

Amministrare GNU/Linux

Terza serata

v. 7

Cos'è il filesystem:

Un **file system**, in [informatica](#), indica informalmente, un meccanismo con il quale i [file](#) sono posizionati e organizzati o su un dispositivo di archiviazione o una [memoria di massa](#), come un [disco rigido](#) o un [CD-ROM](#) e, in casi eccezionali, anche sulla RAM.

Più formalmente, un file system è l'insieme dei tipi di dati astratti necessari per la memorizzazione (scrittura), l'organizzazione gerarchica, la manipolazione, la navigazione, l'accesso e la lettura dei [dati](#). Di fatto, alcuni file system (come [NFS](#)) non interagiscono direttamente con i dispositivi di archiviazione.

I file system possono essere rappresentati sia graficamente tramite [file browser](#) sia testualmente tramite [shell testuale](#). Nella rappresentazione grafica ([GUI](#)) è generalmente utilizzata la metafora delle [cartelle](#) che contengono documenti (i file) ed altre sottocartelle.

(da Wikipedia)

File system del disco

Un file system del disco è un file system progettato per memorizzare dei [file](#) su un'[unità a disco](#), che può essere collegata direttamente o indirettamente al [computer](#).

Esempi di file system del disco sono:

- [Amiga FileSystems](#) - OFS, FFS1 e 2, International, PFS, [SFS](#) usati su [Amiga](#)
- [BFS \(Beos File System\)](#) - file system nativo di [BeOS](#)
- DFS, [ADEFS](#) - file system della [Acorn](#)
- [EFS \(IRIX\)](#) - un vecchio file system a blocchi usato su [IRIX](#) per [unità di memoria di massa](#) rimovibili (DAT, CD-ROM...) precedente all'introduzione di [ISO 9660](#)
- Ext - [Extended file system](#), il primo file system appositamente progettato per [GNU/Linux](#)
- [Ext2](#) - Extended File System 2, diffuso su sistemi [GNU/Linux](#)
- [Ext3](#) - Extended File System 3, diffuso su sistemi [GNU/Linux](#) (ext2+journaling)
- [Ext4](#) - Extended File System 4, rilasciato come stabile dal kernel Linux 2.6.28 (già presente dalla versione 2.6.19 come ext4dev)
- [FAT](#) - Usato su [DOS](#), [Microsoft Windows](#) e su molti dispositivi dedicati, dispone di tabelle a 12 e 16 bit
- [FAT32](#) - versione con tabelle a 32 bit di FAT
- [ExFAT](#) - conosciuto anche come FAT64, creato da [Microsoft](#) e pensato appositamente per [memorie flash](#)
- [FFS](#) - Fast File System, usato in vecchi sistemi [BSD](#) e [Amiga](#)
- [HFS](#) - Hierarchal File System, usato su [Mac OS](#)
- [HFS+](#) - Hierarchal File System Plus, usato su Mac OS a partire dalla [versione 8.1](#) e su [Mac OS X](#)
- [HPFS](#) - High Performance File System, usato su [OS/2](#)
- [ISO 9660](#) - Usato su dischi [CD-ROM](#) e [DVD-ROM](#) (anche con estensioni [Rock Ridge](#) e [Joliet](#))
- [Journaled File System \(JFS\)](#) - disponibile su sistemi [GNU/Linux](#), [OS/2](#), e [AIX](#)
- [LFS](#) - Log-structured File System
- [Minix](#) - Usato su sistemi [Minix](#)
- [NTFS](#) - NT File System. Usato su sistemi basati su [Windows NT](#)
- [NWFS](#) - NetWare File System. Usato da [Novell NetWare](#)



This work is licensed under a [Creative Commons Attribution 4.0 International License](#).

