



Sistema de Archivos

File system

Estructura

Course

Operating System (with focus on Security)

Instructor

Acosta Bermejo Raúl

Lecture notes





Table of contents (outline)

Tabla de contenido

1. Introducción
2. DOS/Windows
3. UNIX
4. SA de red
5. Varios formatos (Mac, ópticos).





Introducción

Temario





Introducción

Temario

1. Definiciones: Modelo: acceso secuencial, aleatorio, registro.
2. Operaciones con archivos
 - Llamadas al sistema, estructura de datos (POSIX)
3. Límites del sistema
 - Número máximo de archivos
 - Tamaño máximo de archivo, etc.
4. Tipos de Archivos
 - Archivos
 - Directorios
 - Binarios, ejecutables (a.out, ELF, comando *file*), etc.
5. Arquitectura del *File System* según el SO
 - DOS (FAT), NTFS
 - UNIX (i-nodo, ext, ext2)
 - Mac (HFS Plus),
 - UFS, ISOFS, VFS, NFS





Introducción

Referencias web

1. http://en.wikipedia.org/wiki/File_system
Vistazo general
2. <http://www.forensics.nl/filesystems>
Artículos en PDF de temas como SAs con cifrado, integridad (ACID), etc.
3. http://en.wikipedia.org/wiki/List_of_file_systems
Lista de sistemas de archivos
4. http://en.wikipedia.org/wiki/Comparison_of_file_systems
Compara **100** SAs en tamaño de archivos.





Introducción

Arquitectura

1. Estructura del Disco Duro (DD)
 - Boot sector.
 - Tabla de particiones.
2. Sistema de archivos
 - La estructura de la información en el DD.
3. Propiedades (Icono, Acciones asociadas al Archivo)

Temas particulares

- Recuperar los archivos borrados: en algunos no se puede (por metodos estandar) y como resolver el problema.
- Arquitectura de Unix (Linux, Mac OSX), Windows, VFS.





Introducción

Definición

File system

- It is used to control **how data is stored and retrieved**. It separates the data (group of data called a **file**.) into pieces and giving each piece a name, the information is easily isolated and identified.
- The structure and logic rules used to manage the groups of information and their names is called a **file system**.
- There are many different kinds of file systems. Each one has different structure and logic, properties of **speed, flexibility, security, size** and more.
- Some file systems have been **designed to be used for specific applications**. For example, the ISO 9660 file system is designed specifically for optical discs.





Introducción

Características

Los FS modernos consideran

- Ligas (soft, hard)
- Fragmentación
- Compresión.
- Cifrado.
- Seguridad
- Transacciones.
- Journaling. Uso de bitácoras y transacciones para recuperación, de por ej. errores.
- Quotas. Se pueden poner límites al espacio en disco que puede usar un usuario.
- Resizing. De particiones.





DOS / Windows

Sistema de archivos

FAT

Directorio raiz

Operaciones avanzadas





DOS/Windows

Historia y tipos

Microsoft Windows

File Allocate Table (FAT)

- Heredado del sistema más viejo DOS.
- Hay 3 tipos:
- FAT12, FAT16, FAT32 y VFAT.

New Technology File System (NTFS)

- Introducido en Windows NT.
- Está presente en todas las versiones mayores de windows.

Resilient File System (ReFS)

- Es el último sistema desarrollado y presente en los servidores Windows 8.
- Usa arboles B+- y es tolerante a fallas mediante el mecanismo Copy-on-Write (CoW).





DOS/Windows

Modelo de datos

Un archivo es la unidad básica de datos del SA y este se implementa como:



Clustes que lo constituyen

<3, 4, 345, 346>

No se almacena de
Forma continua
(fracciona)

Como se guardan y asocian
ambas secuencias?

Una lista
Una tabla
Un árbol

Para llevar el registro de lo
ocupado/libre

Este espacio de
administración se
“desperdicia”

