



Powered By

GNU/Linux



RedTauros Ltda

<http://www.redtauros.com>

Índice de contenido

Introducción.....	3
Instalación del sistema operativo (2 horas).....	4
Estructura de Archivos en Linux	4
Participando un Disco Duro	5
Sistemas de Ficheros.....	6
FAT16 (File Allocation Table).....	7
FAT32 (File Allocation Table).....	7
NTFS (New Technology File System).....	7
EXT3 (Third Extended Filesystem).....	8
EXT4 (Fourth Extended Filesystem).....	9
HPFS (High Performance File System).....	9
ReiserFS	9
ZFS (Zettabyte File System).....	10
XFS.....	10
JFS (Journaling File System).....	11
Journaling	12
Entendiendo la desfragmentación :	13
EXT4 vs NTFS.....	19
Unidades de Información	21

Linux Básico II

Introducción

La instalación de Linux Ubuntu es realmente sencilla, siempre y cuando tengamos claros algunos conceptos.

El instalador de Linux ha evolucionado mucho, hoy en día tenemos los Live CD o Live USB, que nos permiten iniciar el sistema operativo desde un dispositivo extraíble o CD, sin alterar el sistema operativo instalado en la computadora. Esto nos permitirá probar el Linux y tomar la decisión, de instalarse o no.

Objetivos

Al finalizar este nivel *Linux Básico II* el estudiante:

1. Aprenderá a usar un live CD, para realizar la instalación y configuración de Ubuntu Linux.
2. **Disipará las dudas, en cuanto a los posibles términos o nombres usados en la instalación.**
3. **Conocerá las posibles opciones para realizar una instalación, además de la dual (junto a otro sistema operativo).**

