, programas POSIX que funcionan en otros Unix, no tendrán problema para compilarse y ejecutarse en Linux, Para muchas empresas esto es muy importante, a la hora de decantarse por un S.O. u otro (por eso Windows NT es compatible POSIX).

Compatibilidad

Cuando tocamos este tema, siempre pasa lo mismo, la gente dice, JO!!!, No puedo ejecutar mi aplicación favorita XXXX for Windows 95, bueno, tampoco alguien con MAC, ni con IRIX, etc., además en Linux hay peros:

Pero Nº 1: Ficheros

Linux No tiene ningún problema para tragarse cualquier tipo de disco de cualquier cosa que exista, leerlo y usar su contenido, además existen Suites como OpenOffice o Corel WordPerfect que permiten leer y usar ficheros de aplicaciones comunes como puedan ser Word o Excel.

Además cuando se trabaja en red, Linux es capaz de entenderse y de mediar entre todo tipo de redes, permitiendo entornos heterogéneos sin ningún problema.

Pero Nº 2 : Programas

Si puedes ejecutar programas de otros S.O. para MAC tienes basilisk2, capaz de crear un Macintosh virtual y ejecutar MacOs para M68K sin problemas.

Para windows existen varios programas que permiten hacer funcionar programas de windows, crossoffice para entornos de oficina, wineX para juegos, y la versión libre de estos Wine, que permite ejecutar la mayoría de los programas.

Si el programa es para MS-Dos existe DosEmu, un emulador de MS-Dos (bueno no exactamente, mas bien habría que decir de 386) donde podrás ejecutar a pantalla completa, como en la realidad, o en ventana de X Windows, cualquier programa para este S.O.

Además de estos existen vmware (comercial) y bosh que crean PC virtuales donde ejecutar cualquier sistema operativo.

Estabilidad

Linux es robusto, no se colgara (el S.O. una aplicación si esta mal echa por supuesto que se colgara, pero no afectara al resto del sistema, nunca tendrás que reiniciar el ordenador por que un programa lo ha colgado), hay que notar que es posible bloquear el teclado o la pantalla, pero eso no significa que se cuelgue el ordenador, puedes entrar al ordenador por otro sitio (un terminal, por red) y desbloquearlo, y seguir usándolo, o si se te bloquea un programa mientras estabas conectado a Internet y cogiendo correo, quizás no puedas usar el ordenador, pero seguirá bajando el correo sin problemas.

Es libre

Es decir no te costara nada, no tendrás que pagar licencias, podrás copiarlo, venderlo, instalarlo donde quieras sin problemas, pero lo más importante es que dispones del código fuente, esto significa que si un día te encontrases con un problema del S.O. no tendrías que esperar inútilmente a que su creador decidiese que era un problema importante y crease un service pack para el S.O., tu mismo puedes solucionar el problema. O puedes indicarle a una 3º personal el problema, y esta no tendrá que ser de la empresa que creo el S.O. para poder solucionártelo.

Soporte

Parece mentira, siendo gratis, pero aparte del que te da Mandrake, SUSE, o RedHat si le comprar los CD a estas empresas, existen cientos de personas, de todos los idiomas conocidos, que gustosos te ayudaran a solucionar cualquier problema que tengas con Linux, y en pocos días.

No dudes de unirme a las listas de distribución que hay en Internet, no solo aprenderás, podrás ayudar a otros en los problemas que tengan.

Adaptación

Linux es uno de los S.O. que más rápido evoluciona, se adapta al mercado y soluciona los problemas rápidamente, como por ejemplo el bug F00F del Pentium, Linux fue el primero en tener solución, Soporta el sistema FAT32 de Microsoft antes que sus propios Sistemas Operativos (Windows NT 4), a sido de los primeros en estar disponible para las arquitectura Athlon64 y un largo etc de cosas.

Prácticamente las mayores pegas que se le puede echar a Linux son:

- Dificultad de Configuración – No es exacto, simplemente no hay botones, se hace todo por ficheros de configuración ASCII, además actualmente las distribuciones incluyen su propio GUI (Interfaz Grafica para el Usuario) para la configuración del equipo, aunque esta depende de cada distribución.
- Hay pocos juegos - Los hay, no todos pero sí muy buenos, Quake I & II, Doom I & II. Abuse, Freeciv, Descent, y un monton de ellos sencillos.

Las distribuciones

Linux es libre, eso significa que cualquiera puede cogerlo y vendértelo, o regalártelo, pero Linux además es solo un pequeño programa, el kernel, que solo no vale de nada, así que existen una serie de empresas, o grupos de gente que se dedica a unir Linux a un grupo de aplicaciones, en muchos casos libres también, en otros no, meterlo todo en un CD o discos y distribuirlo.

Veremos por encima las principales que hay:

Red Hat

Red Hat es una de las empresas que más influye en el mundo Linux actualmente, tiene dos tipos de distribución una, la de pago, que incluye software comercial, y otra, la gratuita que sustituye este software comercial por otro con licencia Libre.

A Red Hat hay que agradecerle el sistema de paquetes RPM, que es el adoptado por el LBS (Linux Base Standar es un conjunto de normas para conseguir que un programa funcione perfectamente sin importar en que distribución se instale).

Paquete es un fichero donde se han metido un programa, junto con la información necesaria para su correcta instalación, esto ahorra el tener que compilar el fichero, como se hacía antes, te avisa si falta algo en el ordenador que instalas, y te permite desinstalar el programa sin ningún problema.

Básicamente existen 3 tipos de paquetes, los RPM, los Deb y los Tgz, de estos dos hablare mas adelante.

La instalación de Red Hat muy sencillo con menús que guían al usuario en la instalación, y una instalación grafica permiten, junto a una guía bastante bien echa, que cualquiera pueda instalar esta distribución.

La única pega es cierta costumbre de red hat a apurarse al sacar las distribuciones, lo que hace que sea necesario recoger varios parches de Internet si tienes una versión vieja.

Debian

Esta distribución siempre ha sido famosa por su fiabilidad, estabilidad y por ser completamente gratuita (el resto de las distribuciones suelen incorporar programas shareware, o con licencias restrictivas).

Tiene el sistema de paquetes.deb, bastante más potente que el RPM, aunque menos extendido.

Incorpora la herramienta Alien que le permite utilizar paquetes de cualquier otra distribución.

Su instalación es sencilla, destacando sobre las otras en que dispone de varios paquetes de programas para una misma tarea, llegando a acumular mas de 7000 paquetes.

Como consejo si te decantas por debían, no instales ninguno, y cuando necesites hacer X, busca el paquete que te permita hacer X he instálalo.

SUSE

Esta distribución alemana, basada en RPM, es la segunda opcion empresarial mas utilizada, Incluye KDE y la herramienta de configuración YAST, muy cómoda de usar.

Recientemente ha sido adquirida por Novell, he IBM tiene una gran influencia sobre ella.

Slackware

Esta distribución tiene el mérito de haber sido mantenida por una única persona, usa el sistema de paquetes TGZ (este sistema es muy rudimentario, he incluye muy pocas facilidades para el usuario).

No cumple la organización estándar de ficheros de Linux, y no incorpora ningún programa de configuración con lo que todo se configura a mano.

Así todo es una de las más veteranas, y la gente que la instalo (normalmente por primera vez) es reacia a cambiar, además es ideal para experimentar.

Mandrake

Esta distribución francesa esta basada en red hat y se diferencia de esta en:

- Esta compilada para Pentium, con lo que todo funciona mas fluido, en contra no podrás instalarla en un 486.
- Tiene el sistema de instalación más cómodo y automatizado, auto detectando y configurando la mayoría del hardware.

Esta distribución esta muy indicada para entornos de escritorio.

Gentoo

Esta distribución, de reciente aparicion, se diferencia de las demas por instalar todos los programas desde el codigo fuente, recompilandolos, lo cual es lento, pero asegura la máxima optimización y estar a la ultima en cuanto a versiones, requiere amplios conocimientos de Linux.

Y que distribución elijo?

Esa es una pregunta complicada, lo ideal seria que probases varias y eligieses en función de con cual estas mas a gusto.

Si un usuario de escritorio probablemente la mejor sea Mandrake.

Otra opción es red hat, tan fácil como mandrake pero con mucha mas experiencia en el mercado, esta es también la recomendada si necesitas instalar un servidor, o requieres el respaldo y soporte de una empresa detrás.

Si Quieres un sistema GNU/Linux 100% y gratis, entonces Debian.

Total en el fondo da igual, son todas Linux.

Dónde conseguir Linux

Existen dos opciones, comprarla o conseguirla gratis de internet.

Comprarla

Hay algunas distribuciones como Red Hat Enterprise, solo esta disponible a través de suscripciones, con una cuota periódica, otras requieren un único pago.

En Caja, en una tienda de informática, en grandes almacenes o en la web de la distribución.

Tiene las ventajas de incluir un manual, programas comerciales y soporte técnico por un periodo de tiempo.

Con una revista, tiene la ventaja de poder comprarla en el quiosco de la esquina, y es relativamente barato, como desventaja sola suele ser el primer CD y no siempre tiene la calidad espera.

Por último puedes comprar un conjunto de CDs pregrabado (idénticas a las que te bajarías gratis de internet) de <http://www.linuxiso.com> o <http://www.cheapbytes.com> a un precio muy ajustado.

Obtenerla "gratis"

Puedes bajarla de la pagina principal de la distribución.

Además hay sitios dedicados a almacenar imágenes ISO que luego puedes grabar en CD, como <http://www.linuxiso.org>

Instalación de Fedora Red Hat

Para la instalación de Linux es necesario un ordenador que tenga espacio en disco duro disponible, no basta con que desde otro S.O. p.e. windows aparezca el disco con espacio vacío, este espacio tiene que no pertenecer a ninguna partición.

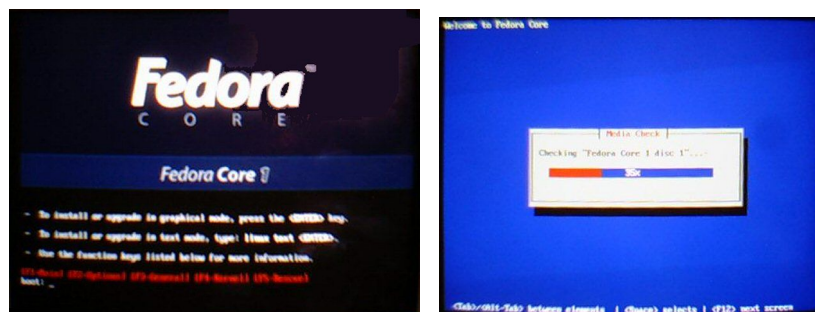
Si se desea conservar el S.O. anterior para un arranque dual, y este ocupa todo el disco duro, habrá que recurrir a una herramienta de terceros, p.e. Partition Magic, para hacer espacio a linux.

Si Linux va a ser el único S.O. del ese ordenador no importa el estado del disco Duro.

El proceso de instalación es muy parecido en todas las distribuciones, para ilustrar este capítulo utilizaremos Fedora Core 1, la versión libre de Red Hat.

Inicio de la instalación

La mayoría de los CD de instalación de Linux son autoarrancables, en ocasiones puede ser necesario buscar en el programa de configuración de la BIOS del ordenador la opción para que arranque desde CD, así que metemos el CD y encendemos o reiniciamos el ordenador.



Pantalla del BootLoader y Verificación del CD

La primera pantalla Pertenece al BootLoader, suele estar en inglés, y además de unas pequeñas instrucciones, nos permite indicar algunos parámetros que pueden ser de ayuda en casos de PC poco compatibles y que den problemas al instalar.

Tras pulsar Intro para la instalación gráfica por defecto, en la instalación de Fedora nos permite comprobar la integridad de los CD, esto es, que no están rallados o mal grabados, lo que nos asegurara un contratiempo menos durante la instalación.

Una vez terminado este paso, Arranca el Instalador gráfico.

Opciones de Instalación

Se nos pregunta el Idioma en que se va a realizar la instalación (independiente del que luego utilice el usuario), la configuración del teclado y del ratón, para asegurar que no hay problemas en los siguientes pasos.

En caso de no detectar el Monitor, se nos pedirá que indiquemos cual es en una lista.

Tras esto se nos pregunta El tipo de Instalación que deseamos, en caso de querer instalar ninguno de los roles predefinidos, la opción personalizada nos permitirá mas adelante decidir uno por uno que programas queremos que lleve el S.O.

La siguiente pregunta es si deseamos que el ordenador Cree las particiones automáticamente, o lo queremos hacer a mano.

En el caso de hacerlo a mano aparecerá Disk-Druid para que las realicemos.

Es necesario al menos crear una partición de swap, idealmente del doble del tamaño de la memoria ram, y al menos 64Mb, para equipos con mucha ram, 256Mb o mas, esto es opcional, aunque recomendable para mejorar su funcionamiento.

Tiene que haber al menos una partición de root, montada en / con espacio suficiente para el sistema.



Pantalla de Presentación y DiskDruid

Se configurara el bootloader, permitiéndonos arrancar Linux u otro S.O. que estuviere en el ordenador.