Disco duro



Secuencia de Clusters
que lo constituyen

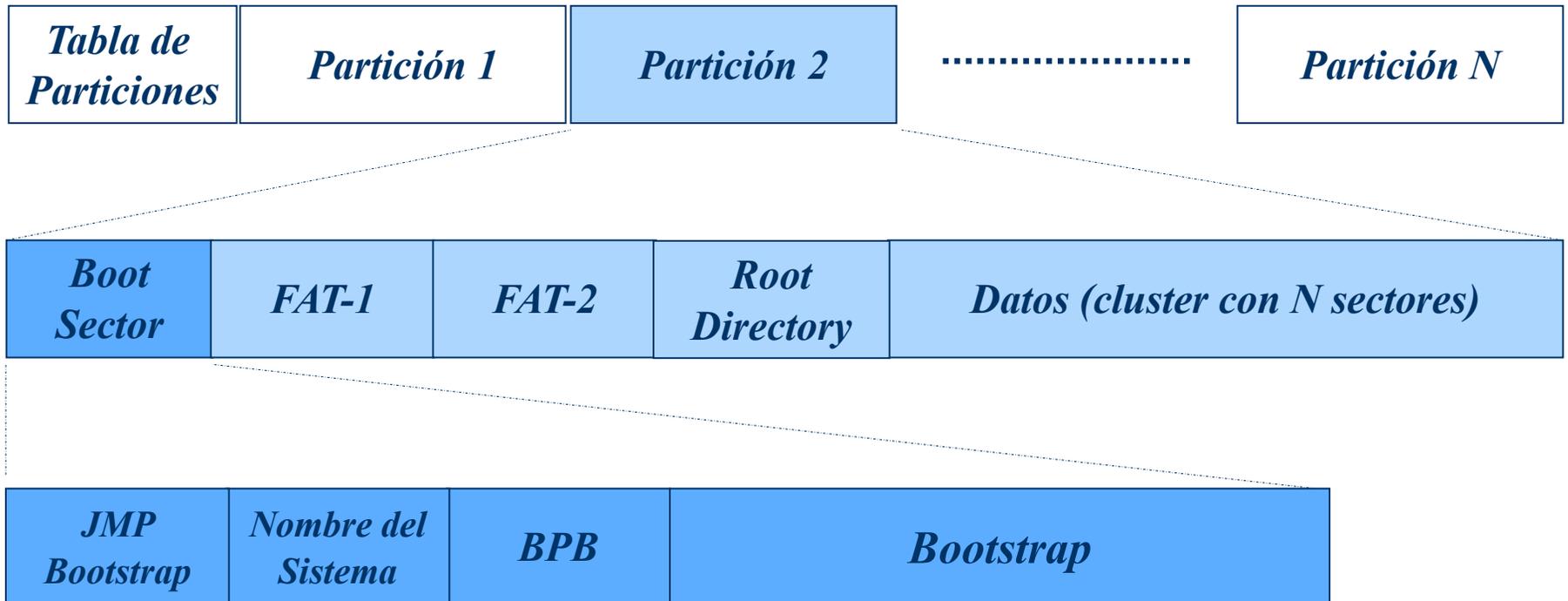
<0, 1, 2, 3, ..., MAX>





DOS/Windows

Sistema de archivos





DOS/Windows

Sistema de archivos

File Allocate Table (FAT)

- Un arreglo de celdas (12 o 16 bits) que describen el estado de cada cluster.
- Clusters 0 y 1 contienen el tamaño del DD y el Id del tipo de formato.

Bios Parameter Block (BPB)

- Contiene, principalmente, las organización del DD.
- Ejemplo:
 - #Bytes x Sec
 - #Sec x Clu
 - #Bytes x Sec
 - #Secs x FAT
 - #Sec x Pista
 - Nombre del Volumen
 - Tipo de Sistema





DOS/Windows

FAT

Modelo de la FAT

La tabla almacena el estado de los clusters y su relación:

indice	Cluster
0	Libre
1	Ocupado por A ₁
2	Ocupado por A ₂
3	Ocupado por B ₂
4	Ocupado por B ₁
...	
Max	

Tamaño de la celda

$$2^{12} = 4,096$$

$$2^{16} = 65,536$$

$$2^{32} = 4,294,967,296$$

Orden





DOS/Windows

FAT

- Para saber los clusters que constituyen un archivo se recorre la FAT dando saltos para obtener la **lista** (cadena) de clusters.
- El estatus del cluster depende de su valor:
 - 00h Libre.
 - E5h Borrado.
 - 2Eh Subdirectorío
 - FFh Fin de archivo

0	1	2	3	4	5	6	...	14	...
			4	6		14		F	

File Chain





DOS/Windows

Directorio raiz (root directory)

Esta formado por entradas de 32 bytes:

<i>Campos</i>	<i>Tamaño (bytes)</i>
File Name	8
File Extension	3
File Attributes	1
Reservado	10
Time of Last Update	2
Date	2
Begininig Cluster	2
File size	4





DOS/Windows

Operaciones avanzadas

Existen herramientas que permiten:

1. Cambiar el tamaño de una partición: si la contigua tiene espacio libre o no se utiliza.
2. Dividir una partición en 2: utiliza el espacio libre.

Ejemplos son:

- Gparted





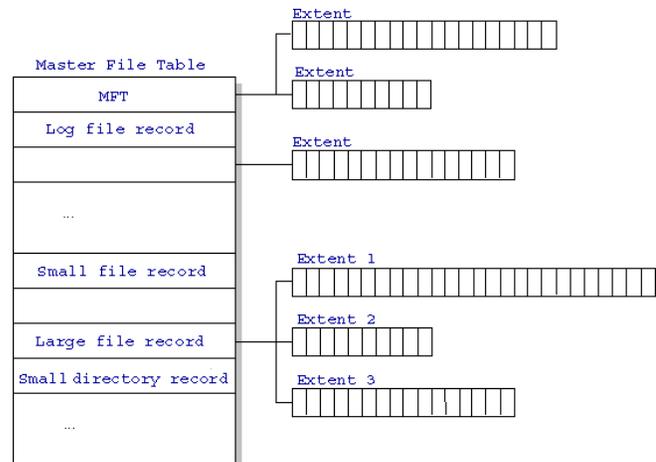
DOS/Windows

NTFS

Con la creación de Windows NT surgió un nuevo sistema de archivos denominado NTFS.

