

PGL – Piccole Guide Linux

<http://lgcrsl.altervista.org/pgi/pgl/index.html>

Slackware

Luigi Carusillo

lgcrsl@email.it



I edizione: 30 aprile 2013

II edizione: 7 luglio 2013

III edizione: 9 ottobre 2013

Indice

Introduzione	1
1 Installazione	2
1.1 Operazioni preliminari	2
1.1.1 Scelta della tastiera italiana	2
1.1.2 Identificazione	2
1.2 Installazione	2
1.2.1 Avvio dell'installazione	2
1.2.2 Partizionamento del disco fisso	2
1.2.3 Scelta della mappatura della tastiera	2
1.2.4 Attivazione della partizione di swap	2
1.2.5 Scelta delle partizioni	3
1.2.6 Scelta della sorgente dell'installazione	3
1.2.7 Scelta delle serie di software	4
1.2.8 Scelta della modalità di installazione	4
1.2.9 Creazione di una chiave USB d'avvio	5
1.2.10 Configurazione ed installazione di <i>LILO</i>	5
1.2.11 Configurazione del mouse	5
1.2.12 Caricamento di <i>GPM</i> all'avvio del sistema	5
1.2.13 Configurazione di rete	5
1.2.14 Scelta dei servizi attivi all'avvio del sistema	5
1.2.15 Configurazione dei caratteri del terminale	5
1.2.16 Impostazione del fuso orario	6
1.2.17 Scelta dell'ambiente grafico predefinito	6
1.2.18 Impostazione della password di root	6
1.2.19 Fine dell'installazione	6
1.3 Pacchetti supplementari	6
2 Configurazione	7
2.1 Impostazione del livello operativo	7
2.2 Impostazione della lingua italiana	7
2.3 Sincronizzazione dell'ora	7
2.4 Impostazione del volume sonoro	7
2.5 Impostazione del montaggio dei file system	8
2.6 Configurazione del firewall	8
2.7 Configurazione di rete	8
2.7.1 Creazione del proprio dominio	8
2.7.2 Configurazione della connessione via cavo	8
2.7.3 Configurazione della connessione senza fili	8
2.7.4 Configurazione della connessione mediante chiavetta	9
2.7.5 Programmi grafici alternativi	9
2.7.6 Criteri di scelta	10
2.8 Installazione e configurazione della stampante	10
2.9 Creazione di un utente	10

3	Caratteristiche di Linux	11
3.1	La multiprocessualità	11
3.2	La multiutenza	11
3.3	La sensibilità alla grandezza delle lettere	12
3.4	I modelli	12
3.4.1	Le stringhe letterali	12
3.4.2	Le espressioni regolari	12
3.5	L'avvio del sistema	14
3.6	La documentazione	14
3.7	Terminologia	14
4	La shell	15
4.1	Le istanze di <i>Bash</i>	15
4.1.1	Istanza interattiva di accesso: la shell di accesso	15
4.1.2	Istanza interattiva non di accesso: la shell di sessione	15
4.1.3	Istanza non interattiva: la shell di script	15
4.1.4	I file temporanei associati	16
4.2	I caratteri speciali	16
4.3	Le variabili	18
4.3.1	Variabili globali o variabili d'ambiente	18
4.3.2	Variabili locali o variabili di shell	18
4.3.3	Le variabili dell'invito al comando	19
4.4	Gli alias	19
4.5	Il controllo dei lavori	20
4.5.1	Esecuzione	20
4.5.2	Piani di esecuzione	21
4.6	I comandi interni	22
4.6.1	Informazioni	22
4.6.2	Documentazione	22
4.6.3	Gestione delle directory e dei file	22
4.6.4	Cronologia dei comandi	22
4.6.5	Esecuzione di comandi	23
4.6.6	Ripetizione	23
4.6.7	Programmi nulli	23
5	Gestione dei gruppi e degli utenti	24
5.1	I gruppi, gli utenti e le password	24
5.1.1	I gruppi	24
5.1.2	Gli utenti	24
5.1.3	Le password	24
5.2	Gestione dei gruppi	25
5.2.1	Creazione	25
5.2.2	Modifica	25
5.2.3	Eliminazione	25
5.3	Gestione degli utenti	25
5.3.1	Creazione	25
5.3.2	Modifica	26
5.3.3	Eliminazione	26
5.4	Programmi grafici alternativi	26
6	Gestione del file system	27
6.1	Il file system	27
6.1.1	Gerarchia del file system	27
6.1.2	I percorsi	28
6.1.3	I file system supportati	29
6.2	Montaggio dei file system	29
6.2.1	Configurazione di <i>Mount</i> : la tabella dei file system	29

6.2.2	<i>Mount</i>	30
6.2.3	<i>Umount</i>	31
6.3	Formattazione: la creazione dei file system	31
6.3.1	Creazione di un file system ext4	31
6.3.2	Creazione di un file system ntfs	31
6.3.3	Creazione di un file system FAT32	32
6.4	Gestione delle partizioni	32
6.5	Programmi grafici alternativi	33
7	Gestione delle directory e dei file	34
7.1	I file	34
7.2	Gli attributi dei file	34
7.2.1	I permessi	34
7.2.2	Le categorie di utenti	35
7.2.3	I permessi predefiniti	35
7.2.4	La marcatura temporale	35
7.3	Gestione degli attributi	36
7.3.1	Gestione dei permessi	36
7.3.2	Gestione delle proprietà	36
7.3.3	Gestione della marcatura temporale	37
7.4	Elencazione	37
7.4.1	<i>Ls</i>	37
7.5	Ricerca	38
7.5.1	<i>Slocate</i>	38
7.5.2	<i>Find</i>	39
7.6	Navigazione	39
7.6.1	<i>Pwd</i>	39
7.6.2	<i>Cd</i>	39
7.7	Visualizzazione e modifica	40
7.8	Creazione	40
7.8.1	<i>Mkdir</i>	40
7.8.2	<i>Touch</i>	40
7.8.3	<i>Ln</i>	40
7.9	Rinomina	41
7.9.1	<i>Mv</i>	41
7.10	Spostamento	41
7.10.1	<i>Mv</i>	41
7.11	Copia	41
7.11.1	<i>Cp</i>	41
7.12	Eliminazione	42
7.12.1	<i>Rm</i>	42
7.13	Gestione di archivi, Compressione e Decompressione	42
7.13.1	<i>Tar</i>	42
7.13.2	<i>Gzip</i> e <i>Gunzip</i>	42
7.13.3	<i>Bzip2</i> e <i>Bunzip2</i>	43
7.13.4	<i>Lzma</i> e <i>Unlzma</i>	43
7.14	Programmi grafici alternativi	43
8	Gestione dei testi	44
8.1	Ricerca	44
8.1.1	<i>Grep</i>	44
8.2	Visualizzazione	45
8.2.1	<i>Cat</i> , <i>Tac</i> , <i>Rev</i>	45
8.2.2	<i>Sed</i>	45
8.2.3	<i>More</i> , <i>Less</i>	45
8.2.4	<i>Head</i> , <i>Tail</i>	45
8.3	Modifica	46

8.3.1	<i>Uniq, Cut</i>	46
8.3.2	<i>Sed</i>	47
8.3.3	<i>Nl, Sort</i>	48
8.3.4	<i>Cat, Paste, Join</i>	49
8.3.5	<i>Split</i>	49
8.4	Confronto	49
8.4.1	<i>Comm, Sdiff</i>	49
8.4.2	<i>Merge</i>	50
8.5	Verifica	50
8.5.1	<i>Sum, Cksum, Md5sum</i>	50
8.5.2	<i>Sha1sum, Sha224sum, Sha256sum, Sha384sum, Sha512sum</i>	51
8.6	Conteggio	51
8.6.1	<i>Wc</i>	51
8.7	Controllo ortografico	51
8.7.1	<i>Aspell</i>	51
8.8	Formattazione	52
8.8.1	<i>Fold</i>	52
8.9	I filtri	52
8.9.1	I filtri obbligati: <i>Tee, Tr, Xargs</i>	52
8.9.2	Esempi vari	53
9	Gestione dei pacchetti	54
9.1	Gestione manuale	54
9.1.1	<i>Installpkg, Upgradepkg e Removepkg</i>	54
9.1.2	<i>Pkgtool</i>	54
9.1.3	<i>Spkg</i>	55
9.2	Gestione automatica	55
9.2.1	Aggiornamento di <i>Slackware</i>	55
9.2.2	<i>Slackpkg</i>	55
9.2.3	<i>Slapt-get</i> e <i>Gslapt</i>	57
9.3	Conversione di pacchetti estranei	58
9.3.1	<i>Alien</i>	58
9.3.2	<i>Rpm2tgz</i>	58
10	Creazione guidata dei pacchetti	59
10.1	Creazione manuale	59
10.1.1	Operazioni preliminari	59
10.1.2	Metodo classico (<i>SlackBuild</i>)	59
10.1.3	Metodo alternativo (<i>SLKBUILD</i>)	59
10.2	Creazione automatica	60
10.2.1	<i>Slapt-src</i> e <i>Sourcery</i>	60
11	Creazione <i>ex novo</i> dei pacchetti	61
11.1	Creazione manuale	61
11.1.1	Operazioni preliminari	61
11.1.2	Configurazione	61
11.1.3	Compilazione	62
11.1.4	Installazione di prova	62
11.1.5	Pulitura	62
11.1.6	Elaborazione della documentazione	62
11.2	Esempi	64
11.2.1	Creazione del pacchetto di <i>Sitecopy</i>	64
11.2.2	Creazione del pacchetto di <i>Medit</i>	65
	Bibliografia	67
	Indirizzi	68

Introduzione

Linux è un kernel, cioè un sistema operativo minimale, simile a quello del sistema operativo *Unix*. *Linux* è stato sviluppato da Linus Torvalds, che ne rilasciò la prima versione (0.01) il 25 agosto 1991. Esistono parecchi sistemi operativi basati su *Linux*, differenti tra loro sia nella forma che nella sostanza, che prendono il nome di distribuzioni.

Slackware (**Slackware Linux**, più correttamente), il cui autore è Patrick J. Volkerding, è la più antica distribuzione Linux attualmente esistente: la sua prima versione (1.0) risale al 17 luglio 1993. *Slackware*, concepita per essere il più possibile simile al sistema operativo *Unix*, ha due caratteristiche che la distinguono: stabilità ed essenzialità.

La stabilità

La stabilità, sinonimo di affidabilità, è dovuta all'inserimento di programmi, non necessariamente nella loro versione più recente, meticolosamente collaudati sia nella loro funzionalità che nella loro capacità di integrazione con il sistema.

L'essenzialità

L'essenzialità, sinonimo di sobrietà, dipende dall'assenza di strumenti grafici per le configurazioni e le operazioni di sistema, che infatti richiedono di essere effettuate in modo consapevole.

L'approccio a *Slackware*

Slackware è fortemente caratterizzata da Volkerding, il quale ha definito in modo rigido la natura della distribuzione: l'esempio più evidente consiste nella gestione dei pacchetti precompilati, che volutamente esclude il controllo automatico delle dipendenze. Questo elemento da solo è sufficiente a distinguere sommariamente due possibili tipi di approccio a *Slackware* da parte dell'utente:

- l'atteggiamento purista, rigorosamente rispettoso dello spirito *Slackware* (e di Volkerding), ma essenzialmente teorico;
- l'atteggiamento libero, di chi adatta la distribuzione ad esigenze di semplificazione, ma tendenzialmente banalizzante.

Un atteggiamento equilibrato sta nell'eventuale libero ricorso alle *comodità*, senza ignorare i meccanismi che ne stanno alla base.

Capitolo 1

Installazione

1.1 Operazioni preliminari

Le operazioni preliminari avvengono in modalità testuale e precedono l'avvio dell'installazione.

1.1.1 Scelta della tastiera italiana

Per la mappatura italiana della tastiera ci sono tre possibilità. Per esempio, scegliere:

qwerty/it.map

1.1.2 Identificazione

Identificarsi come root:

```
slackware login: root
```

1.2 Installazione

1.2.1 Avvio dell'installazione

L'installazione è avviata, in qualità di root, dal programma *Setup*:

```
root@slackware:/# setup
```

Da qui l'installazione avviene in modalità semigrafica, fatta eccezione per l'eventuale partizionamento.

1.2.2 Partizionamento del disco fisso

È possibile dividere il disco fisso in partizioni con *Cfdisk*. Ma sarebbe preferibile aver partizionato il disco fisso precedentemente e separatamente dall'installazione, per evitare di effettuare insieme due operazioni (partizionamento e installazione) per loro natura molto delicate.

Vedere la **Sez. 6.4** a pag. 32.

1.2.3 Scelta della mappatura della tastiera

Ripetere la selezione della tastiera italiana. Come prima:

qwerty/it.map

1.2.4 Attivazione della partizione di swap

Attivare la partizione di swap, individuata automaticamente.

1.2.5 Scelta delle partizioni

Scegliere e formattare (per esempio, nel file system [ext4](#)) la partizione root (/) e scegliere (e, se necessario, formattare) le altre partizioni da inserire nel file `/etc/fstab`.

Se non si hanno particolari esigenze e soprattutto in presenza di più distribuzioni Linux nel disco fisso, conviene includere ogni distribuzione Linux interamente in una partizione root, evitando partizioni dedicate a `/home/`, `/usr/local/` ecc.

I permessi predefiniti in Windows

Come tutte le altre partizioni anche le partizioni Windows sono di proprietà di root (vedere anche la **Sez. 7.2** a pag. 34). La gamma dei permessi ordinari (riferiti a root e agli utenti diversi da root) disponibili sia per le directory che per i file è:

```
rw-rw-rw- 777
```

In `/etc/fstab` sono possibili quattro combinazioni di opzioni `fmask` (relativa ai file) e `dmask` (relativa alle directory).

1. `fmask=177,dmask=077`: accesso solo a root in lettura e scrittura.

```
File      rw----- 600  (= 777 - 177)
Directory rwx----- 700  (= 777 - 077)
```

2. `fmask=333,dmask=222`: accesso a tutti solo in lettura.

```
File      r--r--r-- 444  (= 777 - 333)
Directory r-xr-xr-x 555  (= 777 - 222)
```

3. `fmask=133,dmask=022`: accesso diversificato, a root in lettura e scrittura e agli altri solo in lettura.

```
File      rw-r--r-- 644  (= 777 - 133)
Directory rwxr-xr-x 755  (= 777 - 022)
```

4. `fmask=111,dmask=000`: accesso a tutti in lettura e scrittura.

```
File      rw-rw-rw- 666  (= 777 - 111)
Directory rw-rw-rw- 777  (= 777 - 000)
```

Si produce così il file `/etc/fstab`. Per esempio:

```
/dev/sda5      swap          swap          defaults      0    0
/dev/sda6      /              ext4          defaults      1    1
/dev/sda7      /mnt/sda7     ext4          defaults      1    2
/dev/sda8      /mnt/sda8     ext4          defaults      1    2
/dev/sda9      /mnt/sda9     ext4          defaults      1    2
/dev/sda10     /mnt/sda10    ext4          defaults      1    2
/dev/sda11     /mnt/sda11    ext4          defaults      1    2
/dev/sda1      /mnt/sda1     ntfs          fmask=333,dmask=222 1    0
/dev/sda12     /mnt/sda12    ntfs-3g       fmask=111,dmask=000 1    0
#/dev/cdrom    /mnt/cdrom    auto          noauto,owner,ro,comment=x-gvfs-show 0    0
/dev/fd0       /mnt/floppy   auto          noauto,owner  0    0
devpts        /dev/pts      devpts        gid=5,mode=620 0    0
proc          /proc         proc          defaults      0    0
tmpfs         /dev/shm      tmpfs         defaults      0    0
```

Il file è provvisorio e dovrà essere modificato in seguito: vedere la **Sottosez. 6.2.1** a pag. 29.

1.2.6 Scelta della sorgente dell'installazione

Confermare la sorgente dell'installazione, individuata automaticamente.

1.2.7 Scelta delle serie di software

Le serie di software corrispondono alle categorie in cui sono suddivisi i pacchetti precompilati inclusi nei CD o nel DVD dell'immagine ISO di *Slackware*.

A	Programmi di base.
AP	Programmi che non richiedono il server grafico X.
D	Programmi per lo sviluppo.
E	Il programma di videoscrittura Emacs.
F	Documentazione.
K	Il codice sorgente del kernel (Linux).
KDE	L'ambiente desktop KDE.
KDEI	I pacchetti di internazionalizzazione del KDE.
L	Librerie di sistema.
N	Programmi di rete.
T	La distribuzione te \TeX .
TCL	Il linguaggio di scripting TCL.
X	Il server grafico X (X Window System).
XAP	Programmi che richiedono il server grafico X.
XFCE	L'ambiente desktop Xfce.
Y	Giochi per terminale BSD.

Facoltative le serie E, K, Y; da evitare la serie T (successivamente, per l'installazione di \TeX Live, vedere [2]).

L'immagine ISO di *Slackware* comprende anche la directory `/extra/`, che contiene diversi pacchetti supplementari, installabili successivamente.

1.2.8 Scelta della modalità di installazione

Tra le modalità di installazione disponibili scegliere una delle due seguenti.

full	Installa tutti i pacchetti delle serie di software appena selezionate;
menu	Consente di scegliere, per le serie di software selezionate, i pacchetti da installare.

Installazione completa

La modalità **full** non richiede alcun intervento, ma per il KDE bisogna escludere la serie **KDEI**, per evitare di installare i pacchetti relativi al supporto di tutte le lingue disponibili. L'installazione dei pacchetti relativi all'italiano deve essere rimandata a dopo.

Installazione selettiva

La modalità **menu** mostra di ogni serie, eccetto **KDEI**, tutti i pacchetti già selezionati. È possibile deselectare i pacchetti non desiderati (per esempio, per la serie **N** è possibile scartare i pacchetti **sendmail** e **sendmail-cf** se a *Sendmail* si preferisce *Postfix*). Invece, i pacchetti della serie **KDEI** appaiono non selezionati: scegliere quelli relativi all'italiano.

Poiché alcuni programmi grafici utili appartengono al KDE (a partire dal *KDM*), anche se non si usa abitualmente quell'ambiente desktop conviene installare il KDE almeno parzialmente. Segue un elenco dei pacchetti che potrebbero essere salvati:

ark	kdeartwork	kdf
k3b	kdelibs	ktorrent
kactivities	kdemultimedia	libkexiv2
kcalc	kdepim	libktorrent
kde-baseapps	kdepim-runtime	oxygen-gtk2
kde-runtime	kdepimlibs	oxygen-icons
kde-wallpapers	kdeplasma-addons	kde-l10n-it
kde-workspace	kdesdk	

1.2.9 Creazione di una chiave USB d'avvio

Per i casi di emergenza viene proposto di creare una chiave USB di avvio: si può saltare.

1.2.10 Configurazione ed installazione di *LILO*

La configurazione e l'installazione di *LILO* dipendono dai sistemi operativi presenti nel disco fisso. Assumendo che Windows sia sempre presente (importante è che Windows sia installato prima di tutti), la differenza sta nella presenza di uno solo o più sistemi Linux.