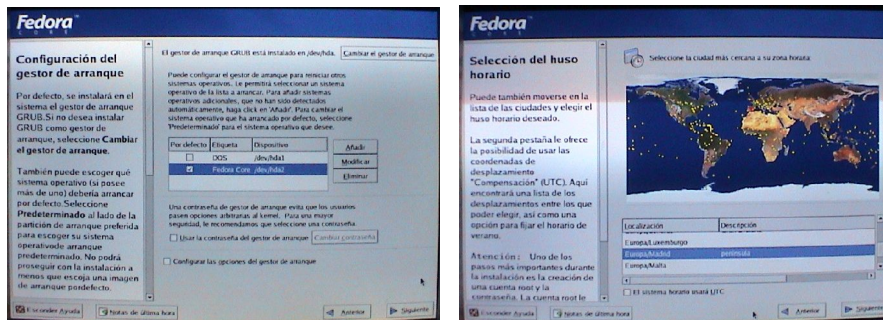
Se nos pedirá información sobre los idiomas que estarán disponibles en el ordenador.

Se configurara la zona horaria.

Se pedirá información sobre las tarjetas de red, dirección IP y otros datos.

Se nos dará la opción de Activar un Firewall en el equipo para aumentar la seguridad.

Se nos pedirá una contraseña para el Administrador del equipo (el root)



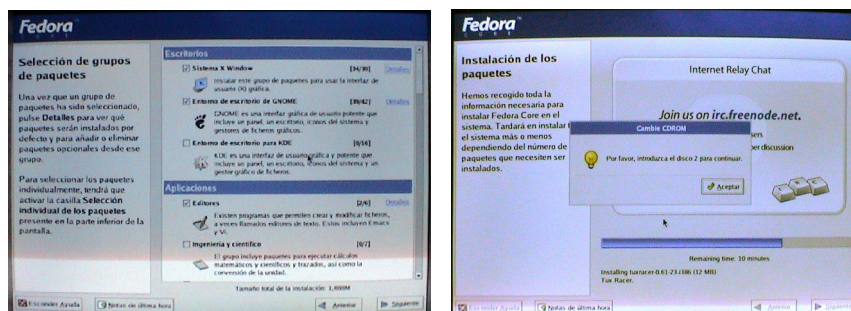
Configuración del BootLoader y de la Zona Horaria

Instalación de los paquetes de software

En el caso de que eligiésemos la instalación personalizada, ahora nos aparecería el listado de paquetes disponibles, deberemos elegir los que deseemos, así como sus dependencias, recordar que mas tarde siempre se puede volver a instalar software que en un primer momento no instalamos, así que si no sabemos que instalar podemos no instalarlo y hacerlo mas adelante cuando lo necesitemos sin ningún problema o merma de rendimiento o calidad.

Se procederá a formatear las particiones, y montarlas según la información que se proporciono antes.

Tras esto el programa de instalación procederá a instalar los programas, pidiéndonos los CD en cada caso y configurando las partes que sean necesarias.



Selección del Software a Instalar y Petición del Segundo CD

Una vez terminada la instalación de los programas nos ofrecerá la opción de crear un disco de arranque para posibles emergencias y tras eso nos mandara reiniciar el Ordenador.

Primer Inicio

Durante el primer inicio se configuraran aquellas partes del S.O. que no pudieron o no debían ser configuradas durante la instalación.

Ahora aparecerá el acuerdo de Licencia, que debemos aceptar si estamos de acuerdo.

También se configurara la Hora del sistema.

Además se nos permite crear los usuarios (al menos uno, recordemos la mala idea que es usar root siempre)

Se configurara hardware menos importante para el funcionamiento del Ordenador, como p.e. la tarjeta de sonido.

Y por ultimo se nos preguntara si tenemos CD adicionales con software, para que el sistema sepa que están disponibles a la hora de listar los programas que se pueden instalar.

Tras esto quedaría preparado para el uso, no siendo necesarios mas reinicios, excepto en el caso de Compilar un Kernel nuevo.

Manejo y Configuración

Usuarios

Por ser multiusuario, varios usuarios, personas (o una misma persona en varios sitios) puede estar usando el ordenador a la vez, bien por varios terminales, en el caso de que estén conectados al ordenador (por el puerto serie, por ejemplo) a través de telnet, o mediante las consolas virtuales (a estas se accede de manera general pulsando **ALT+Fx** siendo x un numero generalmente entre el 1 y el 6, aunque esto varia y se puede configurar a tu gusto, como todo en Linux.

Si estamos en una sesión X, no funcionara, tendremos que usar **CTRL+ALT+Fx**, que nadie se espere encontrar otra sesión X vacía, aparecerá otra sesión en modo texto, si queremos tener dos pantallas X a la vez, deberemos lanzar la 2º a mano desde la sesión de modo texto (es fácil, pero consume mucha memoria) o utilizar los escritorios virtuales, que es bastante mas practico.

Al principio te pide un login, esto es el nombre por el que se te conoce dentro de la maquina, después de eso una clave, te debe haber proporcionado tu administrador, y al menos la de root deberias haberla definido durante la instalacion, si la has hecho tu, si no te gusta después de haber entrado, esto es introducido los dos valores correctamente (el primero aparecerá en pantalla, pero el segundo no, es una medida de seguridad, para que no puedan ver lo que tecleas). Puedes cambiarlo, basta con escribir **passwd**, y te pedirá el antiguo (para comprobar) y que escribas es nuevo dos veces (para verificar que esta bien escrito).

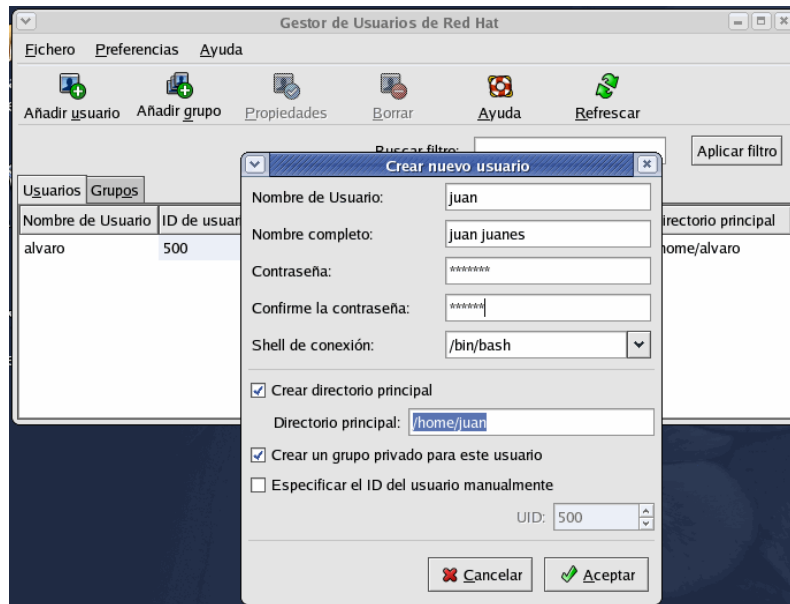
Ahora estas en el prompt del sistema, veras algunas cosas y al final el signo **\$** si entraste como usuario normal y **#** si entraste como root, si has entrado como root, créate un nuevo usuario, con **adduser** nombre, te hará un par de preguntas, nombre, clave, grupo al que pertenece, dirección, teléfono, directorio del home, introduce el nombre que quieras para el login, la clave, y el resto admite los que te propone, por ahora vale con lo que el ordenador da, luego cuando estés mas experimentado, ya comprenderás para que sirven y como modificarlos.

Ahora lo interesante es salir, si quieres acabar la sesión, por ejemplo, para entrar como otro usuario, (muy habitual si eres root, el root, solo debería ser utilizado en casos de necesidad, por ejemplo cuando quieres instalar algo, o quieres cambiar algo de la configuración, en cualquier otro caso es recomendable ser un usuario normal, por seguridad, y por comodidad, aunque al principio te parezca poco cómodo el estar cambiando cada 2x3 de usuario, cuando tengas el ordenador a tu gusto, no necesitaras entrar como root para nada.

Basta con escribir **logout** o **exit**, mas o menos son lo mismo.

Como root ademas puedes cambiar las contraseñas de otros usuarios, incluso sin falta de recordar la antigua, con **passwd** usuario, muy util en casos de olvidar la contraseña. p.e.

Hay ocasiones en que necesitas hacer algo como otro usuario, la orden su usuario nos preguntara la contraseña de usuario, y si es correcta nos transformaremos en ese usuario hasta que salgamos de la session, momento en que volveríamos a ser el usuario anterior, esta opcion puede estar desactivada por motivos de seguridad en algunos ordenadores, así que quizza necesites consultar al administrador del sistema.



La Administración de Usuarios también se puede hacer de manera gráfica

Shell y Comandos

Bash

Una vez entrado en una sesión en Linux, si esta es en modo texto, lo primero que nos encontramos es el prompt del sistema, y esto es debido a que se está ejecutando un Shell, y este espera órdenes, este Shell, es el programa que interactúa con nosotros, y transforma nuestras órdenes en acciones para el S.O., en la mayoría de los casos suele ser BASH, Bourne Again Shell, una versión mejorada del shell creado en la universidad de Bourne y que se considera un estándar dentro de Unix.

Si en este shell escribimos una palabra, el shell comprobará primero si es un comando interno, y lo ejecutará, o después buscará, a través de una serie de directorios predefinidos un archivo con ese nombre, y si es ejecutable lo ejecutará, pasando el control de la sesión hasta que el programa acabe y volvamos al shell en espera de otra orden.

Hay que destacar que no sirve que un programa esté en el directorio en el que estamos, ya que por seguridad, el shell no busca el programa en el directorio actual, solo en una lista de sitios (directorios) donde sabe que hay programas seguros, evitando así una de las formas de entrada de virus.

Para saber por qué sitios busca podemos escribir `set | grep PATH` que nos mostrará la lista separada por “.”.

Opciones de Comandos

Normalmente todos los programas permiten opciones, por ejemplo `ls` permitía `-a -p -l`, etc, las opciones se suelen colocar justo detrás del nombre del programa, antes que los otros parámetros, por ejemplo en el caso de `ls` sería `ls -l /usr/doc`, aunque suele funcionar si lo colocas en otro sitio, lo normal es ahí, por facilidad a la hora de leer el comando y pensar lo que hace.

Cuando hay varias opciones, lo normal es que se puedan colocar juntas, así podríamos poner `ls -pla` para usar las tres opciones, en ciertos casos especiales se podrá además omitir el guión, aunque lo normal es ponerlo.

También en algunos casos es posible utilizar opciones largas, así es lo mismo usar en `ls -a` que `--all`, en estos casos se utiliza un doble guión, para diferenciar `--all` de `-a -l (-all)`

Moviendose por tu Linux

Según entras en una sesión de Linux, estas en tu home, también llamado `~`, es un sitio del disco duro, reservado para ti, donde vas a poder hacer lo que quieras, fuera de este sitio no vas a poder tocar nada, pero no tocar, normalmente suele ser `/home/nombredelogin`, a veces aparece delante del prompt, si no teclea `pwd` y te dirá donde estas.

Si eres root, lo normal es estar el `/root` y puedes ver y tocar todo, absolutamente todo. Por eso no se suele recomendar entrar como root.

El disco duro esta organizado en directorios o carpetas, unas dentro de otras, por ejemplo esta la principal, el raíz `/` de hay cuelga varias, `home`, `usr`, `var`, `lib`, tus estas dentro de `home`, de están cuelgan a su vez otras, tu estarás en una en concreto, yo entro con login pepe, y estoy ahora en `/home/pepe`, como una esta dentro de otra, las puedes escribir juntas separándolas por el signo `/`, la carpeta en la que ahora estas sea cual sea la puedes llamar como. y la carpeta de la que cuelga en la que estas el., cualquier carpeta puede tener mas carpetas, o ficheros, los ficheros son como hojas de papel dentro de las carpetas, puede ser una carta, una foto o un programa, eso da lo mismo. Cada una tiene un nombre que la identifica, y que es diferente al resto de esa misma carpeta, aunque puede ser igual al de otro fichero o carpeta que este en otro sitio diferente.

Para ver lo que hay en la carpeta en la que estas escribe `ls -p`, (el `-p` puede que no sea necesario, según que Linux/Unix utilices) veras que aparecen todos los archivos y las carpetas que cuelgan de él aparecerán con el signo `/` detrás (`-p` hace que aparezca este signo, sin el no habría diferencia entre carpetas y ficheros) también es posible que aparezcan de colores, o con otros signos (`*` y `@` por lo menos son comunes) también puedes ver el contenido de otra carpeta, si escribes su ruta detraes, ej; `ls -p /` , `ls -p /usr/doc` , `ls -p /home/pepe`.

Si lo que quieres es moverte por las carpetas utiliza `cd` seguido de donde quieres ir, así `cd /` te llevara a la raíz, `cd ~` a tu home (`cd` a secas hace lo mismo, útil si te pierdes por hay) `cd /usr/doc` te lleva al directorio de documentación del sistema.

Investiga un poco por tu ordenador con `ls` y `cd`, para ver lo que hay.

Si quieres crear una carpeta (o varias) lo puedes hacer en tu home, usando `mkdir` y el nombre de la carpeta que quieres crear, luego entra en ella con `cd`.

Para borrar una carpeta escribe `rmdir nombre`, pero la carpeta tiene que estar vacía y tienes que tener permiso para borrarla (eso por ahora no importa que es, tienes permiso para borrar todas las que consigas crear, con eso vale).