Links

- <https://en.wikipedia.org/wiki/NTFS>
- <http://www.c-jump.com/bcc/t256t/Week04NtfsReview/Week04NtfsReview.html>
- [https://technet.microsoft.com/en-us/library/cc781134\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc781134(v=ws.10).aspx).



MFT records

	Seq	File address	
312	[...]	0x0040	0040 0000 0138
313	[...]	0x0001	0001 0000 0000 0139
314	[...]	0x000a	000a 0000 0000 013a
315	[...]	0x0003	0003 0000 0000 013b
316	[...]	0x0003	0003 0000 0000 013c

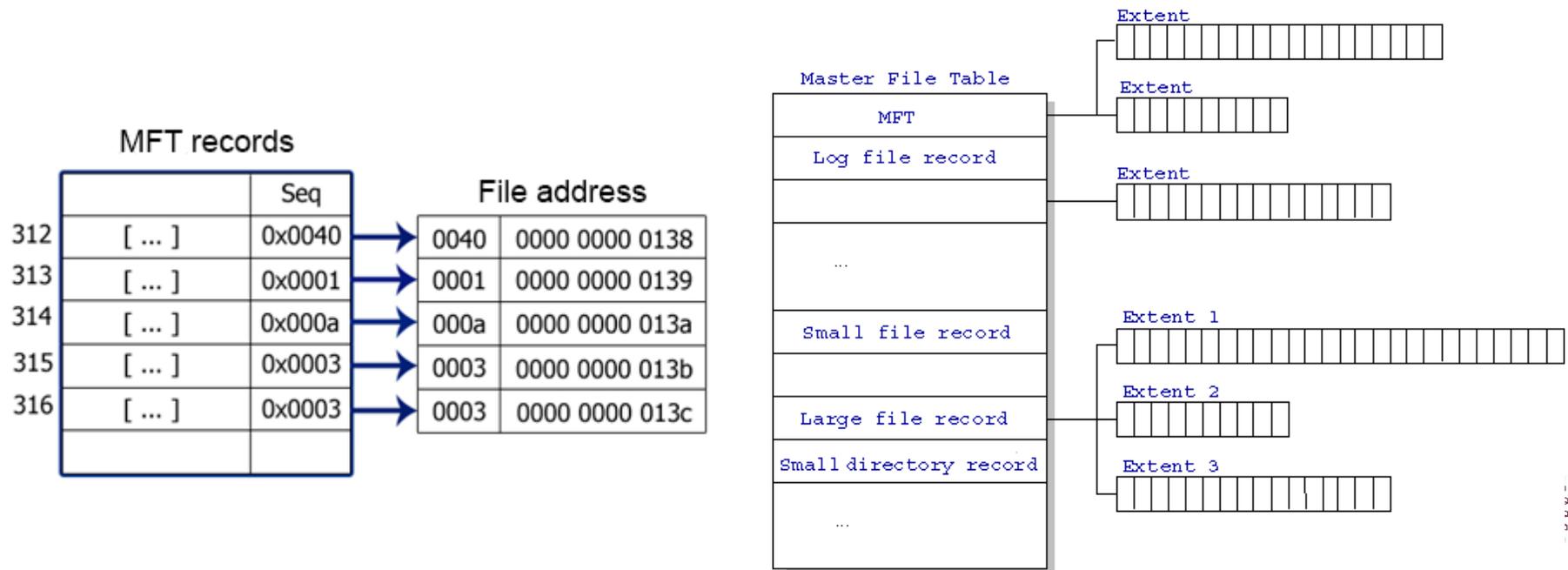




DOS/Windows

NTFS

El diseño central usa MFT records que son como los i-nodos de Unix que contienen la información de: permisos, apuntadores, etc.



