

# MATERIAL COMPLEMENTARIO PARA EL SEMINARIO

## Asignatura: Informática Básica Semestre 1 1er año

**Actividad 1:** Los sistemas operativos sustentados en el software libre.

### Objetivos:

- Identificar los sistemas operativos sustentados Software Libre y Software Propietario.
- Conocer los conceptos de libertad, gratuidad, código fuente, fuente abierta (Open Source), licencias, kernel o núcleo de un sistema, distributiva y versión de un sistema.
- Identificar los requerimientos técnicos para la instalación de una distribución de GNU/Linux

### Contenido:

1. Los sistemas operativos sustentados en el software libre.  
Particiones activas de un HDD. Posibilidades del multisistema para la convivencia de un sistema operativo libre, con Microsoft Windows u otros.  
Cargadores de arranque para el multisistema (LILO, GRUB).
2. Posibilidades del uso de productos informáticos libres en Microsoft Windows.  
(Plataforma Ofimática Open Office y navegador Firefox, etc).
3. Principales distribuciones de GNU/Linux: Red Hat, Mandrake, Knoppix, Debian, Suse, Ubuntu, etc.
4. Requerimientos técnicos para la instalación del sistema objeto de estudio: particiones del HDD, memoria RAM disponible, periféricos conectados a la computadora, y otros
5. Particiones necesarias para la instalación de un sistema operativo de GNU/Linux. Instalación de una versión de la distribución.
6. Sistemas de ficheros más utilizados por los sistemas operativos de GNU/Linux (EXT2, EXT3, RaiserFS, FAT, NTFS).

### Descripción de la bibliografía que se sugiere:

Para el desarrollo de esta unidad se propone la utilización de las bibliografías digitales, entre los que se encuentran:

1. Baig Viñas, R, Francesc Aulí, LL. Sistema operativo GNU/Linux 71Z799002MO
2. Becerra García, F. Manual Básico del Procesador de Texto OpenOffice Writer 1.0
3. Belkin, S. . Manual compacto para nuevos usuarios de Sistemas Linux Versión 2.0 Buenos Aires, Argentina (2005)
4. Cabrera Padrón, D M, Estrada Rodríguez, M. Ofimática básica. (segunda edición). La Habana (.1999)
5. Corriusi Llavita, J. Software libre Utilidades y herramientas
6. DE Gleduwiki, la enciclopedia libre: Manual Calc.
7. García de Jalón, J, Iker; Mora, A. Aprenda LINUX como si estuviera en primero. San Sebastián.(2000)

8. González Barahona, J, Pascual S, J. . Introducción al Software libre. XP03/90783/0. (1935)
9. González, C. Streaming en Linux (AKA MPEG4IP, DSS, MP4WEB howto)
10. Jorba Esteve, J. Administración avanzada de GNU/Linux XP04/90785/00019
11. Laveda José , M.. Linux fácilmente de Europa. ( 1999)
12. Lusky Raul: Linux, otra opción en sistemas operativos
13. Lusky, R. Linux, otra opción en sistemas operativos. Página 2
14. McQuillan, James A . ( 2002 ). LTSP – Linux Terminal Server Project – v3.0
15. Navarrete Navarrete, C. Cómo recuperar el gestor de arranque de un sistema operativo. Centro de Referencia Linux (CRL, UAM-IBM).
16. Rodríguez Lamas, R. Dra. García de la Vega, D, Dr. González Chong, O y otros. ( 2000). “Introducción a la Informática Educativa” Universidad de Pinar del Río Hermanos Sainz Instituto Superior Politécnico José A. Echeverría
17. Santos , Victor . Bulma: Envío de ficheros entre pc's mediante scp (1716 lectures)
18. Sánchez Costa, Andreu. Internet Web Serveis, S.L Departamento Administración y Gestión de Sistemas tanis@lleida.com.
19. Sitio: Smaldone, Javier. (2004). Introducción a Secure Shell <http://www.smaldone.com.ar> Versión 0.2.
20. Sitio: *Bolo* ( [http://www.linuca.org/todos.phtml?id\\_autor=1](http://www.linuca.org/todos.phtml?id_autor=1) ) 2003
21. Sitio: \192.168.170.8\Materiales\Informatica\bibliografia.htm
22. Sitio:\192.168.170.8\Materiales\Informatica\Bibliografia\SistOp1\Linux\sobrelinux.pdf
23. Sitio: Linux. Instalación y Primeros Pasos. Versión 2.2.2 - En castellano ver 1.0, 8 Agosto de. <http://www.cybercursos.net> . ( 1996). Página 1
24. Sitio: <http://memnon.ii.uam.es.crl@ii.uam.es.jam@LTSP.org> .
25. Tipos de software. Software Libre. Vctrnts ( <http://vctrnts.dyndns.org> ) 2004. XP04/90797/00023.

**Ver además carpeta adjunta en la que aparecen los textos, Software y videos con contenido de la asignatura.**

**Actividad 2:** Los sistemas operativos sustentados en el software libre.

**Sumario:** Posibilidades tecnológicas y económicas con relación a sistemas soportados en el software propietario. Política de Cuba en relación con la migración al Software Libre

Conceptos utilizados en el mundo del Software Libre: libertad, gratuidad, código fuente, código abierto (Open Source), código cerrado, licencia, kernel o núcleo de un sistema, distribución y versión de un sistema. Licencia GPL. Características. Surgimiento y desarrollo de GNU/ Linux como plataforma de software libre. Comunidades de programadores para el desarrollo del Software Libre.

Principales distribuciones de GNU/Linux: Red Hat, Mandrake, Knoppix, Debian, Suse, Ubuntu, etc.

Particiones activas de un HDD. Posibilidades del multisistema para la convivencia de un sistema operativo libre, con Microsoft Windows u otros. Cargadores de arranque para el multisistema (LILO, GRUB).

**Objetivo:** Caracterizar los sistemas operativos sustentados en el software libre, sus posibilidades tecnológicas y económicas en comparación con los sistemas operativos

propietarios, así como el surgimiento y desarrollo de las diferentes distribuciones, para su instalación.

### **Bibliografía:**

Seleccionar la bibliografía digital que aparece en la carpeta adjunta.

### **Desarrollo**

Para el desarrollo de esta asignatura es importante partir de los conocimientos que los estudiantes tienen del sistema operativo Windows, donde se valore con ellos las ventajas y desventajas de este sistema desde el punto de vista de potencialidades a la hora de poder resolver los problemas, la seguridad de la información contra virus informáticos, la protección de ficheros y otros aspectos que se consideren necesarios.

En el desarrollo de los temas no se debe dejar de utilizar la analogía, pues existen un grupo de conocimientos de sistema operativo que ya han sido estudiados anteriormente, que han tenido relacionado los software propietarios.

### El Software Libre

Es la libertad de usar un programa, con cualquier propósito, la libertad de estudiar cómo funciona el programa y poder adaptarlo a las necesidades de cada cual gracias a la disponibilidad de acceso al código fuente y a que la licencia lo permita, de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar un soporte técnico. Es la libertad de redistribuir copias.

Entre las empresas productoras de software libres tenemos la Apple, Intel, Sun, AOL Time Warner.

Actualmente, existe toda una variedad de servicios alrededor del software libre. Aparejadas a las grandes empresas, en muchos países tanto desarrollados como subdesarrollados hay inquietudes respecto a este tema y han decidido proponer proyectos de leyes que fomenten el incremento de software libre en las administraciones y en aquellas empresas donde el Estado tenga una mayoría de acciones.

Esto lo promueven basándose no solo en el costo de las licencias que no hay con el software libre, sino por evitar la dependencia de terceros países, fundamentalmente norteamericanos o europeos. Tal dependencia limita mucho la generación de riquezas internas y nacionales al resto del mundo.

Por ejemplo, en el caso particular de Alemania, están diseñando con software libre proyectos de seguridad y productos para la defensa. Otro caso significativo es el caso de Perú, que ha tenido algunas contingencias durante el período de aprobación, debido a la intervención que tuvo Microsoft con propagar una ley que ellos pretendían instaurar en el país y que coartaba las libertades nacionales.

### **Entre las ventajas que nos depara el software libre están:**

- El ahorro por la distribución de las licencias.
- El combate efectivo que se realiza contra la copia ilícita del software.
- La eliminación de las barreras presupuestarias, ya que muchos colaboradores se

disponen a intercambiar unos con otros.

- Las aplicaciones son fácilmente auditable, debido a que el código está disponible, y que todo el mundo tiene derecho a usarlo sin costo alguno. Acceder a su diseño, poder utilizarlo, modificarlo y publicarlo posteriormente sin dificultad alguna.

- Es de libre distribución.

- Los tiempos de desarrollo usualmente son menores cuando existe una amplia disponibilidad de herramientas del software

- Los tiempos de desarrollo sobre algo que no exista son menores por la amplia disponibilidad de herramientas y librerías.

El software libre se convirtió en un producto muy eficiente, todos lo revisan, todos lo optimizan, muy robusto porque todos colaboran en mejorarlo, hacerlo más seguro, se particulariza mucho para cada una de las aplicaciones. No podemos negar que el éxito del software libre está dado por el fenómeno de Internet, el cual gracias a su inmediatez ha posibilitado que se coloquen al alcance de muchos programadores los códigos y módulos de software.

Entre los principales proyectos libres que se están desarrollando hoy está el sistema operativo Linux, que es uno de los más conocidos, el cual fue desarrollado por Linus Torvalds a principios de los años '90 (un estudiante finlandés que, con la ayuda de cinco programadores, desarrolló el kernel del sistema operativo Linux, basado en un sistema similar al UNIX).

También está el proyecto GNU, iniciado por Richard Stallman en 1984, con la idea de producir un sistema operativo totalmente libre cuya mayor contribución han sido las bases filosóficas y legales que ha creado. Entre las bases legales se encuentran las licencias GPL, una de las más usadas por los programadores de software libres en estos momentos.

El proyecto Apache, que es el que más se usa en sitios de Internet (más del 58 por ciento de los sitios lo utilizan), y desde el punto de vista técnico es uno de los proyectos más completos que se conoce hoy en día en el mundo del software libre.

Mozilla, promovido por Netscape (la empresa que creó el Navegador) y que fue el primer proyecto de software libre, nos permite disponer de un navegador de Internet muy eficiente y con muchas prestaciones. El proyecto GENOME, que es el que nos brinda la posibilidad de un entorno de escritorio, con ciertas aplicaciones similares al Excel o el Word de Microsoft.

LINUX (o GNU/LINUX, más correctamente) es un Sistema Operativo como MacOS, DOS o Windows. Es decir, Linux es el software necesario para que tu ordenador te permita utilizar programas como: editores de texto, juegos, navegadores de Internet, etc. Linux puede usarse mediante un interfaz gráfico al igual que Windows o MacOS, pero también puede usarse mediante línea de comandos como DOS.

Linux tiene su origen en Unix. Éste apareció en los años sesenta, desarrollado por los investigadores Dennis Ritchie y Ken Thompson, de los Laboratorios Telefónicos Bell.

### **Distribuciones en GNU-Linux**

Una **distribución Linux**, o **distribución GNU-Linux** es un conjunto de aplicaciones reunidas que permiten brindar mejoras para instalar fácilmente un sistema Linux, también llamado GNU-Linux, se destacan por las herramientas para configuración y sistemas de paquetes de software a instalar.

Existen numerosas distribuciones Linux, cada una de ellas puede incluir cualquier número de software adicional, libre o no, como algunas que facilitan la instalación del

sistema y una enorme variedad de aplicaciones, entre ellos, entornos gráficos, suites ofimáticas, servidores web, servidores de correo, servidores FTP, etcétera. La base de cada distribución incluye el núcleo Linux, con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software, como BSD

El núcleo de Linux se suele instalar en el directorio `/usr/src/linux`. Dependiendo de que distribución de Linux tengas instalada lo tendrás allí o acá que instalar algún paquete en el que estén dichas fuentes. Si lo tienes en ese directorio, o en alguno con el nombre kernel `kernel-source-x.x.x` pueden pasar al siguiente apartado.

**Actividad 3 Tema:** Posibilidades del uso de productos informáticos libres en Microsoft Windows. (Plataforma Ofimática Open Office y navegador Firefox, etc).