Conseguir información

En Linux, lo normal es que tenga información de todos los comandos que puedes utilizar, esta se encuentra principalmente en cuatro sitios:

Las paginas `man`, a estas se acede escribiendo `man capítulo nombre_de_comando` donde capítulo es el capítulo del manual a consultar (déjalo en blanco, y el ordenador suele acertar solo, `capitulo` en un numero, el 1 corresponde a los comandos que los usuarios normales pueden utilizar, 2 es la referencia del programador, 6 son los juegos, 8 los comandos que solo root puede usar, etc., hay varios).

Si simplemente te interesa saber que hace un comando, bastante mas útil que las paginas del `man` será la orden `apropos nombre_del_programa` que te dice que hace el programa, o te informa de programas similares.

Si no sabes que comandos existen pulsa `TAB`, `TAB` te da todas las posibles opciones que tienes a la hora de escribir comandos (quizá tengas que pulsarlo dos veces, y contestar `s` o `y` a la pregunta de si quieres verlos todos) además te permite completar comandos, así para escribir `shutdown`, basta con escribir `shut` y `TAB` para que el ordenador lo complete (si quieres ver todos los comandos que empiezan por `a` pulsa la `a` y luego `tab`).

También te puede ser de utilidad saber que todas las paginas del `man` se encuentran almacenadas en tres o cuatro directorios, `/usr/man`, `/usr/local/man` y `/usr/local/man`, dentro de este encontraras una serie de carpetas, con los idiomas disponibles, ej es para el español, de para el alemán, y carpetas con `man1`, `man2`, etc, que son los capítulos, dentro de estas estarán los ficheros de ayuda, suelen estar comprimidos,

pero te vale para sacar el nombre, así la ayuda del less será algo así como less.gz less.x.gz o less.l.gz o algo similar.

Para salir del man suele bastar con pulsar q e intro.

Otra ayuda muy completa suelen ser las páginas info, para ello teclea info, y accedes a un programa donde veras una lista de todas las opciones, info es un poco suyo a la hora de usarlo, pero esta hay y es útil por que para algunos programas no hay pagina man, solo info.

Además sueles tener mas información y/o ayuda en la documentación, que siempre se encuentra en /usr/share/doc/nombredeprograma. para verla, entra en el directorio que te interese y utilizando cat, more o less (son tres programas que te permitirán ver ficheros, al cual más cómodo, en ese orden) seguido del nombre del fichero, si el fichero acaba el gz lo usual es que este comprimido, usa zless para verlo, si acabase en .html, .tex, .dvi, .ps, .pdf, no podrás verlo necesitas un programa especia,

Además es posible que te interese crear tus propios documentos, podrás crear los que quieras en tu HOME usando vi, el editor que normalmente se usa en Linux, es bastante cómodo cuando le coges el tranquillo, para empezar:

ESC y luego o para empezar a escribir

ESC y luego: wq para guardar los cambios y salir

ESC y luego: q! para salir sin guardar los cambios

ESC y luego: help para una completa información sobre como usarlo. Al final de los apuntes hay un apéndice mucho mas completo sobre vi.

Si tienes suerte y tienes Gnome o Kde instalados, tendrás un bonito programa que te permitirá ver la anterior documentación de una manera muy cómoda, además, si estas en X deberías probar cosas como xman, tkman, tkinfo, que son versiones gráficas de los anteriores programas.

Comodines

Como shell (interprete de comandos, el programa que intenta comprender lo que tu dices y lo ejecuta) normalmente suele ser el bash, este permite usar comodines (como las cartas) a la hora de escribir un nombre de fichero (además la tecla tab te permite completar rutas a la hora de escribir, escribe /ho y pulsa tab veras como aparece /home/)

Los principales son *,? y los corchetes [], así el asterisco sustituye a cualquier grupo de letras, eje pe* vale para pepe, pepa, película y pe. el ? sólo vale para una letra, así pep? vale para pepe, pepa, pero no para película

Los corchetes valen por un conjunto así pep[aAe] vale para pepa, pepA, pepe, pero no para peps, ni para pepE, en unix son muy importantes las mayúsculas y la minúsculas, hasta el punto de que puedes tener dos ficheros que se llamen igual, por ejemplo pepe y Pepe, ya que uno tiene mayúsculas y otro no.

Además puedes usar [letrainicio-letrafinal], ejemplo [a-d]* vale por cualquier fichero que empiece por a,b,c o d.

Hay que tener en cuenta que los nombre que empiezan por . se supone que están ocultos, (al hacer un ls no salen, necesitas ls -a, pruébalo en tu home, verás como aparecen varios ficheros que no sabias que estaban ahí) y el asterisco no los pilla, es necesario indicar.* para que los coja.

Tuberías

En Unix existen los conceptos de entrada estándar, salida estándar y tuberías que debemos aprender a manejar.

En Unix, se utiliza una filosofía diferente a Windows, en Windows estamos acostumbrados a megaprogramas que hacen de todo, en Linux, lo normal es utilizar pequeños programas específicos que saben hacer una y solo una cosa, estos programas normalmente cogen los datos que necesitan de la entrada estándar, y de la línea de comandos (parámetros y opciones), y dan sus resultados por la salida estándar.

Después, puedes coger la salida de un programa y pasárselo como datos a otro, para realizar tareas más complejas, esto se realiza con las tuberías (el carácter | que se obtiene con ALTGR+1), y mediante shell scripts (pequeños programas como los .bat del ms-dos) si la tarea es complicada, o se va a repetir varias veces.

Por defecto la entrada estándar es el teclado, y la salida estándar es la pantalla, pero puedes hacer que sea un fichero con las teclas a pulsar, para ello usar comando `< fichero`, si el resultado del programa quieres que pase a un fichero, en lugar de a pantalla usar comando `> nombrefichero`, pero ojo que `>` borra el fichero que hubiese, y lo sustituye por el resultado del programa, comando `>> fichero` no lo borra, si no que lo añade al final.

Por último se pueden usar tuberías, hay ocasiones en que lo que nos interesa no es conservar el resultado de un programa, si no pasárselo a otro como datos, como entrada estándar, sin necesidad de usar un fichero temporal como podríamos hacer con comando1 `> file.tmp`; comando2 `< file.tmp`; `rm file.tmp` , para esto utilizamos las tuberías, quedando el conjunto de tres comandos anteriores como comando1 | comando2.

Así por ejemplo `sort` coge lo que haya a la entrada y lo ordena y lo pone a la salida, si quieres ver un `ls` ordenado usar `ls | sort`. si además quieres que el resultado aparezca en un fichero usar `ls | sort > fichero`.

Además puedes usar la salida de un programa como opciones para otro, si pones el comando entre acentos graves `find / -name `ls`` por ejemplo buscaría por todo el disco duro fichero que se llamasen igual que los del directorio actual (datos por `ls`).

Uso de unidades de disco

Hasta ahora hemos estado trabajando desde el disco duro, suponemos, en nuestro directorio home, pero en un ordenador normal, (si trabajamos en una red, y estamos detrás de un terminal, esto no tendría tanto sentido) tenemos unidades de disco, Zips, CD-Rom, etc... en los que tendremos cosas que ver.

En Linux, no existe el concepto de unidad de disco, que existe en otros S.O., en Linux, las unidades se "montan" como directorios, que cuelgan de donde elijamos, así lo normal es que la disquetera este en /floppy o en /mnt.

Linux es un sistema multiusuario, eso significa que puede haber varias personas que usan los dispositivos a la vez, imagínate que follón si estáis dos trabajando en el mismo ordenador, y al 2º se le ocurre sacar el CD de donde tu estas copiando cosas para meter uno suyo, tú aplicación daría error y tu si no estas delante de la unidad no tendrías ni idea de por que.

Para evitar eso, Linux tiene dos "defensas", por un lado, el root, es en principio, el único que tiene permiso para montar y desmontar discos, esto es hacer que estén disponibles para el posible usuario. La otra defensa (que no es útil para los discos) es que hasta que no desmontes la unidad (esto es la hagas no disponible a los usuarios) no podrás sacar el CD, Zip, o lo que sea de la unidad. (Aquí es muy importante destacar que a diferencia de los floppy de Macintosh, en que la disquetera es eléctrica y saca ella los discos, en los PC, es manual, es MUY IMPORTANTE desmontar los disquetes antes de sacarlos, ya que de otra manera se puede perder su contenido, es como apagar el PC si esperar. Y NO VALE VOLVER a meter el disco, si lo sacas, los datos se pueden borrar o corromper.

Para montar se utiliza `mount -t tipo_fs dispositivo lugar_montaje`

`tipo_fs` es el tipo de sistema de ficheros, es la manera en que se almacenan los datos en el disco, normalmente cada sistema operativo tiene el suyo propio, así por ejemplo msdos usa fat, Windows 95 usa vfat, iso9660 para los CD-Rom, os2 usar hpfs, Windows nt usar ntfs, y Linux usa ext2, en la practica bastara con montar los disco Linux con ext2, iso9660 para los CDs y el resto con vfat.

Dispositivo es el nombre del dispositivo a utilizar, en Linux, todos los dispositivos del sistema se encuentran en un directorio especial, en /dev/, así la disquetera suele ser /dev/fd0 o /dev/fd1, el disco duro suele ser /dev/MdXY siendo M la letra h si es IDE y la letra s si es SCSI, X una letra entre a y la d si es IDE (en función de donde este, primario master, primario slave, secundario master, etc...) o un numero del ID SCSI, y por ultimo Y es el numero de partición, en aquellos sistemas que tengan.

Es un poco complicado, por lo general el CD-Rom será /dev/hdb o /dev/hdc.

Por último hay que indicar donde quieres que se monte, es la ruta de un directorio que este vacío, así para montar el típico disco de MS-Dos, puesto en la unidad a:, seria `mount -orw -t fat /dev/fd0 /mnt`

Un cd seria mount -t iso9660 /dev/hdb /mnt

Si te has fijado ambos se montan en /mnt, no es posible montar los dos a la vez, así que antes hay que desmontarlos, o montarlos en sitios diferentes, para desmontarlos umount dispositivo o umount lugar_montaje.

Todo este proceso se puede automatizar, con la configuración adecuada, para que baste con poner mount /dev/cdrom por ejemplo para que monte el cdrom, para ellos hay que modificar /etc/fstab, quedando como ejercicio al lector su realizacion.

Organizacion de directorios

Hasta ahora nos hemos estado hablando de unos pocos directorios del disco, si has investigado, veras que hay muchos directorios, cada uno con un significado especial, veamos para que valen:

/

El directorio raíz, de este cuelgan todos los demás.

/sbin

Contiene programas de administración indispensables

/bin

Contiene los programas básicos para el uso normal

/root

El home del administrador del ordenador

/tmp

Es un directorio que todo el mundo puede usar como temporal, normalmente se borra su contenido cada cierto tiempo de manera automática

/lib

Contiene las librerías necesarias para los programas

/etc

Contiene toda la configuración del ordenador, es similar al registro de Windows, cualquier programa que este instalado tendrá un archivo en modo texto (esto es, que lo podemos editar con un editor de textos común como puede ser vi) en algún lugar de este directorio.

/proc

Contiene información del funcionamiento del ordenador, prueba a entrar en el y ver con cat el contenido de los ficheros que tiene.

/lost+found

Contiene los ficheros que se encuentran después de realizar una comprobación del disco, suelen aparecer después de fallos del PC.

/dev

Contiene los dispositivos que tiene el ordenador disponibles.

/var

Contiene datos sobre programas que son comunes a todos los usuarios, records de los juegos, la cola de impresión, los buzones de correo y los log (registros de sucesos en programas) de los demonios (programas que siempre están funcionando aunque no lo veamos ni los notemos).

/mnt /floppy /cdrom /dos

Son directorios donde se suelen montar los sistemas de almacenamiento, depende de cada distribución o administrador.

/home

De aquí cuelgan todos los home de los usuarios del sistema.

/usr

Dentro de este directorio hay varios directorios no imprescindibles

/usr/sbin

Programas de administración no indispensables

/usr/bin

Programas de usuarios no básicos

/usr/games

Juegos y programas que pueden comprometer la seguridad, el superusuario no suele poder usarlos.

/usr/lib

Librerías de los programas de /usr/xxx

/usr/man

Páginas de los manuales

/usr/doc

Documentación de todos los programas instalados

/usr/src

Fuentes de programas para compilar (normalmente el kernel)

/usr/X11R6

Es el sistema de ventanas X Windows system, dentro de él encontrarás man, bin, lib, etc...

/usr/local

Este directorio es como /usr (tiene su bin, su lib, su man, pero se refiere a programas o datos propios de ese ordenador, es decir aquellos que no serán visibles en otros ordenadores conectados en red (es posible compartir el mismo disco duro entre varios ordenadores, de esta manera por ejemplo, da igual en cual te pusieses, siempre tendrías tu home y tus aplicaciones.

Tambien es donde se instalan las cosas que se compilan en el ordenador, sin utilizar un sistema de paquetes y se quiere que esten disponibles para los usuarios.

Otros comandos Interesantes

Supongo que tendrás un montón de fichero basura de tus últimos experimentos `rm fichero(s)` elimina el/los fichero(s) que le indiques, y MUCHO OJO, por que no se pueden recuperar de ninguna manera. además `rm -rf directorio` borra el directorio y todo lo que hubiese dentro (y tampoco se puede recuperar).

