

# 1 Sistemas de arquivos

## 1.1 Sumario

- 1 Estrutura da unidade lóxica ou volume
- 2 FAT12/FAT16/FAT32
- 3 **Fat32** En realidade, FAT32 é un termo "equivocado", xa que só se utilizan os 28 bits de menor orden das direccións de disco. Debería terse chamado FAT28, pero as potencias de dous óense moito mellor.
- 4 HPFS
- 5 NTFS
  - ◆ 5.1 Versións
  - ◆ 5.2 Estrutura
  - ◆ 5.3 Características xerais do sistema de arquivos NTFS
- 6 **Journaling** O *journaling* ben sendo un rexistro que permite ao sistema de arquivos restablecer os datos afectados por unha transacción no caso de que esta falle.
- 7 Compatibilidades FAT - NTFS
- 8 exFAT
- 9 ReFS
- 10 Sistemas de arquivos Linux - Unix
  - ◆ 10.1 Estrutura dos Sistemas de arquivos Linux - *i-nodos*
  - ◆ 10.2 ext2
  - ◆ 10.3 ext3
  - ◆ 10.4 ext4
- 11 **Extents** Os *extents* foron introducidos para substituír ao tradicional esquema de bloques empregado polos sistemas de arquivos ext2/3. Un *extent* é un conxunto de bloques físicos contiguos, mellorando o rendemento ao traballar con ficheiros de grande tamaño e reducindo así a fragmentación. Un *extent* simple en ext4 é capaz de mapear ata 128MB de espazo contiguo cun tamaño de bloque igual a 4KB.
  - ◆ 11.1 ReiserFS
  - ◆ 11.2 UFS
  - ◆ 11.3 XFS
  - ◆ 11.4 JFS
  - ◆ 11.5 ZFS
  - ◆ 11.6 btrfs
  - ◆ 11.7 LVM
  - ◆ 11.8 CEPH
- 12 Sistema de arquivos de CD-ROM ISO 9660
- 13 **Nota** A norma ISO 9660 é descendente directa dun esforzo de estandarización anterior chamado **HSG (High Sierra Group)**, proposto por un conxunto de fabricantes reunidos en 1985 no hotel High Sierra, do Lago Tahoe, Nevada. Aínda que a ISO aceptou unha grande maioría das propostas do HSG, existen algunhas diferencias menores.
- 14 **Little endian vs Big endian** O termo inglés **Endianness** designa o formato no que se almacenan os datos de mais dun byte nun soporte informático. O problema é similar aos idiomas nos que se escriben de dereita a esquerda, como o árabe, fronte aos que se escriben de esquerda a dereita, pero trasladado ao almacenamento de bytes. Así o valor hexadecimal 0x4A3B2C1D codificaríase en memoria na secuencia {4A, 3B, 2C, 1D} no sistema *big endian*, e este mesmo valor codificaríase como {1D, 2C, 3B, 4A} no *little endian*, deste modo faise mais intuitivo o acceso a datos porque vai de menos relevante a mais relevante.
- 15 Particións Cifradas
- 16 Benchmarks de Sistemas de Arquivos
  - ◆ 16.1 dbench

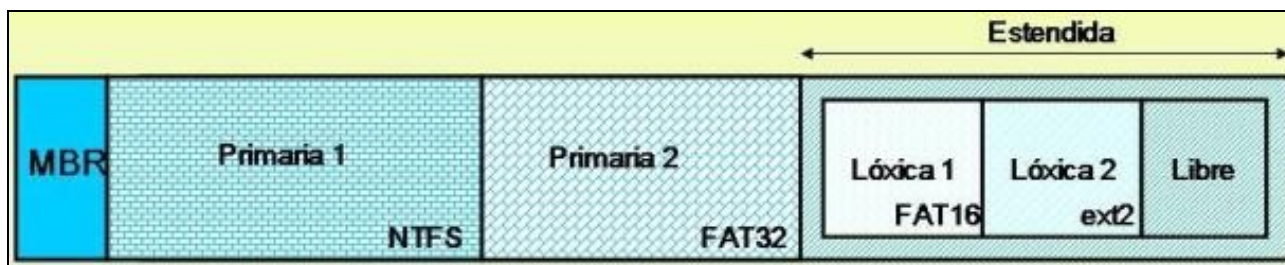
## 1.2 Estrutura da unidade lóxica ou volume

Unha vez feitas as particións hai que proceder a **formatearlas**, que é darlle unha estrutura axeitada para que o SO poida almacenar información no HD.

Dependendo do **Sistema de Arquivos** ou tipo de "formato" empregado variará así o contido de dita estrutura.

Os Sectores agrúpanse para formar **Clusters** que son as unidades mínimas nas que se pode gardar un arquivo.