- [ReiserFS](#) - File system journaling diffuso su sistemi [GNU/Linux](#)
- [Reiser4](#) - File system journaling successore del [ReiserFS](#) diffuso su sistemi [GNU/Linux](#).
- [UDF](#) - File system a pacchetti usato su supporti WORM/RW, CD-RW e DVD
- UFS - [Unix File System](#), usato su vecchi sistemi [BSD](#)
- [UFS2](#) - Unix File System, usato su nuovi sistemi [BSD](#)
- [UMSDOS](#) - File system FAT esteso con permessi e metadata, usato su GNU/Linux
- [XFS](#) - eXtended FileSystem: 64 bit in addressing con possibilità di partizione unica di 9 milioni di TB utilizzato su [IRIX](#)- Tale filesystem è indicato per unità non rimovibili (HDD...).
- [ZFS](#) - Creato dalla [Sun Microsystems](#)
- [Btrfs](#) - Creato dalla [Oracle Corporation](#)
- [WBFS](#)-(Wii Backup File System) Usato per i backup dei giochi della console [Wii](#)
- [Protogon](#) - File System introdotto con la build 7955 di [Windows 8](#), disponibile però nella sola versione Server, successivamente modificato intorno alla fine del 2011 in [ReFS Resilient File System](#)

L'architettura dei file

Uno dei concetti fondamentali dell'architettura di un sistema Unix è il cosiddetto “everything is a file”, cioè il fatto che l'accesso ai vari dispositivi di input/output del computer viene effettuato attraverso un'interfaccia astratta che tratta le periferiche allo stesso modo dei normali file di dati.

Esempio configurazione tastiera:

```
$ nano /etc/default/keyboard
```

Questo significa che si può accedere a qualunque periferica del computer, dalla seriale, alla parallela, alla console, e agli stessi dischi attraverso i cosiddetti file di dispositivo (i device file). Questi sono dei file speciali agendo sui quali i programmi possono leggere, scrivere e compiere operazioni direttamente sulle periferiche, usando le stesse funzioni che si usano per i normali file di dati.

Aprire tty1 e digitare da root:

```
# stop gdm3 # in Ubuntu Gnome oppure Debian
# stop lightdm # in Ubuntu Unity
# /etc/init.d/gdm3 stop # in Debian Gnome
# X -configure # verrà creato il file xorg.conf.new
# X -config xorg.conf.new # per testare la configurazione
# start gdm # per riavviare l'interfaccia graficamente
# /etc/init.d/gdm3 start # per riavviare l'interfaccia graficamente
```



Differenza tra file e directory

creare file >> touch nome_file

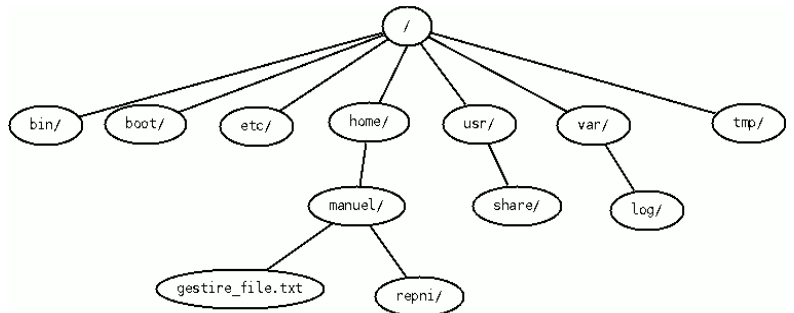
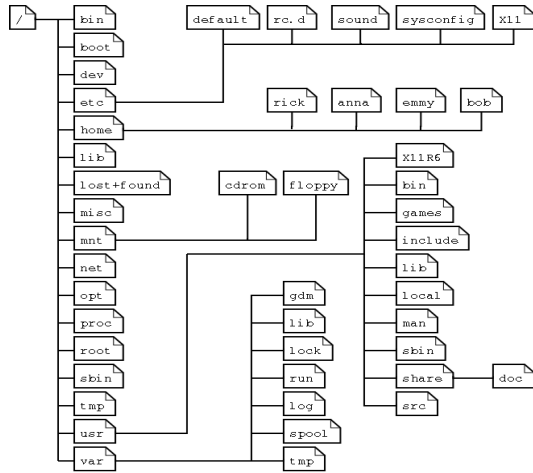
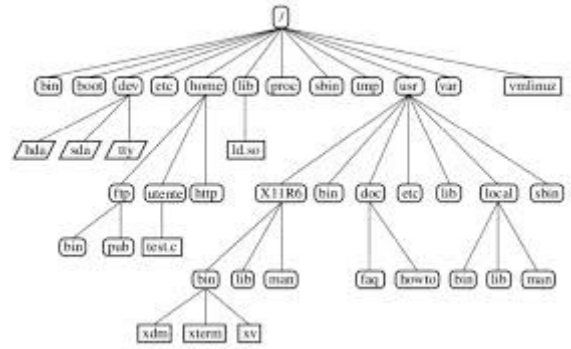
creare directory >> mkdir nome_directory