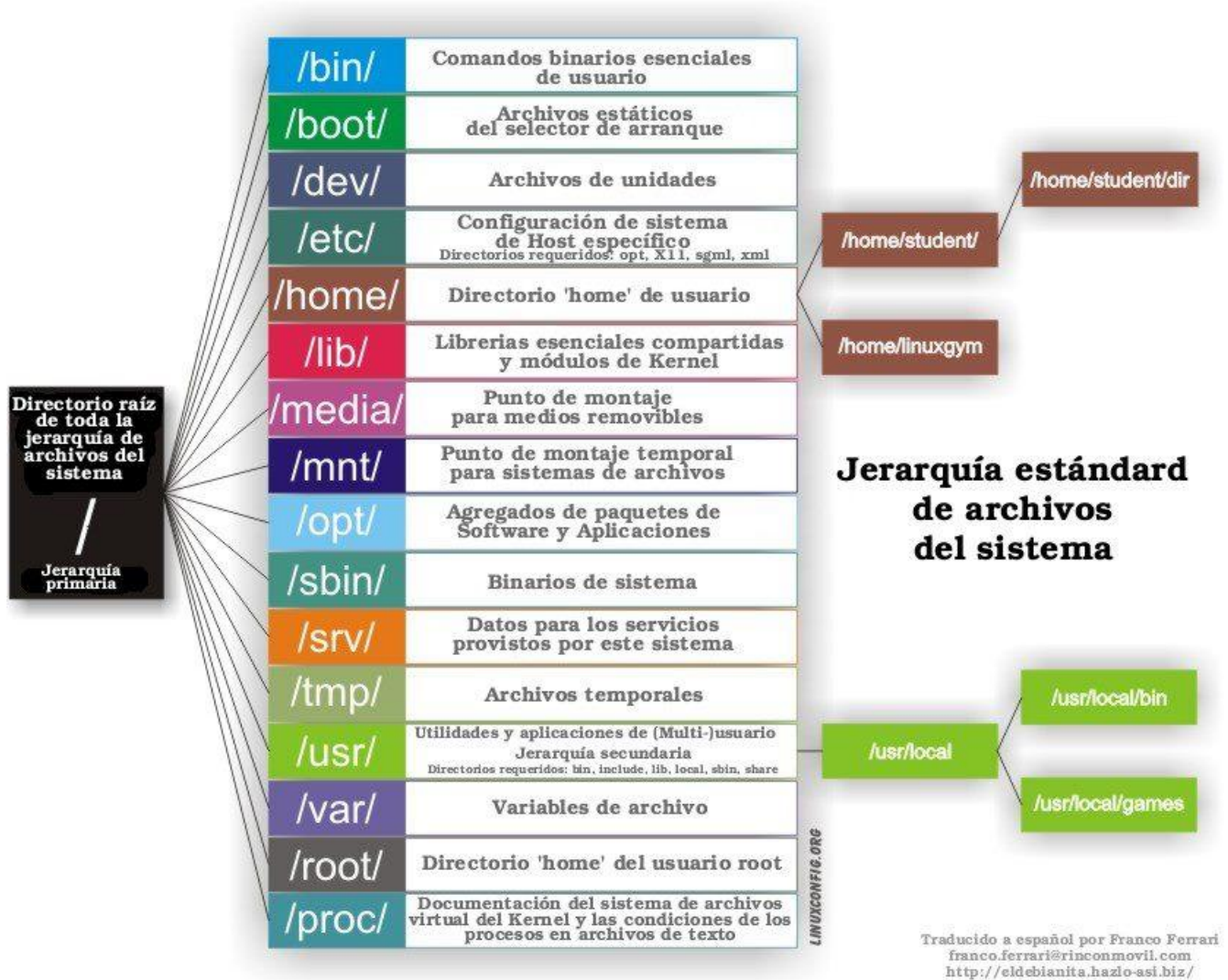
Temario

Instalación del sistema operativo (2 horas)

- Estructura de archivos en Linux
- Requisitos mínimos para la instalación de Ubuntu
- Configuración del idioma
- Configuración de la zona horaria
- Configuración del teclado
- Particionando el Disco Duro
- Introducir datos del usuario
- Confirmación de los datos
- Finalizando

Instalación del sistema operativo (2 horas)

Estructura de Archivos en Linux



- / Es la raíz del sistema de directorios. Aquí se monta la partición principal Linux EXT*
- bin Binarios de comandos esenciales
- boot Archivos estáticos de cargador de arranque (boot-loader)
- dev Archivos de dispositivos
- etc Configuración del sistema local-máquina
- home Directorios home de los usuarios
- lib Librerías compartidas
- mnt Punto de montaje de particiones temporales
- media Punto de montaje de dispositivos como cdrom, usb, memorias.
- root Directorio hogar del usuario root
- sbin Binarios del sistema esenciales
- tmp Archivos temporales
- usr Programas y librerías instalados con la distribución
- /usr/local Programas y librerías instalados por el administrador
- var Información variable, Datos varios como archivos de log (registro de actividad) de

programas, bases de datos, contenidos del servidor web, copias de seguridad

Particionando un Disco Duro

La primera vez que un usuario realiza la instalación de su sistema operativo GNU/Linux (Fedora, Ubuntu o cualquier otra distribución) es común dejar que el asistente gráfico de instalación sea quien realice la configuración por el usuario, cosa que no es recomendable por los siguientes eventos:

- En caso de pérdida del sistema operativo la información sufriría el mismo destino.
- Si se desea migrar a una nueva versión, ejemplo: de Ubuntu 11.10 a Ubuntu 12.10 hay que extraer toda la información y posteriormente instalar y reubicar archivos.
- Tiempo de recuperación del equipo mas prolongado.

A diferencia del particionamiento automático, el manual, permitirá definir cuáles particiones estarán en otra ubicación fuera del sistema operativo para solucionar los problemas citados con anterioridad.

El diseño de particiones que por ejemplo se puede hacer para un equipo de:

Casa

Directorio que usará la partición	Tamaño de la partición máxima	Tipo de partición
/	20Gb	primaria
swap	4Gb	primaria
/var	10Gb	lógica
/tmp	10GB	lógica
/home	resto del disco duro	primaria

Servidor Web

Directorio que usará la partición	Tamaño de la partición máxima	Tipo de partición
/	20Gb	primaria
swap	4Gb	primaria
/var	resto del disco duro	primaria
/tmp	10GB	lógica
/home	10GB	lógica

Sistemas de Ficheros

Piense en un sistema de ficheros como el molde de una estructura metálica y al sistema operativo como el perfil que se acoplara al molde que la contendrá.

Un sistema de ficheros se encarga de estructurar, gestionar y administrar eficientemente la información guardada en una unidad de almacenamiento masivo de datos como puede ser un disco duro, esta información puede ser consultada por un usuario de forma textual o de forma gráfica mediante la utilización de algún gestor de ficheros, que por lo regular son instalados y estructuran la información guardada en una unidad de almacenamiento (normalmente un disco duro) de una computadora, que luego será representada ya sea textual o gráficamente utilizando un gestor de archivos. La mayoría de los sistemas operativos poseen su propio sistema de archivos.

Un Sistema de Ficheros es un componente importante de un Sistema Operativo y suele contener:

- Métodos de acceso relacionados con la manera de acceder a los datos almacenados en archivos.
- Administración de archivos referida a la provisión de mecanismos para que los archivos sean almacenados, referenciados, compartidos y asegurados.
- Administración del almacenamiento auxiliar para la asignación de espacio a los archivos en los dispositivos de almacenamiento secundario.
- Integridad del archivo para garantizar la integridad de la información del archivo.