**Sumario:** Posibilidades del uso de productos informáticos libres en Microsoft Windows. (Plataforma Ofimática Open Office y navegador Firefox, etc). Concepto de Live CD. Facilidades de evaluación de un sistema operativo en versiones Live CD. Kernel del sistema operativo. Versión del sistema según el kernel.

Requerimientos técnicos para la instalación del sistema objeto de estudio: particiones del HDD, memoria RAM disponible, periféricos conectados a la computadora, y otros.

**Objetivo:** Evaluar las posibilidades de la aplicación de los productos informáticos libres en Microsoft Windows. Plataforma Ofimática Open Office y navegador Firefox. Así como los requerimientos técnicos para la instalación del sistema GNU-Linux.

## Desarrollo

### El kernel.

Es el kernel el que se encarga de abstraer las comunicaciones con los dispositivos (la tarjeta de video o de sonido, la controladora del disco duro, la unidad ZIP enchufada al puerto paralelo, la gestión de interrupciones, la coordinación para compartir los recursos entre diferentes aplicaciones que se ejecutan a la vez. El kernel es realmente lo que hace que tú maquina sea algo operativo y funcional, aunque ciertas compañías nos intenten hacer creer que los navegadores de Internet son también parte del sistema operativo.

**OpenOffice.org** es un paquete ofimático que está publicado como software libre y código abierto que incluye dentro de sus aplicaciones un procesador de textos: OpenWriter, hoja de cálculo: OpenCalc, presentaciones electrónicas: OpenImpress, herramientas para el dibujo vectorial: OpenDraw y base de datos: OpenBase. Además está disponible para múltiples plataformas, como son Microsoft Windows, incluido Windows Vista, todo tipo de sistemas Unix con las X Windows, como GNU-Linux, BSD, Solaris y por supuesto también para Mac OS X. Es **compatible con Microsoft Office**, su principal competidor y dispone de soporte nativo para el estándar **OpenDocument** para intercambio de datos. Además **es totalmente gratis**.

### Principales navegadores en GNU-Linux.

Entre los más notables navegadores se encuentran:

Mozilla: En realidad es un conjunto de programas para Internet, entre los que se destacan, el propio Navegador y su mensajero de correo electrónico.

### **Navegador Mozilla**

Mozilla Firefox: También conocido como Firefox a secas. Este navegador está desarrollado por la comunidad de programadores de Mozilla, muchos lo consideran la mejor alternativa a Internet Explorer. Posee más de 120 extensiones que se pueden agregar para aumentar las funcionalidades de la versión estándar. Y al igual que Mozilla se pueden descargar nuevos temas para modificar el aspecto visual del programa.

### **Firefox**

Konqueror

Además de administrador de archivos, sirve para navegar por la Web. Por lo tanto se puede pasar fácilmente de un directorio local a una página web. Es muy útil para usarlo dentro de KDE.

### **Konqueror**

Debido a que Internet Explorer fue el navegador de facto desde hace varios años, los diseñadores web han tenido la costumbre de hacer las páginas web pensadas solamente para que se vean bien en el navegador de Microsoft. Esto sin embargo, significó que muchísimos sitios se apartaran de los estándares propuestos por el World Wide Web Consortium.

La idea original de Internet es que cualquier página se vea correctamente independientemente del software que se utilice. Esto quiere decir que cuando una página web se ve mal en un navegador no quiere decir necesariamente que sea una deficiencia del navegador.

Si un sitio no ve correctamente en navegador distinto de Internet Explorer, se debería notificar al webmaster para que solucione el problema. En especial si se trata de un sitio público o de una empresa de servicios a la cual se está pagando.

No obstante hay que decir que la cantidad de sitios que no se ven correctamente (especialmente en Mozilla o Firefox) es relativamente baja.

### **Instalación. (Ver videos para la instalación de Linux)**

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

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

### **Actividad 4. Tema: Clase práctica**

**Sumario:** Sistemas de ficheros más utilizados por los sistemas operativos de GNU/Linux (EXT2, EXT3, ReiserFS, FAT, NTFS).

**Objetivo:** Ejercitar el trabajo con archivo y carpetas en el sistema operativo Linux.

### **Desarrollo.**

**Comandos para el trabajo con directorios: mkdir, rmdir, cat, more, less. mkdir: crea directorio**

**Comando para crear un directorio.**

· mkdir <ruta> directorio.

**Eliminar directorio.**

## **.rmdir: elimina directorios**

Para borrar directorios se usará el comando rmdir, equivalente al rmdir de MS-DOS. El comando rmdir presenta el siguiente formato:

```
l rmdir [-ri ] directorio
```

Una condición para que el comando funcione correctamente es que los directorios a eliminar **estén vacíos**. Si no lo están, habrá que borrar los ficheros que contiene antes de borrar el directorio.

Para borrar todos los directorios ( vacíos ) que cuelgan de uno dado se utiliza la opción **-r**. Con la opción **-i** entramos en el modo interactivo, el que se nos pregunta antes de eliminar cada directorio.

## **cat, more, less: examinar el contenido de un fichero.**

Una vez que el usuario ya conoce cómo listar los archivos de un directorio, cómo moverse entre directorios y cómo crear y borrar directorios, lo siguiente que surge es la necesidad de examinar el contenido de los ficheros. Los comandos **cat, more y less** permiten hojear el contenido de un fichero, teniendo cada uno sus particularidades que se comentarán a continuación. El formato de los tres comandos es el siguiente:

- **cat fichero**
- **more fichero**
- **less fichero**

cat es el comando más simple, pues muestra el contenido de un fichero mostrándolo por pantalla y sin ningún tipo de pausa. Un caso especial se produce cuando se ejecuta el comando cat sin parámetros. Entonces el comando se queda esperando a que se introduzcan caracteres por pantalla, mostrándolos línea a línea hasta que pulsa Ctrl-D.

more y less sí permiten hacer pausas durante la visualización de los datos. Para avanzar entre pantallas se pulsa cualquier tecla. La diferencia entre ambos consiste en que el comando less utiliza las teclas de flechas para poder avanzar y retroceder por el fichero, cualidad de la que carece el comando more.

Otros comandos para el trabajo con directorios

Comando/Sintaxis	Descripción	Ejemplos
cat <i>fich1</i> [... <i>fichN</i> ]	Concatena y muestra un archivos	cat /etc/passwd
	archivos	cat dict1 dict2 dict
cd [ <i>dir</i> ]	Cambia de directorio	cd /tmp
chmod <i>permisos fich</i>	Cambia los permisos de un archivo	chmod +x miscript
chown <i>usuario:grupo fich</i>	Cambia el dueño un archivo	chown nobody miscript
cp <i>fich1...fichN dir</i>	Copia archivos	cp foo foo.backup
diff [-e] <i>arch1 arch2</i>	Encuentra diferencia entre archivos	diff foo.c newfoo.c
du [-s <i>abr</i> ] <i>fich</i>	Reporta el tamaño del directorio	du -s /home/
file <i>arch</i>	Muestra el tipo de un archivo	file arc_desconocido
find <i>dir test acción</i>	Encuentra archivos.	find . -name ``.bak" -print
grep [-c <i>ilnv</i> ] <i>expr</i> <i>archivos</i>	Busca patrones en archivos	grep mike /etc/passwd
head - <i>count fich</i>	Muestra el inicio de un archivo	head prog1.c
mkdir <i>dir</i>	Crea un directorio.	mkdir temp
mv <i>fich1 ...fichN dir</i>	Mueve un archivo(s) a un directorio	mv a.out prog1
mv <i>fich1 fich2</i>	Renombra un archivo.	mv .c prog_dir
less / more <i>fich(s)</i>	Visualiza página a página un archivo.	more muy_largo.c
	less acepta comandos vi.	less muy_largo.c
ln [-s] <i>fich acceso</i>	Crea un acceso directo a un archivo	ln -s /users/mike/.profile

		.
ls	Lista el contenido del directorio	ls -l /usr/bin
pwd	Muestra la ruta del directorio actual	pwd
rm <i>fich</i>	Borra un fichero.	rm foo.c
rm -r <i>dir</i>	Borra un todo un directorio	rm -rf prog_dir
rmdir <i>dir</i>	Borra un directorio vacío	rmdir prog_dir
tail -count <i>fich</i>	Muestra el final de un archivo	tail prog1.c
vi <i>fich</i>	Edita un archivo.	vi .profile

### **Estructura del sistema de ficheros en GNU-Linux**

Debido a la proliferación de sistemas basados en Unix, se hace claramente y necesaria la estructuración de un estandar de organización para el sistema de archivos, se necesitaba que tanto los programas como los usuarios pudieran predecir con un buen grado de certeza dónde colocar o encontrar los archivos a partir de su función en el sistema.

#### **Un poco de historia**

El proceso de desarrollar este estandar comenzó en 1993 como un esfuerzo por reestructurar la organización de los archivos y directorios en Linux , el FSSTND, una jerarquización del sistema de archivos orientada específicamente a Linux vio la luz el 14 de Febrero de 1994, revisiones posteriores fueron lanzadas en Octubre del mismo año y en Marzo de 1995.

En 1995 el objetivo cambió, ya no se trataba solo de acordar un conjunto de directrices para Linux y su sistema de archivos sino para todos los sistemas operativos basados en Unix. Con la ayuda de la comunidad de desarrolladores de BSD (otro clon de Unix) y un sin número de otros individuos ligados de una u otra forma a comunidades de este tipo, se comenzó a trabajar en este nuevo documento

#### **Clasificación general**

Datos compartibles

Como su nombre lo indica, los datos compartibles son aquellos cuya naturaleza los hace susceptibles de ser `compartidos` con mas de un host, un ejemplo de este tipo de datos son los `home` de los usuarios.

Datos no compartibles

En todo sistema existen datos que solo tienen relación con el `host` que los alberga, fuera de este, pierden toda utilidad (o por lo menos pierden su utilidad oficial, un ejemplo de este tipo de datos son los descriptores de dispositivos o los `lock files` que crean algunos programas en ejecución.

Datos variables o dinámicos

Son datos que varían sin necesitar de la intervención del administrador de sistemas (root), los documentos, como este, son ejemplos de este tipo de datos.

#### **Datos estáticos**

Son datos que solo pueden variar bajo la intervención del administrador de sistemas, por ejemplo, las librerías, los programas instalados, etc.

#### **El sistema de archivos raíz o root (/)**

En un sistema que cumpla con las especificaciones del FHS, los contenidos del sistema de archivos raíz, deben ser suficientes para arrancar, reparar y/o recuperar el sistema.

· Para arrancar el sistema es necesario que se encuentren presentes en `/` las utilidades que permitan `montar` otros sistemas de archivos, archivos de configuración,

información sobre el cargador de arranque (bootloader) y todo otro dato que sea esencial para el arranque de la maquina en cuestión. /usr, /opt y /var están diseñados de manera que puedan estar localizados físicamente en otras particiones.

- Para reparar el sistema, las utilidades que un administrador experimentado o pudiese necesitar deben estar presentes en el sistema de archivos raíz.

- Para restaurar el sistema, deben estar presentes las utilidades que permitan leer desde respaldos en cinta, discos y/o otros medios y copiar al sistema dañado.

Se debe hacer notar que estas instrucciones generales deben ser compatibles con la imperativa necesidad de mantener / lo suficientemente pequeño, muchas veces el sistema de archivos raíz es cargado desde dispositivos con poca capacidad de almacenamiento, además gran parte de los datos en el ubicados son relevantes solo para la máquina a la que pertenecen, por ende, no siempre pueden ser cargados desde otros hosts en la red, se debe hacer notar que un sistema de archivos pequeño es menos vulnerable a la corrupción de datos debida a errores en el medio físico u otros factores .

Software de orientación general no debe nunca almacenar información en / ni requerir archivos o directorios (subdirectorios) especiales en el, para eso existen otras ubicaciones (discutidas mas adelante en este mismo documento).

Directorios requeridos en /

Los siguientes directorios, pueden ser también links a otras ubicaciones en el sistema, pero DEBEN existir como tales en /:

- /bin
- /boot
- /dev
- /etc
- /lib
- /mnt
- /opt
- /sbin
- /tmp
- /usr
- /var
- /bin

Binarios esenciales para el funcionamiento del sistema, un ejemplo de este tipo de binarios en un sistema GNU-Linux son los comandos necesarios para montar otros sistemas de archivos, mount, umount, etc.

