Corso introduttivo all'utilizzo di UNIX
L'informatica per tutti

FreakNet MediaLab Catania
http://www.freaknet.org

this is version **1.0.20031001**

1 ottobre 2003

Copright Notices & more

Prefazione

L’informatica è una cosa seria: per questo ve la presentiamo con leggerezza. Ci sono tanti libri sullo Unix molto più esaurienti di questa raccolta di dispense e anche molto più noiosi. Il FreakNet MediaLab ha maturato negli anni l’esperienza che gli ha permesso di capire che è meglio avere pochi concetti - ma chiari - e sperimentare a partire da fondamenta molto solide. Magari divertendosi. Insomma, più che un corso questo testo vuole appoggiare una serie di workshop, di officine sperimentali dove la gente pratichi più di professare. Dal 1999 il FreakNet MediaLab ha organizzato svariate tornate di corsi rivolti a chi dello Unix non sa niente. Il progetto di fornire le conoscenze di base per muoversi in questo ambiente è partito con l’apertura del laboratorio al pubblico e si è rinnovato grazie al favore che i sistemi Unix-like hanno trovato negli ultimi anni, specie da quando GNU/Linux ha cominciato ad affermarsi anche come alternativa domestica al software commerciale. Sin dalla sua fondazione FreakNet ha lavorato per la diffusione delle libertà elettroniche e della cittadinanza digitale, sposando conseguentemente la causa del Free Software e il particolare quella della GNU/GPL, la licenza che prevede che i diritti sul software sono di chi lo usa e non di chi lo vende. La consapevolezza dell’impotanza delle questioni informatiche non può essere piena senza un’adeguata conoscenza degli aspetti tecnici. Per questo insieme oltre ai corsi di base si sono organizzati corsi di amministrazione, seminari specifici e altre iniziative. Nessuna di queste comunque esauriente fino in fondo: l’unico modo per arrivare alla conoscenza è intraprendere una propria strada. Questo manuale non vi instraderà sul sentiero, ma si limeterà a darvi un orientamento e un buon pai di scarpe.

Lo staff del Freaknet Medialab

Indice

Parte I

Prima dell'inizio

Capitolo 1

Le parti meccaniche di un computer

1.1 L'hardware

Tutte le parti del computer che potete effettivamente toccare con mano costituiscono l'hardware, cioè (letteralmente) la ferraglia. Tutto quello che invece non si può toccare viene chiamato software.

1.1.1 Cosa c'è in periferia

Tutte le cose attaccate con dei cavi allo scatolone centrale del vostro computer vengono dette periferiche. Tra le periferiche più comuni ci sono i monitor, le stampanti, i mouse, le tastiere, i modem, gli scanner, le casse acustiche e tante altre cose che non basterebbero un paio di queste pagine per elencarle tutte.

Il monitor

Il monitor è quella cosa del computer che più somiglia al vostro televisore. I computer non usano dei comuni televisori perché questi a confronto sono di pessima qualità e non vanno bene per visualizzare immagini fisse. Su di esso è possibile visualizzare e leggere un sacco di informazioni utili nel corso del vostro lavoro. In effetti il vostro computer vi parla per mezzo del monitor.

La stampante

La stampante vi fa una copia su carta di quello che vi serve. È utile soprattutto quando ci si imbatte in testi molto lunghi: è molto più riposante leggere dalla carta che da un monitor. Pensate che una volta, quando non c'erano ancora dei buoni monitor, le stampanti venivano usate al posto di questi.

Il modem

Il modem (MOdulator DEModulator) è la magica scatoletta che vi permette di non tenere il vostro computer di casa isolato dal mondo. Non fa altro che trasformare il modo con cui i computer parlano tra di loro nei modi in cui parlano gli umani, cioè sulle frequenze che possono passare per una normale linea del telefono.

La tastiera e il mouse

La tastiera è quella parte del computer che somiglia tanto a una macchina da scrivere. Il mouse invece a noi non ricorda proprio niente. Sono i principali dispositivi con cui informerete il computer delle vostre intenzioni. A essi verranno dedicate delle lezioni apposta. Quindi pazientate.

Altre periferiche

I progettisti dell'hardware sono gente molto fantasiosa e hanno attaccato le cose più strane ai loro computer e persino qualche cosa utile. Le periferiche elencate fin qui le trovate più o meno dappertutto. Se vi imbattete in qualcosa di veramente strano, come macchine del caffè o telecamere nascoste, chiedete al vostro guru.

1.1.2 L'ombelico del computer

Il microprocessore

Il centro assoluto del vostro computer è costituito dal microprocessore, detto anche CPU (Central Processing Unit) da chi ama confondervi con acronimi da tre lettere. Addirittura qualcuno lo chiama persino μP, ma voi non fateci troppo caso. Il microprocessore è quel grosso insetto nero di forma quadrata con un centinaio di piedi che fa i conti. In realtà un computer non è in grado di fare molto altro all'infuori di somme, sottrazioni e poche altre cose che sa fare anche la calcolatrice che avete trovato nel detersivo: certo, l'effetto finale è abbastanza diverso. Molto probabilmente il microprocessore che è presente nel vostro computer appartiene a uno di quelli della tabella qui sotto. Questa tabella vuole dare un'idea delle differenze di velocità tra i diversi modelli di microprocessori di uso domestico.

Processore	
80386	molto lento
80486	lento
Pentium	normale
Celeron	veloce
Pentirum II	molto veloce
Pentium III	velocissimo
Pentium IV	wow!

La scheda madre

Un microprocessore da solo non vi servirebbe a molto. Viene installato su una *scheda madre* (per i puristi *motherboard*) che ospita molte altre cose che vi sono utili almeno quanto il processore, come le schede per il monitor, per l'audio e per la rete.

1.1.3 Le memorie

La memoria volatile

Sulla scheda madre trova per esempio posto la RAM. La RAM (Random Access Memory) è la memoria ad accesso arbitrario, che non vuol dire che vi si leggano e scrivano i dati a casaccio, ma solo che il microprocessore può sfogliarla partendo direttamente dal punto che gli interessa, senza doversela leggere tutta dall'inizio ogni volta. La RAM è la memoria di lavoro, cioè il luogo dove vengono contenuti i vostri programmi quando stanno girando e dove vengono custoditi i dati di questi programmi. Una parte di questa memoria è riservata per cose che riguardano solo il vostro computer (non vale neanche la pena andarci a curiosare). La RAM si svuota ogni volta che il computer viene spento: questo non la rende molto affidabile per conservare le vostre cose. L'unità di misura attuale della Ram in commercio è il MegaByte (MB). Solo per darvi un'idea: i computer oggi vengono venduti solitamente con 256 MB di Ram, il che vuol dire che potreste tenerci sopra 484 copie della Divina Commedia. I due computer a bordo dell'Apollo 13 (la missione spaziale che arrivò sulla Luna) avevano mezzo MB di Ram a testa (che fa all'incirca una sola Divina Commedia per ciascuno).

Le memorie di massa

Se volete ritrovare le vostre cose la prossima volta che accendete il computer, sarebbe utile memorizzarle su un supporto che non perda i vostri dati quando viene tolta la corrente. Per questi scopi vengono usati di solito i dischi. I dischi appartengono alla categoria delle memorie di massa, cioè quel tipo di memoria che può tenere ingenti quantità di dati. I dischi si dividono in due categorie: *removibili* e *fissi*. I primi potete portarveli a spasso, gli altri è meglio lasciarli dove li avete trovati.

I dischi da passeggio

Alcuni removibili li avrete spesso chiamati *floppy*. Ne esistono di varie dimensioni e di varie capacità, a secondo dell'età. L'unica cosa che i dischi hanno in comune è l'essere pressappoco quadrati!

Altri dischi removibili comunemente usati sono i CD-ROM. Fisicamente identici a quelli musicali, possono contenere fino a 650 MB, cioè quanto 450 dischetti della quarta generazione! Dai Cd-rom potete solo leggere i dati che contengono, mentre per tutti gli altri dischi vi è spesso consentita anche la scrittura.

I dischi pantofolai

Che forma abbiano invece i *dischi fissi* (o *rigidi*, o *hard disk*) di solito non vi è dato saperlo, perché sono chiusi nello scatolone del computer e anche se lo aprite troverete una scatoletta rivestita di alluminio e nulla di più. I dischi rigidi attuali hanno dimensione dell'ordine dei Giga Bytes (cioè più di mille Mega Bytes). È su questo supporto che ordinariamente conserverete la vostra giornata di lavoro (sempre se siete interessati a ritrovarla in seguito).

IDE o SCSI?

I guru quando parlano di dischi rigidi e cd-rom fanno distinzione tra IDE o SCSI. A voi la questione non interesserà più di tanto. Tutto quello che può essere utile sapere è che gli SCSI sono più affidabili, ma anche più costosi, tanto che a volte possono non essere particolarmente convenienti.

Capitolo 2

Buongiorno UNIX!

2.1 La tastiera

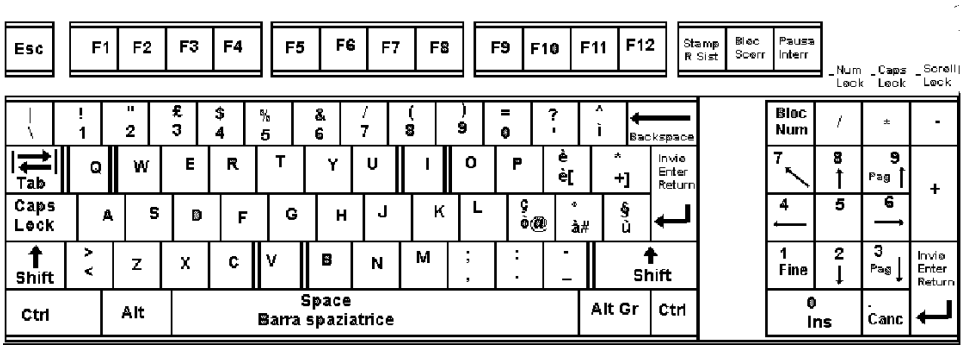

Figura 2.1: Il tipico lay-out della tastiera italiana (si riconosce dalla presenza delle vocali accentate).

La tastiera è il principale dispositivo di ingresso del computer. Quando volete richiamare la sua attenzione o dirgli qualcosa, la tastiera rappresenta un comodo aiuto. Di tastiere ne esistono di tutte le forme e dimensioni, anche molto stravaganti come quelle ergonomiche (specie di tastiere che sembrano spaccate a metà come se avessero ricevuto un colpo ben assestato di mazzuolo). I terminali più vecchi spesso hanno tasti con nomi incomprensibili che non userete mai.

La parte alfanumerica

Tutte (o quasi) hanno in comune la parte alfanumerica, quella che più ricorda la macchina da scrivere.

- A differenza delle macchine da scrivere, per andare a capo non si tira la leva, ma si preme *Invio* (o *Enter* o *Return*, qualche secolo fa *New Line*). Invio va premuto alla fine di ogni comando e dopo aver digitato il proprio nome o la password quando si inizia un colloquio.
- Nascosti in questa parte centrale ci sono alcuni tasti che su una macchina da scrivere non starebbero mai anche se tentasse di darsi delle arie. Questi sono: *Shift* o ⇑: se lo tenete premuto compariranno le lettere MAIUSCOLE. Shift serve anche a visualizzare i caratteri che stanno in alto nei tasti che hanno due segni (senza lo shift vedrete solo quelli in basso).

- *Caps Lock* o icona di un lucchetto o di una chiave: TIENE SEMPRE INSERITO IL MAIUSCOLO. Se caps lock è attivo una lucetta verde ve lo indica da qualche parte, solitamente sulla parte destra della tastiera. Attenzione: MAI, ma proprio MAI tenere caps lock attivo quando si vuole aprire un colloquio (per i guru fare il login) o si vuole lanciare un comando!

- *Control* o *Ctrl*: forse c'avete fatto già l'amore, ma qui sulla tastiera serve ad altro; di solito per richiamare qualche particolare funzione. Control, proprio come quell'altro, non si usa mai da solo, ma in due (cioè insieme ad un altro tasto, di solito un carattere alfabetico): quando vi si dice di fare qualcosa come *ctrl+a*, o *C-a*, o *^a* dovrete in tutti e tre i casi tenere premuto il tasto control e lasciarlo solo dopo avere battuto A.

- *Alt* o *Meta*: anche alt come control si usa in due, ma non esiste la versione in lattice. Per molti Unix alt è un tasto che può semplificarvi tanto il lavoro: se siete direttamente sul computer (non su un terminale!) alt in combinazione con un tasto funzione (calma ci arriviamo presto!) può darvi un'altra possibilità di entrare sul sistema. Anche in questo caso Alt+a, Meta-a e M-a, vogliono dire che bisogna tenere premuto alt e rilasciarlo dopo aver battuto A (ormai dovreste esservici abituati).

- *Alt Gr* o *Alt Graph*: qualcuno dei tasti ha ben tre caratteri. Tenendo premuto alt gr sceglierete quello in basso a destra.

- *Tab* o *Tabulazione* o *due frecce in verso opposto*: alcune macchine da scrivere ce l'hanno, ma non fatevi confondere: qui serve ad altro. Può essere utile per completare il nome di un comando o di un file.

- *Backspace* o *freccia verso sinistra*: cancella il carattere che sta a sinistra del vostro cursore (la barretta lampeggiante) sullo schermo.

Il tastierino numerico

Sulla parte destra della tastiera (spesso) c'è un'insieme di tasti che ricorda tanto da vicino la vostra calcolatrice da tavolo. Gli esperti lo chiamano tastierino numerico.

Bloc num "on"

Ovviamente non funzionerà mai come vi aspettate se non glielo chiedete esplicitamente. Per prima cosa dovrete accenderlo. Provate a battere bloc num: se la sua lucetta si spegne allora era già abilitato. Niente paura: ripremete bloc num per far tornare la luce. Ora tutto funziona come la vostra calcolatrice, cioè i numeri fanno i numeri e gli operatori fanno gli operatori.

Bloc num "off"

A luce spenta il tastierino fa delle altre cose:

- *canc* o *del*: cancella il carattere alla destra del cursore. Approfittiamo di quest'occasione per avvisarvi che quasi mai da un terminale canc si comporta da canc e backspace da backspace. Spesso sono invertiti, mentre delle volte funziona l'uno e non l'altro o peggio uno non funziona e l'altro fa a rovescio. Fate le vostre prove fino a quando riuscite a cancellare quello che avete sbagliato.

- *ins*: cambia tra la modalità inserimento e sovrascrittura. Non funziona quasi mai.

- *home* o *freccia che va verso l'alto a sinistra*: vi porta allinizio della riga. Funziona abbastanza spesso se l'amministratore del vostro computer è in gamba.

- *fine* o *end*: come home, ma va alla fine della riga.

- *pag⇑*o *page up* (tasto 9 del tastierino numerico): alcuni programmi che visualizzano testo usano questo carattere per permettervi di tornare alla pagina precedente.
- *pag⇑* o *page down* (tasto 3 del tastierino numerico): l'omologo di page up ma nel verso opposto.
- 8 (⇑), 2 (⇓), 4 (⇐), 6 (⇒): delle volte il vostro cursore si sposta nella direzione della freccia.

Molte tastiere hanno una parte intermedia tra il blocco alfanumerico e il tastierino numerico che contiene tutte le funzioni non numeriche del tastierino. Bloc num NON ha effetto su questo gruppo di tasti.

La fila superiore

Sulla parte alta della tastiera c'è una fila di bottoni con F e un numero. Li chiamano *tasti funzione* (come promesso ci siamo arrivati) e non sono molto utili se non per cambiare schermo. Alla loro sinistra c'è *esc*, che sta per escape. Delle volte i tasti funzione li trovate in altre parti della tastiera, ma non stenterete a riconoscerli. In ultimo bisognerebbe accennare anche ai tre tastini che completano la fila (*stamp*, *bloc scorr*, *pausa*), ma delle volte neanche i guru sanno esattamente a che servono**.**

Altri tasti utili

C'è qualche altro simbolo di cui è utile conoscere il nome:

- /: sta sopra il 7 nella parte alfanumerica e c'è anche nel tastierino. Si chiama *slash*.
- \: come il primo, però nel senso opposto. Infatti si chiama *backslash*.
- |: pipe, cioè tubo. Condivide il tasto con backslash.
- _: underscore, trattino basso. Nello stesso tasto del *dash*, cioè -
- Nello stesso tasto del dash, cioè *&: ampersand, e commerciale.* Sopra il 6.