Sistemas Operativos como Linux hacen uso de los sistemas de ficheros como ext2, ext3, ext4 y swap, otros sistemas como por ejemplo Windows usan como sistema de ficheros los conocidos FAT y NTFS, por otra parte los sistemas operativos MacOS hacen uso del sistema de ficheros HFS.

La siguiente tabla nos dará una visión mas general sobre los sistemas de ficheros, así como de los sistemas operativos que hacen uso de las antes mencionadas.

Sistema Operativo	Sistema de Fichero Admitido
MS-DOS	FAT16
Windows 95	FAT16
Windows 98	FAT16, FAT32
Windows NT4	FAT32 NTFS
Windows 2000/XP	FAT16, FAT32, NTFS
Windows Vista/7	NTFS
Linux	EXT2, EXT3, EXT4, ReiserFS,Swap
MacOS	HFS
FreeBSD, OpenBSD	UFS
Sun Solaris	UFS, ZFS
IBM AIX	JFS

FAT16 (File Allocation Table)

Es un sistema de archivos desarrollado para MS-DOS, así como el sistema de archivos principal de las ediciones no empresariales de Microsoft Windows hasta Windows Me.

Las implementaciones más extendidas de FAT tienen algunas desventajas. Cuando se borran y se escriben nuevos archivos tiende a dejar fragmentos dispersos de éstos por todo el soporte. Con el tiempo, esto hace que el proceso de lectura o escritura sea cada vez más lento. La denominada desfragmentación es la solución a esto, pero es un proceso largo que debe repetirse regularmente para mantener el sistema de archivos en perfectas condiciones. FAT tampoco fue diseñado para ser redundante ante fallos.

Inicialmente solamente soportaba nombres cortos de archivo: ocho caracteres para el nombre más tres para la extensión. También carece de permisos de seguridad: cualquier usuario puede acceder a cualquier archivo.

FAT32 (File Allocation Table)

FAT32 fue la respuesta para superar el límite de tamaño de FAT16 al mismo tiempo que se mantenía la compatibilidad con MS-DOS en modo real. Microsoft decidió implementar una nueva generación de FAT utilizando direcciones de cluster de 32 bits (aunque sólo 28 de esos bits se utilizaban realmente).

En teoría, esto debería permitir aproximadamente 268.435.538 clusters, arrojando tamaños de almacenamiento cercanos a los dos terabytes. Sin embargo, debido a limitaciones en la utilidad ScanDisk de Microsoft, no se permite que FAT32 crezca más allá de 4.177.920 clusters por partición (es decir, unos 124 gigabytes). Posteriormente, Windows 2000 y XP situaron el límite de FAT32 en los 32 gigabytes. Microsoft afirma que es una decisión de diseño, sin embargo, es capaz de leer particiones mayores creadas por otros medios.

FAT32 apareció por primera vez en Windows 95. Era necesario reformatear para usar las ventajas de FAT32. Curiosamente, DriveSpace 3 (incluido con Windows 95 y 98) no lo soportaba. Windows 98 incorporó una herramienta para convertir de FAT16 a FAT32 sin pérdida de los datos. Este soporte no estuvo disponible en la línea empresarial hasta Windows 2000. El tamaño máximo de un archivo en FAT32 es 4 gigabytes, lo que resulta engorroso para aplicaciones de captura y edición de video, ya que los archivos generados por éstas superan fácilmente ese límite.

NTFS (New Technology File System)

Es un sistema de archivos diseñado específicamente para Windows NT (incluyendo las versiones Windows 2000, Windows 2003, Windows XP y Windows Vista), con el objetivo de crear un sistema de archivos eficiente, robusto y con seguridad incorporada desde su base. También admite compresión nativa de ficheros, cifrado (esto último sólo a partir de Windows 2000) e incluso transacciones (sólo a partir de Windows Vista). Está basado en el sistema de archivos HPFS de IBM/Microsoft usado en el sistema operativo OS/2, y también tiene ciertas influencias del formato de archivos HFS diseñado por Apple.

NTFS permite definir el tamaño del clúster, a partir de 512 bytes (tamaño mínimo de un sector) de forma independiente al tamaño de la partición. Es un sistema adecuado para las particiones de gran tamaño requeridas en estaciones de trabajo de alto rendimiento y servidores. Puede manejar volúmenes de, teóricamente, hasta $2^{64}-1$ clústeres. En la práctica, el máximo volumen NTFS soportado es de $2^{32}-1$ clústeres (aproximadamente 16 Terabytes usando clústeres de 4KB). Los inconvenientes que plantea son: Necesita para sí mismo una buena cantidad de espacio en disco duro, por lo que no es recomendable su uso en discos con menos de 400 MB libres. No es compatible con MS-DOS, Windows 95, Windows 98 ni Windows ME. No puede ser utilizado en disquetes.