- nel caso di un solo sistema Linux (*Slackware*), *LILO* dovrà essere installato nell'MBR e farsi carico di avviare sia Windows che Linux;
- nel caso di più sistemi Linux, il gestore di avvio (*LILO* o *GRUB*) di un sistema Linux dovrà essere installato nell'MBR e assumersi l'incarico di avviare tutti i sistemi operativi (incluso Windows), mentre per ogni altro sistema Linux *LILO* (o *GRUB*) dovrà essere installato nel primo settore della relativa partizione root e potrà limitarsi ad avviare solo il proprio sistema.

Anche maneggiare *LILO* è operazione delicata, che richiede attenzione. Avviare il programma *LILOconfig* nella modalità d'installazione **expert**.

1. Con **Begin** creare il file di configurazione `/etc/lilo.conf` e scegliere: il tipo di terminale (standard o del frame buffer), dove installare *LILO* (**MBR** o **Root**) e il tempo di attesa prima dell'avvio di un sistema operativo (per esempio, 30 secondi).
2. Con **Linux** aggiungere la partizione dove deve essere installata *Slackware*.
3. Eventualmente, ancora con **Linux** aggiungere le altre partizioni Linux (non importa se Linux vi è già presente o deve esservi ancora installato).
4. Eventualmente, con **Windows** aggiungere la partizione Windows.
5. Con **View** controllare che il file `lilo.conf` sia corretto. In caso di errori o ripensamenti, con **Begin** si annulla tutto e si ricomincia.
6. Con **Install** installare *LILO*.

1.2.11 Configurazione del mouse

Scegliere il tipo di mouse corrispondente al proprio. Per esempio: **usb**.

1.2.12 Caricamento di *GPM* all'avvio del sistema

Sì.

1.2.13 Configurazione di rete

No, rimandare a dopo.

1.2.14 Scelta dei servizi attivi all'avvio del sistema

Per esempio:

```
rc.cups
rc.fuse
rc.inetd
rc.messagebus
rc.ntpd
rc.syslog
rc.sshd
```

1.2.15 Configurazione dei caratteri del terminale

No.

1.2.16 Impostazione del fuso orario

Scartare l'ora UTC/GMT e scegliere l'ora CET, specificando:

Europe/Rome

Corrispondenze: CET = UTC+1 (ora solare)/UTC+2 (ora legale).

Acronimi

UTC	: Coordinated Universal Time	= tempo coordinato universale
GMT	: Greenwich Mean Time	= tempo medio di Greenwich
CET	: Central European Time	= tempo dell'Europa centrale

1.2.17 Scelta dell'ambiente grafico predefinito

Scegliere tra i seguenti ambienti desktop e gestori di finestre:

xinitrc.kde
xinitrc.xfce
xinitrc.fluxbox
xinitrc.blackbox
xinitrc.wmaker
xinitrc.fvwm2
xinitrc.twm

1.2.18 Impostazione della password di root

Creare la password di root.

1.2.19 Fine dell'installazione

L'installazione è terminata. Riavviare con: Ctrl-Alt-Canc.

1.3 Pacchetti supplementari

Il DVD (o il quarto CD) d'installazione comprende la directory `/extra/`, che contiene alcuni pacchetti supplementari, installabili successivamente.

Capitolo 2

Configurazione

Generalmente le configurazioni di sistema più rapide e precise si effettuano modificando direttamente i file di configurazione. Una precauzione indispensabile: fare una copia di sicurezza di ciascun file di configurazione che deve essere modificato (i file duplicati vengono rinominati con l'aggiunta del suffisso `.OLD`).

Le configurazioni di sistema sono effettuate da `root` e richiedono il riavvio del sistema per diventare operative.

2.1 Impostazione del livello operativo

Al primo riavvio del sistema la sessione parte in modalità testuale (livello operativo 3), cioè su terminale. Dopo l'autenticazione (come `root`), per avviare la sessione in modalità grafica (livello operativo 4) con l'ambiente desktop selezionato:

```
# startx
```

Sostituire nel file `/etc/inittab` il livello operativo 3 con il 4:

```
id:4:initdefault:
```

Dal riavvio successivo verrà eseguito il *KDM*, che consentirà l'avvio della sessione in modalità grafica, l'autenticazione ed eventualmente il cambiamento dell'ambiente desktop.

2.2 Impostazione della lingua italiana

Sostituire nel file `/etc/profile.d/lang.sh` l'inglese americano con l'italiano:

```
export LANG=it_IT.utf8
```

oppure

```
export LANG=it_IT.ISO8859-1
```

2.3 Sincronizzazione dell'ora

Modificare il file `/etc/ntp.conf`. Per esempio:

```
# Istituto Nazionale di Ricerca Metrologica (I.N.RI.M.)
server ntp1.inrim.it      # server primario 193.204.114.232
server ntp2.inrim.it      # server primario 193.204.114.233
```

2.4 Impostazione del volume sonoro

1. Regolare il volume con *Alsamixer*:

```
# alsamixer
```

o con i programmi grafici alternativi: *KMix* (KDE), *Mixer* (Xfce).

2. Fissare il volume con *Alsactl*:

```
# alsactl store
```

2.5 Impostazione del montaggio dei file system

Per consentire il montaggio dei file system solo quando è necessario, modificare la tabella dei file system contenuta nel file `/etc/fstab`: vedere la **Sottosez. 6.2.1** a pag. 29.

2.6 Configurazione del firewall

In `/etc/rc.d/` inserire il file **rc.firewall** (vedere [3]).

2.7 Configurazione di rete

La connessione ad Internet solitamente avviene:

1. via cavo, mediante interfaccia e router Ethernet;
2. senza fili, mediante interfaccia e router Wi-Fi;
3. mediante modem (chiavetta) USB.

2.7.1 Creazione del proprio dominio

È possibile attribuire un nome di fantasia al proprio computer e al proprio dominio, modificando i file:

- `/etc/HOSTNAME`:

```
macchina.dominio.mio
```

- `/etc/hosts`:

```
127.0.0.1      macchina.dominio.mio macchina
```

2.7.2 Configurazione della connessione via cavo

La configurazione della connessione con l'interfaccia cablata (*wired*), `eth0`, consiste in un paio di modifiche. A titolo di esempio, si assume che il router abbia l'indirizzo IP: 10.0.0.2; di conseguenza, si può attribuire all'interfaccia `eth0` l'indirizzo IP: 10.0.0.3. Si assume anche che il router sia in grado di risolvere gli indirizzi, rendendo superfluo il ricorso a server esterni.

1. Modificare il file `/etc/rc.d/rc.inet1.conf`:

```
# CONFIGURAZIONE DELL'INTERFACCIA CABLATA (WIRED): eth0
IPADDR[0]="10.0.0.3"
NETMASK[0]="255.255.255.0"
GATEWAY="10.0.0.2"
DEBUG_ETH_UP="no"
```

2. Modificare il file `/etc/resolv.conf`:

```
nameserver 10.0.0.2
```

2.7.3 Configurazione della connessione senza fili

Anche la configurazione della connessione con l'interfaccia senza fili (*wireless*), `wlan0`, consiste in un paio di modifiche. A titolo di esempio, si assume che il router abbia l'indirizzo IP: 192.168.0.1; di conseguenza, si può attribuire all'interfaccia `eth0` l'indirizzo IP: 192.168.0.100. Si assume anche che il router sia in grado di risolvere gli indirizzi, rendendo superfluo il ricorso a server esterni.

1. Modificare il file `/etc/rc.d/rc.inet1.conf`:

```
# CONFIGURAZIONE DELL'INTERFACCIA SENZA FILI (WIRELESS): wlan0
IFNAME[0]="wlan0"
IPADDR[0]="192.168.0.100"
NETMASK[0]="255.255.255.0"
USE_DHCP[0]="yes"
DHCP_HOSTNAME[0]="macchina"
HWADDR[0]="indirizzo-MAC"
WLAN_ESSID[0]="ESSID"
WLAN_KEY[0]="s:password"
WLAN_WPA[0]="wpa_supplicant"
WLAN_WPADRIVER[0]="wext"
WLAN_CHANNEL[0]="11"
WLAN_MODE[0]=Managed
```

2. Modificare il file `/etc/wpa_supplicant.conf`:

```
ctrl_interface=/var/run/wpa_supplicant
eapol_version=2
ap_scan=1
fast_reauth=1

network={
    scan_ssid=0
    proto=WPA2
    ssid="SSID"
    #psk="password"
    psk=password-esadecimale
}
```

Per quanto riguarda il blocco `network`:

- le voci sono precedute da uno spazio di tabulazione (tasto `⇧`);
- la password alfanumerica viene inserita commentata;
- per ottenere la password esadecimale (64 caratteri) dalla password alfanumerica:

```
# wpa_passphrase SSID password
```

3. Il file `/etc/resolv.conf` non necessita di modifiche, in quanto all'avvio del sistema viene sovrascritto automaticamente da *Dhcpd* con l'indirizzo IP del router (192.168.0.1).

2.7.4 Configurazione della connessione mediante chiavetta

Per la configurazione della connessione mediante chiavetta USB, vedere [5].

2.7.5 Programmi grafici alternativi

Wicd

Wicd è un programma grafico, incluso tra i pacchetti supplementari nella directory `/extra/` del DVD (o il quarto CD) d'installazione. *Wicd* è compatibile con la configurazione diretta, di cui può essere considerato il completamento.

1. Avviare *Wicd*: si avvia la scansione delle reti senza fili presenti nei dintorni. Individuato il router, cliccare sulle proprietà e inserire la password: la configurazione quasi completa viene automaticamente inserita nel file `/etc/wicd/wireless-settings.conf` (inizialmente vuoto).
2. In qualità di root inserire l'ESSID nel file `wireless-settings.conf`.

NetworkManager

NetworkManager è il programma grafico predefinito. *NetworkManager* è incompatibile con gli altri metodi.

Sakis3G

Sakis3G è uno script prelevabile presso *Salix* (vedere il **Cap. 9** a pag. 54).

2.7.6 Criteri di scelta

1. Per un computer statico (fisso o portatile), che dispone di una rete cablata o senza fili, si può optare tra:
 - a) il metodo diretto, preferibilmente accompagnato da *Wicd*;
 - b) soltanto *Wicd*;
 - c) soltanto *NetworkManager*.
2. Per un computer mobile (portatile), che in situazioni diverse deve connettersi a reti cablate e/o senza fili diverse ed eventualmente anche mediante chiavetta USB, è più pratico l'impiego di *NetworkManager*.
3. Per un computer statico o mobile che può connettersi soltanto mediante chiavetta USB:
 - a) vedere [4];
 - b) *NetworkManager*;
 - c) **Sakis3G**.

2.8 Installazione e configurazione della stampante

Con un programma di navigazione (*browser*) bisogna accedere all'indirizzo `http://localhost:631/`, che corrisponde al server di stampa CUPS.

2.9 Creazione di un utente

Per operare abitualmente come normale utente, creare un utente: vedere la **Sottosez. 5.3.1** a pag. 25.

Capitolo 3

Caratteristiche di Linux

Linux è un sistema operativo:

- multiprocessuale (*multitasking*): più programmi possono essere eseguiti contemporaneamente;
- multiutente: più utenti possono usare il sistema;
- sensibile alla grandezza delle lettere dell'alfabeto (*case sensitive*): fa distinzione tra maiuscole e minuscole.

Linux è un sistema operativo che, nonostante sia dotato di apprezzabili e comode interfacce grafiche, per essere utilizzato appieno non può prescindere dall'interfaccia testuale del terminale.

3.1 La multiprocessualità

La multiprocessualità si basa sul fatto che un processo (programma in esecuzione) non necessita per tutto il tempo della sua esecuzione dell'intervento del processore (CPU). Il processore può perciò dedicarsi e gestire più processi contemporaneamente.

3.2 La multiutenza

La multiutenza è una diretta conseguenza della multiprocessualità: un sistema in grado di gestire più processi (programmi in esecuzione) contemporaneamente è in grado di gestire anche più utenti contemporaneamente.

Si distinguono due categorie di utenti del sistema: A) **root**, e B) gli utenti normali.

Root

Root è il superutente munito di privilegi illimitati, grazie ai quali può accedere a tutto il sistema: ha il compito di amministrare il sistema. A root competono esclusivamente le mansioni delicate: installazione del sistema e amministrazione del sistema.

Gli utenti normali

Gli utenti normali (tizio, caio ecc.) sono gli utenti comuni muniti di diritti limitati, per effetto dei quali possono usare il sistema: svolgono operazioni di carattere ordinario essenzialmente all'interno delle personali directory di lavoro o home directory.

Necessità della multiutenza

La rigorosa distinzione di ruoli (non fa eccezione un computer strettamente personale) è dovuta prima di tutto a ragioni di sicurezza, per evitare che root possa accidentalmente commettere danni irreparabili.

Un esempio limite assai esplicativo è dato dal seguente comando: `rm -r /`. Il comando elimina la directory `/` e tutto il suo contenuto: ossia l'intero file system.

Ma c'è differenza se il comando viene dato da root:

```
# rm -r /
```

o da tizio:

```
$ rm -r /
```

Nel primo caso tutto il file system viene eliminato effettivamente, in quanto root ha poteri illimitati; nel secondo caso al comando segue soltanto un messaggio d'errore, in quanto un utente normale ha privilegi insufficienti per arrecare danno al sistema.

Acquisizione temporanea dei privilegi di root

Da quanto detto, è fondamentale accedere al sistema abitualmente in qualità di utente normale, mentre l'accesso da parte di root deve essere limitato all'amministrazione del sistema. Ciò è possibile sia effettuando la registrazione (*login*) come root sia, preferibilmente, avviando la sessione come utente normale, per poi acquisire temporaneamente, solo quando è necessario, i privilegi di root con *Su*.

Su

Su (da switch user = commutare utente) consente ad un utente normale di accedere alla shell di root per le operazioni di amministrazione del sistema.

```
$ su
Password:
#
```

Il ritorno alla shell di utente normale si ottiene con *exit*, direttamente o attraverso Ctrl-d:

```
# exit
$
```

oppure ancora con *Su* (ma a root non è richiesta la password dell'utente normale):

```
# su tizio
$
```

3.3 La sensibilità alla grandezza delle lettere

Linux distingue in modo predefinito le lettere dell'alfabeto in maiuscole e minuscole (per esempio, in: nomi di directory e file, opzioni di comandi ecc.).

Un esempio è la possibile coesistenza nella stessa directory di più file dallo stesso nome apparente:

```
AB.txt
Ab.txt
aB.txt
ab.txt
```

Quando è necessario ignorare la differenza tra maiuscole e minuscole, bisogna indicarlo esplicitamente (per esempio, nella ricerca di stringhe).

3.4 I modelli

I modelli (*pattern*) comprendono:

- le stringhe letterali,
- le espressioni regolari.

3.4.1 Le stringhe letterali

Una stringa è una sequenza di caratteri normali, dotati esclusivamente del loro valore letterale (visualizzabile).

3.4.2 Le espressioni regolari

Un'espressione regolare (*regular expression*, regexp, regex, RE) è una sequenza di caratteri normali e caratteri speciali (metacaratteri o caratteri jolly). I metacaratteri sono caratteri che hanno un valore simbolico (funzionale), diverso dal loro valore letterale (visualizzabile). In pratica, un metacarattere non rappresenta se stesso, ma un altro carattere o una stringa.

Un'espressione regolare indica un modello (di ricerca o di sostituzione) e produce un valore booleano (sì/no), dipendente dalla sua corrispondenza o non corrispondenza al modello indicato.

Metacaratteri nei modelli di ricerca

<code>\</code>	Effettua il controllo (<i>escape</i>) sul metacarattere che segue: ne disattiva il valore simbolico e ne visualizza il valore letterale.
<code>.</code>	Corrisponde a un qualsiasi carattere singolo, eccetto lo spazio di avanzamento di linea.
<code>*</code>	Corrisponde a un qualsiasi numero (incluso 0) del carattere o dell'espressione regolare che precede.
<code>^</code>	Indica l'inizio di linea o di stringa.
<code>\$</code>	Indica la fine di linea o di stringa.
<code>[]</code>	Indica uno dei caratteri inclusi.
<code>[-]</code>	Indica un intervallo tra caratteri consecutivi.
<code>[^]</code>	Esclude i caratteri inclusi.
<code>\{n\}</code>	Indica il numero (n) di occorrenze del carattere o dell'espressione regolare che precede.
<code>\{n,\}</code>	Indica il numero minimo (n) di occorrenze del carattere o dell'espressione regolare che precede.
<code>\{n,m\}</code>	Indica i numeri minimo (n) e massimo (m) di occorrenze del carattere o dell'espressione regolare che precede.
<code>?</code>	Indica nessuna o una occorrenza dell'espressione regolare che precede.
<code>+</code>	Indica una o più occorrenze dell'espressione regolare che precede.
<code>\(\)</code>	Include e memorizza il testo. Massimo 9 in una linea.
<code>\n</code>	Utilizza il testo n -simo memorizzato. Massimo 9 .

Metacaratteri nei modelli di sostituzione

<code>\</code>	Effettua il controllo (<i>escape</i>) sul metacarattere che segue.
<code>\n</code>	Utilizza il testo memorizzato.
<code>&</code>	Riutilizza il modello di ricerca precedente come parte del modello di sostituzione.

Esempi

<code>^.....\$</code>	Linea di 5 caratteri qualsiasi.
<code>\.\\.\\. \$</code>	Linea terminante con 3 punti.
<code>^stringa</code>	stringa all'inizio di linea o di stringa.
<code>\$stringa</code>	stringa alla fine di linea o di stringa.
<code>[Ss]tringa</code>	Stringa o stringa .
<code>C[aeiou]ro</code>	Caro, Cero, Ciro, Coro o Curo .
<code>[1-5]0</code>	10, 20, 30, 40 o 50 .
<code>10*</code>	1, 10, 100, 1000 ecc.

L'elenco dei metacaratteri validi nei vari programmi cambia. Per esempio:

	<i>Grep</i>	<i>Sed</i>
<code>\</code>	•	•
<code>.</code>	•	•
<code>*</code>	•	•
<code>^</code>	•	•
<code>\$</code>	•	•
<code>[]</code>	•	•
<code>[-]</code>	•	•
<code>[^]</code>	•	•
<code>\{n\}</code>	•	•
<code>\{n,\}</code>	•	•
<code>\{n,m\}</code>	•	•
<code>?</code>	•	
<code>+</code>	•	
<code>\(\)</code>		•
<code>\n</code>		•
<code>\</code>		•
<code>\n</code>		•
<code>&</code>		•

3.5 L'avvio del sistema

1. All'accensione il computer carica dalla memoria secondaria (disco fisso, CD, DVD, chiave USB ecc.) nella memoria primaria (RAM) il kernel del sistema operativo.
2. Il kernel esegue il processo **Init** (il primo processo, con PID 1), che avvia i programmi e i servizi necessari a portare il sistema in uno stato operativo. *Init* (il cui file di configurazione è `/etc/inittab`) può portare il sistema in diversi livelli operativi (*runlevel*): sia impostando il livello iniziale sia portando il sistema da un livello ad un altro.

Il funzionamento di *Init* può essere organizzato in due modi:

- a) secondo il sistema proprio dei sistemi operativi BSD, per cui ad ogni livello corrisponde uno script (`/etc/rc.d/rc.N`, dove N è il numero del livello).
- b) secondo il sistema proprio del sistema operativo Unix System V (five), per cui ad ogni livello corrisponde una directory (`/etc/init.d/rcN.d/`, dove N è il numero del livello).