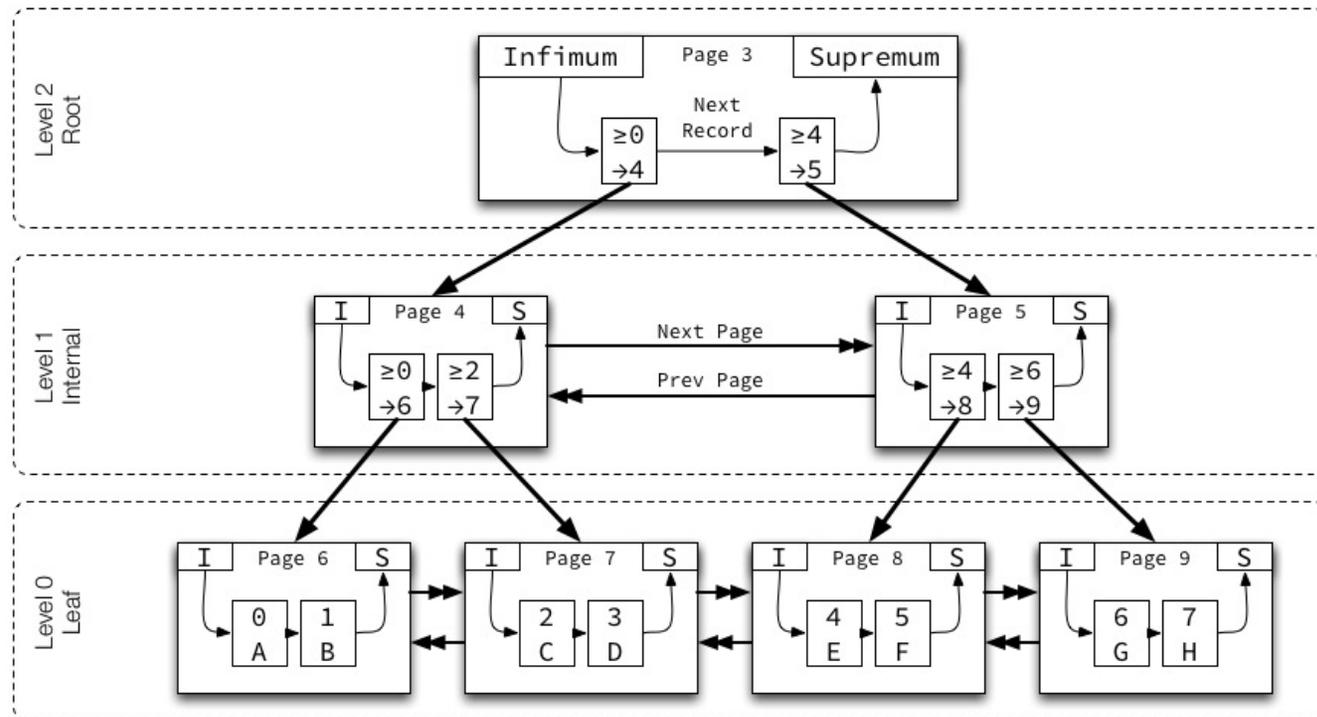


DOS/Windows

ReFS

Diseño

B+Tree Structure





UNIX

Sistema de archivos
Estructura de un archivo
Características
Estructura de un directorio
Tipos de sistemas de archivos
Llamadas al sistema y funciones
Comandos
Tablas de acceso a archivos
Operaciones avanzadas
Prácticas





UFS

Unix File System (UFS)

Historia

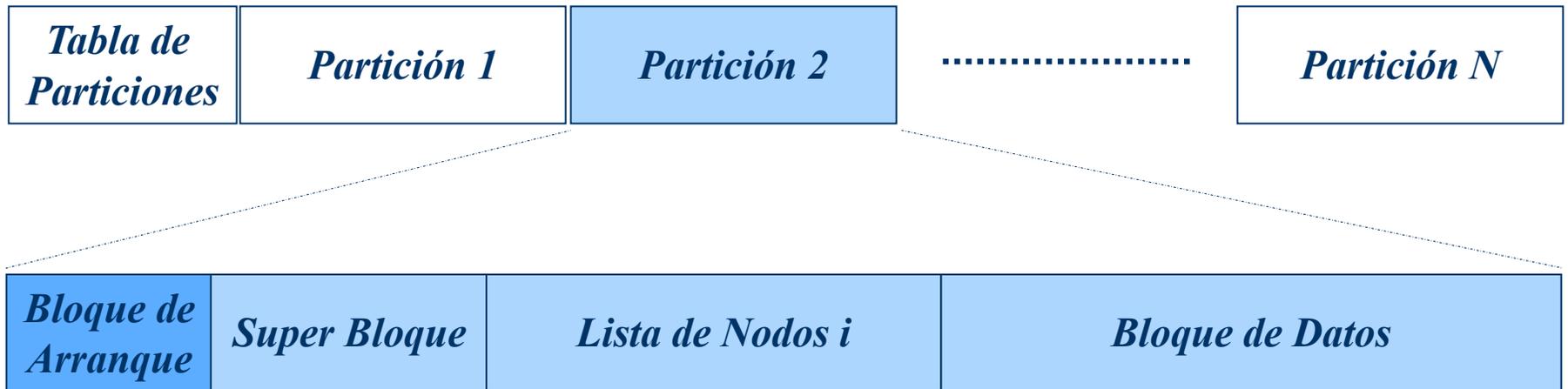
- Es el sistema de archivos utilizado por la mayoría de los UNIXs o like-UNIXs.
- También se le conoce como:
 - Berkeley Fast File System o
 - Fast File System (FFS).
- Es una evolución del sistema de archivos del UNIX Versión 7.
- Algunas variantes de este sistema se implementaron en otros SO, por ejemplo:
 - ZFS en Solaris.





UNIX

Sistema de archivos



BPB

Equivalente a la
FAT





UNIX

Sistema de archivos

Bloque de Arranque: Programa que busca el SO y lo carga en memoria.

Super Bloque: Contiene las características del SA como tamaño, número máximo de archivos, espacio libre y ocupado (listas de nodos-i y de bloques), etc.