Este sistema de archivos posee un funcionamiento prácticamente secreto, ya que Microsoft no ha liberado su código como hizo con FAT. Gracias a la ingeniería inversa, GNU/Linux tiene soporte parcial de escritura y total de lectura en particiones NTFS. Existen varias alternativas, como Captive-NTFS que usa las librerías propietarias de Windows NT para tener acceso completo a NTFS, o NTFS-3G. A Mayo del 2007, NTFS-3g ya es una versión definitiva, y han sido incorporados por múltiples distribuciones como Ubuntu, Gentoo, Debian, openSUSE, Mandriva, Fedora, sólo por mencionar algunas.

EXT2 (Second Extended Filesystem)

Es un sistema de archivos para el kernel de Linux. Fue diseñado originalmente por Rémy Card. La principal desventaja de ext2 es que no implementa el registro por diario (en inglés Journaling) que sí implementa su sucesor ext3, el cual es totalmente compatible. EXT2 fue el sistema de ficheros por defecto de las distribuciones de Linux Red Hat Linux, Fedora Core y Debian hasta ser reemplazado recientemente por su sucesor ext3. El sistema de ficheros tiene un tipo de tabla FAT de tamaño fijo, donde se almacenan los i-nodos. Los i-nodos son una versión muy mejorada de FAT, donde un puntero i-nodo almacena información del archivo (ruta o path, tamaño, ubicación física). En cuanto a la ubicación, es una referencia a un sector del disco donde están todos y cada una de las referencias a los bloques del archivo fragmentado. Estos bloques son de tamaño especificable cuando se crea el sistema de archivos, desde los 512 bytes hasta los 4 kB, lo cual asegura un buen aprovechamiento del espacio libre con archivos pequeños.

Los límites son un máximo de 2 TB de archivo, y de 4 TB de partición.

EXT3 (Third Extended Filesystem)

Es un sistema de archivos con registro por diario (journaling). La principal diferencia con ext2 es el registro por diario. Un sistema de archivos ext3 puede ser montado y usado como un sistema de archivos ext2. Otra diferencia importante es que ext3 utiliza un árbol binario balanceado (árbol AVL, es un tipo especial de árbol binario, en si es un arbol de busqueda) e incorpora el asignador de bloques de disco.

Aunque su velocidad y escalabilidad es menor que sus competidores, como JFS, ReiserFS o XFS, tiene la ventaja de permitir actualizar de ext2 a ext3 sin perder los datos almacenados ni formatear el disco y un menor consumo de CPU. El sistema de archivo ext3 agrega a ext2 lo siguiente:

- Registro por diario.
- Índices en árbol para directorios que ocupan múltiples bloques.
- Crecimiento en línea.

EXT4 (Fourth Extended Filesystem)

Es un sistema de archivos con registro por diario (en inglés Journaling), anunciado el 10 de octubre de 2006 por Andrew Morton, como una mejora compatible de ext3. Es el sistema de archivo más usado en distribuciones Linux. Las principales mejoras son:

- Soporte de volúmenes de hasta 1024 PiB.
- Soporte añadido de extent.

Actualmente, el ext4 es compatible con su anterior versión, el ext3. Esto quiere decir que se puede montar como una partición ext3. También se pueden montar las particiones ext3 como ext4. Aunque, si la partición ext4 usa extent (una de las mayores mejoras), la compatibilidad con la versión anterior, y por lo tanto, montar la partición como ext3, no es posible. Extents están configurados por defecto desde la versión del kernel 2.6.23.

Nota :

1024 GiB = 1 TiB; 1024 TiB = 1 PiB; 1024 PiB = 1 EiB

HPFS (High Performance File System)

Fue creado específicamente para el sistema operativo OS/2 para mejorar las limitaciones del sistema de archivos FAT. Fue escrito por Gordon Letwin y otros empleados de Microsoft, y agregado a OS/2 versión 1.2, en esa época OS/2 era todavía un desarrollo conjunto entre Microsoft e IBM.

Se caracterizaba por permitir nombres largos, metadatos e información de seguridad, así como de autocomprobación e información estructural. Otra de sus características es que, aunque poseía tabla de archivos (como FAT), ésta se encontraba posicionada físicamente en el centro de la partición, de tal manera que redundaba en menores tiempos de acceso a la hora de leerla/escribirla.