/bin contiene binarios que pueden ser usados tanto por el administrador del sistema como por otros usuarios, pero que deben estar disponibles cuando aun no se ha montado ningún otro sistema de archivos adicional. También puede contener binarios que son usados indirectamente por scripts.

### Requerimientos .

/bin no puede contener subdirectorios, los siguientes comandos binarios o links a los mismos DEBEN encontrarse en /bin:

- cat
- chgrp
- chmod
- chown
- cp
- date
- dd
- df
- dmesg
- echo
- false
- hostname
- kill
- ln
- login
- ls
- mkdir
- mknod
- more
- mount
- mv
- ps
- pwd
- rm
- rmdir
- sed
- sh (no la real, un link)
- stty
- su
- sync< item>true
- umount
- uname

### **Opciones específicas**

Pueden existir y en tal caso deben estar en bin pueden ser links.

- csh
- ed
- tar
- cpio
- gzip
- gunzip
- zcat
- netstat
- ping
- /boot

Contiene datos asociados a la tarea de arrancar el sistema, tienen el carácter de `estáticos` y son operados por el gestor de arranque lilo, grub, etc. cabe hacer notar que el map installer y los archivos de configuración del arranque NO pueden estar en /boot.

Opciones específicas

El kernel del sistema operativo DEBE estar en / o en /boot

/dev

Archivos de dispositivo, como bien haz de saber unix fue diseñado con una serie de ideas explícitas en mente, una de las cuales imponía: "todo es un archivo", es en /dev donde reside un gran numero de archivos que representan dispositivos de hardware.

Opciones específicas

Es posible que los descriptores de dispositivo de /dev deban ser creados manualmente, en ese caso, debe existir un comando llamado MAKEDEV en este mismo directorio, que sea capaz de crearlos en caso de ser requeridos. (puede existir tb un MAKEDEV. local, para cualquier dispositivo local)

/etc

Aquí residen los archivos de configuración del sistema, son usados por el administrador y/o otros programas para almacenar y leer datos que alteran de una u otra forma su funcionamiento.

### **Requerimientos**

No pueden existir binarios en /etc

*Se requiere la existencia de el directorio /etc/opt para los datos de configuración de /opt*

Opciones específicas

En caso de estar instalados los subsistemas asociados, los siguientes directorios DEBEN residir en /etc

- X11
- sgml

Los siguientes archivos deben residir en /etc en caso de estar instalados los subsistemas asociados, todos opcionales.

- csh.login
- exports
- fstab
- ftpusers
- gateways
- gettydefs
- group
- host.conf
- hosts
- hosts.allow
- hosts.deny
- hosts.equiv
- hosts.lpd
- inetd.conf
- inittab
- issue
- ld.so.conf
- motd
- mtab
- mtools.conf
- networks
- passwd
- printcap
- profile
- protocols
- resolv.conf
- rpc
- securetty
- services
- shells
- syslog.conf
- **lib**

Contiene librerías esenciales para ejecutar los binarios de /bin y /sbin , contiene también, en caso de existir, los módulos del kernel.

## Requerimientos

Por lo menos uno de cada uno de los siguientes archivos o link a los mismos debe existir

- 
- libc.so.\*
- ld\*
- /mnt

Puntos de montaje temporal para otros sistemas de archivos.

/opt

Contiene paquetes de software opcional.

/sbin

Contiene binarios de sistema esenciales para el funcionamiento del mismo, un ejemplo son los comandos insmod y rmmmod que sirven para instalar y remover módulos en el kernel respectivamente.

### **Requerimientos**

shutdown DEBE existir y residir en /sbin.

Opciones específicas

Los siguientes archivos deben residir en sbin en caso de que sus respectivos subsistemas se encuentren instalados.

fastboot

- fasthalt
- fdisk
- fsck
- fsck.\*
- getty
- halt
- ifconfig
- init

· mkfs

- mkfs.\*
- mkswap
- reboot
- route
- swapon
- swapoff
- update
- /tmp

Contiene archivos temporales, creados por programas o usuarios mientras realizan otras funciones.

## **/usr**

Esta jerarquía es la segunda en importancia y en nivel de complejidad, contiene datos `compatibles` y de solo lectura, los paquetes de software muy grandes no deben usar un subdirectorio directamente sobre /usr como parecen creer algunos malos programadores/empaquetadores.

### **Requerimientos**

Los siguientes directorios o links a los mismos deben existir y residir en /usr.

.bin

- include
- lib
- local
- sbin
- share

### **Opciones Específicas**

Los siguientes directorios pueden existir y DEBEN residir en /usr en caso de estar instalados sus subsistemas asociados:

- X11R6 (contiene su propia jerarquía, no discutida en este documento)

- games
- lib
- src

### **Opciones para compatibilidad**

Los siguientes enlaces simbólicos pueden existir para guardar la compatibilidad con sistemas mas antiguos, hasta que todos aprendan a usar /var

- /usr/spool → /var/spool
- /usr/tmp → /var/tmp
- /usr/spool/locks → /var/lock

### **Opciones específicas para /usr/bin**

/usr/bin contiene binarios que no son necesarios cuando el sistema esta operando en modo `monousario` single user mode, los siguientes directorios o links a los mismos pueden existir y DEBEN residir en /usr/bin si sus subsistemas correspondientes se encuentran instalados.

- mh comandos para el sistema manejador de mails.
- /usr/bin/X11

Debe ser un link a /usr/X11R6/bin en caso de que este ultimo este instalado .

Los siguientes archivos o links a los mismos pueden existir y en ese caso DEBEN residir en /usr/bin si sus correspondientes subsistemas se encuentran instalados.

- perl
- python
- tclsh
- wish
- expect

### **La jerarquía /usr/local**

La jerarquía /usr/local debe ser usada por el administrador de sistemas cuando instala software localmente, debe estar protegida de manera que no sea afectada por upgrades al software de sistema, puede ser usada para compartir programas y/o datos con otros host. Requerimientos

Los siguientes directorios o links a los mismos deben encontrarse en /usr/local.

- .bin
- games
- include
- lib
- man
- sbin
- share
- src

Opciones específicas en caso de existir /usr/lib y/o /lib sus equivalentes deben existir también en /usr/local

### **/usr/sbin**

Aquí deben estar los binarios no esenciales que solo usará el administrador de sistemas /usr/share. Destinado a almacenar datos que son independientes de la arquitectura del host ojo, solo lectura, diseñado de manera de ser compartido con otros host de la red en caso de existir esta necesidad y/o posibilidad.

Se recomienda que cualquier programa que necesite guardar datos que no son modificables, lo haga bajo subdirectorios de /usr/share o en /usr/local/share si esta instalado localmente.

Los datos de juegos y/o programas educativos en /usr/share/games deben ser datos estáticos, para cualquier otro tipo (puntuaciones, etapas, etc ) se debe usar /var/games. Los siguientes directorios o links a los mismos deben encontrarse en /usr/share

- man
- misc

### **Opciones específicas**

Los siguientes directorios pueden existir y DEBEN residir en /usr/share en caso que sus subsistemas específicos se encuentren instalados:

- dict
- docs
- games
- info
- locale
- nls
- sgml
- terminfo
- tmac
- zoneinfo
- /var

Contiene solo datos de carácter variable, en esta categoría están especialmente incluidos los archivos y directorios de `spooling` (mail, impresión) y los archivos y directorios de `logging` (mensajes del kernel, mensajes de los servidores, etc).

La naturaleza de algunos archivos y directorios de var los hacen `no compartibles`, por ejemplo: /var/lock

- /var/log
- /var/run

### **Otros pueden ser perfectamente compartidos; por nombrar algunos:**

- /var/mail
- /var/cache/man
- /var/cache/fonts
- /var/spool/news

Se recomienda a las aplicaciones no crear directorios directamente sobre /var, una posible excepción a esta regla podrían constituirlos los archivos que tienen implicaciones para todo el sistema (seguirá siendo en todo caso, una violación a los estándares de no existir previa consulta y acuerdo con las listas de correo del FHS)

*Los siguientes directorios o links, los mismos deben existir en /var*

- cache
- lib
- local
- lock
- log
- opt
- run
- spool
- tmp

Se debe hacer notar que en /var existen directorios `reservados` que no deben ser usados por ninguna nueva aplicación pues generaría problemas con programas `históricos` o de práctica local, estos son:

- backups
- cron
- msgs
- preserve

Los siguientes directorios o links a los mismos pueden existir y DEBEN residir en /var siempre que sus subsistemas correspondientes se encuentren instalados:

- account
- crash
- games
- mail
- yp

Los siguientes deben estar presentes en un sistema solo si la orientación del mismo lo requiere provisto que cada uno contenga o apunte a una dirección que contenga los datos asociados a los mismos

/root

Directorio `home` para el administrador del sistema.

/home

Debe contener los directorios home de los usuarios normales del sistema /lib

Formato alternativo para librerías compartidas esenciales, en sistemas que soporten más de un formato de binarios, requiriendo por ende, librerías separadas.

El estandar discutido en este breve documento es aplicable a toda la gama de sistemas basados en unix actualmente en desarrollo, pues bien a pesar de ser GNU/Linux uno de los más populares, no es el único y el estandar necesita adaptar algunas de sus reglas para satisfacer a cada una de las variantes significativas, he aquí entonces las excepciones aplicables a GNU/Linux.

### **Directorio raíz (/)**

El estandar recomienda a los usuarios y desarrolladores que, de existir un kernel en el directorio raíz, este tome alguno de estos dos nombres:

- vmlinux
- vmlinuz

*Estos nombres están siendo actualmente utilizados en los paquetes fuentes del kernels más recientes.*

*/bin binarios de los comandos de uso esencial*

*Los sistemas GNU/Linux que lo requieran, deben poner el programa (comando, binario) `setserial` en este directorio.*

/dev

Archivos de dispositivo.

Todos los archivos de dispositivo y archivos especiales de /dev deben, en un sistema GNU/Linux, cumplir con lo establecido en el documento, "Linux Allocated Devices" (Linux, dispositivos reservados), mantenido por H. Peter Anvin esta simple regla también es aplicable a los enlaces simbólicos que puedan residir en este directorio.

/etc

Archivos de configuración del host

Se permite a los sistemas GNU/Linux que lo requieran, ubicar en esta carpeta al archivo de configuración del gestor de arranque `lilo.conf` (lilo)

/proc

Sistema de archivos virtual de información de procesos y el kernel

El sistema de archivos /proc es la ubicación estandar de la información del sistema y los procesos en ejecución en GNU/Linux, se recomienda su uso a desarrolladores y usuarios por sobre /proc/kmem y otros métodos.

*/sbin Binarios `esenciales` de sistema .*

Los sistemas GNU/Linux que lo requieran pueden ubicar los siguientes binarios extras en este directorio.

- badblocks

- *dumpe2fs*
- *e2fsck*
- mke2fs*
- *mklost+found*
- *tune2fs*
- *lilo* (*map installer*)
- *ldconfig*
- *sln*
- *ssync*
- *ctrlaltdel*
- *kbdrate*

*/usr/include* Archivos de cabecera incluidos por programas escritos en C

Los siguientes enlaces simbólicos deben estar presentes en */usr/include* en caso de sistemas que no están basados en glibc, cuando hay un compilador de C o C++ instalado.

- */usr/include/asm* → */usr/src/linux/include/asm-*
- */usr/include/linux* → */usr/src/linux/include/linux*

*/usr/src* Archivos fuentes

En un sistema basado en glibc no existen recomendaciones especiales para este directorio. En sistemas basados en la libc de Linux anterior a glibc, se deben seguir estos simples principios:

- Se exige que ``Linux`` sea el subdirectorio que almacene las fuentes del sistema operativo
- Cuando esté instalado un compilador de C o C++, pero las fuentes de Linux no lo estén, se requiere existan los archivos ``include`` del código fuente del kernel en uso:

**Actividad 5 Tema:** Particiones necesarias para la instalación de un sistema operativo de GNU/Linux

**Sumario:** Particiones necesarias para la instalación de un sistema operativo de GNU/Linux: boot, home, swap, root. Instalación de una versión de la distribución objeto de estudio. (Debian o u otra). Selección del paquete a instalar. Personalización de la instalación según las necesidades del usuario.

**Objetivo:** particionar el disco para la instalación de la distribución seleccionada, aplicando los procedimientos manual o automático, empleando los discos de instalación.