Nodos-i: Contienen la características de los Archivos como propietario, permisos de acceso, ubicación en el disco, tipo de archivo, Apuntadores a Bloques de Datos, etc. Nodo-i(1)Bloques Defectuosos, Nodo-1(2)/Dir Raiz

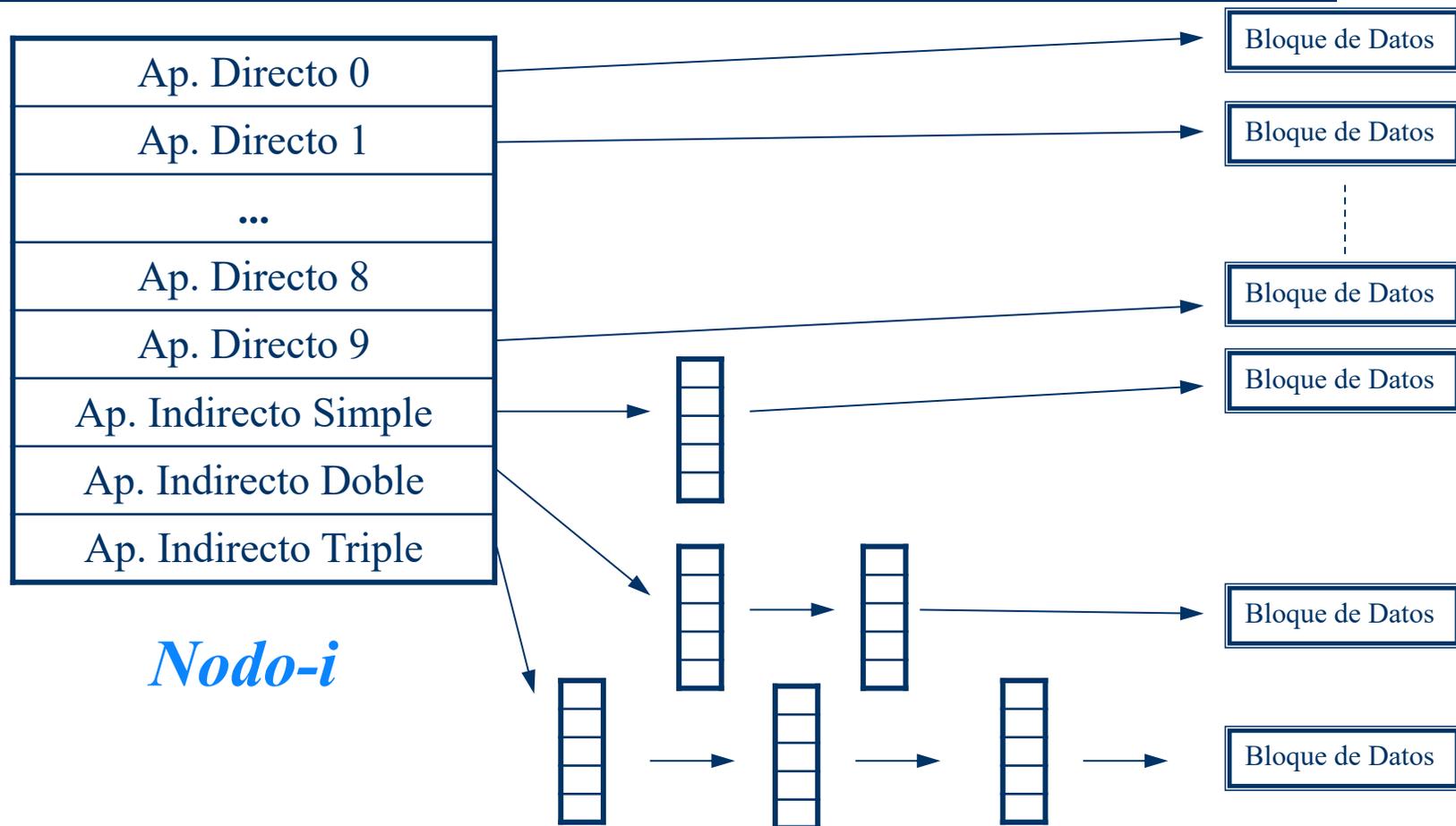
Bloques de Datos: Contienen la información de los Archivos. Los bloques de datos son asignados de forma exclusiva a cada archivo.





UNIX

Estructura de un archivo





UNIX

Características – Tamaños

Dada la información anterior de número de apuntadores y tamaños de bloque se tiene:

Apuntador 32 bits

Bloque de Datos = 1Kb

10 Apuntadores Directos = 10 Kb

1 Ap. Ind. Simple = 256 Kb

1 Ap. Ind. Doble = 64 Mb

1 Ap. Ind. Triple = 16 Gb

Como se calcula el:

Tamaño total de un archivo?

Número máximo de archivos en un directorio?





UNIX

Características

- Como las operaciones con archivos utilizan mucho el Superbloque y la lista de nodos-i, estos se copian en memoria para agilizarlas.
- Para asegurar la consistencia de la información existe un *demonio* que actualiza la información en el DD periódicamente.
- Existen comandos para asegurarse que la información del SA es consistente, por ejemplo: `shutdown`, `sync`.