ReiserFS

ReiserFS es un sistema de archivos de propósito general, diseñado e implementado por un equipo de la empresa Namesys, liderado por Hans Reiser. Actualmente es soportado por Linux y existen planes de futuro para incluirlo en otros sistemas operativos. También es soportado bajo windows (de forma no oficial), aunque por el momento de manera inestable y rudimentaria (ReiserFS bajo windows). A partir de la versión 2.4.1 del núcleo de Linux, ReiserFS se convirtió en el primer sistema de ficheros con journal en ser incluido en el núcleo estándar. También es el sistema de archivos por defecto en varias distribuciones, como SuSE (excepto en openSuSE 10.2 que su formato por defecto es ext3), Xandros, Yoper, Linspire, Kurumin Linux, FTOSX, Libranet y Knoppix.

Con la excepción de actualizaciones de seguridad y parches críticos, Namesys ha cesado el desarrollo de ReiserFS (también llamado reiser3) para centrarse en Reiser4, el sucesor de este sistema de archivos. ReiserFS ofrece funcionalidades que pocas veces se han visto en otros sistemas de archivos:

- Journaling. Esta es la mejora a la que se ha dado más publicidad, ya que previene el riesgo de corrupción del sistema de archivos.
- Reparticionamiento con el sistema de ficheros montado y desmontado. Podemos aumentar el tamaño del sistema de ficheros mientras lo tenemos montado y desmontado (online y offline). Para disminuirlo, únicamente se permite estando offline (desmontado). Namesys nos proporciona las herramientas para estas operaciones, e incluso, podemos usarlas bajo un gestor de volúmenes lógicos como LVM o EVMS.
- Tail packing, un esquema para reducir la fragmentación interna.

Comparado con ext2 y ext3 en el uso de archivos menores de 4k, ReiserFS es normalmente más rápido en un factor de 10–15. Esto proporciona una elevada ganancia en las news, como por ejemplo Usenet, caches para servicios HTTP, agentes de correo y otras aplicaciones en las que el tiempo de acceso a ficheros pequeños debe ser lo más rápida posible.

Algunas de las desventajas de ReiserFS son:

Los usuarios que usen como sistema de ficheros ext2, deben formatear sus discos, aunque no así los que usen ext3.

- ReiserFS en versiones del kernel anteriores a la 2.4.10 se considera inestable y no se recomienda su uso, especialmente en conjunción con NFS
- Algunas operaciones sobre archivos (por ejemplo unlink(2)) no son síncronas bajo ReiserFS, lo que pueden causar comportamientos extraños en aplicaciones fuertemente basadas en locks de archivos.
- No se conoce una forma de desfragmentar un sistema de archivos ReiserFS, aparte de un volcado completo y su restauración.
- Tempranas implementaciones de ReiserFS (anteriores a la incluida en el kernel 2.6.2), eran susceptibles de problemas de escrituras fuera de orden, lo que provocaba que archivos siendo escritos durante una caída del sistema, ganaran un pico de bytes extras de basura en el siguiente montado del sistema de archivos. La implementación actual de journaling, es correcta en este aspecto, manteniendo el journaling ordenado, del estilo de ext3.

ZFS (Zettabyte File System)

Es un sistema de ficheros desarrollado por Sun Microsystems para su sistema operativo Solaris. El significado original era 'Zettabyte File System', pero ahora es un acrónimo recursivo.

El anuncio oficial de ZFS se produjo en Septiembre del 2004. El código fuente del producto final se integró en la rama principal de desarrollo de Solaris el 31 de octubre del 2005 y fue lanzado el 16 de noviembre de 2005 como parte del build 27 de OpenSolaris.

ZFS fue diseñado e implementado por un equipo de Sun liderado por Jeff Bonwick.

ZFS destaca por su gran capacidad, integración de los conceptos anteriormente separados de sistema de ficheros y administrador de volúmenes en un solo producto, nueva estructura sobre el disco, sistemas de archivos ligeros, y una administración de espacios de almacenamiento sencilla.

Sun ha indicado que está investigando el port del producto a Linux, aunque no hay planes para llevarlo a HP-UX o AIX. FreeBSD 7, a lanzarse a fines del 2007, también dará soporte a ZFS. Recientemente, Apple ha confirmado que utilizará ZFS en la próxima versión Server de su sistema operativo Mac OS X 10.6 Snow Leopard.

XFS

XFS es un sistema de archivos de 64 bits con journaling de alto rendimiento creado por SGI (antiguamente Silicon Graphics Inc.) para su implementación de UNIX llamada IRIX. En mayo del 2000, SGI liberó XFS bajo una licencia de código abierto.