Slackware adotta il sistema BSD, ma per motivi di compatibilità presenta anche il sistema System V, per l'eventuale installazione di programmi che ammettono solo il sistema System V.

3.6 La documentazione

Nel sistema la documentazione, collocata principalmente in tre subdirectory nella directory `/usr/` (`/doc/`, `/info/` e `/man/`), è accessibile in vario modo:

- accedendo direttamente in `/usr/doc/`, `/usr/info/`, `/usr/man/`;
- accedendo attraverso i programmi specifici **Info** e, soprattutto, **Man**:

```
$ info bash
$ man locate
```

La documentazione molto lunga può essere convertita in documenti (PS e, meglio ancora, PDF) consultabili più agevolmente:

1. `$ man -t bash > bash.ps`
2. `$ ps2pdf bash.ps`

3.7 Terminologia

Programma	File binario eseguibile.
Script	File di testo eseguibile.
Processo	Programma in esecuzione.
Lavoro	Attività, comprendente uno o più processi, originata (attraverso la shell) da un comando eseguito da un utente. Si distingue da un processo di sistema.

Capitolo 4

La shell

I livelli di interazione tra computer e utente si possono rappresentare nel modo seguente:

Utente	Dotato dei cinque sensi.
Programma grafico	Interprete di secondo livello dei comandi tra utente e kernel.
Shell	Interprete di primo livello dei comandi tra utente e kernel.
Kernel	Linux in senso stretto, che comunica direttamente con il computer.
Computer	Dotato di linguaggio macchina.

Figurativamente il *kernel* (= nucleo) e la *shell* (= conchiglia) si possono immaginare rappresentati da una perla racchiusa in una conchiglia, la quale fa da tramite con l'esterno.

Slackware, in quanto insieme organizzato di kernel, shell e programmi grafici, è Linux in senso lato. La shell predefinita di *Slackware* è: **Bash**.

4.1 Le istanze di *Bash*

In base alle modalità e ai file di avvio (= di inizializzazione), sono possibili i seguenti tipi di istanze di *Bash*:

- istanza interattiva di accesso;
- istanza interattiva non di accesso;
- istanza non interattiva.

4.1.1 Istanza interattiva di accesso: la shell di accesso

Questa istanza di *Bash* è visibile al primo avvio dopo l'installazione di *Slackware*: vedere la **Sez. 2.1** a pag. 7. Richiede l'immissione di un comando (interattiva) e delle credenziali di identificazione (accesso o *login*) per l'accesso alla sessione Linux. La shell di accesso si basa:

- sul file generale `/etc/profile`;
- eventualmente, sul file personale `~/.bash_profile` (oppure `~/.bash_login` oppure `~/.profile`);
- eventualmente, per la sua cessazione, sul file personale `~/.bash_logout`.

Per chiudere una shell di accesso:

```
$ logout
```

4.1.2 Istanza interattiva non di accesso: la shell di sessione

Questa istanza di *Bash* è visibile all'apertura di un terminale (ad ogni terminale aperto corrisponde una shell di sessione). Richiede l'immissione di un comando (interattiva) all'interno della sessione Linux. La shell di sessione si basa sul file `~/.bashrc`.

Per chiudere una shell di sessione (cioè, un terminale):

```
$ exit oppure Ctrl-d
```

4.1.3 Istanza non interattiva: la shell di script

Questa istanza di *Bash* è visibile, all'interno di una shell di sessione, durante l'esecuzione di uno script. Non ammette l'immissione di un comando (non interattiva), ma consente l'esecuzione di uno script di shell. La shell di script si basa eventualmente sulla variabile `BASH_ENV`.

Per chiudere una shell di script:

```
$ exit
```

4.1.4 I file temporanei associati

In ogni istanza di *Bash* vengono creati e aperti tre file temporanei predefiniti:

- il file associato alla sorgente di immissione dei dati, corrispondente alla tastiera: lo *standard input* (abbreviato in: *stdin*), cui corrisponde il descrittore di file 0;
- il file associato alla destinazione di visualizzazione dei dati, corrispondente al terminale: lo *standard output* (abbreviato in: *stdout*), cui corrisponde il descrittore di file 1;
- il file associato alla destinazione di visualizzazione dell'eventuale errore, corrispondente anch'essa al terminale: lo *standard error* (abbreviato in: *stderr*), cui corrisponde il descrittore di file 2;

Più schematicamente:

0	Standard input (stdin)	Tastiera
1	Standard output (stdout)	Terminale
2	Standard error (stderr)	Terminale

4.2 I caratteri speciali

I caratteri speciali hanno un valore simbolico (funzionale) diverso dal loro valore letterale (visualizzabile). L'applicazione del loro valore simbolico è detta espansione.

Il carattere di controllo

Un carattere di controllo (o di *escape*) è un carattere non visualizzabile, che non corrisponde ad un simbolo scritto. Il carattere di controllo valido in *Bash* è la barra inversa: « `\` ».

La barra inversa (`\`) precede un carattere speciale: ne impedisce il valore simbolico e ne mostra il valore letterale. Fa eccezione lo spazio di avanzamento di linea (LF o Newline: pag. 44), che dopo la barra inversa indica la continuazione di una linea troppo lunga.

I caratteri di citazione

La citazione (*quoting*) è la delimitazione di una stringa (= sequenza di caratteri): ne impedisce il valore simbolico e ne mostra il valore letterale.

I caratteri di citazione sono gli apici singoli e doppi e la barra inversa: « `'` », « `"` » e « `\` ».

<code>'...'</code>	La stringa delimitata viene presa alla lettera. Fa ovviamente eccezione, conservando il suo valore speciale, il carattere: « <code>'</code> ».
<code>"..."</code>	La stringa delimitata viene presa alla lettera. Fanno eccezione, conservando il loro valore speciale, i caratteri: « <code>"</code> », « <code>\$</code> », « <code>`</code> ».
<code>\</code>	In quanto carattere di controllo, consente la citazione tra doppi apici dei caratteri: « <code>"</code> », « <code>\$</code> », « <code>`</code> ».

Il carattere di esecuzione in secondo piano

Normalmente un comando esegue un processo in primo piano (*foreground*), ma può essere necessario eseguire contemporaneamente un processo in secondo piano (*background*). Ciò è possibile in quanto Linux è un sistema multiprocessuale (vedere la **Sez. 3.1** a pag. 11).

<code>&</code>	Consente l'esecuzione di un comando in secondo piano. Agisce anche come carattere di separazione da un eventuale comando successivo.
--------------------	--

I caratteri di separazione

HT	Separa parole.
LF	Separa parole.
Spazio	Separa parole.
<code>;</code>	Separa comandi da eseguire in sequenza.
<code>&&</code>	Separano comandi da eseguire in sequenza, condizionata al successo del precedente (= AND).
<code> </code>	Separano comandi da eseguire in sequenza, in alternativa al fallimento del precedente (= OR).

I caratteri di raggruppamento

<code>(...)</code>	Raggruppamento di comandi nidificati, da eseguire in una subshell.
<code>{...}</code>	Blocco di comandi, da eseguire nella shell corrente.

I caratteri di reindirizzamento

<code> </code>	Incanala l'output di un comando come input di un altro comando (= Pipe).
<code>></code>	Dirige l'output di un comando in un file, creando il file o sovrascrivendolo.
<code>>></code>	Dirige l'output di un comando in un file, aggiungendolo in coda.
<code><</code>	Prende da un file l'input di un comando.

I caratteri di sostituzione

I caratteri di sostituzione utilizzano il risultato (output) di un comando come argomento del comando precedente.

<code>`...`</code>	Sostituiscono un comando.
<code>\$</code>	Sostituisce un comando o una variabile.

Combinazioni di comandi

<code>comando &</code>	Esegue un comando in secondo piano.
<code>comando1 & comando2</code>	Esegue un comando in secondo piano e un comando in primo piano.
<code>comando file</code>	Esegue un comando che interessa un file.
<code>comando1 ; comando2</code>	Esegue nella shell corrente più comandi in sequenza.
<code>comando1 && comando2</code>	Esegue un comando successivo in caso di successo di un comando precedente.
<code>comando1 comando2</code>	Esegue un comando successivo in caso di fallimento di un comando precedente.
<code>(comando1 ; comando2)</code>	Esegue in una subshell più comandi in sequenza.
<code>{comando1 ; comando2}</code>	Esegue nella shell corrente più comandi in sequenza.
<code>comando1 comando2</code>	Utilizza l'output di un comando precedente come input di un comando successivo.
<code>comando > file</code>	Salva l'output di un comando in un file nuovo. Un file esistente viene sovrascritto.
<code>comando >> file</code>	Aggiunge in coda l'output di un comando in un file esistente.
<code>comando < file</code>	Prende da un file esistente l'input di un comando.
<code>comando1 `comando2`</code>	Utilizza l'output di un comando successivo come input di un comando precedente.
<code>comando1 \$(comando2)</code>	Utilizza l'output di un comando successivo nidificato (in una subshell) come input di un comando precedente.

I caratteri jolly

I caratteri jolly per le directory e i file sostituiscono caratteri alfanumerici (lettere e/o numeri) nei nomi delle directory e dei file.

<code>*</code>	Sostituisce una serie qualsiasi di caratteri.
<code>?</code>	Sostituisce un qualsiasi singolo carattere.
<code>[...]</code>	Sostituiscono uno dei caratteri inclusi o l'intervallo di caratteri tra due estremi separati da <code>-</code> , estremi inclusi.
<code>~</code>	Sostituisce il percorso della home directory.
<code>!</code>	Esclude il carattere successivo.

Il carattere di commento

Negli script una parte del codice sorgente contiene informazioni (commento) utili a chi legge il codice, ma, non dovendo essere eseguita, deve essere ignorata da *Bash*.

Precede ogni linea di commento.

4.3 Le variabili

Le variabili indicano dei dati, che devono essere usati da programmi. Esse sono caratterizzate da un nome (che può essere scritto in lettere maiuscole o minuscole, indifferentemente), a cui viene attribuito un valore, cioè i dati utilizzati dai programmi cui sono rivolti.

Le variabili comprendono:

- variabili globali;
- variabili locali.

4.3.1 Variabili globali o variabili d'ambiente

Sono valide per tutte le shell. Sono inizializzate con la shell di accesso e rese disponibili per tutte le shell di sessione (e per i programmi eseguiti in esse).

Elencazione

Per elencare le variabili globali:

```
$ env (oppure $ printenv)
```

o, preferibilmente,

```
$ env | sort (oppure $ printenv | sort)
```

Creazione

Per creare una variabile globale in modo provvisorio:

```
$ export nome-variabile=valore
```

Per creare una variabile globale in modo permanente, aggiungere nel file `~/.bashrc`:

```
export nome-variabile=valore
```

Ad esempio:

```
export PS1='\u[W]\$ '
export PAGER=most
export SLRNPULL_ROOT=~/.slrnpull
```

4.3.2 Variabili locali o variabili di shell

Sono valide solo per la shell corrente.

Elencazione

Per elencare le variabili locali (comprehensive delle globali):

```
$ set
```

Creazione

Per creare una variabile locale in modo provvisorio:

```
$ set nome-variabile=valore
```


4.3.3 Le variabili dell'invito al comando

Le variabili dell'invito al comando (*prompt*) sono quattro: PS1, PS2, PS3, PS4. Le principali sono le prime due.

L'invito primario

PS1 è l'invito primario, che *Bash* mostra quando è pronta a leggere ed eseguire un comando. Il suo aspetto dipende da una stringa di codici (caratteri speciali preceduti dal carattere di controllo « \ »), spazi e caratteri normali.

Codici	
\h	Inserisce il nome della macchina, presente in <code>/etc/HOSTNAME</code> .
\u	Inserisce il nome dell'utente corrente.
\W	Inserisce la sola directory corrente.
\w	Inserisce il percorso completo della directory corrente.
\D{ }	Inserisce la data e l'ora corrente nel formato <i>strftime</i> (<code>\$ man strftime</code>).
\A	Inserisce l'ora corrente nel formato HH:MM di 24 ore.
\t	Inserisce l'ora corrente nel formato HH:MM:SS di 24 ore.
\!	Inserisce il numero di successione del comando corrente nella cronologia.
\#	Inserisce il numero di successione del comando corrente nella shell corrente.
\\$	Inserisce: <ul style="list-style-type: none"> il simbolo del cancelletto (#) per root (UID = 0); il simbolo del dollaro (\$) per tutti gli altri utenti.

Esempi:

```
PS1='\u[\w]\$ '
PS1='\u@\h[\W]\$ '
PS1='\D{%a %d %b %Y}[\W]\$ '
PS1='\D{%c}[\W]\$ '
PS1='\A|\h|\u[\W] -> '
PS1="All'ora \t esegui: " (apici doppi per la presenza dell'apostrofo)
PS1='Questo è il comando N.\# \$ '
```

L'invito secondario

PS2 è l'invito secondario, che appare quando è necessario un ulteriore input per completare un comando. Il suo aspetto predefinito è dato da:

```
PS2='> '

$ echo parola1 \
> parola2 \
> parola3
```

4.4 Gli alias

Gli alias sono comandi personalizzati, cioè sono sinonimi semplificati che sostituiscono i comandi originali, sia singoli che combinati.

Elencazione

Per elencare gli alias:

```
$ alias
```

Creazione

Per creare un alias in modo provvisorio:

```
$ alias nome-alias='comando'
```

Per eliminare un alias provvisorio:

```
$ unalias nome-alias
```

Per creare un alias in modo permanente, aggiungere nel file `~/.bashrc` (oppure `~/.bash_aliases`):

```
alias nome-alias='comando'
```

Ad esempio:

```
alias lista='ls -Fl --group-directories-first'
alias lpdf='ls *.pdf'
alias ltp='ls *.txt && ls *.pdf'
alias es='env | sort'
```

Per usare i comandi *Cp*, *Mv* e *Rm* sempre con l'opzione « **-i** », che per sicurezza chiede conferma, senza indicarla:

```
alias cp='cp -i'
alias mv='mv -i'
alias rm='rm -i'
```

Un alias appena inserito sarà efficace nelle shell di sessione aperte successivamente. Per renderlo efficace già nella shell corrente:

```
$ . ~/.bashrc (oppure $ source ~/.bashrc)
```

4.5 Il controllo dei lavori

Oltre a controllare l'input e l'output dei comandi, è possibile controllare l'esecuzione dei comandi o, meglio, dei lavori (*job*) di shell.

4.5.1 Esecuzione

Avvio

Per avviare un lavoro:

```
$ comando _Invio
```

Negli esempi la digitazione del tasto `Invio` è solitamente sottintesa.

Interruzione

Per interrompere (annullare) un lavoro:

```
$ comando
...
Ctrl-c
```

Sospensione

Per sospendere (interrompere provvisoriamente) un lavoro:

```
$ comando
...
Ctrl-z
```

4.5.2 Piani di esecuzione

Esecuzione in primo piano (I)

L'esecuzione in primo piano è quella predefinita.

```
$ comando(_Invio)
```

Esecuzione in secondo piano

Per eseguire sin dall'inizio un lavoro in secondo piano:

```
$ comando &(_Invio)
[1] 6005
```

Di ogni lavoro eseguito in secondo piano vengono mostrati due numeri:

- l'identificativo di lavoro dell'utente (1) tra parentesi quadre;
- l'identificativo di processo nel sistema (6005).

Per eseguire un lavoro in secondo piano dopo averlo avviato in primo piano, è necessaria prima la sospensione del lavoro in primo piano:

```
$ comando(_Invio)
...
Ctrl-z
$ bg
```

Per eseguire più lavori in secondo piano dall'inizio:

```
$ comando1 & comando2 &(_Invio)
[1] 7925
[2] 7926
```

Per elencare i lavori eseguiti in secondo piano:

```
$ jobs
```

Per interrompere (annullare) un lavoro in secondo piano (oltre che con il generico [Ctrl-c](#)):

```
$ kill %2 (oppure $ kill 7926)
```

Normalmente, quando un lavoro eseguito in secondo piano giunge al termine durante l'esecuzione di un lavoro in primo piano, al termine del lavoro in primo piano appare la notifica della conclusione del lavoro in secondo piano: il lavoro in primo piano, pertanto, non viene interrotto. Perché la fine di un lavoro in secondo piano venga notificata appena accade, nonostante il lavoro in primo piano sia ancora in esecuzione:

```
$ notify %2
```

Esecuzione in primo piano (II)

Quando in secondo piano c'è un solo lavoro, per riportarlo in primo piano:

```
$ fg
```

Quando in secondo piano c'è più d'un lavoro, bisogna indicare quale riportare in primo piano:

```
$ fg %2
```

Riattivazione

Un lavoro sospeso (con [Ctrl-z](#)) resta in attesa, finché non viene riattivato. La riattivazione può avvenire sia in primo piano (con [\\$ fg](#)) che in secondo piano (con [\\$ bg](#)).

4.6 I comandi interni

Bash è un programma provvisto di programmi interni (o incorporati). Quelli relativi alle variabili, agli alias e ai piani di esecuzione sono stati già descritti.

4.6.1 Informazioni

Type

Alcuni programmi interni sono identici e sostituiscono gli omonimi e corrispondenti programmi Linux. *Type* mostra se un comando è un comando Linux (di cui mostra il percorso) o un comando interno a *Bash*.

```
$ type comando
$ type comando1 comando2
```

4.6.2 Documentazione

Info e *Man* non forniscono documentazione (eccetto che per *Bash* in generale).

Help

Help mostra un elenco dei comandi interni di *Bash* e un testo d'aiuto su ognuno dei comandi interni:

```
$ help
$ help comando
```

4.6.3 Gestione delle directory e dei file

Umask

Umask mostra e crea la maschera dei permessi per la creazione delle directory e dei file (predefinita: 0022). Vedere la **Sottosez. 7.2.3** a pag. 35.

```
$ umask
```

Pwd

Vedere la **Sottosez. 7.6.1** a pag. 39.

Cd

Vedere la **Sottosez. 7.6.2** a pag. 39.

4.6.4 Cronologia dei comandi

La cronologia dei comandi viene registrata nel file `~/.bash_history`, che elenca in via predefinita gli ultimi 500 comandi eseguiti.

History

History mostra l'intera cronologia (500 comandi).

```
$ history
```

Fc

Fc mostra una parte della cronologia.

```
$ fc -l -20 (gli ultimi 20 comandi)
```

Scorrimento manuale della cronologia

Partendo dall'ultimo comando eseguito, i tasti ↑ e ↓ scorrono la cronologia, mostrando, uno alla volta, i comandi elencati nel file `.bash_history`.

4.6.5 Esecuzione di comandi

`.` e *Source*

I programmi `.` (punto) e *Source*, identici, eseguono nella shell corrente i comandi contenuti in un file.

```
$ . file
$ source file
```

Vedere anche a pag. 20.

4.6.6 Ripetizione

Echo

Echo mostra tal quali una o più linee di testo, terminate da uno spazio di avanzamento di linea. *Echo* mostra anche il valore di una o più variabili.

```
$ echo Testo in una linea.
$ echo Testo \
> in più \
> linee.
$ echo $HISTFILE $HISTSIZE
```

Per il secondo esempio, vedere anche a pag. 16 e a pag. 19.

4.6.7 Programmi nulli

I comandi nulli non fanno nulla, ma restituiscono un risultato di successo o di insuccesso.