Tipos de Archivos:

Ordinarios o de Datos

Directorios

Dispositivos

Especiales (pipes o fifos)





UNIX

Estructura de un directorio

Los directorios son manejados igual que los archivos (un nodo-i y bloques de datos). El kernel es el único que puede modificar los directorios (seguridad).

Unix System V

	Nodo-i (2 Bytes)	Nombre del Archivo (14 Bytes)
0	45	.
16	2	..
32	2395	inittab
48	371	mount
64	4921	shutdown
...

Bloque de Directorio. Operaciones atómicas con en el DD.





UNIX

Estructura de un directorio

4.3 BSD

No de Nodo-i	Longitud de la entrada	Longitud del Nombre	Nombre	...	
--------------	------------------------	---------------------	--------	-----	--





UNIX

Tipos de sistemas de archivos

El sistema de archivos de BSD ha evolucionado de la siguiente manera:

1. Ext.- Es el nombre que se le da a la primera versión del SA de UNIX en MINIX la cual fue remplazada por 2 propuestas xiafs (actualmente obsoleta) y el Ext2.
2. Ext2.- Es la implementación más básica de Linux.
3. Ext3.-Journaling. Todas las operaciones se hacen con una bitácora. Es posible hacer "undos".
4. Ext4.- Anunciada en el 2006 y liberada en el 2008 por Theodore Ts'o creador de la Ext3. No ha sido adoptada completamente en Linux y se ha propuesto usar Btrfs (entre otros).





UNIX

Llamadas al sistema

Llamadas al sistema

- Creación y contenido
 - creat, open, read, write, lseek, close, dup, dup2, mknod.
 - remove, rename, sync, fsync
- Permisos y máscaras
 - chmod, fchmod, chown, fchown, lchown
 - stat (sticky bit), lstat, fstat, access, umask, utime
- Varios
 - link, unlink, symlink, readlink

Se verán algunos en detalle en la sesión de práctica
Ver slides anexos





UNIX

Funciones

Funciones de Archivos

fopen, fread, fwrite, fclose, fflush,
freopen, fdopen, setbuf, setvbuf, ftell, fseek, rewind, fgetpos, fsetpos,
getc, fgetc, getchar, ungetc, putc, fputc, putchar, fgets, gets, fputs, puts,
feof, ferror, (f,s,v,vf,vs)printf, (f,s)scanf

Funciones de directorios

opendir, readdir, rewinddir, closedir, seekdir, telldir, chdir, fchdir
chroot, getcwd, mkdir, rmdir





UNIX

Notificaciones

Los syscall que existen son:

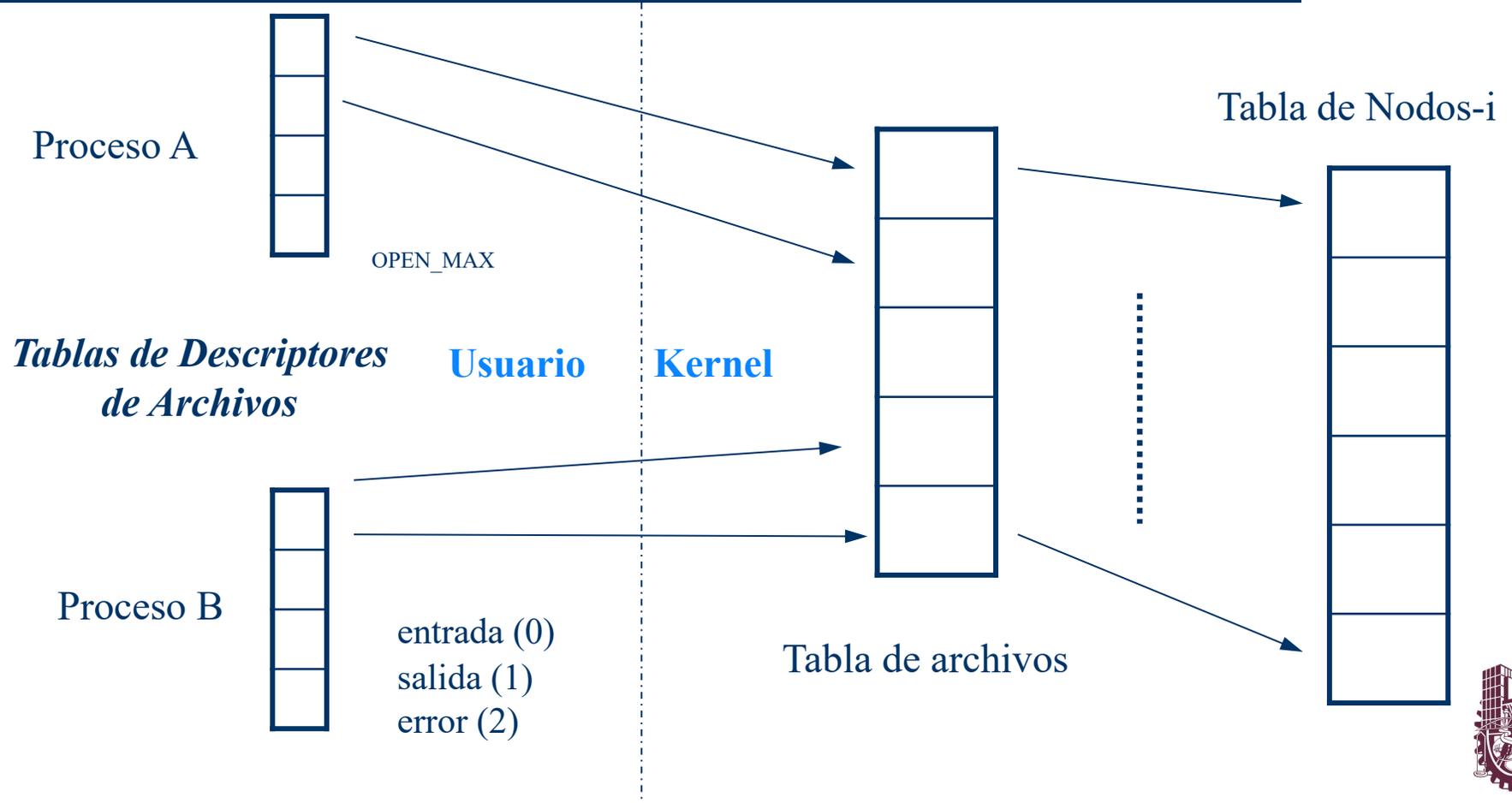
- Sólo en Linux: poll, ppoll.
Espera por un evento en un archivo (descriptor).
- Select.
Ver slides de Mecanismos de sincronización.