XFS se incorporó a Linux a partir de la versión 2.4.25, cuando Marcelo Tosatti (responsable de la rama 2.4) lo consideró lo suficientemente estable para incorporarlo en la rama principal de desarrollo del kernel. Los programas de instalación de las distribuciones de SuSE, Gentoo, Mandriva, Slackware, Fedora Core, Ubuntu y Debian ofrecen XFS como un sistema de archivos más. En FreeBSD el soporte para solo-lectura de XFS se añadió a partir de Diciembre de 2005 y en Junio de 2006 un soporte experimental de escritura fue incorporado a FreeBSD-7.0-CURRENT.

JFS (Journaling File System)

Es un sistema de archivos con respaldo de transacciones desarrollado por IBM y usado en sus servidores. Fue diseñado con la idea de conseguir "servidores de alto rendimiento y servidores de archivos de altas prestaciones, asociados a e-business". Según se lee en la documentación y el código fuente, va a pasar un tiempo antes de que la adaptación a Linux este finalizada e incluida en la distribución estándar del kernel. JFS utiliza un método interesante para organizar los bloques vacíos, estructurándolos en un árbol y usa una técnica especial para agrupar bloques lógicos vacíos.

JFS fue desarrollado para AIX. La primera versión para Linux fue distribuida en el verano de 2000. La versión 1.0.0 salió a la luz en el año 2001. JFS está diseñado para cumplir las exigencias del entorno de un servidor de alto rendimiento en el que sólo cuenta el funcionamiento. Al ser un sistema de ficheros de 64 bits, JFS soporta ficheros grandes y particiones LFS (del inglés Large File Support), lo cual es una ventaja más para los entornos de servidor.

También está disponible para las últimas versiones de OS/2 y eComstation Las principales ventajas de JFS son:

- Eficiente respaldo de transacciones (Journaling).

JFS, al igual que ReiserFS, sigue el principio de metadata only. En vez de una completa comprobación sólo se tienen en cuenta las modificaciones en los metadatos provocadas por las actividades del sistema. Esto ahorra una gran cantidad de tiempo en la fase de recuperación del sistema tras una caída. Las

actividades simultáneas que requieren más entradas de protocolo se pueden unir en un grupo, en el que la pérdida de rendimiento del sistema de ficheros se reduce en gran medida mediante múltiples procesos de escritura.

- Eficiente administración de directorios.

JFS abarca diversas estructuras de directorios. En pequeños directorios se permite el almacenamiento directo del contenido del directorio en Inode. En directorios más grandes se utiliza Btrees, que facilitan considerablemente la administración del directorio. Mejor utilización de la memoria mediante adjudicación dinámica de Inodes.

Con ext2 debe dar por anticipado el grosor del Inode (la memoria ocupada por la información de administración). Con ello se limita la cantidad máxima de ficheros o directorios de su sistema de ficheros. JFS le ahorra esto, puesto que asigna memoria Inode de forma dinámica y la pone a su disposición cuando no se está utilizando.

Journaling

Es un mecanismo por el cual un sistema informático puede implementar transacciones. También se le conoce como "registro por diario". Se basa en llevar un journal o registro de diario en el que se almacena la información necesaria para restablecer los datos afectados por la transacción en caso de que ésta falle. El procedimiento es básicamente el siguiente:

1. Se bloquean las estructuras de datos afectadas por la transacción para que ningún otro proceso pueda modificarlas mientras dura la transacción.
2. Se reserva un recurso para almacenar el journal. Por lo general suelen ser unos bloques de disco, de modo que si el sistema se para de forma abrupta (corte eléctrico, avería, fallo del sistema operativo...) el journal siga disponible una vez reiniciado el sistema.
3. Se efectúan una a una las modificaciones en la estructura de datos. Para cada una:
 1. Se apunta en el journal como deshacer la modificación y se asegura de que esta información se escribe físicamente en el disco.
 2. Se realiza la modificación.
4. Si en cualquier momento se quiere cancelar la transacción se deshacen los cambios uno a uno leyéndolos y borrándolos del journal.
5. Si todo ha ido bien, se borra el journal y se desbloquean las estructuras de datos afectadas.

Las aplicaciones más frecuentes de los sistemas de journaling se usan para implementar transacciones de sistemas de bases de datos y, más recientemente, para evitar la corrupción de las estructuras de datos en las que se basan los sistemas de archivos modernos.

En el caso concreto de los sistemas de archivos, el journaling se suele limitar a las operaciones que afectan a las estructuras que mantienen información sobre:

- Estructuras de directorio.
- Bloques libres de disco.
- Descriptores de archivo (tamaño, fecha de modificación...)

