

Dischi e Filesystem

Stefano Sanfilippo

POuL – Politecnico Open unix Labs, Corsi Linux Avanzati

27 marzo 2014

FHS è lo standard per la struttura delle directory nei sistemi Linux.

<http://www.pathname.com/fhs/>

- Estensione della gerarchia BSD
- Curato dalla Linux Foundation
- Versione attuale è la 2.3 (29 gennaio 2004)
- Successore di FSSTND e `hier`

In pratica: definisce *dove* mettere *cosa*.

- 1 tutti i files sono collocati sotto /, anche se memorizzati su dispositivi fisici o virtuali differenti.
- 2 non tutte le directory specificate devono essere presenti sul sistema.
- 3 tutto è un file e se non è un file allora è un processo. Anche i dispositivi di memoria (cfr. C:\, D:\ di Windows) sono files speciali.

Le principali distribuzioni sono FHS-compliant, ma presentano alcune piccole deviazioni.

Distribuzione	Riferimento
Debian	debian.org
SuSe	novell.com
RHEL/Fedora	redhat.com
Ubuntu	ubuntu.com

Ma non tutte!

Ad esempio, GoboLinux

Percorso	Contenuto
/	cartella <i>root</i> , la <i>radice</i> del filesystem.
/boot	files di avvio: bootloader, kernel e initrd.
/etc	files di configurazione del sistema
/home	cartelle personali degli utenti
/media	punto di mount per i filesystem esterni (cd...)
/mnt	punto di mount per i filesystem interni temporanei
/opt	pacchetti software <i>opzionali</i> (concetto elastico)
/root	home dell'utente root
/usr	eseguibili e librerie per le applicazioni utente
/var	dati (mutabili), log e cache di sistema

Definizione

Struttura

Generale

Eseguibili

Speciali

Deviazioni dallo
standard

FHS specifica una gerarchia a *tre livelli*.

Eseguibili e librerie sono posizionati in tre punti del FS:

<code>/</code>	essenziali al funzionamento del sistema
<code>/usr</code>	applicazioni utente
<code>/usr/local</code>	applicazioni locali (<code>make install...</code>)

Definizione

Struttura

Generale

Eseguibili

Speciali

Deviazioni dallo
standard

In ciascun percorso, i binari sono suddivisi in 3 cartelle:

.../bin	binari
.../lib	librerie
.../sbin	binari di sistema

Perché separare bin e sbin?

Di solito, i binari in .../sbin richiedono privilegi di amministratore.

Definizione

Struttura

Generale

Eseguibili

Speciali

Deviazioni dallo
standard

Oltre a `/sbin`, `/bin` e `/lib`, contiene

<code>/usr/include</code>	Include files (cfr. *.h del C/C++)
---------------------------	------------------------------------

<code>/usr/share</code>	Dati delle applicazioni (immagini, suoni...)
-------------------------	--

<code>/usr/src</code>	Codice sorgente (es. kernel)
-----------------------	------------------------------

Definizione

Struttura

Generale

Eseguibili

Speciali

Deviazioni dallo
standard

Il contenuto di `/usr/share` non è coperto da FHS.

Tuttavia, è consuetudine avere:

- `.../icons`
- `.../wallpapers`
- `.../sounds`
- `.../pixmaps` (icone e immagini dei programmi)
- `.../doc`

E altre cartelle:

- specifiche per i singoli programmi, es. `.../amarok/`

Definizione

Struttura

Generale

Eseguibili

Speciali

Deviazioni dallo
standard

Sono rappresentazioni di alcuni aspetti del sistema, nello spirito della *regola III* (tutto è un file).

`/dev` riferimenti ai dispositivi

`/proc` informazioni sui processi e sul kernel

Sono il punto di mount di FS (speciali) opportuni.

Sono memorizzati in due posizioni diverse:

<code>/tmp</code>	Cancellata, <i>di solito</i> , ad ogni riavvio
-------------------	--

<code>/var/tmp</code>	Persistente
-----------------------	-------------

Regola

Se vanno persi, il sistema non ne sentirà la mancanza.

Dati condivisi delle applicazioni durante il loro funzionamento.

Definizione

Struttura

Generale

Eseguibili

Speciali

Deviazioni dallo
standard

<code>/var/cache</code>	Cache di sistema
<code>/var/lib</code>	Informazioni di stato
<code>/var/lock</code>	Lock files
<code>/var/log</code>	Log di sistema (demoni, firewall, accessi...)
<code>/var/run</code>	Informazioni sul sistema (utenti connessi...)
<code>/var/spool</code>	Coda per le operazioni lunghe (stampa, mail...)