### Desarrollo

Particiones para instalar linux

Si hay un tema que pueda decirse que es el más preguntado en los foros sobre linux, sin duda es éste: ¿Que particiones necesita linux? ¿que tamaños tienen que tener las particiones?

La respuesta es fácil y normalmente la respuesta suele estar en esos mismos foros. Pero el problema de los foros es que la información está desperdigada.

En Linux (GNU/Linux) necesitamos al menos una partición. Cada sistema operativo usa su sistema de archivos, así que para instalar Linux necesitamos una partición exclusiva para él. Suele ser de tipo ext3 o ReiserFS, aunque en un futuro próximo se impondrá ext4.

Realmente siempre usamos al menos dos particiones, por una simple cuestión de rendimiento. Esa segunda partición es la de intercambio. En otros sistemas operativos se

usa un fichero para la memoria virtual (la memoria que se usa cuando la RAM se acaba, simplificando mucho), pero crear el fichero con una forma de asignación totalmente distinta a la memoria, y sobre la marcha, produce fragmentación (sobre todo en el sistema operativo más usado), y todo ello se traduce en un menor rendimiento. En linux usaremos una partición específica del tipo swap para intercambio (aunque también puede usarse un fichero), con un rendimiento óptimo. El tamaño generalmente ha sido del doble de la memoria física (la RAM), pero a partir de 512 MB se considera que es un desperdicio de espacio. Incluso se dice que no debe ser de más de ese tamaño. Hay razones para considerar que la regla debe seguir así, por las características de hibernación y suspensión, que pueden usar el intercambio para guardar la memoria al desactivarla, pero normalmente con usar el mismo tamaño que la RAM es suficiente incluso usando dichas características.

En general es recomendable usar tres particiones por cuestiones de seguridad y fiabilidad. El crear más particiones lo que significa es que separamos partes del sistema (directorios) en particiones, por lo que serán independientes, y en caso de algún error (comúnmente humano) o saturación, tendremos lo demás a salvo. Entonces tendremos lo siguiente:

Partición	del	sistema	/
Partición	de	intercambio	
Partición	de	los usuarios	/home

¿Que tamaños hay que dar a esas particiones? La del sistema es raro que supere los 6Gb, así que éste es un buen tamaño para usar. En caso de instalar aplicaciones propietarias (por ejemplo, juegos comerciales) en sistemas con varios usuarios, posiblemente éstas se instalen en la partición del sistema, así que quizá sea mejor usar 15Gb para no quedarse corto. De todas maneras éstas pueden instalarse en la partición de los usuarios, que veremos ahora. La partición de los usuarios es la partición que contendrá los directorios de los usuarios, ( /home/el64 ) por tanto todos sus datos, el contenido de su escritorio, sus configuraciones, etc. Esta es la partición que mayor tamaño debe tener, sobre todo si almacenamos películas (por ejemplo, extraídas de DVD). Ésta ocupará el resto del disco.

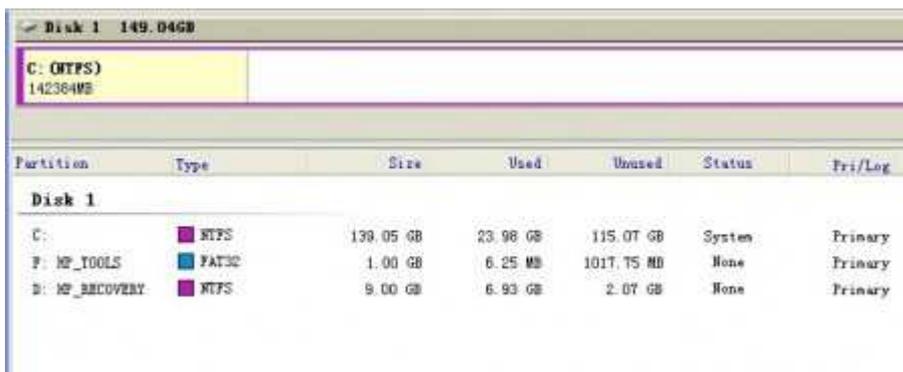
Pueden crearse aun más particiones. La partición extra más común, al menos hasta hace pocos años "gracias" a lilo, hoy en desuso, es la del arranque ( /boot). Contiene el núcleo del sistema (linux) y los archivos de configuración de éste y del cargador grub. Un tamaño de unos simples 20 megas es suficiente. Si le das 80 tampoco pasa nada. Entonces una configuración general podría ser esta (exactamente por este orden):

Arranque	/boot	(50Mb)
Sistema	/	(>5Gb)
Intercambio	(= memoria RAM o hasta el doble de ésta)	
Usuarios	/home (resto del disco)	

Si tienes un servidor, tienes alguna necesidad especial, o simplemente eres un maniático, puedes querer crear aun más particiones. Por ejemplo para /usr, directorio donde se encuentran casi todos los programas, archivos compartibles(pero no modificables) entre varios servidores, lo que llega a ser casi todo el sistema. Para el caso de un servidor es recomendable crear una partición para /var, directorio que varía su tamaño enormemente en ese entorno; el crear esta partición protegería a las demás de quedarse sin espacio.

Un caso más relacionado con rendimiento puro (sobre todo para profesionales del vídeo) podría ser crear una partición con un sistema de ficheros que tuviera un mejor rendimiento para el caso de almacenaje de ficheros grandes, como el XFS, dado que ext3, aun no teniendo ningún tipo de problema con estos ficheros, está orientado a tener mayor velocidad con ficheros pequeños. Podría ser algo así como /home/el64/DVD, o en cualquier otro lugar. Las particiones se pueden montar como se quiera, siempre con respeto a la estructura de directorios de linux.

Así tenía el disco duro antes de comenzar la partición:



Partition	Type	Size	Used	Unused	Status	Pri/Log
<b>Disk 1</b>						
C:	NTFS	139.05 GB	23.98 GB	115.07 GB	System	Primary
F: MP_TOOLS	FAT32	1.00 GB	6.25 MB	1017.75 MB	None	Primary
D: MP_RECOVERY	NTFS	9.00 GB	6.93 GB	2.07 GB	None	Primary

Así pues redimensiono para dejar espacio para Linux, con el EASEUS es tan sencillo como reducir la partición C: haciendo clic en el bloque rectangular que representa esa partición y desde el borde derecho arrastrarlo hasta dejarlo a nuestro gusto.

El resto de particiones F: y D: simplemente arrastro sus bloques con el puntero del ratón justo al lado derecho del bloque que representa la partición C:



**Imagen ampliada y detallada.**

Así aparecerá en fondo gris el espacio libre, que dedicaremos a instalar Linux.

Una vez conformes con el resultado del espacio asignado a cada partición de Windows Vista, le damos al botón "Apply". Nos preguntará si queremos reiniciar para aplicar los cambios. Aceptamos.

Una vez que se reinicie, el particionador EASEUS comenzará a dimensionar y a mover las particiones D y F al sitio que elegimos (continuación de la partición C tal y como estaban). Se reiniciará el sistema.

Cuando reinicie Windows nos aparecerá una pantalla con letras blancas y fondo negro, pondrá varios mensajes cómo que efectivamente ha habido cambios en el sistema, con la flecha del cursor nos movemos y elegimos la opción "Iniciar Windows normalmente".

Y ya tendremos todas las particiones con sus respectivos archivos igual que antes, pero con menos espacio para C.

Ahora no nos queda más que elegir una distribución de linux para instalar, mientras lo medito, será una distribución que por esta vez no será Fedora y como de costumbre dejaré una partición para mis archivos personales en /home.

Sin duda un proceso muy sencillo. Para cuándo instale un CD, DVD o pendrive con la ISO de linux, tan solo tendré que elegir:

-La opción de ocupar el espacio libre del disco. El proceso sería automatizado por la distribución de linux que pretendamos instalar.

-Personalizar la partición. Requerirá más atención por parte del usuario.

### ***Compilación e instalación***

La compilación e instalación del núcleo tiene un conjunto de pasos que son siempre los mismos, y que haremos de forma mecánica una vez que los hayamos seguidos dos o tres veces.

Antes de configurar el núcleo recuerda que hay que ejecutar "make mrproper" para limpiar posibles "restos" (ficheros .o y dependencias) de compilaciones anteriores.

Con el núcleo configurado a nuestro gusto, ejecutamos dentro del directorio donde están las fuentes el comando: "make depend".

Este comando actualiza las dependencias entre las diferentes partes, de forma que el orden de compilación sea correcto.

Con el comando "make zImage" es con el que se lanza la compilación. Ante él se pondrá el compilador gcc a compilar todas las partes del núcleo, un proceso que puede tardar desde varias horas (386 con 4 Mb de RAM) hasta 5 minutos (Pentium II o K6-2 con 32 Mb de RAM).

Una vez terminado este proceso tenemos ya la imagen de núcleo preparada para ser cargada y controlar el sistema. Dicha imagen se deja tras la compilación en el fichero "/usr/src/kernel-source-2.x.x/arch/i386/boot/zImage", si la arquitectura es Intel.

Dicha imagen está comprimida por lo que verás que su tamaño es pequeño, entre 400 y 500 Kb si no has metido muchas opciones dentro del núcleo. Si el núcleo generado supera estas cantidades fallará la generación del nuevo núcleo. Debes utilizar make "bzImage" en esos casos.

Una vez generada la imagen se han de generar los módulos, aquellas partes de l núcleo que cargaremos cuando nos hagan falta. Para ello hay que ejecutar las ordenes: "make modules", "make modules\_install".

Estas órdenes compilan los módulos y los dejan en el directorio adecuado para sus cargas.

Para que nuestro ordenador utilice este nuevo núcleo, en el caso de utilizar LILO, debemos copiar la nueva imagen del núcleo al directorio donde guardemos los núcleos, normalmente la imagen del núcleo será /boot/zImage, y añadimos una nueva opción al fichero "/etc/lilo.conf" para que sepa como arrancar este nuevo núcleo. Tras ello ejecutamos "lilo" para que se actualice el arranque del ordenador y reiniciamos el ordenador con "reboot". Al arrancar, elegimos dentro del LILO el nuevo núcleo y con ello, nuestro GNU/Linux utilizará nuestro nuevo núcleo.

Una vez arrancado el sistema, si quieres insertar dentro del núcleo uno de los módulos que compilaste, basta con ejecutar "insmod " y el núcleo ya podrá trabajar como si hubieras compilado soporte para ese módulo dentro del propio núcleo.

### ***Evolución del núcleo***

El núcleo de Linux está en constante evolución y hoy por hoy, no existe ningún otro sistema operativo que se adapte, actualice y corrija como Linux.

A finales de Enero de 1999 apareció la versión 2.2 del núcleo, un acontecimiento muy importante dentro de la comunidad Linux, y que lleva a Linux a soportar nuevas tecnologías como Video, USB ...

Dentro del presente artículo no se ha tocado en profundidad todas las mejoras que introduce este nuevo núcleo, pero desde aquí os damos la URL a un excelente artículo en el que se describe este nuevo núcleo: <http://www.openresources.com/>

EL núcleo de los sistemas operativos constituye la parte más compleja de los mismos, al tener controlar el uso de todos los recursos del sistema.

Linux posee uno de los núcleos mejor desarrollados dentro de los sistemas operativos. Su eficiencia y robustez han logrado llevar a Linux a obtener una cuota importante dentro del mercado de los servidores, y a ser el sistema preferido dentro del mundo de los desarrolladores.

Su constante evolución y la rapidez con la que solucionan los problemas que se detectan están logrando que los sistemas Unix vuelvan a recuperar terreno frente al pujante Windows NT.

Su desarrollo abierto y la disponibilidad de las fuentes hacen que Linux sea el sistema ideal para aprender, por lo que su implantación dentro de las Universidades está siendo masiva.

Es una realidad el que el 95% de PC's del mundo tienen instalada alguna versión de Windows, lo que hace casi seguro que virtualmente todos los usuarios finales y un enorme porcentaje de gente encargada de desarrollo y administración de redes sólo conoce este sistema operativo, lo que cierra las posibilidades de poder comenzar de lleno en cualquier otro, ya sea MacOS, Solaris, FreeBSD o GNU/Linux. Conclusión: casi seguro que comenzaremos con una computadora con sistema operativo dual; de hecho, se considera que la gran mayoría de usuarios de GNU/Linux comenzaron de esta forma y muchos, por cuestiones de trabajo o costumbre (de los hermanos que utilizan la PC, claro), lo mantienen a lo largo de la vida de su computadora.