2.2 Aprire un colloquio (il login)

Qualche volta si vorrebbe cominciare a lavorare ma lo schermo del computer è assolutamente nero. Ciò può avere due spiegazioni:

1. Il monitor è spento. Accendetelo e tutto andrà per il meglio.
2. Un salva schermo (programma che riduce il logorio dei monitor) è attivo.

Risvegliate il sistema dal suo torpore premendo un tasto qualsiasi.

Succede anche, da qualche vecchio terminale, che possa essere necessario battere qualche C-d (o ^d, ricordate?) per attirare l'attenzione del computer centrale. In tutti i casi ci si troverà davanti a una schermata del tipo:

```
Minix Release 2.0 Version 2

smarterm login: |
```

A questo punto ci si presenta rispondendo a login con il nome che si è scelto (o che è stato imposto) per l'accesso al computer. In effetti anche il computer si è presentato dichiarando il proprio nome (smarterm) sulla stessa riga del login. Prima del login il computer ci aveva dato anche una serie di informazioni poco utili come la versione del suo sistema operativo e altre cose per addetti ai lavori.

Facciamo il caso dell'utente luther (ricordate di scrivere il vostro nome sempre in minuscolo, visto che quasi sicuramente vi è stato dato in questa forma).

Prima di andare avanti: notate che Unix è molto permaloso e farà finta di non capire se provate a scrivere un comando o un nome con qualche lettera maiuscola dove in effetti non c'è. Per lui luther, Luther e LUTHER (ma anche lutheR o lUthEr) sono persone diverse.

```
Minix Release 2.0 Version 2

smarterm login: luther|
```

Fin qui luther non ha concluso niente di buono, perché il computer non è ancora stato informato del suo arrivo. Per farlo bisogna premere il tasto invio.

```
Minix Release 2.0 Version 2

smarterm login: luther
password:|
```

Ok. Il computer si è accorto dell'arrivo di luther e vuole vedere se si tratta davvero di luther o di qualcuno che si spaccia per luther. Ecco perché gli chiede una parola d'ordine (*password*).

```
Minix Release 2.0 Version 2

smarterm login: luther
password:
Terminal type? (network) vt100
$
```

Dopo aver digitato la sua password, luther ha finalmente cominciato un colloquio con il computer. Qui è successo che il sistema gli ha posto una domanda imbarazzante alla quale non avremmo voluto rispondere neanche noi. Voi non preoccupatevi perché non ve la chiederà mai nessuno.

Le password

Quando scegliete una password, ricordate che essa non deve essere facilmente rintracciabile. Sono da escludere tutti i nomi di fidanzate/i, figli/e, nonni/e, animali domestici, date di nascita e tutto quello che può essere facilmente messo in relazione con voi. Ricordate anche che certi ragazzacci usano dei dizionari per trovare la vostra password e che questi dizionari contengono molti più lemmi dei normali Devoto-Oli, come nomi propri, ecc. Una buona password è un misto di caratteri minuscoli, MAIUSCOLI e numeri. Ricordate anche che è una buona norma avere password sempre differenti per ogni computer su cui si ha accesso, perché altrimenti se qualcuno scopre la vostra password può spacciarsi per voi su tutti quei computer. Chi entra in modo fraudolento può cancellare tutto il vostro lavoro o fare incazzare qualcuno (per esempio mandandogli delle lettere poco carine) facendo ricadere la colpa su di voi. È una buona abitudine cambiare la propria password di frequente. Per farlo date in pasto al *prompt* il comando *passwd* e prestate attenzione a quello che vi richiederà il computer.

2.3 Chiudere il colloquio

Visto che per ora non sapete fare altro, una volta dentro digitate *exit* per chiudere il colloquio con Unix.

Parte II

Alcuni concetti basilari su programmi (comandi), file e directory

Capitolo 3

Gli amici dello scrittore (gli editor)

3.1 Cosa vi serve per fare letteratura

Per adesso vi siete iscritti con entusiasmo a un corso di computer. Ma mettiamo il caso che vi venga il pallino di buttarvi nella narrativa. Potreste conciliare le due cose? Probabilmente sì, se conosceste un programma in grado di farvi scrivere qualcosa di nuovo e di modificarlo in seguito.

3.1.1 Un po' di storia. *Ed* il «nonno» degli editors UNIX

Quando nacque Unix c'era un solo programma in grado di svolgere questo compito. I programmi che servono a comporre del testo si chiamano *editor*.

Indovinate un po' come venne chiamato quest'unico programma a quel tempo in cui andavano di moda nomi da due lettere?

Sì, bravi! semplicemente *ed*!

Su qualsiasi computer ne troverete una versione ma solo perché, come scrive Andrew S. Tanenbaum:

> «*ed* è fornito per il suo valore affettivo».

Molto difficilmente vi capiterà di imbattervi in un computer dove *ed* sia l'unico editor a disposizione. Se sarete così fortunati protestate energicamente con l'amministratore di sistema. L'uso di `ed` è così rigido e inamichevole che non proveremo neanche a illustrarvelo.

3.1.2 Rock'n Roll e solo testo: vi.

Uno standard di fatto è il programma *vi*. Se non c'è proprio *vi* in persona (che è protetto da astrusi copyright) troverete almeno il suo clone: il grande *elvis* (che comunque potrete invocare sempre come *vi* dalla riga di comando), oppure altri cloni come *vim* (*vi* improved).

Se utilizzate, ad esempio, una distribuzione Slackware provate ad eseguire i comandi

```
$ vi prova
```

Digitate :q e premere invio per uscire.

```
$ elvis prova
```

Fate lo stesso per uscire.

```
$ vim prova
```

Digitate ancora :q e invio per uscire!

Sorpresi? Dovete sapere che tutte le volte che invocate il comando *vi*, in realtà, non fate altro che chiamare *elvis, vim* o un altro clone. Questi cloni, oltre che implementare tutti i comandi tipici di *vi*, ne aggiungono di altri, insieme a modalità aggiuntive e feature varie.

vi non è esattamente uno di quei programmi amichevoli come li intendiamo oggi, anche se il suo uso è molto più semplice rispetto a *ed*.

Il modo più comodo è di lanciarlo specificando il file da editare direttamente sulla riga di comando (questo metodo funziona per qualsiasi altro editor e diamo per scontato che d'ora in avanti farete sempre così).

Ad esempio per editare il file "appunti", invieremo il comando:

```
$ vi appunti
```

Se il file appunti non esiste, *vi* ve lo comunicherà attraverso l'ultima riga in basso dello schermo. All'inizio di ogni riga visibile sul monitor andrà a piazzare una ~ per dirvi che la riga è vuota. Se invece il vostro appunti contiene già qualcosa, *vi* in basso vi informerà su quanti caratteri e quante righe compongono il testo. Il contenuto del file sarà visualizzato piazzando l'inizio sulla prima riga dello schermo. Se tutto il file è più corto dello schermo stesso, le righe mancanti verranno anchesse visualizzate con una ~ all'inizio.

Adesso *vi* è pronto ad accettare i vostri comandi. Se volete inserire del testo, dovete prima portare il cursore dove intendete iniziare a scrivere e battere la *i*. A questo punto potete cominciare.

3.1.2.1 Un programma, diversi stili di lavoro.

L'editor *vi* ha due diversi modi operativi: la *modalità di inserimento* (*input*) e la *modalità di comando* (*command*). Da quest'ultima si può passare a una sorta di sottomodalità in cui è possibile visualizzare l'inserimento dei comandi sull'ultima riga dello schermo, detta *last line* (questi comandi vanno finiti premendo il tasto di invio).

Ecco qui uno schema che vi sarà utile per meglio comprendere e per effettuare i vostri esperimenti con maggiore consapevolezza.

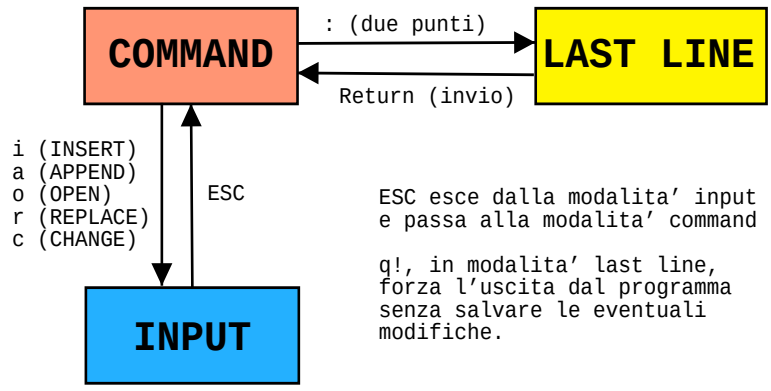

Figura 3.1: Modalità di vi e loro relazioni

Nella pagina succesiva è riportata una tabella con i comandi più utilizzati ed importanti.

Comando	**Descrizione**
(invio)	Cominciare una nuova riga (solo in *modalità inserimento*)
(esc)	Per tornare alla *modalità comando*
+	Andare all'inizio della prossima riga
-	Tornare all'inizio della riga sopra
a	*Inserire testo dopo il cursore*
A	*Inserire testo alla fine della riga*
dd	Cancellare l'intera riga attuale
D	Cancellare dal cursore alla fine della riga
G	Andare alla fine del file
lG	Andare all'inizio del file
h	Spostare il cursore a sinistra
i	*Aggiungere testo a sinistra del cursore*
j	Andare alla riga sotto
k	Andare alla riga sopra
l	Spostare il cursore a destra
O	*Aggiungere testo su una nuova riga prima dell'attuale*
:q! (invio)	Chiudere il programma senza salvare
R	*Scrivere in modalità* sovrascrittura
u	Annullare l'ultima modifica (*undo*)
U	Annullare tutte le modifiche al file
x	Cancellare il carattere sul cursore
:w (invio)	Salvare
ZZ	Chiudere il programma salvando i cambiamenti

Tabella 3.1: In corsivo i comandi che fanno passare vi dalla modalità comando a quella inserimento

Alla partenza si pone automaticamente nella prima modalità (*command*) e aspetta pazientemente che gli diciate cosa fare. Buona parte delle cose possibili sono elencate nella tabella qui sopra. Forse la vostra versione ha qualche comando in più, ma quelli della tabella funzionano sempre.

Comandi come *a, A, i, O, R,* lo fanno passare al modo inserimento: cioè vi permettono finalmente di aggiungere e/o modificare quello che vi serve. Per tornare a dare i comandi (indispensabile anche solo per spostare il cursore), basta premere il tasto *esc*.

Oltre ai comandi *h, j, k, l,* alcuni computer di epoca pressappoco contemporanea accettano l'uso dei tasti con le frecce per spostare il cursore, sempre se vi trovate in modalità comando.

I due punti (*:*) sono quelli che vi fanno digitare il comando sull'ultima riga (*last line*).

Quando non ricordate più in quale modalità state operando (*vi* è timido e non vi dice niente), battete *esc* e ricominciate (tanto esc non ha effetto quando siete già in modalità *command*).

3.1.2.2 Prima di chiamare aiuto in caso di emergenza

Se avete fatto molto, ma molto casino, tanto da non bastare il comando U che dovrebbe riportarvi il file nelle condizioni in cui l'avete pescato, non entrate nel panico ma cominciate a battere *esc* fino a quando il computer arrabbiato vi apostrofa con un sonoro beep; a questo punto la sequenza *:q!* completata dall'invio vi toglierà dai guai facendovi riguadagnare il prompt della shell. Alla prossima ritroverete il vostro file esattamente comera l'ultima volta che l'avete salvato.

3.2 Non solo editing

Sono veramente pochi gli elaboratori Unix che già all'installazione possiedono emacs. Ma grazie al fatto che la sua versione più potente, il *GNU emacs*, è assolutamente gratuita, emacs è presente su qualsiasi computer (talvolta chiamato anche *gmacs*) che abbia una discreta capacità di calcolo.

Come editor è molto più facile da usare di *vi*, seppure i suoi comandi non siano intuitivi.

Ma il *GNU emacs* non è solo un editor: conosciamo gente capace di usarlo per leggersi la posta elettronica, giocare a solitario, calcolare le fasi lunari e portare a passeggio il cane. Fa anche un ottimo espresso.

3.2.1 Un anticipo sulla storia di ...

... *Richard Stallman*, il suo creatore, avremo modo di parlarne in seguito (è veramente un tipo interessante!).

Qui diremo solo che *emacs* (che sta per editor macro) in origine era solo un'appendice a un altro editor chiamato *teco* (non provate a lanciarlo, non risponderà: per fortuna si è estinto grazie alla sua avversione nei confronti dell'utente).

Poi è cresciuto ed è andato avanti da solo.

3.2.2 Scrivete dove vi pare

La grande novità che vi porta *emacs* è che non ci sono due modalità distinte, ma stavolta cominciando a scrivere automagicamente il testo compare sul monitor esattamente dove avete posizionato il cursore.

Vi pare poco? Dentro emacs tutto si svolge in maniera abbastanza semplice: questi sono i principali comandi per funzioni diverse dallo scrivere.

Comando	Descrizione
C-a	Andare all'inizio della riga
C-b	Un passo a sinistra
C-d	Cancellare un carattere
C-e	Andare alla fine della riga
C-f	Un passo a destra
C-k	Cancellare fino a fine riga
C-n	Un passo verso giu'
C-p	Un passo verso l'alto
C-x C-c	Uscita dal programma
C-x C-s	Salvare il file
M-<	Andare all'inizio del file
M->	Andare alla fine del file
M-d	Cancellare fino alla fine della parola

3.3 Il vostro amico Joe!

Sugli Unix più evoluti (ma purtroppo solo su quelli), il perfetto equilibrio fra facilità di funzionamento e potenza si è raggiunto (a nostro avviso) con *joe*.

Questo editor è il più indicato specie per chi ha avuto esperienza con il vecchio Word Star dell'ambiente Dos.

Addirittura se invocato come *jstar* emula questo programma anche nell'aspetto, oltre che nelle combinazioni di tasti per i comandi. Invece lanciando proprio *joe* avrete un editor a pieno schermo.

Sporadicamente l'ultima riga verrà utilizzata come riga di stato, mentre in alto vi si dice quale file avete per le mani e vi si ricorda che per ottenere l'elenco delle combinazioni di tasti utili bisogna premere *Ctrl-K*. Se non vi disturba troppo potete lasciare visualizzato l'elenco e scorrerlo con *Meta-.* in avanti e *Meta-,* indietro per trovare il comando che cercate. Le combinazioni fondamentali di joe sono queste due:

- Ctrl-KD per salvare il file sul quale state lavorando;

- Ctrl-C per chiudere il programma.

Le altre, prese da le prime due schermate dell'aiuto, sono queste:

Cursore		Spostare	Selezione	Cancellare	Varie	Uscire
^B sinistra	^F destra	^U indietro di una pagina	^KB inizio	^D un carattere	^KJ reimpagina	^KX salva e esci
^P su	^N giù	^V avanti di una pagina	^KK fine	^Y tutta la riga	^T opzioni	^C esci e basta
^Z indietro di una parola		^A inizio riga	^KM sposta	^W fino alla fine della parola	^R rinfresca lo schermo	^KZ sospendi e torna alla shell
^X avanti di una parola		^E fine riga	^KC copia	^O fino all'inizio della parola	^@ inserisci	**File**
Ricerca		^KU inizio del file	^KW salva	^J fino alla fine della riga	**Controllo ortografico** *	^KE apri
^KF cerca una stringa		^KV fine del file	^KY cancella	^_ annulla	^[N sulla parola	^KR inserisci
^L prossima occorrenza		^KL vai alla riga numero...	^K/ filtra	^^ ripristina	^[L sull'intero file	^KD salva
Operazioni sulle finestre						
^KO Dividi la finestra in due				^KE Apri un file nella finestra		
^KG Allarga la finestra attiva				^KT Riduci la finestra attiva		
^KN Vai alla finestra inferiore				^KP Vai alla finestra superiore		
^C Chiudi la finestra attiva				^Kl Mostra tutte le finestre / Mostra un solo file		

Capitolo 4

Le operazioni con i file

La vera essenza di un computer è sepolta nelle profondità della macchina persa in milioni di righe di codice. E certamente non si preoccupa di capire cosa gli utenti vorrebbero fare. Questo lavoro è svolto da un altro programma: la shell, cioè l'interprete dei comandi. La shell è il primo programma che viene lanciato dopo il login, ed il suo compito è quello di tradurre quello che gli utenti gli dicono in ciò che il computer può capire. Solitamente dopo aver aperto il colloquio con la macchina si ci trova di fronte ad una schermata del tipo:

```
vostro_login_name@nome_del_computer:cartella/$
```

Ora siamo pronti a parlare con il computer...