Sobre el manejo de ficheros pocas cosas quedan por ver, únicamente destacaría el comando para mover, `mv`, este permite dos operaciones, por un lado nos permite cambiar un fichero de sitio, para esto es necesario que tengamos permiso de escritura sobre ese fichero y el lugar de destino, simplemente indicando `mv nombre_origen ruta_destino`, cabe destacar que es posible no solo mover fichero, si no también directorios, o todos los ficheros de un directorio utilizando comodines.

La otra operación que se puede realizar con `mv` es el cambio de nombre, así basta con indicar el nombre de un fichero o directorio, y luego su nuevo nombre.

Otra cosa interesante es saber cuanto espacio estamos ocupando en un determinado directorio, basta con escribir `du`.

Para saber el espacio que hay libre en el disco, utilizaremos `df`.

Si queremos borrar la pantalla utilizaremos `clear`.

Con esto estarían vistos los comandos básicos para desenvolverse por el sistema operativo.

Procesos y Niveles de Inicio

Control de Procesos

Con el comando `ps` o `top` que es interactivo se pueden ver los procesos (programas) que están funcionando en ese momento en el ordenador.

Algunos de estos programas están funcionando de manera transparente desde el inicio del S.O. y se les denomina `daemons`, demonios, y son los encargados de que funcionen los distintos servicios que da el S.O.

Entre otra información muestran el PID, un número único en ese momento para cada proceso, que nos permitirá más adelante modificar el proceso.

Estos procesos se pueden controlar de varias maneras, pues es posible cambiarles la prioridad, para hacer que estén funcionando pase lo que pase, o solo en momentos de poco uso del ordenador, así como enviarlos a segundo plano, para poder seguir trabajando en el shell mientras se ejecutan, o terminarlos antes de tiempo:

Si cuando lanzamos un programa añadimos `&` al final, este se ejecutará en segundo plano, devolviéndonos inmediatamente al prompt.

Un programa normal puede ser suspendido pulsando `ctrl.-Z` (recordemos el `ZZZ` de dormir) o cancelado con `ctrl.-Z`.

Para continuar la ejecución de un programa suspendido escribiremos `fg`, o `bg` si queremos que se siga ejecutando en segundo plano.

`Jobs` nos muestra que programas hemos ejecutado en esa sesión, así como cuáles están parados y cuáles funcionando en segundo plano, muestra además un número después del signo `%`, que es relativo y puede ser utilizado en lugar del PID, p.e. `bg %2` llevaría a segundo plano el penúltimo (el último sería `%1`) programa que hallamos ejecutado (y que aun no halla terminado).

`Nice` prioridad programa, o `renice` prioridad PID, permiten lanzar un programa con una prioridad diferente, o cambiar la de uno que ya se está ejecutando, desde 20, que es la mínima prioridad, 0 la prioridad normal o -19 máxima prioridad (solo `root` puede poner prioridades negativas).

`Kill` PID, termina el programa que tenga ese PID, `killall` programa, termina todos los programas que se ejecuten con ese nombre, es posible terminar el programa de diversas maneras, con `Kill -9` o `Kill -15`, en función del tipo de terminación que queramos.

Servicios

Desde que se arranca el ordenador hasta que aparece `login`: se ejecutan una serie de programas, que se encargan de configurar el ordenador, y lanzar los diferentes demonios o servicios que estarán funcionando todo el tiempo.

Para hacer más versátil al ordenador, se le dotó de la posibilidad de definir niveles (quizás fuese mejor llamarlos roles, pues no todos son tienen un orden con respecto a los demás) en función de los servicios necesarios.

Este sistema está basado en el Unix System V, existe otra opción basada en el Unix BSD, pero no es tan utilizada.

Así el nivel 0 se ejecuta al arrancar, el 6 al reiniciar y 1 es el modo monousuario, para administración.

Los niveles 2 al 5 están disponibles para que el usuario defina hasta 4 roles de funcionamiento.

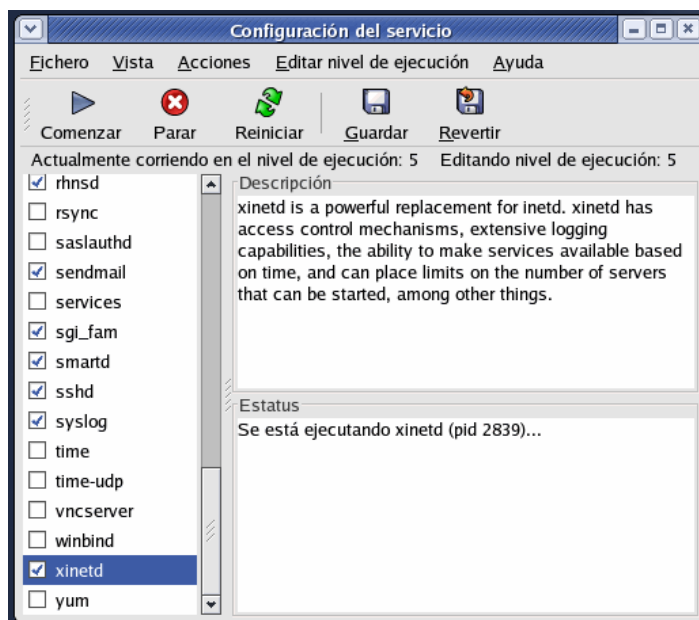
Así p.e. el 5 suele ser el único en que se ejecuta el servidor gráfico X, aunque todo esto es configurable.

Existe un directorio `/etc/init.d` donde están `shellscripts` de cada uno de los servicios disponibles.

Existen además una serie de directorios `/etc/rc.N` donde N es el `runlevel`, que contienen enlaces simbólicos a los `shellscripts` de `/etc/init.d`.

El nombre del enlace tiene una forma especial: XNNshellscrip, donde X puede ser una S (Star) o un K (kill) y controlara que ese servicio se inicie o se termine al entrar en ese runlevel, después NN es la prioridad de ese servicio, servicios con la prioridad mas baja se iniciaran antes que aquellos con la prioridad mas alta, de esta manera es posible p.e. controlar que el servidor web (S80apache por ejemplo) no se inicie hasta que se haya iniciado la red (S30 network por ejemplo).

Además de administrar los servicios a través de los nombre y los enlaces, están disponibles herramientas graficas una administración mas cómoda.



Herramienta de Configuración de Servicios, aka Runlevels

Inetd

Existe un tercer tipo de servicios, normalmente de red, que no se ejecutan como demonios, si no bajo demanda, estos servicios se controlan desde un archivo de texto en `/etc/inetd.conf`.

El que un servicio, p.e. un servidor de paginas web, se ejecute como demonio o bajo demanda, dependerá del uso, si va ha ser intensivo, o solo se utilizara de vez en cuando. Además los demonios consumen mas memoria del sistema, pero los servicios de inetd tardan mas tiempo en empezar a funcionar.

Administración del equipo

Permisos

Linux/Unix son sistemas multiusuario, donde mucha gente puede estar usando el mismo disco duro, así que debe haber un sistema que sea capaz de poner un poco de orden a todo esto y decidir de quien es cada cosa, y quien puede usarla.

Todo en Linux tiene un dueño y un grupo, el dueño es generalmente el que lo creo, aunque se puede cambiar, y el grupo es un grupo de usuarios que tiene privilegios especiales sobre el resto del mundo, imagina que una empresa usa un ordenador, y dentro de esa empresa, hay departamentos de marketing, investigación y dirección, es posible crear grupos de usuarios que pertenezcan a uno o mas de esos grupos, y tener ficheros que un grupo vea y otro no (por ejemplo para evitar que los de marketing cotilleen lo que hacen los de investigación.

El dueño, adema puede tener otro tipo de privilegios sobre el grupo, e.j. puede además de verlo, modificarlo.

Para ver los permisos y dueños de un fichero utiliza `ls -l`, veras que aparecen columnas, la primera es una letra (que indica el tipo de archivo que es) y un grupo de guiones, `x,w`, y `r`, luego un numero, y luego dos nombres o números, el primero, el dueño del fichero y el segundo el grupo, luego aparece tamaño y fecha de creación y por ultimo el nombre del fichero.

Para cambiar el dueño `chown` nombre fichero(s), para cambiar el grupo `chgrp` nombre fichero(s), o todo de una vez, `chown nombre.grupo fichero(s)`.

La primera columna de `ls -l` indica los privilegios o permisos de cada tipo de usuario, además del tipo de fichero.

La primera letra será una `d` si es un directorio, un guión si es un fichero normal y otro signo si es un fichero especial.

Luego hay tres conjuntos de letras `xrw`, o en su lugar guiones, indican los permisos del dueño, del grupo y del resto de usuarios.

La `x` indica que es un fichero ejecutable, un programa, y según donde aparezca lo podrán ejecutar el dueño, grupo y/o todos los usuarios.

La `r` indica quien puede leer ese fichero.

La `w` indica quien puede escribir (o modificar o borrar) ese fichero.

Así puedes hacer que un documento solo lo puedas ver tu dándole permisos `-rw-----`, o que lo pueda ver y modificar la gente de tu grupo, y el resto del mundo solo leerlo, `-rw-rw-r-`.

Cambiar los permisos es un poco complicado, hay que usar `chmod` permisos nombredeficheros

Permisos puede ser `gente+accion+permiso`, siendo `gente` `u` - usuario, `g` - grupo, o - los otros, la acción un guión para quitar el permiso y el signo `+` para darlo y luego `x,r` y/o `w` según el permiso, así:

`chmod go+rw fichero` hace que el grupo y todo el mundo puede leer y escribir el fichero.

`chmod o-rwx fichero` hace que los otros no pueden ni leer ni escribir ni ejecutar el fichero.

Además si sabes un poco de binario y octal puedes sustituir eso por un numero, así `xrwxr-xr` (tu puede leer/modificar y ejecutar, el resto solo leer y ejecutar) seria `chmod 755 fichero`, permisos `-rw-r----` seria `640`, sustituye cada letra por un uno binario, y cada guión por un cero binario, y luego pásalo a octal, con la practica es muchísimo mas rápido de usar que la opción anterior.

Apagar el sistema

El ordenador no se puede apagar por las buenas, como harías con la tele, debes antes decirle que lo vas a apagar, para que el ordenador deje las cosas en orden, (el ordenador no guarda todos los datos en disco duro, los guarda en memoria, y cuando lo cree conveniente los pasa al disco duro, así esa carta, que acabas de hacer, puede que aun no este en el disco duro, si no en memoria, debes decirle que pase todo lo que tiene pendiente al ordenador antes de apagar, esto se conoce como cache, además ciertos procesos (otros programas que están funcionando y que no ves, he aquí la multitarea) tienen que cerrarse civilizadamente.

Apagar es algo que solo puede hacer el root, o habitualmente el que esta situado en la consola (consola es el conjunto de monitor-teclado principal, es decir el que esta conectado directamente al ordenador y no a un puerto serie, o por telnet)

Si estas situado en la consola, pulsa `ctrl-alt-del`, esto debería iniciar el proceso de apagado o reinicio de la maquina, según lo configures, al final, deberías ver un mensaje que pone `SYSTEM HALTED`, o el típico mensaje de la bios al arrancar, ahora es cuando debes apagar el equipo.

Suponiendo que seas root, para apagar el equipo teclea `shutdown -fh now`, no es necesario que estés en la consola

La `f` significa que el próximo arranque sea rápido, es decir que no compruebe los discos al iniciarse.

La `h` significa que se pare, (al final veras el mensaje `system halted`) si hubieses puesto una `r` se reiniciaría, y verías el mensaje de la bios.

`Now`, significa que reinicies en ese instante, si pones una hora, el reinicio se realizara a esa hora, si pones `+n`, el reinicio se hará en `n` minutos, esto es útil, por que se envía un mensaje a todos los usuarios que hay

indicándoles que en n (o a la hora x) se reinicia el ordenador, y de paso impide nuevas entradas de usuarios.

Si hubiese un proceso de reinicio en curso (por ejemplo has dicho que reinicie dentro de 5 minutos) puedes cancelarlo con shutdown -c.

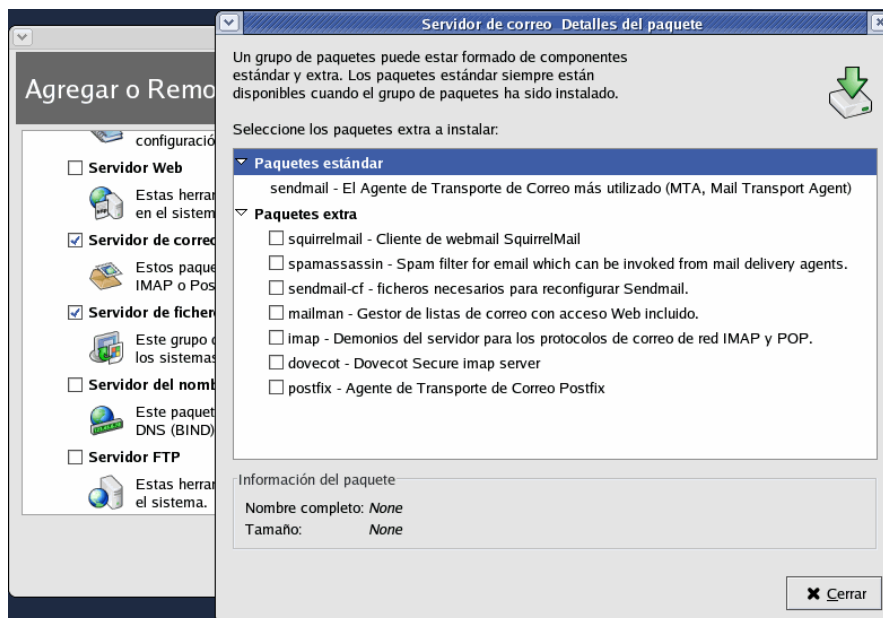
Instalacion de Software

En Linux los programas y las librerías vienen en paquetes, en el caso de redhat con la extensión .RPM, en estos paquetes ademas de los programas y archivos necesarios, viene información de donde hay que instalarlo (que es un sitio fijo), si necesita algun otro paquete para funcionar y si dara problemas con otros ya instalados.