### **Material para comenzar**

Si debemos tener una lista de requerimientos previos, es la siguiente:

- Respaldar nuestra información y si es necesario, los programas de nuestra computadora.
- Verificar la compatibilidad del hardware con Linux.
- Chequear el respaldo de nuestra información.

- Tener a la mano disquetes nuevos y los CD's de la distro (distribución de Linux) de nuestra preferencia y de alguna aplicación que particione discos duros. Si el disco duro es nuevo o lo acabamos de formatear, también necesitaremos el disco de instalación de Windows.
- Volver a chequear el respaldo de nuestra información.

### **El primer paso: Windows en la PC**

Si nuestro disco duro está limpio, lo debemos particionar con fdisk, mismo que se puede obtener tanto en los discos de arranque de Windows o en la distro de Linux. Debemos crear al menos dos particiones para cada sistema operativo, con el tamaño suficiente para que tanto el SO como sus aplicaciones y nuestra información quepan; recuerden que conforme pasa el tiempo se va necesitando más espacio, así que sean inteligentes, una partición de 5 GB para cada uno será suficiente para asegurarles un futuro con pocas preocupaciones, a menos a que sean unos adictos a las descargas de software y MP3. Particiones más pequeñas son posibles (3 GB o menos) pero deben fijarse muy bien cuando instalen o actualicen el sistema. La primera partición se recomienda para Windows, mismo que debe instalarse primeramente.

En caso que Windows ya esté instalado, entonces chequeamos cuánto espacio está ocupando y ejecutamos algún particionador de nuestra confianza ([Partition Magic](#) es considerado de los mejores), donde le asignamos un "apartado" a Linux.

Debemos recordar la limitación de muchos BIOS de no poder iniciar un sistema operativo después del cilindro 1024, por lo que si no es posible hacerlo, especialmente con un Windows ya muy crecido, podemos levantar Linux con un disquete de arranque.

### **Segundo paso: Linux en la PC**

Luego de tener Windows en la computadora (no afecta si se configuran todos los parámetros de red y dispositivos o no), se puede comenzar a instalar Linux. El primer paso es obtener un disco de inicio de la instalación de la distro (en caso que nuestra PC no pueda iniciar desde el CD-ROM), que generamos de la siguiente forma:

1. Insertamos un disquete nuevo en la unidad correspondiente.
2. Insertamos el primer CD de nuestra distro.
3. Buscamos en el CD la carpeta "dosutils" (regularmente en su raíz). Se recomienda leer el archivo "README" para tener un poco más de info.
4. Ejecutar el programa "rawrite.exe".
5. A continuación se le preguntará por la "imagen" que se utilizará, teclee: D:\images\boot.img (o la unidad del CD-ROM).
6. En seguida se le preguntará por la unidad en donde se instalará esta "imagen". En este caso nos referimos al floppy de 3½ pulgadas, así que teclee "a" o bien "b", según corresponda.
7. Después de algunos segundos, el disquete que usted introdujo, estará listo.

Una vez creado el disco, reiniciamos el equipo de forma normal, con lo que pasamos a la instalación propiamente; nos podemos apoyar, dependiendo de la distribución, en alguno de los siguientes documentos.

- [Red Hat 7.2 \(HTML - PDF\)](#)
- [Mandrake 8.1 \(HTML\)](#)
- [Slackwave \(HTML\)](#)
- [Debian \(HTML\)](#)

En un momento determinado del procedimiento, se nos preguntará si deseamos instalar **Lilo** (Linux Loader) o **Grub** (este puede no existir en ciertas distribuciones o versiones) para iniciar Linux. Debemos verificar que se reconozca la partición de Windows. Especificamos que se instaló cualquiera de los dos en hda1 (unidad C en Windows® y MS-DOS®) o en la partición donde se encuentra Linux®. Este pequeño programa permite

arrancar distintos sistemas operativos en una misma PC si los hubiese. De forma predeterminada, si usted tiene instalado Windows® o MS-DOS® en la unidad C del disco duro, al reiniciar el sistema podrá acceder a uno de estos sistemas operativos tecleando "dos" en el prompt de Lilo o a Linux® con solo presionar la tecla ENTER. Si por alguna razón desea desinstalar Lilo del disco duro, utilice fdisk de MS-DOS® o Windows® y utilice el siguiente comando en el símbolo de sistema de MS-DOS®:

```
fdisk /mbr
```

Si la instalación de Linux no queda antes del cilindro 1024, será recomendable que la inicie mediante un disquete de arranque, mismo que podrá crear al terminar la instalación.

### **Inicio dual con Windows NT/2000/XP**

Las versiones de Windows 95/98/Me pueden ser inicializadas con el procedimiento anterior, pero para las versiones de windows a 32 bits, como Windows NT/2000/XP, es necesario utilizar el siguiente procedimiento:

1. Instalamos Windows en una partición como lo vimos anteriormente.
2. Instalamos Linux como lo vimos anteriormente.
3. Inician Linux desde un disquete de arranque.
4. En el prompt tecleamos "cp /dosc/bootsect.dos /dosc/bootsect.lnx".
5. Hacemos una copia de nuestro kernel en /dosc/linux/vmlinux. El kernel lo encuentran en la partición /boot.
6. Utilizando un editor de texto (vi, emacs o pine), abren el archivo "/dosc/boot.ini" y lo modifican para que se vea así:
  - 7.
  8. [boot loader]
  9. timeout=30
  10. default=c:\bootsect.lnx
  11. [operating systems]
  12. c:\bootsect.lnx="Linux"
  13. multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Workstation
  14. Version 4.00"
  15. multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Workstation
  16. Version 4.00 [VGA mode]" /basevideo /sos
  17. C:\="Microsoft Windows"
  - 18.
19. De la misma forma, hay que modificar el archivo /etc/lilo.conf de la siguiente forma
  20. boot=/dosc/bootsect.lnx
  21. map=/boot/map.lnx
  22. install=/boot/boot.b
  23. image=/dosc/linux/vmlinux
  24. label=linux
  25. root=/dev/hdb2
  26. read-only
  - 27.
28. Salve los cambios de lilo.conf y desde el prompt corra "lilo" (sin comillas) para que se tomen los cambios en el arranque.
29. Cuando se reinicie la máquina, el arranque de NT deberá presentar la entrada "linux", la que al ser seleccionada, deberá iniciar el arranque de Linux.

**Actividad 6. Tema:** Los ambientes de trabajo de un sistema GNU/Linux.

**Sumario:** Interacción con los ambientes de trabajo de un sistema GNU/Linux (modo consola o de comandos y modo gráfico).

Concepto de comando en un sistema operativo. Comandos básicos para el modo consola: direccionar, copiar, mover, renombrar, borrar, crear, modificar directorios y ficheros. Ayuda del sistema para el trabajo con comandos.

**Trabajo independiente:** Comando MC y MAN.

**Objetivo:** Caracterizar los ambientes de trabajo del sistema operativo GNU/Linux, el modo consola o entorno texto y entorno gráfico GENOME y KDE, así como los comandos más generales aplicados en el modo texto.

## Desarrollo

### Entorno de escritorio

Un entorno de escritorio (en inglés, Desktop Environment) es un conjunto de software para ofrecer al usuario de una computadora una interacción amigable y cómoda.

El software es una solución completa de interfaz gráfica de usuario o GUI, ofrece iconos, barras de herramientas, programas e integración entre aplicaciones con habilidades como, arrastrar y soltar.

En general cada entorno de escritorio se distingue por su aspecto y comportamiento particulares, aunque algunos tienden a imitar características de escritorios ya existentes.

El primer entorno moderno de escritorio que se comercializó fue desarrollado por Xerox en los años 1980. Actualmente el entorno más conocido es el ofrecido por la familia Windows aunque existen otros como los de Macintosh (Classic y Cocoa) y de código abierto (o software libre) como GNOME, KDE, CDE o Xfce.

En algunos contextos es muy útil el disponer de una forma gráfica de interacción con el Sistema Operativo en que se trabaje. De hecho, todos los Sistemas Operativos modernos cuentan con esta utilidad y Linux no es la excepción. Las interfaces gráficas para Linux y para todos los sistemas tipo Unix en general son conocidas como ambientes X Windows (observe que no lleva ``s" a diferencia del Sistema Operativo Windows). Para el caso de Linux se utiliza una implementación de X Windows con código abierto (open source) denominada XFree86 que también está disponible para otros sistemas tipo Unix como los BSD (FreeBSD, OpenBSD y NetBSD), Solaris, etc.

En Linux todo se puede lograr con comandos, no así con las aplicaciones gráficas que pueden no ofrecer todas las opciones posibles en aras de ser más simples o porque no siempre se desarrollan a la par de los comandos y programas con interfaz textual. Es común encontrar muchas aplicaciones gráficas con la misma funcionalidad pero con diferente diseño y apariencia.

La interfaz principal entre el usuario y un servidor X la define el gestor o manejador de ventanas (Window Manager). Este se encarga de definir y manipular la apariencia de las ventanas: bordes, menues, botones, barras de desplazamiento, de herramientas y de estado, entre otras múltiples funciones.

### Comando

Medio por el cual se le ordena una acción determinada al sistema operativo a través de un intérprete de **comandos**, tal y como puede ser Bash.

Entorno de Escritorio K , es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix / Linux .

De acuerdo con su página web , «KDE es un entorno gráfico contemporáneo para estaciones de trabajo Unix . KDE llena la necesidad de un escritorio amigable para estaciones de trabajo Unix , **similar a los escritorios de MacOS o Windows**

La «K», originariamente, representaba la palabra « Kool », pero su significado fue abandonado más tarde. Actualmente significa simplemente «K», la letra inmediatamente anterior a la «L» (inicial de Linux) en el alfabeto .

La mascota del proyecto es un pequeño dragón llamado Konqi .

### Historia

El proyecto fue iniciado en octubre de 1996 por el programador alemán Matthias Ettrich , quien buscaba crear una interfaz gráfica unificada para sistemas Unix . En sus inicios imitó a CDE ( Common Desktop Environment ), un entorno de escritorio utilizado por varios Unix .

Dos factores llevaron a la creación del proyecto alternativo GNOME en 1997 : la elección de la biblioteca Qt , que por aquel entonces poseía una licencia incompatible con la GPL de GNU , aunque libre: la QPL y en menor medida la importancia del lenguaje C++ para el desarrollo de KDE. La rivalidad actual entre ambos proyectos se considera beneficiosa generalmente y existe, de hecho, una constante cooperación e inspiración mutua.

- KDE 1: Al año siguiente , se publicó KDE 1.0. Esta versión contenía un panel (barra de tareas y lanzador de aplicaciones), un escritorio sobre el cual dejar iconos, un administrador de archivos ( Kfm ) y un gran número de utilidades.
- KDE 2: KDE 2.0, lanzado en el año 2000 , fue reescrito casi por completo. Esta versión incluía Konqueror , un navegador Web y gestor de archivos , además de muchas nuevas tecnologías con el objetivo de mejorar la integración entre aplicaciones. En esta versión mejoró parcialmente el aspecto visual.
- KDE 3: KDE 3.0 fue publicado en el año 2002 , y es la evolución de KDE 2. El aspecto de la interfaz no varió hasta KDE 3.1, en el que consta una importante mejora referente al tema visual: Keramik es incluido como nuevo tema por omisión junto con el conjunto de iconos Crystal GT de y el antialiasado de fuentes . En KDE 3.2 Crystal GT fue reemplazado por Crystal SVG. En KDE 3.4 Keramik fue reemplazado por Plastik .

GNOME es un acrónimo de 'GNU Network Object Model Environment', entorno de trabajo en red orientado a objetos, por lo que forma parte del más amplio proyecto GNU.

Es un entorno gráfico, escritorio de trabajo, amigable que permite a los usuarios usar y configurar sus ordenadores de una forma sencilla. Incluye un panel para arrancar aplicaciones y presentar el estado de funcionamiento, un escritorio donde se pueden situar los datos y las aplicaciones, un conjunto estándar de aplicaciones y herramientas de escritorio, y un conjunto de convenciones que facilitan la operación y consistencia de las aplicaciones entre sí.

