

1° Corso Introduttivo a GNU/Linux

19 Maggio 2001

LUG Roma3

Filesystem, Comandi di base, Documentazione

19 Maggio 2001

LUG Roma3 Stefano Francesi, Bruno Pelaia, Roberto Manicardi

Riferimenti bibliografici:

Italian Linux Documentation Project

<http://www.pluto.linux.it/ildp>

Guida dell'amministratore di sistema di Linux

<http://www.pluto.linux.it/ildp/guide/GuidaSysadm>

Definizione di Filesystem

Un'organizzazione di dati (*file*) su memorie di massa.

Filesystem esistenti

Linux	<i>EXT2</i>
DOS/Windows	<i>FAT</i>
CDROM	<i>ISO9660 / Rockridge</i>
Win NT/2K	<i>NTFS</i>
OS/2	<i>HPFS</i>
MacOS	<i>HFS</i>

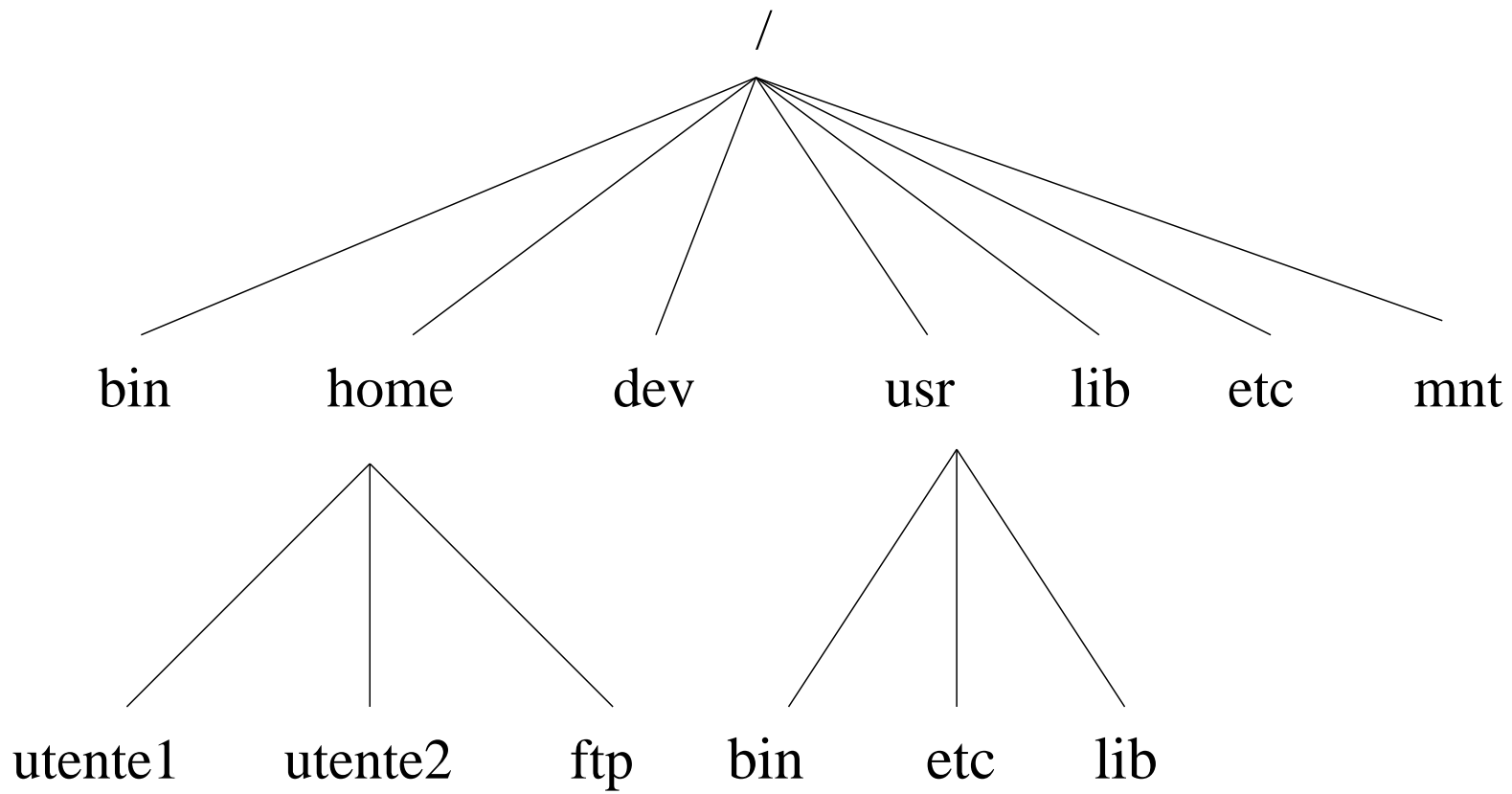


Figure 1: Filesystem EXT2

Filesystem EXT2

Un unico albero delle directory che può essere diviso in parti più piccole (*partizioni*)