El hecho de que no se suele implementar el journaling de los datos concretos de un archivo suele carecer de importancia, puesto que lo que persigue el journaling de sistemas de archivos es evitar los engorrosos y largos chequeos de disco que efectúan los sistemas al apagarse bruscamente, ya que el sistema al arrancar solo deberá deshacer el journal para tener un sistema coherente de nuevo.

Entendiendo la desfragmentación :

Es una pregunta que sale con bastante frecuencia, ¿porqué no es necesario desfragmentar en Linux?

En lugar de intentar explicarlo con términos muy técnicos, he optado por un cuadro de ASCII dado que una imagen vale más que mil palabras. Aquí está el cuadro que utilizaré para explicarlo:

```
  a b c d e f g h i j k l m n o p q r s t u v w x y z
a 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
d 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
e 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
g 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
h 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
j 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
k 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
l 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
m 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
n 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
o 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
p 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
q 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
r 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
s 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
t 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
u 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
v 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
w 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
x 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
y 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
z 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Esto es una representación (muy pequeña) de un disco duro, completamente vacío - por eso a ceros. El a-z en la parte superior y en el lado izquierdo del cuadro son utilizados para localizar cada byte individual de datos: el izquierdo superior es aa, superior derecho za y el último de la izquierda es az.

Empezaremos con un sistema de ficheros muy sencillo de una clase que seguro están muy familiarizados: uno que necesita defragmentar ocasionalmente. Desde que los usuarios tanto de Windows como de Linux usan sistemas de ficheros FAT, aunque sea para discos flash USB, es un sistema de ficheros importante, aunque desafortunadamente padece de fragmentación.

Añadiremos un archivo a nuestro sistema de ficheros, y nuestro disco duro se parecerá a esto:

```
  a b c d e f g h i j k l m n o p q r s t u v w x y z
```

```

a TOChello.txtae/e0000000000
b 00000000000000000000000000000000
c 00000000000000000000000000000000
d 000000000000000000000000000000TOC
e Hello, _world0000000000000000
f 00000000000000000000000000000000

```

(Las filas vacías g-z se han omitido para más claridad)

Para explicar lo que están viendo: las primeras cuatro filas del disco se reservan para una tabla de contenidos o TOC. Este TOC almacena la localización de cada fichero en el sistema de ficheros. En el ejemplo anterior, TOC contiene un fichero, denominado “hello.txt”, y nos dice que el contenido de este fichero se puede localizar entre ae y le. Miramos esas posiciones y vemos que el contenido del fichero es “Hello, world”.

¿Todo bien hasta aquí? Añadimos ahora otro fichero:

```

a b c d e f g h i j k l m n o p q r s t u v w x y z
a TOChello.txtae/ebye.txtm ez
b e0000000000000000000000000000000
c 00000000000000000000000000000000
d 000000000000000000000000000000TOC
e Hello, _worldGoodbye, _world
f 00000000000000000000000000000000

```

Como se puede ver, el segundo fichero se ha añadido inmediatamente después del primero. La idea aquí es que si todos los ficheros se guardan juntos, acceder a ellos será más rápido y fácil: la parte lenta del disco duro es la aguja, cuanto menos se tiene que mover, tanto más rápido se podrá leer/escribir.

El problema que esto causa lo podemos ver cuando decidimos editar nuestro primer fichero. Digamos que queremos añadir algún símbolo de exclamación a nuestro “Hello” para que parezca más entusiasta. Entonces tenemos un problema: no hay sitio para nuestro símbolo de exclamación en nuestro sistema de ficheros: el fichero “byte.txt” está en medio. Tenemos solamente dos opciones, ninguna ideal:

- Borrar el fichero de su posición original, y añadimos el nuevo y más grande al final del segundo fichero.
- Fragmentamos el archivo, de manera que exista en dos sitios pero sin espacios vacíos.

Para ilustrarlo, aquí tenemos una aproximación a la primera opción:

```

a b c d e f g h i j k l m n o p q r s t u v w x y z
a TOChello.txtafnbye.txtm ez
b e0000000000000000000000000000000
c 00000000000000000000000000000000
d 000000000000000000000000000000TOC
e 000000000000Goodbye, _world
f Hello, _world!!0000000000000000

```

Y aquí a la segunda:

En Windows el sistema de archivos de disco es NTFS (New Technology FileSystem) cuyo debut se produjo por allá en Julio de 1993 con el lanzamiento de Windows NT 3.1. Por supuesto NTFS se ha venido actualizando de a poquitos sin romper la compatibilidad, siendo la versión más reciente la 3.1 que, a todas estas, no ha cambiado desde su introducción en Windows XP y es usado en Windows Server 2003, Windows Server 2008, Windows Vista, and Windows 7.

En Linux el sistema de archivos por tradición ha sido EXT (Extended FileSystem) (En Linux a diferencia de Windows hay varios sistemas de archivos y pueden seleccionarse a la hora de la instalación) siendo el más reciente ext4 introducido el 10 de octubre de 2006 en el kernel Linux 2.6.19 y la opción por defecto para las distribuciones más populares como Ubuntu y Fedora.

Veamos en detalle lo que pueden hacer estos sistemas de archivos:

Límites	
De entrada, diré que un usuario casero posiblemente nunca llegue a alcanzar estos límites, sin embargo, es bueno tenerlos en mente por propósitos científicos.	
NTFS	Ext4
Tamaño máximo de volumen = 264TB (262.144 GB)	Tamaño máximo de volumen = 1024PiB (1'073.741.824 GB)
Tamaño máximo de archivo = 16TiB (16.384 GB)	Tamaño máximo de archivo = 16TiB (16.384 GB)
Número máximo de archivos = 4.294.967.295	Número máximo de archivos = 4.000.000.000

Confiabilidad	
NTFS	Ext4
<p>NTFS Log. NTFS es un sistema basado en Journaling (Registro por diario) que usa el \$LogFile para grabar todos cambios en el sistema de archivos. Básicamente mantiene coherente la estructura de archivos incluso si el sistema falla.</p> <p>También NTFS utiliza el USN (Update Sequence Number Journal) que graba todos los cambios sobre los archivos. Sin embargo, el journal no preserva las versiones originales de los archivos modificados por lo que no es posible deshacer los cambios realizados.</p>	<p>Ext4 Journal Ext4 también hace uso del registro por diario. Sin embargo también realiza comprobaciones sobre el diario (Checksums) para mejorar la fiabilidad ya que el archivo de log es uno de los más leídos del disco.</p> <p>Este proceso hace que se evite la lectura/escritura del disco mientras se realiza el proceso de registro en el journal aumentando ligeramente el rendimiento.</p>

Fragmentación
Con el uso, todos los sistemas de archivos son susceptibles a fragmentarse, lo cual, eventualmente disminuye el rendimiento y el tiempo de respuesta del disco.

NTFS	Ext4
Ordena y accede a los archivos secuencialmente. Esto lo hace particularmente susceptible a la fragmentación siendo recomendado ejecutar la utilidad de desfragmentación cuando el índice de archivos fragmentados supera el 10%.	Ordena y accede a los archivos desde la mitad de la unidad, reduciendo la fragmentación y el tiempo acceso a archivos en 1/2 de entrada. Incorpora características para reducir la fragmentación siendo extraño en equipos caseros que supere el 10%. Pensando en sistemas altamente transaccionales , existe una utilidad de desfragmentación en línea.


Rendimiento	
Varios test sugieren que ext4 puede hacer ciertas operaciones de lectura/escritura mucho más rápido que NTFS[1]	
NTFS	Ext4
Como ya lo habíamos mencionado anteriormente, NTFS es particularmente susceptible a la fragmentación por lo que su rendimiento disminuye con el paso del tiempo.	Si bien ext4 es de por sí rápido, pueden realizarse trucos (No siempre recomendados) como desactivar el journal para aumentar las velocidades de lectura/escritura en el disco.

La evolución de los sistema de archivos es particularmente lenta y debe ser especialmente cuidadosa. Quizás en este sentido los sistemas operativos abiertos han gozado de mayores libertades en implementación y adopción de éstos sistemas dando como resultado una pequeña ventaja sobre su homólogo de Windows.

Unidades de Información

Unidades de información (del Byte)			
Sistema Internacional (Decimal)		ISO/IEC 80000-13 (Binario)	
Múltiplo - (Símbolo)	SI	Múltiplo - (Símbolo)	ISO/IEC
kilobyte (kB)	10^3	Kibibyte (KiB)	2^{10}
Megabyte (MB)	10^6	Mebibyte (MiB)	2^{20}
Gigabyte (GB)	10^9	Gibibyte (GiB)	2^{30}

Terabyte (TB)	10^{12}	Tebibyte (TiB)	2^{40}
Petabyte (PB)	10^{15}	Pebibyte (PiB)	2^{50}
Exabyte (EB)	10^{18}	Exbibyte (EiB)	2^{60}
Zettabyte (ZB)	10^{21}	Zebibyte (ZiB)	2^{70}
Yottabyte (YB)	10^{24}	Yobibyte (YiB)	2^{80}



Linux Ubuntu online Grafico:

<https://www.onworks.net/playonline/index.php>

Linux Ubuntu online Consola:

<https://copy.sh/v86/?profile=linux26>