Evolución de los sistemas de archivos en Linux

Mario Medina C.

Depto. Ing. Eléctrica, UdeC

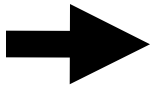
mariomedina@udec.cl

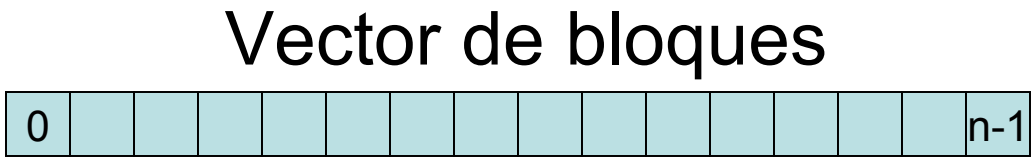
Tópicos a tratar

- Sistemas de archivos tradicionales
  - El sistema de archivos Ext2
- Sistemas de archivos basados en bitácoras (journaling)
  - El sistema de archivos Ext3
- Sistema de archivos actuales
  - El sistema de archivos Ext4
  - Otros

Sistemas de archivos

- Sistemas de archivos ven un medio de almacenamiento masivo como un *vector de bloques*
  - Bloques típicamente son de 1, 2 ó 4 KiB
  - Datos se almacenan en sectores de 512 bytes





Almacenando un archivo

- Almacenar los datos del archivo y los metadatos
  - Nodo índice contiene índices de los bloques de datos
  - Nombre se almacena en el directorio

Nodo índice (Nodo-i)

Dueño			
Tipo			
Tamaño			
1	11	13	14
17	18	20	5
6	7		

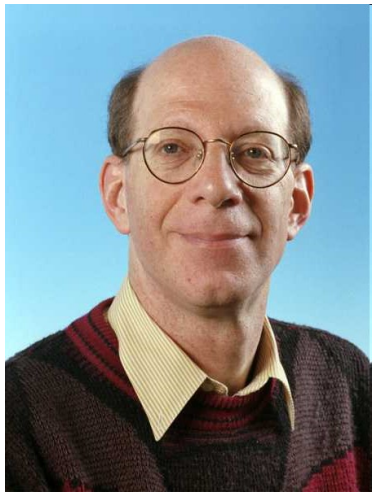
Índice de bloques

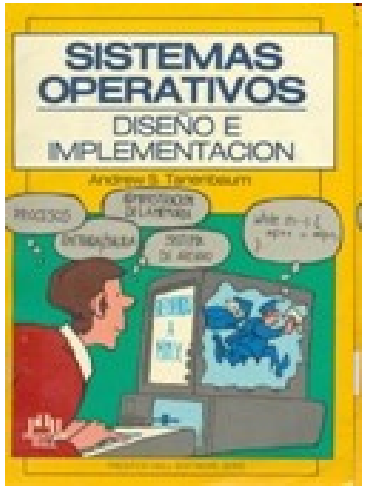
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

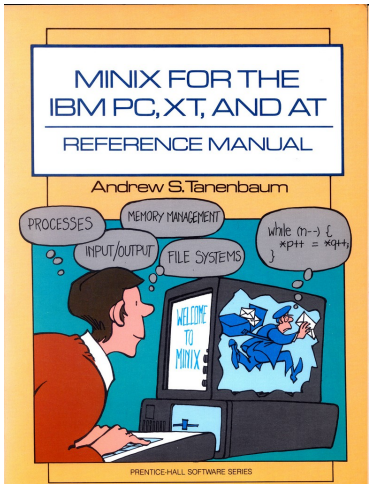
Bloques de disco

Breve historia

- Linus Torvalds
- MINIX
- Andy Tanenbaum





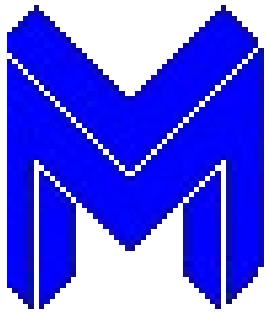


MINIX y Linux

- MINIX
  - Plataforma, guía y fuente de inspiración para desarrollar Linux
- Linux
  - Primera versión del kernel en 1991
  - Inicialmente, utilizaba el sistema de archivos de MINIX
    - Basado en Berkeley FFS
    - Simplificado para la enseñanza

MINIX FS

- Sistema de archivos de tamaño máximo 64 MiB
- Archivos de tamaño máximo 64 MiB
- Directorios tienen tamaño fijo
- Nombres de archivos limitados a 14 caracteres
- Bloques de 1 KiB
- Punteros a bloques de 16 bits