UNIX

Tablas de acceso a archivos





UNIX

Enlaces

- Para no tener repetido un archivos en varios lugares, UNIX utiliza los enlaces. Para efectuar un enlace se utiliza el comando *ln*.
- Existen 2 tipos de enlace: duros y simbólicos.
 - Enlace duros: Se usa el número de i-node.
 - Enlaces lógicos (simbólicos): Se usan cadenas (paths).
- Por default se crean ligas duras. Existen varias notaciones.

\$ ln Origen Destino

Ejemplos

Ligas Duras

\$ ln Target Link_name

Ligas simbólicas

\$ ln -s /path/file Nuevo-nombre





UNIX

Mount

- Es el proceso que coloca en la memoria RAM la información del disco duro y de la partición (sistema de archivos).
- Un SA no puede utilizarse sino se monta. Para ello se usa el comando mount.
- Operaciones como fsck requieren que la partición sea desmontada.
- La primera partición montada forma el árbol inicial (SA) y las siguientes que se montan formaran ramas (subcarpetas).
- En sistemas like-UNIX se guarda en el archivos /etc/fstab las particiones que se van a montar en el arranque.





Linux

Ext (historia)

LVS

VFS





Linux

Historia y tipos de FS

Los tipos son:

- Ext2 Aun se usa en dispositivos flash por su número de ciclos de escritura.
- Ext3 Es sólo una extensión de Ext2 con journaling.
- Ext4
- ReiserFS
- XFS derivado de SGI el cual fue utilizado inicialmente por IRIX.
- JFS Desarrollado por IBM y su version actual es la JFS2.





Linux

Ext2

- Liberado en 1992 para resolver 2 problemas en el minix: tamaño de la partición y de los nombres de archivos (14 car).
- Ext2 aun se usa en dispositivos flash por su número reducido de ciclos de escritura.
- No existe indexado de directorios por lo que hay problemas de rendimiento para directorios con un número grande de archivos (mas de 10 mil).
- El límite teorico de archivos en un directory es 1.3×10^{20} lo cual no es práctico.
- Límites teóricos de Ext2 en Linux:

Block size	1KB	2KB	4KB	8KB
Max. File size	16GB	256GB	2TB	2TB
Max. Filesystem size	4TB	8TB	16TB	32TB





Linux

Ext3

- Su desarrollo (por Stephen Tweedie) empezó antes de 1998 y fue liberado hasta el 2001 en el Linux 2.4.15.
- Su principal ventaja es el Journaling. Hay 3 niveles:
 - Journal (lowest risk)
 - Ordered (medium risk)
 - Writeback (highest risk)
- Varían en diferentes combinaciones para los Meta-datos y contenido de archivos. Se guarda en el journaling antes de hacer el committed. Protocolo de 2 fases.
- Algunas desventajas: sin defragmentación integrada, no se puede verificar (fsck) sin desmontarlo, no se pueden recuperar los archivos borrados.
- Algunos límites
 - Máx número de bloques 2^{32} .
 - Un directorio puede tener a lo mas 31,998 subdirectorios.

