

La gestione di Linux

Davide Rondini

3 gennaio 2005



MonteLUG
Montebelluna Linux User Group

Indice

1	Struttura base del file system	3
1.1	Struttura del file system Linux	3
1.2	La gestione dei filesystem	5
1.3	Il montaggio dei filesystem	6
1.3.1	Montaggio di CD-ROM e DVD-ROM	6
1.3.2	Montaggio di floppy disk	7
1.3.3	Montaggio di memorie USB e fotocamere digitali	7
1.4	Il montaggio in automatico dei filesystem	8
1.5	La struttura delle cartelle	9
2	Utenti e permessi	13
2.1	Un sistema multiutente	13
2.1.1	La creazione degli utenti	14
2.2	I permessi	15
2.2.1	Impostare i permessi	17
2.2.2	Cambiare proprietario di un file	19
3	Comandi shell di base	20
3.1	Comandi built-in	20
3.2	Comandi per tutti gli utenti	21
3.3	Comandi per l'amministratore di sistema	31

Elenco delle figure

1	Struttura standard del filesystem Linux	10
2	L'interfaccia di KUser	14
3	Impostazione dei permessi da Konqueror	17
4	La rappresentazione ottale dei permessi di un file	19

Elenco delle tabelle

1	Tipi principali di device	5
---	-------------------------------------	---

1 Struttura base del file system

1.1 Struttura del file system Linux

Un file system è un sistema di immagazzinamento e organizzazione dei file in una periferica di memorizzazione. In pratica il filesystem è l'interfaccia che organizza i dati presenti sul disco in modo che siano comprensibili dall'utente. In altre parole maschera all'utente la disposizione fisica dei dati sul supporto (dischi fissi, floppy, CD memorie flash, ecc...) facendo vedere soltanto la organizzazione in cartelle e file che è comune. Linux supporta quasi tutti i filesystem comunemente utilizzati dalle perifeche di memorizzazione. La lista che segue contiene tutti i tipi di filesystem che si possono specificare attraverso il comando `mount`, anche se la lista di quelli supportati è ancora più lunga¹:

- `adfs`
- `affs`
- `autofs`
- `coda`
- `devpts` - Linux swap
- `ext2` - Filesystem nativo delle partizioni Linux
- `ext3` - Versione journalized di ext2 (è il filesystem che si usa di default nelle partizioni di dischi su cui viene installato Linux)
- `hfs`
- `hpfs`
- `iso9660` - Filesystem dei CD-ROM e DVD-ROM
- `msdos` - Filesystem del DOS: FAT o FAT16.
- `nfs` - Filesystem per condividere file tra macchine UNIX/Linux
- `ntfs` - Filesystem di Windows[®] NT, Windows[®] 2000 e Windows[®] XP

¹Per rendersene conto è sufficiente utilizzare un tool di partizionamento come quelli disponibili nelle principali distribuzioni

- `proc` - È un filesystem virtuale: viene usato da Linux per memorizzare i parametri in memoria
- `qnx4`
- `romfs`
- `smbfs` - Condivisione di file con Windows[®] con Samba
- `sysv` - UNIX filesystem
- `ufs`
- `umsdos`
- `vfat` - Filesystem di Windows[®] 95 e Windows[®] 98. Noto come FAT32.

Le partizioni dei dischi rigidi su cui si installa Linux sono abitualmente di tipo *ext* (`ext2` o `ext3`), anche se recentemente si sta affermando anche un altro filesystem, **ReiserFS** [3]. `ext2` è il filesystem nativo di Linux da molto tempo; `ext3` ne è una evoluzione: è basato sulla stessa struttura di base², ma offre in più la funzione di *journalising*. Il *journalising* è una funzionalità offerta da molti filesystem moderni che consente di evitare il controllo di integrità del filesystem in caso di blocchi del sistema imprevisti. Per far questo il sistema tiene sempre traccia in un file “diario” (appunto il file *journal*) di tutte le operazioni di scrittura in corso. In questo modo in caso di interruzione di una operazione di scrittura, la perdita di dati viene minimizzata, e il filesystem viene ripristinato in pochi secondi; il prezzo da pagare è una piccola perdita di prestazioni del sistema. Un sistema Linux deve poi sempre prevedere una partizione di tipo `swap`, che consente di gestire la memoria virtuale della macchina. Questa partizione non è accessibile agli utenti, ma serve al sistema per estendere la memoria utilizzabile oltre il limite fisico della memoria RAM di sistema, anche se al prezzo di notevoli rallentamenti.

Oltre alle proprie partizioni native, Linux è in grado di leggere e scrivere su quasi tutti i filesystem supportati. L’eccezione, parziale, è rappresentata da NTFS, che fino a non molto tempo fa era supportato soltanto in lettura, ma non in scrittura; le ultime distribuzioni hanno introdotto un supporto parziale in scrittura, e alcuni tool di partizionamento ormai consentono anche di ridimensionare partizioni di questo tipo.

²Tanto è vero che si può passare da un sistema `ext2` a `ext3` e viceversa senza neppure formattare la partizione, ma usando semplicemente usando il comando `tune2fs`.

<code>hd</code>	Dischi fissi e CD-ROM IDE
<code>sd</code>	Dischi fissi SCSI, ma anche memorie USB
<code>fd</code>	Floppy disk
<code>tty</code>	Terminali
<code>lp</code>	Porte parallele
<code>ttyS</code>	Porte seriali

Tabella 1: Tipi principali di device

1.2 La gestione dei filesystem

Anche se può gestire diversi file system Linux utilizza una struttura del file (derivata del modello UNIX) in cui le unità non sono caricate indipendentemente l'una dall'altra, ma dipendono da una radice comune. Ciascuna unità fisica che viene collegata al sistema e *montata*, sia essa un disco fisso, un floppy, un CD-ROM, ecc., apparirà all'interno della stessa struttura delle directory. L'operazione di collegamento di una unità fisica alla struttura delle cartelle si chiama appunto *montaggio* (*mount* in inglese).

Ciascuna unità che viene collegata o inserita nel sistema sarà automaticamente associata ad un *device*³. Un device è un particolare tipo di file che individua, in maniera virtuale, una periferica⁴. Ciascun tipo di periferica è individuato con dei nomi convenzionali: ad esempio i dischi fissi IDE, quelli comunemente presenti nei PC, vengono indicati dalle lettere `hd`. In aggiunta, nel nome del device, la terza lettera distingue eventuali differenti unità dello stesso tipo, mentre il numero che lo conclude indica la partizione dell'unità. Tutti i file dei device sono contenuti nella cartella `/dev`. Per esempio il file `/dev/hda1` identifica la prima partizione del disco IDE primario, ovvero quella in cui abitualmente è installato Windows[®], il quale la identificherebbe come `C:\.` `/dev/fd0` individua invece il lettore floppy disk. La tabella 1 indica i principali tipi di device standard.

Per poter visualizzare il contenuto di un device all'interno dell'albero delle directory è necessario eseguire una operazione di montaggio, attraverso il comando `mount`.

³Questa operazione, se esistono i driver per la periferica in oggetto, viene eseguita in automatico dal sistema operativo.

⁴GNU/Linux segue un principio architetturale di UNIX, comunemente detto "everything is a file", che associa a qualsiasi componente del sistema, comprese le periferiche e la CPU stessa, la possibilità di utilizzarlo come se si trattasse di un comune file.

1.3 Il montaggio dei filesystem

Il comando `mount` permette di collegare all'albero delle directory di Linux il contenuto dei device e di cartelle remote. La sintassi base del comando è :

```
mount -t filesystem device percorso
```

dove *filesystem* è il tipo di filesystem del dispositivo da montare, e va scelto dalla lista del paragrafo 1.1; *device* è il percorso del device da montare; *percorso* è il nome di una directory dentro la quale si vuole far apparire il contenuto del device. In genere in ogni distribuzione viene creata una cartella chiamata `/mnt` in cui sono presenti le cartelle che fanno riferimento a tutte le periferiche escluse le partizioni native Linux, come ad esempio, CD-ROM, floppy, partizioni di altri sistemi operativi, dischi rimovibili, ecc.

Una volta che si voglia togliere l'unità dal sistema, è necessario eseguire l'operazione di smontaggio o *unmount*. Il comando corrispondente è `umount`:

```
umount percorso
```

oppure

```
umount device
```

Avendo infatti precedentemente inserito tutte le informazioni nella fase di `mount`, `umount` conosce già a quale dispositivo è associata la directory e viceversa.