4.1 Cominciamo... quali file abbiamo?

Per ottenere una lista dei vostri file digitate:

```
quest@medialab:~$ ls
```

Questo comando come molti altri in Unix deriva dall'inglese e precisamente dalla parola list, lista. È molto comune in Unix che il nome di un comando sia ricavato dal codice fiscale della corrispettiva parola inglese. Ma torniamo a *ls*. Il comando fa semplicemente vedere i nomi dei file in ordine alfabetico, ma se noi passiamo un parametro aggiuntivo al comando questo ci potrà dare qualche informazione in più. Proviamo a digitare:

```
quest@medialab:~$ ls -l
total 2778
drwxr-xr-x 2 quest staff 1024 Oct 23 22:45 News/
drwxr-xr-x 2 quest staff 1024 Nov 12 22:59 c/
-rw-r--r-- 1 quest staff 2820098 Oct 19 23:24 cpt_harlock.mp3
drwxr-xr-x 2 quest staff 1024 Nov 15 20:03 corsi/
-rw------- 1 quest users 1170 Nov 16 21:38 dead.letter
drwxr-xr-x 6 quest staff 1024 Nov 12 22:41 documenti/
drwx------ 2 quest staff 1024 Nov 16 21:39 mail/
drwx------ 2 quest staff 1024 Oct 19 22:53 nsmail/
drwxr-xr-x 2 quest staff 1024 Oct 17 23:17 public_html/
-rw-r--r-- 1 quest users 11 Nov 11 22:07 xenix
quest@medialab:~$
```

Come possiamo vedere ai nomi dei file si sono aggiunte altre incompresibili informazioni, che per il momento è bene tralasciare.

-l in gergo viene chiamata opzione. Le opzioni in Unix vengono messe subito dopo il nome del comando a cui si riferiscono e sono generalmente precedute dal simbolo - .

4.2 Copiare i file

È possibile fare dei duplicati esatti dei file che possediamo: copiarli. Per copiare un file si usa il programma *cp*, il cui nome deriva direttamente, come facile immaginare, dalla parola inglese copy. Copy, o, cp che dir si voglia funziona così:

```
$ cp file_da_copiare nome_della_copia
```

oppure, per i più smaliziati sarà più comprensibile la forma:

```
$ cp file_sorgente file_destinazione
```

Se per esempio ora noi volessimo fare una copia del file contente il testo del primo Canto della Divina Commedia, dovremmo scrivere qualcosa del tipo:

```
quest@medialab:~$ cp primo_canto.dv canto
```

Abbiamo così copiato il nostro primo file, ma che cosa succede se per caso esiste già un file che si chiama canto? Unix lo cancella sostituendolo con la copia di primo_canto_dv. È una bellissima e grandissima idea prima di fare operazioni di questo tipo controllare con un *ls* che non esista già un file con il nuovo nome che abbiamo scelto. Nella recenti versioni di Unix a *cp* si può chiedere di avvisarci se per caso esiste già un file con il nuovo nome che abbiamo scelto, questo si ottiene con l'opzione *-i*. Provatela:

```
quest@medialab:~$ cp -i primo_canto.dv canto
cp: overwrite ‘canto‘?
```

4.3 Scegliere dei bei nomi

Quando copiamo un file, o, più in generale quando creiamo un file, dobbiamo stare attenti al nome che gli diamo. Unix in tal senso ha delle regole ben precise: I nomi possono essere lunghi a piacere: non hanno nessuna limitazione di spazio, quindi per esempio potreste chiamare un file:

```
Alcune_cose_che_vorrei_digitare_se_vivrò_abbastanza_a_lungo
```

È un'ottima cosa non utilizzare caratteri strani come i segni di interpunzione, ottimi per le password, ma disastrosi per tutto il resto. Quindi state lontani dai questi caratteri:

/	!	@
#	$	^
&	*	+
(	)	‘
“	?	|

Non inserire spazi bianchi se il vostro nome è più lungo di una sola parola. In questi casi ci sono dei piccoli trucchi che possono fare al caso nostro:

- È molto comune utilizzare come separatore il carattere underscore _:

  ```
  Alcune_cose_che_vorrei_digitare_se_vivrò_abbastanza
  ```

- Ancora è possibile separare le parole scrivendo in MAIUSCOLO la prima lettera di ogni parola:

  ```
  AlcuneCoseCheVorreiDigitareSeVivròAbbastanza
  ```

4.4 Rimandare i file all'età della pietra

I file si possono anche cancellare utilizzando il comando *rm*, remove in inglese. *rm* funziona cosi:

```
$ rm nome_del_file_che_voglio_cancellare
```

Ora provate a cancellare il file che contiene la nostra copia del primo Canto della Divina Commedia.

```
quest@medialab:~$ rm canto
```

È possibile cancellare anche un lista di file basta scrivere qualcosa come:

```
$ rm nome_primo_file nome_secondo_file nome_terzo_file
```

Attenzione: una volta cancellato un file, anche per l'hacker più esperto sarebbe quasi impossibile recuperarlo. Quindi per essere sicuri di non cancellare file utili sarebbe una buona cosa utilizzare l'opzione *-i* del comando rm che così facendo ci chiederà, ogni volta che cancella un file, la conferma dell'operazione.

```
quest@medialab:~$ rm i canto
rm: remove 'canto'?
quest@medialab:~$
```

Bisogna sbarazzarsi dell'immondizia. Cancellare i file che non servono più è una abitudine che è ben prendere per almeno due motivi:

1. Avere molti file inutili crea confusione;

2. i file occupano spazio su disco e solitamente un buon amministratore di sistema ci chiederà di fare spazio cancellando tutto il surplus.

D'altro canto, è una buona idea creare copie extra dei file. Se per esempio stessimo lavorando su una relazione da tre settimane, facendone una copia in più al giorno, saremmo sicuri di poter recuperare una versione precedente della relazione qualora avessimo apportato una qualche stupida modifica nell'ultimo lavoro.

4.5 Cambiamo nome.

Avendo un dato file, può anche essere che magari una mattina ci alziamo e decidiamo di cambiare il suo nome, a tal proposito ci può essere di aiuto il comando *mv*, che vuol dire move, cioè sposta.

```
$ mv nome_file_nome_corrente nome_file_nome_nuovo
```

Provate a digitare quanto segue:

```
$ mv canto primo_canto.dv
```

Visto e considerato che non si possono avere due file con lo stesso nome, se un file ha già il nome che volete utilizzare, *mv* lo getta via perdendolo e sostituendolo con quello nuovo; ma magari questo non è proprio quello che volevate. Allora ancora una volta le opzioni vi vengono in soccorso. Utilizzando mv con l'opzione *-i*, vi verrà chiesta la conferma prima di effettuare la sostituzione.

4.6 Guardare dentro un file

Finora avete giocato a taglia e cuci con i file, ma non avete ancora visto cosa c'è dentro. Bisogna premettere che esistono quattro tipi di file:

1. File *ordinari*;
2. File *device* (cioè "finti" che si riferiscono ad una periferica);
3. Le *directory* (abbiate pazienza, le vedremo in seguito);
4. I *link simbolici* (abbiate ancora più pazienza);
5. *Tubi, spinotti o altra ferraglia* (che non vale la pena di conoscere).

Quelli che noi abbiamo trattato finora e di cui tratteremo fino alla fine del capitolo sono file ordinari, che a loro volta sono di due tipi fondamentali:

1. I *file di testo* (in inglese *text files*), che contengono solo caratteri alfanumerici e che Unix può visualizzare bene;
2. I *file binari* (in inglese *binary files*), che contengono codici speciali e che sembrano scritti da scimmie quando vengono visualizzati sullo schermo.

I file delle immagini, dei database, dei fogli di calcolo, della video scrittura e quasi tutto il resto fanno parte del secondo tipo. Per visualizzare un file di testo sullo schermo si utilizza il comando *cat*:

```
$ cat nome_file_da_visualizzare
```

Potreste decidere ora di dare un'occhiata al file del primo canto della Divina Commedia:

```
quest@medialab:~$ cat canto
```

Se invece provate ad utilizzare *cat* con un file di tipo immagine o di qualunque altro tipo vedremo scorrere sullo schermo il dialetto meridionale di un abitante di Marte ma provateci:

```
quest@medialab:~$ cat freaknet.gif
GIF87AVqgv1/a‘R Sq L@ W@p3Ybt>) 5{S[{8sft Gwe r@6rY%O9
q‘J?|b[cTDgX’o9qeF7kjhE%cG@a#erar |&+[G0b\DK{u:?,Wxt_Z
||+,(y xc{#D{R#-49 r>’cYs{c!bEv J ;% qqMy#yow]lYFuhhF‘
I @Ox%MA# Ux>v(u?u:}_v)rZ(ao-?f tyViuuqtF %pkxqkbBciJ(
Tdsct aB N %FYZ[@Kz^ 1q||I#‘O{pe4FwKf<Pr>X‘n A)c6]08k‘
s?iI#F‘bUpG(QP8rRi|pM|^tamrp>rpI(OR@-yQop|~_NJX!=r9^uq
$$!SKoo<cAc%aHCU\oBvu
quest@medialab:~$
```

Il programma *cat* nonostante la sua bontà ha una pecca, se il file che vorremmo vedere è troppo lungo ci scorre via davanti agli occhi senza che noi possiamo leggere niente. Per vedere solo le prime dieci righe di un file c'è il comando *head*. Proviamo a leggere le prime dieci righe del primo Canto della Divina Commedia:

```
quest@medialab:~$ head canto
LA DIVINA COMMEDIA
di Dante Alighieri INFERNO




<B>Inferno: Canto I</B>


Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura
```

Se al contrario invece volessimo sapere che fine hanno fatto Dante e Virgilio alla fine del primo Canto dovremmo usare il comando *tail*.

```
quest@medialab:~$ tail canto
tal era io a quella vista nova:
veder voleva come si convenne
l’imago al cerchio e come vi s’indova;
ma non eran da ci le proprie penne:
se non che la mia mente fu percossa
da un fulgore in che sua voglia venne
A l’alta fantasia qui manc possa;
ma già volgeva il mio disio e ’l <I>velle</I>,
s come rota ch’igualmente mossa,
l’amor che move il sole e l’altre stelle.
```

Ora se invece volessimo vedere, finalmente, tutto il contenuto di un file di testo e scorrerlo con calma, dovremmo usare i comandi *more* e *less*. Proviamo a digitare:

```
quest@medialab$ more canto
LA DIVINA COMMEDIA
di Dante Alighieri INFERNO
```

```
<B>Inferno: Canto I</B>


Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura
che la diritta via era smarrita.
Ahi quanto a dir qual era cosa dura
esta selva selvaggia e aspra e forte
che nel pensier rinova la paura!
Tant' amara che poco più morte;
ma per trattar del ben ch'ì vi trovai,
dir de l'altre cose ch'ì v'ho scorte.
Io non so ben ridir com'ì v'intrai,
tant'era pien di sonno a quel punto
che la verace via abbandonai.
Ma poi ch'ì fui al più d'un colle giunto,
là dove terminava quella valle
--More--(0%)
```

come possiamo vedere in fondo alla pagina c'è un indice che ci dice quale percentuale abbiamo visto fin'ora del documento che stiamo leggendo, nel nostra caso (0%). Con *more* il testo si scorre premendo in tasto invio. Proviamo ora ad usare *less*.

```
quest@medialab$ less canto
LA DIVINA COMMEDIA
di Dante Alighieri INFERNO



<B>Inferno: Canto I</B>


Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura
chÈ la diritta via era smarrita.
Ahi quanto a dir qual era cosa dura
esta selva selvaggia e aspra e forte
che nel pensier rinova la paura!
Tant' amara che poco pi morte;
ma per trattar del ben ch'ì vi trovai,
dir de l'altre cose ch'ì v'ho scorte.
Io non so ben ridir com'ì v'intrai,
tant'era pien di sonno a quel punto
che la verace via abbandonai.
Ma poi ch'ì fui al pi d'un colle giunto,
l dove terminava quella valle
canto 1/14753 0%
```

il risultato è più o meno identico; cambia l'indice in basso, che sta volta ci indica pure a che riga stiamo leggendo. In *less* il testo si scorre utilizzando i tasti cursore.

Ora che sappiamo come leggere un file vorremmo anche cercarci dentro le cose che ci interessano, magari vorremmo sapere tutte le righe del primo canto della Divina commedia in cui compare la parola Dio.

A questo scopo ci può aiutare il commando *grep*:

```
$ grep parola_che_vogliamo_cercare file_in_cui_cercare
```

per esempio:

```
quest@medialab:~$ grep dio canto
vagliami lì lungo studio e l grande amoren fiamma d'esto
incendio non m'assale.che nvidiosi son d'ogne altra sorte.da
fastidiosi vermi era ricolto.Ovidio l tergo, e l'ultimo
Lucano.che con misura nullo spendio ferci.portando dentro
accidioso fummo: D'ogne malitia, ch'odio in cielo
acquista,quelle fiere selvagge che n odio hannolo ncendio e
giace dispettoso e torto,gent' avara, invidiosa e superb:del
misero Sabello e di Nasidio, Taccia di Cadmo e d'Aretusa
Ovidio;converte poetando, io non lo nvidio;
```

Come previsto *grep* vi ha mostrato tutte le righe in cui lui ha trovato la stringa ***dio***. Attenti all'uso di *grep*, perché il programma ha trovato anche i versi in cui dio non è una parola, ma solo parte di una parola più grande, come o**DIO** o invi**DIO**so.

Ora se noi avessimo una stampante potremmo anche stampare, basta usage il commando *lpr*:

```
$ lpr nome_file_da_stampare
```

quindi volendo stampare il primo Canto della Divina Commedia dovrete digitare:

```
quest@medialab:~$ lpr canto
quest@medialab:~$
```

Capitolo 5

Anche il file discende dall'albero

5.1 Le directory

Ogni volta che avrete lavorato sul computer, avete registrato il vostro lavoro su uno o più file (vi siete ricordati di salvare, vero? :). In breve tempo, quando digiterete il comando *ls* non riuscirete più a farvi bastare lo schermo per visualizzare l'elenco di tutti i files che avete generato.

```
$ ls
budget_agosto iano.jpg lettera_sindac rapporto_sette
budget_aprile lettera_alfred mara.jpg relazione
budget_dicembr lettera_banca pippo.jpg riepilogo_gene
budget_febbrai lettera_carla prova.tar.gz siti_belli
budget_gennaio lettera_cetty rapporto_agost siti_computer
budget_giugno lettera_direzi rapporto_april siti_intranet
budget_luglio lettera_giovan rapporto_dicem siti_lavoro
budget_maggio lettera_inail rapporto_febbr siti_trash
budget_marzo lettera_inps rapporto_genna spese
budget_novembr lettera_lele rapporto_giugn torta_mele
budget_ottobre lettera_luigi rapporto_lugli torta_ricotta
budget_settemb lettera_patriz rapporto_maggi turi.jpg
budino_cioccol lettera_rita rapporto_marzo budino_vanigli
lettera_robert rapporto_novem carmelo.jpg lettera_sergio
rapporto_ottob
$
```

L'utente luther ha conservato insieme le foto dei suoi amici (tutti quei file il cui nome finisce per .jpg), le ricette per i budini e le torte, le lettere personali e di lavoro, i bilanci, le relazioni e altri appunti.