Problema

`/var` spesso è montato come una partizione separata, quindi potrebbe non essere accessibile nelle prime fasi di avvio.

Le moderne distribuzioni Linux creano altre due directory, non standard:

`/run` come `/var/run`, ma montata come filesystem virtuale, **subito accessibile**.

`/sys` contiene i parametri del kernel e le opzioni per configurarne il funzionamento.

NOTA sui vecchi unix, era un collegamento al codice sorgente del kernel.

Alcune distribuzioni (cfr. Archlinux e Fedora) appiattiscono la gerarchia a tre livelli:

⇒ `/lib`, `/sbin` e `/bin` sono collegamenti a `/usr/...`

Distinguiamo tre tipi di dispositivo:

A blocchi operazioni di I/O per *blocchi* di dimensione predeterminata (*es.* hard disk, che si leggono a settori – p.e. di 4KB)

A caratteri operazioni di I/O per singoli byte (o, comunque, senza blocchi di dimensione fissata) (*es.* standard output)

Speciali sono «astrazioni», non necessariamente legate ai dispositivi di I/O.

Per ragioni di sicurezza, i dispositivi possono essere creati solo da un utente privilegiato e non sono direttamente accessibili ai normali utenti.

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsdf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Le partizioni sono le sezioni in cui è suddiviso il disco:

- Ogni disco deve avere almeno una partizione, sulla quale si può creare il FS.

I metadati (dimensione, posizione, FS contenuto e attributi) delle partizioni sono contenuti in una tabella su ciascun disco.

- Il formato e l'organizzazione di questa tabella sono standard.
 - In questo modo è garantita l'interoperabilità.

MBR è uno schema legacy, comparso negli anni '80:

- 1 la struttura dati precede la prima partizione.
- 2 ogni partizione è descritta da un numero
- 3 la tabella delle partizioni descrive al massimo 4 partizioni fisiche
- 4 la dimensione max della singola partizione è di 2TiB
- 5 sono state aggiunte le partizioni logiche (al costo di una fisica), memorizzate in una lista concatenata

Attenzione!

se si danneggia il record di una partizione logica, si perdono i dati delle seguenti

GPT è il successore di MBR.

- 1 viene usata su UEFI e su alcuni sistemi BIOS moderni
- 2 innalza i limiti di MBR
 - max 8ZiB per partizione
 - default a 128 partizioni. Nessun *hard-limit*.
- 3 controllo di integrità della tabella e backup
- 4 partizioni identificate da un GUID (id univoco), possibilità di conflitto quasi nulle

I due schemi NON possono coesistere!

In particolare, parted (e altri) scrivono una *protective MBR*, una MBR finta che impedisce agli strumenti che non riconoscono GPT di far danni.

MBR andrebbe evitato, a meno che non sia necessario per la «convivenza» con altri OS più vecchi.

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsuf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Nel caso si proceda con un'installazione manuale di Linux su di un disco con GPT, occorre ricordare che:

- **GRUB2 su BIOS con GPT** ha bisogno di una partizione vuota da 1007kiB all'inizio del disco per memorizzare la sua `core.img`
- **GRUB2 su UEFI (con GPT)** ha bisogno della ESP (*EFI System Partition*), una partizione contenente un filesystem VFAT, da circa 512MiB e con il flag `boot` attivato.
 - La ESP è montata in `/boot/efi`

Nel caso di *dispositivi a blocchi*, abbiamo definito il:

Blocco

unità minima di informazione che

- può essere **letta o scritta**
- in **una sola operazione**
- sul dispositivo.

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsuf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

In realtà, dobbiamo distinguere due dimensioni di blocco:

Blocco fisico

dimensione di blocco effettivamente gestita dall'hardware

⇒ negli HDD era di 512B, in quelli moderni arriva a 4kiB

⇒ negli array RAID è di 64kiB

⇒ negli SSD può arrivare a 128kiB

Blocco logico

dimensione di blocco «apparente» per il software

⇒ quasi sempre 512B, per compatibilità con il software legacy

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsfd

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Allineare le partizioni significa posizionarle in modo da sovrapporre la struttura logica (a blocchi di 512B) con quella fisica sottostante (a blocchi di 4kiB~128kiB).

Ovvero:

- 1 la partizione inizia ad un multiplo della dimensione di blocco fisico
- 2 la dimensione della partizione è multipla della dimensione di blocco fisico

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e
umount

lsdf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Quando l'OS scrive 4kiB, 64kiB o 128kiB,
scrive un solo blocco (intero).