1 Cluster =  $2^n$  Sectores (sendo  $n = \{0, 1, 2, 3, 4, 5, 6\}$ )



• As funcións básicas do Sistema de Arquivos son:

- ◊ Control do espazo dispoñible e asignado.
- ◊ Mantemento de directorios e nomes de arquivos.
- ◊ Control do lugar onde se encontran almacenadas fisicamente as distintas porcións dun arquivo.

• **Sistemas de Arquivos** actuais:

FAT - FAT32 - NTFS - Ext2 - Ext3 - ReiserFS - ...

### 1.3 FAT12/FAT16/FAT32

Cada partición formateada neste sistema de arquivos terá a seguinte estrutura:

BOOT	FAT (Dúas copias)	Directorio raíz (32 sectores)	Datos e directorios
------	-------------------	-------------------------------	---------------------

• **BOOT (Sector de Arranque):** Ó formatear un volume, créase sempre no primeiro sector do volume o sector de arranque. Nel atópase información acerca do tamaño, da estrutura do volume e sobre todo do **BOOTSTRAP-LOADER**, mediante a cal se pode arrancar o PC dende o SO.

• **FAT (File Allocation Table - Táboa de Asignación de Ficheiros):** Cada entrada da táboa corresponde a un *cluster*, en cada entrada indícase: Se o *cluster* está baleiro; se o *cluster* está defectuoso; cal é o *cluster* seguinte que ten parte do arquivo; se é o último *cluster* que ten información do arquivo. (Vemos un exemplo na imaxe de abaixo).

◊ O tamaño da FAT depende do sistema de arquivos empregado:

- FAT12 implica que da direccionado  $2^{12}$  clusters
- FAT16 implica que da direccionado  $2^{16}$  clusters
- FAT32 implica que da direccionad  $2^{28}$  clusters

### 1.4 Fat32

En realidade, FAT32 é un termo "equivocado", xa que só se utilizan os 28 bits de menor orden das direccións de disco. Debería terse chamado FAT28, pero as potencias de dous óense moito mellor.

Isto fai que se ten que adaptar o tamaño dos sectores dependendo do tamaño da partición. Así, na seguinte táboa vemos os tamaños máximos teórico das particións dependendo do tamaño do *cluster*:

Tamaño de <i>cluster</i>	Partición FAT 12	Partición FAT 16	Partición FAT 32
0,5 KB	2 MB	32 MB	
1 KB	4 MB	64 MB	
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB	32 MB	512 MB	2 TB

Tamaño de <i>cluster</i>	Partición FAT 12	Partición FAT 16	Partición FAT 32
16 KB		1 GB	4 TB
32 KB		2 GB	8 TB

- ◊ O tamaño da partición está limitado, pois hai que ter en conta que non se admiten *clusters* maiores de 32 KB (¡Olo! Aplicacións como Partition Magic permiten formatear con *clusters* de 64 KB).
- ◊ En teoría, FAT32 soporta particións máis grandes de 2 TB. Sen embargo, debido a unha limitación no BOOT, que utiliza unha variable de 32 bits para numerar os sectores físicos presentes no disco duro, **o límite real dunha partición en FAT32 é de 2 TB**.
- ◊ As utilidades incluídas en Windows permiten crear e formatear unidades FAT32 de tamaño máximo 32 GB. Sen embargo, poden acceder en modo lectura e escritura a volumes FAT32 de ata 2 TB creados usando outras utilidades.