Il disordine regna sovrano tra i suoi file. Per riorganizzare queste cose seguendo un ordine logico, seppure arbitrario, esistono le *directory*, letteralmente *elenchi*.

5.2 Creare un elenco/directory

Si può cominciare a mettere ordine creando la directory per le foto degli amici.

```
$ mkdir foto
$ ls
budget_agosto foto/ lettera_sergio rapporto_ottob
```

```
budget_aprile iano.jpg lettera_sindac rapporto_sette
budget_dicembr lettera_alfred mara.jpg relazione
budget_febbrai lettera_banca pippo.jpg riepilogo_gene
budget_gennaio lettera_carla prova.tar.gz siti_belli
budget_giugno lettera_cetty rapporto_agost siti_computer
budget_luglio lettera_direzi rapporto_april siti_intranet
budget_maggio lettera_giovan rapporto_dicem siti_lavoro
budget_marzo lettera_inail rapporto_febbr siti_trash
budget_novembr lettera_inps rapporto_genna spese
budget_ottobre lettera_lele rapporto_giugn torta_mele
budget_settemb lettera_luigi rapporto_lugli torta_ricotta
budino_cioccol lettera_patriz rapporto_maggi turi.jpg
budino_vanigli lettera_rita rapporto_marzo carmelo.jpg
lettera_robert rapporto_novem
$
```

Per creare la nuova directory che ospiterà le immagini è stato usato il comando *mkdir* seguito dal nome del nuovo elenco. Dopo un nuovo *ls*, tra i file di luther sarà apparso anche *foto/*. Molti computer hanno la buona abitudine di far capire subito di cosa si tratta facendo apparire uno slash alla fine del nome di una directory. A questo punto si può provare a spostare uno dei file all'interno della nuova directory.

```
$ mv turi.jpg foto/turi.jpg
```

Il comando mv non serve solo a cambiare il nome di un file, ma viene impiegato anche per spostarlo da una directory a un'altra. Di fatto funziona allo stesso modo. Per spostare un file dovete però indicare il *percorso* (in inglese *path*) della directory di destinazione, poi porre uno slash e infine mettere il nome (il tutto senza spazi e rispettando sempre MAIUSCOLE e minuscole).

```
$ mv carmelo.jpg foto/melo.jpg
```

Con questo comando l'immagine carmelo.jpg è stata spostata nella directory foto cambiandone contemporaneamente il nome in melo.jpg.

Se invece non viene specificato il nome per la destinazione del file, il computer assume che intendete chiamarlo allo stesso modo.

```
$ mv iano.jpg foto
```

Dopo questo comando iano.jpg sarà stato spostato nella directory foto mantenendo lo stesso nome. Con questa tecnica si può anche specificare più di un file per volta, scrivendo la destinazione sempre per ultima.

```
$ mv mara.jpg pippo.jpg foto
```

Andate a vedere cosa sia successo dopo questi spostamenti. Per prima cosa bisogna che vi spostiate nella directory che vi interessa digitando il comando *cd* (abbreviazione di *change directory*) seguito dal nome della directory. Poi chiedete la lista completa dei file con ls:

```
$ cd foto
$ ls -a
 . .. iano.jpg mara.jpg melo.jpg pippo.jpg turi.jpg
$
```

I file ci sono tutti, ma ci sono anche dei puntini che non sapete proprio chi li abbia lasciati lì. In effetti non sono proprio dei file, ma la rappresentazione della *directory corrente (.)*, cioè la foto, e di quella *superiore (..)*.

Notate l'opzione *-a*. Essa dice al comando *ls* di visualizzare anche quella categoria di files *nascosti* e cioè tutti i file precedenti da un punto (.) .

5.3 La vostra directory di lavoro (la home directory)

Sì, perché anche quella in cui sta quella massa disordinata di files è una directory. Per la precisione è la vostra directory di lavoro iniziale, che vi abituerete a chiamare *home directory*. Il vostro computer probabilmente la chiamerà semplicemente ~. Subito dopo il login vi ritroverete normalmente nella vostra directory di lavoro. La directory foto è sotto la vostra home. Perché sotto?

Perché le directory si dispongono esattamente come un albero (anche se a testa in giù). Se dentro la directory foto create un'altra directory che chiamate album, quest'ultima sarà un ramo di foto, che a sua volta era già un ramo della home.

Per discendere verso un ramo più basso (tenete sempre presente che l'albero è rovesciato), bisogna chiamarlo con un comando del tipo:

```
$ cd sottodirectory
```

dove con sottodirectory si intende il nome della directory a cui volete accedere (viene chiamata anche *directory figlia*).

Per salire su un ramo superiore userete invece sempre il comando:

```
$ cd ..
```

perché .. rappresenta, appunto, la directory madre della directory all'interno della quale vi trovate.

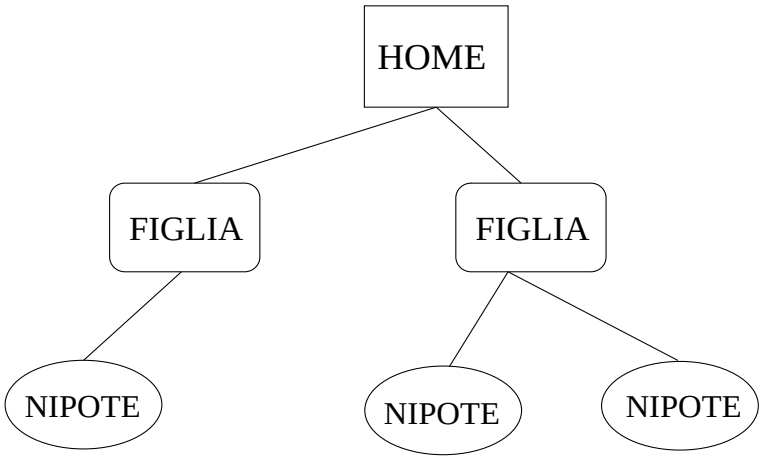

Figura 5.1: Relazioni fra le directory

In figura potete notare le relazioni che sussistono fra le directory.

5.4 Dove sono?

Alcuni computer sono tanto gentili da informarvi sempre riguardo alla directory in cui vi trovate. Da altri questa informazione va estorta. Digitate quanto segue e saprete ritrovare la posizione:

```
$ pwd /usr/luther/foto
```

Questo computer appartiene al partito tradizionalista e vi mostra la posizione corrente sotto la directory /usr. Quelli del partito innovatore vi direbbero invece /home/luther/foto.

5.5 Il file system di UNIX

Che diavolo saranno mai */usr* e */home*?

Potete immaginare già a questo punto che la vostra home è figlia di qualche altra directory superiore. Un tipico albero, con la radice completa, di un computer Unix è fatto come nella figura.

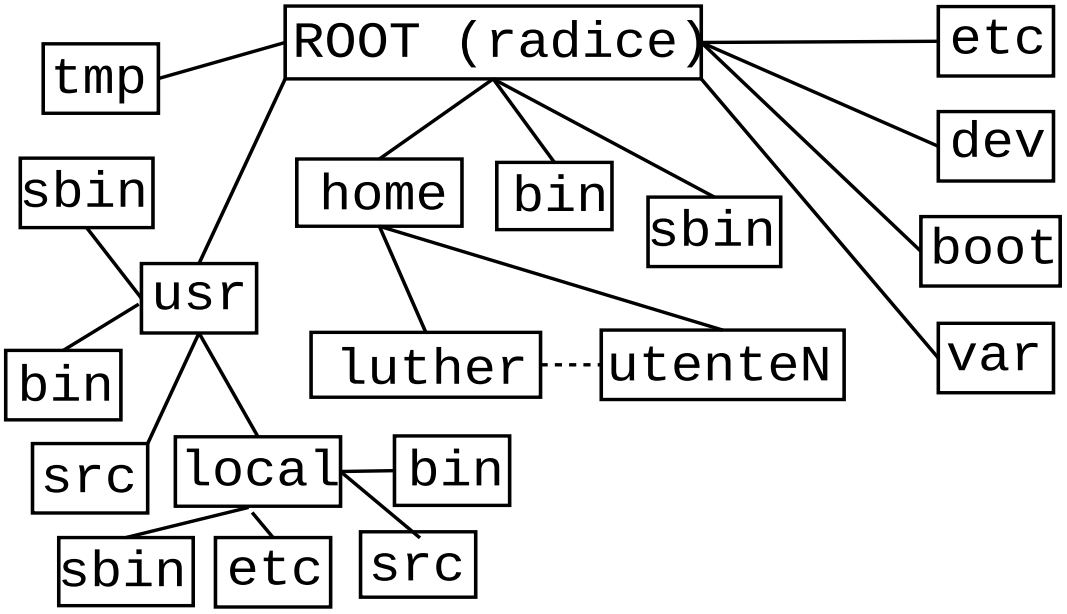

Figura 5.2: un tipico file system di UNIX

Ovviamente non su tutti i sistemi esiste una directory luther (anzi, quasi su nessuno), ma nello schema è riportata per darvi idea di dove si trovi la vostra home.

La directory da cui tutto ha origine si chiama *root*, o *radice*; per i più concisi semplicemente /: ecco perché anticipa *usr* e *home* negli esempi precedenti.

Ogni directory contiene dei file usati per fini specifici. I programmi stanno sotto la bin, la configurazione sotto etc, ecc. Non vale la pena approfondire per il momento. Tenete presente che partendo da una directory avete sempre due modi per raggiungere un'altra directory.

Le due forme vi portano allo stesso posto (**partendo dalla vostra ~**):

- *percorso relativo*: ../../usr/src;
- *percorso assoluto*: /usr/src.

5.6 Il file smarrito

A questo punto avrete già fatto ordine nella vostra home. Ma dove avete cacciato quel file che vi serviva tanto e che ora non è più a vista?

Prima di entrare nel panico cercate con un comando come questo:

```
$ find directory_di_partenza -name nome_del_file_da_cercare -print
```

Il comando inizierà una ricerca a partire dalla directory di partenza alla ricerca del vostro file, entrando in **tutte** le sottocartelle che stanno sotto la cartella di partenza. Non vi resta che aspettare una risposta, sperando che non sia negativa.

Supponiamo di trovarci su ~ e di dare il comando:

```
$ find . -name mio_file -print
```

In questo caso il comando find cerca il file indicato dopo l'opzione *-name* a partire dalla directory specificata subito dopo il comando stesso (in questo caso la directory corrente, che corrisponde a ~). Se omettete *-print* alcuni computer si rifiuteranno di rispondervi.

Capitolo 6

Il dischetto, dove lo metto?

6.1 Un disco da monta

L'albero delle directory di Unix non si riferisce solo a ciò che è depositato sull'hard disk. A differenza di altri sistemi operativi che assegnano una lettera di unità a tutte le memorie di massa, Unix considera ogni disco come una directory del suo albero monolitico. Per attaccare un disco a un ramo dell'albero bisogna dirlo al computer esplicitamente. Ciò è necessario ogni volta che volete che Unix si renda conto dell'esistenza del dischetto sul quale volete fare una copia dei vostri file. Anche per vedere il contenuto di un cd-rom c'è questa necessità. Il programma che si occupa di quest'innesto si chiama *mount*.

Non sempre vi verrà permesso di lanciare mount. In quel caso cercate di giungere a un accordo col vostro amministratore, anche se probabilmente vi chiederà di usare qualche altro programma a suo avviso più sicuro. Tra i comandi che abbiamo visto finora, mount è tra i più rognosi per la quantità e la qualità degli argomenti e delle opzioni obbligatorie.

```
$ mount perif_da_innestare dir_da_innestare -t
tipo_di_partizione
```

***riferimento_alla_periferica*:** indica il file (completo di percorso) della directory */dev* associato alla periferica che state tentando di innestare. È di solito *fd0* ("fd" sta appunto per Floppy Disk) per il primo floppy del computer (*fd1* per il secondo e così via). Comincia a diventare un po' più complicato se quello che si vuole montare è un cdrom. Se non esiste un link simbolico chiamato cdrom collegato al giusto file di periferica, è probabile che sitratti di *hdb*, *hdc* o *hdd*; tentate e se non ottenete gran risultati allora avete un cd-rom scsi o siete l'unico fortunato possessore di un prototipo di cd-rom respinto dal mercato perché troppo fuori standard. Nell'ultimo caso, consigliate all'amministratore di sistema di sostituirlo con qualcosa di contemporaneo.

***directory_da_innestare*:** rappresenta il ramo a partire dal quale sarà visibile il vostro floppy (o cd o quello-che-sia) nell'albero delle directory.

-t ***tipo_di_partizione***: con l'opzione -t si specifica il tipo di formattazione logica del vostro disco. Vale *vfat* per un dischetto preparato con Windows 9x, *minix* o *ext2* per un floppy creato con Linux. Vale sempre *iso9660* per un cdrom.

6.2 Un'utile tabella

L'amministratore può avere già impostato una serie di directory predefinite per attaccare i vostri dischi. Queste impostazioni sono contenute nel file */etc/fstab* (sta per *File System TABle*) ed è utile conoscerle.

```
$ cat /etc/fstab
/dev/hdc4 swap swap defaults 0 0
/dev/hdc1 / ext2 defaults 1 1
/dev/hdc2 /usr ext2 defaults 1 1
/dev/hdc3 /home ext2 defaults 1 1
/dev/hda1 /DOS vfat uid=1000,gid=100,noexec 1 0
/dev/cdrom /cdrom iso9660 user,noauto 0 0
/dev/fd0 /floppy vfat noexec,user 0 0
none /dev/pts devpts gid=5,mode=620 0 0
none /proc proc defaults 0 0
$
```

Qualcosa di utile si riesce a intravederla.

Il cdrom viene montato di default sulla directory */cdrom*. Il che vuol dire che in pratica per montare il cdrom con i giusti parametri basta dare un comando come quello che segue:

```
$ mount /cdrom
mount: block device /dev/cdrom is write-protected, mounting
read-only
$
```

Analogamente, per montare il floppy basta dare:

```
$ mount /floppy
$
```

Attenzione alla formattazione del dischetto. Il comando che avete visto andrà bene per montare un dischetto di Windows, ma non va bene per tutti i dischetti preparati in altro modo: nel */etc/fstab* è specificato come default il tipo di partizione *vfat*. Per tutti gli altri tipi di partizione sarà necessario specificare l'opzione relativa al tipo di partizione e scrivere tutto nella forma estesa.