1.3.1 Montaggio di CD-ROM e DVD-ROM

Uno dei più comuni filesystem che vengono montati sull'albero delle directory è ovviamente quello dei CD-ROM. La maggior parte dei PC è dotata di uno o due lettori ottici di tipo IDE, che in genere corrispondono ai device `/dev/hdc` oppure `/dev/hdd`. Per leggere il contenuto di un CD-ROM (oppure un DVD-ROM) è sufficiente quindi inserire il disco e avviare il comando:

```
mount -t iso9660 -o ro /dev/hdc /mnt/cdrom
```

A quel punto, se il device è corretto, il contenuto del CD sarà visibile in `/mnt/cdrom`. Se sono installati più di un lettore, probabilmente le cartelle saranno `/mnt/cdrom0` e `/mnt/cdrom1`. L'opzione `-o ro` dice al sistema di caricare il filesystem in modalità sola lettura, visto che i dischi non supportano la scrittura diretta. Il comando è piuttosto complesso e scomodo, ma nel 99% dei casi inutile: a meno di problemi di configurazione il lettore viene sempre configurato e installato, perciò il sistema operativo conosce già il device che corrisponde all'unità e il tipo di filesystem da utilizzare, perciò il comando si semplifica in:

```
mount /mnt/cdrom
```

Spesso, anche questo è superfluo. Alcune distribuzioni, come Mandrake, Fedora e SUSE, utilizzano un tool chiamato *supermount* che riconosce in automatico l'inserimento del disco e provvede a montarlo.

Quando si decide di rimuovere un disco dal lettore bisogna eseguire prima lo smontaggio (`umount /mnt/cdrom`), altrimenti il cassetto non si aprirà⁵.

1.3.2 Montaggio di floppy disk

Il montaggio dei floppy segue gli stessi principi di quello dei CD-ROM. Il device dove viene caricato il floppy è abitualmente `/dev/fd0`, mentre il tipo di filesystem è `vfat`. Nel caso si fosse in dubbio (i floppy possono essere anche formattati con `ext2`), si può usare il filesystem `auto` o addirittura ometterlo; in questo modo il sistema provvederà da solo ad individuare il tipo di formattazione del dischetto. Il comando è quindi:

```
mount /dev/fd0 /mnt/floppy
```

Anche in questo caso tipicamente il lettore floppy viene riconosciuto in fase di installazione, perciò sarà quasi sempre possibile scrivere semplicemente:

```
mount /mnt/floppy
```

ATTENZIONE: Contrariamente a quanto accade con i CD, il sistema non può bloccare il disco nella sede finché non viene rimosso, perciò fare attenzione a eseguire `umount` *prima* di togliere il dischetto.

1.3.3 Montaggio di memorie USB e fotocamere digitali

Per motivi legati alla implementazione nel kernel Linux delle periferiche USB, esse vengono indicate dal sistema come se fossero dei dischi fissi di tipo SCSI. L'unica conseguenza pratica di questa scelta è che le unità verranno attivate sui device `/dev/sda1`, `/dev/sdb1`, ecc... Tutte le periferiche che memorie di massa che si collegano attraverso l'interfaccia USB avranno questo tipo di device, indipendentemente dal fatto che siano effettivamente pendrive, dischi esterni, lettori MP3, fotocamere digitali o altro. Il comando di montaggio è perciò sempre lo stesso:

```
mount -t vfat /dev/sda1 /mnt/usb
```

⁵Questo per ragioni di sicurezza: se si toglie una unità al sistema senza averlo "avvertito", questo potrebbe andare in crash.

Queste unità abitualmente vengono montate in lettura ed in scrittura, quindi non è quasi mai necessario specificare l'opzione `-o rw`, che lo specifica esplicitamente. Contrariamente ai lettori di floppy e di CD-ROM, questo tipo di periferiche non vengono mai riconosciuti in fase di installazione, ed è quindi sempre necessario eseguire a mano il montaggio con il comando completo. Anche in questo caso comunque alcune distribuzioni recenti appena si collega una nuova periferica eseguono tutto in automatico: caricano il device, creano una cartella in `/mnt`, eseguono il mount, e magari vi aprono una finestra per navigare il contenuto della periferica. Anche se non fosse così è sempre possibile istruire il sistema sulle caratteristiche della periferica e semplificarne il montaggio: ciò è possibile tramite il file `/etc/fstab`, che vedremo nel prossimo paragrafo.

Se la periferica è una fotocamera digitale, una volta eseguito il mount è sufficiente entrare nella cartella di collegamento e sarà possibile aprire le foto con un qualsiasi visualizzatore o software di fotoritocco, copiarle in una cartella sul disco fisso, e anche cancellarle dalla memoria della fotocamera come si fa per qualsiasi altro file.

1.4 Il montaggio in automatico dei filesystem

Come detto, è piuttosto scomodo eseguire il mount di alcune periferiche che si utilizzano spesso dovendo inserire tutti i parametri di configurazione, in particolare se si tratta di partizioni di disco fisso, sempre presenti sul sistema. Per questo motivo esiste un file particolare, chiamato `fstab` e presente nella directory `etc`, che indica al sistema, in fase di avvio, e in caso di mount manuali, tutti i parametri delle periferiche da montare, come device, tipo del filesystem, cartella di mount e altri parametri, come la possibilità di montare in sola lettura, l'accesso degli utenti, ecc. . .

Un esempio di file `fstab` è il seguente:

```

/dev/hda6      swap          swap          defaults      0  0
/dev/hda5      /             ext3          defaults      1  1
/dev/hda7      /home        ext3          defaults      1  2
/dev/hda1      /mnt/windows vfat          auto,rw,noexec 1  0
/dev/cdrom     /mnt/cdrom   iso9660       noauto,user,ro 0  0
/dev/fd0       /mnt/floppy  auto          noauto,user    0  0
/dev/sda1     /mnt/usb     vfat          noauto,user,rw 0  0
devpts        /dev/pts     devpts        gid=5,mode=620 0  0
proc          /proc        proc          defaults      0  0

```

Ciascuna riga del file rappresenta un periferica oppure una partizione del disco. In ogni riga sono indicati nell'ordine: il nome del device della periferica,

la cartella su cui viene montato, il tipo di filesystem e la lista delle opzioni⁶. Le ultime due colonne, quelle con i numeri, indicano rispettivamente la sequenza di smontaggio in fase di spegnimento e la opportunità di eseguire il controllo del filesystem in avvio da parte del comando `fsck`. In genere non è necessario modificarle.

Un volta che un filesystem è stato indicato sul file `fstab` verrà caricato automaticamente in avvio, a meno che non sia specificata l'opzione `noauto`, che consente comunque di montare il sistema specificando soltanto il nome del device o quello della cartella di mount, come già visto per i CD-ROM o i floppy.

Nel file di esempio mostrato le prime tre righe sono relative alle partizioni del disco fisso su cui è installato Linux, in particolare la prima contiene la memoria virtuale, la seconda la radice di tutto il filesystem (`/`), ovvero la partizione su cui è installato il sistema operativo, mentre la terza contiene le cartelle personali degli utenti. La quarta riga include nel filesystem anche una partizione su cui è installato Windows[®]. La quinta e la sesta riga aggiungono il supporto per CD-ROM e dischetti (si noti che contengono l'opzione `noauto` perché devono essere montati al momento dell'inserimento del disco). La settima riga inserisce la possibilità di collegare una periferica USB come una memoria flash. Questa è l'unica riga aggiunta a mano nel file, tutte le altre sono state generate automaticamente in fase di installazione. Le ultime due righe fanno riferimento a file system virtuali, che non corrispondono a dei dispositivi fisici: `devpts` implementa le console virtuali (come ad esempio i terminali che vengono aperti dall'interfaccia grafica, oppure le connessioni remote al sistema), mentre `proc` è una cartella su cui il sistema operativo scrive le informazioni relative all'hardware della macchina.

Oltre a `fstab`, che indica i file system montabili in modo automatico dal sistema, esiste un altro file `/etc/mstab`, che indica invece i filesystem attualmente montati; questo file viene generato dal sistema e non è editabile.