Sistema de archivos Ext

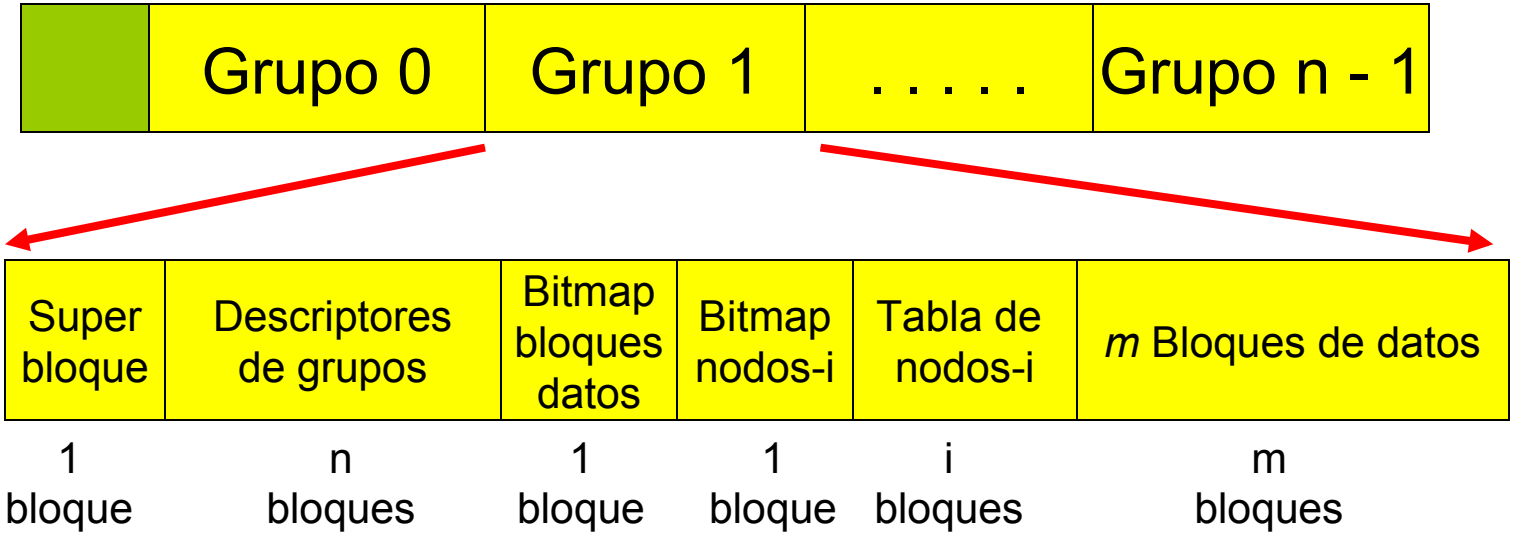
- Sistema de archivos extendido (*Extended File System*)
  - Liberado en 1992 para reemplazar MINIX FS
  - Sistema de archivos de tamaño máximo 4 TiB
  - Archivos de tamaño máximo 2 GiB
  - Nombres de archivos de 255 caracteres
  - Ineficiente!
    - Maneja bloques libres y nodos-I como listas encadenadas

Sistema de archivos Ext2

- Segundo sistema de archivos extendido (*Extended File System 2*)
  - Aparece en 1993
  - Sistema de archivos de tamaño máximo 4 TiB
  - Archivos de tamaño máximo 2 GiB
  - Directorios de tamaño variable
  - Nombres de archivos de 255 caracteres
  - Eficiente y robusto

Sistema de archivos Ext2

- Divide el disco en *n* grupos de bloques
  - *m* bloques de datos
  - *i* nodos-i asociados a los archivos en el grupo



Contenido de un grupo

- *m* bloques de datos
- *i* nodos-i
- Bitmap de bloques libres en el grupo
- Bitmap de nodos-i libres en el grupo
- Copia del *superbloque*
- Copia de los descriptores de todos los grupos

Superbloque

- Contiene información de todo el sistema
  - Número total de nodos-i
  - Tamaño total en bloques
  - Tamaño del bloque (1, 2 ó 4 KiB)
  - Tamaño del nodo-i (128 bytes)
  - Número total de bloques libres
  - Número total de nodos-i libres
  - Número de bloques y nodos-i por grupo
  - Contador de operaciones de montaje

Descriptor de grupo

- Cada grupo tiene su propio descriptor
- Replicado en cada grupo (*n* copias)
  - Número de bloque del bitmap de bloques de datos
  - Número de bloque del bitmap de nodos-i
  - Número de bloque del primer bloque de datos
  - Contador de bloques de datos libres
  - Contador de nodos-i libres
  - Número de directorios en el grupo