Altrimenti, una scrittura da 4kiB potrebbe capitare «a cavallo» tra due blocchi fisici, implicando due letture/scritture (una per blocco fisico coinvolto) al posto di una.

I dispositivi compaiono in `/dev` e hanno *nomi convenzionali*.

Le notazioni riguardano *solo* Linux!
(es. BSD usa altre convenzioni).

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsdf

df

Loop

UDev

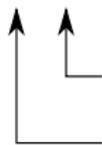
fstab

S.M.A.R.T.

Hard e soft
link

I dischi PATA/SATA (la quasi totalità di quelli usati attualmente) usano una prefisso *sd* e una convenzione a due livelli:

`/dev/sda1`



Numero: m-sima partizione sull'n-simo disco

Lettera: n-simo dispositivo (~disco)

`/dev/sda` ← Senza numero finale, n-simo dispositivo

Su MBR, la prima partizione logica è sempre `sdX5`

Framebuffer	/dev/fb
CD-ROM	/dev/cdromX
Lettori Floppy Disk	/dev/fdX
Terminali virtuali	/dev/ptX
Terminali	/dev/{ttyX,ttyS,ttyUSBX}
Loop	/dev/loopX

Esistono ancora altri nomi, meno usati (legacy o rari), non li tratteremo.

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsdf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsdf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Dispositivo	Input	Output
/dev/null	Scarta l'input	N/A
/dev/zero		Flusso continuo di <code>NULL</code>
/dev/full	«Disco pieno»	
/dev/random	N/A	Flusso <i>truly random</i>
/dev/urandom		Flusso pseudorandom

Anche `/dev/stdin`, `/dev/stdout` e `/dev/stderr` sono file!

La cartella `/dev/disk` contiene link simbolici ai dispositivi e alle partizioni riconosciute dal sistema, secondo varie classificazioni:

Subpath	Nome costruito a partire da
<code>.../by-id</code>	ID del dispositivo (e numero della partizione)
<code>.../by-label</code>	Label del filesystem
<code>.../by-path</code>	Path del dispositivo (bus, porta, ...)
<code>.../by-uuid</code>	UUID del filesystem
<code>.../by-partlabel</code>	Label della partizione GPT
<code>.../by-partuuid</code>	UUID della partizione GPT

Ogni schema ha un caso d'uso:

by-id permette di individuare subito il disco a cui ci si riferisce, nel caso sia necessario sostituirlo.

by-path come sopra, ma individua subito la porta (e non direttamente il disco) a cui è connesso. Utile:

- in presenza di più dischi con ID molto simile (p.e. stesso produttore e modello)
- in presenza di molti dischi

by-label utile se è possibile dare un nome significativo al dispositivo

by-uuid usa lo UUID delle partizioni, il cui formato dipende dal filesystem. Utile in presenza di un solo o di pochi dischi (dato che non permette di ricavare subito la posizione della partizione).

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e
umount

lsuf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Rifersi alle partizioni per UUID, ID o *path*, aiuta ad evitare errori.

Se un dispositivo o una partizione non è classificabile secondo uno degli schemi sopra (p.e. non ha una label), allora non comparirà alcun link per quel dispositivo/partizione nella cartella relativa.

- Ogni dispositivo ha un file speciale corrispondente nella cartella /dev.
- I dispositivi sono direttamente leggibili e scrivibili (ad esempio, standard input)
 - a volte sono disponibili interfacce di livello superiore, specialmente se la lettura/scrittura avviene con protocollo.
- Per accedere ai dati contenuti in un dispositivo a blocchi (i files), bisogna «montarlo».

Il comando per montare i filesystem è `mount`:

```
# mount $nome_device $punto_di_mount
```

Eeguire `mount` senza argomenti mostrerà i FS attualmente montati, con le relative opzioni.

È il comando simmetrico: serve a «smontare» un dispositivo, in modo da effettuare una rimozione sicura:

```
# umount $punto_di_mount
```

```
# umount $nome_device
```

Non è un errore, il comando è `umount`, non `unmount`!

Non è possibile smontare un filesystem occupato (es. c'è un trasferimento di file in corso).

In tal caso si può usare `lsuf` per mostrare il PID dei programmi che stanno usando il dispositivo.

```
# lsuf /nome/del/file
```

Processi che hanno aperto *il file* specificato

```
# lsuf -c nomeprocesso
```

File usati dal processo specificato

```
# lsuf +D /nome/della/directory
```

Processi che hanno aperto un file nella directory specificata (o in una sottodirectory)

Potrebbe essere necessario eseguire `lsuf` da utente privilegiato.