1.5 La struttura delle cartelle

Come già visto in Linux (come in tutti i sistemi UNIX) non esistono le lettere di unità, ma tutti i filesystem dipendono gerarchicamente da un'unica cartella, chiamata radice (*root* in inglese), indicata dalla barra avanti (`/`). Il nome e il contenuto delle cartelle figlie della radice sono abbastanza ben standardizzate, anche se purtroppo esistono molte varianti che differiscono

⁶Per avere la lista di tutte le opzioni e il loro significato, è sufficiente consultare il manuale di `mount` con il comando `man mount`: le opzioni sono quelle specificabili anche attraverso l'opzione `-o` del comando.

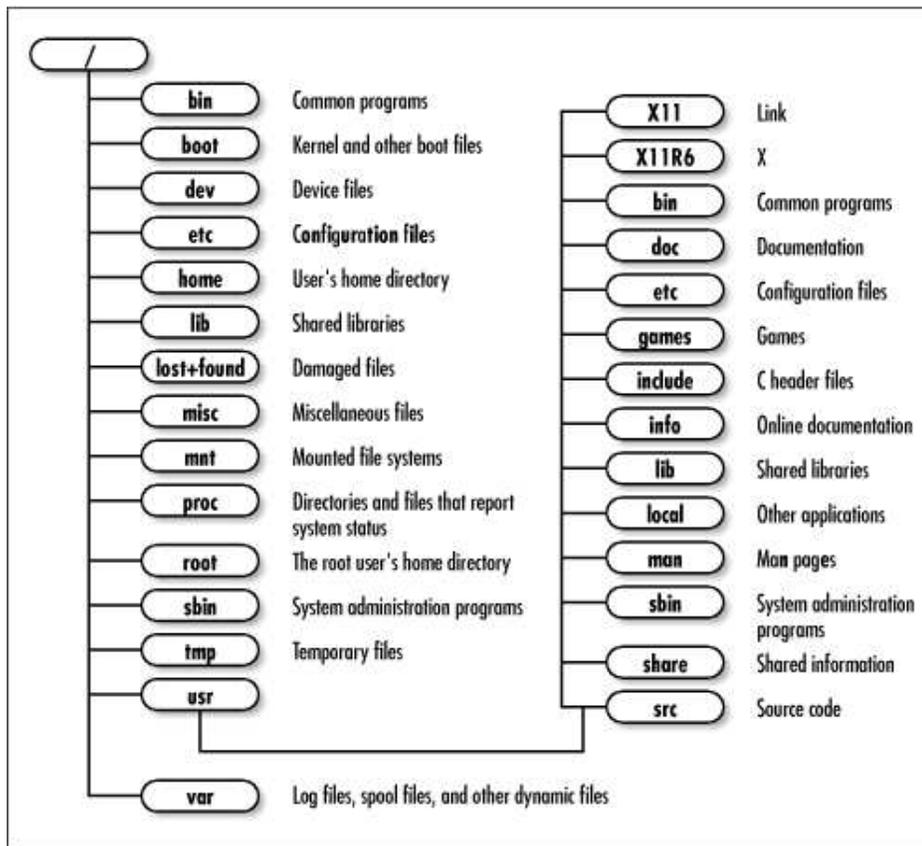


Figura 1: Struttura standard del filesystem Linux

dallo schema di base. La Figura 1 mostra l'albero standard delle cartelle principali. Il contenuto delle cartelle è :

- `/bin` Contiene gli eseguibili dei programmi principali.
- `/boot` Contiene il/i kernel e tutti i file di avvio del sistema operativo.
- `/dev` Contiene i device delle periferiche.
- `/etc` Contiene tutti i file di configurazione del sistema e dei vari programmi.
- `/home` È la cartella con i documenti degli utenti; ciascun utente è proprietario di una sottocartella di `/home`
- `/lib` Contiene le librerie condivise del sistema; in particolare la sottocartella `modules` contiene i moduli del kernel, ovvero i driver delle periferiche.

- /lost+found** Contiene frammenti di file danneggiati e recuperati da fsck (l'equivalente di scandisk per Windows[®]).
- /mnt** Come già visto, contiene le cartelle su cui sono montate le periferiche e le partizioni di altri sistemi.
- /proc** È un filesystem virtuale contenente informazioni sull'hardware della macchina.
- /root** È la cartella *home* dell'amministratore del sistema.
- /sbin** Contiene tutti i comandi di amministrazione del sistema, eseguibili soltanto da **root**.
- /tmp** File temporanei.
- /usr** È una cartella che contiene molti file, e genrealmente tutto ciò che non ha una naturale collocazione in qualche altra cartella. Contiene abitualmente i sorgenti dei programmi (**include**), gli eseguibili dei programmi non di sistema (**bin**), la documentazione (**doc**), le pagine di manuale dei comandi (**man**), librerie condivise (**lib**), file multi-piattaforma (**share**), e molto altro.
- /var** Contiene file dal contenuto variabile, come i log di sistema, le code di stampa, e come **usr**, molto altro.

Contrariamente a Windows[®], come sarà ormai chiaro, il carattere separatore delle cartelle nell'albero è /, e non \. Questa differenza è fonte di non poche difficoltà nelle applicazioni che devono operare in maniera multi-piattaforma, nonché di confusione per gli utenti che si trovano a passare da un sistema all'altro. In realtà questa differenza non ha alcun motivo di essere, è soltanto un retaggio delle prime versioni di DOS, che si sono differenziate dal mondo UNIX, che adotta universalmente la stessa sintassi di GNU/Linux, in maniera del tutto arbitraria.

I file possono essere nascosti alla visualizzazione normale. Per ottenere questa caratteristica è sufficiente che il primo carattere del loro nome sia un punto. Per rendersi conto di questo basta lanciare il comando **ls** nella propria cartella *home*, e poi ripetere il comando con l'opzione **-a**; si vedranno comparire una serie di file che alla prima esecuzione del comando non venivano elencati, tutti con un punto all'inizio del nome.

```
$ ls
Desktop
```

```
Mail
Montellug
pa230001.jpg
play.pls
stampa.ps
$ ls -a
.
..
.aumixrc
.bash_history
.bash_logout
.config
.fonts
.fonts.conf
.local
.opera
.profile
.xfce4
Desktop
Mail
Montellug
pa230001.jpg
play.pls
stampa.ps
```

Oltre ai file nascosti si nota che sono comparsi anche 2 file particolari, indicati con “.” e “..”. Questi file sono dei file speciali che rappresentano rispettivamente il percorso della cartella che si sta visualizzando e la sua cartella genitrice nell’albero delle directory (*parent directory*). Ad esempio in alcune distribuzioni per eseguire comandi che siano installati nella cartella locale e che non siano comandi di sistema è necessario specificarne il percorso, ovvero la cartella “.”: se il comando è, per esempio, `mycommand`, sarà sufficiente scrivere:

```
$ ./mycommand
```

2 Utenti e permessi

2.1 Un sistema multiutente

Il concetto di sistema multiutente risale ai tempi dei primi sistemi UNIX, di cui GNU/Linux riprende l'impostazione generale, quando i computer erano principalmente grandi mainframe a cui erano connessi semplici terminali remoti. Un sistema multiutente è concettualmente l'opposto di un personal computer, ovvero di una macchina che dispone di tutte le periferiche necessarie già da sola e ha abbastanza potenza di calcolo per le operazioni del singolo utente che la utilizza. Anche se per molti versi l'utilizzo tipo che si fa oggi di un PC è molto più vicino a quello del personal computer che di un mainframe, l'architettura di sistema multiutente non solo è rimasta viva in Linux e UNIX, ma molti concetti sono stati implementati anche in sistemi operativi che inizialmente non la prevedevano. I vantaggi sono molteplici:

- ◊ In un sistema multiutente ben configurato, ciascun utente avrà accesso solo e unicamente alle risorse che gli competono, e ai file di sua proprietà, più quelli condivisi, incrementando notevolmente la sicurezza.
- ◊ Anche gli accessi dall'esterno sono vincolati alle risorse disponibili per gli utenti: se un attacco alla macchina dalla rete riuscisse a violare la password di un utente, i dati di tutti gli altri rimarrebbero al sicuro.
- ◊ Soltanto l'amministratore di sistema può modificare le impostazioni, evitando che gli utenti meno esperti possano provocare danni e perdite di dati.
- ◊ Ciascun utente può personalizzare la configurazione nel modo che preferisce, senza influire su quella degli altri utenti.
- ◊ Più utenti possono collegarsi contemporaneamente alla stessa macchina in rete. Un utente può lavorare sempre sulla stessa macchina, pur accedendo in remoto da qualsiasi punto di una rete.