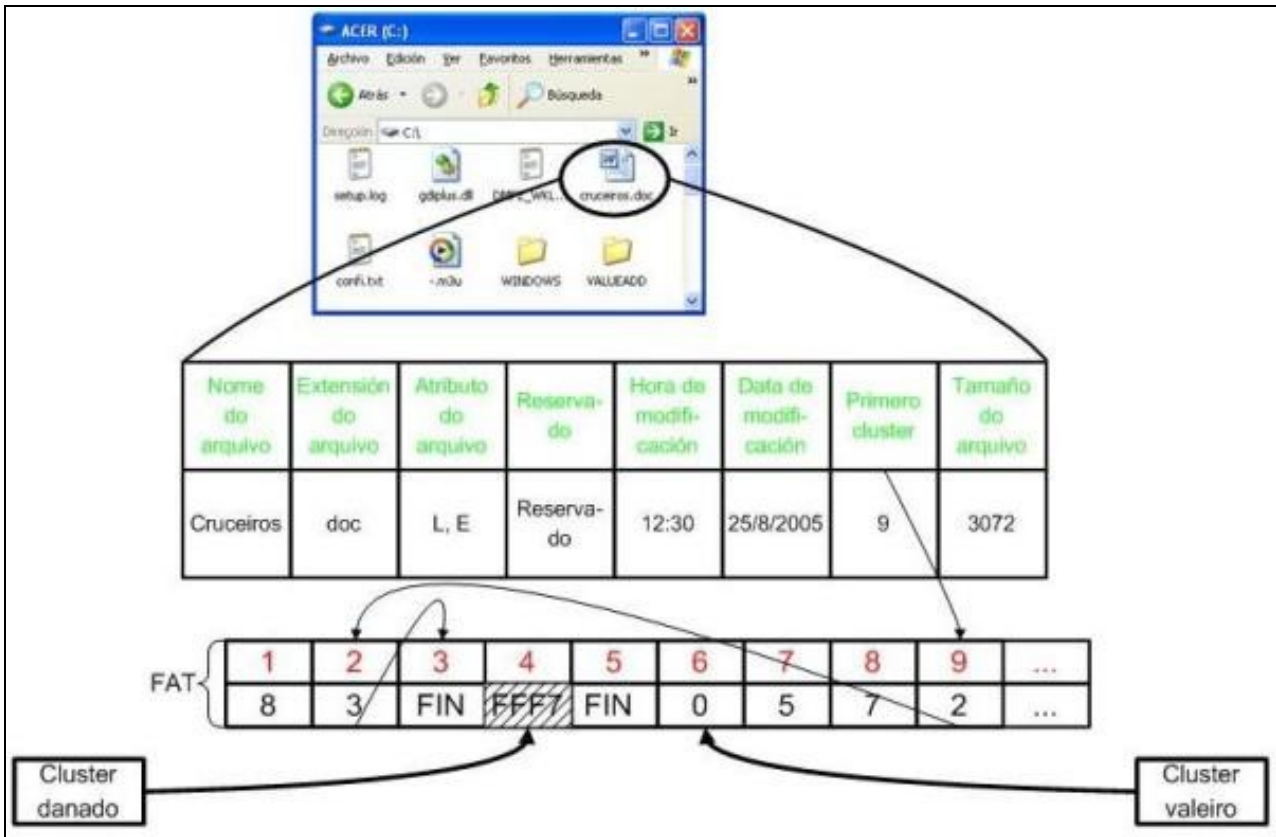
#### Tamaño máximo arquivo

Partición FAT 12	Partición FAT 16	Partición FAT 32
32 MB	2 GB	4 GB

- **Unha ou mais copias da FAT:** Normalmente fanse dúas copias idénticas da FAT. Se o SO atopa un erro na FAT primaria, copia a secundaria para evitar a perda de datos.
- **O directorio Raíz:** Ten un tamaño fixo que se establece cando se formatea o disco.
  - ◊ Contén unha entrada por cada ficheiro que se almacena no directorio raíz do sistema.
  - ◊ Cada entrada é de 32 bytes e componse dos seguintes campos:

nome ficheiro <b>8 bytes</b>	extensión <b>3 bytes</b>	atributos <b>1 byte</b>	reservado <b>10 bytes</b>	hora creación ou modificación <b>2 bytes</b>	data creación ou modificación <b>2 bytes</b>	primeiro clúster <b>2 bytes</b>	tamaño <b>4 bytes</b>
------------------------------------	-----------------------------	----------------------------	------------------------------	--	--	---------------------------------------	--------------------------

- **A Zona de Datos:** É a parte do disco duro na que se almacenan os datos dun arquivo. Esta zona depende en case a súa totalidade das interrelacións entre as estruturas de datos que forman o sistema de arquivos do SO, e do camiño que se leva dende a FAT cara os diferentes sectores dun arquivo. Os *clusters* enuméranse para poder ser accedidos polo SO.



## 1.5 HPFS

- HPFS acrónimo de *High Performance File System* (Sistema de Arquivos de Altas Prestacións), foi creado por *Microsoft* especificamente para o sistema operativo OS/2 para mellorar as limitacións do sistema de arquivos FAT.
- Caracterizábase por permitir nomes longos, metadatos e información de seguridade, así como de autocomprobación e información estrutural.
- Outra das súas características é que, aínda que posuía táboa de arquivos (como a FAT), esta estaba posicionada fisicamente no centro da partición, de tal xeito que redundaba en menores tempos de acceso á hora de lela/escrivila.

## 1.6 NTFS

### 1.6.1 Versións

Hai tres versións de NTFS:

- ◊ v1.2 ou v.4.0 en NT 3.51 e NT 4
- ◊ v3.0 ou v5.0 en Windows 2000
- ◊ v3.1 ou v5.1 en Windows XP e Windows 2003 Server

### 1.6.2 Estrutura

Cada partición formateada neste sistema de arquivos terá a seguinte estrutura:

<b>BOOT</b>	<b>Superbloque</b>	<b>Mapas de bits</b>	<b>Descritores físicos de arquivos</b>	<b>Datos e directorios</b>
-------------	--------------------	----------------------	--	----------------------------

- **Boot:** Sector de arranque.
- **Superbloque:** Contén información sobre a estrutura, organización e contido do volume.
- **Mapas de bits ou vectores de bits:** Representa o espazo existente nun sistema de arquivos. Inclúen un bit por recurso, por descritor de arquivo existente; se o recurso está libre, o valor do bit asociado ó mesmo é 1, se está ocupado é 0.