Albero root con tree

\$ tree

\$ tree -L 1 (primo livello)

Esempio di albero root:



Creare directory con:

\$ mkdir pippo

\$ ls -a

definizione '.' e '..'

. directory corrente

.. directory genitrice

~ (AltGr+1) home utente

Percorso assoluto e relativo

relativo: cioè a partire dalla directory corrente. In questo caso, il nome non comincia mai con la barra (/).

assoluto: a partire dalla radice (/). In questo caso, il nome comincia sempre con la barra.



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

Il programma pwd

L'architettura di un filesystem e le proprietà dei file

Come già accennato Linux (ed ogni sistema unix-like) organizza i dati che tiene su disco attraverso l'uso di un filesystem. Una delle caratteristiche di Linux rispetto agli altri Unix `è quella di poter supportare, grazie al VFS, una enorme quantità di filesystem diversi, ognuno dei quali ha una sua particolare struttura e funzionalità proprie.

Cenni sul VFS da Wikipedia

Virtual file system o Virtual filesystem switch (VFS) è un componente [software](#) che permette al [kernel](#) del [sistema operativo](#) (per esempio il [kernel Linux](#)) un accesso al [file system](#) attraverso delle funzioni standard e indipendenti dal reale file system o dal supporto utilizzato per la memorizzazione dei dati.

Tipi di file

\$ ls -l

Tipo di file		Descrizione
<i>regular file</i>	file regolare	- un file che contiene dei dati (ciò che si intende normalmente per file).
<i>directory</i>	cartella o direttore	d un file che contiene una lista di nomi associati a degli <i>inode</i> .
<i>symbolic link</i>	collegamento simbolico	l un file che contiene un riferimento ad un altro file o directory.
<i>char device</i>	dispositivo a caratteri	c un file che identifica una periferica ad accesso a caratteri.
<i>block device</i>	dispositivo a blocchi	b un file che identifica una periferica ad accesso a blocchi.
<i>fifo o pipe</i>	"coda" o "tubo"	p un file speciale che identifica una linea di comunicazione unidirezionale.
<i>socket</i>	"presa"	s un file speciale che identifica una linea di comunicazione bidirezionale.

Esempi:

Tabella 1.1: I vari tipi di file riconosciuti da Linux

```
$ touch pippo # creare file dati
$ ls -l pippo
```

```
$ mkdir pippo # creare directory
$ ls -l
```

Spiegazione link e link simbolico

```
$ nano pippo # file originale
$ ln pippo paperino
$ ls -i # notare stesso numero di inode
$ ln -s pippo paperino
$ rm pippo # cancelliamo pippo, paperino rimane invariato
$ nano pippo # nuovo file
```

Notare numero inode, permessi (link simbolico prende i permessi dal file originale), hard-link devono essere all'interno dello stesso filesystem, i soft-link possono risiedere su filesystem diversi.