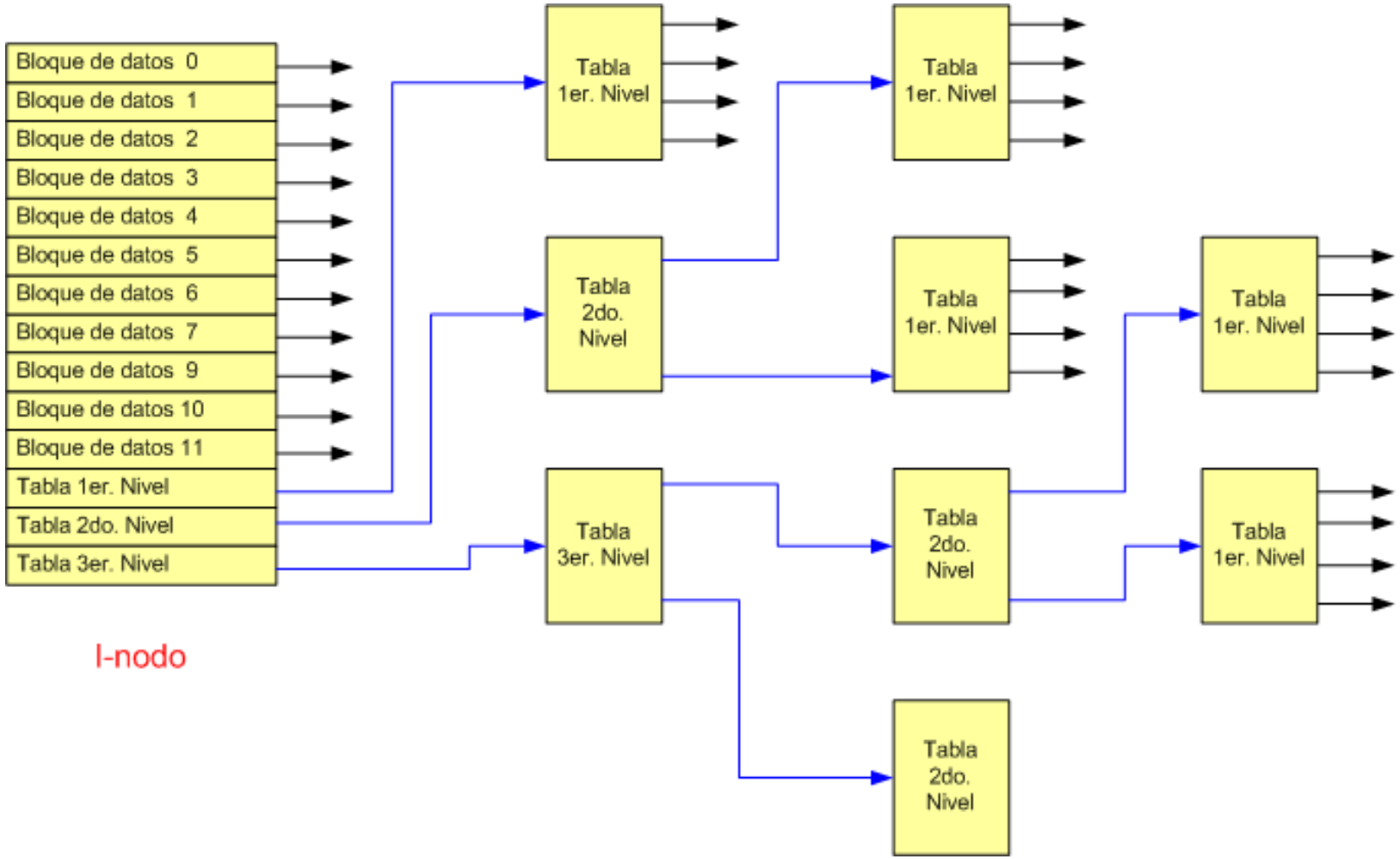
Nodo-i

- 128 bytes de longitud
  - Tipo del archivo (archivo, directorio, socket, enlace, etc.)
  - Tamaño del archivo en bytes
  - Tamano del archivo en bloques
  - Dueño del archivo
  - 3 marcas de tiempo
    - Fecha y hora de creación
    - Fecha y hora de última modificación
    - Fecha y hora de último acceso

Punteros en nodo-i

- 12 punteros directos
  - Contienen número del bloque de datos
- Puntero a tabla de primer nivel
  - Tabla contiene punteros directos
- Puntero a tabla de segundo nivel
  - Tabla de punteros a tablas de primer nivel
- Puntero a tabla de tercer nivel
  - Tabla de punteros a tablas de segundo nivel

Direccionamiento de bloques



Direccionamiento de bloques

- Para bloques de 4 KiB y punteros de 32 bits
  - 12 punteros directos a bloques de datos: 48 KiB
  - Puntero a tabla de primer nivel: 4 MiB
    - Ocupa un bloque de datos
  - Puntero a tabla de segundo nivel: 4 GiB
  - Puntero a tabla de tercer nivel: 4 TiB
- Máximo tamaño de un archivo
  - 4 TiB + 4 GiB + 4 MiB + 48 KiB

Directorios

- Archivo especial
- Estructura de datos contiene
  - Número del nodo-I
  - Tamaño de esta estructura de datos
  - Longitud del nombre
    - Máximo 255 caracteres
  - Tipo del archivo
  - Nombre del archivo

Estructura de un directorio

	I-nodo	Largo registro	Largo nombre	Tipo	Nombre del archivo
0		12	1	2	. \0 \0 \0
12	22	12	2	2	. . \0 \0
24	53	16	5	2	h o m e
40	67	28	3	2	u s r \0
52	0	16	7	1	o l d f l l e \0
68	34	12	4	2	s b i n
	4 Bytes	2 Bytes	1	1	Largo Variable

- En el ejemplo, archivo `oldfile` está borrado
  - Nodo-i es 0

Límites de Ext2

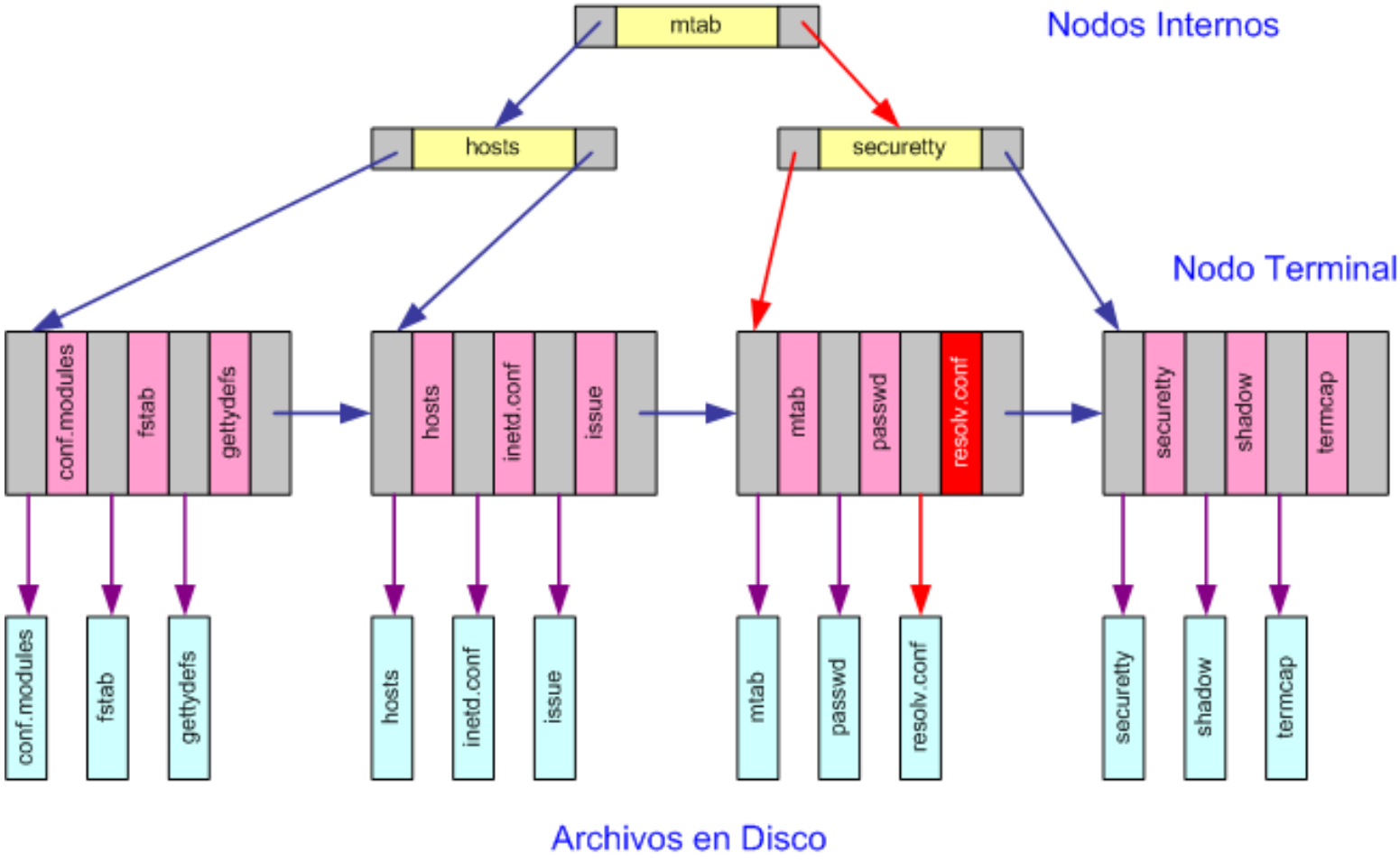
- Bloques por grupo 32 Ki
- Bytes por grupo 128 MiB
- Nodos-i por grupo 32 Ki
- Número máximo de grupos 128 Ki
- Tamaño máximo de un disco 16 TiB
- Bloques por archivo 2 Gi
- Tamaño máximo de un archivo 4 TiB