In ogni PC Linux l'accesso avviene sempre identificando l'utente attraverso un nome utente e una password, che è memorizzata nel sistema in forma criptata. In ogni sistema è presente un utente chiamato *root*, che rappresenta l'amministratore di sistema ed ha accesso a tutti i file e le risorse. In pratica l'utente *root* è l'unico a cui qualsiasi operazione è consentita. Tutti gli altri utenti sono identificati con un nome (in genere una abbreviazione del nome vero, oppure un nickname) e hanno la propria password, ma il loro accesso è limitato e ci sono azioni che non possono compiere. Ciascun

utente ha una cartella in cui salvare i propri documenti, a cui gli altri utenti (tranne root) non hanno accesso. Le cartelle degli utenti, come accennato nel § 1.5, sono tutte nella cartella `/home`, e ciascuna ha lo stesso nome dell'utente corrispondente. Solo root ha una cartella a sé stante, appunto `/root`.

2.1.1 La creazione degli utenti

La creazione degli utenti può avvenire in diversi modi: spesso viene eseguita durante il processo di installazione, ma può essere fatta anche a posteriori con diversi tool grafici oppure dalla linea di comando.

Quasi tutte le distribuzioni hanno un tool grafico che permette la gestione degli utenti. Anche KDE dispone di KUser, che è un tool apposito che consente di creare, modificare e rimuovere gli utenti, impostando i parametri di durata della password, i dati anagrafici, la cartella *home*, ecc.

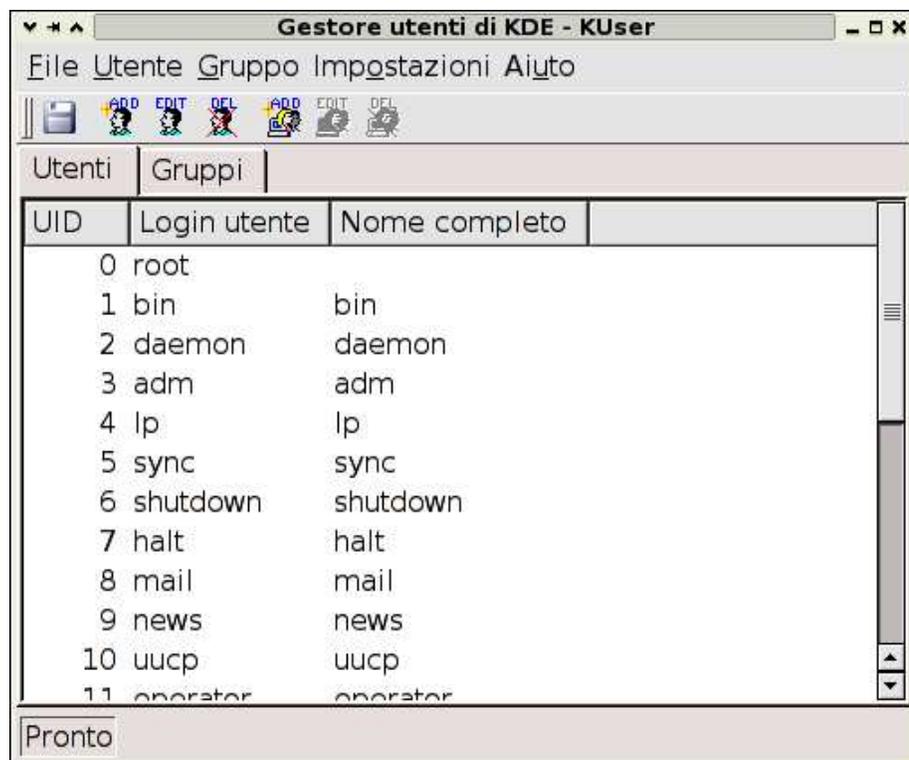


Figura 2: L'interfaccia di KUser

È possibile anche gestire gli utenti anche attraverso la linea di comando, attraverso i comandi `useradd`, `usermod` e `userdel`. Per creare un nuovo utente la sintassi è :

```
useradd -m nome-utente
```

L'opzione `-m` impone di creare la cartella personale dell'utente, utilizzando il percorso di default, (`/home/nome-utente`) se non esistesse già. Se si volesse specificare un percorso diverso alla cartella dell'utente sarebbe necessaria l'opzione `-d percorso`. È possibile specificare anche il gruppo a cui far appartenere il nuovo utente, con l'opzione `-g` e anche specificare il suo numero identificativo (UID, User IDentifier), attraverso l'opzione `-u`. Ciascun utente ha un numero di identificazione univoco, che di default parte da 500 per gli utenti comuni del sistema. L'utente root ha invece l'UID 0. Una volta creato l'utente, per motivi di sicurezza il suo account è disabilitato. È necessario impostare la password dell'utente per renderlo attivo. Questo si fa con il comando `passwd`:

```
passwd nome-utente
```

Verrà richiesto di digitare 2 volte la password e l'utente sarà abilitato.

In maniera analoga alla creazione degli utenti si può intervenire sui gruppi, attraverso i comandi `groupadd`, `groupmod` e `groupdel`. Nel caso dei gruppi non è necessario fornire una password. L'appartenenza degli utenti ai vari gruppi è gestita attraverso il file `/etc/group`, che contiene una riga per ogni gruppo presente, con campi separati da ":". Un riga di esempio:

```
users::100:mario,claudia,francesca
```

Il primo campo rappresenta il nome del gruppo, il secondo la password (che in questo caso è assente, il terzo il Group ID (GID), che è l'equivalente dello User ID per i gruppi, e infine la lista degli utenti appartenenti al gruppo, separati da virgole. Si tenga presente che ciascun utente può appartenere a più gruppi contemporaneamente.

2.2 I permessi

L'architettura di un sistema multiutente è strettamente connessa alla possibilità di controllare l'accesso degli utenti alle risorse e a consentire loro soltanto le operazioni che gli competono. Tutti i sistemi UNIX e quelli da essi derivati implementano in modo nativo nei propri filesystem la presenza per ciascun file o directory di alcune informazioni aggiuntive ai dati in essa contenuti, come la data e l'ora di creazione, il gruppo e l'utente proprietari, e i permessi. Per visualizzare queste informazioni sui file è sufficiente usare

il comando `ls` con l'opzione `-l`. Ecco per esempio le informazioni sul file sorgente di questa guida:

```
-rw-r--r-- 1 davide users 28774 2004-12-18 12:43 lezione.tex
```

Ciascun file dispone di ben nove permessi, raggruppati a tre a tre per il tipo di utente a cui si riferiscono. Ciascun utente ha perciò su ciascun file contenuto nel sistema tre diversi permessi: lettura (**r**, **read**), scrittura, (**w**, **write**) ed esecuzione (**x**, **exec**). Mentre i primi due permessi sono evidenti, il permesso di esecuzione, se è chiaro per il file eseguibile di un programma, o per uno script, lo è meno per una directory. Per una cartella il permesso di esecuzione indica la possibilità per l'utente di aprirla e vederne il contenuto; una cartella di cui si ha il permesso di lettura ma non quello di esecuzione ci consentirà di vedere una lista dei file contenuti, ma non di aprire la cartella stessa. Al contrario si potrà accedere ad una cartella di cui si ha soltanto il permesso di esecuzione ma senza poterne vedere il contenuto. Si tratta di casi un po' particolari, ben rari nella pratica, ma utili a distinguere il significato dei permessi.

I tre permessi appena descritti sono applicati per ciascuno dei seguenti tipi di utente:

- Il proprietario del file (**User**): si tratta di chi ha creato il file, oppure di colui a cui il sistema o l'utente `root` ha assegnato il file⁷;
- Il gruppo di utenti (**Group**) a cui appartiene il proprietario del file;
- Tutti gli altri utenti (**Other**).

Considerando quindi tre permessi per ogni gruppo di utenti si hanno in totale nove permessi per ciascun file o cartella presenti nel sistema. L'utente `root` non viene considerato tra i vari gruppi di utenti, semplicemente possiede di default tutti i permessi per tutti i file.

Per vedere quindi quali permessi sono assegnati ad un file basta interpretare la stringa iniziale nella riga relativa dell'output del comando `ls -l`. Come si vede nell'esempio mostrato sopra sono indicate 10 posizioni: la prima di queste serve ad indicare il tipo di file: un trattino indica un file ordinario, una `d` indica le directory. Seguono i permessi relativi nell'ordine al proprietario, al gruppo e agli altri utenti, raccolti a gruppi di tre, e sempre nell'ordine `rwx`. Se un permesso non è presente viene sostituito da un trattino. Ad esempio il file sorgente di questa guida è un file ordinario (`-`), il proprietario

⁷Il proprietario di un file si può cambiare attraverso il comando `chmod`, che può essere eseguito soltanto da `root`

ne ha soltanto i permessi di lettura e scrittura, ma non di esecuzione⁸ (`rw-`), il gruppo `users`, così come tutti gli altri utenti del sistema possono soltanto leggerlo, ma non modificarlo (`r--r--`).