- **Descritores físicos de arquivos:** Cada un ocupa 4 KB. Cando se crea o sistema de arquivos o SO habilita un número de descritores físicos proporcional ó tamaño da partición. Este tamaño é moito maior que os homólogos en UNIX (os i-nodos) o que fai que xa neles se almacenen datos do arquivo, isto optimiza o acceso ós arquivos pequenos ó evitar accesos a disco para obter o descritor e despois os datos. Para arquivos grandes, o servidor de arquivos mantén unha estrutura xerárquica en forma de árbore, onde en cada bloque se almacenan datos e apuntadores ós bloques seguintes.
- **Bloques de datos (*Cluster*):** Estes bloques son asignados ós arquivos polo servidor de arquivos a través dos descritores de arquivos. O tamaño deste bloque é a mínima unidade de transferencia que manexa o SO (*cluster*).

### 1.6.3 Características xerais do sistema de arquivos NTFS

- ◊ 64 bits para direccionar bloques, polo que o tamaño máximo da unidade será de 16 HexaBytes.
- ◊ Non se recomenda utilizar NTFS nun volume de menos de 400 MB, debido á sobrecarga de espacio que implica NTFS. Esta sobrecarga de espacio refírese aos arquivos de sistema de NTFS que normalmente utilizan polo menos 4 MB de espacio de unidade nunha partición de 100 MB.
- ◊ Operacións de busca (B-Tree):  $\log(N) \times$  Tempo Base (sendo N o número de arquivos existentes).
- ◊ ACLs: Listas de Control de Acceso a arquivos e directorios.
- ◊ Journaling: Rexistro de transaccións.
- ◊ Nomes de arquivos: 256 caracteres
- ◊ Compresión individual: compact.exe
- ◊ Cifrado de arquivos (EFS): cipher.exe
  - Simétricos ou Clave Privada
  - Asimétricos ou Clave Pública
- ◊ Cotas de disco a partires da v5.0
- ◊ Descritores de Arquivos (Os i-nodos de Linux) de 4 KB.
- ◊ Pódese facer a conversión, dende o Windows e sen perda de información, dende unha partición en FAT/FAT32 a NTFS, emprégase o comando: **convert Unidade: /FS:NTFS**

O paso de NTFS a FAT/FAT32 non se pode dende o sistema operativo Windows. Si o permiten algunhas aplicacións do tipo Partition Magic (¡Olo! facer sempre antes unha copia de seguridade dos datos importantes).

## 1.7 Journaling

O *journaling* ben sendo un rexistro que permite ao sistema de arquivos restablecer os datos afectados por unha transacción no caso de que esta falle.

## 1.8 Compatibilidades FAT - NTFS

Temos o seguinte cadro de compatibilidades sistemas de arquivos / sistemas operativos:

Sistema Operativo	FAT16	FAT32	NTFS
Win 95	X		
Win 95-OSR2	X	X	
Win 98	X	X	
NT	X	X <sup>1</sup>	X
W2k	X	X	X
WXP	X	X	X
W2k3	X	X	X

**Nota 1:** Só soporta FAT32 unha vez instalado e mediante a aplicación do correspondente parche. Durante a instalación non recoñecerá ningunha partición formateada con FAT32.

**Nota importante:** Á hora de elixir un sistema de arquivos para unha partición, ter en conta que os Sistemas de Arquivos FAT son moi lentos cando se accede aleatoriamente a un arquivo grande (moito mais lentos que o NTFS e os Sistemas de Arquivos de Linux-UNIX). A razón é que, en FAT, non se sabe onde está un bloque (*cluster*) dun arquivo se non se sigue toda a cadea de bloques do mesmo. Por iso, si unha aplicación salta dun lado a outro do arquivo, pode ser necesario comezar cada vez dende o principio. Este comportamento débese a que o mecanismo está deseñado pensando en accesos secuenciais a arquivos pequenos.

## 1.9 exFAT

**exFAT (Extended File Allocation Table)** é un sistema de arquivos especialmente adaptado para memorias flash presentado con Windows Embedded CE 6.0. exFAT utilízase cando se decide non empregar o sistema de arquivos NTFS debido á sobrecarga das estruturas de datos.

Características básicas:

- Límite teórico para o tamaño dun arquivo de  $2^{64}$  bytes (16 Exbibytes) mellorando o límite de FAT32 de 4GB.
- Tamaño de cluster de ata  $2^{255}$  bytes (o implementado por defecto é de 32MB).
- Melloras no rendemento da asignación de espazo libre gracias á introdución dun **free space bitmap**.
- Soporte para mais de 1000 ficheiros nun só directorio.
- Soporte para ACLs.
- NON implementa journaling.