De esta manera es muy sencillo controlar que programas y donde están instalados evitando tener programas duplicados, obsoletos, que provoquen errores o que no funcionen, y garantizando que la desinstalación es limpia y completa.

RedHat incorpora la herramienta YUM que permite instalar programas de una manera cómoda, si es necesario descargara de Internet los paquetes que no estén disponibles.

Para configurar YUM hay que utilizar el archivo /etc/yum.conf donde de manera similar a los archivos .ini de Windows se configura de que sitios se pueden conseguir los paquetes.



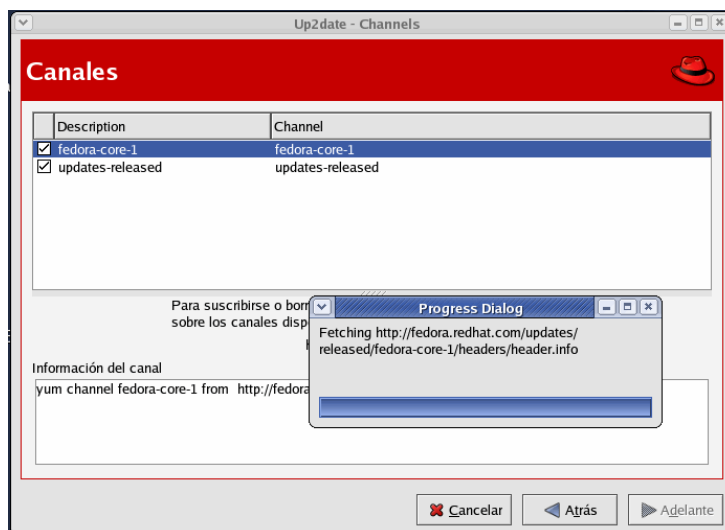
Programa de Instalacion de Software Mostrando un Grupo de Programas

Una vez echo esto, instalar algo es tan sencillo como escribir `# yum install programa`.

Con el programa rpm se puede controlar de manera independiente que paquetes hay instalados y instalar o desinstalar en función de las necesidades.

Ademas se ofrecen tres metodos para instalar software:

- Si durante el arranque se hace con el CD de Instalacion el programa nos permite actualizar o instalar nuevo software sin modificar lo ya instalado.
- Si se introduce el CD de instalacion en cualquier momento, y se ejecuta el autorun, aparecera el programa de instalacion de Software donde podremos elegir que programas deseamos instalar, este programa tambien esta accesible desde el menú de aplicaciones
- El programa up2date mantiene el software actualizado avisandonos y permitiendonos instalar nuevas versiones en cuanto estan disponibles.



Up2date comprobando la existencia de actualizaciones

Entorno Grafico

El sistema X Windows

Linux es un sistema en modo texto, como puede serlo el MS-Dos, además, incluye una extensión, el sistema X Windows versión 11, conocido como las X, o como X11.

Este sistema se carga encima del de modo texto, pero a diferencia de lo que sucede con MS-Dos + Windows 3.1, no disminuye su rendimiento, al ser ya de por sí Linux un sistema multitarea de 32 bits, en cierto modo es como Windows 95.

Por lo anterior dicho, es conveniente empezar a aprender a manejar Linux desde modo texto.

Además hay que tener en cuenta que el sistema de ventanas de Linux, se basa en una arquitectura Cliente-Servidor, mas o menos esto significa que el programa no tiene por que ejecutarse en el mismo ordenador en que lo estas viendo, y que un mismo ordenador puede ejecutar varios programas o sesiones X para uno o mas pantallas,

En la práctica esto simplemente significa que si tuvieses que proporcionar ordenadores a 10 personas que en un determinado momento necesitasen mucha potencia, podrías darles 10 ordenadores muy potentes, o 10 ordenadores normales y un gran ordenador que todos compartirán cuando necesiten potencia (que suele ser lo mas barato, quizás no con 10 pero sí con 100).

El sistema X Windows, en un principio se creo para solucionar la necesidad de ciertas personas de ver gráficos de sus aplicaciones, o de tener mas de una pantalla a la vez, estas personas normalmente eran científicos o programadores, así que las X se limitan a dar un trozo de pantalla donde puedes dibujar o leer.

Más tarde la gente descubrió que lo del ratón era intuitivo y que en lugar de teclear largas líneas de comandos era más fácil hacer clic con el ratón apuntando sobre un cuadrado con unas letras (botón).

Además como les quedaba muy soso lo de los tres o cuatro trozos en la pantalla decidieron añadir un marco a esos trozos llamados ventanas.

Surgen así dos temas, bastante controvertidos en Linux.

- Los Widget, o ventartilugios, como alguien llamo (los botones y demás)
- Los Windows Manager, los administradores de ventanas (lo bordes)

Los Windows Manager son el programa que se encarga de dar marco a las ventanas, como cada uno hace lo que quiere, hay cientos, desde cutres como el TWM, eficaces como el Fvwm 2. A vistosos y barrocos como el Enlightenment, además suelen proporcionar unos menús y alguna aplicación para hacer un poco menos inhóspitas las X, (principalmente menús, o una barra de herramientas para ejecutar aplicaciones).

Los widget son los que dan aspecto a los programas, es lo que diferencia un botón de Mac de uno de Win95 y de uno de SCO, aquí ya no depende de lo que tu elijas, si no de lo que elija el programador, aunque se puede cambiar a través de temas, algunos de los mas conocidos son el polémico QT, Motif, Gtk, Athena, Xform. Como cada programador hace lo que quiere, suele pasar que tengas varias aplicaciones y no se parezcan nada entre ellas.

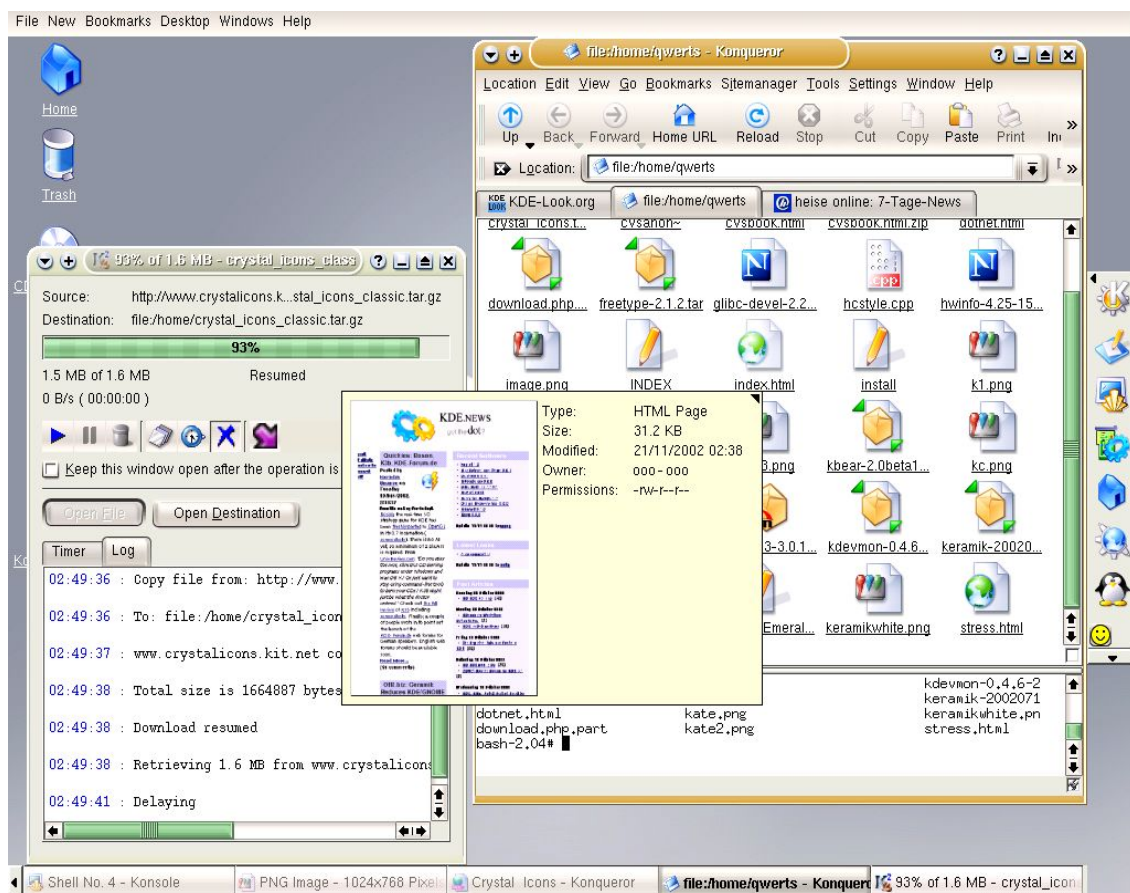
Últimamente ha surgido una serie de movimientos que intentan cambiar esto, así aparecen los gestores de escritorio, que básicamente son:

Un gestor de ventanas, una librería de widgets, un montón importante de aplicaciones tontas (el buscaminas, el reloj, la agenda), y algunas más importantes, (Procesador de Textos, Hoja de Calculo, Base de Datos, Navegador de Internet, Programa de Presentaciones) y algunas utilidades para configurar todo esto a base de ventanas.

Su interés es conseguir unas X atractivas y fáciles de manejar, algunos como OpenStep, un entorno que imita NextStep, o xfce que intenta hacer un gnome más liviano, aunque hay dos que destacan sobre los demás en cuanto a porcentaje de utilización:

Kde

Ya en su versión 3.1 junto con Koffice, su principal achaque es que el widget esta desarrollado por una empresa privada, y puede cortar su futuro desarrollo, aunque esta posibilidad es mínima gracias a que QT ya es software libre, el aspecto es soberbio. Esta programado en su mayoría en C++



Entre los programas que destacan de KDE están:

Quanta: Editor de paginas Web, con opción wysiwyg, similar a dreamweber

Qcad: Programa de CAD

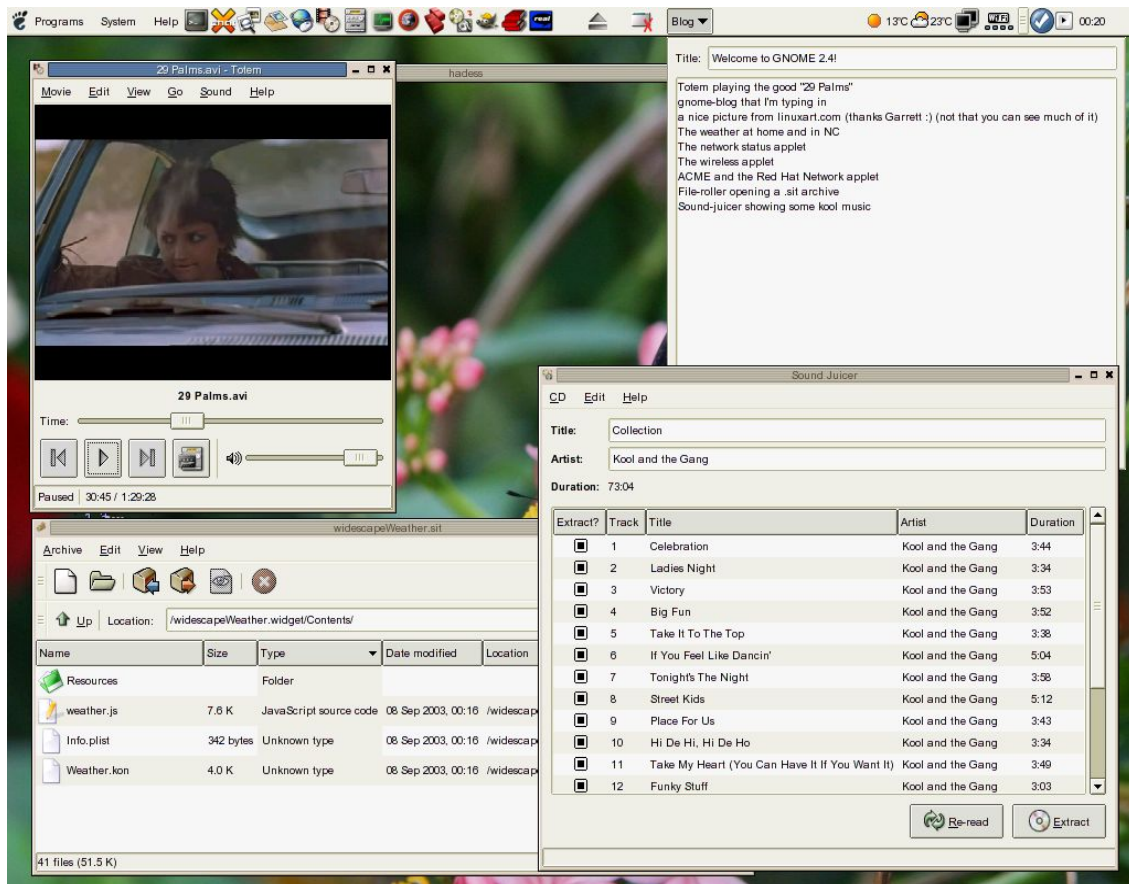
Scribus: Programa de autoedición similar al Quark Express

K3B: Programa de grabación de CD similar al Nero

Gnome

Esta basado en gtk, un widget completamente libre, y con unas vistas un poco mas altas (gnome no solo da las aplicaciones con un aspecto único, intenta dar toda una serie de objetos para el desarrollo de

aplicaciones, programables en una multitud de lenguajes) Ya va por la versión 2.0, y también tiene un conjunto ofimático tipo Office, aunque menos integrado que este y con la falta de algunos programas.