- Suponiendo bloques de 4 KiB

Problemas de Ext2

- Directorios son listas encadenadas
  - Parche a Ext2 en 2002 implementa directorios como árboles Htree (opción `dir_index`)
    - Similar a árboles B+
  - Árboles de altura constante (1 ó 2 niveles)
  - Factor de *fanout* alto
  - Función de dispersión (*hash*) aplicada al nombre del archivo
  - Compatible con lista de Ext2 rev. 0

Árboles Htree



Superbloques ralos

- Copia del superbloque en todos los grupos
- Modificaciones deben actualizar todas las copias
  - Ineficiente!
- Ext2 revisión 1 agrega opción `sparse_super`
  - Copias del superbloque y los descriptores de grupo se almacenan sólo en los grupos 0, 1, y potencias de 3, 5, y 7

Soporte para archivos grandes

- Ext2 rev. 0 limita tamaño máximo de un archivo a 2 GiB
  - Límite fue sobrepasado por evolución de los discos magnéticos
- Ext2 rev.1 agrega opción `large_file`
  - Permite utilizar archivos de tamaño hasta 4 TiB
  - Este límite está siendo sobrepasado hoy en día.

Tamaño máximo de un grupo

- Bitmap de bloques libres debe caber en un bloque
  - Si bloque es 4 KiB, bitmap describe a lo más 32 Ki bloques
  - Cada grupo de bloques puede almacenar a lo más 128 MiB de datos

Número fijo de nodos-i

- Número de nodos-i por grupo se define al momento de crear el sistema de archivos
  - Tamaño de nodo-i: 128 bytes
  - 32 nodos-i en un bloque de 4 KiB
  - Debe haber un nodo-i por cada bloque de datos
    - 32 Ki bloques $\rightarrow$ 32 Ki nodos-i
  - Máximo 4 MiB dedicados a almacenar nodos-i

Metadatos en Ext2

- Ext2 mantiene 6 tipos de metadatos
  - Superbloques
  - Descriptores de grupo
  - Nodos-i
  - Tablas de punteros de primer, segundo y tercer nivel
  - Bitmaps de bloques libres
  - Bitmaps de nodos-i libres

Actualizando metadatos

Si un archivo crece de 3 a 5 bloques, actualizar

- Superbloque y todas sus copias
  - Actualizar contador de total de bloques libres
- Descriptor del grupo y todas sus copias
  - Actualizar contador de bloques libres en el grupo
- Grupo
  - Actualizar bitmap de bloques libres
- Nodo-i del archivo
  - Actualizar tamaño, bloques usados, punteros a bloques, tablas de punteros, etc.

Metadatos en memoria

- Información crítica del sistema de archivos se lee a memoria al momento del montaje
- Memoria RAM no puede almacenar todas las estructuras de datos
- Qué hacer?
  - Guardar datos en cache!

Metadatos en memoria

- Superbloque y descriptores
  - Siempre en memoria
- Nodos-i
  - En buffer cache
- Bloques de datos
  - En buffer cache
- Bitmaps de bloques y de nodos-i libres
  - En cola FIFO en memoria

Consistencia de metadatos

- Metadatos se almacenan en memoria
  - Mayor desempeño por acceso rápido a metadatos frecuentemente usados
- Copias en disco son actualizadas periódicamente por el proceso `sync`
  - Escribe superbloques, nodos-i, bitmaps modificados desde memoria al disco
- Qué pasa si hay un problema?
  - Inconsistencia en los metadatos!

Reparación usando `fsck`

- `fsck` asegura consistencia del sistema de archivos
  - Corrige errores en metadatos
  - Debe revisar todo el sistema de archivos
- Tiempo de ejecución proporcional al tamaño del disco
  - Puede demorar muchísimo para discos grandes
- Solución: *journaling*

Tercer sistema de archivos extendido (Ext3)

- Compatible con Ext2
  - Partición Ext3 puede ser montada como partición Ext2
  - Conversión in situ entre Ext2 y Ext3 posible
  - Bitácora almacenada como archivo `/.journal`
  - Bitácora se escribe al disco con mayor frecuencia que sync de Ext2
  - Volumen de datos a escribir es menor

La bitácora o *journal*

- Estructura de datos almacenada en disco
  - Bitácora registra cambios a realizar a los metadatos
  - Después se realizan los cambios a los datos
  - Luego, se marcan los cambios como realizados en la bitácora (*commit*)
- Restauración de consistencia
  - Revisar últimas modificaciones a la bitácora
  - Tiempo de ejecución proporcional al tamaño de la bitácora

Modos de operación de Ext3

Journal

- Bitácora guarda cambios a los datos y a los metadatos
- Luego, los datos modificados son escritos a disco y se hace el *commit*
  - Minimiza chances de pérdidas de datos
  - Modo más lento y seguro
  - Alto número de accesos a disco

Modos de operación de Ext3

Ordered

- Bitácora sólo almacena cambios a los metadatos
  - Asegura que los datos son escritos antes que el *commit* en la bitácora
  - Opción por omisión en Ext3
  - Minimiza chances de corrupción