[Enlace interesante onde vemos o tamaño do cluster fronte o tamaño da partición para FAT - exFAT - NTFS.](#)

## 1.10 ReFS

- **ReFS (Resilient File System)** é un novo sistema de arquivos que aparece có SO Windows Server 2012 R2.
- Este sistema de arquivos ven a ofrecer practicamente un tamaño ilimitado nos arquivos e nos directorios (superior a petabytes), de aí o nome de "Resistente - Elástico".
- Tamén incrementa a resistencia a a aparición de erros e elimina a necesidade de empregar ferramentas de busca de erros (como **chkdsk.exe**, que xa non funcionará sobre este sistema de arquivos).
- Por outro lado, **ReFS non soporta moitas das características de NTFS** como: compresión de arquivos, EFS ou cuotas.
- Por último, comentar que ReFS non poderá ser lido por sistemas operativos "antigos" como Windows Server 2012 e Windows 8.
- [Moi bo artigo sobre ReFS.](#)

## 1.11 Sistemas de arquivos Linux - Unix

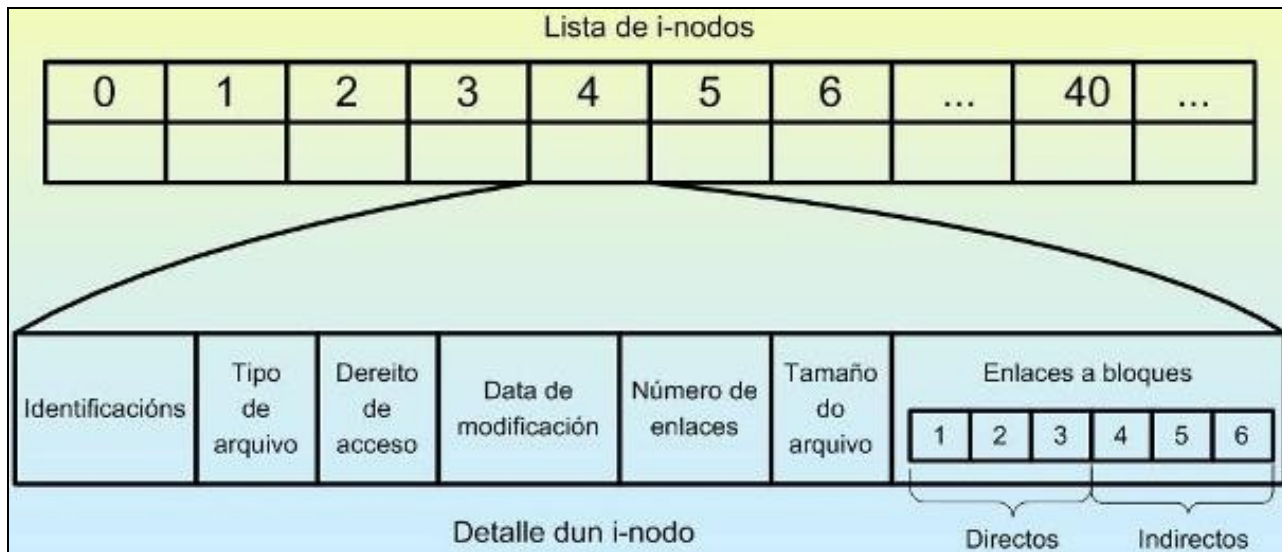
### 1.11.1 Estrutura dos Sistemas de arquivos Linux - *i-nodos*

BOOT	Superbloque	Mapas de bits	i-Nodos	Datos e directorios
------	-------------	---------------	---------	---------------------

- **Boot:** sector de arranque.
- **Superbloque:** almacena información de control do sistema de ficheiros.
- **Mapas de bits ou vectores de bits:** representa o espazo existente nun sistema de arquivos. Inclúen un bit por recurso, por descriptor de arquivo existente; se o recurso está libre, o valor do bit asociado ó mesmo é 1, se está ocupado é 0.
- **i-Nodos:** contén a lista de i-Nodos. Un i-nodo é unha estrutura que almacena información relativa a un ficheiro.
- **Datos e directorios:** almacena os arquivos e directorios do sistema de ficheiros.