6.3 Prima e dopo

La sequenza di istruzioni che segue è più eloquente di mille righe di spiegazione:

```
$ mount /floppy/
$ ls -l /floppy/
total 7
drwxr-xr-x 3 operator sys 112 Dec 10 1999 documenti/
-rw-r--r-- 1 operator sys 0 Dec 10 1999 fnml.shadow
-rw-r--r-- 1 operator sys 0 Dec 10 1999 posta.pgp
-rw-r--r-- 1 13 sys 3790 Dec 10 1999 program.txt
drwxr-xr-x 2 operator sys 32 Dec 10 1999 tutorial/
drwxr-xr-x 2 operator sys 32 Dec 10 1999 web/
$ ls -l /floppy/documenti
total 1
-rw-r--r-- 1 operator sys 0 Dec 10 1999 aurora.txt
drwxr-xr-x 2 operator sys 32 Dec 10 1999 foto/
-rw-r--r-- 1 operator sys 0 Dec 10 1999 mail1.pgp
-rw-r--r-- 1 operator sys 0 Dec 10 1999 treni.txt
```

```
-rw-r--r-- 1 operator sys 0 Dec 10 1999 vito.txt
$ umount /floppy/
$ ls -l /floppy/
total 0
$ ls -l /floppy/documenti
/bin/ls: /floppy/documenti: No such file or directory
$
```

6.4 Il disco smontato

Volete essere certi che le vostre operazioni di copia dei file su dischetto siano andate a buon fine? Non ve lo potremmo garantire, ma vi assicuriamo che non tutto andrà per il meglio se dimenticate di smontare il dischetto. Analogamente al modo con cui avete innestato il ramo, c'è una procedura per la potatura. Il programma è *umount* e su molti Unix l'unico parametro che richiede è la directory su cui è innestato il disco che si vorrà rimuovere. Ricordate che, smontando un disco, la vostra directory di lavoro non può appartenere a un ramo che si trova sul disco stesso. È impossibile dimenticare di smontare un cdrom, perché il computer si rifiuterà di restituirvelo fino a quando non sarà stato rimosso dal file system.

```
$ umount /cdrom
```

Capitolo 7

L'interprete capisce un TUBO

7.1 L'idraulica sotto Unix

Delle volte non riuscirete a leggere il risultato di un comando che avete lanciato perché la sua risposta è così logorroica da non bastare una singola schermata. Per esempio, se nella nostra directory di lavoro avete tanti file ancora non ripartiti tra le necessarie sottodirectory, la lista di questi può richiedere più spazio di quello che avete sul monitor. Nella schermata qui in basso non si vede il prompt con il comando digitato perché è stato spinto in alto dall'elenco dei file e la prima parte dello stesso elenco è fuoriuscita dallo schermo.

```
budget_dic1998 budget_set1998 rapporto_apr98 rapporto_ott98
budget_dic1999 budget_set1999 rapporto_apr99 rapporto_ott99
budget_feb1996 budino_cioccol rapporto_dic96 rapporto_set96
budget_feb1997 budino_vanigli rapporto_dic97 rapporto_set97
budget_feb1998 carmelo.jpg rapporto_dic98 rapporto_set98
budget_feb1999 def rapporto_dic99 rapporto_set99
budget_gen1996 ghi rapporto_feb96 relazione budget_gen1997
iano.jpg rapporto_feb97 riepilogo_gene budget_gen1998 jkl
rapporto_feb98 rubrica budget_gen1999 la_divin.txt
rapporto_feb99 siti_belli budget_giu1996 lettera_alfred
rapporto_gen96 siti_computer budget_giu1997 lettera_banca
rapporto_gen97 siti_intranet budget_giu1998 lettera_carla
rapporto_gen98 siti_lavoro budget_giu1999 lettera_cetty
rapporto_gen99 siti_trash budget_lug1996 lettera_direzi
rapporto_giu96 spese budget_lug1997 lettera_giovan
rapporto_giu97 stu budget_lug1998 lettera_inail
rapporto_giu98 torta_limone budget_lug1999 lettera_inps
rapporto_giu99 torta_mele budget_mag1996 lettera_lele
rapporto_lug96 torta_ricotta budget_mag1997 lettera_luigi
rapporto_lug97 turi.jpg budget_mag1998 lettera_patriz
rapporto_lug98 vwx budget_mag1999 lettera_rita
rapporto_lug99 yz budget_mar1996 lettera_robert
rapporto_mag96 budget_mar1997 lettera_sergio rapporto_mag97
$
```

Non c'è modo di dire al comando *ls* di aspettarvi e darvi il tempo di leggere tutto. Però si può prendere l'output di *ls* e spedirlo a un programma più educato.

```
$ ls | more
```

Guardate l'effetto del comando che è stato appena dato:

```
abc
budget
budget_ago1996
budget_ago1997
budget_ago1998
budget_ago1999
budget_apr1996
budget_apr1997
budget_apr1998
budget_apr1999
budget_dic1996
budget_dic1997
budget_dic1998
budget_dic1999
budget_feb1996
budget_feb1997
budget_feb1998
budget_feb1999
budget_gen1996
budget_gen1997
budget_gen1998
budget_gen1999
budget_giu1996
standard-input, 1-23 (Top)
```

Adesso è *more* a occuparsi della visualizzazione della lista dei vostri file. Questo effetto magico è stato ottenuto grazie a `|`, cioè il *tubo* (*pipe*). È come se l'uscita di *ls* sia stata incanalata attraverso la condotta e fatta arrivare fino a *more*.

Pipe risulta utile in molte situazioni, non solo per vedere l'elenco dei file.

Per esempio quando vi viene voglia di cercare la parola "cielo" all'interno della Divina commedia e volete leggere tutti i versetti che la contengono potendo facilmente anche tornare indietro.

```
$ grep cielo la_divin.txt | less
```

Qui è stato lanciato il comando. Andate a vedere nella pagina successiva cosa è successo.

```
$ grep cielo la_divin.txt | less
per quel ch'ì ho di lui nel cielo udito.
curan di te ne la corte del cielo,
Non isperate mai veder lo cielo:
D'ogne malizia, ch'odio in cielo acquista,
veggendo il cielo a te così benigno,
che mostri in cielo, in terra e nel mal mondo,
Lascian'andar, chi nel cielo È voluto
quando i cavalli al cielo erti levorsi,
Giove del cielo ancora quando tuona.
ni Tanai l' sotto 'l freddo cielo, Da questa
parte cadde giù dal cielo; Vedi come l'ha
dritte verso 'l cielo, che decreto del cielo
orazion pieghi; Li occhi miei ghiotti andavan
pur al cielo, al cerchio che più tardi in cielo
h torto. Più ch'altra creatura, giù dal cielo
Chiamavi 'l cielo e 'ntorno vi si gira,
la famiglia del cielo", a me rispuose:
d'ogne pianeto, sotto pover cielo, chi nel cielo uno,
e un qua giù la pone". pur suso al cielo,
pur come se tutto Lo cielo i vostri movimenti inizia;
rivolga il cielo a si, saprai; ma prima
line 1/62 68%
```

7.2 Fare ordine anche "laddentro"

Un altro comando con il quale piazzare un tubo può esservi utile è *sort*. Sort ordina alfabeticamente quello che trova all'interno di un file. Guardate la differenza:

```
$ cat rubrica
Umberto u.scapagnini@comune.catania.it
George president@whitehouse.gov
Vladimir putin@kremlin.ru
Liz queen@royalpalace.go.uk
Luciano benetton@benetton.it
Cicciobello rutelli@ulivo.it
Baffetto DALEMA_M@camera.it


$ cat rubrica | sort
Baffetto DALEMA_M@camera.it
Cicciobello rutelli@ulivo.it
George president@whitehouse.gov
Liz queen@royalpalace.go.uk
Luciano benetton@benetton.it
Umberto u.scapagnini@comune.catania.it
Vladimir putin@kremlin.ru


$ cat rubrica | sort -r
Vladimir putin@kremlin.ru
Umberto u.scapagnini@comune.catania.it
```

```
Luciano benetton@benetton.it
Liz queen@royalpalace.go.uk
George president@whitehouse.gov
Cicciobello rutelli@ulivo.it
Baffetto DALEMA_M@camera.it
```

- *sort* con l'opzione *-r* ve li ordina a rovescio.
- *sort* ha anche un'altra opzione di una qualche utilità. Specificando *-n* ordinerà per progressione numerica e non alfabetica. La differenza con *-r* la capirete da soli con un po' di esperienza.

Si possono usare anche due o più *pipe* all'interno della stessa riga di comando.

Quella di questo esempio serve a mandare direttamente in stampa la vostra rubica già ordinata:

```
$ cat rubrica | sort | lpr
```

State sicuri che troverete tantissime modi utili di far scorrere i dati nei tubi.

7.3 Più grande di tutti i file: operatori di reindirizzamento

Non è detto che i dati all'uscita di un comando vi interessino nello stesso momento che lo lanciate.

Potreste volere una lista dei file della vostra directory per guardarla da spedirvi attaccata alla posta elettronica. Vi serve cioè reindirizzare l'uscita di un comando verso un file. Per farlo dovete comunicarlo all'interprete dei comandi (da ora in poi semplicemente *shell*) con questo segno > *(maggiore)*.

```
$ ls > lista_file
```

Sul monitor non vedrete apparire assolutamente niente, ma un nuovo file di nome lista_file è stato creato nella vostra directory. Provate a lanciare un cat su di esso.

Se rifate il comando *ls > lista_file* il vecchio lista_file sarà sostituito dal nuovo e il suo contenuto irremediabilmente perso. Piazzando due segni di maggiore il vecchio contenuto sarà salvaguardato e il nuovo piazzato in coda (i guru chiamano questo comportamento "scrittura in modalità *append*"). Provatelo.

```
$ ls >> lista_file
```

La shell capisce anche il simbolo < (minore). Mentre > si occupa del reindirizzamento dell'output, < di quello dell'input. Difficilmente vi capiterà di usarlo, per cui è meglio tagliarla qua.

7.4 Jolly e altri mezzi per barare: le wildcards (? e *)

Spostare a uno a uno i file del budget dentro la loro naturale directory budget può essere lungo e noioso. Per fortuna chi ha scritto la shell vi aveva abbastanza a cuore da darvi una scorciatoia, anzi due:

1. *?* sostituisce un singolo carattere;
2. * sostituisce un numero imprecisato di caratteri.

Dando in pasto alla shell la riga qui sotto, sposterete in un sol colpo tutti i file i cui nomi cominciano per budget_:

```
$ mv budget_* budget
```

Analogamente, per avere la lista di tutti i file i cui nomi finiscono per 96 digiterete quello che segue:

```
$ ls *96
budget_ago1996 budget_lug1996 rapporto_ago96 rapporto_lug96
budget_apr1996 budget_mag1996 rapporto_apr96 rapporto_mag96
budget_dic1996 budget_mar1996 rapporto_dic96 rapporto_mar96
budget_feb1996 budget_nov1996 rapporto_feb96 rapporto_nov96
budget_gen1996 budget_ott1996 rapporto_gen96 rapporto_ott96
budget_giu1996 budget_set1996 rapporto_giu96 rapporto_set96

$ ls ???
abc def ghi jkl mno pqr stu vwx
$
```

Con il secondo comando avrete invece chiesto la lista di tutti i file i cui nomi sono di tre lettere.

Rapido, no?

7.5 Collegamento per mezzo di simboli

Volete arrivare da qualche parte più veloci della luce? Con Unix si può.

Basta costruire un *link simbolico* verso la vostra destinazione. Il comando è *ln* e dovrete sempre specificare lopzione *-s*. In questo esempio abbiamo fatto in modo che attraverso un link si arrivi subito alla directory ~/foto/album come se partisse direttamente sotto la vostra home. Guardate la figura se vi gira un po' la testa.

```
$ ln -s foto/album album
$ cd album
$ pwd
~/album
$
```

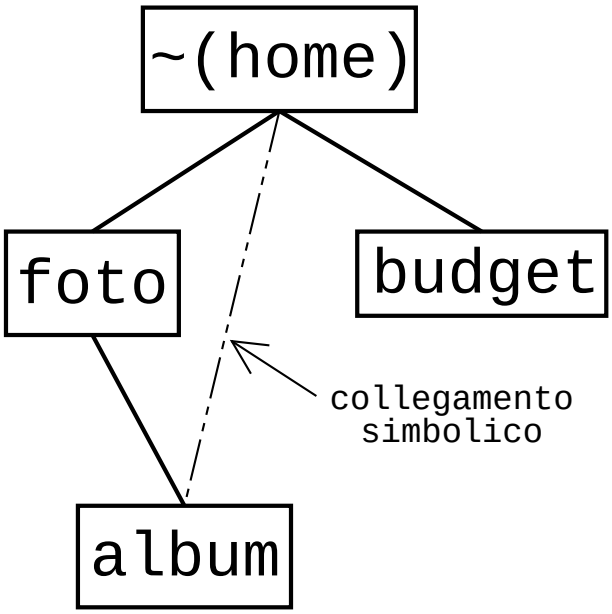

Figura 7.1: Collegamento simbolico - la vostra home directory e il collegamento creato con il comando ln

Parte III

I primi passi

Capitolo 8

Tutto quello che può tornare utile

8.1 Preparare i file di Natale

Tra le operazioni possibili con i file ce n'è una che non avrete mai immaginato: *l'impacchettamento*.

Pur non essendo ancora in grado di attaccarci un bel fiocco, è possibile prendere un file e impacchettarlo, prepararlo cioè per la spedizione o per essere messo in cantina per un lungo periodo.

Per preparare questa confezione molto speciale bisogna innanzi tutto comprimere il file.

Guardate la schermata che segue e che effetto fanno i comandi:

```
$ ls -l la_divin.txt*
-rw-r--r-- 1 luther users 537610 Nov 23 14:41 la_divin.txt

$ gzip -9 la_divin.txt

$ ls -l la_divin.txt*
-rw-r--r-- 1 luther users 219056 Nov 23 15:10 la_divin.txt.gz
$
```

Inizialmente la_divin.txt era un corposo file da più di mezzo milione di caratteri. Dopo il comando *gzip* il file che conteneva la Divina Commedia è scomparso e ne avete trovato uno dal nome molto simile che è grande la metà e che ha il suffisso *.gz* alla fine.

Se adesso provate a leggerla non riuscirete a trovare lo stile di Dante, ma una cosa del genere:

```
$ more la_divin.txt.gz
^Tt^EhJ$ZhtNN^DGP^@^_"^E^WwY5$^@.X+\]
ca^C^M^?p^M?>j^|p{^O??j^^?rSO_>yaU_^?rNM^[4,ms*KWw9^
]r_^?rCO_}{w^_^?yk_t^?~c^[^?|a84S0>h^^WO^NM^O^?q?>yO
|(i~n;&o_?^_^[syS)u}^^^Z2h0NK^T^Z{^R_^?RgfJK4^SnnmT,
Cjf6;'f^\OkT>qzwgS%r^Kes^K~-5myu9OSy^Wtox*Wors_+/^@^
_D_:~lCs^Zgro^\*^]^W^X^]^?{=b0&^?=^LSRb^K[f(W|k^G9\9
>!\^^.a^VJ0_a^K^?+_uxP$r^Q)A/\FSX>l^V^?Qtfe95D;*wZ.{
bCxn^GriC>^T[htWG<hiSyqS^]R7L-.8^M^O^]^^MM|<Q^_Frx^F
f^^9T^Tq/'1WGsP>^C.8:\}-7|ds8^Lx4^?5b>Vr^\t^[e^^nes'
^V^O5=?'a<^N)s;~*??f_HeMMy$7|!9^_Vr^WcW5M5c, r^M]^_M
y<7MRNe-';>C'r9}Y^?Hw^V/GO1IeR^L<|^[Gi^Zk^Z&s^Yeg1^^
^[\Yy^Mr]sZ~^?MC]K^[L^K>?|o{h&<J>a[NC;$^?c[fS^YsfDpS
*96^^Gr|1^R&OW^UK8,u^M^WI(wYn!^FB^DhRUs9|>g^CF<BGJ/^
Nc2Dk;%y.wq*/o^B/BSu[\p^K!3'^N>Sa:Se^R^F^A#-s)^WK>,#
```

```
&M-eWs^NNy^>1gf>vW^V^?vH’2]qgyZeLv^VOqI|"e2}Dg^F\^OM
^_K#-^W^X^F^So9<5;>&af^L75e^b?9OS^Y6]H;q^DAdkR9<O|v=
Vm^Y^X^Xw]a/^Z4^_ J^OJE^TG)^W~^[1B^Lw=e[Tb;3^V
la_divin.txt.gz, 1-9 (Top)
```

È successo semplicemente che il programma gzip ha cercato le cose più ricorrenti e le ha trascitte su una tabella che può capire solo lui.

Insomma, *gzip* ha fatto una cosa come quella qui sotto:

```
trentatre trentini entrarono a trento tutti e trentatre
trotterellando.
tre tigri contro tre tigri

#=tr,@=tre,*=ti

@nta@ @n*ni en#arono a @nto tut* e @nta@ #otterellando.
@ *gri con#o @ *gri

98 caratteri
90 caratteri (75 testo, 15 tabella)
```

Su un file piccolo come questo non c’è un gran risparmio e la tabella occupa molto spazio in proporzione al resto, ma su testi più grandi il risparmio diventerà qualcosa di ben più tangibile (avete visto di quanto si sia ridotta la Divina Commedia!).

Quando è stato lanciato *gzip* è stata specificata l’opzione *-9*.

In questo modo avete detto al programma “Comprimi più che puoi, anche se devi perdere più tempo”.

Se aveste scritto 1 gli avreste detto “Sbrigati subito, anche se mi lasci il file poco compresso.”