Interesa destacar que OpenOffice.org va a sacar próximamente una versión nativa de su suite ofimática para Gnome, además de la propia que gnome desarrolla.

Uno de sus productos estrella es Evolution, un clon del Microsoft Outlook, del que es compatible y supera en prestaciones.

Gimp, un programa a la altura de photoshop para el retoque grafico.

GnuCash es un excelente programa de contabilidad tanto a nivel empresarial como del hogar.

Administración Avanzada

Recompilar el Kernel

En Linux el 99% de los drivers de hardware, así como algunas características especiales, vienen en un paquete llamado Kernel, que se distribuye en código fuente, y que es necesario configurar y compilar cuando se quiera cambiar o añadir alguna de estas características.

El 1% restante lo forman fabricantes de Hardware que en lugar de proporcionar el código fuente del driver, dan ya el driver compilado, en forma de modulo insertable, siendo las principales desventajas de este sistema el que debes utilizar la versión de kernel para la que el fabricante creo el modulo (que nunca suele ser la ultima), y que no puedes confiar en su seguridad (ni puede ser auditada externamente).

El kernel tiene un numero de versión compuesto por tres cifras, la segunda si es impar indica un kernel en desarrollo, y no utilizable en entornos de producción, si es par, es un kernel estable, la ultima indica el numero de revisión, y suele actualizarse cada poco tiempo, con soluciones a fallos y nuevos drivers.

Cada cierto tiempo la versión en desarrollo se convierte en estable, añadiendo a Linux nuevas características.

En el momento de escribir esto acaba de salir la versión 2.6.0, la primera estable desde hace tiempo, con nuevas características, aun así la versión 2.4.22, anterior estable, aun sufrirá alguna revisión para asegurar la estabilidad en aquellos sistemas que no quieran o no puedan usar la 2.6.

Debido a que cada vez son mas y mas las características que tiene el kernel, muchas de ellas están separadas en módulos, piezas de kernel que solo se utilizan cuando son necesarias, y permiten reducir el tamaño general del kernel y hacerlo mas manejable, durante la configuración del kernel deberemos decidir que partes del kernel estarán disponibles como módulos, cuales de manera permanente y cuales no estarán disponibles.

Así p.e. si no tenemos un sistema bluetooth, podemos configurarlo como no disponible. El sistema de Acceso a CD podemos ponerlo como modulo, si el CD no lo utilizamos constantemente, en cambio el soporte para Discos Duros deberemos tenerlo disponible constantemente, pues de lo contrario no podremos usar el S.O.

Como durante el proceso hay que compilar el kernel, es necesario tener instalado el compilador de C, así como sus herramientas auxiliares, y las librerías de TK, o a partir de la versión 2.6 QT o GTK si quieres configurarlo de manera grafica.

El kernel puede ser bajado desde www.kernel.org donde encontraremos una lista de mirrors, esto es sitios iguales, que tienen el paquete, ya que es un archivo muy grande, con tamaños próximos a los 30 Mb, también están disponibles parches, para en el caso de tener la versión anterior del kernel, ahorrar tiempo de descargar.

El kernel debe ser descomprimido en `/usr/src`.

Se ejecuta `make clean` para asegurar una compilación limpia, o si quieres empezar de cero `make mrproper`.

Se ejecuta el programa de configuración `make xconfig` (modo grafico), `make gtkconfig` (modo gnome solo a partir de 2.6), `make qtconfig` (modo KDE solo a partir de 2.6), `make menuconfig` (modo texto).

Una vez echo esto se procede a compilar he instalar.

`Make dep` (solo en versiones previas a la 2.6, o si cambias opciones y recompilas)

`make bzImage; make modules; make modules-install`

Tras lo cual queda copiar el “kernel” un archivo llamado vmlinuz desde /usr/src/Linux/arch/i386/boot/bzImage a /boot o al lugar donde le tengamos indicado a LILO que busque el kernel, ejecutamos LILO para actualizar los cambios y reiniciamos.

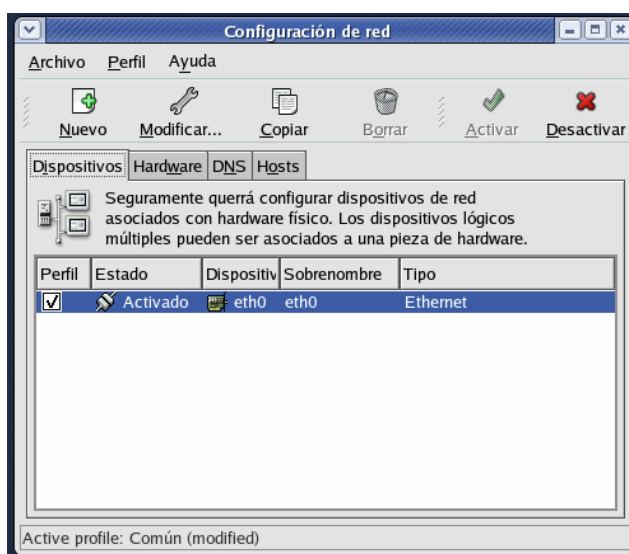
Opcionalmente puedes hacer make install, que realizara las ultimas opciones por ti.

Configuración de Redes

La configuración de las redes en Linux se realiza principalmente en el archivo /etc/network/interfaces, donde se especificaran las características de las tarjetas de red, su IP si es fija, puerta de acceso, redes, etc... en la pagina man (man 1 interfaces) hay una completa guía de cada uno de los parámetros que se le pueden poner.

La IP de los servidores de nombre de dominio se pondrán en el archivo /etc/resolv.conf así como el dominio al que pertenece ese ordenador. El nombre de la red debe estar también en /etc/networks

El nombre del PC debe estar en /etc/hostname, el par nombre IP debería estar también en el archivo /etc/hosts para asegurar el correcto funcionamiento de todos los programas.



Utilidad Grafica para configurar la red

Iptables

En caso de que el PC tenga varias tarjetas de red, o que proporcione servicios de red a otros ordenadores, debe activarse las opciones de iptables y ipforwarding en el núcleo y pasarle al inicio las reglas de funcionamiento.

Todo esto se realiza a través de ficheros en /etc/init.d como ya vimos en la sección de Niveles de Inicio.

Servicios SAMBA

Samba es un conjunto de programas que permiten la integración de Linux con otros Unix o con los sistemas operativos de Microsoft, gracias a samba es posible sustituir un Servidor NT o XP Profesional con un servidor Linux, principalmente proporciona dos servicios:

- Compartir Archivos
- Compartir Impresoras

Adicionalmente proporciona acceso a archivos e impresoras en sistemas con Windows, servicios de identificación, gestion de dominios, y otras opciones.

Se controla desde el archivo `/etc/samba/smb.conf`, y si esta instalado `swat`, tiene un interfaz gráfico funcionando como un servidor web en `http://localhost:691`, además también esta disponible un simple interfaz gráfico.

Servicios de DNS

DNS es el sistema que convierte las direcciones IP en nombre de dominio, más fáciles de recordar.

Configuración local

La resolución de nombres en Linux se controla con dos archivos:

`/etc/resolv.conf` controla el orden de los sistemas a los que se preguntará, aquí es donde hay que poner las IP de los DNS

`/etc/hosts` contiene una lista de pares, IP – Nombre de dominio que pueden ser resueltos sin falta de consultar al dns, útil en pequeñas intranets donde no merezca la pena montar un dns.

En cuanto a los servidores de DNS, tenemos dos opciones en función de las necesidades:

Pdns

Es un proxy dns, su misión es reducir y acelerar el tráfico entre la intranet e internet, es útil en pequeñas intranets con acceso a internet.

Sirve además para resolver nombres de dominio a pequeña escala, cogiéndolos del archivo `/etc/hosts`.

Su principal ventaja es su pequeño tamaño y fácil configuración, en contra su limitada funcionalidad

Bind

En el caso de necesitar servicios de resolución de nombres más avanzadas, bien por tener que proporcionar un gran número de respuestas, por servir varios dominios u otras características se utiliza `bind`.

`Bind` se controla a través de una serie de archivos, de sintaxis concreta.

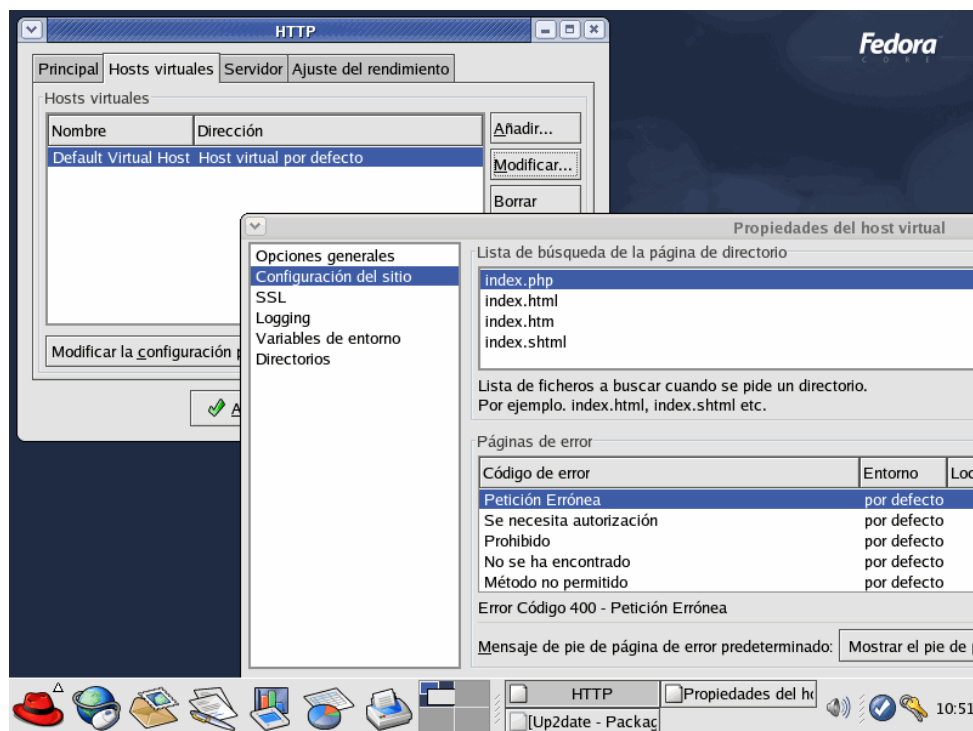
Servicios de web y ftp

Servicios Web

Existen una gran multitud de servidores web disponibles bajo Linux, aunque el más ampliamente utilizado es `apache`.

La configuración de `apache` se realiza a través de varios archivos localizados en `/etc/apache/` siendo el principal `/etc/apache/httpd.conf` donde se configura los posibles servidores virtuales (`apache` puede responder a distintos nombres y proporcionar webs diferentes a cada uno de ellos), y todo lo relacionado con su funcionamiento.

`Apache` tiene una serie de módulos que le permiten aumentar o mejorar las características que trae, son controlados desde `/etc/apache/modules.conf`



Utilidad Grafica para Configurar Apache

Servicios de FTP

Entre todos los servidores de ftp disponibles, porFTPD destaca por ser seguro y fácil de administrar.

En /etc/proftpd.conf en archivo con una estructura similar a la de los html, es posible controlar por un lado el aspecto del servidor, nombre, puerto que escucha, etc...

Por defecto deja entrar a los usuarios no root, y navegar por el sistema de ficheros como si estuviesen en una sesión de shell. Limitarlos a su propio directorio es tan simple como añadir una línea que ponga DefaultRoot ~ y reiniciar el demonio.

Es posible controlar además el acceso de usuarios anónimos, a que directorios pueden acceder, subir cosas, etc...

Así mismo se puede controlar la velocidad, numero de accesos simultáneos, cuota de disco y al igual que en apache, crear servidores virtuales que respondan de manera diferente a distintos nombres.

Servicios de Acceso Remoto

Existen cuatro posibilidades de acceso remoto a un ordenador con Linux

Servidor X

Un ordenador ejecutando un servidor X, puede recibir peticiones desde otro conectado en red para mostrar en su pantalla un programa, este sistema solo es util en equipos conectados en la misma red local.

VNC

Este sistema es similar al anterior, pero además permite ser utilizado en conexiones lentas, incluso de modem y tiene persistencia, esto es, si desconectas el cliente, y conectas desde otro sitio, todo sigue como lo dejaste.