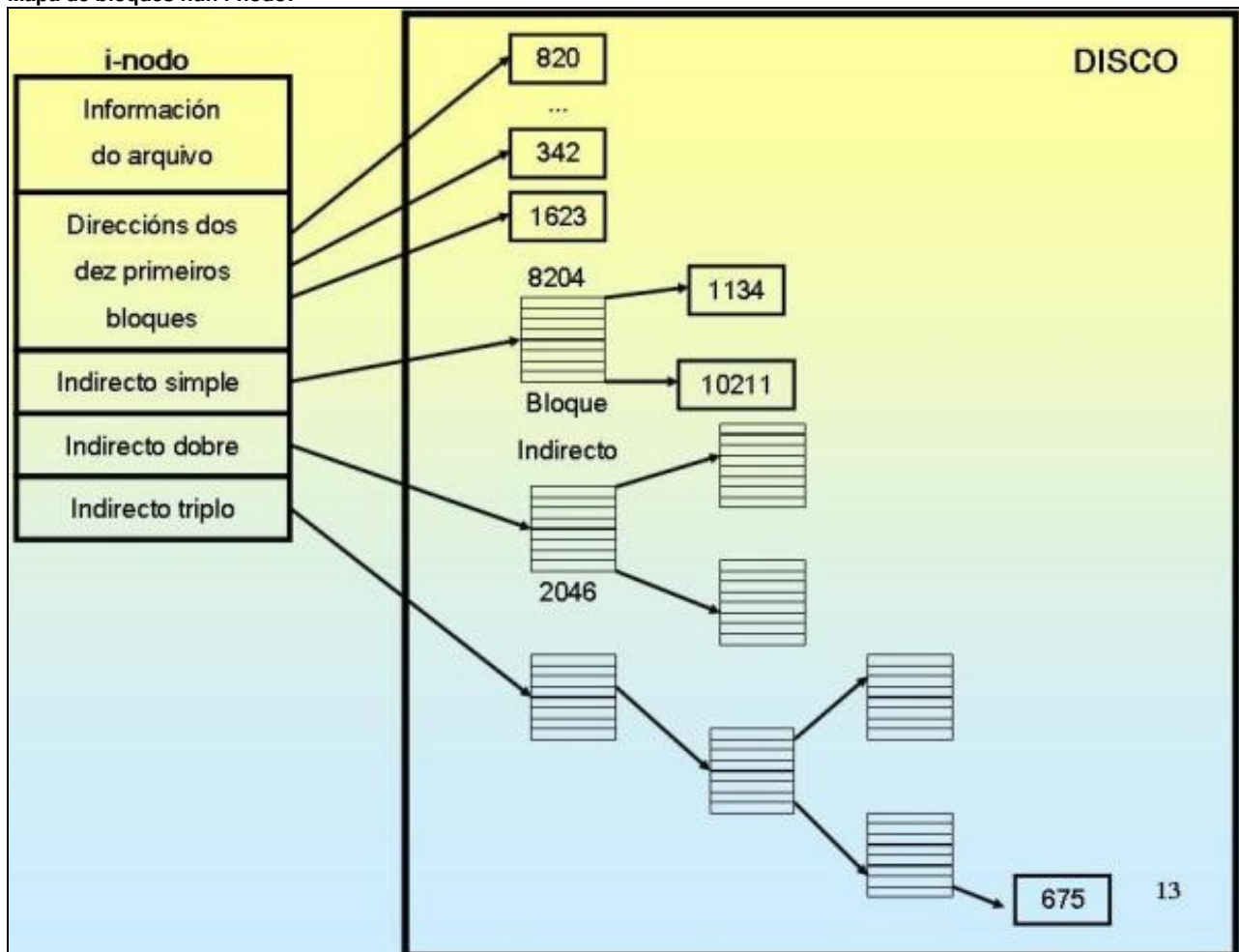
Características:

- ◇ Características similares a NTFS.
- ◇ i-nodos: Son os equivalentes dos "descriptores físicos de arquivos" de NTFS; son pequenos, 128 bytes por KB da área de Datos e Directorios. Non gardan nada do contido do arquivo, gardan só información sobre el.
- ◇ Cada i-nodo almacena información do arquivo: tipo de ficheiro, permisos, dono, datas de creación e modificación, número de enlaces, tamaño e punteiros a bloques de datos.



◊ Cada i-nodo ten varios punteiros a bloques de datos que especifican a ubicación física dos datos no disco. Hai 10 punteiros a bloques de disco, 1 indirecto simple, 1 indirecto dobre e 1 indirecto triple.

**Mapa de bloques nun i-nodo:**



◊ Estes bloques son dun tamaño que se especificou no momento da creación do sistema de arquivos, con un tamaño vai dende os 512 bytes ata os 4 kB, o que asegura un bo aproveitamento do espazo libre con arquivos pequenos.

Límites teóricos para ext2

**Tamaño do bloque:** 1 KB 2 KB 4 KB 8 KB



**Tamaño do bloque:** 1 KB 2 KB 4 KB 8 KB  
**Tamaño máximo arquivo:** 16 GB 256 GB 2 TB 2 TB  
**Tamaño máximo do sistema de ficheiros:** 4 TB 8 TB 16 TB 32 TB

◊ **Directorios:** os directorios impleméntanse como un tipo especial de ficheiros. Un directorio é un ficheiro que contén unha lista de entradas. Cada entrada contén un número de inodo e un nome de ficheiro.

i-nodo	nome arquivo
100	.
54	..
143	subdirectorío
150	arquivo.txt

Así as características particulares dos sistemas de arquivos linux máis coñecidos podémolos ver nos seguintes puntos:

### 1.11.2 ext2

**ext2** (*second extended filesystem*) é un sistema de arquivos para o kernel Linux. Foi deseñado orixinalmente por Rémy Card.

ext2 foi o sistema de arquivos por defecto das distribucións de Linux Red Hat, Fedora Core e Debian ata ser substituído recentemente polo seu sucesor ext3.