2.2.1 Impostare i permessi

La modifica dei permessi di un file può essere eseguita soltanto dal suo proprietario e dal superutente. Il modo più immediato per eseguire la modifica è utilizzare un file manager da interfaccia grafica (ad es. Konqueror) e cliccare con il tasto destro sul file interessato. Selezionando “proprietà” e cliccando sul tabulatore “Permessi” apparirà una maschera che permetterà di impostare i permessi in modo rapido (vedi Figura 3). Cliccando su “Permessi avanzati” si avrà accesso alla matrice di permessi completa, con in più i permessi speciali.



Figura 3: Impostazione dei permessi da Konqueror

I permessi si possono impostare anche da linea di comando con il comando `chmod`. Il comando ha una sintassi di questo tipo:

⁸peraltro inutile per un file di testo, a meno che non si tratti di uno script

```
chmod utenti±permessi nomefile
```

il codice indicato come *utenti±permessi* viene indicato in questo modo: si inseriscono prima delle lettere corrispondenti ai tipi di utenti (**u** per il proprietario, **g** per il gruppo, **o** per gli altri utenti, **a** per tutte le categorie insieme, cioè per tutti gli utenti), poi un segno di addizione o di sottrazione a seconda che si voglia aggiungere o togliere un permesso, infine i codici che identificano i permessi. Ad esempio il comando:

```
chmod g+w lezione.tex
```

aggiunge a tutti gli utenti del gruppo **users** il permesso di scrittura al file **lezione.tex**. Invece il comando:

```
chmod go-wx prova
```

toglie agli utenti del gruppo e agli altri utenti la possibilità sia di editare che di eseguire il file “prova”.

Esiste un altro modo di utilizzare il comando **chmod**, che consente di impostare in modo rapido tutti i permessi di un file contemporaneamente, ed è basato sulla rappresentazione ottale dei permessi. Questo metodo è concettualmente piuttosto complesso, ma molto pratico una volta che se ne è appreso il meccanismo. In pratica ciascuna tripletta di permessi **rxw** può essere vista come una cifra ottale, che può essere convertita in un numero decimale da 0 a 7. Supponiamo per esempio che di un file un certo utente abbia solo i permessi di lettura ed esecuzione; la sua rappresentazione sarà: **r-x**. Sostituendo i permessi dati con 1 e quelli negati con 0, si ottiene la cifra ottale 101. Convertendo in decimale questo corrisponde a 5^9 . Ripetendo il procedimento per tutte le categorie di utenti si ottiene che con una tripletta di cifre si riesce a rappresentare tutti i permessi di un file. Se per esempio si vuole impostare i permessi del file **lezione.tex** come **rxr-xr-x** si scriverà il comando:

```
chmod 755 lezione.tex
```

La Figura 4 cerca di rendere più chiaro il procedimento di rappresentazione ottale dei permessi.

$${}^9_5 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

<i>Tipo utente</i>	User	Group	Other
<i>Rappresentazione letterale</i>	r w x	r - x	r - -
<i>Rappresentazione ottale</i>	1 1 1	1 0 1	1 0 0
<i>Potenze di 2</i>	4 2 1	4 2 1	4 2 1
<i>Prodotto</i>	4 2 1	4 0 1	4 0 0
<i>Rappresentazione decimale</i>	7	5	4

Figura 4: La rappresentazione ottale dei permessi di un file

2.2.2 Cambiare proprietario di un file

A volte, senza dover consentire l'accesso di un file a tutti gli utenti, è necessario cambiare il proprietario del file. Questa operazione è consentita soltanto all'utente `root`. Il comando da riga di comando è `chown`, e la sintassi è:

```
chown [-R] utente:gruppo nomefile
```

dove `utente` e `gruppo` sono il nuovo utente e il gruppo a cui assegnare il file o la cartella. È importante sottolineare che bisogna sempre specificare entrambi, separandoli con “:”.

L'opzione `-R`¹⁰ permette di cambiare proprietario ricorsivamente a una cartella e a tutti i file e le sottocartelle contenuti in essa.

¹⁰Si noti come mentre in molti comandi l'opzione ricorsiva `-r` è in minuscolo, in `chown` l'opzione è maiuscola.

3 Comandi shell di base

Qui di seguito è ripostata una lista di comandi più esaustiva possibile in modo che possa essere da riferimento veloce per l'uso della shell¹¹. In ogni caso la via più semplice per verificare che un comando sia installato sulla propria distribuzione e per conoscerne la sintassi il è sufficiente digitare:

```
man nome-comando
```

3.1 Comandi built-in

I comandi built-in sono integrati nella shell che si utilizza, e non sono riconducibili a dei file. Praticamente tutte le shell incorporano un linguaggio di scripting molto potente che consente di essere utilizzato per realizzare operazioni anche molto complesse, con i costrutti classici di un linguaggio di programmazione, come le istruzioni condizionali (`if...then...else...`) e i cicli (`for`, `while`). Queste istruzioni non sono state incluse per semplicità, ma l'intera documentazione sulla shell può essere ottenuta semplicemente con il comando `man bash`.

<code>alias</code>	definisce alias di comandi, ovvero permette di eseguire comandi complessi usati di frequente dando loro nuovi nomi.
<code>bg</code>	manda un processo sospeso in background.
<code>cd</code>	cambia la directory corrente.
<code>exec</code>	sostituisce la shell corrente con un nuovo processo.
<code>exit</code>	chiude la shell.
<code>export</code>	esporta una variabile d'ambiente nelle shell figlie.
<code>fg</code>	porta in foreground un processo.
<code>help</code>	richiama l'help per i comandi builtin.
<code>history</code>	mostra l'history della shell. L'history permette di recuperare e editare i comandi già immessi per velocizzare le operazioni.
<code>jobs</code>	mostra i processi fatti partire dalla shell corrente.
<code>logout</code>	esce da una shell di login.
<code>set</code>	setta una variabile.

¹¹La lista che segue è stata adattata e rimaneggiata sulla base di una reperibile su Internet, all'indirizzo:

<http://guide.supereva.it/manuali/interventi/2003/04/133253.shtml>

Un ringraziamento quindi all'autore del documento originale, Giuseppe Ciaburro

<code>type</code>	mostra dove si trova l'eseguibile di un comando e indica se si tratta di un comando built-in, di uno script o altro.
<code>ulimit</code>	controlla le risorse disponibili per la shell.
<code>umask</code>	setta i permessi di default per la creazione di nuovi file.

3.2 Comandi per tutti gli utenti

I seguenti comandi sono a disposizione di tutti gli utenti. È possibile che non tutti siano installati o configurati di default nella distribuzione che si ha a disposizione.

<code>arch</code>	informazioni sull'architettura del sistema.
<code>bash</code>	la shell (interprete di comandi) normalmente usata.
<code>cat</code>	mostra il contenuto di un file di testo. -n numera le righe. -b salta le righe vuote.
<code>chgrp</code>	cambia il gruppo di appartenenza di un file.
<code>chmod</code>	modifica i permessi di accesso di un file. Vedi § 2.2.1 per le opzioni.
<code>chown</code>	cambia il proprietario di un file o directory.
<code>cp</code>	copia file e directory. -r ricorsivo. -a mantiene gli attributi. -f forza.
<code>cpio</code>	lavora su archivi di file (come i .tar).
<code>cut</code>	taglia un file di testo. -b x-y mostra le colonne da x a y del file; conta i byte. -c x-y mostra le colonne da x a y; conta i caratteri. -f x mostra i campi x separati da TAB. -d specifica un altro delimitatore al posto di TAB.
<code>date</code>	mostra la data e l'ora.
<code>dd</code>	data duplicator, copia da un dispositivo di input su un output. Permette per esempio di creare dischi di avvio. if=xxx nome del file o device di input. of=yyy nome del file o device di output.
<code>df</code>	mostra lo spazio libero sul disco fisso.
<code>dmesg</code>	riporta i messaggi mostrati durante il boot.