`:` e *True*

I programmi `:` (due punti) e *True*, identici, non fanno nulla, ma restituiscono un risultato di successo.

```
$ :
$ true
```

False

False non fa nulla, ma restituisce un risultato di insuccesso.

```
$ false
```

Capitolo 5

Gestione dei gruppi e degli utenti

5.1 I gruppi, gli utenti e le password

5.1.1 I gruppi

I gruppi sono elencati nel file `/etc/group`. Ogni voce è divisa in quattro campi:

```
gruppo:password:GID:utenti
```

- La password, di solito assente, è sostituita da: « x ».
- Il GID (Group ID) è il numero identificativo del gruppo.
- Gli utenti possono essere assenti; se presenti più di uno, sono separati da una virgola: « , ».

Esempio minimo di file `group`:

```
root:x:0:root
bin:x:1:root,bin
...
users:x:100:
```

5.1.2 Gli utenti

Gli utenti sono elencati nel file `/etc/passwd`. Ogni voce è divisa in sette campi:

```
utente:password:UID:GID:informazioni-opzionali:home-directory:shell
```

- La password, presente o assente, è sempre sostituita da: « x ».
- L'UID (User ID) è il numero identificativo dell'utente.
- Le informazioni opzionali possono essere assenti; se presenti più di una, sono separate da una virgola: « , ».

Esempio minimo di file `passwd`:

```
root:x:0:0:./root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
...
tizio:x:1000:100:./home/tizio:/bin/bash
```

5.1.3 Le password

Gli utenti sono elencati anche nel file `/etc/shadow`. Ogni voce è divisa in nove campi (attualmente, il nono è inutilizzato):

```
utente:password:I1:I2:I3:I4:I5:I6:
```

- La password, se presente, è criptata e, se assente, è sostituita da un asterisco: « * ».
- I1: intervallo tra il 1 gennaio 1970 e l'ultimo cambio della password.
- I2: intervallo minimo prima di poter cambiare la password (0 = subito).
- I3: intervallo massimo dopo cui si deve cambiare la password, cioè tempo di scadenza della password (99999 = mai).

- I4: intervallo prima della scadenza della password per avvisare l'utente.
- I5: intervallo dopo la scadenza della password per considerare inattivo e disabilitare l'utente.
- I6: intervallo tra il 1 gennaio 1970 e la disabilitazione dell'utente.

Gli intervalli sono espressi in giorni. Esempio minimo di file `shadow`:

```
root:password:15273:0:0:0:
bin:*:9797:0:0:0:
...
tizio:password:15273:0:99999:7:0:
```

5.2 Gestione dei gruppi

5.2.1 Creazione

Groupadd

Groupadd crea un gruppo, attribuendogli il relativo GID:

```
# groupadd -g GID gruppo
```

Esempi

```
# groupadd -g 100 users (predefinito)
# groupadd -g 200 postfix
```

5.2.2 Modifica

Groupmod

Groupmod modifica il nome e/o il GID di un gruppo:

```
# groupmod gruppo nuovo-gruppo
# groupmod -g nuovo-GID gruppo
```

5.2.3 Eliminazione

Groupdel

Groupdel elimina un gruppo:

```
# groupdel gruppo
```

5.3 Gestione degli utenti

5.3.1 Creazione

Adduser

Adduser crea un utente, attribuendogli il relativo UID:

```
# adduser
```

Il programma è semplice da usare. In particolare, con il tasto `↑` evidenziare i gruppi predefiniti necessari a varie attività (tra cui: condivisione dell'audio, uso dell'unità CD/DVD, uso di programmi video):

```
...
Press ENTER to continue without adding any additional groups
Or press the UP arrow key to add/select/edit additional groups
:  audio cdrom floppy plugdev video power netdev lp scanner
```

ed apportare le eventuali modifiche (per esempio, in caso anche di utilizzo di Usenet):

```
:  audio cdrom floppy plugdev video power netdev lp scanner news
```

In alternativa, le appartenenze ai gruppi possono essere attribuite direttamente nel file `/etc/group`, aggiungendo o togliendo il nome dell'utente in corrispondenza dei gruppi considerati tra quelli elencati.

L'uso di *Adduser* è comodo per la creazione di utenti normali comuni.

Useradd

Anche *Useradd* crea un utente, attribuendogli il relativo UID:

```
# useradd -u UID -d home-directory -s shell -g gruppo utente
```

Esempi

```
# useradd -u 100 -d /home/tizio -s /bin/bash -g users tizio
# useradd -u 200 -d /dev/null -s /bin/false -g postfix postfix
```

L'uso di *Useradd* è preferibile per utenti normali particolari.

5.3.2 Modifica

Usermod

Usermod modifica le diverse caratteristiche di un utente:

```
# usermod -u nuovo-UID utente
# usermod utente nuovo-utente
# usermod -G gruppo1,gruppo2 -a utente
```

Passwd

Passwd modifica la password di un utente:

```
# passwd utente
$ passwd
```

5.3.3 Eliminazione

Userdel

Userdel elimina un utente, inclusa la sua home directory con tutto il suo contenuto:

```
# userdel -r utente
```

5.4 Programmi grafici alternativi

- *KUser* (KDE).
- *Usersetup* (prelevabile presso *Salix*) è un programma semigrafico, che si avvia da terminale:

```
# usersetup
```

Gtkusersetup (Xfce) è l'interfaccia grafica.

Capitolo 6

Gestione del file system

6.1 Il file system

Il *file system* è il meccanismo con cui i dati (file e directory) sono immagazzinati e organizzati su un dispositivo di archiviazione (disco fisso o dispositivo rimovibile).

In Linux ogni cosa (sia i dispositivi che i dati) è rappresentata sotto forma di file (*“everything is a file”*).

6.1.1 Gerarchia del file system

In Linux il file system è organizzato in una struttura ad albero rovesciato originante da un'unica directory: */*, detta *root directory* (directory radice, cioè la radice dell'albero). La root directory contiene tutte le directory che costituiscono il file system.

Rappresentazione testuale

La rappresentazione più semplice del file system è quella testuale (su terminale), che qui è limitata al primo livello:

```
$ tree -d -L 1 /
```

```
/
|-- bin
|-- boot
|-- dev
|-- etc
|-- home
|-- lib
|-- lib64
|-- lost+found
|-- media
|-- mnt
|-- opt
|-- proc
|-- root
|-- run
|-- sbin
|-- srv
|-- sys
|-- tmp
|-- usr
`-- var

20 directories
```

Rappresentazione grafica

La rappresentazione grafica del file system si avvale di metafore: i file sono rappresentati come documenti, le directory chiuse sono rappresentate sotto forma di cartelle, le directory aperte sono rappresentate sotto forma di finestre contenenti i documenti e le sottocartelle (chiuse).

Elenco e descrizione delle directory principali

/bin/	Contiene i programmi (file binari) indispensabili eseguibili da tutti gli utenti.
/boot/	Contiene i file necessari al programma (<i>LILO</i> o <i>GRUB</i>) di avvio del sistema.
/dev/	Contiene i file corrispondenti ai dispositivi hardware.
/etc/	Contiene i file di configurazione del sistema e gli script di avvio del sistema. <i>etc</i> = <i>et cetera</i> .
/home/	Contiene le <i>home directory</i> degli utenti normali, ciascuna contenente i file personali di ogni utente.
/lib/	Contiene le librerie condivise di sistema, per i file binari in /bin/ e /sbin/ .
/lib64/	Contiene le librerie condivise di sistema specifiche dei sistemi a 64 bit.
/lost+found/	Contiene le directory e i file rimasti senza collocazione nella gerarchia del file system dopo il ripristino di un file system danneggiato.
/media/	Contiene le directory usate per il montaggio di file system temporanei riferibili ai dischi rimovibili.
/mnt/	Contiene le directory usate per il montaggio di file system temporanei riferibili sia alle partizioni del disco fisso sia ai dischi rimovibili.
/opt/	Contiene i programmi opzionali.
/proc/	Contiene i file virtuali con le informazioni sull'hardware, sulle periferiche e sui processi attivi.
/root/	È la <i>home directory</i> dell'utente root, contenente i file personali di root.
/run/	Contiene i file necessari all'esecuzione dei programmi, i cui dati non possono essere eliminati come in /tmp/ .
/sbin/	Contiene i programmi (file binari di sistema) indispensabili eseguibili esclusivamente da root.
/srv/	Contiene dati specifici forniti dal sistema.
/sys/	Evoluzione di /proc/ , che contiene e modifica le informazioni dei dispositivi connessi al sistema.
/tmp/	Contiene i file temporanei necessari all'avvio del sistema o durante i processi: in parte restano in parte sono eliminati automaticamente.
/usr/	Gerarchia secondaria, che contiene il grosso del sistema: la maggior parte dei programmi e la documentazione. <i>usr</i> = <i>unix system resources</i> .
/var/	Contiene i file variabili, i cui dati sono soggetti a continue modifiche, e i file temporanei della posta elettronica.

Vedere anche [7].

6.1.2 I percorsi

I percorsi sono indirizzi inequivocabili interni al file system.

Percorsi assoluti

I percorsi assoluti sono indipendenti da riferimenti.

```
/
/etc/rc.d
/etc/rc.d/rc.inet1.conf
/home/tizio (abbreviazione: ~)
/home/tizio/bozze.txt (abbreviazione: ~/bozze.txt)
```

Percorsi relativi

I percorsi relativi sono riferiti alla directory corrente.

```
..      Directory immediatamente superiore alla directory corrente.
../..   Directory successivamente superiore alla directory corrente.
./directory Directory nella directory corrente.
./file   File nella directory corrente.
```

6.1.3 I file system supportati

I file system supportati da Linux sono elencati nel file `/proc/filesystems`. Degni di menzione:

ext4	Linux.
ntfs (NT)	Windows.
vfat (FAT32)	Chiavette USB.
iso9660	CD-ROM.
udf	CD-RW, DVD.

6.2 Montaggio dei file system

Come si è detto, nel file system di Linux qualsiasi cosa è rappresentata da un file. Bisogna distinguere:

- il file system principale, cioè la propria partizione root (/);
- i file system secondari (altre partizioni del disco fisso, dispositivi rimovibili ecc.), che di volta in volta è necessario aggiungere, collegare e integrare nel file system principale.

Il montaggio dei file system secondari, pianificato dal file `/etc/fstab` (il file di configurazione di *Mount*), è effettuato da *Mount* in una directory del file system principale: tradizionalmente, ma non necessariamente, si tratta della directory `/mnt/`, che costituisce il punto di montaggio dei file system secondari nel file system principale. La directory `/mnt/` e la subdirectory specifica per il file system secondario da montare (per esempio, `/mnt/sda1/`) devono essere create prima di poter essere montate. Di solito questo avviene durante l'installazione: vedere la **Sottosez. 1.2.5** a pag. 3.

Lo smontaggio dei file system secondari è effettuato da *Umount*.

6.2.1 Configurazione di *Mount*: la tabella dei file system

La tabella dei file system contenuta nel file `/etc/fstab` creata automaticamente durante l'installazione va modificata principalmente per motivi di sicurezza. Per esempio:

<code>/dev/sda5</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>/dev/sda6</code>	<code>/</code>	<code>ext4</code>	<code>defaults</code>	<code>1</code>	<code>1</code>
<code>/dev/sda7</code>	<code>/mnt/sda7</code>	<code>ext4</code>	<code>noauto,users</code>	<code>1</code>	<code>2</code>
<code>/dev/sda8</code>	<code>/mnt/sda8</code>	<code>ext4</code>	<code>noauto,users</code>	<code>1</code>	<code>2</code>
<code>/dev/sda9</code>	<code>/mnt/sda9</code>	<code>ext4</code>	<code>noauto,users</code>	<code>1</code>	<code>2</code>
<code>/dev/sda10</code>	<code>/mnt/sda10</code>	<code>ext4</code>	<code>noauto,users</code>	<code>1</code>	<code>2</code>
<code>/dev/sda11</code>	<code>/mnt/sda11</code>	<code>ext4</code>	<code>noauto,users</code>	<code>1</code>	<code>2</code>
<code>/dev/sda1</code>	<code>/mnt/sda1</code>	<code>ntfs</code>	<code>noauto,users,fmask=333,dmask=222</code>	<code>1</code>	<code>0</code>
<code>/dev/sda12</code>	<code>/mnt/sda12</code>	<code>ntfs-3g</code>	<code>noauto,users,fmask=111,dmask=000</code>	<code>1</code>	<code>0</code>
<code>#/dev/cdrom</code>	<code>/mnt/cdrom</code>	<code>auto</code>	<code>noauto,owner,ro,comment=x-gvfs-show</code>	<code>0</code>	<code>0</code>
<code>#/dev/fd0</code>	<code>/mnt/floppy</code>	<code>auto</code>	<code>noauto,owner</code>	<code>0</code>	<code>0</code>
<code>devpts</code>	<code>/dev/pts</code>	<code>devpts</code>	<code>gid=5,mode=620</code>	<code>0</code>	<code>0</code>
<code>proc</code>	<code>/proc</code>	<code>proc</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>tmpfs</code>	<code>/dev/shm</code>	<code>tmpfs</code>	<code>defaults</code>	<code>0</code>	<code>0</code>

La tabella è divisa in sei colonne:

- dispositivo (partizione del disco fisso, lettore CD/DVD, chiavetta USB ecc.);
- directory;
- tipo di file system;
- opzioni;
- numero indicante se **Dump** deve fare copie di sicurezza (backup) dei file system:
 - 0: disattivato;
 - 1: attivato.
- numero indicante l'ordine in cui **E2fsck** (per i file system Linux estesi ext2, ext3, ext4) deve controllare l'integrità dei file system Linux all'avvio del sistema:
 - 0: nessun controllo;
 - 1: controllo prima (la partizione root: /);
 - 2: controllo dopo (le altre partizioni Linux, in successione).

6.2.2 Mount

Mount monta un file system secondario. Il comando senza opzioni mostra i file system attualmente montati, elencati in `/etc/mstab`:

```
$ mount
```

Trascurabili le opzioni da linea di comando. Essenziali le opzioni in `fstab`, di cui segue la descrizione.

Opzioni in fstab

defaults	Comprende le opzioni predefinite: auto , dev , exec , rw , suid , async , nouser .
atime	Aggiorna atime . Obsoleta.
noatime	Non aggiorna atime : accesso più veloce al file system all'avvio. Il mancato aggiornamento di atime continua anche dopo l'avvio del sistema.
relatime (predefinita)	Aggiorna atime solo se è anteriore a ctime o a mtime .
strictatime	Aggiorna atime . Sostituisce atime . Vedere la Sottosez. 7.2.4 a pag. 35. Vedere anche le opzioni predefinite nel file <code>/proc/mounts</code> .
auto (predefinita)	Monta automaticamente il file system all'avvio.
noauto	Non monta automaticamente all'avvio il file system, che deve essere montato esplicitamente.
dev (predefinita)	Interpreta i dispositivi speciali a blocchi o a caratteri sul file system.
nodev	Non interpreta i dispositivi speciali a blocchi o a caratteri sul file system.
exec (predefinita)	Consente l'esecuzione dei file eseguibili nel file system.
noexec	Non consente l'esecuzione dei file eseguibili nel file system.
ro	Consente di montare il file system solo in lettura.
rw (predefinita)	Consente di montare il file system in lettura e in scrittura.
suid (predefinita)	Rispetta i permessi speciali dei file: SUID, SGID e Sticky.
nosuid	Ignora i permessi speciali dei file: SUID, SGID e e Sticky.
async (predefinita)	Non sincronizza le operazioni di entrata/uscita (I/O: input/output) nel file system.
sync	Sincronizza le operazioni di entrata/uscita (I/O: input/output) nel file system. Può abbreviare la vita dei dispositivi rimovibili a limitato numero di cicli di scrittura (chiavi USB).
owner	Consente ad un utente normale di montare e smontare il file system solo se proprietario del dispositivo. Include in modo predefinito: nodev , nosuid .
user	Consente ad un solo utente normale di montare e smontare il file system: solo l'utente non root che ha montato il file system può anche smontarlo. Include in modo predefinito: nodev , noexec , nosuid .
users	Consente a diversi utenti normali di montare e smontare il file system: il file system può essere montato da un utente e smontato da un altro utente. Include in modo predefinito: nodev , noexec , nosuid .
nouser (predefinita)	Non consente agli utenti normali di montare e smontare il file system: cioè, consente solo a root di montare e smontare il file system.

Con l'inserimento delle opzioni nel file `fstab`, a **Mount** basta l'indicazione del dispositivo o della directory punto di montaggio:

```
# mount dispositivo (oppure directory)
```

```
$ mount dispositivo (oppure directory)
```

Le partizioni Windows

Se si escludono le opzioni **defaults** e **auto**, dopo l'avvio del sistema le partizioni Windows possono essere montate solo da root (l'opzione **users** non ha effetto). Per esempio:

```
# mount /dev/sda1
```

I dispositivi rimovibili

Il montaggio dei dispositivi rimovibili (CD/DVD, chiavette USB ecc.) avviene automaticamente e viene visualizzato graficamente nel pannello laterale di un gestore di finestre (*window manager*).

6.2.3 Umount

Umount smonta un file system secondario.

```
# umount dispositivo (oppure directory)
```

```
$ umount dispositivo (oppure directory)
```

Le partizioni Windows

L'opzione *users* consente lo smontaggio delle partizioni Windows da normale utente. Per esempio:

```
$ umount /dev/sda1
```

I dispositivi rimovibili

Lo smontaggio dei dispositivi rimovibili si effettua sempre graficamente dal pannello laterale di un gestore di finestre.

6.3 Formattazione: la creazione dei file system

La creazione di un file system destinato ad ospitare un sistema operativo è parte integrante del processo di installazione del sistema operativo, per cui non si interviene direttamente.

Per la creazione di un file system destinato ad ospitare dati (in una partizione del disco fisso dedicata o in una chiavetta USB), bisogna avviare direttamente i programmi necessari. La partizione o il dispositivo da formattare non devono essere montati.

6.3.1 Creazione di un file system ext4

Mke2fs

Mke2fs crea tra l'altro il file system ext4:

```
# mke2fs -t ext4 -L nuova-etichetta /dev/sdb1
```

Mkfs.ext4 è un collegamento simbolico a *Mke2fs*:

```
# mkfs.ext4 -L nuova-etichetta /dev/sdb1
```

6.3.2 Creazione di un file system ntfs

Mkntfs

Mkntfs crea il file system ntfs:

```
# mkntfs -L nuova-etichetta /dev/sdb1
```

Mkfs.ntfs è un collegamento simbolico a *Mkntfs*:

```
# mkfs.ntfs -L nuova-etichetta /dev/sdb1
```

6.3.3 Creazione di un file system FAT32

Mkdosfs

Mkdosfs crea tra l'altro il file system FAT32:

```
# mkdosfs -F 32 -n nuova-etichetta /dev/sdb1
```

Mkfs.vfat è un collegamento simbolico a *Mkdosfs*:

```
# mkfs.vfat -n nuova-etichetta /dev/sdb1
```

6.4 Gestione delle partizioni

La situazione ideale è quella di poter partizionare il disco fisso *ex novo*, stabilendo con comodo il numero di partizioni da creare per ospitare i sistemi operativi, le partizioni di paginazione e swap e le partizioni destinate ai dati.