**Este sistema de arquivos ten un tipo de táboa FAT de tamaño fixo, onde se almacenan os i-nodos.**

- Tamaño mínimo do bloque: 1 KB
- Tamaño máximo do bloque: 4 KB (8 KB nas Alpha)
- Lonxitude máxima do nome: 255 caracteres
- Reserva de espacío para i-nodos: 1 por cada 4 KB

### 1.11.3 ext3

**ext3** (*third extended filesystem*) é un sistema de arquivos con *journaling*. É, sen dúbida, o sistema de arquivos máis empregado nas distribucións Linux, aínda que, na actualidade está sendo substituído polo seu sucesor, **ext4**.

- ◊ A principal diferenza con ext2 é que incorpora *journaling*.
- ◊ Outra diferenza importante é que ext3 emprega unha árbore binaria balanceada (árbore AVL).

### 1.11.4 ext4

**ext4** (*fourth extended filesystem*) é un sistema de arquivos anunciado en outubro do 2006 como unha mellora compatible de ext3. O 25 de decembro de 2008 publicouse o kernel Linux 2.6.28, que elimina xa a etiqueta de "experimental" de código de ext4. As principais melloras son:

- Soporte engadido de *extent*.
- Menor uso da CPU.
- Melloras na velocidade de lectura e escritura.
- ext4 é capaz de traballar con volumes de ata 1EB e ficheiros de ata 16TB.

## 1.12 Extents

Os *extents* foron introducidos para substituír ao tradicional esquema de bloques empregado polos sistemas de arquivos ext2/3. Un *extent* é un conxunto de bloques físicos contiguos, mellorando o rendemento ao traballar con ficheiros de grande tamaño e reducindo así a fragmentación. Un *extent* simple en ext4 é capaz de mapear ata 128MB de espazo contiguo cun tamaño de bloque igual a 4KB.



### 1.12.1 ReiserFS

ReiserFS ofrece funcionalidades que poucas veces se viron noutros sistemas de arquivos:

- *Journaling*.
- Reparticionamiento có sistema de ficheiros montado e desmontado. Podemos aumentar o tamaño do sistema de ficheiros mentres o temos montado e desmontado (*online* y *offline*). Para diminuílo, unicamente se permite estando *offline* (desmontado).
- *Tail packing*, un esquema para reducir a fragmentación interna.
- Tamaño máximo do sistema de ficheiros é de  $2^{32} \times 4 \text{ KB} = 16 \text{ TB}$
- Tamaño máximo dun ficheiro é de  $2^{60}$  - Uns cantos bytes = 1 EB (está limitado a 8 TB en arquitecturas de 32 bits).
- Pensando en posibles perdas de datos recoméndase empregar *ext3* fronte a *ReiserFS*.

### 1.12.2 UFS

- UFS é o acrónimo para denominar ó sistema de arquivos **UNIX file system (UFS)** que é empregado por varios sistemas operativos UNIX e POSIX. En realidade é un derivado do **Berkeley Fast File System (FFS)**.
- Case todos os derivativos de BSD incluíndo FreeBSD, NetBSD, OpenBSD, NeXTStep, e Solaris empregan unha variante de UFS. En MacOS X esta dispoñible como una alternativa ó HFS. En Linux, existe soporte parcial ó sistema de arquivos UFS, de só lectura.
- FreeBSD 5.0 introduciu **UFS2**, que engade soporte para volumes sobre un 1TB (TeraByte).
- Dende Solaris 7, Sun Microsystems introduciu **UFS Logging** en Solaris, para dar soporte de Journaling ó sistema de arquivos UFS. O UFS de Solaris tamén ten extensións para arquivos grandes, discos grandes e outras características.

### 1.12.3 XFS

XFS é un sistema de arquivos de 64 bits con *journaling* de alto rendemento creado por **SGI (Silicon Graphics Inc.)** para a súa implementación de UNIX chamada IRIX. No, SGI liberou XFS baixo unha licenza de código aberto.

- ◇ XFS é o mais antigo dos sistema de arquivos con *journaling* dispoñible para a plataforma UNIX.
- ◇ XFS incorporouse a Linux a partires da versión 2.4.25.
- ◇ Os programas de instalación das distribucións de SuSE, Gentoo, Mandriva, Slackware, Fedora Core, Ubuntu e Debian ofrecen XFS como un sistema de arquivos mais.
- ◇ XFS soporta un sistema de arquivos de ata 9 EB, aínda que isto pode variar dependendo dos límites impostos polo sistema operativo (En sistemas Linux de 32 bits, o límite é de 16 TB).