Modos de operación de Ext3

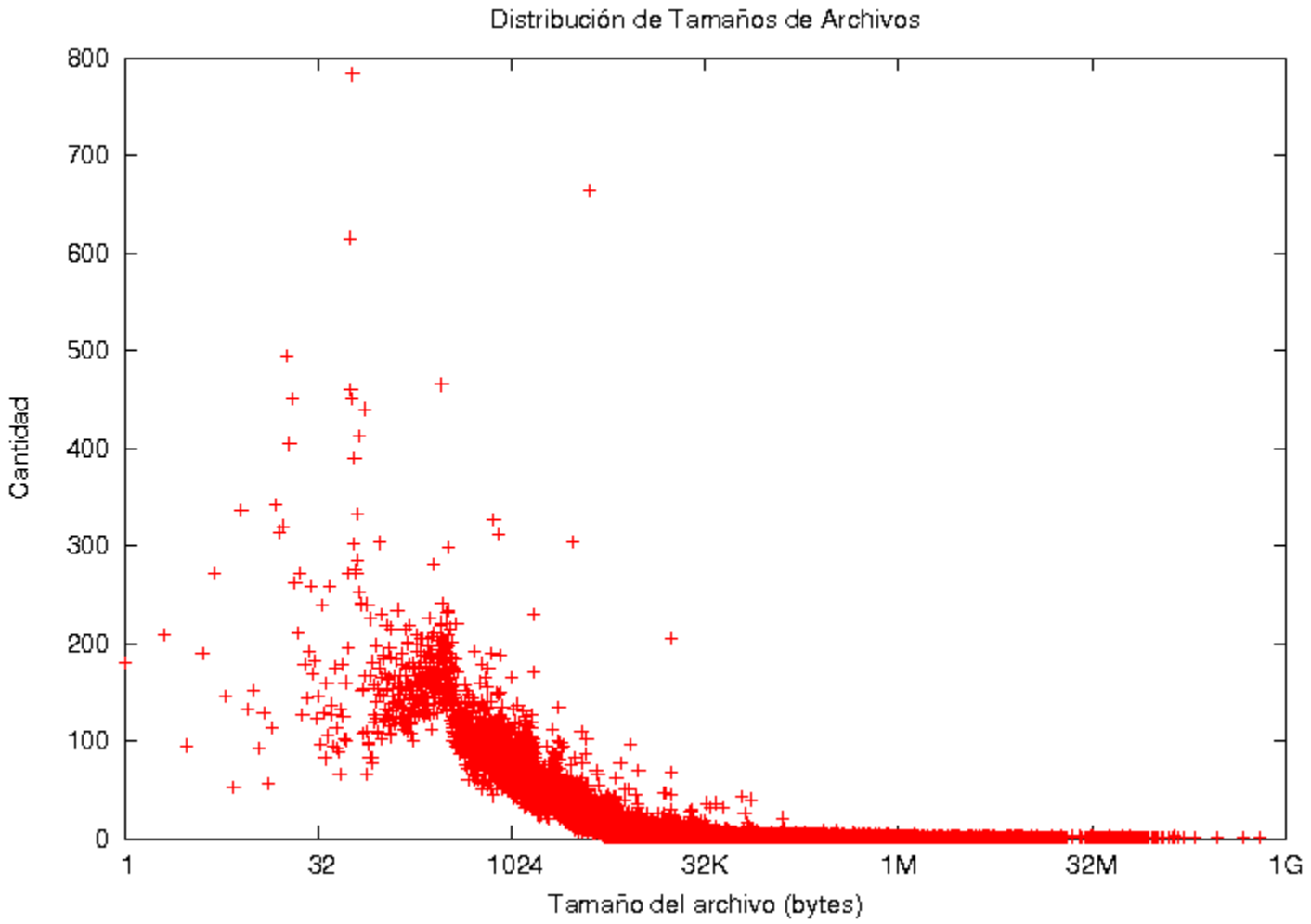
Writeback

- Bitácora sólo almacena cambios a los metadatos
  - No hay reordenamiento de escrituras a disco
  - Es posible que ocurra un *commit* antes que los datos sean escritos al disco
  - Modo más rápido, pero que ofrece menos robustez