Indica quanto spazio libero rimane su ogni partizione *montata*:

```
$ df
```

⇒ Viene spesso usato con l'opzione *-h*, che riporta le misure in unità «*human-readable*»:

```
$ df -h
```

Lo spazio libero non corrisponde alla differenza tra la capacità massima e la somma della dimensione dei files!
(una parte potrebbe essere riservata per usi particolari). Gestione del filesystem

Il loop è uno *pseudo-device* che rende un file ordinario accessibile come se fosse un dispositivo a blocchi.

- È utile per montare immagini di dischi o chiavette senza doverle mettere su di un supporto fisico.
- Su Linux, `mount` si occupa di tutto, deve solo essere informato con l'opzione `loop`:

```
# mount -o loop nome_file punto_di_mount
```

- i dispositivi (*device*) vengono creati con `mknod`.
- Sui sistemi Linux moderni i dispositivi a blocchi sono creati e gestiti automaticamente da un demone, **UDev**.
 - Udev ascolta gli eventi di sistema (uevent, appunto) e reagisce opportunamente.
 - È un componente articolato, ma non ce ne occuperemo. *Sappiate che esiste e funziona.*

Osservate

Quando attaccate una chiavetta, viene creato automaticamente `/dev/sdb`

- file di configurazione, situato in `/etc`
- elenca tutti i dispositivi disponibili sul sistema e indica come vanno inizializzati o integrati nel FS del sistema.
- è usato ancora per la configurazione dell'hard disk principale e i FS di startup, ma **è stato rimpiazzato** da UDev per i dispositivi esterni.

Linea di `/etc/fstab`

```
# device name mpoint fs-type options dump-freq pass-num  
/dev/sda2 / ext4 errors=remount-ro 0 1
```

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e
umount

lsdf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

I dischi rigidi (quasi tutti) hanno un sistema di diagnostica interna, detto S.M.A.R.T.

S.M.A.R.T. fornisce diversi indicatori di affidabilità per anticipare i guasti e salvaguardare i dati.

Alcuni dati forniti sono:

- Temperatura
- Numero di settori danneggiati
- Ore di funzionamento

Gli indicatori S.M.A.R.T. sono essenziali per prevedere in anticipo il guasto di un disco.

I dati S.M.A.R.T. possono essere visualizzati con `smartctl`, `gsmartcontrol` o `palimpsest`.

Per ottenere tutte le informazioni da linea di comando:

```
# smartctl -x /dev/sda
```

Stefano
Sanfilippo

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsdf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Negli ambienti desktop (GNOME, KDE, Unity, . . .), un demone rileva eventuali allarmi e li segnala all'utente.

Alcuni concetti importanti:

- 1 Nel FS, i files sono rappresentati da *inode*.
 - Un inode contiene la posizione, la dimensione, i permessi e gli attributi di ciascun file
- 2 Un *path* nel FS è rappresentato da un *hard link* ad un *inode*.
 - Dà un nome all'*inode*
- 3 Quando cancellate un file, viene rimosso un link all'*inode*.
- 4 I link si creano con il comando `ln`

Un *hard link* crea un **altro** link
allo **stesso** *inode*.

L'*inode* viene marcato come cancellabile quando non ci sono più hard link che lo puntano.

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsuf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

ls

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Un *soft link* è un **collegamento** (riferimento)
ad un altro file, per **path** (nome).

- 1 Spostare o rinominare un file non influenzerà un hard link, perché è un collegamento all'inode
 - i cambiamenti all'inode si riflettono su tutti gli hard link
- 2 Gli hard link sono validi solo nello stesso FS
- 3 I soft link possono espandersi ad altri FS
 - sono semplici riferimenti al nome del file

Ora sappiamo quasi tutto su FHS.

Ma in concreto?

Come organizzo le partizioni e i files?

Dispositivi

Partizioni

MBR

GPT

Note

GRUB

Allineamento

Notazione

Standard

PATA/SATA

Legacy

Speciali

/dev/disk

mount e

umount

lsdf

df

Loop

UDev

fstab

S.M.A.R.T.

Hard e soft
link

Uno schema su disco prevede più partizioni per gestire meglio i dati.

Non ci sono regole assolute – dipende dall'uso che la distribuzione fa di ciascuna partizione.

Uno schema usuale prevede:

Percorso	Dimensioni consigliate	Variabile principale
/	10~20GB	Applicazioni installate
/boot	50MB~100MB	Numero di kernel
/home	?	Utente
/var	5~10G	Cache varie

- Partizionate l'uso del disco!
 - es. separare /var/mail limita l'effetto dello spam sul sistema, se la macchina è un server di posta
- Se possibile, lasciare un po' di spazio libero per futuri assestamenti.