du mostra lo spazio usato da file o directory.
 -m dati in megabyte.
 -s mostra solo la directory passata come parametro, non tutte le sue sottodirectory.

echo stampa una stringa.

ed editor di testo line-oriented.

false ritorna 1 come codice di uscita.

fuser identifica i processi che stanno usando un file.

grep trova testo all'interno di un file.
 -c conta solo le righe che contengono il testo cercato.
 -i ignora Maiuscolo/minuscolo.
 -v inverte il senso della ricerca.
 -l elenca solo il nome del file.

gzip comprime e decomprime file (estensione .gz).
 -d decomprime.
 -f forza.
 -r ricorsivo.
 -1 piu' veloce.
 -9 miglior compressione.

hostname mostra e cambia il nome dell'host.
 -f mostra il nome completo (host.dominio).

kill termina forzatamente l'esecuzione di un processo. Per indicare il processo da terminare si utilizza il suo PID. Può essere anche usato per inviare segnali diversi ai processi.
 -s specifica che deve inviare il segnale s.
 -l lista dei segnali.

ln crea collegamenti a file o directory.
 -s crea un link simbolico.

loadkeys carica un layout della tastiera.

ls mostra il contenuto di una directory.
 -a mostra anche i file nascosti (quelli che iniziano per ".").
 -d mostra le directory (senza elencarne il contenuto).
 -f disordinato.
 -i mostra il numero di inode.
 -k dimensione in Kb.
 -l formato lungo.
 --color colorizza i file secondo il tipo.
 -F classifica i file a seconda del tipo.

mkdir crea una directory.

<code>mknod</code>	crea un <i>device</i> (file speciale) a caratteri o a blocchi.
<code>more</code>	visualizza file separando l'output in più pagine.
<code>mount</code>	monta un filesystem. Vedi § 1.3 per la sintassi.
<code>mv</code>	muove o rinomina un file o una directory. -b crea copie di backup. -i chiede conferma.
<code>netstat</code>	mostra informazioni sulle connessioni di rete.
<code>ping</code>	verifica la connessione con un altro computer in rete inviando pacchetti di prova. Per interromperne l'esecuzione premere CTRL-C.
<code>ps</code>	visualizza un elenco dei processi correnti. l formato esteso. u nome utente ed ora di avvio. m informazioni sull'utilizzo della memoria. a mostra anche i processi di altri utenti. r mostra solo i processi attivi. x mostra anche i processi che non controllano un terminale.
<code>pwd</code>	mostra la directory corrente.
<code>rm</code>	cancella file e directory. -d anche directory. -i chiede conferma. -f forza. -r ricorsivo.
<code>rmdir</code>	rimuove una directory.
<code>sed</code>	legge un file e lo processa con determinati comandi.
<code>setserial</code>	setta la porta seriale.
<code>sh</code>	la shell base di UNIX.
<code>sleep</code>	attende senza fare nulla per il numero di secondi specificato.
<code>stty</code>	setta il terminale. Esempio: " <code>stty sane < /dev/ttyX</code> " reimposta al default il terminale X.
<code>su</code>	login come un altro utente (default root). -p preserva l'ambiente. -c esegue un singolo comando.
<code>sync</code>	svuota la cache del disco.
<code>tar</code>	crea od estrae backup di file. x estrae. c archivia. v verbose.

	f file (in cui archiviare o da estrarre).
	z comprime il file con gzip (per file .tar.gz o .tgz).
	j comprime il file con bzip2 (per file .tar.bz2).
touch	cambia la data di un file (se non esiste lo crea). -a ora di accesso. -d cambia la data. -m cambia la data di modifica.
true	ritorna 0 come codice di uscita.
umount	smonta un filesystem. -a smonta tutti i filesystem inclusi in fstab. -t smonta solo i filesystem di un certo tipo.
uname	mostra informazioni sul computer. -m architettura del processore. -n nome dell'host. -r release del kernel. -s nome del sistema operativo. -v data di compilazione del kernel. -a tutte le informazioni.
zcat	mostra il contenuto di un file compresso con gzip (file .gz).
alien	converte pacchetti da/a vari formati (debian deb, redhat rpm, tgz).
apropos	cerca tra i man un determinato argomento.
ar	crea, modifica ed estrae file da un archivio.
arj	comprime file con arj (file .arj).
as	assembler per Linux.
awk	linguaggio di ricerca ed elaborazione di testo (anche gawk, nawk o mawk).
basename	elimina directory e suffissi dai nomi dei file.
bc	una calcolatrice solo testo.
biff	avvisa dell'arrivo di posta.
bison	parser generator (anche yacc).
cal	mostra il calendario.
cdp	player CD audio da riga di comando.
cdplay	versione non interattiva di cdp.
cdrecord	masterizza CD-ROM.
cdrodao	altro comando per masterizzare.
chfn	cambia le proprie informazioni visibili con finger.
chsh	cambia la propria shell di login.
chvt	passa ad un altro terminale virtuale.
clear	pulisce lo schermo del terminale.

<code>cmp</code>	compara due file.
<code>colrm</code>	rimuove le colonne da un file.
<code>column</code>	formatta l'output di un comando o di un file in colonne.
<code>comm</code>	compara due file ordinati linea per linea.
<code>compress</code>	comprime un file (estensione <code>.Z</code>).
<code>convert</code>	converte immagini tra vari formati grafici e ne cambia la risoluzione.
<code>cpp</code>	preprocessore C.
<code>crontab</code>	avvia un processo ad una determinata ora.
<code>csplit</code>	spezza un file in sezioni predeterminate.
<code>ddate</code>	converte la data da gregoriana a discordian.
<code>dialog</code>	per creare finestre e dialog box da shell script.
<code>diff</code>	visualizza le differenze tra due file. -b ignora gli spazi. -B ignora le linee vuote. -i ignora Maiuscole/minuscole.
<code>diff3</code>	confronta 3 file.
<code>dircolors</code>	per settare il colore dei file mostrati da <code>ls</code> .
<code>dirname</code>	stampa solo la directory di un riferimento.
<code>dos</code>	lancia l'emulatore DOSemu.
<code>dpkg</code>	gestore dei pacchetti sulle distribuzioni Debian. -i installa un pacchetto. -r rimuove un pacchetto (<code>-purge</code> rimuove anche i file di configurazione). -s [<code>--info</code>] stampa informazioni su un pacchetto [non] installato. -L [<code>--contents</code>] mostra i file contenuti in un pacchetto [non] installato. -l mostra l'elenco dei pacchetti installati.
<code>dselect</code>	interfaccia per gestire i pacchetti Debian.
<code>dumpkeys</code>	stampa la mappa dei tasti.
<code>emacs</code>	editor di testo molto potente: esiste anche la versione per X-Window, <code>XEmacs</code> .
<code>env</code>	esegue un programma in un determinato ambiente oppure mostra le variabili ambientali.
<code>expand</code>	converte le tabulazioni in spazi.
<code>expr</code>	valuta espressioni (anche aritmetiche).
<code>fdformat</code>	formatta un dischetto. -n non verifica la formattazione.
<code>fdmount</code>	monta un dischetto.

<code>file</code>	determina il tipo di file. -z controlla all'interno dei file compressi.
<code>filesize</code>	stampa la dimensione di un file.
<code>find</code>	cerca un file tra le directory. -name 'xxx' cerca file di nome xxx. -mount esclude la ricerca nei file system nella cartella /mnt). -type X cerca file di tipo X (directory, file).
<code>finger</code>	mostra le informazioni di un utente di un sistema.
<code>flex</code>	per creare analizzatori lessicali (anche lex).
<code>free</code>	mostra lo stato della memoria RAM del sistema. -m dati in megabyte.
<code>fromdos</code>	converte un testo dal formato DOS a quello Unix (anche dos2unix).
<code>ftp</code>	client ftp da riga di comando.
<code>funzip</code>	filtro per utilizzare unzip in una pipe.
<code>g++</code>	compilatore C++.
<code>gcc</code>	compilatore C.
<code>gdb</code>	debugger a riga di comando.
<code>gpm</code>	programma che controlla il mouse nell'interfaccia testuale.
<code>groff</code>	interfaccia per la compilazione di manuali.
<code>groups</code>	stampa il nome del gruppo di un utente.
<code>gzexe</code>	crea eseguibili compressi che si decomprimono al volo.
<code>head</code>	stampa le prime 10 righe di un file. -c x primi x byte. -n y prime y righe.
<code>hexdump</code>	mostra un file in un determinato formato.
<code>id</code>	stampa l'ID e l'UID.
<code>indent</code>	indenta in vari modi un sorgente C.
<code>info</code>	sistema di consultazione dei manuali tramite ipertesti.
<code>install</code>	copia dei file ed assegna permessi e proprietario.
<code>installpkg</code>	installa un pacchetto .tgz sulla distribuzione Slackware.
<code>irc</code>	client irc.
<code>ispell</code>	controllo grammaticale su un file.
<code>kbd_mode</code>	setta la tastiera.
<code>killall</code>	invia un messaggio a tutti i processi con uguale nome. -s specifica che deve inviare il segnale s. -i chiede conferma per ogni processo.
<code>jed</code>	editor di testo con interfaccia.