Con 5 avreste fatto metà e metà. È valido qualsiasi numero sulla scala tra 1 e 9.

8.1.1 Aprire i doni

Avete ricevuto il vostro file bell’e impacchettato. Per togliere il nastro basta fare quanto segue:

```
$ gunzip la_divin.txt.gz
```

Et voilà!

```
$ ls -l la_divin.txt*
-rw-r--r-- 1 luther users 537610 Nov 23 14:41 la_divin.txt
$
```

Come per i pacchetti veri, è più facile disfare la confezione che prepararla.

In questo caso a sparire dopo aver lanciato *gunzip* sarà il file compresso.

8.1.2 Schiacciare un ramo

Facciamo il caso che i file che volete regalare per questo Natale (a proposito: auguri!) a vostro cugino comprendano un’intera collezione di francobolli.

Per quello che sapete fare finora potreste impacchettarglieli uno per uno.

Sapete che divertimento con centotrenta cosi da far passare manualmente a gzip!.. Anche vostro cugino sarà felice di spacchettarli tutti e centotrenta uno per uno.

Come scherzo non sarebbe male, specie perché sto cugino vi è sempre stato antipatico.

Comunque ci sono metodi più rapidi che vi permetteranno di risparmiare fatica (anche a vostro cugino, purtroppo).

Il programma *tar* vi potrà essere d'aiuto in situazioni del genere e in tanti altri casi.

Basta lanciarlo nella maniera più opportuna. Per esempio:

```
$ tar cfv nome_archivio.tar *
```

Creerà un nuovo file (la creazione è specificata dall'opzione *c*) di nome *"nome_archivio.tar"* (il nome è specificato dall'opzione *f*) e ci metterà dentro i file specificati, cioè tutti quelli presenti nella directory e nelle directory inferiori.

Prestate attenzione a due cose: *tar* non cancellerà i file che ha accorpato, ma li lascerà al loro posto; l'opzione *f* deve essere sempre specificata per ultima tra le opzioni, altrimenti *tar* si confonderà e non farà quello che desiderate. Se provate a farlo girare con una forma del tipo:

```
$ tar cfv archivio_foto.tar foto
```

otterrete l'accorpamento solo della sottodirectory foto.

Il programma tar ha solo messo assieme i file e non li ha compressi (quindi il vostro archivio_foto.tar occuperà lo stesso spazio che occupano i file che vi sono stati accorpati). Potreste usare gzip per comprimere il file con estensione .tar che avete generato.

Oppure avreste potuto provare a chiedere direttamente a tar di farlo, inserendo una z tra le opzioni. Tipo:

```
$ tar czfv archivio_foto.tar.gz foto
```

8.1.3 Reinnestare un ramo

Per scompattare un file creato con tar utilizzate sempre *tar* con queste opzioni (la *x* sta per *eXtract*):

```
$ tar xfv nome_archivio.tar
```

Se invece il file è un archivio compresso basta aggiungete l'opzione *z* come quando è stato creato:

```
$ tar xzfv nome_archivio.tar.gz
```

8.2 Che anno è? Che giorno è?

Il vostro computer, grazie al programma *date*, è una macchina finalmente in grado di dare una risposta agli interrogativi che Mogol ha posto e che hanno angosciato generazioni di italiani.

```
$ date
Tue Nov 23 17:35:26 GMT 1999
```

In più ci dice anche l'ora esatta! Se poi voleste anche sapere qualcosa in più a proposito di quel famoso 29 settembre, vi tornerà utile il programma *cal*.

Si usa mettendo come argomenti prima il numero del mese e poi l'anno a quattro cifre.

```
$ cal 9 1967
        Sep 1967
 S  M Tu  W Th  F  S
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

8.3 Fatevi pubblicità

Se avete propria la ferma intenzione di farvi notare, potrebbe essere utile fare la conoscenza con bban e banner

```
quest@medialab:~$ bban shining rules!
```

Con bban potrete ottenere scritte visibili sullo schermo. Su molte versioni di Unix, con *banner* avrete scritte così grosse che sarà meglio reindirizzarle subito verso la stampante. Sui computer non cè *bban*, *banner* vi mostrerà scritte più maneggevoli.

Potete anche utilizzare *figlet* che è un po' meno invasivo come programma.

8.4 I gemelli del file

Se sospettate che due file abbiano lo stesso contenuto, provatelo con *cmp* e *diff*. Si usano così:

```
$ cmp la_divin.txt rubrica la_divin.txt
rubrica differ: char 1, line 1
$
```

In questo caso sono due file diversi. Se fossero stati uguali, Unix per discrezione non vi avrebbe detto niente.

```
$ diff budget_set1997 budget_set1998
$
```

Questi due invece erano uguali. Se fossero stai diversi, *diff* sarebbe stato ben più loquace di *cmp*.

8.5 Fare i conti con l'host

Finalmente avete una buona ragione per usare il tastierino numerico: il programma *bc*.

```
$ bc
bc 1.02 (Mar 3, 92)
Copyright (C) 1991, 1992 Free Software Foundation, Inc.
3*2
6
quit
$
```

Dopo aver lanciato bc potete scrivete l'operazione e premete invio. Per uscire digitate *quit*.

Capitolo 9

Toc! Toc! È permesso?!?

9.1 Di chi sono questi file?

Come avete visto un computer Unix è pensato per far lavorare molte persone contemporaneamente. Che ne pensereste se chiunque possa mettere il naso tra le vostre cose, i vostri file, le vostre lettere personali? Quanto meno che Unix è una grande str*nz@t@. Ma per fortuna chi lo ha pensato aveva a cuore la vostra privacy quanto e più di Rodotà.

Guardate un po' che ha pensato, date il seguente comando su una qualsiasi directory contenente dei files:

```
$ ls -l
```

Ricordatevi che *-l* sta per *long*.

```
-rw-r--r-- 1 luther users 0 Nov 18 14:44 rapporto_ott96
-rw-r--r-- 1 luther users 0 Nov 18 14:44 rapporto_ott97
-rw-r--r-- 1 luther users 0 Nov 18 14:44 rapporto_ott98
-rw-r--r-- 1 luther users 0 Nov 18 14:44 rapporto_ott99
-rw-r--r-- 1 luther users 0 Nov 18 14:44 rapporto_set96
-rw-r--r-- 1 luther users 0 Nov 18 14:44 rapporto_set97
-rw-r--r-- 1 luther users 0 Nov 18 14:44 rapporto_set98
-rw-r--r-- 1 luther users 0 Nov 18 14:44 rapporto_set99
-rw-r--r-- 1 luther users 1641 Nov 16 14:48 relazione
-rw-r--r-- 1 luther users 0 Nov 18 14:53 riepilogo_gene
-rw-r--r-- 1 luther users 209 Nov 20 17:05 rubrica
-rw-r--r-- 1 luther users 0 Nov 18 14:52 siti_belli
-rw-r--r-- 1 luther users 0 Nov 18 14:53 siti_computer
-rw-r--r-- 1 luther users 0 Nov 18 14:53 siti_intranet
-rw-r--r-- 1 luther users 0 Nov 18 14:53 siti_lavoro
-rw-r--r-- 1 luther users 0 Nov 18 14:52 siti_trash
-rw-r--r-- 1 luther users 0 Nov 18 14:51 spese
-rw-r--r-- 1 luther users 0 Nov 19 13:27 stu
-rw-r--r-- 1 luther users 0 Nov 19 12:13 torta_limone
-rw-r--r-- 1 luther users 0 Nov 18 14:45 torta_mele
-rw-r--r-- 1 luther users 0 Nov 18 14:45 torta_ricotta
-rw-r--r-- 1 luther users 0 Nov 18 14:47 turi.jpg
-rw-r--r-- 1 luther users 0 Nov 19 13:27 vwx
-rw-r--r-- 1 luther users 0 Nov 19 13:27 yz
 $
```

A destra si riconosce il nome del file. Subito prima c'è l'orario dell'ultima modifica. Ancora prima la sua dimensione espressa in numero di caratteri. E prima ancora ci sono due colonne molto importanti:

- la prima vi dice a quale utente appartiene il file (in questo caso tutti i file appartengono a luther);
- la seconda a quale gruppo (in questo caso users).

Cos'è sta storia del gruppo?

Finora vi abbiamo omesso qualcosa a proposito della vostra vera identità a bordo di Unix. Fin dal momento in cui è stato creato l'account sul computer, siete stati inseriti in un insieme ben preciso di utenti. Tutti gli utenti di un sistema hanno delle caratteristiche comuni con gli altri utenti appartenenti allo stesso insieme: questo insieme è detto *gruppo*. Per esempio gli utenti di un certo gruppo possono lanciare dei programmi il cui uso è vietato ad utenti di altri gruppi, o condividere dei file il cui accesso è consentito solo agli utenti dello stesso gruppo.

È un po come l'appartenenza a una famiglia.

9.2 Che ci posso fare con sto file?

La prima colonna della schermata mostra le possibilità di leggere, scrivere o eseguire i file.

Tutti gli elementi della prima colonna hanno esattamente dieci caratteri.

Il primo vi dice se è una directory quando il suo valore è *d*, che è un link quando c'è la *l*, o che è un file speciale (tipo il riferimento a una periferica) se c'è scritto *b* o *c* (nella directory /dev sono tutti così).

Gli altri nove si leggono come tre distinti blocchi di tre lettere nel formato rwx.

- Il primo blocco rwx si riferisce al proprietario del file;
- Il secondo ai componenti del gruppo specificato (che non è necessariamente lo stesso gruppo a cui appartiene il proprietario);
- Il terzo al resto del mondo, cioè a tutti quegli utenti che non sono il proprietario, né fanno parte del gruppo a cui appartiene il file.

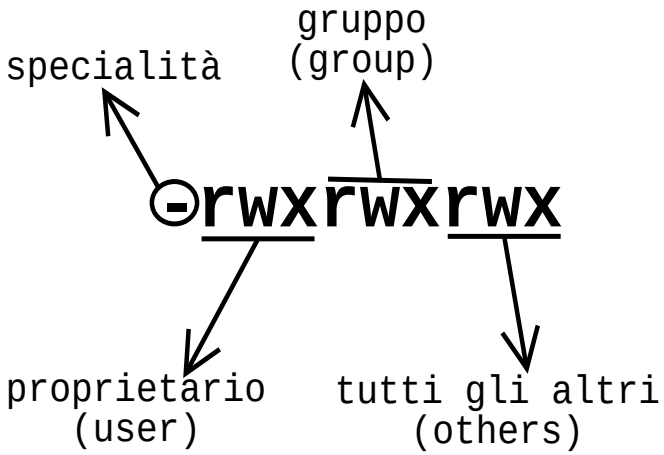

Figura 9.1: I permessi di un file, si notano i tre gruppi (user, group, others) e l'indicazione di specialità

I simboli r,w,x hanno i seguenti significati:

- La r indica la possibilità di leggere il file (Read);
- la w la possibilità di modificarlo (Write);
- la x la possibilità di eseguirlo (eXecute)

Notate che per le directory la x significa un'altra cosa, cioè la possibilità si essere esplorate (eXplore). Il - indica invece che la possibilità è negata. Cosa esattamente sia negato si capisce dalla posizione.

Ecco qui una tabella che vi renderà le idee molto più chiare:

Stato	Significato
- - -	Non è possibile fare alcunché
- - x	Non è possibile leggerlo ne modificarlo, è possibile eseguirlo
- w -	Non è possibile leggerlo, è possibile modificarlo, non è possibile eseguirlo
- w x	Non è possibile leggerlo, è possibile modificarlo ed eseguirlo
r - -	Si può leggere, ma non modificare o eseguire
r - x	Si può leggere. Non li si può modificare ma lo si può eseguire
r w -	Lo si può leggere e scrive ma non eseguire
r w x	Si può fare tutto: leggere, modificare ed eseguire

Adesso che avete chiaro come funzionano i permessi, andate a vedere se è tutto a posto tra i vostri file più riservati.

```
$ ls -l rubrica -rw-r--r-- 1 luther users 209 Nov 20 17:05
rubrica
$
```

In questo momento non solo gli altri utenti del gruppo users, ma tutti (proprio tutti!) possono leggere la rubrica di luther. Ciò non è carino!

9.3 C'è modo e modo

Vediamo cosa potete fare per la vostra privacy.

```
$ chmod u=rw,g=-rwx,o=-rwx rubrica
$ ls -l rubrica -rw------- 1 luther users 209 Nov 20 17:05
rubrica
$
```

Ora, grazie al comando *chmod*, solo voi potete leggere e modificare la rubrica, mentre il resto del mondo è escluso da qualsiasi accesso al file. Se non vi spiegate come sia stato possibile, pazientate ancora un po'!

Il comando *chmod* sta per "change mode" e il suo scopo è quello di cambiare i permessi a file e directory.

Come vedete il comando accetta informazioni sui gruppi di cui abbiamo parlato qualche riga più sopra.

- u è l'owner: qui anche inteso come "user";
- g è il gruppo
- o sta per "others": cioè tutti gli altri.

con il comando sopra abbiamo assegnato al possessore solo i permessi di scrittura e lettura (u=rw) e al gruppo abbiamo tolto tutti i permessi (g=-rwx). Lo stesso per tutti gli altri utenti (o=-rwx).

Avete già intuito il suo uso. Per assegnare dei permessi è sufficiente specificare per ogni gruppo le lettere in base alle nostre esigenze, mentre, per togliere un permesso è sufficiente anticipare il segno meno.

In realtà utilizzare questo tipo di sintassi per cambiare i permessi di un file farebbe rabbrividire i guru... che risolverebbero comodamente con un:

```
$ chmod 600 rubrica
$ ls -l rubrica -rw------- 1 luther users 209 Nov 20 17:05
rubrica
$
```

Cambiare i permessi in questo modo è molto più veloce, vero?

Ma che diavolo di complessi calcoli matematici stanno dietro? In realtà non c'è niente di più semplice... a patto che la matematica vi stia simpatica!

Per i più curiosi diciamo solo che viene utilizzato un modo di contare e di rappresentare i numeri in base 8, detto anche *ottale.* Per semplicità diciamo solo che è un modo di contare i numeri utilizzando solo 8 cifre invece che le classiche 10 cifre che noi siamo abituati a utilizzare nei nostri conti.

Avrete quindi intuito come il comando chmod non si usa solo nel modo sopra esposto, ma anche in modo più complesso.

Ci auguriamo che vogliate approfondire! L'approfondimento alla fine di questo testo fa al caso vostro.

Date anche solo una lettura. Vi renderete conto di come funzioni logicamente il vostro computer.

Deformando lo slogan di una famossisima campagnia publicitaria di prevenzione contro l'Aids diremmo che "se lo conosci non ti uccide mica!".

9.4 Bit, byte e altre cose che mordono

Abbiamo appena introdotto la parola bit (cosa che ci eravamo ben guardati dal fare finora). Ormai siete grandi abbastanza da non averne paura e neanche tanto bisogno. Per cui accontentatevi di una definizione simil accademica (anche se avremo modo di ritornarci):

> ***Il bit è l'unità fondamentale dell'informazione. Esso può avere solo due stati: acceso o spento; bianco o nero; simpatico o antipatico; attivo o inattivo. Per convenzione useremo più spesso 1 o 0.***

Per lo strano modo di ragionare dei computer, otto bit fanno qualcosa insieme: *un agglomerato di 8 bit costituisce un byte*.

Di fatto i computer mettono in memoria (qualunque essa sia, di lavoro o di massa) sempre otto bit insieme per volta. Per questo è nata quest'unità di misura. Ogni singolo carattere di questa pagina è costituito dall'insieme di otto bit. Un file che contiene un testo lungo 9000 battute, occupa esattamente 9000 byte.

Il byte ha dei multipli:

- il kilobyte (kB) equivale a 1024 byte (2^{10});
- il megabyte (Mb), 1048576 byte (2^{20});
- il gigabyte (GB), 1073741824 (2^{30});
- il terabyte (TB), 1099511627776 byte (2^{40}).

Capitolo 10

Il processo del lunedì.

10.1 Come evitare una collera al vostro capo

Quante cose per volta può fare un computer? Ok, tante, avete ragione. E quante cose per volta potete fare voi davanti al computer? Qui la risposta non è per niente scontata. Si può pensare "a me già una basta e avanza", ma presto vi renderete conto che invece sapersi muovere tra più programmi aperti contemporaneamente risulta comodo. Immaginate di essere immersi in un'entusiasmante partita di backgammon invece di completare la relazione che vi è stata richiesta ormai da una settimana, quando il boss entra coll'aria tuttaltro che tranquilla nella vostra stanza. Per giunta il tipo è arrivato proprio mentre state per fare un gammon in un match dove si è già raddoppiato due volte.

10.1.1 Dammi la console, presto!

Cè una soluzione rapida e immediata. Tenete premuto il tasto *alt* e schiacciate un tasto funzione. La combinazione *alt+Fn* sostituirà la vostra scacchiera con una richiesta di login nuova di zecca. Se foste stati previdenti avreste già aperto il documento con un editor sulla nuova console prima, in previsione di un'eventualità del genere.

Peccato, siete stati beccati lo stesso! I sistemi Unix costruiti in tempi successivi alla caduta di Costantinopoli vi danno l'opportunità di usare contemporaneamente più console. Cioè, è come se sulla vostra tastiera e sul vostro monitor coesistano nello stesso tempo più terminali. Per passare dall'uno all'altro generalmente si usa la combinazione che avete usato prima. Il numero di terminali disponibili dipende dalle capacità del computer e dalle impostazioni dell'amministratore. Dopo la cazziata, per consolarvi, cercate tra le vostre console quella su cui stavate giocando e finite di battere il computer.

10.1.2 Partita sospesa

Siccome oggi proprio non è il vostro giorno fortunato, avete premuto *alt+Fn* ma non funziona.

È un modo alquanto brusco di ricordarvi che non vi trovate direttamente sul computer ma su un terminale. Ma non tutto è perduto. Non sia mai detto che questa benedetta partita debba essere persa. La shell vi offre unaltra possibilità: provate a premere *ctrl+z*

```
[1]+ Stopped backgammon
$
```

La scacchiera è sparita. Al suo posto è comparso un messaggio che vi avvisa che la partita è stata temporaneamente sospesa. Vedete un po qual è la situazione dei programmi che avete lanciato.