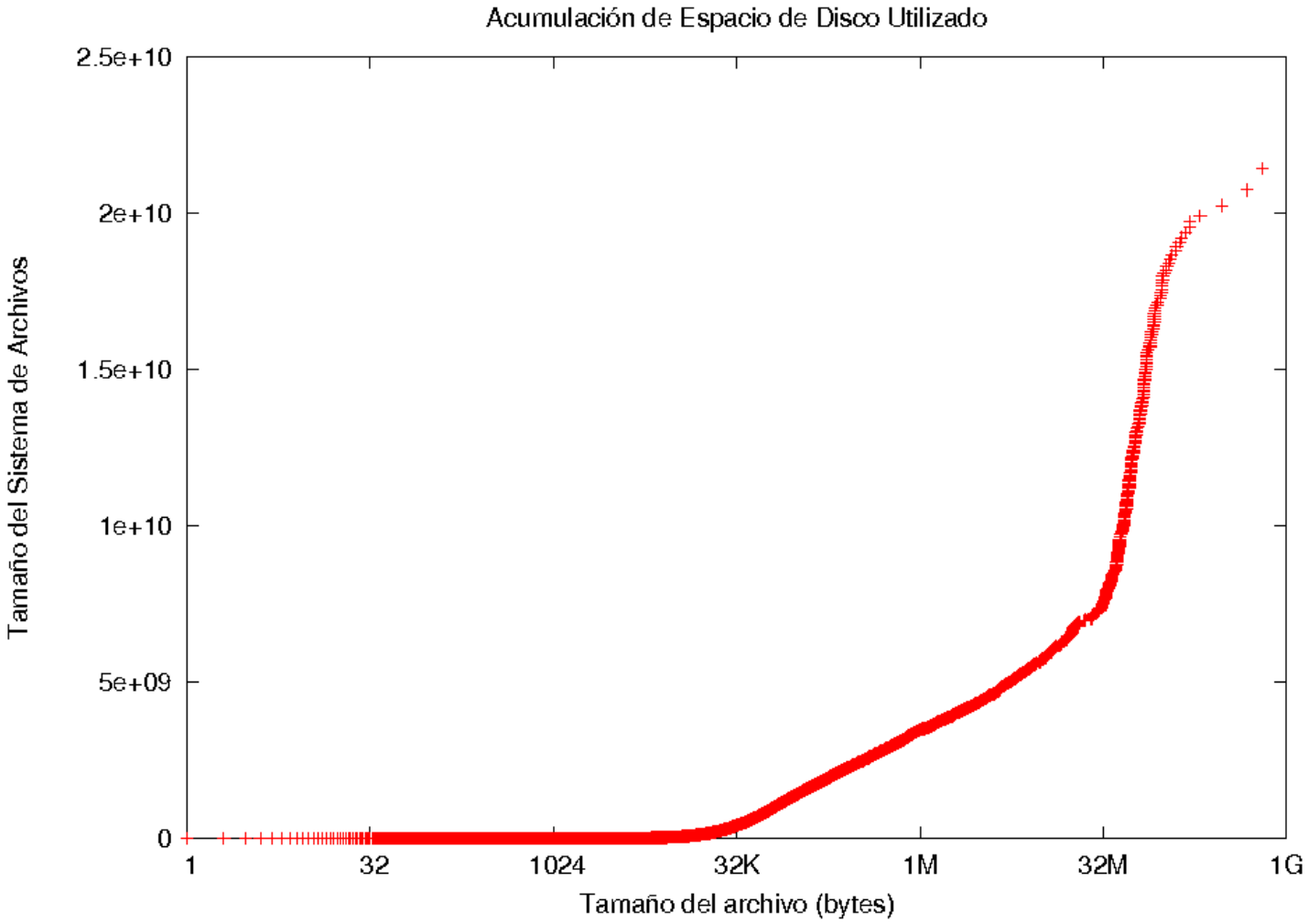
Limitaciones de Ext2/Ext3

- Desempeño es muy bueno
  - Ext3 acelera recuperación ante fallas
  - Ext3 aumenta confiabilidad del sistema
- Diseñados para
  - Discos “pequeños”
  - “Bajo” número de archivos
  - Archivos “pequeños”
  - Directorios con “bajo” número de archivos

Tamaños de archivos



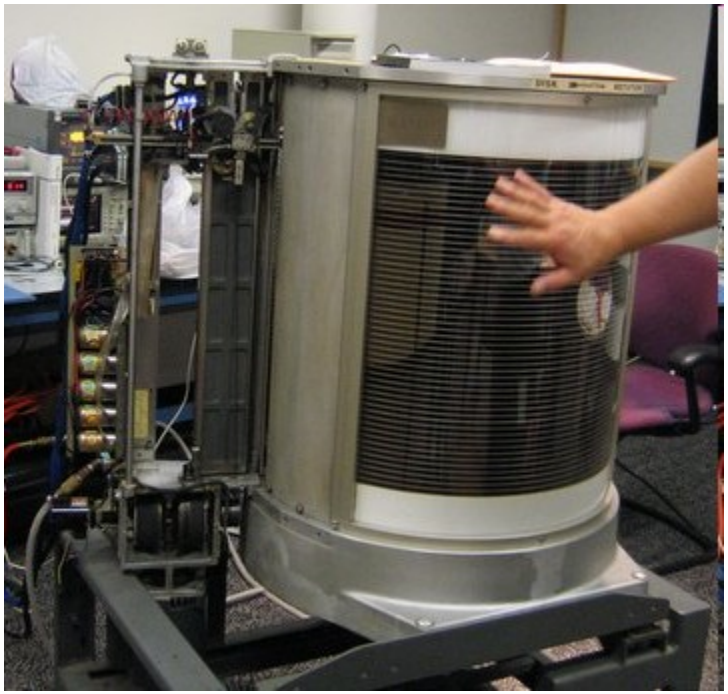
Tamaños de archivos



Evolución de los discos duros

IBM RAMAC 305

- Primer disco magnético
- Salió en 1956
- 50 discos en una torre
- Cada disco de 100 KiB
- Costo del arriendo mensual: US$160000



Evolución de los discos duros

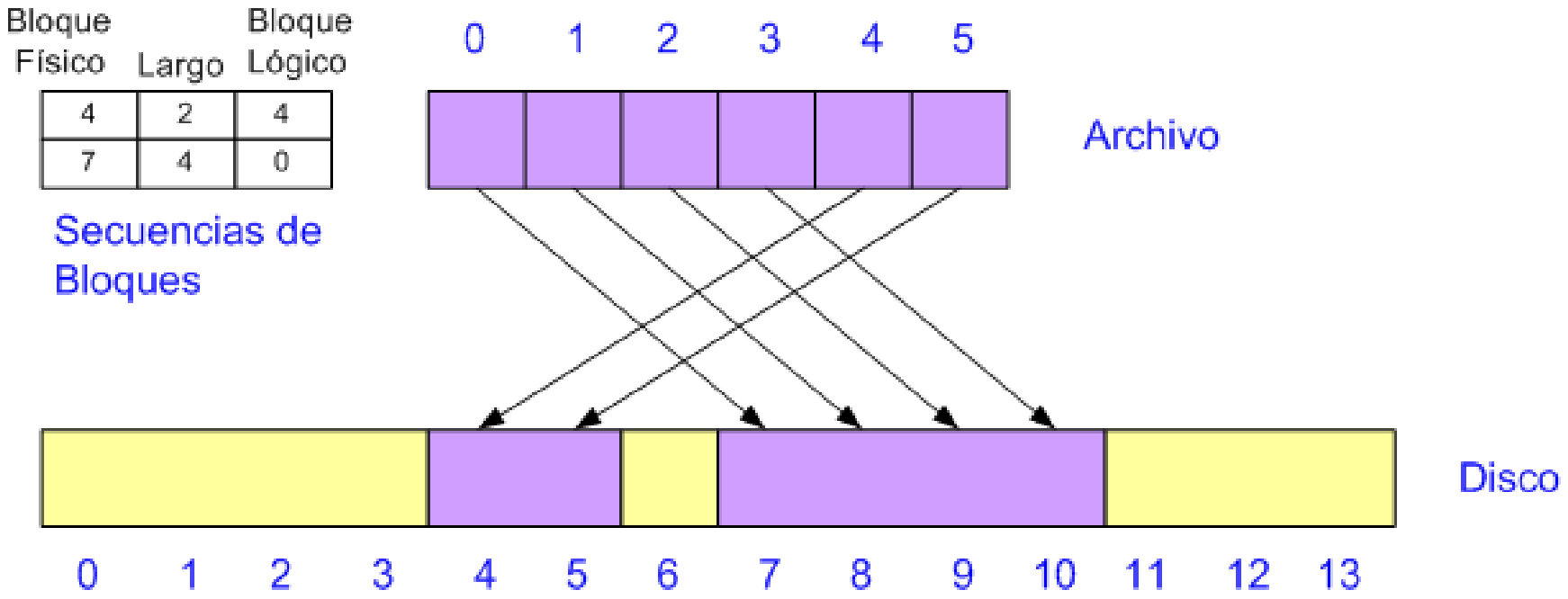
Seagate Barracuda XT modelo ST32000641AS