Los usuarios de otros sistemas operativos y entornos se sentirán como en casa cuando usen el potente entorno gráfico que proporciona GNOME. Es en su totalidad código abierto software libre, con el código fuente disponible libremente y desarrollado por cientos de programadores en todo el mundo.

Si quiere saber más sobre el proyecto GNOME puede visitar la página web de [GNOME](#).

Este entorno tiene una serie de ventajas para los usuarios, facilita el uso y la configuración de aplicaciones sin interfaces puramente textuales (console).

Es altamente configurable, permitiéndole ajustar su escritorio con el aspecto que desee. El gestor de sesiones de GNOME recuerda la configuración previa, de manera que una vez que haya configurado las cosas a su gusto, las mantendrá así.

Soporta muchos idiomas y puede añadir más sin cambiar el software, soporta incluso varios protocolos de arrastrar y soltar (drag and drop D&D) para una máxima interoperabilidad con aplicaciones que no sean compatibles con GNOME.

Tiene también una serie de ventajas para los programadores que indirectamente benefician también a los usuarios. Los programadores no necesitan comprar una cara licencia de software para hacer compatibles sus aplicaciones comerciales.

De hecho, GNOME es independiente de cualquier compañía, ningún componente del interfaz está controlado únicamente por una compañía o tiene restringidas la modificación o redistribución.

Las aplicaciones GNOME pueden desarrollarse en una gran variedad de lenguajes de programación, por lo que el programador no se ve limitado a un solo lenguaje, usa la arquitectura universal de agentes de petición de objetos (CORBA) para permitir que los componentes software interactúen 'sin costuras', independientemente del lenguaje en el que estén implementados, o incluso de la máquina en la que se estén ejecutando. Finalmente, funciona en numerosos sistemas operativos de tipo Unix, incluido Linux.

### Comandos internos del shell (bash).

Comandos	Funciones
date -----	Da la hora y la fecha del sistema.
cal -my -----	Muestra el calendario.
clear -----	Borra la pantalla.
uname -a -----	Versión del kernel
hostname -----	Visualiza el nombre de la máquina.
who -a -H -----	Muestra información de los usuarios conectados
id -----	Muestra información del usuario actual, (grupos a los que pertenece,uid,gid)
startx -----	Arranca el entorno gráfico con el gestor especificado en el .initrc de tu home.
startx /usr/bin/X11/icewm	Arranca el entorno gráfico con el icewm (para el caso de que no fuera el predeterminado)
Ctrl+C -----	Salir de un programa que corre en consola, parando su ejecución.
Ctrl+Alt+F2 -----	Salir del entorno gráfico a un terminal. (Fnº según gettys activados)
./script -----	Ejecuta un script de shell.
java -jar fichero. --	Ejecuta un programa java.
base-config -----	Configura el sistema base.
reportbug -----	Programa en modo texto para enviar bugs a Debian
init 0 -----	Apaga la máquina
init 6 -----	Reinicia la máquina.
shutdown -t1 -h ----	Apaga la máquina
shutdown -t1 -r now	Reinicia la máquina
----	
su -----	Entrar a la sesión como root u otro usuario

su nom\_usuario ---- Estando como root entramos como otro usuario.  
passwd ----- Cambio de contraseña.

## Comandos para el arranque, salir de un programa, salir del entorno gráfico, apaga la máquina y reinicia la máquina

**startx** ----- Arranca el entorno gráfico con el gestor especificado en el .initrc de tu home.  
**startx /usr/bin/X11/icewm** ----- Arranca el entorno gráfico con el icewm (para el caso de que no fuera el predeterminado)  
**Ctrl+C** ----- Salir de un programa que corre en consola, parando su ejecución.  
**Ctrl+Alt+F2** ----- Salir del entorno gráfico a un terminal.(Fn<sup>0</sup> según gettys activados)  
**./script** ----- Ejecuta un script de shell.  
**java -jar fichero.jar** ----- Ejecuta un programa java.  
**base-config** ----- Configura el sistema base.  
**reportbug** ----- Programa en modo texto para enviar bugs a Debian  
**init 0** ----- Apaga la máquina.  
**init 6** ----- Reinicia la máquina.  
**shutdown -t1 -h now** -----Apaga la máquina .  
**shutdown -t1 -r now** -----Reinicia la máquina.  
**su** ----- Entrar a la sesión como root u otro usuario.  
su nom\_usuario ----- Estando como root entramos como otro usuario.  
passwd ----- Cambio de contraseña.

## Comandos de ayuda: man, help, info.

Algunos comandos en consola pueden servirnos de ayuda, veamos algunos ejemplos.

*Comando man:*

Nombre : **man**

Descripción : Interfaz de los manuales de referencia electrónicos.

Sintaxis : **man <un\_comando>**

Manual : **man man**

Ejemplo : **man time**

En el ejemplo anterior la salida sería luego de ejecutarlo: el manual del comando time. Es posible caminar dentro de él apretando la tecla "Enter". Se puede salir del manual apretando la tecla "q".

*Comando help:*

Nombre : **help**

Descripción : Interfaz de los manuales de referencia electrónicos.

Sintaxis : **<un\_comando> -- help**

Manual : **--help**

Ejemplo : **mkdir --help**

La salida para el ejemplo mostrado será:

Modo de empleo: **mkdir [OPCIÓN] DIRECTORIO...**

Crea los DIRECTORIO(s), si no existen ya.

-Z, --context=CONTEXT (SELinux) set security context to CONTEXT  
Los argumentos obligatorios para las opciones largas son también obligatorios para las opciones cortas.  
Crea el/los DIRECTORIO(s), si no existen ya.  
-m, --mode=MODE establece los permisos (como en chmod), en lugar de rwxrwxrwx - umask  
-p, --parents no hay error si existen, crea los directorios padres en caso necesario  
-v, --verbose muestra un mensaje por cada directorio creado  
--help muestra esta ayuda y finaliza  
--version informa de la versión y finaliza

#### Comando **info**:

Nombre : **info**

Descripción : Interfaz para leer información de los archivos.

Sintaxis : **info <un\_comando>**

Manual : **info info**

Ejemplo : **info time**

En el ejemplo anterior la salida sería luego de ejecutarlo: el manual del comando time. A diferencia de man es posible desplazarse dentro del manual utilizando las flechas de “arriba” y “abajo”. Se puede salir del manual apretando la tecla “q”.

### Comandos para la administración de usuarios. Adicionar y eliminar usuarios del sistema.

#### Adicionar

Nombre: **adduser**

Descripción: Comando para adicionar usuarios al sistema.

Sintaxis: **adduser <nombre\_usuario>**

Manual: **man adduser**

Ejemplo: adduser **Rudy**

La salida de este comando será un prompt donde se le preguntarán una serie de datos que deberá ir llenando hasta al final que será creado el usuario, finalmente para verificar que se ha creado el usuario podemos cerrar sesión para verificarlo o simplemente con el comando su <nombre\_del\_usuario\_creado> (**Rudy** para este caso) y entrando la contraseña ya podremos entrar y ver el nuevo usuario verificando así la correcta creación del mismo.

#### Eliminar

Nombre : **deluser**

Descripción : Comando para eliminar usuarios del sistema.

Sintaxis : **adduser <nombre\_usuario>**

Manual : **man adduser**

Ejemplo : adduser **Rudy**

Luego de aplicar este comando la información del usuario será eliminada, cuando nos referimos a la información del usuario queremos decir la entrada durante el proceso de creación del mismo así como que se le negará el acceso al sistema.

### Comandos para el trabajo con directorios: mkdir, rmdir, cat, more, less. **mkdir: crea directorio**

## Comando para crear un directorio.

· mkdir <ruta> directorio.

## Eliminar directorio.

### .rmdir: elimina directorios

Para borrar directorios se usará el comando rmdir, equivalente al rmdir de MS-DOS. El comando rmdir presenta el siguiente formato:

| rmdir [-ri ] directorio

Una condición para que el comando funcione correctamente es que los directorios a eliminar **estén vacíos**. Si no lo están, habrá que borrar los ficheros que contiene antes de borrar el directorio.

Para borrar todos los directorios ( vacíos ) que cuelgan de uno dado se utiliza la opción **-r**. Con la opción **-i** entramos en el modo interactivo, el que se nos pregunta antes de eliminar cada directorio.

### cat, more, less: examinar el contenido de un fichero.

Una vez que el usuario ya conoce cómo listar los archivos de un directorio, cómo moverse entre directorios y cómo crear y borrar directorios, lo siguiente que surge es la necesidad de examinar el contenido de los ficheros. Los comandos **cat, more y less** permiten hojear el contenido de un fichero, teniendo cada uno sus particularidades que se comentarán a continuación. El formato de los tres comandos es el siguiente:

- **cat fichero**
- **more fichero**
- **less fichero**

cat es el comando más simple, pues muestra el contenido de un fichero mostrándolo por pantalla y sin ningún tipo de pausa. Un caso especial se produce cuando se ejecuta el comando cat sin parámetros. Entonces el comando se queda esperando a que se introduzcan caracteres por pantalla, mostrándolos línea a línea hasta que pulsa Ctrl-D. more y less sí permiten hacer pausas durante la visualización de los datos. Para avanzar entre pantallas se pulsa cualquier tecla. La diferencia entre ambos consiste en que el comando less utiliza las teclas de flechas para poder avanzar y retroceder por el fichero, cualidad de la que carece el comando more.

Otros comandos para el trabajo con directorios

Comando/Sintaxis	Descripción	Ejemplos
cat <i>fich1</i> [... <i>fichN</i> ]	Concatena y muestra un archivos	cat /etc/passwd
	archivos	cat dict1 dict2 dict
cd [ <i>dir</i> ]	Cambia de directorio	cd /tmp
chmod <i>permisos fich</i>	Cambia los permisos de un archivo	chmod +x miscript
chown <i>usuario:grupo fich</i>	Cambia el dueño un archivo	chown nobody miscript
cp <i>fich1...fichN dir</i>	Copia archivos	cp foo foo.backup
diff [-e] <i>arch1 arch2</i>	Encuentra diferencia entre archivos	diff foo.c newfoo.c
du [-sabr] <i>fich</i>	Reporta el tamaño del directorio	du -s /home/
file <i>arch</i>	Muestra el tipo de un archivo	file arc_desconocido
find <i>dir test acción</i>	Encuentra archivos.	find . -name ``.bak" -print
grep [-cilmv] <i>expr archivos</i>	Busca patrones en archivos	grep mike /etc/passwd
head -count <i>fich</i>	Muestra el inicio de un archivo	head prog1.c
mkdir <i>dir</i>	Crea un directorio.	mkdir temp
mv <i>fich1 ...fichN dir</i>	Mueve un archivo(s) a un directorio	mv a.out prog1
mv <i>fich1 fich2</i>	Renombra un archivo.	mv .c prog_dir
less / more <i>fich(s)</i>	Visualiza página a página un archivo.	more muy_largo.c
	less acepta comandos vi.	less muy_largo.c

<code>ln [-s] fich acceso</code>	Crea un acceso directo a un archivo	<code>ln -s /users/mike/.profile .</code>
<code>ls</code>	Lista el contenido del directorio	<code>ls -l /usr/bin</code>
<code>pwd</code>	Muestra la ruta del directorio actual	<code>pwd</code>
<code>rm fich</code>	Borra un fichero.	<code>rm foo.c</code>
<code>rm -r dir</code>	Borra un todo un directorio	<code>rm -rf prog_dir</code>
<code>rmdir dir</code>	Borra un directorio vacío	<code>rmdir prog_dir</code>
<code>tail -count fich</code>	Muestra el final de un archivo	<code>tail prog1.c</code>
<code>vi fich</code>	Edita un archivo.	<code>vi .profile</code>

## Crear, borrar, copiar, renombrar y mover ficheros por la consola.

**cp -dpR fichero1 ruta\_fichero2** ----- Realiza una copia del fichero1 a ruta\_fichero2, cambiándole el nombre.

- **cp -dpR fichero1 /directorio** ----- Copia fichero1 a directorio, conservando fichero1 el nombre.
- **-R** ----- Copia un directorio recursivamente, salvo los ficheros especiales.
- **-p** ----- Copia preservando permisos, propietario, grupos y fechas.
- **-d** ----- Conserva los enlaces simbólicos como tales y preserva las relaciones de los duros.
- **-a** ----- Lo mismo que -dpR .