```
$ jobs
[1]- Stopped (signal) vi relazione
[2]+ Stopped backgammon
$
```

Non vi resta che digitare rapidamente:

```
$ fg 1
```

In effetti in quest'occasione avete rischiato grosso. Un po' sprovveduti, non avete annotato prima il numeretto magico del vostro lavoro sospeso e avete dovuto cercarlo di corsa con jobs (la prossima volta appuntatevelo prima: tanto una volta che gli viene assegnato non cambia). Quando avete digitato *fg* la shell è andata a prendere il programma sospeso 1 e ve lo ha messo in foreground, cioè in primo piano. Ecco come mai è riapparso vi con tutta la vostra relazione dentro.

10.1.3 Vade retro, lumaca

Delle volte capita di lanciare un programma e solo successivamente rendersi conto di quanto sia lento. *Ctrl+z* può tornare utile anche in questo caso. Questa combinazione di tasti non riuscirà a velocizzarlo, ma almeno ci farà riguadagnare il prompt della shell. C'è da rispettare un unico accorgimento: il programma non va lasciato in sospeso, ma messo in background.

```
[3]+ Stopped lumaca
$ bg 3
```

Attenti quando mettete un processo in background con *bg*, perché il programma continuerà a parlarvi come se fosse lui il protagonista. Quindi vedrete i suoi eventuali messaggi sullo schermo e si fermerà se ha bisogno che digitiate qualcosa. In questo caso dovrete riesumarlo subito in primo piano con *fg* (trattandolo proprio come un processo sospeso) e inserire quello di cui ha bisogno. Poi potrete tornare a metterlo in background. Potete ripetere queste operazioni tutte le volte che volete.

10.1.4 Partenza lenta

Adesso che sapete quanto è lento quel programma siete avvisati. La prossima volta potreste farlo partire direttamente in background. Se esiste qualcosa di facile sotto Unix, forse è proprio questo: per far partire il programma in questo stato, basta aggiungere un *&* prima di battere invio.

```
$ lumaca &
$ jobs
[1]- Stopped (signal) vi relazione
[2]+ Stopped backgammon
[3] Running lumaca &
$
```

jobs si ricorderà di metterlo nell'elenco anche se il programma è proprio partito in background. Per la shell un programma in background è un programma comunque funzionante, per questo viene indicato come Running.

10.2 L'assassino dei lavori

Adesso che vi siete accorti che lumaca non è sto granché di programma e che sostanzialmente non fa niente, avrete deciso che è ora che si tolga da lì. Il comando che si usa è *kill*.

È molto importante mettere un % prima del numero del programma da uccidere. Quello che vedrete adesso si commenta da solo.

```
$ kill %3
$ [3]+ Terminated lumaca &
$
```

10.3 La distribuzione equa delle risorse

Non di rado i programmi lenti possono appesantire il sistema. Il che significa che sottrarete velocità agli altri programmi, compresi quelli degli altri utenti. Prima di buscarle potreste adottare un grazioso accorgimento: scrivere la riga per il vostro programma anticipandola con *nice*

```
$ nice lumaca &
$
```

nice fa in modo che il vostro programma non disturbi gli altri. Lo renderà un po più lento, ma che vi importa? Tanto già lo era.

10.4 Le statistiche sui processi

Ora, se tutto questo vi è sembrato inconsuetamente facile per uno Unix, sappiate che in realtà le cose sono molto diverse e, ovviamente, molto più complicate.

Sorpresi? Prima di confondervi del tutto è opportuno dire che diavolo sono questi processi di cui si continua a parlare. Considerateli come una proiezione simbolica dei programmi che stanno girando sul computer. Ogni programma che viene lanciato genera uno o anche più processi. I programmi che prima avete messo in background, o che avete riesumato, avevano a loro volta generato dei processi. Il sistema operativo li cataloga, li numera, procede a un'identificazione completa che tiene conto anche di chi ha lanciato il programma. Ricordate che un programma che gira a vostro nome (perché lanciato da voi o perché vostro e con lo sticky bit attivo) può leggere e/o modificare tutti i file su cui avete i rispettivi permessi. I numeri che avete messo dopo i vari fg, bg, kill sono validi solo per voi. Unix invece tiene il proprio conto e lo stesso programma che avete fatto fuori con kill %1 per lui era magari il processo numero 739 o 3471.

Per vedere quali sono in effetti i processi che avete aperto lanciando i programmi (considerate che anche la shell ne genera uno) digitate quanto segue e vi appariranno schermate simili.

```
$ ps
PID TTY TIME CMD
210 tty2 00:00:00 bash
304 tty2 00:00:00 backgammon
301 tty2 00:00:00 vi
325 tty2 00:00:00 ps
$
```

Se avete lanciato un programma che non ha bisogno di una console o di un terminale per funzionare non lo vedrete nell'elenco. Il comando *ps* può darvi informazioni anche sui processi non vostri. Per ottenere un elenco completo dei processi, compresi quelli che non usano la console e

quelli degli altri utenti, provate a lanciarlo con le opzioni *aux*. Per estrapolare tra tutti i processi proprio quelli vostri basta fare passare l'uscita di ps da un opportuno grep.

```
$ ps xua | grep luther
luther  210  0.0  1.7  1168  552  tty2  S 05:34 0:00 bash
luther  212  0.0  0.0  1168    0  tty4 SW 05:34 0:00 [bash]
luther  235  0.0  0.0   836    0  tty4 SW 05:35 0:00 [rlogin]
luther  236  0.0  0.0   836    0  tty4 SW 05:35 0:00 [rlogin]
luther  303  0.0  1.9  1216  600  tty2  T 05:41 0:00 vi relazion
luther  304  0.0  1.3  1120  428  tty2  T 05:41 0:00 backgammon
luther  320  0.0  2.0  2164  628  tty2  R 05:48 0:00 ps xau
luther  321  0.0  1.1   956  344  tty2  S 05:48 0:00 grep luther
$
```

Questa cosa vi tornerà utile per sbarazzarvi di qualche programma che proprio non vuole collaborare. Se non siete riusciti a fermarlo in alcun modo, non vi resta che identificarlo e sparargli con un arma veramente pesante: il *kill* con l'opzione *-9*. Prendete bene la mira prima di tirare un colpo di questo calibro, perché sbagliare significa far fuori un innocente. L'informazione in fondo a destra è il nome del programma (completo di eventuali opzioni). Sulla seconda colonna da sinistra (a fianco del vostro nome) c'è il PID. Dovrete riportarlo come argomento per *kill*.