Le parti principali sono i filesystem radice, /usr, /var, /etc e /home (Figura 1), e ciascuna ha uno scopo diverso. I loro nomi (e relativi contenuti) sono *convenzioni* del mondo UNIX.

Il meccanismo si presta molto bene alla condivisione di parti di filesystem residenti su dispositivi esterni o su altri computer.

La directory `/etc`

La directory `/etc` contiene tutti i file di configurazione del sistema; sono file di testo e possono essere modificati sia manualmente sia tramite opportuni programmi di utilità.

La directory `/usr`

La directory `/usr` contiene tutti i programmi installati, dunque è spesso contenuta in una partizione a parte, la quale può addirittura risiedere su un'altra macchina. Una sottodirectory comunemente presente è `/usr/local` dove è possibile installare programmi destinati all'uso locale.

La directory `/var`

La directory `/var` contiene i dati che vengono modificati quando il sistema lavora normalmente. È anch'essa (come `/usr/local`) specifica per ciascun sistema, cioè non viene condivisa in rete con altri computer. Contiene le caselle di posta degli utenti, i documenti in attesa di essere stampati e i log del sistema.

La directory /home

Contiene i dati e le configurazioni personali di ciascun utente; ad esempio l'utente `utente1` avrà a disposizione una directory `/home/utente1`

La directory `/lib`

Contiene le librerie *condivise* necessarie al funzionamento dei programmi installati.

La directory `/proc`

La directory `/proc` contiene un filesystem virtuale che non esiste sul disco ma viene creato in memoria. Viene usato per fornire informazioni sul sistema (originariamente elencava i processi attivi, da cui il suo nome).

La directory `/dev`

In Linux ad ogni dispositivo (CDROM, floppy, HD, porte seriali, telecamere, schede sonore e video) sono associati uno o più file speciali detti *device* e contenuti nella directory `/dev`.

I nomi dei file rispecchiano la natura del dispositivo: ad esempio il primo disco sul canale primario IDE si chiamerà `/dev/hda`, mentre il secondo disco SCSI si chiamerà `/dev/sdb`.