**mv ruta\_fichero1 ruta\_fichero2** ----- Mueve y/o renombra ficheros o directorios.

### Comandos para borrar ficheros y todo el directorio

**rm fich** ----Borra un fichero. -----**rm foo.c**

**rm -r dir**----- Borra todo un directorio----- **rm -rf prog\_dir**

**rmdir dir**----- Borra un directorio vacío----- **rmdir prog\_dir**

## Comandos para instalar y desinstalar dispositivos

### Puntos de Montajes.

Todos los dispositivos del sistema de archivos de Linux se colocan en: /media, para que puedan verse en el icono del escritorio "Mi PC". Sin embargo es correcto montarlo también en /mnt.

Así por ejemplo:

El floppy se monta en /media/floppy

El cdrom se monta en /media/cdrom

### Comandos mount – umount.

Ahora que se ha creado el sistema de archivos, puede montar la partición. Inicialmente, la misma estará vacía. El comando para montar sistemas de archivos es **mount**, y su sintaxis es la siguiente:

**mount [opciones] <-t tipo> [-o opciones\_de\_montado] <dispositivo>  
<punto\_de\_montaje>**

En este caso, queremos montar temporalmente nuestra partición sobre **/mnt** (o cualquier otro punto de montaje que Usted haya elegido – recuerde que debe existir); el comando para montar nuestra partición creada recientemente es el siguiente:

**\$ mount /dev/hdb1 /mnt**

Frecuentemente se presentan problemas con el sistema de archivo de los dispositivos, y se nos hace difícil montar un disquete o un flash que ha sido utilizada, o formateada, en Windows. Veamos el siguiente ejemplo:

**\$ mount -t ext2 /dev/hdb1 /mnt**

La opción **-t** se usa para especificar el tipo de sistema de archivos que se supone alberga la partición. Entre los sistemas de archivos que encontrará con mayor frecuencia, están ext2 (el sistema de archivos de GNU/Linux), vfat (para todas las particiones DOS/Windows: FAT 12, 16 o 32) e ISO9660 (sistema de archivos de CD-ROM). Si no especifica tipo alguno, **mount** intentará adivinar el sistema de archivos que alberga la partición leyendo el superbloque. Es poco común que falle en esta operación.

Ahora que se copiaron los archivos, podemos desmontar nuestra partición. Para hacerlo utilice el comando **umount**. La sintaxis es simple:

**umount <punto\_de\_montaje | dispositivo>**

Entonces, para desmontar nuestra partición, podemos ingresar:

**\$ umount /mnt**

o bien:

**\$ umount /dev/hdb1**

Montaje y desmontaje de unidades.

**Para realizar estas operaciones es necesario conocer la contraseña del root.**

root> mkfs.ext3 /dev/fd0 ----- formateamos el disquete.

root> mount -t ext3 /dev/fd0 /floppy ----- lo montamos en /floppy.

root> cp /ruta\_de\_acceso/stage1 /floppy ----- copiamos stage1

root> cp /ruta\_de\_acceso/stage2 /floppy ----- copiamos stage2

root> cp /boot/vmlinuz /dev/fd0 ----- si hay sitio se puede incluir el núcleo en el disquete.

root> umount /dev/fd0 ----- se desmonta floppy sin sacarlo.

root> grub ----- inicia la interfaz de comandos de GRUB

grub> root (fd0) ----- instala las herramientas de GRUB...

grub> setup (fd0) ----- ...en el MBR del disquete.

grub> quit ----- salimos.

-mount p.de montaje disco extraible ----- montaje del disco extraíble.

-mount -o rw -t ntfs /dev/had(#/home/(Windows) ----montaje de la torre de Windows en Linux

Montar equivale a crear un acceso desde un directorio a una unidad o dispositivo. No significa copiar sino establecer un enlace entre tal directorio y el dispositivo. Desmontarla es eliminar ese enlace. A los usuarios de ms-dos puede parecerles a lo que se realizaba con las unidades comprimidas dblspace o drivespace.

De forma predeterminada GNU-LINUX nos ofrece directorios en los que se montan dispositivos comúnmente utilizados, **floppy** en el que accedemos a las disqueteras y otro llamado **cdrom** para unidades ópticas. Esto no es común para todas las distribuciones.

Cada quien puede montar las cosas tal y como prefiera, lo más normal es que los dispositivos se monten dentro del directorio **/mnt**. El montaje, tanto de GNU-Linux en modo gráfico, como de windows, se hace automáticamente al intentar acceder a la unidad, pero en el modo texto de GNU-Linux se montan de forma prácticamente manual.

**Nota:** En este encuentro se aplicará la prueba de control parcial

**Guía 7. Tema:** Gestores de ventanas

**Sumario:** Gestores de ventanas: KDE y GNOME. Facilidades de cada uno. Selección del gestor que se utilizará.

Componentes del escritorio: paneles, iconos, fondo, etc.

Utilización de escritorios virtuales. Personalización de cada escritorio. Posibilidades del uso de escritorios múltiples en la personalización del sistema.

**Objetivo:** Caracterizar los gestores de ventana del sistema operativo Linux, sus componentes y posibilidades de utilización de escritorios múltiples.

### **Desarrollo**

#### **Entornos de escritorio y gestores de ventanas**

Una vez que un servidor X se esté ejecutando, las aplicaciones cliente X pueden conectarlo y crear una GUI para el usuario. Un rango de GUIs está disponible con Red Hat Enterprise Linux, desde el rudimentario *Administrador de pestañas de ventanas* hasta un entorno de escritorio altamente desarrollado, interactivo como *GNOME*, con el que la mayoría de los usuarios de Red Hat Enterprise Linux están familiarizados.

Para crear lo último, una GUI más avanzada, se deben conectar dos clases principales de aplicaciones clientes X al servidor X: un *entorno de escritorio* y un *gestor de ventanas*.

#### **Entornos de escritorio**

Un entorno de escritorio une diferentes clientes de X, los cuales cuando se usan juntos crean un ambiente de usuario gráfico común y una plataforma de desarrollo.

Los entornos de escritorio tienen características avanzadas las cuales permiten a los clientes X y a otros procesos en ejecución, comunicarse unos con otros a la vez que se permite a todas las aplicaciones escritas para funcionar en ese ambiente a que realicen tareas avanzadas, tales como operaciones de arrastrar y soltar.

Red Hat Enterprise Linux proporciona dos entornos de escritorio:

- *GNOME* — Es el entorno de escritorio por defecto en Red Hat Enterprise Linux basado en el conjunto de herramientas gráficas GTK+ 2.
- *KDE* — Un entorno de escritorio alternativo basado en el conjunto de herramientas gráficas Qt 3.

Ambos entornos GNOME y KDE tienen aplicaciones de productividad avanzadas, tales como procesadores de palabras, hojas de cálculo y navegadores Web así como herramientas para personalizar la apariencia de la GUI. Adicionalmente, si ambas bibliotecas están presentes, la GTK+ 2 y la Qt, las aplicaciones KDE pueden ejecutarse en GNOME y viceversa.

#### **Gestores de ventanas**

Los *gestores de ventanas* son programas clientes de X que son o parte del entorno de escritorio o, en algunos casos, independientes. Su propósito principal es controlar la forma en que las ventanas gráficas son posicionadas, redimensionadas o movidas. Los manejadores de ventanas controlan las barras de títulos, el comportamiento del foco, los vínculos del botón del ratón y teclas especificadas por el usuario.

Se incluyen cuatro gestores de ventanas con Red Hat Enterprise Linux:

- *kwin* — El gestor de ventanas *KWin* es el manejador por defecto para el entorno KDE. Es un manejador de ventanas eficiente que soporta temas personalizados.

- metacity — El gestor de ventanas *Metacity* es el manejador de ventanas por defecto del entorno GNOME. Es un manejador de ventanas simple y eficiente que también soporta temas personalizados.
- mwm — El gestor de ventanas *Motif*, es un gestor básico independiente. Puesto que está diseñado para ser un gestor que se ejecuta de forma independiente, no se debería utilizar en conjunto con los entornos de escritorios GNOME o KDE.
- twm — El minimalista *Administrador de pestañas de ventanas*, el cual proporciona el conjunto de herramientas más básicas de cualquier gestor de ventanas y puede ser usado bien sea de forma independiente o con un entorno de escritorio. Es instalado como parte de la versión X11R6.8.

Estos gestores de ventanas pueden ejecutarse sin los entornos de escritorio para obtener una mejor impresión de sus diferencias. Para hacer esto, teclee el comando `xinit -e <ruta-al-gestordeventanas>`, donde `<ruta-al-gestordeventanas>` es la ubicación del archivo binario de gestor de ventanas. El archivo binario puede ser encontrado escribiendo `which <nombre-de-gestordeventanas>`, donde `<nombre-de-gestordeventanas>` es el nombre del gestor de ventanas sobre el cual está preguntando.

El sistema gráfico en sistemas tipo Unix esta basado en componentes o unidades independientes, algo así como piezas sueltas que pueden acoplarse unas a otras, primero tenemos el servidor X (X Window System -Xorg Xfree86), después están los gestores o manejadores de ventanas (Window Managers--Icewm, Fluxbox, etc) y por último los escritorios (Desktop--KDE, Gnome; Xfce, etc)

Cada uno es independiente entre si, aunque siguen reglas especificadas y aprobadas (o sea que nada es arbitrario ni anárquico)

un gestor de ventanas lo único que hace es dibujar las ventanas y presentarlas en pantalla, definen el tamaño y la posición, así como los elementos en ellas (iconos para abrir, cerrar ampliar, etc.) y cual esta activa (lista para trabajar) entre otras cosas, que varían de gestor en gestor, y como lo único a lo que se dedican es básicamente a presentar las ventanas sin mayor adorno eso los hace ligeros

Un escritorio hace uso de un gestor de ventanas (ya sea propio o de terceros) para lo antes mencionado y además especifica el poder poner imágenes (iconos) en la pantalla (escritorio), eso entre otras cosas, como definir temas para cursores, menus emergentes en el escritorio, monitores de recursos mas llamativos, etc todo eso hace que el entorno grafico (escritorio) se vuelva pesado.

ahora la cosa se pone mejor aun, a un gestor de ventanas el cual no puede poner por si sólo iconos, le puedes agregar un manejador de iconos para convertirlo en un casi escritorio, digo casi porque para lograr algo como un KDE o Gnome se requieren de distintos componentes que llegarían a hacer igual de pesado el entorno de trabajo, por lo regular a un gestor de ventanas para mantenerlo ligero pero a la vez "fácil" de usar se le incluyen un manejador de iconos para tener algo más parecido a un escritorio sin llegar a ser tan pesado.

Es difícil entender la diferencia entre gestor y escritorio, la única forma más fácil y rápida

de entenderlo es usando cada uno de ellos, incluso entre los distintos gestores de ventanas hay diferencias en cuanto a su manejo y presentación, pero lo que tienen más en común es que para configurarlos se suele hacer a través de archivos de texto, en cambio en un escritorio todos los cambios los haces desde asistentes gráficos (otro elemento que los hace pesados)

Te recomiendo si quieres conocer más a fondo es empezar con un gestor de ventanas como Icewm, es muy fácil de usar y de configurar y de inmediato veras la diferencia entre un gestor y un escritorio, después puedes cambiar a Fluxbox, el cual es algo diferente a Icewm, desde mi punto de vista Icewm es más fácil de entender que Fluxbox, luego puedes continuar con windowmaker, el cual es completamente diferente a los otros dos.

Cuando veas la diferencia entre un gestor y otro podrás hacer más fácil el entendimiento de como funcionaria un escritorio como Gnome si le cambias su gestor predeterminado por cualquiera de los tres que te he mencionado (o algún otro), y te darás cuentas de las distintas combinaciones que puedes obtener y que cada uno trabajara de manera un poco distinta y ya dependerá de ti que forma de trabajar es mejor, así no estas atado a una sola forma de presentar la información y de manejarla.

En sistemas como Windows, donde todo esta integrado lo único que puedes modificar son las decoraciones, y el trabajo sigue siendo el mismo, en cambio en GNU/Linux puedes encontrar la forma más apropiada para trabajar con entornos gráficos, todo de acuerdo a tus gustos y necesidades, tanto ergonómicas, funcionales o de recursos de hardware.