- SATA 6 Gb/s
- 7200 RPM
- 64 MB cache
- Sólo US$300

El cuarto sistema de archivos extendido (Ext4)

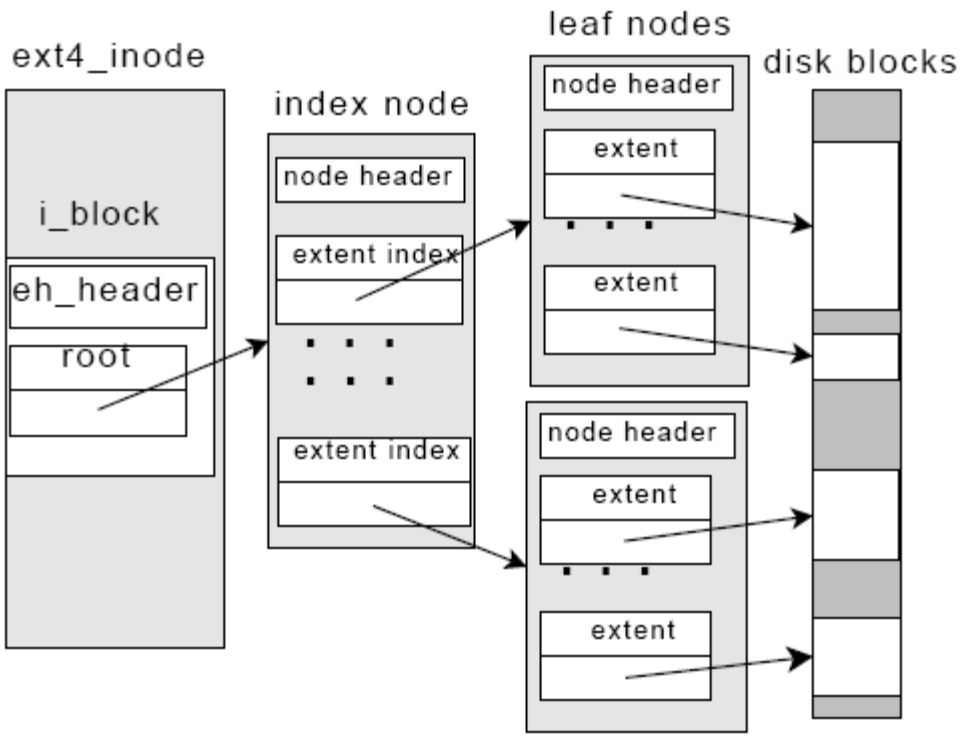
- Anunciado en 2006
- Soporte para sistemas de archivos de gran tamaño usando secuencias de bloques (*extents*)
- Número de bloque se aumenta a 48 bits
- Permite sistemas de archivos de 1 EiB
- Permite archivos de hasta 16 TiB
- Aumenta número máximo de subdirectorios de 32000 en Ext3 a 640000

Secuencias de bloques



Secuencias de bloques (*extents*)

- E*xtent* representa hasta 128 MiB
  - 4 extents en el nodo-i
  - Raíz de árbol de *extents* en nodo-i
- Nodo-i por omisión tiene 256 bytes



Número de bloque de 48 bits

- Ampliar contadores y punteros a 48 bits
- Superbloque
  - Contadores de bloques totales y libres
- Descriptor de grupo
  - Punteros a bitmaps y tablas de nodos-I
- Bitácora
  - Debe almacenar direcciones de bloques como 48 bits

Asignación multibloques

- Ext3 asigna un nuevo bloque de disco a la vez
  - Implica revisar bitmap de bloques libres
  - Operación costosa
- Ext4 permite realizar asignaciones de múltiples bloques en una operación
  - Reduce uso de CPU
  - Uso más eficiente de bloques libres

Asignación diferida

- Posterga la asignación de bloques libres en disco hasta el tiempo de writeback
- Se usa en conjunto con la asignación multibloque
  - Reduce fragmentación del sistema de archivos
  - Reduce el uso de la CPU
  - Reduce actualizaciones para archivos temporales
  - Escritura puede demorarse hasta 60 s.

Preasignación persistente

- Permite preasignar espacio en disco para un archivo que se sabe va a crecer
  - Asegura asignación contigua de bloques de disco
  - Mejora desempeño
  - Reduce fragmentación
  - Útil para aplicaciones como p2p
  - Importante para operaciones de tiempo real

Otras mejoras

- `fsck` no revisa grupos o nodos-i vacíos
- Marcas de tiempo con granularidad de nanosegundos
- Checksums de la bitácora
- Agranda nodos-i a 256 bytes
- Reserva nodos-I iniciales de un directorio
- Permite deshabilitar bitácoras
- Desfragmentación online

Convirtiendo Ext2/3 a Ext4

- Para convertir Ext3 a Ext4

```
tunefs -O extents,uninit_bg,dir_index /
  dev/DEV
```

- Ejecutar obligatoriamente `fsck`

```
e2fsck -pDf /dev/DEV
```

- Para convertir Ext2 a Ext4, además es necesario crear una bitácora

```
tune2fs -j /dev/DEV
```