Uno schema minimo (comprendente anche Windows) potrebbe essere, per esempio:

<i>Primaria</i>	<i>Estesa</i>				
	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>
sda1	sda5	sda6	sda7	sda8	sda9
Windows	Swap	Linux	Dati Linux	Dati Windows	Paginazione

Poiché è frequente l'installazione di più distribuzioni Linux, uno schema un po' più allargato (comprendente anche Windows) potrebbe essere, per esempio:

<i>Primaria</i>	<i>Estesa</i>								
	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>	<i>Logica</i>
sda1	sda5	sda6	sda7	sda8	sda9	sda10	sda11	sda12	sda13
Windows	Swap	Linux	Linux	Linux	Linux	Linux	Dati Linux	Dati Windows	Paginazione

L'utilità di installare più sistemi Linux ha più motivi: osservazione comparata di distribuzioni diverse, uso provvisorio della nuova versione della propria distribuzione prima di renderla operativa, copia o modifica dei file di configurazione da una distribuzione all'altra ecc.

Il partizionamento del disco fisso si effettua con *Fdisk* o preferibilmente con *Cfdisk*.

Fdisk si avvia con:

```
# fdisk
```

Cfdisk si avvia con:

```
# cfdisk
```

I due programmi sono quasi equivalenti tra loro: conviene conoscerli entrambi. Entrambi sono semplici da usare, ma, poiché il loro ambito operativo è molto delicato, vanno usati sempre con grande attenzione (vietato distrarsi), ma con maggiore attenzione se ci sono già delle partizioni che ospitano sistemi operativi funzionanti o dati. In questi casi, per sicurezza è consigliabile trascrivere su carta la situazione di partenza prima di procedere con le modifiche: in caso di errori o di ripensamenti sarà possibile tornare indietro e ricominciare da capo.

Esiste anche *Parted*:

```
# parted
```

da prendere in considerazione più che altro per la sua interfaccia grafica.

A volte sono possibili delle leggere sbavature, benché trascurabili, se in precedenza il partizionamento del disco fisso è stato effettuato con programmi per Windows. Quindi, in caso di partizionamenti a più riprese è preferibile utilizzare lo stesso programma.

Le partizioni presenti sul disco fisso sono elencate nel file `/proc/partitions`.

6.5 Programmi grafici alternativi

- *Midnight Commander* è un programma semigrafico, che si avvia da terminale:

```
$ mc
```

- *Dolphin* (KDE), *Konqueror* (KDE), *Thunar* (Xfce), *PCManFM*, *SpaceFM*.
- *KDiskFree* (KDE), *KwikDisk* (KDE), *Plugin di montaggio* (Xfce).
- *GParted*, *KDE Partition Manager*.

Capitolo 7

Gestione delle directory e dei file

7.1 I file

Tipi di file

file (normale)	-	contiene dati.
dispositivo a blocchi	b	periferica ad accesso casuale (disco fisso, lettore CD/DVD).
dispositivo a caratteri	c	periferica ad accesso sequenziale (terminale).
directory	d	contiene un elenco di nomi: subdirectory e file.
collegamento simbolico	l	si riferisce ad una directory o ad un file.
pipe	p	comunicazione unidirezionale tra processi.
socket	s	comunicazione bidirezionale tra processi.
porta	D	comunicazione bidirezionale tra client e server.

Qui interessa la distinzione tra: directory e file. Le directory e i file possiedono: a) caratteristiche evidenti, che li qualificano; e b) informazioni di servizio, dette attributi. Le caratteristiche e gli attributi si assegnano alla creazione ed eventualmente alla modifica delle directory e dei file.

Caratteristiche dei file

Le caratteristiche evidenti sono:

- per le directory, che sono soltanto degli elenchi: il nome;
- per i file: il nome, il contenuto e il tipo (indicato dall'estensione: txt, html, pdf ecc.).

Attributi dei file

Gli attributi sono delle informazioni specifiche, che tra l'altro comprendono:

- i permessi;
- le proprietà: l'utente proprietario e il gruppo proprietario;
- la marcatura temporale: la data e l'ora.

7.2 Gli attributi dei file

7.2.1 I permessi

I permessi comprendono:

1. i permessi ordinari di accesso alle directory e ai file (in lettura, scrittura ed esecuzione), che possono essere indicati con lettere o numeri:

read	r	4
write	w	2
execute	x	1

2. i permessi speciali di accesso alle directory e ai file eseguibili (SUID = Set UID, SGID = Set GID, Sticky), che si aggiungono e modificano i permessi ordinari in esecuzione e che possono essere indicati con lettere o numeri:

SUID	s	4
SGID	s	2
Sticky	t	1

- SUID interessa i file eseguibili e le directory: ne modifica i permessi in esecuzione del proprietario. Un esempio è il programma `/usr/bin/passwd` (`rws--x--x/4711`), di proprietà di root: può essere utilizzato pienamente da root (per modificare la password di tutti gli utenti) e limitatamente da qualsiasi utente normale (per modificare solo la propria password, dopo aver fornito la password corrente).
- SGID interessa i file eseguibili e le directory: ne modifica i permessi in esecuzione del gruppo. Un esempio è il programma `/usr/bin/slocate` (`-rwxr-sr-x/2755`), di proprietà di root e del gruppo `slocate`.
- Sticky interessa le directory: si riferisce a tutti gli utenti diversi dal proprietario della directory. Un esempio è `/tmp/` (`rw-rwxrwt/1777`), di proprietà di root: ad eccezione di root, tutti gli utenti possono accedere alla directory per leggere, creare e modificare soltanto i propri file temporanei.

Esempi di directory con i permessi SUID, SGID e Sticky sono le directory corrispondenti alle partizioni contenenti altri sistemi Linux, come nella seconda tabella della **Sez. 6.4** a pag. 32. Assumendo *Slackware* in `sda6` = `/`: `/mnt/sda7/` - `/mnt/sda8/` - `/mnt/sda9/` - `/mnt/sda10/` (`rwsrwsrwt/7777`).

7.2.2 Le categorie di utenti

I permessi vengono attribuiti alle varie categorie di utenti (utente proprietario, gruppo proprietario, tutti gli altri, tutti), che possono essere indicate con lettere:

```
user      u
group     g
others    o
all       a   = ugo (predefinito, si omette)
```

o dall'ordine dei numeri indicanti i permessi.

Le due notazioni (rappresentazioni) in lettere e in numeri sono tra loro equivalenti, ma in pratica si utilizzano diversamente: vedere la **Sottosez. 7.3.1**.

L'utente che crea una directory o un file è automaticamente il proprietario di quella directory o di quel file.

7.2.3 I permessi predefiniti

La dotazione iniziale dei permessi ordinari disponibili per creare una directory o un file è:

```
directory  rwxrwxrwx  777
file       rw-rw-rw-  666
```

Come si vede, la dotazione iniziale è differente, in quanto i file sono creati senza i permessi in esecuzione. In realtà, in base alla configurazione di **Umask** (programma interno alla shell *BASH*), presente in `/etc/profile`:

```
...
# Default umask.  A umask of 022 prevents new files from being created group
# and world writable.
umask 022
...
```

che in modo predefinito nega i permessi in scrittura per il gruppo proprietario e per gli altri, le nuove directory e i nuovi file sono creati con i seguenti permessi ordinari predefiniti:

```
directory  rwxr-xr-x  755  (= 777 - 022)
file       rw-r--r--  644  (= 666 - 022)
```

7.2.4 La marcatura temporale

La marcatura temporale delle directory e dei file è l'attribuzione del tempo (data e ora) relativo allo stato delle directory e dei file e comprende:

```
atime  Tempo dell'ultimo accesso. Non si aggiorna mai se nel file fstab è presente l'opzione noatime.
ctime  Tempo dell'ultimo cambiamento di stato.
mtime  Tempo dell'ultima modifica.
```

Directory

Nelle directory (semplici elenchi di directory e file):

`atime` si aggiorna in caso di: apertura, creazione/eliminazione di subdirectory e/o file all'interno.

`ctime` si aggiorna in caso di: creazione/eliminazione di subdirectory e/o file all'interno.

`mtime` si aggiorna in caso di: creazione/eliminazione di subdirectory e/o file all'interno.

File

Nei file:

`atime` si aggiorna in caso di: lettura, scrittura.

`ctime` si aggiorna in caso di: scrittura, cambiamento dei permessi.

`mtime` si aggiorna in caso di: scrittura.

7.3 Gestione degli attributi

7.3.1 Gestione dei permessi

Chmod

Chmod attribuisce e modifica i permessi di una directory o di un file. Come già detto, anche se equivalenti tra loro, la notazione in lettere e la notazione in numeri si usano diversamente.

La notazione in lettere viene essenzialmente usata in modo relativo, per aggiungere o togliere alcuni permessi.

```
$ chmod u+x file
$ chmod o-w file
$ chmod u+s file
$ chmod +t directory
```

La notazione in numeri è usata normalmente in modo assoluto: i permessi sono indicati dalla successione di tre numeri (uno per ogni categoria di utenti). I permessi speciali, quando presenti, sono indicati dal primo di quattro numeri (gli altri tre corrispondono, come di norma, alle categorie di utenti). Invece, la notazione in lettere usata in modo assoluto è poco pratica: basta confrontare gli stessi esempi espressi nei due modi.

\$ chmod 644 directory-vuota	\$ chmod u=rw,g=r,o=r directory-vuota
\$ chmod -R 644 directory-piena	\$ chmod -R u=rw,g=r,o=r directory-piena
\$ chmod 640 file	\$ chmod u=rw,g=r file
\$ chmod 640 file1 file2	\$ chmod u=rw,g=r file1 file2
\$ chmod 4755 file	\$ chmod u=rws,g=rx,o=rx file
\$ chmod 1755 file	\$ chmod u=rwt,g=rx,o=rx file

L'ordine degli utenti e dei permessi è importante nella notazione in numeri, è indifferente nella notazione in lettere.

7.3.2 Gestione delle proprietà

Chown

Chown attribuisce e modifica le proprietà di una directory o di un file, ossia l'utente proprietario e/o il gruppo proprietario.

```
# chown utente:gruppo directory-vuota
# chown -R utente:gruppo directory-piena
# chown utente:gruppo file
# chown UID:GID file
# chown utente: file
# chown :gruppo file
```

7.3.3 Gestione della marcatura temporale

Touch

Touch aggiorna la marcatura temporale di una o più directory o di uno o più file, già esistenti, alla data e all'ora correnti, senza che i file siano stati modificati.

Opzioni

-a	Aggiorna atime e ctime .
-m	Aggiorna mtime e ctime .

Il comando senza opzioni aggiorna **atime**, **mtime** e **ctime**.

```
$ touch directory
$ touch directory1 directory2
$ touch file
$ touch file1 file2
```

7.4 Elencazione

7.4.1 *Ls*

Ls elenca le directory (subdirectory) e i file contenuti in una directory. La directory predefinita (senza indicazione) è la directory corrente. Elenca anche singoli file.

La sintassi generale è:

```
ls opzione/i directory/file/modello
```

Opzioni

-a, --all	Elenca tutto: la directory corrente (.), la directory superiore (..), le (sub)directory nascoste e visibili e i file nascosti e visibili.
-A, --almost-all	Elenca quasi tutto: le (sub)directory nascoste e visibili e i file nascosti e visibili.
-d, --directory	Elenca la directory senza il contenuto.
-F, --classify	Aggiunge un simbolo alle voci elencate (* ai file eseguibili, / alle directory, @ ai collegamenti simbolici, alle pipe, = ai socket, > alle porte).
--group-directories-first	Elenca prima le (sub)directory.
-lc	Mostra un elenco dettagliato per ctime , in ordine alfabetico.
-ltc	Mostra un elenco dettagliato per ctime , in ordine cronologico da più recente a più remoto.
-l	Mostra un elenco dettagliato per mtime , in ordine alfabetico.
-lt	Mostra un elenco dettagliato per mtime , in ordine cronologico da più recente a più remoto.
-lu	Mostra un elenco dettagliato per atime , in ordine alfabetico.
-ltu	Mostra un elenco dettagliato per atime , in ordine cronologico da più recente a più remoto.
-r, --reverse	Inverte l'ordine dell'elenco.
-R, --recursive	Elenca le (sub)directory in modo ricorsivo.
-S	Mostra un elenco per dimensioni, in ordine decrescente.
-X	Mostra un elenco per estensione, in ordine alfabetico.

```
$ ls -F --group-directories-first
$ ls -l --group-directories-first /etc
$ ls -d /etc
```

```
$ ls -dl /etc
$ ls -dlR --group-directories-first /etc/*
$ ls -dlR --group-directories-first /etc/*/*
$ ls -l bozze.txt
$ ls -l /etc/fstab
$ ls *.pdf
```

Il comando per gli elenchi dettagliati (`$ ls -l ...`) produce il seguente risultato tipo:

```
totale 37444
...
drwxr-xr-x  8 tizio users    4096 giu 15 03:32 Archivio
...
```

Dopo la prima riga, che informa sulle dimensioni totali del contenuto della directory, ogni voce è divisa in sette colonne:

1. colonna suddivisa in 10 sottocolonne: tipo di file (**Sez. 7.1**) e permessi (**Sottosez. 7.2.1**);
2. numero di collegamenti fisici;
3. utente proprietario;
4. gruppo proprietario;
5. dimensioni;
6. mtime suddiviso in data e ora (per gli anni precedenti al corrente c'è solo la data);
7. nome.

Il numero di collegamenti fisici:

- per le directory è minimo 2 (*a.* la directory stessa con il proprio nome, e *b.* il collegamento interno ad ogni directory: `.`) più un collegamento alla directory superiore in ogni subdirectory eventualmente presente: `..`;
- per i file è minimo 1 (il file stesso con il proprio nome).

Inoltre, il comando:

```
$ ls -l /usr/bin
```

produce tra l'altro:

```
...
lrwxrwxrwx  1 root  root           7 mar 19 18:32 locate -> slocate
...
-rwxr-sr-x  1 root  slocate    36288 ago 30  2012 slocate
...
```

da cui si vede che:

- il file `locate` è un collegamento simbolico (1) a `slocate` (`-> slocate`);
- il file `slocate` ha un solo collegamento (fisico: il collegamento simbolico `locate` non è considerato).

7.5 Ricerca

La ricerca dei file si basa sui modelli (vedere la **Sez. 3.4** a pag. 12) o sugli attributi dei file.

7.5.1 Slocate

Slocate cerca i file in base a modelli (intesi come stringhe letterali) all'interno del file system non direttamente, ma attraverso il file database `/var/lib/slocate/slocate.db`. **Locate** è un collegamento simbolico a **Slocate**.

```
$ locate modello
```

Quando è troppo lungo, l'elenco ottenuto può essere stampato in un file di testo:

```
$ locate modello > modello.txt
```

L'aggiornamento del database viene effettuato da root con **Updatedb**:

```
# updatedb
```

7.5.2 Find

Find cerca i file in una gerarchia di directory, cioè ricorsivamente a partire dalla directory di partenza, in base a modelli o attributi.

Opzioni

-maxdepth <i>n</i>	Imposta il numero massimo di livelli gerarchici da non superare nella ricerca.
-mindepth <i>n</i>	Imposta il numero minimo di livelli gerarchici da cui avviare la ricerca.
-print	Mostra le directory e i file trovati, completi di percorso.
-name <i>modello</i>	Cerca i file i cui nomi contengano dei modelli (in stile <i>Bash</i>). I metacaratteri vanno preceduti dal carattere di controllo (\) o citati tra apici (' ').
-regex <i>modello</i>	Cerca i file i cui nomi contengano dei modelli (in stile <i>Grep</i>).
-user <i>utente/UID</i>	Cerca i file di un particolare utente o UID.
-uid <i>UID</i>	Cerca i file di un particolare UID.
-group <i>gruppo/GID</i>	Cerca i file di un particolare gruppo o GID.
-gid <i>GID</i>	Cerca i file di un particolare GID.
-type <i>tipo</i>	Cerca i file del tipo specificato (Sez. 7.1 a pag. 34): <ul style="list-style-type: none"> • f: file normale, • b: dispositivo a blocchi, • c: dispositivo a caratteri, • d: directory, • l: collegamento simbolico, • p: pipe, • s: socket, • D: porta.
-perm <i>permessi</i>	Cerca i file con tutti i permessi specificati.
/permessi	Cerca i file con alcuni permessi specificati.
-exec <i>programma {} \;</i>	Esegue un programma nei file trovati.
-ok <i>programma {} \;</i>	Esegue un programma nei file trovati, dopo conferma.
!	Nega le opzioni successive.

```
$ find directory
$ find directory -print
$ find directory -name \*.pdf (oppure '*.pdf')
$ find directory -regex [Ff]ile
$ find /bin -type l
$ find directory -perm /111 (cioè, le directory e i file eseguibili)
$ find directory ! -perm /111 (cioè, le directory e i file non eseguibili)
$ find directory -print -exec programma {} \;
```

7.6 Navigazione

7.6.1 Pwd

Pwd mostra la directory corrente. **Pwd** è un programma interno alla shell **BASH**.

```
$ pwd
```

7.6.2 Cd

Cd cambia la directory corrente. La directory predefinita è la propria home directory. **Cd** è un programma interno alla shell **BASH**.

```
$ cd
$ cd directory
```

Gli esempi seguenti includono percorsi assoluti e relativi generici.

<code>\$ cd ..</code>	Alla directory superiore.
<code>\$ cd ../../</code>	Alla directory ancora superiore.
<code>\$ cd ../subdirectory</code>	Alla subdirectory nella directory superiore.
<code>\$ cd</code>	Ritorno alla propria home directory.
<code>\$ cd ~</code>	Ritorno alla propria home directory.
<code>\$ cd /home/utente</code>	Ritorno alla propria home directory.
<code>\$ cd ~/subdirectory</code>	Alla subdirectory nella propria home directory.
<code>\$ cd /home/utente/subdirectory</code>	Alla subdirectory nella propria home directory.
<code>\$ cd subdirectory</code>	Alla subdirectory nella directory corrente.
<code>\$ cd ./subdirectory</code>	Alla subdirectory nella directory corrente.
<code>\$ cd subdirectory/subdirectory</code>	Alla subdirectory di secondo livello nella directory corrente.
<code>\$ cd ./subdirectory/subdirectory</code>	Alla subdirectory di secondo livello nella directory corrente.

7.7 Visualizzazione e modifica

Per la visualizzazione e modifica dei file di testo (intesi come: file di testo semplice, script ecc.), vedere [4].

7.8 Creazione

7.8.1 *Mkdir*

Mkdir crea una o più directory vuote.

```
$ mkdir directory
$ mkdir directory1 directory2
```

7.8.2 *Touch*

Touch crea uno o più file vuoti.

```
$ touch file
$ touch file1 file2
```

7.8.3 *Ln*

Ln crea uno o più collegamenti. I collegamenti possono essere: a) fisici, e b) simbolici.

Collegamenti fisici

Un collegamento fisico ad un file corrisponde al file di origine con un nome alternativo. I dati contenuti nel file di origine non vengono duplicati, ma vi si può accedere aprendo sia il file di origine sia il collegamento.

```
$ ln file collegamento
$ ln file collegamento1 collegamento2
```

L'eliminazione del file di origine lascia intatto il collegamento: i dati contenuti nel file di origine restano accessibili aprendo il collegamento.