### 1.12.4 JFS

**Journaling File System (JFS)** É un sistema de arquivos de 64-bit con respaldo de transaccións creado por IBM. Está dispoñible baixo la licenza GNU GPL. Existen versións para AIX, eComStation, OS/2, sistemas operativos Linux e HP-UX

- ◇ A primeira versión para Linux foi distribuída no verán do 2000.
- ◇ Foi deseñado coa idea de conseguir "servidores de alto rendemento e servidores de arquivos de altas prestacións, asociados a e-business".
- ◇ JFS utiliza un método interesante para organizar os bloques baleiros, estruturándoos nunha árbore e emprega unha técnica especial para agrupar bloques lóxicos baleiros.
- ◇ Ao ser un sistema de ficheiros de 64 bits, JFS soporta ficheiros grandes e particins **LFS (Large File Support)**.
- ◇ Acada mellor utilización do disco mediante adxudicación dinámica de lnodes, conseguindo aumentar, con respecto aos sistemas de arquivos ext, a cantidade máxima de ficheiros que se poden gardar nun volume.

### 1.12.5 ZFS

ZFS é un sistema de arquivos moderno deseñado para ter características moito máis avanzadas das que ofrece un sistema de arquivos UNIX tradicional.

- [ZFS na web de Freenas](#)
- [ZFS Na Wikipedia](#)

### 1.12.6 btrfs

...

## 1.12.7 LVM

...

## 1.12.8 CEPH

...

## 1.13 Sistema de arquivos de CD-ROM ISO 9660

Trátase do estándar mais común para os sistemas de CD-ROM e data do 1.998.

### 1.14 Nota

A norma ISO 9660 é descendente directa dun esforzo de estandarización anterior chamado **HSG (High Sierra Group)**, proposto por un conxunto de fabricantes reunidos en 1985 no hotel High Sierra, do Lago Tahoe, Nevada. Aínda que a ISO aceptou unha grande maioría das propostas do HSG, existen algunhas diferencias menores.

No definido ni por High Sierra ni por el estándar ISO 9660
Descriptor del volumen primario
Descriptor del volumen secundario - opcional
Descriptores de partición - opcional
Descriptores de arranque (Boot) - opcional
Descriptor final de volumen
Tabla de rutas (path table) 4 posibles copias en High Sierra 2 posibles copias en ISO 9660
Directorio raíz Entrada para el directorio raíz Entrada para el directorio padre Entrada para el primer archivo o directorio ... Entrada para el último archivo o directorio
Subdirectorio Entrada para el directorio raíz Entrada para el directorio padre Entrada para el primer archivo o directorio ... Entrada para el último archivo o directorio
Contenido de archivo

Os CD-ROMs están compostos dunha única espiral continua que contén os bits nunha secuencia lineal.

Os bits ao longo da espiral divídense en bloques lóxicos de 2.352 bytes, sendo a porción aproveitábel para cada bloque lóxico de 2.048 bytes.

#### Estrutura do sistema de arquivos:

- ◊ Cada CD-ROM comeza con **16 bloques non definidos** polo estándar que calquera fabricante podería empregar para, por exemplo, facelo "arrancable" e así poder ter no seu interior un SO ou outro tipo de aplicacións.
- ◊ A continuación séguelle un bloque que contén o "**descriptor de volume primario**", que contén certa información xeral sobre o

CD-ROM:

- O identificador do sistema: 32bytes.
  - O identificador do volume: 32bytes.
  - O identificador do publicador: 128bytes.
  - O identificador do publicador de datos: 128 bytes.
  - Os nomes de tres arquivos que poden conter: o resumo do contido do CD, o aviso de copyright e a información bibliográfica.
  - Varios números clave presentes: Tamaño do bloque lóxico, número de bloques no CD-ROM, as datas de creación e expiración do CD\_ROM.
  - Unha entrada de directorio para o directorio raíz, que indica onde se pode atopar no CD-ROM.
- ◊ O **descriptor de volume secundario**, se existe, contén información similar á do primario.
- ◊ O **directorio raíz** con todos os demais directorios relacionados:
- Consiste dun número variable de entradas, o último das cales contén un bit que a marca como a última.
  - As entradas de directorio son tamén de lonxitude variable.
  - Cada entrada de directorio pode ter de 10 a 12 campos.
  - Algúns campos están en ASCII e outros son campos numéricos en binario. Os campos binarios están codificados dúas veces, unha vez en formato *little endian* (Pentium) e *big endian* (Motorola).

## 1.15 **Little endian vs Big endian**

O termo inglés **Endianness** designa o formato no que se almacenan os datos de mais dun byte nun soporte informático. O problema é similar aos idiomas nos que se escriben de dereita a esquerda, como o árabe, fronte aos que se escriben de esquerda a dereita, pero trasladado ao almacenamento de bytes. Así o valor hexadecimal 0x4A3B2C1D codificaríase en memoria na secuencia {4A, 3B, 2C, 1D} no sistema *big endian*, e este mesmo valor codificaríase como {1D, 2C, 3B, 4A} no *little endian*, deste modo faise mais intuitivo o acceso a datos porque vai de menos relevante a mais relevante.

## 1.16 Particións Cifradas

PGP

## 1.17 Benchmarks de Sistemas de Arquivos

### 1.17.1 dbench

◊ [dbench man page](#)