### **Escritorio KDE (Escritorio Kubuntu):**

**Opinión:** Un escritorio con muchas opciones para su configuración , escritorio fiable, con mucho software para su entorno. Para mi el mejor y va por su versión 4 , muy Vistera pero bueno...

#### **Instalación de KDE:**

Simple. En la Consola escribe **"sudo apt-get install kubuntu-desktop"**.

Luego password y aceptan la descarga con "Y" ó "S". Instalará todas las dependencias

### **Escritorio GNOME (Ubuntu Desktop)**

**Opinión:** Algo rápido. Sencillo. No con tantos programas como los de KDE pero es una alternativa a los otros si quieren algo diferente al gñindows. (Aunque personalmente con unos cambios al KDE, no sentirán que están en el otro SO , que tanto miedo da!).

#### **Instalación de Gnome:**

También Sencillo. En la consola escribir: **"Sudo apt-get install ubuntu-desktop"**. y las mismas instrucciones de las otros escritorios.

**Componentes del escritorio: paneles, iconos, fondo, etc.**

**Guía 8. Tema:** Repositorios en INTERNET para la distribución objeto de estudio

**Sumario:** Aplicaciones generales para:

La navegación en redes: Firefox y Konquerour.

Clientes de correo: Evolution y Kmail.

Chateo en GNU/Linux: XChat

Mensajería Instantánea en GNU/Linux: AIM, Yahoo Mensajer y ICQ

**Objetivo:** Caracterizar las principales aplicaciones para la navegación en la red mediante los repositorios de Internet.

### Desarrollo

La palabra repositorio deriva de la palabra repositorium (latín) que significa alacena o armario, y bueno un repositorio no es más que eso un "armario" centralizado de información, en este caso de paquetes para una distribución....

una vez explicado esto procedo a explicar los beneficios de tener un repositorio local...

La utilidad de un repositorio local se encuentra cuando no se tiene una conexión rápida de Internet o esta conexión es lenta en ciertos horarios del día (léase al mediodía) o si se quiere recomendar una distribución específica a alguien q no tiene Internet y quiere instalar paquetes, que hacemos ante esta situación? bueno lo que tenemos que hacer primero es encontrar las direcciones de los repositorios de nuestra distro en Internet, para el caso de pclinuxos los repositorios son los siguientes:

\*<http://spout.ussg.indiana.edu/linux/pclinuxos/pclinuxos/apt/>

\*<http://ftp.heanet.ie/pub/pclinuxos/apt/>

\*<http://distro.ibiblio.org/pub/linux/distributions/texstar/pclinuxos/apt/>

\*<http://www.gtlib.gatech.edu/pub/metalab/distributions/texstar/pclinuxos/apt/>

\*<ftp://ftp.belnet.be/pub/mirror/pclinuxonline.com/apt/>

\*<ftp://ftp.ch.debian.org/mirror/pclinuxos/apt/>

\*<http://ftp.sh.cvut.cz/MIRRORS/pclinuxos/apt/>

\*<http://ftp.nl.freebsd.org/os/Linux/distr/texstar/pclinuxos/apt/>

\*<ftp://ftp.pbone.net/pub/pclinuxos/apt/>

\*<http://linuxstation.net/pub/pclinuxos/apt/>

\*<http://ftp.leg.uct.ac.za/pub/linux/pclinuxos/apt/>

\*<http://mirror.pacific.net.au/pub1/linux/texstar/pclinuxos/apt/>

\*<ftp://mirror.internode.on.net/pub/pclinuxos/apt/>

luego necesitamos alrededor de 40gb libres en nuestro disco, una conexión a Internet, y necesitaremos instalar los siguientes programas del repositorio de pclinuxos: rsync y createrepo

una vez que tenemos instalados todos los programas y verificamos el espacio en disco procedemos a usar rsync para descargar los archivos del repositorio que más nos guste.

### BAJANDO Y CONFIGURANDO EL REPOSITORIO

\*Creamos la carpeta repositorio en algún lugar por ejemplo en /mnt/datos.

\*Luego abrimos una consola y escribimos `rsync -av --progress --delete --exclude=SRPM*/ftp.heanet.ie::pub/pclinuxos/apt/ /mnt/datos/PCLinuxOS-repo`

-av es para q este en archive mode y sea verboso (o sea q nos indique todo el tiempo q es lo q esta haciendo)

--progress muestra el progreso de la transferencia de archivos

--delete borra los archivos q se encuentran en la carpeta de destino y q no corresponden

con la lista de archivos del servidor

--exclude=SRPM\* le indica a rsync que excluya cierto tipo de paquetes, en este caso q excluya los SRPM q son las fuentes de algunos programas  
/ftp.heanet.ie::pub/pclinuxos/apt/ le indica el repositorio o la dirección desde donde tiene q bajar los archivos (puede ser cualquier repositorio)  
/mnt/datos/PCLinuxOS-repo le indica adonde tiene q guardar los archivos (puede ser cualquier carpeta, en este caso es un ejemplo)

una vez escrito el comando, apretamos enter y rsync se va a conectar al repositorio va a mirar la lista de archivos y va a comenzar a bajar los archivos mostrando el progreso de cada uno, si tenemos q apagar la maquina o algo lo q hacemos es cancelar el comando con control+c y se va a detener pero si lo arrancamos otra vez va a continuar desde donde se había quedado.

Una vez q rsync termine de bajar todos los archivos (puede llevar bastante tiempo debido a q son alrededor de 30 gb), se puede actualizar el repositorio mediante el mismo comando, q automáticamente elimina los paquetes viejos y baja los nuevos.

Cuando finalicemos de bajar el repositorio y tenerlo bien actualizado tenemos q instalar el programa createrepo q está en el manejador de paquetes.

Luego abrimos una consola y nos dirigimos a la carpeta donde hayamos guardado el repositorio (nuestro caso seria cd /mnt/datos/PCLinuxOS-repo/)

una vez ahí hacemos

```
[lab@ServerLab PCLinuxOS-repo]$ cd pclinuxos/2007/
```

nos ponemos como root

y luego ejecutamos el siguiente comando

```
[root@ServerLab 2007]# createrepo base/
```

y les va a generar esta salida

```
Saving Primary metadata
```

```
Saving file lists metadata
```

```
Saving other metadata
```

luego si ingresamos a la carpeta base y hacemos un ls nos vamos a dar cuenta q hay una carpeta dice repodata, esta carpeta contiene en su interior varios archivos de extensión .xml, estos archivos son necesarios para q synaptic/apt-get puedan leer la información de los paquetes y cargar la lista de paquetes.

## CONFIGURANDO SYNAPTIC

Ingresamos a synaptic que es nuestro manejador de paquetes predeterminado y nos dirigimos al menú settings y hacemos clic en repositorios.

Nos va a aparecer otra ventana en la cual tenemos la lista de repositorios y abajo del todo tres botones, uno q dice New, otro q dice Cancel, y otro q dice Ok.

Hacemos clic en New y se nos van a habilitar tres campos en blanco.

en el que dice url tenemos q poner la dirección de la carpeta donde esta guardado el repositorio, en nuestro caso seria así la dirección

file:/mnt/datos/PCLinuxOS-repo/

luego en el segundo campo que dice Distribución escribimos  
pclinuxos/2007

y en el tercer y ultimo campo "Sections" escribimos las secciones q qremos q estén en nuestro repositorio, en nuestro caso la elección fue  
main extra nonfree kde kde4 (se escribe tal cual esta escrito ahi).

una vez echo esto buscamos el repositorio que estaba chequeado y lo "deschequeamos" , luego hacemos clic en ok y nos va a salir un cartel que nos avisa q la información del repositorio ah cambiado y que debemos actualizarlo para ver los paquetes correctamente, hacemos clic en Ok y luego hacemos clic en el botón reload.

Si siguieron paso a paso este tutorial y no saltaron ningún paso eh hicieron todo correctamente deberían encontrar el repositorio bajar los archivos de información y actualizar la lista de paquetes correctamente, y ya podemos probar la velocidad de tenerlo local.

## CONFIGURANDO EL REPOSITORIO EN UNA LAN (SIN PROXY)

Cuando ya tengamos el repositorio debidamente configurado y actualizado le podemos dar su verdadera utilidad, distribuirlo en una red de computadoras, si bien todas pueden tener Internet, la transferencia va a ser mucho más veloz por la red q por Internet.

Lo primero q tenemos q hacer es levantar una red entre las maquinas (samba, NIS, NFS, etc) o un servidor apache2 en el server, todo esto se explica en otro tutorial por eso no vamos a profundizar mas q en darles los nombres.

En nuestro caso elegimos apache, en la carpeta /var/www/html hacemos un enlace simbólico (acceso directo) a la carpeta q esta conteniendo nuestro repositorio local y luego probamos a ver si vía navegador web podemos ver las carpetas, para esto es necesario escribir la dirección ip del servidor en el navegador y tendrían q aparecer las carpetas q tenemos en apache.

una vez q esto este funcionando correctamente, nos dirigimos a alguna de las maquinas clientes y probamos si podemos ver las carpetas vía navegador web, si esto funciona abrimos el synaptic de dicha pc, vamos al menu settings y luego a repositorios, se va a abrir la lista de repositorios, hacemos clic en new y donde dice url escribimos "http://ip.del.servidor/PCLinuxOS-repo" (sin las comillas), en distribución escribimos "pclinuxos/2007", y en sections escribimos "main extra nonfree kde kde4". "Deschequeamos" el repositorio q estaba chequeado y chequeamos el nuestro, le damos Ok y nos va a aparecer el cartelito q nos informa q la información del repositorio ah cambiado y debemos hacer clic en reload para actualizarla, hacemos là ya tenemos nuestro repositorio funcionando a velocidad "copada".

## CONFIGURANDO EL REPOSITORIO EN UNA LAN (CON PROXY)

En nuestro caso, tenemos un proxy entonces si usamos la configuración anterior no va a funcionar, porq la pc se va a dirigir al proxy con esa ip y el proxy no lo va a dejar pasar asi q no va a encontrar los archivos necesarios para actualizar la información del registro.

Lo que debemos hacer es en la maquina cliente q ya tiene q tener configurado el proxy, ejecutar el drakconf o "configure your computer", ir a la sección network & internet, y allí a "setup a proxy server for files and web browsing" y cuando ingresemos vamos a ver un

recuadro q dice "no proxy for (coma separated list)" y ahi ponemos la ip del servidor y le damos aceptar.

Luego vamos a synaptic al menu settings y a repositories y hacemos lo mismo q en el paso anterior.

Le hacemos clic al botón reload y ya funciona el repositorio.

## **Principales navegadores en GNU-Linux.**

Entre los más notables navegadores se encuentran:

Mozilla: En realidad es un conjunto de programas para Internet, entre los que se destacan, el propio Navegador y su mensajero de correo electrónico.

### **Navegador Mozilla**

Mozilla Firefox: También conocido como Firefox a secas. Este navegador está desarrollado por la comunidad de programadores de Mozilla, muchos lo consideran la mejor alternativa a Internet Explorer. Posee más de 120 extensiones que se pueden agregar para aumentar las funcionalidades de la versión estándar. Y al igual que Mozilla se pueden descargar nuevos temas para modificar el aspecto visual del programa.

### **Firefox**

Konqueror

Además de administrador de archivos, sirve para navegar por la Web. Por lo tanto se puede pasar fácilmente de un directorio local a una página web. Es muy útil para usarlo dentro de KDE.

### **Konqueror**

Debido a que Internet Explorer fue el navegador de facto desde hace varios años, los diseñadores web han tenido la costumbre de hacer las páginas web pensadas solamente para que se vean bien en el navegador de Microsoft. Esto sin embargo, significó que muchísimos sitios se apartaran de los estándares propuestos por el World Wide Web Consortium .

La idea original de Internet es que cualquier página se vea correctamente independientemente del software que se utilice. Esto quiere decir que cuando una página web se ve mal en un navegador no quiere decir necesariamente que sea una deficiencia del navegador.

Si un sitio no ve correctamente en navegador distinto de Internet Explorer, se debería notificar al webmaster para que solucione el problema. En especial si se trata de un sitio público o de una empresa de servicios a la cual se está pagando.

No obstante hay que decir que la cantidad de sitios que no se ven correctamente (especialmente en Mozilla o Firefox) es relativamente baja.