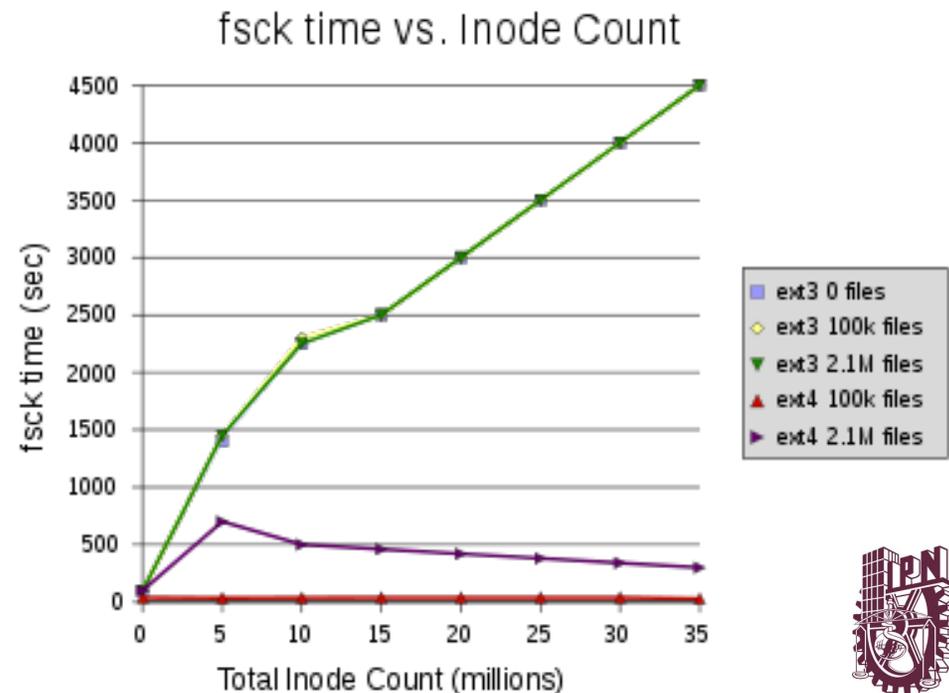




Linux

Ext4

- Su desarrollo (por Theodore Ts'o) empezó en 2006 y fue liberado hasta el 2008 en el Linux 2.6.28.
- No tiene muchas nuevas funcionalidades.
- Algunas de las nuevas características son:
 - Sistema de archivos más grande: volúmenes de hasta 1 exabyte y archivos de hasta 16 TB.
 - RedHat recomienda XFS para volúmenes mayores de 100TB.
 - Se modificó el uso de los bloques. Se creó Extents que usa un mapa de bloques más eficiente.
 - Se incrementó el límite de los subdirectorios a 32mil.





Linux

Journaling

Sistemas de archivos con registro

Al hacer alguna operación se suele realizar lo siguiente:

- Encontrar bloques libres (en lo posible lo mas continuos posible).
- Almacenar los datos.
- Marcar los bloques como ocupados.
- Crear un inodo, y referenciar los bloques.
- Guardar la informacion del archivo en el inodo.
- Marcar la operacion en el registro como completada.

Se verificar si quedo alguna **accion/entrada incompleta** en los registros y se termina.





Linux

Journaling

- Aumenta la consistencia del sistema de archivos.
- Guarda la meta-información (i-nodos nuevos, bloques liberados, etc.) en un log.
- Si el sistema falla y no se ha volcado los datos del cache al disco, se lee el log para hacer que el sistema sea consistente.
- Ejemplos de SAs con journaling:
 - XFS
 - ReiserFS





Linux

Software

Para leer sistema de archivos Exts desde otros SO se puede utilizar:

- **Explore2fs.-** Lee desde windows Ext2, Ext3, LVM, etc. Hay muchos software similares.
<http://www.chrysocome.net/explore2fs>
- Etc.





Otros formatos de Sistema de Archivos

HFS (Mac)

ISO 9660 (óptico)

UDF (óptico)





HFS

Para Mac OS X

- Existen los siguientes formatos:
 - **HFS** (Mac OS Standard)
 - **HFS plus o HFS+** (Mac OS Extended)
 - **HFSX** (Mac OS Extended)
- Herramientas para leer el FS de mac en Windows y Linux
 - hfsexplorer: <http://www.catacombae.org/hfsx.html>





UDF

Universal Disk Format

- También conocido como ISO/IEC 13346 y ECMA-167.
- Fue desarrollado por la OSTA (*Optical Storage Technology Association*).
- Es un sistema de archivos **neutral** de vendedores para almacenar datos de una computadora.
- Se usa normalmente para **DVDs** y nuevos formatos de disco ópticos.
- Se usa en lugar del ISO 9660.
- Su diseño se adapta a actualizaciones incrementales de medios ópticos grabables (re-writable)
- Varias versiones han sido publicadas:
 - La última en el 2005, la 2.60.
 - <http://www.osta.org/specs/pdf/udf260.pdf>





The end

Contacto

Raúl Acosta Bermejo

<http://www.cic.ipn.mx>

<http://www.ciseg.cic.ipn.mx/>

racostab@ipn.mx

racosta@cic.ipn.mx

57-29-60-00

Ext. 56652





ISO 9660

CD, DVD, etc.

- .





UDF

Universal Disk Format

- .





Oracle Cluster File System

Version 2

- .