joe	editor di testo.
join	unisce linee di due file in campi comuni.
last	stampa informazioni sull'ultimo login.
latex	linguaggio di editing di documenti professionale. Questo documento è scritto in L ^A T _E X.
ld	linker (programma complementare al compilatore).
ldd	stampa informazioni sulle librerie condivise.
less	visualizza un file di testo anche di grandi dimensioni (simile a more).
locale	mostra e setta le informazioni sul LOCALE (settaggi internazionali).
locate	cerca un file nel database locale.
logger	scrive un messaggio nei log di sistema.
logname	stampa il nome di login di un utente.
look	mostra le linee che cominciano con una certa stringa.
lpr	stampa un file.
lpq	mostra la coda di stampa.
lprm	cancella un lavoro dalla coda di stampa.
lsdev	mostra informazioni sul proprio hardware.
lynx	browser web solo testo.
m4	macro processor.
make	Consente la compilazione di progetti complessi con un semplice comando.
mail	semplice programma da riga di comando per gestire le e-mail.
man	formatta e mostra le pagine di guida in linea.
mc	Midnight Commander: shell Norton Commander-like.
md5sum	crea una checksum CRC di un file, ovvero calcola una stringa che è univoca per ogni file. Se il file viene modificato, la sua checksum cambia. L'algoritmo MD5 usato da questo comando è considerato molto sicuro.
mesg	mostra o meno messaggi da altri utenti inviati da talk o write.
mev	riporta i mouse-events.
mpg123	player MP3 da riga di comando.
mpg321	altro player MP3 da riga di comando.
mkfifo	crea un file FIFO.
mkisofs	crea un filesystem iso9660 (CD-ROM o DVD), in pratica crea l'immagine di un CD da masterizzare con cdrecord .
nano	un editor di testo.

<code>newgrp</code>	cambia il proprio gruppo di appartenenza.
<code>nice</code>	<p>esegue un programma e ne modifica la priorità d'esecuzione.</p> <p>-20: priorità maggiore (solo root può settare valori negativi).</p> <p>19: minima priorità.</p> <p>n1: numera le righe di un file.</p> <p>nohup: esegue un comando immune da hangup: continua a girare anche dopo un logout.</p>
<code>od</code>	dump di un file in ottale e altri formati.
<code>open</code>	apre un comando in un altro terminale.
<code>passwd</code>	cambia la password di un utente.
<code>paste</code>	unisce linee di files.
<code>patch</code>	applica il risultato di un diff (una patch) ad un file.
<code>pathchk</code>	controlla se un dato file_name è valido.
<code>perl</code>	interprete per script in linguaggio perl.
<code>pico</code>	editor di testo, derivato da nano.
<code>pkgtool</code>	gestore di pacchetti Slackware.
<code>printenv</code>	stampa informazioni sull'ambiente.
<code>procinfo</code>	mostra informazioni tratte dal filesystem /proc.
<code>pstree</code>	mostra un albero di processi.
<code>quota</code>	mostra le percentuali di dischi utilizzabili dagli utenti.
<code>removepkg</code>	elimina un pacchetto Slackware.
<code>renice</code>	modifica la priorità di un processo attivo.
<code>reset</code>	resetta il terminale alle impostazioni iniziali.
<code>rev</code>	inverte le linee di un file.
<code>rpm</code>	<p>gestire pacchetti RedHat, SUSE e Mandrake.</p> <p>-i installa un pacchetto.</p> <p>-e rimuove un pacchetto.</p> <p>-qi [-qip] mostra informazioni su un pacchetto [non] installato.</p> <p>-ql [-qlp] mostra i file contenuti in un pacchetto [non] installato.</p> <p>-qa mostra l'elenco dei pacchetti installati.</p>
<code>scp</code>	permette di copiare file tra diversi computer; utilizza il protocollo di ssh.
<code>script</code>	stampa su un file tutto l'input da tastiera.
<code>sdiff</code>	confronta due file ed elimina le differenze.
<code>setfont</code>	seleziona il font da usare.
<code>setleds</code>	setta i led della tastiera.
<code>setterm</code>	setta gli attributi del terminale.

shar	crea shell archives.
showfont	mostra il font usato.
showkey	mostra i codici della tastiera.
sort	ordina linee di testo. -b ignora gli spazi all'inizio di riga. -f ignora Maiuscolo/minuscolo. -i ignora i caratteri estesi. -n segue l'ordine numerico. -r inverte l'ordine.
split	spezza un file in blocchi di dimensioni date. -l xxx ogni blocco contiene xxx linee. -b xxx [b,k,m] ogni blocco è di xxx byte, kilo, mega.
ssh	permette di collegarsi in remoto ad un altro PC, aprendovi una shell come se si fosse in locale. La connessione è crittografata con algoritmi ritenuti sicuri.
strings	trova linee stampabili in un file.
sudo	esegue un programma come superutente (root).
sum	crea una checksum CRC di un file, ovvero calcola una stringa che è univoca per ogni file. Se il file viene modificato, la sua checksum cambia.
superformat	formatta floppy ad alta capacità.
tail	mostra la fine di un file. -s solo le ultime s righe. -c x ultimi x byte. -f continua a leggere un file, utile se questo viene modificato.
tee	legge da standard input e stampa su stdout e su file. -a append sul file. -i ignora segnali di interruzione.
telnet	apre una sessione telnet. Telnet serve per collegarsi in remoto ad un altro computer come se si fosse loggati sul proprio terminale. Non essendo crittografato, non viene considerato sicuro, e viene progressivamente sostituito da ssh .
test	consente di fare delle verifiche su file (ad esempio se esistono, se sono file ordinari o cartelle, ecc.) oppure di verificare condizioni booleane (confrontare variabili e valori).

<code>time</code>	esegue un programma e mostra informazioni sul tempo di esecuzione e carico.
<code>tload</code>	rappresenta graficamente il carico medio del sistema.
<code>todos</code>	converte un testo da Unix a MSDOS (anche <code>unix2dos</code>).
<code>top</code>	mostra i processi che usano più tempo CPU o memoria.
<code>tr</code>	converte o cancella caratteri.
<code>troff</code>	formatta i documenti come le manpage.
<code>tty</code>	stampa il nome del file terminale connesso allo standard input.
<code>unarj</code>	decomprime file <code>.arj</code> .
<code>uncompress</code>	decomprime file compressi (file <code>.Z</code>).
<code>unexpand</code>	converte spazi in tabulazioni.
<code>uniq</code>	rimuove linee indentiche in file ordinati.
<code>unzip</code>	decomprime file <code>.zip</code> .
<code>updatedb</code>	update di un file database di nomi di file (usato da <code>locate</code>).
<code>uptime</code>	mostra da quanto tempo il sistema è attivo.
<code>users</code>	mostra il nome degli utenti correntemente collegati.
<code>uudecode</code>	decodifica un file ASCII in un binario.
<code>uuencode</code>	codifica un file binario in ASCII.
<code>vi</code>	editor di testo molto potente, ma non intuitivo.
<code>vim</code>	versione migliorata di <code>vi</code> .
<code>vmstat</code>	riporta statistiche sulla memoria virtuale.
<code>w</code>	mostra chi è loggato e cosa sta facendo.
<code>wall</code>	invia un messaggio a tutti i terminali.
<code>wc</code>	conta il nr di byte, parole e linee di un file. <code>-c</code> conta i byte. <code>-w</code> conta le parole. <code>-l</code> conta le newline.
<code>whatis</code>	ricerca nel database della guida per una parola.
<code>whereis</code>	indica dove si trova un determinato file.
<code>which</code>	indica il percorso completo di un file.
<code>who</code>	mostra chi e' loggato.
<code>whoami</code>	indica l'username con cui si e' attualmente loggati.
<code>whois</code>	interroga un database whois, che contiene informazioni sugli utenti di una rete.
<code>wish</code>	interprete per script in linguaggio Tk.
<code>workbone</code>	interfaccia solo testo per ascoltare CD audio.
<code>write</code>	invia un messaggio ad un altro utente.