Collegamenti simbolici

Un collegamento simbolico ad una directory o ad un file corrisponde ad un percorso alternativo alla directory o al file di origine. I dati contenuti nella directory o nel file di origine non vengono duplicati, ma vi si può accedere aprendo sia la directory o il file di origine sia il collegamento.

```
$ ln -s directory collegamento
$ ln -s directory collegamento1 collegamento2
$ ln -s file collegamento
$ ln -s file collegamento1 collegamento2
```

L'eliminazione della directory o del file di origine rende inutilizzabile il collegamento: i dati contenuti nella directory o nel file di origine non sono più accessibili aprendo il collegamento.

7.9 Rinomina

7.9.1 *Mv*

Mv rinomina una directory o un file.

```
$ mv nome-directory nuovo-nome-directory
$ mv nome-file nuovo-nome-file
```

7.10 Spostamento

7.10.1 *Mv*

Mv sposta una o più directory o uno o più file in un'altra directory.

```
$ mv directory /tmp
$ mv directory1 directory1 /tmp
$ mv file /tmp
$ mv file1 file2 /tmp
```

L'eventuale aggiunta dell'opzione `-i` mostra una richiesta di conferma (s/n) in caso di sovrascrittura.

7.11 Copia

7.11.1 *Cp*

Cp copia una o più directory o uno o più file.

Nella stessa directory:

```
$ cp nome-directory-vuota nuovo-nome-directory-vuota
$ cp -r nome-directory-piena nuovo-nome-directory-piena
$ cp nome-file nuovo-nome-file
```

In un'altra directory:

```
$ cp directory-vuota /tmp
$ cp directory-vuota1 directory-vuota2 /tmp
$ cp -r directory-piena /tmp
$ cp -r directory-piena1 directory-piena2 /tmp
$ cp file /tmp
$ cp file1 file2 /tmp
```

L'eventuale aggiunta dell'opzione `-i` mostra una richiesta di conferma (s/n) in caso di sovrascrittura.

7.12 Eliminazione

7.12.1 *Rm*

Rm elimina una o più directory o uno o più file.

```
$ rm directory-vuota
$ rm directory-vuota1 directory-vuota2
$ rm -r directory-piena
$ rm -r directory-piena1 directory-piena2
$ rm file
$ rm file1 file2
```

L'eventuale aggiunta dell'opzione *-i* mostra una richiesta di conferma (s/n) prima di procedere.

7.13 Gestione di archivi, Compressione e Decompressione

Sia *file* che *archive* si traducono con: archivio. Entrambi sono file, ma in base al contenuto si distinguono:

- il file, in quanto raccolta di dati;
- l'archivio, in quanto raccolta di directory e file.

Tar converte directory o singoli file in archivi. Sia gli archivi che i file (le directory vanno prima convertite in archivi) possono essere compressi, per risparmiare spazio. Programmi di compressione via via più potenti sono: *Gzip*, *Bzip2* e *Lzma*.

7.13.1 *Tar*

Tar gestisce archivi di directory o di singoli file.

Crea archivi (raccolge directory o singoli file in archivi) e archivi compressi con *Gzip* o con *Bzip2*:

```
$ tar cvf archivio.tar directory
$ tar czvf archivio.tar.gz directory
$ tar cjvf archivio.tar.bz2 directory
$ tar cvf archivio.tar file
$ tar czvf archivio.tar.gz file
$ tar czvf archivio.tar.bz2 file
```

L'eventuale aggiunta dell'opzione *k* salva (non elimina) la directory o il file di origine.

Estrae il contenuto di archivi (libera directory o singoli file dagli archivi) e archivi compressi con *Gzip* o con *Bzip2*:

```
$ tar xvf archivio.tar
$ tar xzvf archivio.tar.gz
$ tar xjvf archivio.tar.bz2
```

L'eventuale aggiunta dell'opzione *k* salva (non elimina) l'archivio (compresso o meno).

7.13.2 *Gzip* e *Gunzip*

Gzip comprime archivi e file. L'archivio o il file di origine viene eliminato.

```
$ gzip -v9 archivio.tar
$ gzip -v9 file
```

Gunzip decompime archivi e file compressi con *Gzip*. L'archivio o il file compresso viene eliminato.

```
$ gunzip -v archivio.tar.gz
$ gunzip -v file.gz
```


7.13.3 *Bzip2* e *Bunzip2*

Bzip2 comprime archivi e file.

```
$ bzip2 -v archivio.tar
$ bzip2 -v file
```

L'eventuale aggiunta dell'opzione **-k** salva (non elimina) l'archivio o il file di origine.
Bunzip2 decompime archivi e file compressi con *Bzip2*.

```
$ bunzip2 -v archivio.tar.bz2
$ bunzip2 -v file.bz2
```

L'eventuale aggiunta dell'opzione **-k** salva (non elimina) l'archivio o il file compresso.

7.13.4 *Lzma* e *Unlzma*

Lzma comprime archivi e file.

```
$ lzma -v9 archivio.tar
$ lzma -v9 file
```

L'eventuale aggiunta dell'opzione **-k** salva (non elimina) l'archivio o il file di origine.
Unlzma decompime archivi e file compressi con *Lzma*.

```
$ unlzma -kv archivio.tar.lzma
$ unlzma -kv file.lzma
```

L'eventuale aggiunta dell'opzione **-k** salva (non elimina) l'archivio o il file compresso.

7.14 Programmi grafici alternativi

- *Midnight Commander*:

```
$ mc
```

- *Dolphin* (KDE), *Konqueror* (KDE), *Thunar* (Xfce), *PCManFM*, *SpaceFM*.
- *Ark* (KDE), *Engrampa/File Roller*.

Capitolo 8

Gestione dei testi

Oltre ai programmi di videoscrittura interattivi, come *Vim* (vedere [4]), in cui la modifica dei testi avviene attraverso la tastiera, esistono numerosi programmi non interattivi deputati alla gestione dei file di testo, che mostrano il risultato (output) su schermo (standard output), senza modificare il file di input. Questi programmi non interattivi sono anche in grado di agire su flussi (*stream*) di dati, cioè in modo sequenziale, e per questo sono definiti filtri (vedere la **Sez. 8.9** a pag. 52).

In sintesi, questi programmi possono agire:

- su un file di input, se il nome del file è indicato esplicitamente;
- sullo standard input, se il nome del file non è indicato o è sostituito dall'opzione: « - ».

Spazi

La tabella seguente elenca alcuni tipi di spazi presenti nei testi.

	Tasto	Codice*
Spazio di tabulazione (orizzontale), Horizontal Tab (HT)	⇐⇒	\t
Spazio di avanzamento di linea (o riga), Line feed (LF), Newline	Invio	\n
Spazio (semplice)	Spazio	

* Codice di controllo (*escape*) in linguaggio C.

8.1 Ricerca

La ricerca all'interno dei file si basa sui modelli (vedere la **Sez. 3.4** a pag. 12).

8.1.1 *Grep*

Grep ricerca in uno o più file le linee sede di corrispondenze di modelli: stringhe letterali ed espressioni regolari.

Opzioni

<code>-c, --count</code>	Mostrano il numero di linee sede di corrispondenze.
<code>-e modello, --regexp=modello</code>	Precedono un modello. Facoltative (si omettono) in caso di un singolo modello, sono obbligatorie in caso di più modelli e se il modello inizia con: « - ».
<code>-f, --file</code>	Accettano un file contenente un elenco di modelli (uno per linea).
<code>-i, --ignore-case</code>	Ignorano la grandezza delle lettere (la differenza tra maiuscole e minuscole).
<code>-n, --line-number</code>	Mostrano le linee sede di corrispondenze e il relativo numero di linea.
<code>-r, --recursive</code>	Mostrano, in una directory esplorata ricorsivamente (directory e subdirectory), le linee sede di corrispondenze e i file che le contengono.
<code>-v, --invert-match</code>	Mostrano le linee che non sono sede di corrispondenze.

```
$ grep modello file
$ grep -c modello file
$ grep -n modello file
$ grep modello file1 file2
$ grep -r modello directory
```

8.2 Visualizzazione

La visualizzazione di un file di testo ne consente principalmente la lettura.

8.2.1 *Cat, Tac, Rev*

Cat

Cat mostra un testo (ma anche un file binario), partendo dall'inizio.

```
$ cat file
```

Tac

Tac mostra un testo (ma anche un file binario), partendo dalla fine.

```
$ tac file
```

Rev

Rev mostra un testo invertito.

```
$ rev file
```

8.2.2 *Sed*

Sed mostra un testo (ma anche un file binario).

```
$ sed ' ' file
```

8.2.3 *More, Less*

More

More mostra un testo come una successione di schermate: non consente lo scorrimento.

Comandi di spostamento	
Spazio	Alla schermata successiva.
b	Alla schermata precedente.

```
$ more file
```

Less

Less mostra un testo come una successione di schermate, ma consente lo scorrimento. *Less* è usato da *Man*.

```
$ less file
```

8.2.4 *Head, Tail*

Head

Head mostra la parte iniziale di un testo (ma anche di un file binario), in via predefinita le prime 10 linee.

Opzioni	
-n, -n n, --lines n	Impostano le prime n linee da mostrare.

```
$ head file
```

```
$ head -5 file (oppure: -n 5 file)
```

Tail

Tail mostra la parte finale di un testo (ma anche di un file binario), in via predefinita le ultime 10 linee. Usato per la visualizzazione dei file di registro (*log*) presenti in `/var/log/`, che sono continuamente aggiornati (nella loro parte finale).

Opzioni

`-n, -n n, --lines n` Impostano le ultime *n* linee da mostrare.

```
$ tail file
$ tail -20 file (oppure: -n 20 file)
```

8.3 Modifica

8.3.1 Uniq, Cut

Uniq

Mostra le linee uniche e/o le linee consecutive ripetute.

Opzioni

<code>-c, --count</code>	Mostrano sia le linee uniche sia una volta sola le linee ripetute, tutte precedute dal conteggio delle loro occorrenze.
<code>-d, --repeated</code>	Mostrano una volta sola le linee ripetute, ma non le linee uniche.
<code>-u, --unique</code>	Mostrano le linee uniche, ma non le linee ripetute.

Senza opzioni, *Uniq* mostra sia le linee uniche sia una volta sola le linee ripetute.

```
$ uniq file
$ uniq -c file
```

Cut

Cut mostra soltanto (taglia) alcune colonne di un file:

Opzioni

<code>-ccolonne, --characterscolonne</code>	Mostrano le colonne indicate.
<code>-ddelimit., --delimiterdelimiter.</code>	Impostano un delimitatore: in via predefinita, il carattere di tabulazione. Lo spazio semplice va racchiuso tra apici (' '). Usate insieme a <code>-f</code> .
<code>-fcampi, --fieldscampi</code>	Mostrano i campi indicati. Usate insieme a <code>-d</code> .

Le colonne e i campi sono indicati dai numeri corrispondenti. Gli eventuali separatori sono rappresentati da virgole e/o trattini:

`,` = Colonne/campi distinti.
`-` = Intervallo di colonne/campi.

```
$ cut -c3 file
$ cut -c2,5 file
$ cut -c2-5 file
$ cut -c-5 file (oppure: 1-5 file)
$ cut -c5- file (cioè: dalla quinta all'ultima)
$ cut -d: -f1,3 file
$ cut -d' ' -f1,3 file
```

8.3.2 Sed

Sed è un programma di videoscrittura non interattivo (modifica i file in modo automatico), che non modifica direttamente il file di input, ma crea e modifica una sua copia in uno spazio di memorizzazione. Il nome deriva dal fatto che, come filtro, può agire su flussi (*stream*) di dati: Sed = Stream EDitor.

Opzioni

-e, --expression	Precedono un comando. Facoltative (si omettono) in caso di un singolo comando, sono obbligatorie in caso di più comandi.
-f, --file	Indicano uno script contenente comandi.
-i.suffisso, --in-place=.suffisso	Senza suffisso, modificano il file. Con suffisso (come .bak), creano anche una copia di sicurezza del file immutato.
-n, --quiet, --silent	Non mostrano l'output predefinito, evitando raddoppi di linee.

Comandi

=	Mostra i numeri di linea.
p	Mostra le linee.
a	Aggiunge del testo dopo una linea.
c	Sostituisce del testo.
i	Inserisce del testo prima di una linea.
d	Cancella delle linee.
s	Sostituisce del testo /modello/ con del testo /sostituto/ . Comprende dei comandi aggiuntivi: n : numero che indica l' <i>n</i> -esima occorrenza del /modello/ di ogni linea indicata dagli indirizzi; se assente, indica la prima occorrenza, predefinita; g : interessa tutte le occorrenze del /modello/ di ogni linea indicata dagli indirizzi; p : mostra le linee sede di sostituzioni; w file : scrive in un file le linee sede di sostituzioni.

I comandi possono essere uno o più, indicati direttamente tra apici singoli o indirettamente all'interno di uno script. Pertanto, la sintassi semplificata della linea di comando prevede due forme possibili:

```
sed -e 'comando' file
sed -f script file
```

Se il comando manca, viene mostrato il file (vedere anche la **Sottosez. 8.2.2** a pag. 45).

```
$ sed ' ' file
```

I comandi possono essere preceduti da indirizzi, che a loro volta possono essere di due tipi:

Indirizzi

n	Indica un numero di linea.
/modello/	Indica un modello: stringa letterale o espressione regolare.

Il numero degli indirizzi può variare:

Nessun indirizzo: il comando agisce su tutte le linee.

Un indirizzo : il comando agisce su tutte le linee corrispondenti.

Due indirizzi : il comando agisce su tutte le linee corrispondenti tra i due indirizzi inclusi.

! : il comando agisce su tutte le linee non corrispondenti all'indirizzo/agli indirizzi.

Tornando alla sintassi della linea di comando:

```
sed 'comando' file
sed 'indirizzocomando' file
sed 'indirizzo!comando' file
sed 'indirizzo,indirizzocomando' file
sed 'indirizzo,indirizzo!comando' file
```

```
$ sed '1,5=' file
$ sed 'p' file
$ sed -n 'p' file
$ sed -n '5s/modello/sostituto/p' file
$ sed -n '3,5s/modello/sostituto/w file2' file1
$ sed -n '3,5s/modello/sostituto/2w file2' file1
$ sed -i '/modello/d' file
$ sed -i.bak '/modello/d' file
$ sed -i '7a Una singola linea di testo.' file
```

I comandi **a**, **c**, **i** tal quali agiscono su una singola linea di testo. Perché possano agire su un testo di più linee, bisogna creare uno script, in cui ogni linea di testo (tranne l'ultima) termini con il carattere di controllo « \ », prima del consueto spazio di avanzamento di linea. Segue uno script d'esempio:

```
# Aggiungere un testo di più linee dopo la settima linea.
7a Prima linea di testo aggiunta.\
Seconda linea di testo aggiunta.\
Terza linea di testo aggiunta.

# Sostituire un testo di più linee alle linee 18, 19, 20.
18,20c Prima linea di testo sostituta.\
Seconda linea di testo sostituta.\
Terza linea di testo sostituta.

# Inserire un testo di più linee prima della linea 25.
25i Prima linea di testo inserita.\
Seconda linea di testo inserita.\
Terza linea di testo inserita.
```

Quindi:

```
$ sed -f script file
```

8.3.3 *Nl, Sort*

Nl

Nl numera le linee di un file. Il risultato appare su schermo.

Opzioni

-b, --body-numbering	Numerano le linee secondo una delle seguenti modalità: a : numera tutte le linee; t : numera solo le linee piene.
-----------------------------	---

Senza opzioni, **Nl** numera solo le linee piene (come **-bt**).

```
$ nl file (oppure: -bt file)
$ nl -ba file
```

Sort

Sort ordina le linee di un file. Il risultato appare su schermo.

Opzioni

-d, --dictionary-order	Effettuano un ordinamento alfabetico (prima i numeri, poi le lettere) similmente ai dizionari.
-n, --numeric-sort	Effettuano un ordinamento aritmetico (prima le lettere, poi i numeri).
-r, --reverse	Invertono l'ordine.
-u, --unique	Mostrano solo una volta le linee ripetute.

Senza opzioni, *Sort* effettua un ordinamento alfabetico (prima i numeri, poi le lettere).

```
$ sort file
$ sort -n file
```

8.3.4 *Cat, Paste, Join*

Cat

Cat mostra uniti in sequenza (concatena) i testi di due file.

```
$ cat file1 file2
```

Paste

Paste mostra affiancati in colonne (incolla) i testi di due file.

```
$ paste file1 file2
```

Join

Join mostra uniti gli elenchi numerati (con numeri o con lettere) presenti rispettivamente in due file: ogni voce del primo elenco viene completata dalla voce corrispondente del secondo elenco. I contrassegni degli elenchi devono essere uguali per entrambi. Per esempio, per le prime voci:

File 1	File 2
1.	1.
A -	A -
a)	a)

```
$ join file1 file2
```

8.3.5 *Split*

Split divide un file di grandi dimensioni in frammenti della lunghezza predefinita di 1000 linee e nominati con il prefisso predefinito x seguito da: aa, ab, ac ecc. L'indicazione di un prefisso sostituisce il prefisso predefinito.

Opzioni

-d, --numeric-suffixes	Impostano un prefisso numerico in luogo di quello predefinito alfabetico.
-l, --lines	Impostano un numero di linee differente da quello predefinito.

```
$ split file
$ split -dl 500 file BOZZA
```

8.4 Confronto

8.4.1 *Comm, Sdiff*

Comm

Comm confronta due file linea per linea. Mostra tre colonne:

Sinistra	Centro	Destra
Linee presenti solo nel primo file.	Linee presenti solo nel secondo file.	Linee presenti in entrambi i file.

```
$ comm file1 file2
```

Sdiff

Sdiff confronta due file linea per linea. Mostra due colonne, corrispondenti ai due file, separate da spazi e/o simboli:

Spazio = Linee uguali.
< = Linea in più a sinistra.
> = Linea in più a destra.
| = Linee differenti.

Opzioni

<code>-l, --left-column</code>	Mostrano le linee uguali solo nella colonna di sinistra.
<code>-s, --suppress-common-lines</code>	Non mostrano le linee uguali.

```
$ sdiff file1 file2
$ sdiff -l file1 file2
$ sdiff -s file1 file2
```

8.4.2 Merge

Merge confronta tre file linea per linea.

Opzioni

<code>-p</code>	Mostra uniti in sequenza i tre file.
-----------------	--------------------------------------

```
$ merge -p file1 file2 file3
```

8.5 Verifica

Durante l'invio/ricezione tra computer i dati possono corrompersi. Per ovviare a questa eventualità:

1. chi invia i dati calcola una stringa, sulla base dei dati da inviare, e la mette a disposizione in un file, a beneficio di chi deve ricevere i dati;
2. a sua volta chi riceve i dati calcola (con lo stesso programma) una stringa sulla base dei dati ricevuti: dal confronto tra le due stringhe si può verificare l'integrità dei dati trasmessi.

A seconda del programma utilizzato per il calcolo, la stringa in questione è definita: somma di controllo (*checksum*) o impronta del messaggio (*message digest*).

Esistono vari programmi di calcolo, con indicazioni differenti:

- per verificare l'alterazione accidentale dei dati, imputabile a difetti nella trasmissione, sono sufficienti i programmi: *Sum*, *Cksum*, *Md5sum*;
- per verificare eventuali alterazioni dolose sono necessari i programmi: *Sha1sum*, *Sha224sum*, *Sha256sum*, *Sha384sum*, *Sha512sum*.