Telnet

Es el sistema mas simple y antiguo, abre una session en modo texto en otro ordenador, el principal problema es que todo viaja por Internet sin seguridad, incluyendo contraseñas o otros datos sensibles.

SSH

SSH permite la ejecución remota de programas, incluyendo el shell, lo que en la practica lo hace una alternativa a Telnet, su diferencia esta en que todo viaja por la red de forma cifrada, lo que aumenta la seguridad, también es posible comprimir los datos o utilizarlo como túnel y así tener p.e. una sesión de VNC segura gracias al ssh.

Apéndice A – Editor de Textos VI

Rápido tutorial de Vi

El propósito de este tutorial es que comience a usar el editor vi. Este tutorial asume que no tiene ninguna experiencia con vi, así que le mostraremos los diez comandos más básicos de vi. Estos comandos fundamentales son suficientes para realizar la mayoría de sus necesidades de edición, y puede expandir su vocabulario de vi cuanto necesite. Se recomienda que tenga un ordenador para practicar, según avanza con el tutorial.

Ejecutar vi

Para ejecutar vi, simplemente tiene que teclear las letras vi seguidas del nombre de fichero que desea crear. Verá una pantalla con una columna de tildes (~) en el lado izquierdo. vi está ahora en modo de comando. Cualquier cosa que teclee será interpretada como un comando, no como texto que usted desea escribir. Para introducir texto, tiene que teclear un comando. Los dos comandos de entrada básicos son los siguientes:

i insertar texto a la izquierda del cursor.

a añadir texto a la derecha del cursor.

Dado que está al comienzo de un fichero vacío, no importa cual de estos usar. Escriba uno de ellos, y después teclee el siguiente texto (un poema de Augustus DeMorgan encontrado en The Unix Programming Environment por B.W. Kernighan y R. Pike):

Las pulgas grandes tienen pequeñas pulgas<Intro>

sobre sus espaldas para que les muerdan.<Intro>

Y las pulgas pequeñas tienen pulgas más pequeñas<Intro>

y así hasta el infinito.<Intro>

Y las pulgas grandes, a su vez,<Intro>

tienen pulgas más grandes sobre las que estar;<Intro>

Mientras que estas de nuevo tienen otras más grandes aun,<Intro>

y más grandes aun, y así.<Intro>

<Esc>

Fíjese que tiene que pulsar la tecla `|_Esc_|` para finalizar la inserción y volver al modo de comando.

Comandos de movimiento del cursor

h mueve el cursor un espacio a la izquierda.

j mueve el cursor un espacio abajo.

k mueve el cursor un espacio arriba.

l mueve el cursor un espacio a la derecha.

Estos comandos se pueden repetir manteniendo la tecla pulsada. Intente mover el cursor por el texto ahora. Si intenta un movimiento imposible, por ejemplo, pulsar la tecla `[_k_]` cuando el cursor está en la línea superior, la pantalla parpadeará momentáneamente o el terminal sonará. No se preocupe, no muerde, y su fichero no saldrá dañado.

Borrar texto

x borra el carácter que hay en el cursor.

dd borra la línea donde está el cursor.

Mueva el cursor a la primera línea y póngalo de modo que esté bajo la n. Pulse la letra `[_x_]`, y la n desaparecerá. Ahora pulse la letra `[_i_]` para cambiarse al modo de inserción y vuelva a teclear la n. Pulse `[_Esc_]` cuando haya terminado.

Salvar un fichero

:w salvar (escribir al disco).

:q salir.

Asegúrese de que está en modo de comando pulsando la tecla `[_Esc_]`. Ahora pulse: w. Esto salvará su trabajo escribiéndolo a un fichero de disco.

El comando para salir de vi es `[_q_]`. Si quiere combinar el salvar y salir, escriba: wq. También hay una abreviación para: wq ZZ. Dado que gran parte del trabajo de programación consiste en ejecutar un programa, encontrar algún problema y llamar el programa en el editor para hacer algún pequeño cambio, y luego volver a salir del editor para ejecutar el programa otra vez, ZZ será un comando que usará a menudo. (En realidad, ZZ no es un sinónimo exacto de: wq si no ha hecho ningún cambio al fichero que está editando desde la última vez que lo salvo, ZZ simplemente saldrá del editor mientras que :wq salvará (redundantemente) el fichero antes de salir).

Si usted se lo ha cargado todo sin esperanzas y simplemente quiere volver a empezar todo, puede teclear: q! (recuerde pulsar la tecla `[_Esc_]` primero). Si omite el !, vi no le permitirá salir sin salvar.

¿Qué viene a continuación?

Los diez comandos que acaba de aprender deberían ser suficientes para su trabajo. Sin embargo, solamente ha rozado la superficie del editor vi. Hay comandos para copiar material de un lugar del fichero a otro, para mover material de un lugar a otro, para mover material de un fichero a otro, para ajustar el editor a sus gustos personales, etc. Con todo, hay unos 150 comandos.

Tutorial avanzado de Vi

La ventaja y la potencia de vi está en la habilidad de usarlo con éxito con sólo saber unos pocos comandos. La mayoría de los usuarios de vi se sienten un poco incómodos al principio, sin embargo después de algún tiempo se encuentran ávidos de mayor conocimiento de los comandos disponibles.

El siguiente tutorial asume que el usuario ha completado el tutorial rápido (arriba) y por lo tanto se siente a gusto con vi. Este tutorial expone algunas de las características más poderosas de ex/vi desde copiar texto hasta la definición de macros. Hay una sección sobre ex y los parámetros para ajustar el editor a su gusto. Este tutorial describe los comandos, en vez de llevarle grupo por grupo.

Se recomienda que usted disponga de algún tiempo para probar los comandos en algún texto de ejemplo, que se pueda permitir destruir.

Este tutorial no expone todos los comandos de vi, aunque se cubren los más comúnmente usados y aún otros adicionales. Aunque usted elija usar otro editor de texto, se espera que apreciará vi y lo que ofrece a quienes deciden usarlo.

Movimiento

La funcionalidad más básica de un editor es el mover el cursor por el texto. Aquí están los comandos de movimiento.

h mueve el cursor un espacio a la izquierda.

j mueve el cursor una línea hacia abajo.

k mueve el cursor una línea hacia arriba.

l mueve el cursor un espacio a la derecha.

Algunas implementaciones también permiten el uso de las teclas de cursor para mover el cursor.

w mueve al principio de la siguiente palabra.

e mueve al final de la siguiente palabra.

E mueve al final de la siguiente palabra antes de un espacio.

b mueve al principio de la palabra anterior.

0 mueve al principio de la línea actual.

^ mueve a la primera palabra de la línea actual.

\$ mueve al final de la línea.

<CR> mueve al principio de la siguiente línea.

- mueve al principio de la línea anterior.

G mueve al final del fichero.

lG mueve al principio del fichero.

nG mueve a la línea n.

<Cntl> G muestra el número de línea actual.

% va al paréntesis correspondiente.

H mueve a la línea superior en pantalla.

M mueve a la línea de en medio de la pantalla.

L mueve al final de la pantalla.

n_ mueve el cursor a la columna n.

El texto se desplaza automáticamente cuando el cursor alcanza la parte superior o inferior de la pantalla. También hay comandos que permiten controlar el desplazamiento del texto.

<Cntl> f desplaza una pantalla hacia delante.

<Cntl> b desplaza una pantalla hacia atrás.

<Cntl> d desplaza media pantalla hacia abajo.

<Cntl> u desplaza media pantalla hacia arriba

Los comandos anteriores controlan el movimiento del cursor. Algunos de los comandos usan un modificador de comandos en la forma de un número que precede al comando. Esta característica normalmente repite el comando ese número de veces.

Para mover el cursor ocho posiciones a la izquierda.

8l mueve el cursor 8 posiciones a la izquierda.

Si desea introducir un número de espacios delante de un texto, podría usar el modificador de comandos con el comando insertar. Introduzca el número de repeticiones (n) y después `|_i_` seguido por el espacio y luego pulse la tecla `|_Esc_`.

`ni` inserta algún texto y lo repite n veces.

Los comandos que tratan con líneas usan el modificador para referirse al número de línea. El `|_G_` es un buen ejemplo.

`1G` mueve el cursor a la primera línea.

`vi` tiene un voluminoso grupo de comandos que se puede usar para mover el cursor por el texto.

`vi` también puede posicionar el cursor en una línea desde la línea de comandos.

`vi +10 mi_fichero.tex`

Este comando abre el fichero llamado `mi_fichero.tex` y pone el cursor 10 líneas más abajo del comienzo del fichero.

Pruebe los comandos de esta sección. Muy poca gente los recuerda todos en una sola sesión. La mayor parte de los usuarios solamente utiliza una parte de estos comandos.

Ahora se puede mover, pero ¿cómo modificar el texto?

Modificación del texto

El propósito es cambiar el contenido del fichero y `vi` ofrece un amplio conjunto de comandos para ayudarnos en este proceso.

Esta sección se concentra en añadir texto, cambiar el texto existente y borrar texto. Al final de esta sección usted tendrá el conocimiento para crear cualquier fichero de texto que desee. Las secciones restantes se concentran en comandos más especializados y convenientes.

Cuando usted teclea nuevo texto puede introducir varias líneas usando la tecla `|_Intro_`. Si hay que corregir un error tipográfico mientras introduce texto, puede usar la tecla `|_Retroceso_` para mover el cursor sobre el texto. Las varias implementaciones de `vi` se comportan de manera diferente. Algunas simplemente mueven el cursor hacia atrás y el texto sigue viéndose y es aceptado. Otras borran el texto cuando pulsa la tecla de borrado. Otras versiones, incluso, permiten usar las teclas de cursores para mover el cursor mientras se está en modo de entrada de texto. Este modo de funcionamiento no es el normal de `vi`. Si el texto es visible y usa la tecla `|_Esc_` mientras está en la línea en la que pulsó `|_Retroceso_`, entonces el texto que hay después del cursor se borrará. Use el editor para acostumbrarse a este comportamiento.

`a` Añadir texto a partir de la posición actual del cursor.

`A` Añadir al final de la línea.

`i` Insertar texto a la izquierda del cursor.

`I` Insertar texto a la izquierda del primer carácter que no sea espacio en la línea actual.

`o` Abrir una nueva línea y añade texto debajo de la línea actual.

`O` Abrir una nueva línea y añade texto encima de la línea actual.

`vi` tiene un pequeño grupo de comandos para borrar texto que se pueden mejorar con el uso de modificadores.

`x` Borrar el carácter que está debajo del cursor.

`dw` Borrar desde la posición actual al final de la palabra.

`dd` Borrar la línea actual.

`D` Borrar desde la posición actual al final de la línea.

Los modificadores se pueden usar para aumentar la potencia de los comandos. Los siguientes ejemplos son un subgrupo de las posibilidades.

`nx` Borrar n caracteres desde el que está bajo el cursor.

ndd Borrar n líneas.

dnw Borrar n palabras (igual que ndw).

dG Borrar desde la posición actual hasta el final del fichero.

d1G Borrar desde la posición actual hasta el principio del fichero.

d\$ Borrar desde la posición actual al final de la línea. (Esto es igual que D).

dn\$ Borrar desde la línea actual al final de la enésima línea.

La lista de comandos anterior muestra que la operación de borrado puede ser muy útil. Esto se hace patente cuando se aplica en combinación con los comandos de movimiento de cursor. Hay que hacer notar que el comando `|_D_|` ignora los modificadores que se le apliquen.

En ocasiones usted puede necesitar deshacer los cambios al texto. Los siguientes comandos recuperan el texto que había antes de hacer cambios.

u Deshacer el último comando.

U Deshacer todos los cambios ocurridos en la línea actual.

:e! Editar otra vez. Recupera el estado del fichero desde la última vez que se salvó.

vi no sólo le permite deshacer cambios, también puede dar la vuelta al comando "deshacer". Por ejemplo, si usamos el comando `|_5dd_|` para borrar 5 líneas, se pueden recuperar usando el comando `|_u_|`. Si usted usa `|_u_|` una vez más, las 5 líneas desaparecerán de nuevo.

vi ofrece comandos que permiten que se hagan cambios al texto sin tener que borrar y volver a escribir:

rc Reemplaza del carácter bajo el cursor con la letra "c". Se Mueve el cursor al modificador de repetición, por ejemplo: 2rc.

R Sobreescribe el texto con el nuevo texto..

cw Cambia el texto de la palabra actual.

c\$ Cambia el texto desde la posición actual al final de la línea.

cnw Cambia las siguientes n palabras. (Igual que ncw).

cn\$ Hacer cambios hasta el final de la enésima línea.

C Hacer cambios hasta el final de la línea actual.

cc Hacer cambios en la línea actual.

s Sustituye el texto que escriba por el carácter actual.

ns Sustituye el texto que escriba por los siguientes n caracteres.

Las series de comandos de cambio que permiten que se introduzca un grupo de caracteres se terminan con la tecla `|_Esc_|`.

El comando `|_cw_|` empieza en la posición actual en la palabra y termina al final de la palabra.

Cuando use un comando de cambio que especifica la distancia sobre la que se aplicará el cambio, vi pondrá un \$ en la última posición de carácter. El nuevo texto puede sobrepasar o no llegar a la longitud original de texto.