Sistema de archivos XFS

- Desarrollado por SGI para IRIX
- Uso extensivo de árboles B+ y *extents*
  - Bloques libres y ocupados, directorios, etc.
- Sistema de archivos de 64 bits
  - Tamaño máximo teórico: 16 EiB
  - Tamaño máximo de un archivo: 8 EiB
  - Bloques de datos de 512 B a 64 KiB
  - Limitaciones en Linux: 16 TiB

Sistema de archivos XFS

- Disco dividido en grupos de asignación
  - Cada grupo maneja sus bloques libres y nodos-i
  - Punteros internos de 32 bits
- Bitácora de metadatos almacena operaciones lógicas
- Soporte para archivos ralos
- Asignación diferida y dinámica de nodos-i

Bitácora física vs lógica

- Bitácora física
  - Bitácora almacena bloques de datos que contienen metadatos modificados
  - Facilita la recuperación de información
  - Desperdicia espacio en la bitácora
- Bitácora lógica
  - Almacena sólo cambios a los metadatos
  - Tamaño de bitácora mucho menor
  - Código más complejo

Ext4 vs XFS

- XFS es full 64-bit
- Código fuente es 100K líneas
- XFS hereda código y compatibilidad con IRIX
- Máximo 16 EiB
- Bitácora lógica

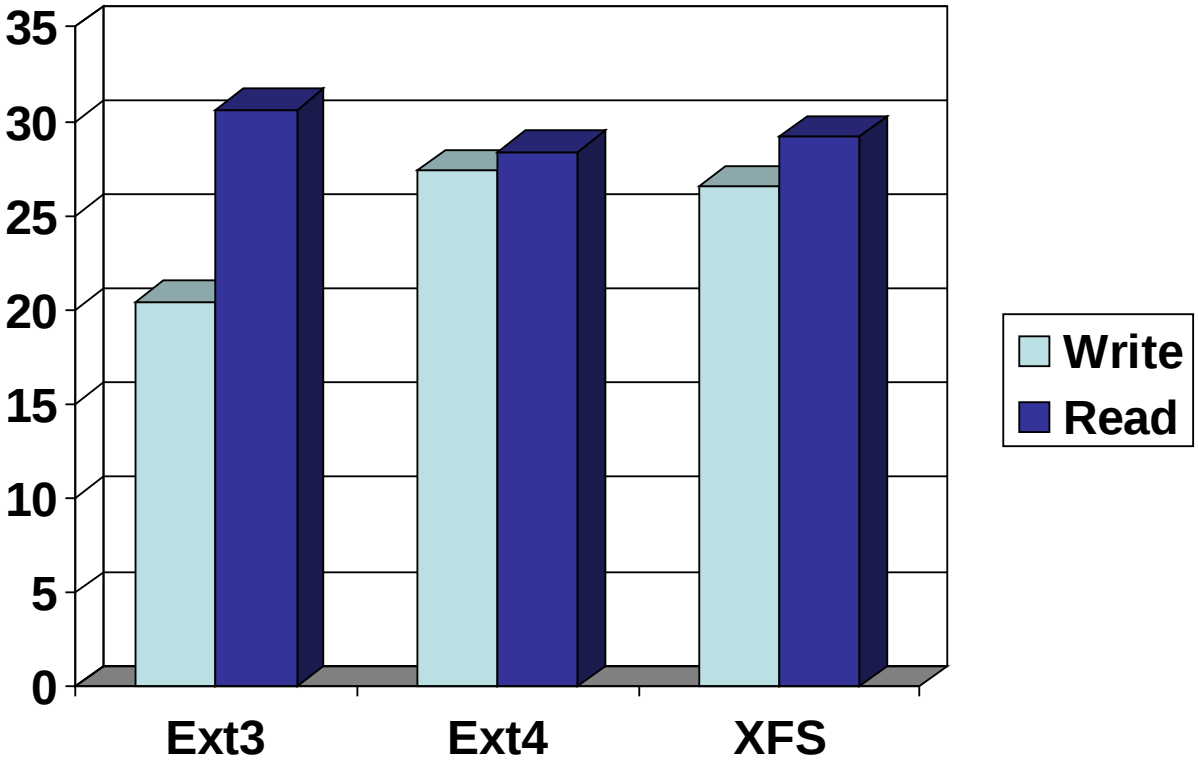
- Ext4 aún no los es
- Código fuente es 25K líneas
- Ext4 hereda robustez y estabilidad de Ext3 y Ext2
- Máximo 1 EiB
- Bitácora física

ReiserFS/Reiser4

- Sistema de archivos con bitácoras
- Árboles almacenan todos los objetos del sistema
- Excelente desempeño para archivos pequeños
- Soporte para archivos ralos
- Desarrollo detenido por arresto de Hans Reiser por asesinar a su mujer

Desempeño

- Iozone v.3.322, Disco Seagate 80GiB, 7200RPM



Desempeño en la vida real

- Benchmarks de Phoronix, Dic. 2008
- Comparación de
  - Ext3
  - Ext4
  - XFS
  - ReiserFS
- Resultado: no hay diferencias!

- World of Padman
- Unreal Tournament
- 7-Zip Compression
- Parallel BZip2 Compression
- LZMA Compression
- LAME MP3 Encoding
- FFmpeg
- GnuPG
- OpenSSL
- Bork File Encrypter

Futuro de Ext4

- Soporte para archivos ralos
- Soporte para archivos pequeños
- Soporte para archivos muy fragmentados
- Archivos mayores a 16 TiB
- Creación y asignación dinámica de nodos-i
- Nodos-i en el directorio

BTRFS

- Sistema de archivos para Linux en desarrollo por Oracle
  - Copy-On-Write
  - Snapshots
  - Manejo dinámico de dispositivos
  - Compresión transparente
  - Balance de carga
  - Manejo dinámico de volúmenes

Conclusiones

- Ext4 es un sistema de archivos
  - Compatible con sistemas Ext2 y Ext3
  - Robusto, eficiente y de buen desempeño
  - Apto para medios de almacenamiento masivos
- Ventajas se notan en sistemas con gran número de archivos y/o directorios, que ejecutan aplicaciones de alto desempeño