<code>xargs</code>	appende ad un comando opzioni prese dallo stdin.
<code>yes</code>	emette continuamente una stringa finche' non viene terminato (default <code>y</code>).
<code>zcmp</code>	usa <code>cmp</code> su file compressi.
<code>zdiff</code>	usa <code>diff</code> su file compressi.
<code>zgrep</code>	esegue <code>grep</code> su file compressi.
<code>zip</code>	comprime file con <code>zip</code> .
<code>znew</code>	ricomprime file da <code>.Z</code> a <code>.gz</code> .

3.3 Comandi per l'amministratore di sistema

I seguenti comandi sono a disposizione, a meno di utilizzare `su` oppure `sudo`, soltanto per l'utente `root`, e si tratta di comandi per la configurazione del sistema. È possibile che non tutti siano installati o configurati di default nella distribuzione che si ha a disposizione.

<code>apt-get</code>	software per la gestione dei pacchetti della distribuzione Debian. Permette la risoluzione automatica delle dipendenze.
<code>badblocks</code>	controlla la superficie di un disco fisso.
<code>chattr</code>	cambia gli attributi di un file. <code>-R</code> ricorsivo. <code>a</code> in scrittura appende al file. <code>i</code> il file non può essere modificato, spostato, eliminato, linkato. <code>s</code> quando il file viene cancellato lo spazio su disco viene azzerato. <code>S</code> il file viene sincronizzato immediatamente.
<code>dosfsck</code>	controlla un filesystem DOS.
<code>dumpe2fs</code>	stampa info sul super block e sui blocks del disco fisso.
<code>e2fsck</code>	controlla una partizione <code>ext2fs</code> .
<code>fdisk</code>	manutenzione delle partizioni del disco fisso (anche <code>cfdisk</code>).
<code>fsck</code>	controlla una partizione.
<code>fsck.minix</code>	controlla una partizione <code>minix</code> .
<code>getty</code>	apre una porta <code>tty</code> con richiesta di login (anche <code>agetty</code> , <code>mgetty</code> o <code>mingetty</code>).
<code>halt</code>	spegne il sistema.
<code>hwclock</code>	setta il clock hardware.
<code>ifconfig</code>	configura una interfaccia di rete.

<code>init</code>	lancia i processi di <code>inittab</code> e cambia il runlevel (è il primo processo eseguito dal sistema).
<code>insmod</code>	installa un modulo nel kernel. -f forza anche se le versioni sono diverse.
<code>iptables</code>	amministrazione del firewall IP.
<code>iwconfig</code>	configurazione delle schede di rete wireless.
<code>kbdrate</code>	cambia l'intervallo di ripetizione della tastiera.
<code>kerneld</code>	demone che rimuove/installa automaticamente i moduli non usati/richiesti.
<code>ldconfig</code>	aggiorna l'elenco delle librerie.
<code>lilo</code>	installa il boot loader che consente di selezionare il sistema operativo all'avvio.
<code>losetup</code>	associa dispositivi loop a file.
<code>lsattr</code>	elenco degli attributi dei file. -R ricorsivo. -a tutti i file.
<code>lsmod</code>	mostra informazioni sui moduli del kernel caricati.
<code>mkdosfs</code>	crea una partizione DOS.
<code>mke2fs</code>	crea una partizione <code>ext2fs</code> (il filesystem nativo di Linux).
<code>mkfs</code>	crea una partizione del tipo specificato.
<code>mkfs.minix</code>	crea una partizione minix.
<code>mkswap</code>	crea un dispositivo di swap.
<code>pidof</code>	mostra il PID di un processo.
<code>route</code>	configura i percorsi di rete dei pacchetti. Viene utilizzato per impostare il gateway.
<code>runlevel</code>	stampa il system runlevel corrente e precedente.
<code>shutdown</code>	spegne o riavvia il sistema. -t x aspetta x secondi. -r dopo la chiusura effettua un riavvio (reboot). -h blocca il sistema (halt). -f effettua un reboot veloce. -c blocca uno shutdown in corso.
<code>sysctl</code>	imposta i parametri del kernel.
<code>swapon</code>	attiva un dispositivo o una partizione di swap.
<code>swapoff</code>	disattiva un dispositivo o una partizione di swap.
<code>tune2fs</code>	setta una partizione <code>ext2fs</code> . -c x nr. di reboot prima di un filesystem check. -g setta il gruppo che puo' beneficiare dei blocchi riservati. -l mostra le impostazioni correnti.

	-r setta i blocchi riservati.
	-u setta l'utente beneficiario dei blocchi riservati.
update	svuota periodicamente il buffer del filesystem.
	-S usa il metodo tradizionale (Chiama sync ogni 30 sec.).
	-s x chiama sync ogni x secondi.
	-f y svuota il buffer senza chiamare sync ogni y sec (default: 5).
addgroup	aggiunge un gruppo.
adduser	aggiunge un nuovo utente al sistema (anche useradd).
cfdisk	manipola le partizioni del disco fisso.
chroot	cambia la directory root del sistema.
ctrlaltdel	setta la funzione della combinazione CTRL+ALT+DEL, imponendo il reset immediato oppure permettendo alcune operazioni di recupero. Vista la stabilità di Linux, un comando quasi superfluo...
debugfs	debugga un filesystem ext2.
depmod	verifica le dipendenze del kernel e genera il file modules.dep.
groupdel	elimina un gruppo di utenti.
grub	bootloader alternativo a LILO.
hdparm	mostra e setta molti parametri del disco fisso.
inetd	gestore di molti servizi di rete.
liloconfig	script che aiuta a configurare lilo.
makewhatis	genera il database per il comando whatis.
psupdate	aggiorna il file /etc/psdatabase che rappresenta l'immagine della mappa di sistema attuale (dopo aver ricompilato il kernel).
rdev	setta i dispositivi di un kernel (HD, swap, video mode...).
traceroute	mostra la strada percorsa per raggiungere un dato host.
userdel	elimina un utente del sistema.

Ringraziamenti

Il ringraziamento principale va a tutto il gruppo del MontelLUG, che ha reso possibile questa dispensa e la lezione da cui è nata.

In merito, una citazione va anche alla Fondazione del Museo dello Scarpone di Montebelluna che ha messo a disposizione gli spazi per le attività dell'associazione.

Per questa dispensa in particolare, un ringraziamento particolare va a al Grande Vecchio, il mio supporto tecnico ufficiale per \LaTeX .

Licenza

Copyright © 2005, Davide Rondini. Questo documento viene rilasciato secondo i termini della licenza Creative Commons (<http://creativecommons.org>). L'utente è libero di:

distribuire, comunicare al pubblico, rappresentare o esporre in pubblico la presente opera.

alle seguenti condizioni:

Attribuzione. Deve riconoscere la paternità dell'opera all'autore originario.

Non commerciale. Non può utilizzare quest'opera per scopi commerciali.

No opere derivate. Non può alterare, trasformare o sviluppare quest'opera.

In occasione di ogni atto di riutilizzazione o distribuzione, deve chiarire agli altri i termini della licenza di quest'opera.

Se ottiene il permesso dal titolare del diritto d'autore, è possibile rinunciare a ciascuna di queste condizioni.

Le utilizzazioni libere e gli altri diritti non sono in nessun modo limitati da quanto sopra.

Questo è un riassunto in lingua corrente dei concetti chiave della licenza completa (codice legale), reperibile sul sito Internet <http://creativecommons.org/licenses/by-nc-nd/2.0/legalcode>.

Riferimenti bibliografici

- [1] <http://www.montellug.it>, sito del MontellUG
- [2] <http://www.linux.com>, il sito principale di Linux.
- [3] <http://www.reiserfs.org>, il sito di ReiserFS
- [4] <http://www.linuxquestions.org>, sito colmo di forum per la risoluzione di problemi con Linux
- [5] <http://linux.html.it/guida/index.html>, guida introduttiva di buon livello, come tradizione di HTML.IT
- [6] <http://www.allcommands.com>, sito che raccoglie tutti i comandi di tutti i sistemi operativi. Da non perdere.
- [7] Richard L. Petersen, *La guida completa Linux*, quinta edizione, McGraw-Hill, 2002.