8.5.1 *Sum*, *Cksum*, *Md5sum*

Sum

Sum mostra: la somma di controllo e il numero dei kilobyte di un file.

```
$ sum file
```

Cksum

Cksum mostra: la somma di controllo, il numero dei byte e il nome di un file.

```
$ cksum file
```


Md5sum

Md5sum (da Message Digest algorithm 5) mostra: l'impronta e il nome di un file.

```
$ md5sum file
```

8.5.2 *Sha1sum, Sha224sum, Sha256sum, Sha384sum, Sha512sum*

L'SHA (Secure Hash Algorithm = algoritmo di criptazione sicura) produce l'impronta mediante criptazione.

Sha1sum

Sha1sum mostra: l'impronta e il nome di un file.

```
$ sha1sum file
```

Sha224sum, Sha256sum, Sha384sum, Sha512sum

Sha224sum, Sha256sum, Sha384sum, Sha512sum sono i più sicuri. Mostrano: l'impronta e il nome di un file.

```
$ sha224sum file
$ sha256sum file
$ sha384sum file
$ sha512sum file
```

8.6 Conteggio

8.6.1 *Wc*

Wc mostra il conteggio degli elementi che compongono un testo.

Opzioni	
-c, --bytes	Mostrano il conteggio dei byte.
-l, --lines	Mostrano il conteggio delle linee.
-m, --chars	Mostrano il conteggio dei caratteri.
-w, --words	Mostrano il conteggio delle parole.

Senza opzioni, *Wc* mostra i numeri delle linee, delle parole e dei byte. Il numero dei byte è uguale o superiore al numero dei caratteri.

```
$ wc file
$ wc -w file
```

8.7 Controllo ortografico

8.7.1 *Aspell*

Aspell, che richiede l'installazione del pacchetto *aspell-it*, contenente il dizionario italiano (il dizionario predefinito è quello inglese: pacchetto *aspell-en*), confronta ogni parola di un testo con il dizionario italiano. Lo schermo appare suddiviso in due parti:

- superiore, contenente il testo, in cui in successione vengono evidenziate le parole non corrispondenti;
- inferiore, contenente:
 - l'elenco (contrassegnato con numeri) delle possibili alternative;
 - l'elenco (contrassegnato con lettere) delle possibili azioni.

Opzioni

-c, check	Effettuano il controllo ortografico.
------------------	--------------------------------------

```
$ aspell -c file
```

8.8 Formattazione

8.8.1 *Fold*

Fold spezza le linee alla larghezza massima specificata (larghezza predefinita = 80 colonne).

Opzioni

-s, --spaces	Spezzano le linee in corrispondenza degli spazi, possibilmente.
-w, --width	Impostano la larghezza massima.

```
$ fold file
$ fold -s file
$ fold -w 100 file
```

8.9 I filtri

I programmi non interattivi di gestione dei testi possono fungere da filtri, agendo in modo sequenziale: infatti, un filtro è in grado di ricevere l'output di un comando precedente come proprio input, elaborarlo e produrre un proprio output, che appare nello standard output.

In altri termini, i filtri operano su flussi (*stream*) di dati: i dati dell'output di un comando precedente diventano i dati dell'input del comando (filtro) successivo, scorrono (fluiscono) attraverso di questo, che li elabora e produce i dati del proprio output.

I filtri possono collaborare tra loro: più filtri possono susseguirsi nella stessa linea di comando, in una sorta di reazione a catena, per mezzo dei caratteri speciali propri della shell e dei modelli (le stringhe letterali e le espressioni regolari con, a loro volta, i propri metacaratteri).

I filtri operano principalmente su input di tipo testo, ma non solo. Infatti, la shell può interpretare ed espandere i metacaratteri delle espressioni regolari in due modi:

- inclusi tra caratteri di citazione, come (appartenenti a) nomi di file: per la ricerca di file;
- tal quali, come (appartenenti a) modelli di espressioni regolari: per la ricerca all'interno dei file e per la modifica di file.

I filtri possono essere:

- facoltativi, come i programmi descritti finora in questo Capitolo;
- obbligati, come quelli che seguono.

8.9.1 I filtri obbligati: *Tee*, *Tr*, *Xargs*

Tee

Tee mostra l'output e contemporaneamente lo copia in un file (biforcazione a «T» dell'output). Usato per la creazione dei file di registro (*log*).

Opzioni

-a	Aggiunge l'output in coda al testo già presente, senza sovrascrivere il file.
-----------	---

Senza opzioni, *Tee* crea un file non esistente o sovrascrive un file esistente.

```
$ cat file1 | tee file2 (crea un file o sovrascrive un file)
$ ls -l | tee -a file
```

Tr

Tr cancella, sostituisce o accorcia delle serie di caratteri (*set*) dallo standard input. La sintassi prevede due forme possibili:

```
tr -d/s 'serie'
sed -s/t 'serie1' 'serie2'
```

Opzioni

-d, --delete	Cancellano una serie.
-s, --squeeze-repeats	Sostituiscono un carattere ripetuto nella prima serie con una sua singola occorrenza nella seconda serie.
-t, --truncate-set1	Accorcia la prima serie alla lunghezza della seconda serie.

Senza opzioni, **Tr** sostituisce la prima serie con la seconda serie.

```
$ tr a-z A-Z < file
$ tr A-Z a-z < file1 > file2
$ echo testo minuscolo. | tr a-z A-Z
$ echo Testo con spazi doppi. | tr ' ' ' '
$ echo Testo con spazi singoli. | tr ' ' '\t'
$ tr -d serie < file
$ cat file | tr -d serie
$ cat file | tr -s ' '
$ ls -l | tr -s ' ' oppure $ ls -l | tr -s ' ' ' '
```

Xargs

Xargs riceve elenchi (per esempio, prodotti da *Ls* e da *Find*) e li cede al programma seguente come argomenti multipli da elaborare in successione.

```
$ ls -l directory1 | xargs mv *.txt directory2
$ find directory | xargs grep modello
```

8.9.2 Esempi vari

```
$ ls | wc -l
$ ls -l | tr -s ' ' | cut -d' ' -f1,6-
$ ls -l | tr -s ' ' | cut -d' ' -f1,6- | tr ' ' '\t'

$ find directory -print | sort
$ find directory -print -exec grep [Ss]tringa {} \; > file
$ find directory | xargs grep [Ss]tringa > file
$ find directory -print | xargs grep [Ss]tringa > file

$ sed -f script.sed < file1 > file2
$ cat file | sed 's/.../.../g'

$ sort file | uniq
$ wc -l * | sort
$ wc -l * | sort -r

$ cat file1 | paste - file2 oppure $ cat file1 | paste file1 file2
$ cat file | split -dl 100 - new.
```

Capitolo 9

Gestione dei pacchetti

Ogni programma è prodotto in forma di codice sorgente. Il sorgente va compilato, per poter essere installato. I pacchetti precompilati, essendo già stati sottoposti a compilazione, possono essere installati direttamente. La gestione dei pacchetti è compito di root. I pacchetti specifici di *Slackware* attualmente sono compressi con *Lzma* e terminano con *.txz*, ma si incontrano ancora pacchetti compressi con *Gzip* e terminanti con *.tgz*.

I pacchetti precompilati sono disponibili negli appositi archivi di software (*repository*) presenti sui siti di *Slackware*, delle distribuzioni derivate, di comunità e di singoli; esistono anche dei siti di ricerca dei pacchetti.

Va detto che *Slackware* riguardo ai pacchetti è una distribuzione chiusa, cioè contempla come ufficialmente autorizzati soltanto i pacchetti inclusi nella ISO e quelli presenti nei propri archivi di software, che dopo il rilascio della ISO continuano ad essere aggiornati. Tutti gli altri archivi sono considerati come non ufficiali: Volkerding si fa garante soltanto dei propri pacchetti.

D'altra parte, è frequente il ricorso agli altri archivi per prelevare i pacchetti di programmi non inclusi nella distribuzione, ma di uso comune (*Adobe Reader*, *LibreOffice*, *Opera* ecc.). Per quanto riguarda i pacchetti di dubbia sicurezza, si può provare ad installarli preventivamente in un'altra partizione non operativa, dedicata alle personali sperimentazioni.

9.1 Gestione manuale

Slackware adotta per la gestione manuale dei pacchetti i programmi del pacchetto *pkgtools*: a) i programmi di primo livello: *Installpkg*, *Upgradepkg* e *Removepkg*; b) il programma di secondo livello *Pkgtool*, interfaccia semigrafica con numerose funzioni, tra cui quella di gestire i pacchetti attraverso i programmi di primo livello.

La distribuzione *Salix*, una derivata di *Slackware* pienamente compatibile, adotta come gestore di pacchetti di primo livello il più veloce *Spkg*, prelevabile dagli archivi di *Salix*.

9.1.1 *Installpkg*, *Upgradepkg* e *Removepkg*

- *Installpkg* installa:

```
# installpkg pacchetto.txz
```

- *Upgradepkg* aggiorna:

- se i pacchetti vecchio e nuovo hanno lo stesso nome:

```
# upgradepkg pacchetto-nuovo.txz
```

- se i pacchetti vecchio e nuovo hanno nomi diversi:

```
# upgradepkg pacchetto-vecchio.txz%pacchetto-nuovo.txz
```

- *Removepkg* disinstalla:

```
# removepkg pacchetto.txz
```

9.1.2 *Pkgtool*

Pkgtool installa, aggiorna e disinstalla; inoltre, consente di visualizzare le informazioni relative ai singoli pacchetti e di selezionare più pacchetti da disinstallare:

```
# pkgtool
```

9.1.3 *Spkg*

Spkg effettua: installazione, aggiornamento e disinstallazione.

```
# spkg -i pacchetto.txz
# spkg -u pacchetto.txz
# spkg -d pacchetto.txz
```

9.2 Gestione automatica

La gestione automatica richiede la connessione ad Internet, in particolare agli archivi di software da cui prelevare i nuovi pacchetti e le nuove versioni dei pacchetti già installati, e può essere operata da:

- *Slackpkg*;
- *Slapt-get* e *Gslapt*.

9.2.1 Aggiornamento di *Slackware*

Slackware è una distribuzione in continuo sviluppo, disponibile in due versioni:

- una versione stabile (attualmente la 14.0), che corrisponde a *Slackware* fissata –come in una fotografia– in un certo momento del suo sviluppo;
- una versione sempre provvisoria (*current*), che corrisponde a *Slackware* nel suo continuo divenire.

Aggiornamento della versione stabile

La versione stabile non è statica e può essere aggiornata con i pacchetti che man mano sono considerati stabili. L'aggiornamento può essere effettuato in due modi alternativi tra loro: a) per mezzo di *Slackpkg*, rigorosamente dagli archivi di *Slackware*; oppure b) per mezzo di *Slapt-get*/*Gslapt*, dagli archivi di *Salix* o di *Slacky* (anche questi alternativi tra loro).

La scelta del criterio di aggiornamento va ponderata e possibilmente stabilita una volta per tutte, perché in seguito eventuali ripensamenti diventano sempre meno praticabili quanto più tardi vengono attuati. Inoltre, bisogna sapere che in caso di abbandono del sistema di aggiornamento ufficiale con *Slackpkg* il passaggio al sistema di aggiornamento alternativo con *Slapt-get*/*Gslapt* è irreversibile.

Aggiornamento della versione provvisoria

La versione provvisoria (*current*) può essere aggiornata soltanto con *Slackpkg* dagli archivi di *Slackware*.

9.2.2 *Slackpkg*

Slackpkg è un programma testuale di terzo livello, che agisce attraverso *Pkgtool*: aggiorna *Slackware* esclusivamente dagli archivi di *Slackware*, relativamente ai soli pacchetti ufficiali.

Eventuali pacchetti estranei all'installazione ufficiale di *Slackware* vanno installati/rimossi manualmente.

Configurazione

In `/etc/slackpkg/` si trovano i file: `blacklist`, `mirrors` e `slackpkg.conf`.

1. Nel file **mirrors** bisogna decommentare un sito speculare (*mirror*), da cui prelevare le informazioni relative ai pacchetti da aggiornare e i pacchetti stessi. Per esempio:

```
#-----
# Slackware64-14.0
#-----
...
# IRELAND (IE)
# ftp://ftp.heanet.ie/mirrors/ftp.slackware.com/pub/slackware/slackware64-14.0/
# http://ftp.heanet.ie/mirrors/ftp.slackware.com/pub/slackware/slackware64-14.0/
...
```

oppure

```
#-----
# Slackware64-current
#-----
...
# IRELAND (IE)
# ftp://ftp.heanet.ie/mirrors/ftp.slackware.com/pub/slackware/slackware64-current/
http://ftp.heanet.ie/mirrors/ftp.slackware.com/pub/slackware/slackware64-current/
...
```

2. Nel file **blacklist** bisogna elencare i pacchetti che devono essere esclusi dall'aggiornamento automatico, come i pacchetti relativi al kernel (non necessariamente: basta avviare *LILO* alla fine dell'aggiornamento), altri pacchetti di sistema (per esempio, **aaa_elflibs**) e i pacchetti non ufficiali installati manualmente. Riguardo a questi ultimi segue un esempio:

```
# Pacchetti prelevati da Salix
libreoffice
libreoffice-help-it
libreoffice-l10n-it
# Pacchetti prelevati da Slackers
postfix
tor
# Pacchetti prelevati da Slacky
flash-player-plugin
medit
```

3. Prelevare i file che occuperanno la directory `/var/lib/slackpkg/` (inizialmente vuota):

```
# slackpkg update
```

In particolare, interessano i file: `ChangeLog.txt`, `PACKAGES.TXT` e `pkglist`.

La sintassi generale è:

```
slackpkg comando pacchetto/file/espressione
```

Comandi

update	Preleva gli elenchi aggiornati dei pacchetti.
search	Cerca i pacchetti per mezzo di espressioni.
file-search	Cerca un file (per esempio, una libreria) nei pacchetti.
install	Preleva ed installa un pacchetto.
upgrade	Aggiorna un pacchetto alla nuova versione.
remove	Disinstalla un pacchetto.
download	Preleva un pacchetto.
reinstall	Reinstalla un pacchetto già installato.
install-new	Preleva ed installa tutti i nuovi pacchetti.
upgrade-all	Aggiorna tutti i pacchetti di cui esista una nuova versione.
clean-system	Disinstalla i pacchetti ufficiali diventati superflui e i pacchetti non ufficiali.
info	Mostra informazioni d'aiuto sulla sintassi generale.
help	Mostra informazioni d'aiuto sui comandi.

A regime la normale sequenza di comandi per l'aggiornamento dell'intera distribuzione (relativamente ai pacchetti ufficiali) è la seguente:

```
# slackpkg update
# slackpkg install-new
# slackpkg upgrade-all
# slackpkg clean-system
```

Al termine dell'installazione dei nuovi pacchetti (con: `# slackpkg install-new`) e dell'aggiornamento dei pacchetti presenti (con: `# slackpkg upgrade-all`) appare la richiesta di scegliere il destino dei nuovi file di configurazione:

```
...
Some packages had new configuration files installed.
You have four choices:

(K)eep the old files and consider .new files later

(O)verwrite all old files with the new ones. The
  old files will be stored with the suffix .orig

(R)emove all .new files

(P)rompt K, O, R selection for every single file
```

What do you want (K/O/R/P)?

- **k** lascia intatti i vecchi file di configurazione e aggiunge i nuovi con l'estensione `.new`: consente di rimandare la visualizzazione dei nuovi file ed eventualmente modificare le varie configurazioni;
- **o** sostituisce i vecchi file di configurazione, rinominati con l'estensione `.orig`, con i nuovi: costringe a rifare subito tutte le configurazioni;
- **r** lascia intatti i vecchi file di configurazione ed elimina i nuovi: può essere imprudente;
- **p** consente di scegliere separatamente per ogni file.

Da preferire senz'altro: **k**.

9.2.3 Slapt-get e Gslapt

La distribuzione *Salix* adotta **Slapt-get**, programma testuale di secondo livello che agisce attraverso *Spkg*. **Gslapt** è l'interfaccia grafica di **Slapt-get**. Anche **Slapt-get** e **Gslapt** sono prelevabili dagli archivi di *Salix*.

Configurazione

In `/etc/slapt-get/` si trova il file di configurazione: **slapt-getrc**. Per esempio:

```
WORKINGDIR=/var/slapt-get
EXCLUDE=~aaa_elflibs,~devs,~glibc-.*,~kernel-.*,~udev,.*-[0-9]+dl$,x86_64
EXCLUDE=~aaa_elflibs,~aaa_base,~devs,~glibc.*,~kernel-.*,~udev,~rootuser-settings,~zzz-settings.*,~i?86-
SOURCE=http://salix.enialis.net/x86_64/slackware-14.0/:OFFICIAL
SOURCE=http://salix.enialis.net/x86_64/slackware-14.0/extra/:OFFICIAL
SOURCE=http://salix.enialis.net/x86_64/14.0/:PREFERRED
```

- **WORKINGDIR** indica la directory `/var/slapt-get/`, che contiene:
 - i pacchetti di volta in volta prelevati dai vari archivi;
 - il file `package_data`, che elenca tutti i pacchetti presenti negli archivi (disponibili e installati) con le relative informazioni.
- **EXCLUDE** elenca i pacchetti normalmente esclusi da modifiche.
- **SOURCE** indica l'indirizzo di uno degli archivi.

La sintassi generale è:

`slapt-get opzione e/o comando ± pacchetto/espressione`

Opzioni

<code>-d, --download-only</code>	Prelevano un pacchetto.
<code>--reinstall</code>	Reinstalla un pacchetto già installato.
<code>--ignore-excludes</code>	Include i pacchetti normalmente esclusi da modifiche.

Comandi

<code>-u, --update</code>	Aggiornano i dati relativi ai pacchetti presenti negli archivi.
<code>--search</code>	Cerca i pacchetti per mezzo di espressioni.
<code>--list</code>	Elenca tutti i pacchetti: disponibili e installati.
<code>--available</code>	Elenca solo i pacchetti disponibili.
<code>--installed</code>	Elenca solo i pacchetti installati.
<code>--show</code>	Mostra la descrizione di un pacchetto.
<code>--filelist</code>	Mostra l'elenco dei file di un pacchetto.
<code>-i, --install</code>	Prelevano ed installano uno o più pacchetti.
<code>--upgrade</code>	Aggiorna tutti i pacchetti di cui esista una nuova versione.
<code>--remove</code>	Disinstalla uno o più pacchetti.
<code>--clean</code>	Elimina tutti i pacchetti contenuti in <code>/var/slapt-get/</code> .
<code>-h, --help</code>	Mostrano informazioni d'aiuto.

```
# slapt-get --filelist pacchetto
# slapt-get --search editor
# slapt-get --clean
```

9.3 Conversione di pacchetti estranei

In caso di irreperibilità di pacchetti nativi, si può provare a convertire pacchetti (non di sistema) estranei a *Slackware*: il risultato non è sempre garantito. La conversione può essere effettuata da:

- *Alien*;
- *Rpm2tgz*.

9.3.1 *Alien*

La conversione di pacchetti specifici di altre distribuzioni, in particolare *Debian* e *Red Hat* (e loro derivate o simili), è prodotta da *Alien*.