Checksumming calcolo di codici di controllo per dati e metadati, dà una certa resistenza alle «silent corruptions»

Deduplicazione mappatura di copie identiche della stessa risorsa sullo stesso blocco (cfr. COW)

Copy on Write (COW) creazione «al volo» di una copia delle risorse condivise, in caso di modifica unilaterale (duplico solo se necessario).

Frammentazione Suddivisione di un file in più blocchi o gruppi di blocchi non contigui. Rallenta I/O su dischi a rotazione (obbliga ad un seek), sfrutta meglio lo spazio (tappa i buchi).

Deframmentazione Rimozione della f., ottenuta muovendo i blocchi per rendere contiguo il più grande numero possibile (100% limite teorico).

Extent area di memorizzazione contigua (blocchi contigui) riservata per un file. Riduce la frammentazione.

Journaling scrittura su disco per transazioni. Garantisce correttezza e consistenza del FS in caso di spegnimenti improvvisi.

- Le operazioni (metadati) vengono inserite nel *log* (un'area riservata del FS), poi applicate.
- Ogni successo/fallimento viene registrato.
- In caso di spegnimento, Linux analizza il log e prende opportune misure.

Mirroring copia esatta del contenuto del FS: in caso di rottura del disco, potrà essere sostituito e ripristinato.

Quota quantità massima di spazio sul FS usabile da un utente, gruppo, processo (o una combinazione).

Wear leveling (TRIM) distribuzione dell'uso delle celle di memoria su un SSD, per bilanciare l'usura della flash. Viene effettuata dall'hardware, ma ha bisogno della collaborazione del SO (che deve sapere e indicare quali blocchi non sono più in uso).

In uso attuale, sono diffusi 5 filesystem:

Nome	Stato	Journal	COW	C'sum	Dedupe	Defrag	Mirror	Quota
ext4	stabile	✓	~	✗	✗	✓	✗	✓
btrfs	«testing»	✓	✓	✓	✓	~	✓	~
zfs	stabile	✓	✓	✓	✓	✓	✓	✓
vfat	legacy	✗	✗	✗	✗	✗	✗	✗
ntfs	«stabile»	✓	✗	?	✗	~	~	✓

VFAT DEVE essere **EVITATO** su qualsiasi disco fisso, ma è utile per chiavette (compatibile con Windows): no Journal!!

- 1 BTRFS e ZFS offrono circa le stesse caratteristiche, ma ZFS è nato nel mondo OpenSolaris, BTRFS in quello Linux.
- 2 BTRFS e ZFS supportano la compressione trasparente, ma è consigliata solo in sistemi multi-core/CPU, dove l'overhead è difficilmente notato (1 core == NO!)
- 3 ZFS non può essere distribuito assieme al Kernel di Linux per questione di licenza. Va installato a parte.

- 1 EXT4 è stabile. BTRFS (che lo rimpiazzerà) è dichiarato sperimentale, ma il formato su disco è stabile – salvo «bug che rendano la modifica assolutamente necessaria».
 - Se i dati memorizzati non sono essenziali (es. backup), è consigliato provarlo.
- 2 Un volume EXT4 può essere convertito in BTRFS: è possibile installare EXT4 e, in futuro, convertire.

- procfs** montato in `/proc`, fornisce informazioni sui processi attivi, sotto forma di pseudo-FS (es. `/proc/$PID`)
- sysfs** montato in `/sys`, permette di modificare i parametri del kernel e apprendere la configurazione del sistema.
- tmpfs** FS creato nella RAM, utile per `/run`, `/var/lock` e altre strutture «piccole» (RAM limitata) che hanno bisogno di velocità ma non devono resistere al reboot
- aufs** raggruppa più FS, facendoli apparire come un solo FS virtuale. Successore di `unionfs`.

squashfs FS compresso in sola lettura. Viene spesso usato nei Live CD, facendo risparmiare spazio.

ISO 9660 (e UDF) FS in sola lettura usato nei CD dati. UDF è il successore (usato nei CD-RW e nei DVD)

eCryptfs overlay che supporta la cifratura trasparente a livello di singoli file e path (es. si può abilitare solo per una cartella).

Questa opera è licenziata sotto i termini della Creative Commons, Attribuzione – Non commerciale – Condividi allo stesso modo 3.0 Italia License.

Per una copia della licenza, consultare

<http://creativecommons.org/licenses/by-nc-sa/3.0/it/>.