Copiar y mover bloques de texto

El movimiento de texto supone la combinación de un número de comandos para conseguir el propósito final. Esta sección introduce los buffers¹ con y sin nombre junto a los comandos que cortan y pegan el texto.

La copia de texto se realiza en tres pasos principales:

¹ N. del T.: "buffer" es una palabra inglesa que se refiere a un espacio reservado en memoria para guardar alguna información temporalmente. En este caso se usan buffers para almacenar una copia temporal del texto que usted desea copiar o mover.

1. Copia del texto a un buffer.
2. Movimiento del cursor al lugar de destino.
3. Pegar (poner) el texto en el buffer de edición.

El buffer de edición no es sino el propio fichero en el que usted está trabajando.

Para Copiar texto al buffer sin nombre use el comando `[_y_]`.

`yy` Mueve una copia de la línea actual al buffer sin nombre.

`Y` Mueve una copia de la línea actual al buffer sin nombre.

`nyy` Mueve las siguientes `n` líneas al buffer sin nombre.

`nY` Mueve las siguientes `n` líneas al buffer sin nombre.

`yw` Mueve una palabra al buffer sin nombre.

`ynw` Mueve `n` palabras al buffer sin nombre.

`nyw` Mueve `n` palabras al buffer sin nombre.

`y$` Mueve el texto desde la posición actual al final de la línea al buffer sin nombre.

El buffer sin nombre es un buffer temporal que puede ser fácilmente afectado por otros comandos comunes. Hay ocasiones en las cuales necesitamos mantener el contenido de un buffer durante un período más extenso de tiempo. En este caso se usaría un buffer con nombre. `vi` tiene 26 buffers con nombre. Los buffers usan las letras del alfabeto como el nombre de identificación. Para distinguir un comando de un buffer con nombre, `vi` usa el carácter `[_"_]`. Cuando use un buffer con nombre de letra minúscula los contenidos se sobre escriben, mientras que el uso del nombre en letra mayúscula añade el nuevo texto al contenido actual.

"`ayy` Mueve la línea actual al buffer con nombre `a`.

"`aY` Mueve la línea actual al buffer con nombre `a`.

"`byw` Mueve la palabra actual al buffer con nombre `b`.

"`Byw` Añade la palabra actual al contenido del buffer `b`.

"`by3w` Mueve las siguientes 3 palabras al buffer `b`.

Use el comando `[_p_]` para pegar el contenido del buffer al buffer de edición.

`p` Pegar del buffer sin nombre a la DERECHA del cursor.

`P` Pegar del buffer sin nombre a la IZQUIERDA del cursor.

`nP` Pegar `n` copias del buffer sin nombre a la IZQUIERDA del cursor.

"`aP` Pegar del buffer con nombre `"a"` a la DERECHA del cursor.

"`b3P` Pegar 3 copias del buffer con nombre `"b"` a la IZQUIERDA del cursor.

Cuando use `vi` dentro de un `xterm` tiene otra opción para copiar texto. Marque el bloque de texto que desea copiar arrastrando el cursor del ratón sobre el texto. Si mantiene pulsado el botón izquierdo del ratón y arrastra el ratón desde el comienzo al final, invertirá el vídeo del texto. Esta operación automáticamente coloca el texto en un buffer reservado para el servidor de X Windows.

Para pegar el texto, pulse el botón del medio. Acuérdesse de poner `vi` en modo de inserción dado que la entrada de texto se podría interpretar como comandos y el resultado sería cuando menos desagradable. Usando la misma técnica se puede copiar una sola palabra haciendo doble click con el botón izquierdo del ratón sobre la palabra. Solamente se copiará esa única palabra. Para pegarla se usa el mismo procedimiento que anteriormente. Los contenidos del buffer sólo cambian cuando se marca otro bloque de texto.

Para mover un bloque de texto son necesarios tres pasos.

1. Borrar el texto para ponerlo en un buffer con o sin nombre.
2. Mover el cursor a la posición de destino.
3. Pegar el buffer.

El proceso es el mismo que el de copia con un solo cambio en el primer paso para borrar. Cuando se usa el comando `|_dd_|`, la línea se borra y se coloca en el buffer sin nombre. Luego puede poner el Contenido del mismo modo que lo hizo cuando copió el texto en la posición deseada.

"add Borrar la línea y ponerla en el buffer con nombre a.

"a4dd Borrar cuatro líneas y ponerlas en el buffer con nombre a.

dw Borrar una palabra y ponerla en el buffer sin nombre.

Vea la sección sobre el modificado de texto para ver más ejemplos de borrado de texto.

En el caso de que el sistema falle, el contenido de todos los buffers se pierde excepto el contenido de los buffers de edición, que se puede recuperar (ver comandos Útiles).

Búsqueda y cambio de texto

vi dispone de varios comandos para búsqueda. Usted puede buscar desde un solo carácter hasta una expresión completa.

Los dos comandos principales de búsqueda basados en caracteres son `|_f_|` y `|_t_|`.

fC Encuentra el siguiente carácter c. Se mueve a la derecha del siguiente.

Fc Encuentra el siguiente carácter c. Se mueve a la izquierda del anterior.

tc Se mueve a la DERECHA al carácter anterior al siguiente c.

Tc Se mueve a la IZQUIERDA al carácter que sigue al c anterior. vi(En algunas esto es lo mismo que Fc).

; Repite el último comando f,F,t,T.

, Igual que (;) pero cambiando la dirección del comando original.

Si el carácter que usted buscaba no se encuentra, vi emitirá un pitido o le dará otro tipo de señal (tal vez un parpadeo de toda la pantalla).

vi le permite buscar una cadena de texto en el buffer de edición.

/tira Busca hacia la derecha y abajo la siguiente instancia de "tira".

?tira Busca hacia la izquierda y arriba la siguiente instancia de "tira".

n Repite el último comando / o ?.

N Repite el último comando / o ? en la dirección opuesta.

Cuando use los comandos `||_|` o `|_?_|`, la última línea en la parte baja de la pantalla se vaciará.

Introduzca en esa línea la tira de caracteres que desea buscar y pulse `|_Intro_|`.

La tira en el comando `||_|` o `|_?_|` puede ser una expresión regular. Una expresión regular puede ser una descripción de un conjunto de tiras. La descripción se construye usando texto entremezclado con caracteres especiales. Los caracteres especiales de una expresión regular son. `* [] ^ $`.

. Vale por cualquier carácter excepto el carácter de nueva línea.

" Tecla de "escape" para cualquier carácter especial.

* Vale por 0 o más instancias del carácter anterior.

[] Busca exactamente uno de los caracteres incluidos entre los corchetes.

^ El carácter que sigue a ^ debe estar al principio de la línea.

\$ El carácter que precede a \$ se busca al final de la línea.

[^] Busca cualquier cosa que no se encuentre después de ^ dentro de los corchetes.

[-] Busca un rango de caracteres.

El único modo de acostumbrarse a usar las expresiones regulares es usándolas. A continuación mostramos varios ejemplos de uso:

c.ch Coincide con coche, cacho, cochera, etc.

c".pe Coincide con c.pe, c.per, etc.

sto*p Coincide con stp, stop, stoop, etc.

car.*n Coincide con cartón, carrillón, carmen, etc.

xyz.* Coincide con xyz al final de la línea.

^Los Encuentra cualquier línea que empiece con Los.

abcdef\$ Encuentra cualquier línea que termine con abcdef.

^Solo\$ Coincide con cualquier línea que sólo tenga la palabra "Solo".

p[aiue]so Coincide con paso, piso, puso, peso.

Ver[D-F] Coincide con VerD, VerE, VerF.

Ver[^1-9] Coincide con Ver siempre que a continuación NO haya un número.

the[ir][re] Coincide con their, therr, there, theie.

[A-Za-z][A-Za-z]* Coincide con cualquier palabra.

vi usa el modo de comandos de ex para llevar a cabo búsquedas y sustituciones. Todos los comandos que empiezan con dos puntos son comandos en modo ex.

Los comandos de búsqueda y sustitución permiten que se usen expresiones regulares sobre un rango de líneas y sustituir la tira de caracteres que se busca. El usuario puede pedir confirmación antes de que se realice la sustitución. Merece la pena hacer una revisión de la representación de los números de línea en el tutorial de ed.

:<prim>,<ult>s/<busca>/<sust>/g Comando general.

:1,\$s/los/Los/g Buscar el fichero entero y sustituir los por Los.

:%s/the/The/g % significa el fichero entero (igual que arriba).

::,5s/^.*//g Borra el contenido de la línea actual hasta la quinta.

:%s/los/Los/gc Sustituye "los" con "Los" preguntando antes de hacer la operación.

:%s/^.*/g Borra los primeros cuatro caracteres de cada línea.

El comando de búsqueda es muy potente cuando se combina con las expresiones regulares. Si no se incluye la directiva |_g_| entonces el cambio sólo se aplica a la primera instancia que se encuentre en cada línea.

En ocasiones usted quiere usar los caracteres de búsqueda en los caracteres de sustitución. Se puede reescribir todo el comando en la línea pero vi permite que los caracteres de sustitución incluyan algunos caracteres especiales.

:1,5s/ayuda/&ndo/g Sustituye ayuda por ayudando en las cinco primeras líneas.

:%s/ */&&/g Duplica el número de espacios entre palabras.

El uso de la tira de caracteres completa para búsqueda tiene sus límites y por lo tanto vi usa los paréntesis en secuencia de escape |_(_| y |_)_| para seleccionar el rango de la sustitución. El uso de un dígito en secuencia de escape |_1_| identifica el rango en el orden de la definición y la sustitución se puede hacer.

:s/^^(".*").*/"1/g Borra todo después de e incluyendo los dos puntos.

:s/(".*"):"(.*")/"2:"1/g Da la vuelta a las palabras a los dos lados de los dos puntos.

vi ofrece comandos muy potentes que muchos editores de texto modernos no tienen o no pueden ofrecer. El coste de esta potencia suele ser también el principal argumento en contra de vi. Los comandos pueden ser difíciles de aprender y leer. Sin embargo, la mayoría de las cosas buenas suelen ser un poco extrañas al principio. Con un poco de práctica y tiempo, los comandos de vi se convertirán en una segunda naturaleza para Ud..

Apéndice B - Sitios de Internet

Esta es una pequeña lista de sitios con Libros en formato electrónico sobre Linux.

<http://es.tldp.org> - contiene la mayoría de la documentación que existe para Linux traducida al español, los libros de linuxdoc, los "comos" traducción de los howto y las paginas del man. Todas estas traducciones están echas por un equipo de voluntarios por lo que no siempre están completas o actualizadas, aun así son una referencia imprescindible.

Destacan entre otro el *Guía de instalación Oficial de RedHat*, la *Guía del usuario de Linux*, y *Linux, instalación y 1º Pasos*

<http://www.tldp.org> contiene el mismo tipo de información que Lucas, pero debido a que la mayoría de la documentación se escribe en ingles, su contenido es mas completo y/o actualizado.

De <http://andercheran.aiind.upv.es/toni/personal/index.html> puedes bajarte un interesante libro llamado *Introducción al sistema operativo Unix* y otro manual sobre seguridad.

<http://www.tayuda.com> tiene una serie de libros sobre informática llamada "*aprende informática como si estuvieses en primero*" con uno dedicado a Linux en la sección de sistemas operativos.

Además puedes encontrar información y noticias (en español) sobre Linux en:

<http://www.barrapunto.com> – Noticias del S.L. Y en general.

<http://www.hispalinux.org> – Asociacion de Usuarios de Linux (España)

<http://www.asturlinux.org> – Asociacion de Usuarios de Linux (Asturias)

<http://www.laespiral.org> – Informacion sobre Debian

<http://www.debian-es.org> - Usuarios de Debian

<http://www.fedora-es.com> - Usuarios de Red Hat

<http://www.gentoo-es.org> - Usuarios de Geento

Y en ingles en:

<http://www.linux.com> – Portar comercial sobre Linux

<http://www.freshmeat.net> – Noticias sobre Software Nuevo o Actualizado

<http://www slashdot.org> – Noticias de S.L. Y en general

<http://www.linuxgames.com> – Juegos para Linux

<http://www.happypenguin.org> – Juegos para Linux

Web de las distribuciones:

<http://www.redhat.com>

<http://www.mandrakesoft.com>

<http://www.debian.org>

<http://www.gentoo.org>

Apéndice C -Recursos

Material Externo

El apéndice A – Editor de Textos VI ha sido sacado de Linux Instalación y Primeros Pasos, de Matt Wells, disponible en <http://es.tldp.org>.

Las capturas de pantalla del Tema 3 – Entorno Grafico han sido obtenidas de <http://www.kde.org> y <http://www.gnome.org> respectivamente.

El Autor

Alvaro Alea Fernández, (email: alvaro@asturlinux.org) es Ingeniero Técnico Electrónico, trabaja como administrador de sistemas Linux, y Diseño grafico, es socio fundador de la asociación Asturlinux, (<http://www.asturlinux.org>)y en sus ratos libres intenta programar un emulador de spectrum en C bajo licencia GPL (<http://aspectrum.sourceforge.net>)

Versión

2002/XX/XX - Primera Versión.

2003/12/20 - Añadido Información sobre la instalación y servicios de Red, Capturas de Pantalla.

Licencia

Se permite la libre copia y distribucion de este documento en lo terminos de la Licencia GPL V2.

Las marcas registradas son propiedad de sus respectivos dueños.