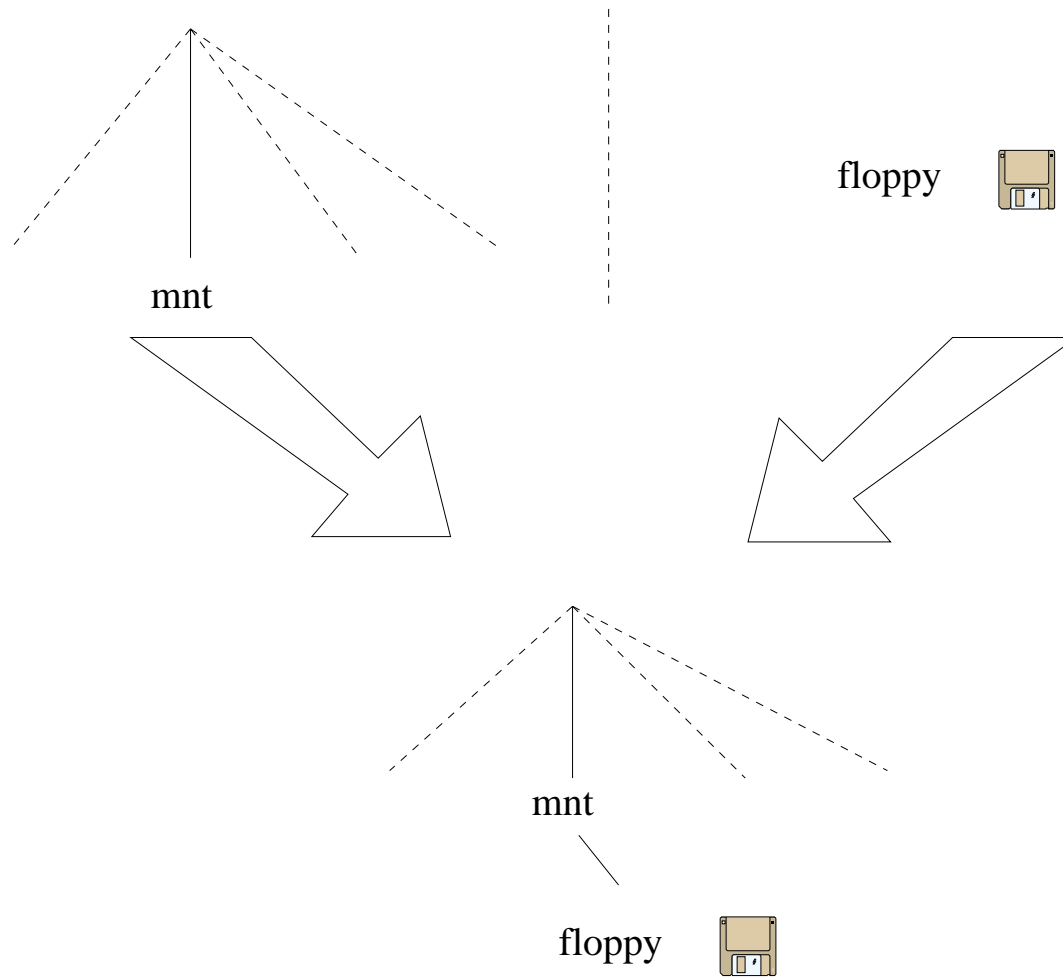


Figure 2: Montaggio dispositivi

Montaggio dispositivi

Per utilizzare un nuovo filesystem all'interno dell'albero delle directory è necessario *montarlo*, ovvero far eseguire al S/O una serie di operazioni che lo facciano sembrare contenuto in una sua sottodirectory.

Una volta montato, il S/O si occupa in maniera trasparente dell'accesso al particolare tipo di filesystem (qualunque esso sia).

Montaggio dispositivi 2

È possibile montare su Linux un qualsiasi filesystem esistente, nonché interi filesystem memorizzati su file (immagini ISO, floppy avviabili, archivi compressi).

Smontaggio dispositivi

È ovviamente necessaria un'operazione inversa di *smontaggio* quando si è finito di utilizzare il dispositivo. È bene dunque non sfilare un floppy montato, pena la perdita di parte dei dati su di esso memorizzati; Linux, infatti, implementa un efficiente sistema di caching dei dati sia in lettura che in scrittura.

Supermount

Una tecnica recentemente introdotta da alcune distribuzioni di Linux è il **supermount** che realizza in maniera trasparente il meccanismo di montaggio/smontaggio dei dispositivi esterni (floppy/cdrom) dando più libertà all'utente.

Permessi e gruppi

Un altro concetto di base per comprendere i comandi di Linux è quello di **permesso**. Linux è un sistema multiutente che punta molto sulla sicurezza del sistema e dei dati dei suoi utenti; ciò significa che il filesystem stesso è in grado di proteggere con opportuni *permessi* singoli file e directory dall'accesso ad opera di utenti non autorizzati od esterni.

Un **gruppo** è un insieme di utenti che condividono permessi particolari tra di loro.

Shell e Terminali

Il sistema operativo UNIX e dunque anche Linux sono nati con la sola interfaccia testuale; ciò significa che è possibile eseguire una qualsiasi operazione utilizzando opportuni comandi di *shell*. Esistono dunque molti comandi per fare piccole cose facilmente componibili l'uno con l'altro. Recentemente si è diffuso molto l'utilizzo delle interfacce grafiche che tuttavia risulta opzionale e non completamente in grado di realizzare tutte le operazioni possibili con una *shell*.

Shell: Comandi Importanti

- Gestione file e directory
- Gestione permessi
- Gestione processi
- Altri comandi utili
- Documentazione

Shell: Gestione dei file

- ★ `cp` : copia file o directory
- ★ `rm` : cancella file (o directory)
- ★ `mv` : sposta file o directory

Shell: Gestione delle directory

- ★ `mkdir` : crea una directory
- ★ `pwd` : mostra la directory corrente
- ★ `cd` : cambia la directory corrente
- ★ `ls` : mostra il contenuto di una directory

Comandi per file : cp

cp SORGENTE DESTINAZIONE

copia file o directory da SORGENTE a DESTINAZIONE

Es.

cp * utente1 copia tutti i file dalla directory corrente alla directory *utente1*

Comandi per file : `rm`

```
rm [OPZIONI] FILE
```

`rm` permette di cancellare file o directory

Es.