Opzioni

<code>-t, --to-tgz</code>	Convertono un pacchetto <code>.deb</code> o <code>.rpm</code> in un pacchetto <code>.tgz</code> .
<code>-i, --install</code>	Convertono, installano ed eliminano in successione il pacchetto creato.

```
$ alien -t pacchetto.deb
# alien -t -i pacchetto.rpm
```

9.3.2 *Rpm2tgz*

La conversione di pacchetti specifici della distribuzione *Red Hat* (e simili) è prodotta anche da *Rpm2tgz*:

```
$ rpm2tgz pacchetto.rpm
```


Capitolo 10

Creazione guidata dei pacchetti

La creazione guidata di un pacchetto consiste nella compilazione del sorgente attraverso uno script. La creazione dei pacchetti è compito di root.

10.1 Creazione manuale

Esistono due modi:

- il metodo classico, che si basa sugli SlackBuild;
- il metodo alternativo, adottato da *Salix*, che si basa sugli SLKBUILD.

10.1.1 Operazioni preliminari

Le operazioni preliminari sono comuni.

1. Nella propria home directory creare una directory di lavoro con il nome del pacchetto:

```
$ mkdir pacchetto
```

2. Prelevare i file necessari (SlackBuild/SLKBUILD ecc.) e salvarli nella directory.
3. Accedere alla directory:

```
$ cd pacchetto
```

4. Per sicurezza, conviene evitare l'intervento diretto di root e avviare *bash* mediante **Fakeroot** (prelevabile presso *Salix*), che simula i privilegi di root:

```
$ fakeroot bash
```

10.1.2 Metodo classico (SlackBuild)

Lo **SlackBuild** è uno script che compila il sorgente e crea il pacchetto. Gli SlackBuild, insieme con altri file utili ed eventualmente i relativi sorgenti (o gli indirizzi necessari a prelevarli), sono disponibili negli appositi archivi di *Slackware*, delle derivate, di comunità e di singoli.

1. Controllare ed eventualmente modificare i permessi dello script:

```
# ls -l pacchetto.SlackBuild
# chmod 755 pacchetto.SlackBuild
```

2. Creare il pacchetto:

```
# ./pacchetto.SlackBuild
```

10.1.3 Metodo alternativo (SLKBUILD)

Lo **SLKBUILD** è un metafile, cioè un file di testo contenente i dati sufficienti a generare lo script che compila il sorgente e crea il pacchetto. Gli SLKBUILD, insieme con altri file utili e i relativi sorgenti, sono disponibili negli archivi di *Salix*.

1. Avviare **Slkbuild** (disponibile negli archivi di *Salix*), che preleva il sorgente, se assente, e crea lo script **build-pacchetto.sh**:

```
# slkbuild
```

2. Creare il pacchetto:

```
# build-pacchetto.sh
```

10.2 Creazione automatica

La creazione automatica richiede la connessione ad Internet, in particolare agli archivi di software da cui prelevare gli SlackBuild/SLKBUILD (e il resto). La creazione automatica, adottata da *Salix*, è operata da *Slapt-src* e *Sourcery* (prelevabili presso *Salix*).

10.2.1 *Slapt-src* e *Sourcery*

Slapt-src è un programma testuale, che crea i pacchetti indifferentemente attraverso gli SlackBuild o gli SLKBUILD. *Sourcery* è l'interfaccia grafica di *Slapt-src*.

Configurazione

In `/etc/slapt-get/` si trova il file di configurazione: **slapt-srcrc**. Per esempio:

```
BUILDDIR=/usr/src/slapt-src
PKGEXT=txz
SOURCE=http://salix.enialis.net/slkbuilt/14.0/
SOURCE=http://salix.enialis.net/sbo/14.0/
```

- BUILDDIR indica la directory `/usr/src/slapt-src/`, che contiene:
 - i componenti necessari a creare i pacchetti, di volta in volta prelevati dagli archivi;
 - il file `slackbuilds_data`, che elenca tutti gli SlackBuild presenti negli archivi (disponibili e installati) insieme con le relative informazioni.
- PKGEXT indica l'estensione dei pacchetti che verranno creati.
- SOURCE indica l'indirizzo di uno degli archivi.

La sintassi generale è:

```
slapt-src comando ± pacchetto/espressione
```

Comandi

-u, --update	Aggiornano i dati relativi agli SlackBuild presenti negli archivi.
-s, --search	Cercano gli SlackBuild per mezzo di espressioni.
-l, --list	Elencano tutti gli SlackBuild: disponibili e installati.
-w, --show	Mostrano la descrizione di uno SlackBuild.
-f, --fetch	Prelevano uno o più SlackBuild.
-b, --build	Prelevano uno o più SlackBuild e ne creano il pacchetto corrispondente.
-i, --install	Prelevano uno o più SlackBuild, ne creano e installano il pacchetto corrispondente.
-e, --clean	Eliminano tutti gli SlackBuild e i pacchetti contenuti in <code>/usr/src/slapt-src/</code> .
-h, --help	Mostrano informazioni d'aiuto.

Capitolo 11

Creazione *ex novo* dei pacchetti

La creazione *ex novo* di un pacchetto consiste nella compilazione del sorgente previa configurazione. Anche in questo caso la creazione dei pacchetti è compito di root.

11.1 Creazione manuale

Come esempio generale si descrive la creazione del pacchetto del programma *Programma* (versione *N.N.N*) per architetture a 64 bit.

11.1.1 Operazioni preliminari

1. Prelevare (dal sito dove è disponibile) l'archivio compresso **programma-N.N.N.tar...**, contenente il sorgente, salvarlo nella propria home directory e decomprimerlo. Si crea la directory **/programma-N.N.N/**, contenente varie directory e diversi file, tra cui eseguibili e documentazione.
2. Nella propria home directory creare la directory di lavoro **/PROGRAMMA/**, dove verrà effettuata l'installazione di prova, e la subdirectory **/install/**, destinata a contenere dei file informativi:

```
$ mkdir PROGRAMMA PROGRAMMA/install
```

3. Accedere nella directory **/programma-N.N.N/** e cambiare i permessi dei file eseguibili e non eseguibili:

```
$ cd programma-N.N.N
$ find . -perm /111 -exec chmod 755 {} \;
$ find . ! -perm /111 -exec chmod 644 {} \;
```

4. Per sicurezza, avviare *Bash* simulando i privilegi di root:

```
$ fakeroot bash
```

11.1.2 Configurazione

1. Visualizzare il testo d'aiuto dello script **configure** con le informazioni necessarie alla configurazione:

```
# ./configure --help
```

Opzioni

--prefix=PREFIX	Imposta la destinazione di installazione: /usr/ .
--libdir=DIR	Imposta la destinazione delle librerie: /usr/lib64/ .
--mandir=DIR	Imposta la destinazione della documentazione di <i>Man</i> : /usr/man/ .

Variabili

CFLAGS	Parametri per il compilatore <i>Gcc</i> . In particolare: <ul style="list-style-type: none">• i parametri di ottimizzazione (-O);• i parametri relativi al processore (-m). Vedere: https://it.wikipedia.org/wiki/CFLAGS http://gcc.gnu.org/onlinedocs/gcc/i386-and-x86_002d64-Options.html <pre>\$ man -t gcc > gcc.ps && ps2pdf gcc.ps && rm gcc.ps</pre>
---------------	--

2. Procedere con la configurazione. Per esempio:

```
# ./configure --prefix=/usr --libdir=/usr/lib64 --mandir=/usr/man
CFLAGS='-O2 -march=k8 -mtune=k8'
```

In alternativa, si può creare in `/programma-N.N.N/` uno script, per esempio **myconfig** (modificabile e riutilizzabile in altre occasioni):

```
#!/bin/bash

configure \
--prefix=/usr \
--libdir=/usr/lib64 \
--mandir=/usr/man

CFLAGS='-O2 -march=k8 -mtune=k8'
```

e dare:

```
# ./myconfig
```

11.1.3 Compilazione

Compilare il sorgente:

```
# make
```

11.1.4 Installazione di prova

Indirizzare l'installazione nella directory di lavoro `/PROGRAMMA/`:

```
# make install DESTDIR=~/PROGRAMMA
```

L'installazione è contenuta nella subdirectory `/usr/`.

11.1.5 Pulitura

1. Accedere alla directory `/PROGRAMMA/`:

```
# cd ~/PROGRAMMA
```

2. Alleggerire i file eseguibili e le eventuali librerie delle parti di codice non necessarie al loro funzionamento.

Opzioni

--strip-unneeded	Elimina tutte le parti di codice usate nell'elaborazione preinstallazione, ma non più necessarie.
-------------------------	---

```
# strip --strip-unneeded usr/bin/programma
# strip --strip-unneeded usr/lib64/libreria
```

11.1.6 Elaborazione della documentazione

L'elaborazione della documentazione è in parte automatica e in parte manuale.

1. In `/usr/man/` comprimere con *Gzip* i file di documentazione per *Man*, che possono essere presenti in uno o due livelli:

```
# gzip -v9 usr/man/*/*
```

ed eventualmente:

```
# gzip -v9 usr/man/*/*/*
```

2. In `/usr/doc/` o in `/usr/share/doc/` (secondo i casi; ma la sostanza non cambia, perché in *Slackware* `/usr/share/doc/` è un collegamento a `/usr/doc/`), accanto ad un'eventuale directory già presente, creare la directory **/programma-N.N.N/**:

```
# mkdir usr/doc/programma-N.N.N (oppure # mkdir usr/share/doc/programma-N.N.N)
```

e copiarvi i file di qualche interesse (dai nomi a lettere maiuscole) presenti in `~/programma-N.N.N/` (come: AUTHORS, BUGS, COPYING, COPYING.GPL, ChangeLog, ChangeLog.CVS, INSTALL, LICENSE, NEWS, README, THANKS, TODO).

3. In `/install/` creare il file descrittivo **slack-desc**:

```
# touch install/slack-desc
```

e inserirvi (cercando di non superare le 80 colonne), per esempio:

```
# HOW TO EDIT THIS FILE:
# The "handy ruler" below makes it easier to edit a package description. Line
# up the first '|' above the ':' following the base package name, and the '|'
# on the right side marks the last column you can put a character in. You must
# make exactly 11 lines for the formatting to be correct. It's also
# customary to leave one space after the ':'.

      |-----handy-ruler-----|
programma: programma (descrizione sintetica)
programma:
programma: Programma descritto in modo più esauriente.
programma: Indirizzo in rete del sorgente
programma:
programma:
programma:
programma:
programma: Pacchetto creato da Nome Cognome
programma: Indirizzo in rete dell'autore
```

La parte generica è facoltativa. La parte specifica (undici linee) è obbligatoria: quello che viene inserito a destra dei due punti (« : ») comparirà con l'opzione **--infobox** di *Installpkg*:

```
# installpkg --infobox programma
```

Chiudere il terminale usato finora.

Creazione

Aprire un nuovo terminale.

1. Accedere alla directory **/PROGRAMMA/**:

```
$ cd PROGRAMMA
```

2. Creare il pacchetto con **Makepkg**:

Opzioni

-c y/n, --chown y/n L'affermazione (y) imposta la proprietà e i permessi di tutte le directory, rispettivamente: `root:root` e `755`.

Sono necessari i privilegi effettivi di root:

```
$ su
# makepkg -c n ../programma-N.N.N-x86_64-Nnc.txz
```

Il pacchetto **programma-N.N.N-x86_64-Nnc.txz** viene creato nella propria home directory, ma è proprietà di root (che possiede i privilegi necessari ad installare il pacchetto nel sistema). Il gruppo Nnc comprende la versione del pacchetto (N) e le iniziali dell'autore del pacchetto (nc = nome cognome).

3. Cambiare il proprietario del pacchetto:

```
# chown utente:gruppo programma-N.N.N-x86_64-Nnc.txz
```

11.2 Esempi

11.2.1 Creazione del pacchetto di *Sitecopy*

Operazioni preliminari

Indirizzo del sorgente: <http://www.manyfish.co.uk/sitecopy/>.

```
$ tar xzvf sitecopy-0.16.6.tar.gz (si crea la directory /sitecopy-0.16.6/)
$ mkdir SITECOPY SITECOPY/install (per creare le directory /SITECOPY/ e /SITECOPY/install/)
$ cd sitecopy-0.16.6
$ find . -perm /111 -exec chmod 755 {} \;
$ find . ! -perm /111 -exec chmod 644 {} \;
$ fakeroot bash
```

Configurazione

```
# configure --help
# configure --prefix=/usr --mandir=/usr/man --disable-debug CFLAGS='-O2 -march=k8 -mtune=k8'
```

oppure

```
# myconfig
```

Compilazione

```
# make
```

Installazione di prova

```
# make install DESTDIR=~ /SITECOPY
```

Pulitura

Non è necessaria.

Elaborazione della documentazione

```
# cd ~/SITECOPY
# gzip -v9 usr/man/*/*
# gzip -v9 usr/man/*/*/*
# mkdir usr/doc/sitecopy-0.16.6 (per creare la directory /sitecopy-0.16.6/, nella quale inserire i file:
                                BUGS, COPYING, ChangeLog, ChangeLog.CVS, INSTALL, NEWS,
                                README, THANKS, TODO)
# touch install/slack-desc (per creare il file slack-desc)
```

In slack-desc inserire, ad esempio:

```
...
sitecopy: sitecopy (gestione di siti web)
sitecopy:
sitecopy: Sitecopy sincronizza un sito web con il corrispondente sito locale
sitecopy: con un solo comando.
sitecopy: http://www.manyfish.co.uk/sitecopy/
sitecopy:
sitecopy:
sitecopy:
sitecopy:
sitecopy: Pacchetto creato da Luigi Carusillo --- PGL - Piccole Guide Linux
sitecopy: http://lgcrsll.altervista.org/pgi/pgl/index.html
```

Chiudere il terminale.

Creazione

In un nuovo terminale:

```
$ cd SITECOPY
$ su
# makepkg -c n ../sitecopy-0.16.6-x86_64-1lc.txz
# chown luigi:users sitecopy-0.16.6-x86_64-1lc.txz
```

11.2.2 Creazione del pacchetto di *Medit*

Operazioni preliminari

Indirizzo del sorgente: <http://moedit.sourceforge.net/>.

```
$ tar xjvf medit-1.1.1.tar.bz2 (si crea la directory /medit-1.1.1/)
$ mkdir MEDIT MEDIT/install (per creare le directory /MEDIT/ e /MEDIT/install/)
$ cd medit-1.1.1
$ find . -perm /111 -exec chmod 755 {} \;
$ find . ! -perm /111 -exec chmod 644 {} \;
$ fakeroot bash
```

Configurazione

```
# configure --help
# configure --prefix=/usr --mandir=/usr/man CFLAGS='-O2 -march=k8 -mtune=k8'
```

oppure

```
# myconfig
```

Aggiungendo l'opzione `-fPIC` ai parametri per *Gcc* (... `CFLAGS='-O2 -fPIC -march=k8 -mtune=k8'`), si produce un eseguibile di dimensioni di gran lunga minori, ma comunque destinato alla pulitura con *Strip*.

Compilazione

```
# make
```

Installazione di prova

```
# make install DESTDIR=~ /MEDIT
```

Pulitura

```
# cd ~/MEDIT
# strip --strip-unneeded usr/bin/medit
```

Elaborazione della documentazione

```
# gzip -v9 usr/man/*/*
# mkdir usr/share/doc/medit-1.1.1 (per creare la directory /medit-1.1.1/, nella quale inserire i file:
AUTHORS, COPYING, INSTALL, NEWS, README, THANKS)
# touch install/slack-desc (per creare il file slack-desc)
```

In slack-desc inserire, ad esempio:

```
...
medit: medit (programma di videoscrittura)
medit:
medit: Medit è un programma di videoscrittura dotato di numerose
medit: caratteristiche.
medit: http://moedit.sourceforge.net/
medit:
medit:
medit:
medit: Pacchetto creato da Luigi Carusillo --- PGL - Piccole Guide Linux
medit: http://lgcrsll.altervista.org/pgi/pgl/index.html
```

Chiudere il terminale.

Creazione

In un nuovo terminale:

```
$ cd MEDIT
$ su
# makepkg -c n ../medit-1.1.1-x86_64-1lc.txz
# chown luigi:users medit-1.1.1-x86_64-1lc.txz
```


Bibliografia

- [1] AA.VV. Documentazione presente in *Slackware*.
\$ `man ...`
\$ `locate ...`
- [2] Luigi Carusillo. *LaTeX*.
<http://lgcrsll.altervista.org/pgi/pgl/latex.html>
- [3] Luigi Carusillo. *Netfilter e Iptables*.
<http://lgcrsll.altervista.org/pgi/pgl/internet.html>
- [4] Luigi Carusillo. *Vim*.
<http://lgcrsll.altervista.org/pgi/pgl/videoscrittura.html>
- [5] Luigi Carusillo. *WvDial*.
<http://lgcrsll.altervista.org/pgi/pgl/internet.html>
- [6] Alan Hicks ed al. *Slackware Linux Essentials*.
<http://book.slackware.it/>
- [7] N.N. *HIER(7)*.
\$ `man hier`
- [8] Mauro Sacchetto. *Slackware for dummies*.
<http://www.slackyeu/slacky/Slackware4Dummies>

Indirizzi

Achtung alles lookenspeepers!	http://www.slackware.com/~alien/
Linux Packages Search	http://pkgs.org/
Robby Workman's Homepage	http://rlworkman.net/
Salix OS	http://www.salixos.org/wiki/index.php?title=Home
SlackBuilds.org	http://slackbuilds.org/
Slackel	http://www.slackel.gr/
Slackers	http://www.slackers.it/
Slackfind	http://slackfind.net/en/
Slacky.eu - Italian Slackware Community	http://www.slacky.eu/slacky/Pagina_principale
Slackware Linux Project Italia	http://slackware.it/
Slakfinder	http://slakfinder.org/
The Slackware Linux Project	http://www.slackware.com/



LICENZA PGI – PICCOLE GUIDE D'INFORMATICA

Le PGI – PICCOLE GUIDE D'INFORMATICA, o più brevemente PGI, sebbene siano redatte con grande accuratezza, sono sprovviste di qualsiasi forma di garanzia: è sempre opportuno procedere a verifiche.

Tutte le PGI possono essere riprodotte e/o distribuite totalmente o parzialmente in piena libertà. La ridistribuzione commerciale delle PGI non è vietata, ma neanche incoraggiata.

Tutte le PGI possono essere modificate totalmente o parzialmente in piena libertà. Le modifiche fatte in accordo con l'autore originale rendono l'autore delle modifiche coautore. Le modifiche fatte senza l'accordo con l'autore originale sono ammesse ma non incoraggiate: sarebbe auspicabile che, semplicemente in ossequio al galateo, riportassero il nome dell'autore originale e il nome del revisore.

Lavori derivati dalle PGI o comprendenti parti di PGI sono non solo ammessi ma incoraggiati: sarebbe auspicabile che, semplicemente in ossequio al galateo, riportassero i riferimenti alle PGI da cui derivano o che comprendono.

Tutte le PGI possono essere tradotte in piena libertà: sarebbe auspicabile che le traduzioni, semplicemente in ossequio al galateo, riportassero il nome dell'autore e il nome del traduttore.