`rm -R *` cancella tutti i file e directory anche nelle directory sottostanti a quella corrente

`rm -f *` cancella tutti file nella directory corrente senza chiedere la conferma. **ATTENZIONE!** pericoloso!

`rm -i *` cancella tutti file nella directory corrente chiedendo conferma.

NOTA: Il comportamento di default può variare a seconda della distribuzione.

Comandi per file : mv

```
mv SORGENTE DESTINAZIONE
```

sposta file o directory da SORGENTE a DESTINAZIONE. È usato anche per rinominare file o directory

Es.

```
mv -v pippo pluto
```

 sposta il file *pippo* nella directory *pluto* e mostra cosa sta facendo

```
mv u1 u2
```

 rinomina il file u1 in u2

Comandi per directory : `mkdir`

```
mkdir [OPZIONI] DIRECTORY
```

`mkdir` crea la directory di nome `DIRECTORY`

Comandi per directory : pwd

pwd

pwd mostra la directory in cui si lavora

Es.

Se si sta nella directory `/home/utente1/prova/` il prompt si presenta così:

```
<utente1@dia1 prova>_
```

il comando `pwd` mostrerà proprio `/home/utente1/prova`

Comandi per directory : cd

```
cd [DESTINAZIONE]
```

Il comando `cd` permette di spostarsi nella directory `DESTINAZIONE`

Es.

```
cd /home/utente1/ per spostarsi nella directory /home/utente1
```

```
cd per spostarsi nella propria home directory
```

```
cd .. per spostarsi nella directory del livello superiore
```

Comandi per directory : `ls` (1)

```
ls [OPZIONI] [DIRECTORY]
```

`ls` (abbreviazione di *list*) fornisce la lista dei file contenuta nella directory corrente o nella directory indicata in `[DIRECTORY]`

Es.

`ls` elenco solo dei nomi dei file

`ls -l` elenco dettagliato dei file (tipo di file, permessi, proprietario, gruppo, dimensione, data e ora dell'ultima modifica, nome del file)

Comandi per directory : ls (2)

```
[utente@spartacus tmp]$ ls -l
total 3316
-rw-r--r--  1 utente users      80295 May 18 16:01 Cicli_limite.pdf
-rw-r--r--  1 utente users    138779 May 18 16:01 Funz_descr.pdf
-r--r--r--  1 utente users    283557 May 18 16:01 HStabilita.pdf
-r--r--r--  1 utente users    234103 May 18 16:01 Prima.pdf
-r--r--r--  1 utente users    169493 May 18 16:01 Quarta.pdf
-r--r--r--  1 utente users     77691 May 18 16:01 Quinta.pdf
-r--r--r--  1 utente users    225185 May 18 16:01 Seconda.pdf
-r--r--r--  1 utente users    231629 May 18 16:02 Terza.pdf
-rw-r--r--  1 utente users    815121 May 18 16:01 diagrammi.pdf
-rw-r--r--  1 utente users    182325 May 18 16:01 eserciziavolti1.pdf
-r--r--r--  1 utente users     86688 May 18 16:01 fda_1.pdf
-rw-r--r--  1 utente users       311 May 18 16:01 fda_12.pdf
-rw-r--r--  1 utente users    199750 May 18 16:01 fda_14.pdf
-r--r--r--  1 utente users    160862 May 18 16:01 fda_2.pdf
-r--r--r--  1 utente users    224435 May 18 16:01 fda_3.pdf
-r--r--r--  1 utente users     98553 May 18 16:01 fda_4.pdf
-r--r--r--  1 utente users     60331 May 18 16:01 fda_5.pdf
-rw-r--r--  1 utente users     21525 May 18 12:46 plugger-debug
[utente@spartacus tmp]$
```

Shell: Gestione dei permessi

- ★ `chown` : cambia il proprietario di un file o di una directory
- ★ `chgrp` : cambia il gruppo di un file o di una directory
- ★ `chmod` : cambia i permessi di un file o di una directory

Gestione permessi: chown

```
chown [OPZIONI] UTENTE FILE
```

Cambia il proprietario di FILE in UTENTE

Gestione permessi: `chgrp`

```
chgrp [OPZIONI] GRUPPO FILE
```

Cambia il gruppo di FILE in GRUPPO

Gestione permessi: chmod

```
chmod [OPZIONI] MODO FILE
```

Imposta a MODO i permessi per FILE.

Es.

```
chmod a+rwx miofile tutti i permessi settati per ogni utente
```

```
chmod g-w,o-w miofile impedisce a qualunque utente diverso dal  
proprietario di modificare il file
```

Shell: Gestione processi

- ★ `ps` : fa una lista dei processi attualmente in esecuzione
- ★ `kill` : uccide uno o più processi in esecuzione

Gestione processi : ps

ps [OPZIONI]

ps mostra la lista dei processi attivi

Es. L'esecuzione di ps genera:

```
[utente1@spartacus utente1]$ ps
  PID TTY          TIME CMD
 28344 pts/11    00:00:00 bash
 28357 pts/11    00:00:00 ps
[utente1@spartacus utente1]$
```

Gestione processi : `kill`

```
kill [OPZIONI] PID
```

`kill` manda dei messaggi ai processi indicati con PID. Il messaggio più usato è il SIGKILL (OPZIONI = -9) che permette di *uccidere* un programma.

Shell : Altri comandi utili

- ★ `cat` : visualizza uno o più file
- ★ `less` : visualizza con la possibilità di scorrimento un file
- ★ `halt` : Arresta il sistema

Altri comandi utili : cat

```
cat FILE
```

cat permette di direzionare il contenuto di un file verso un output. Senza redirezionamenti questo viene riversato a video.

cat sfrutta anche il fatto che in Linux le periferiche vengano viste come file, e questo permette di mandare il contenuto di un file audio verso una scheda audio (**Es.** `cat audio.wav > /dev/audio`)

Altri comandi utili : `less`

```
less [OPZIONI] FILE
```

`less` permette di visualizzare un qualsiasi file (in genere di testo) e ne permette lo scorrimento in avanti e indietro. Permette anche la ricerca di una stringa all'interno del file stesso.

Per attivare la modalità ricerca digitare `/` e scrivere la parola desiderata.

Altri comandi utili : halt

```
halt [OPZIONI]
```

`halt` permette di preparare il sistema per lo spegnimento. Nel caso che il computer lo supporti si spegnerà fisicamente alla fine del processo. Per il riavvio o si preme la combinazione di tasti CTRL-ALT-CANC o si digita il comando "reboot".

Shell: Documentazione

- ★ `man` : fornisce la sintassi completa di un comando
- ★ `info` : fornisce altre informazioni dettagliate, quali definizioni, esempi e concetti di base.
- ★ `apropos` : fornisce una lista di comandi attinenti ad un argomento

Documentazione: man

man [OPZIONI] COMANDO

man fornisce la sintassi completa di COMANDO; le OPZIONI sono in genere la *sezione* del manuale legata al COMANDO: questo per motivi di omonimia tra i comandi della shell e alcune funzioni dei linguaggi di programmazione.

Documentazione: info

info COMANDO

`info` arricchisce le informazioni fornite da `man` tramite esempi pratici e la spiegazione di concetti fondamentali legati al comando. Fornisce anche un'ottima documentazione sulle principali librerie standard.

A differenza di `man` le distribuzioni più recenti forniscono sempre maggiori dettagli utilizzando il comando `info`.

Documentazione: apropos

`apropos ARGOMENTO`

Fornisce una lista di comandi e funzioni dei linguaggi di programmazione attinenti all'argomento specificato.

Va ovviamente utilizzato in combinazione con `man` e `info`.

Documentazione

CONSIGLIO: Per imparare ad usare a fondo Linux è necessario **leggere** una gran quantità di documenti. Esistono numerose fonti da cui trarre informazioni, tradotte in moltissime lingue.

Documentazione: Fonti

- ★ *Man pages*
- ★ *HOWTOs*
- ★ *Linux Documentation Project*
<http://www.linuxdoc.org/>
- ★ *Italian Linux Documentation Project*
<http://www.pluto.linux.it/ildp>
- ★ *FAQs*
- ★ *Newsgroups*
- ★ *Mailing lists*
- ★ *Forums*
ad es. <http://forum.lugroma3.org>