```
$ kill -9 303
$ ps xau | grep luther
luther  210  0.0  1.7  1168  552 tty2 S  05:34 0:00 -bash
luther  212  0.0  0.0  1168    0 tty4 SW 05:34 0:00 [bash]
luther  235  0.0  0.0   836    0 tty4 SW 05:35 0:00 [rlogin]
luther  236  0.0  0.0   836    0 tty4 SW 05:35 0:00 [rlogin]
luther  304  0.0  1.3  1120  428 tty2  T 05:41 0:00 backgammon
luther  320  0.0  2.0  2164  628 tty2 R  05:48 0:00 ps xau
luther  321  0.0  1.1   956  344 tty2 S  05:48 0:00 grep luther
[1]- Killed vi relazione
$
```

Il processo 303 è stato spazzato via e non ne è rimasta più alcuna traccia. Notate che non è stato anteposto il % al numero del processo da far fuori, perché stavolta avete attaccato il vero *PID* (*Process IDentification*). Ora che vi siete finalmente liberati della noiosa relazione potete riprendere la partita a backgammon. Buon divertimento!

10.5 La respirazione bocca a bocca alla console

Delle volte il vostro schermo può fare i capricci. Succede spesso se vi ostinate a lanciare un cat su un file binario (cioè un file che contiene tuttaltro che testo, come un programma o un'immagine). Vi potrete trovare nell'imbarazzante situazione di non vedere più caratteri sensati sullo schermo e di non riuscire a produrne neanche voi. In questo caso premete ^*U* per cancellare quello che stavate scrivendo sul prompt della shell mentre vi siete lasciati prendere dal panico. Subito dopo digitate la parolina magica *reset*. La shell lo capirà, anche se voi non potrete vederla mentre la battete. Dopo l'invio tutto tornerà a posto.

Parte IV

La filosofia

Capitolo 11

Il manifesto del Partito linuxista

11.1 LICENZA PUBBLICA GENERICA (GPL) DEL PROGETTO GNU

Versione 2, Giugno 1991 Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA
Tutti possono copiare e distribuire copie letterali di questo documento di licenza, ma non è permesso modificarlo.

11.1.1 Preambolo

Le licenze per la maggioranza dei programmi hanno lo scopo di togliere all'utente la libertà di condividerlo e di modificarlo. Al contrario, la Licenza Pubblica Generica GNU è intesa a garantire la libertà di condividere e modificare il free software, al fine di assicurare che i programmi siano "liberi" per tutti i loro utenti. Questa Licenza si applica alla maggioranza dei programmi della Free Software Foundation e a ogni altro programma i cui autori hanno scelto questa Licenza. Alcuni altri programmi della Free Software Foundation sono invece coperti dalla Licenza Pubblica Generica per Librerie (LGPL). Chiunque può usare questa Licenza per i propri programmi.

Quando si parla di free software, ci si riferisce alla libertà, non al prezzo. Le nostre Licenze (la GPL e la LGPL) sono progettate per assicurare che ciascuno abbia la libertà di distribuire copie del software libero (e farsi pagare per questo, se vuole), che ciascuno riceva il codice sorgente o che lo possa ottenere se lo desidera, che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi liberi e che ciascuno sappia di potere fare queste cose. Per proteggere i diritti dell'utente, abbiamo bisogno di creare delle restrizioni che vietino a chiunque di negare questi diritti o di chiedere di rinunciarvi. Queste restrizioni si traducono in certe responsabilità per chi distribuisce copie del software e per chi lo modifica.

Per esempio, chi distribuisce copie di un Programma coperto da GPL, sia gratuitamente sia facendosi pagare, deve dare agli acquirenti tutti i diritti che ha ricevuto. Deve anche assicurarsi che gli acquirenti ricevano o possano ricevere il codice sorgente. E deve mostrar loro queste condizioni di Licenza, in modo che conoscano i loro diritti.

Proteggiamo i diritti dell'utente attraverso due azioni:

1. proteggendo il software con un diritto d'autore (una nota di copyright), e

2. offrendo una Licenza che concede il permesso legale di copiare, distribuire e/o modificare il Programma.

Infine, per proteggere ogni autore e noi stessi, vogliamo assicurarci che ognuno capisca che non ci sono garanzie per i programmi coperti da GPL. Se il Programma viene modificato da qualcun

altro e ridistribuito, vogliamo che gli acquirenti sappiano che ciò che hanno non è l'originale, in modo che ogni problema introdotto da altri non si rifletta sulla reputazione degli autori originari.

Infine, ogni programma libero è costantemente minacciato dai brevetti sui programmi. Vogliamo evitare il pericolo che chi ridistribuisce un Programma libero ottenga brevetti personali, rendendo perciò il Programma una cosa di sua proprietà. Per prevenire questo, abbiamo chiarito che ogni prodotto brevettato debba essere reso disponibile perché tutti ne usufruiscano liberamente; se l'uso del prodotto deve sottostare a restrizioni allora tale prodotto non deve essere distribuito affatto.

Seguono i termini e le condizioni precisi per la copia, la distribuzione e la modifica.

11.1.2 LICENZA PUBBLICA GENERICA GNU: TERMINI E CONDIZIONI PER LA COPIA, LA DISTRIBUZIONE E LA MODIFICA

- **0.** Questa Licenza si applica a ogni Programma o altra opera che contenga una nota da parte del detentore del diritto d'autore che dica che tale opera può essere distribuita nei termini di questa Licenza Pubblica Generica. Il termine "Programmà' nel seguito indica ognuno di questi programmi o lavori, e l'espressione "lavoro basato sul Programmà' indica sia il Programma sia ogni opera considerata "derivatà' in base alla legge sul diritto d'autore: cioè un lavoro contenente il Programma o una porzione di esso, sia letteralmente sia modificato e/o tradotto in un'altra lingua; da qui in avanti, la traduzione è in ogni caso considerata una "modificà'. Vengono ora elencati i diritti dei detentori di licenza.
  Attività diverse dalla copiatura, distribuzione e modifica non sono coperte da questa Licenza e sono al di fuori della sua influenza. L'atto di eseguire il programma non viene limitato, e l'output del programma è coperto da questa Licenza solo se il suo contenuto costituisce un lavoro basato sul Programma (indipendentemente dal fatto che sia stato creato eseguendo il Programma). In base alla natura del Programma il suo output può essere o meno coperto da questa Licenza.

- **1.** È lecito copiare e distribuire copie letterali del codice sorgente del Programma così come viene ricevuto, con qualsiasi mezzo, a condizione che venga riprodotta chiaramente su ogni copia un'appropriata nota di diritto d'autore e di assenza di garanzia; che si mantengano intatti tutti i riferimenti a questa Licenza e all'assenza di ogni garanzia; che si dia a ogni altro acquirente del Programma una copia di questa Licenza insieme al Programma.
  È possibile richiedere un pagamento per il trasferimento fisico di una copia del Programma, è anche possibile a propria discrezione richiedere un pagamento in cambio di una copertura assicurativa.

- **2.** È lecito modificare la propria copia o copie del Programma, o parte di esso, creando perciò un lavoro basato sul Programma, e copiare o distribuire queste modifiche e questi lavori secondo i termini del precedente comma 1, a patto che vengano soddisfatte queste condizioni:

  a) Bisogna indicare chiaramente nei file che si tratta di copie modificate e la data di ogni modifica.
  b) Bisogna fare in modo che ogni lavoro distribuito o pubblicato, che in parte o nella sua totalità derivi dal Programma o da parti di esso, sia utilizzabile gratuitamente da terzi nella sua totalità, secondo le condizioni di questa licenza.
  c) Se di solito il programma modificato legge comandi interattivamente quando viene eseguito, bisogna fare in modo che all'inizio dell'esecuzione interattiva usuale, stampi un messaggio contenente un'appropriata nota di diritto d'autore e di assenza di garanzia (oppure che specifichi che si offre una garanzia). Il messaggio deve inoltre specificare agli utenti che possono ridistribuire il programma alle condizioni qui descritte e deve indicare come consultare una copia di questa licenza. Se però il programma di partenza è interattivo ma

normalmente non stampa tale messaggio, non occorre che un lavoro derivato lo stampi.
Questi requisiti si applicano al lavoro modificato nel suo complesso. Se sussistono parti identificabili del lavoro modificato che non siano derivate dal Programma e che possono essere ragionevolmente considerate lavori indipendenti, allora questa Licenza e i suoi termini non si applicano a queste parti quando vengono distribuite separatamente. Se però queste parti vengono distribuite all'interno di un prodotto che è un lavoro basato sul Programma, la distribuzione di questo prodotto nel suo complesso deve avvenire nei termini di questa Licenza, le cui norme nei confronti di altri utenti si estendono a tutto il prodotto, e quindi a ogni sua parte, chiunque ne sia l'autore.
Sia chiaro che non è nelle intenzioni di questa sezione accampare diritti su lavori scritti interamente da altri, l'intento è piuttosto quello di esercitare il diritto di controllare la distribuzione di lavori derivati o dal Programma o di cui esso sia parte.
noltre, se il Programma o un lavoro derivato da esso viene aggregato a un altro lavoro non derivato dal Programma su di un mezzo di memorizzazione o di distribuzione, il lavoro non derivato non ricade nei termini di questa licenza.

- **3.** È lecito copiare e distribuire il Programma (o un lavoro basato su di esso, come espresso al comma 2) sotto forma di codice oggetto o eseguibile secondo i termini dei precedenti commi 1 e 2, a patto che si applichi una delle seguenti condizioni:
a) Il Programma sia corredato dal codice sorgente completo, in una forma leggibile dal calcolatore e tale sorgente deve essere fornito secondo le regole dei precedenti commi 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.
b) Il Programma sia accompagnato da un'offerta scritta, valida per almeno tre anni, di fornire a chiunque ne faccia richiesta una copia completa del codice sorgente, in una forma leggibile dal calcolatore, in cambio di un compenso non superiore al costo del trasferimento fisico di tale copia, che deve essere fornita secondo le regole dei precedenti commi 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.
c) Il Programma sia accompagnato dalle informazioni che sono state ricevute riguardo alla possibilità di ottenere il codice sorgente. Questa alternativa è permessa solo in caso di distribuzioni non commerciali e solo se il programma è stato ricevuto sotto forma di codice oggetto o eseguibile in accordo al precedente punto b).
Per "codice sorgente completo" di un lavoro si intende la forma preferenziale usata per modificare un lavoro. Per un programma eseguibile, "codice sorgente completo" significa tutto il codice sorgente di tutti i moduli in esso contenuti, più ogni file associato che definisca le interfacce esterne del programma, più gli script usati per controllare la compilazione e l'installazione dell'eseguibile. In ogni caso non è necessario che il codice sorgente fornito includa nulla che sia normalmente distribuito (in forma sorgente o in formato binario) con i principali componenti del sistema operativo sotto cui viene eseguito il Programma (compilatore, kernel, e così via), a meno che tali componenti accompagnino l'eseguibile.
Se la distribuzione dell'eseguibile o del codice oggetto è effettuata indicando un luogo dal quale sia possibile copiarlo, permettere la copia del codice sorgente dallo stesso luogo è considerata una valida forma di distribuzione del codice sorgente, anche se copiare il sorgente è facoltativo per l'acquirente.

- **4.** Non è lecito copiare, modificare, sublicenziare, o distribuire il Programma in modi diversi da quelli espressamente previsti da questa Licenza. Ogni tentativo contrario di copiare, modificare, sublicenziare o distribuire il Programma è legalmente nullo, e farà cessare automaticamente i diritti garantiti da questa Licenza. D'altra parte ogni acquirente che abbia ricevuto copie, o diritti, coperti da questa Licenza da parte di persone che violano la Licenza come qui indicato non vedranno invalidare la loro Licenza, purché si comportino conformemente a essa.

- **5.** L'acquirente non è obbligato ad accettare questa Licenza, poiché non l'ha firmata. D'altra parte nessun altro documento garantisce il permesso di modificare o distribuire il Programma o i lavori derivati da esso. Queste azioni sono proibite dalla legge per chi non accetta

questa Licenza; perciò, modificando o distribuendo il Programma o un lavoro basato sul programma, si accetta implicitamente questa Licenza e quindi di tutti i suoi termini e le condizioni poste sulla copia, la distribuzione e la modifica del Programma o di lavori basati su di esso.

- **6.** Ogni volta che il Programma o un lavoro basato su di esso vengono distribuiti, l'acquirente riceve automaticamente una licenza d'uso da parte del licenziatario originale. Tale licenza regola la copia, la distribuzione e la modifica del Programma secondo questi termini e queste condizioni. Non è lecito imporre restrizioni ulteriori all'acquirente nel suo esercizio dei diritti qui garantiti. Chi distribuisce programmi coperti da questa Licenza non è comunque responsabile per la conformità alla Licenza da parte di terzi.

- **7.** Se, come conseguenza del giudizio di un tribunale, o di un'imputazione per la violazione di un brevetto o per ogni altra ragione (anche non relativa a questioni di brevetti), vengono imposte condizioni che contraddicono le condizioni di questa licenza, che queste condizioni siano dettate dal tribunale, da accordi tra le parti o altro, queste condizioni non esimono nessuno dall'osservazione di questa Licenza. Se non è possibile distribuire un prodotto in un modo che soddisfi simultaneamente gli obblighi dettati da questa Licenza e altri obblighi pertinenti, il prodotto non può essere distribuito affatto. Per esempio, se un brevetto non permettesse a tutti quelli che lo ricevono di ridistribuire il Programma senza obbligare al pagamento di diritti, allora l'unico modo per soddisfare contemporaneamente il brevetto e questa Licenza è di non distribuire affatto il Programma.
  Se parti di questo comma sono ritenute non valide o inapplicabili per qualsiasi circostanza, deve comunque essere applicata l'idea espressa da questo comma; in ogni altra circostanza invece deve essere applicato il comma 7 nel suo complesso.
  Non è nello scopo di questo comma indurre gli utenti a violare alcun brevetto né ogni altra rivendicazione di diritti di proprietà, né di contestare la validità di alcuna di queste rivendicazioni; lo scopo di questo comma è solo quello di proteggere l'integrità del sistema di distribuzione del software libero, che viene realizzato tramite l'uso della licenza pubblica. Molte persone hanno contribuito generosamente alla vasta gamma di programmi distribuiti attraverso questo sistema, basandosi sull'applicazione consistente di tale sistema. L'autore/donatore può decidere di sua volontà se preferisce distribuire il software avvalendosi di altri sistemi, e l'acquirente non può imporre la scelta del sistema di distribuzione.
  Questo comma serve a rendere il più chiaro possibile ciò che crediamo sia una conseguenza del resto di questa Licenza.

- **8.** Se in alcuni paesi la distribuzione e/o l'uso del Programma sono limitati da brevetto o dall'uso di interfacce coperte da diritti d'autore, il detentore del copyright originale che pone il Programma sotto questa Licenza può aggiungere limiti geografici espliciti alla distribuzione, per escludere questi paesi dalla distribuzione stessa, in modo che il programma possa essere distribuito solo nei paesi non esclusi da questa regola. In questo caso i limiti geografici sono inclusi in questa Licenza e ne fanno parte a tutti gli effetti.

- **9.** All'occorrenza la Free Software Foundation può pubblicare revisioni o nuove versioni di questa Licenza Pubblica Generica. Tali nuove versioni saranno simili a questa nello spirito, ma potranno differire nei dettagli al fine di coprire nuovi problemi e nuove situazioni.
  Ad ogni versione viene dato un numero identificativo. Se il Programma asserisce di essere coperto da una particolare versione di questa Licenza e "da ogni versione successivà', l'acquirente può scegliere se seguire le condizioni della versione specificata o di una successiva. Se il Programma non specifica quale versione di questa Licenza deve applicarsi, l'acquirente può scegliere una qualsiasi versione tra quelle pubblicate dalla Free Software Foundation.

- Se si desidera incorporare parti del Programma in altri programmi liberi le cui condizioni di distribuzione differiscano da queste, è possibile scrivere all'autore del Programma per chiederne l'autorizzazione. Per il software il cui copyright è detenuto dalla Free Software

Foundation, si scriva alla Free Software Foundation; talvolta facciamo eccezioni alle regole di questa Licenza. La nostra decisione sarà guidata da due scopi: preservare la libertà di tutti i prodotti derivati dal nostro software libero e promuovere la condivisione e il riutilizzo del software in generale.

11.1.3 NESSUNA GARANZIA

- **11.** POICHÉ IL PROGRAMMA È CONCESSO IN USO GRATUITAMENTE, NON C'È ALCUNA GARANZIA PER IL PROGRAMMA, NEI LIMITI PERMESSI DALLE VIGENTI LEGGI. SE NON INDICATO DIVERSAMENTE PER ISCRITTO, IL DETENTORE DEL COPYRIGHT E LE ALTRE PARTI FORNISCONO IL PROGRAMMA "COSÌ COM'È", SENZA ALCUN TIPO DI GARANZIA, NÉ ESPLICITA NÉ IMPLICITA; CIÒ COMPRENDE, SENZA LIMITARSI A QUESTO, LA GARANZIA IMPLICITA DI COMMERCIABILITÀ E UTILIZZABILITÀ PER UN PARTICOLARE SCOPO. L'INTERO RISCHIO CONCERNENTE LA QUALITÀ E LE PRESTAZIONI DEL PROGRAMMA È DELL'ACQUIRENTE. SE IL PROGRAMMA DOVESSE RIVELARSI DIFETTOSO, L'ACQUIRENTE SI ASSUME IL COSTO DI OGNI MANUTENZIONE, RIPARAZIONE O CORREZIONE NECESSARIA.

- **12.** NÉ IL DETENTORE DEL COPYRIGHT NÉ ALTRE PARTI CHE POSSONO MODIFICARE O RIDISTRIBUIRE IL PROGRAMMA COME PERMESSO IN QUESTA LICENZA SONO RESPONSABILI PER DANNI NEI CONFRONTI DELL'ACQUIRENTE, A MENO CHE QUESTO NON SIA RICHIESTO DALLE LEGGI VIGENTI O APPAIA IN UN ACCORDO SCRITTO. SONO INCLUSI DANNI GENERICI, SPECIALI O INCIDENTALI, COME PURE I DANNI CHE CONSEGUONO DALL'USO O DALL'IMPOSSIBILITÀ DI USARE IL PROGRAMMA; CIÒ COMPRENDE, SENZA LIMITARSI A QUESTO, LA PERDITA DI DATI, LA CORRUZIONE DEI DATI, LE PERDITE SOSTENUTE DALL'ACQUIRENTE O DA TERZE PARTI E L'INABILITÀ DEL PROGRAMMA A LAVORARE INSIEME AD ALTRI PROGRAMMI, ANCHE SE IL DETENTORE O ALTRE PARTI SONO STATE AVVISATE DELLA POSSIBILITÀ DI QUESTI DANNI.

FINE DEI TERMINI E DELLE CONDIZIONI

11.2 Letture, link e libri interessanti

- La storia del progetto GNU (di Richard Stallman) - tratto da www.gnu.org
  `http://www.gnu.org/gnu/thegnuproject.it.html`

- Perchè il software libero non deve avere padroni (di Richard Stallman) - tratto da www.gnu.org
  `http://www.gnu.org/philosophy/why-free.it.html`

- Il trionfo dell'anarchia: Il Software Libero e la Morte del Diritto d'Autore - Traduzione di Francesco Paparella (fatta per il Pluto Journal) dell'articolo "Anarchism Triumphant: Free Software and the Death of Copyright" di Eben Moglen, apparso per la prima volta in First Monday, peer-reviewed journal on the internet vol. 4, n. 8.
  `http://www.pluto.linux.it/journal/pj0201/anarchism-it.html`

- La cattedrale ed il Bazar (di E. Raymond)
  `http://www.unimo.it/corsi/linux/presentazione/cattedrale/cathedral-bazaar.html`

- Piedone il Finnico - Articolo scritto nell'Aprile 1998, riprodotto col permesso del "Giornale della natura"
  `http://www.linux.it/GNU/articoli/piedone.shtml`

- Spaghetti Hacker - di Stefano Chiccarelli, Andrea Monti, Ed. Apogeo

- Hacktivism - La libertà nelle maglie della rete - di Arturo Di Corinto e Tommaso Tozzi
- Italian Crackdown - di Carlo Gubitosa e Associazione Peacelink, Ed.Apogeo
- L'etica hacker - di Pekka Himanen con prefazione di Linus Torvalds, Ed. Feltrinelli
- M.F.Banahan, A.Rutter - UNIX - The Book - Sigma Technical Press, Wilmslow, Gran Bretagna, 1982
- L.Blackburn, M.Taylor - POCKET GUIDE UNIX - Pitman Publishing LT D, 1984
- H.Hahn - Mastering XENIX on the IBM PC - Scott, Foresman & Co., Glenview (USA) 1986
- Paolo Attivissimo - Da Windows a Linux - `http://www.attivissimo.net` - 2000
- Manuale per i nuovi utenti - Asbesto, Martin, Vinx - FreakNet MediaLab 1999 `http://medialab.dyndns.org/manuale.txt`

Capitolo 12

S.O.S. (S.ave O.ur S.hells)

Non per volervi demoralizzare subito, però sappiate che anche chiedere aiuto richiede un certo "skill" (abilità in english). Ci sono due modi di ottenerlo: uno è chiedere direttamente al computer e l'altro è di interrogare un esperto. Imparerete presto a saper scegliere tra le due forme, riuscendo a capire quale è più conveniente in ogni specifica situazione.

12.1 L'aiuto endogeno

12.1.1 Spingere un programma ad essere cordiale

Come avete notato, Unix non è molto loquace di per sé, ma se lo spingete a parlare qualcosa ve la dice. Talvolta anche troppo. Lanciare un programma dando come unico argomento - - help può esservi di una qualche utilità.

```
$ ls - - help
Usage: ls [-acdfgilnqrstu1ACDFLMRTX] [file ...]
$
```

Su altri sistemi il comando *ls –help* genera una esaurientissima pagina.

Provare per credere!

Qui, per esigenze tipografiche, abbiamo preferito mostrarvi la risposta di ls nella sua versione per Minix (una versione di Unix scritta da quel fanatico di Tanenbaum che mette ancora in giro ed per «ragioni affettive»). Sinteticamente il programma illustra quali sono i caratteri validi per le sue opzioni. Dovrete accontentarvi finché andremo avanti a fotocopie oppure non sperimentate voi stessi il comando. Lo stesso comando su Linux ci racconta il significato di ogni possibile opzione, ma avrebbe preso da solo tre pagine. Fate caso alle parentesi quadre tra cui stanno racchiuse alcune opzioni (in questo caso tutte, meglio così) e gli eventuali percorsi o file. Una cosa [tra parentesi quadre] vuol dire che la si può mettere o che si può non metterla, a discrezione dell'utente. Ovviamente bisogna metterla (e nella riga di comando senza le [parentesi quadre]) se si vuole che quella precisa opzione sia attiva. Insomma, quelle parentesi sono solo una notazione per dire, scusate il gioco di parole, che l'opzione è opzionale!

12.2 Il manuale in linea

C'è un programma che si occupa di darvi informazioni su tutti gli altri. Questo comando è *man* e sta per manual. L'uso base del manuale è come nell'esempio che segue:

```
$ man su
SU(1)        Minix Programmer's Manual     SU(1)
NAME
    su - temporarily log in as superuser or another user
SYNOPSIS
    su [name]
EXAMPLES
    su      # Become superuser
    su ast  # Become ast
DESCRIPTION
Su can be used to temporarily login as another user. It prompts
for the superuser password. If the correct password is entered,
su creates a shell with the desired uid. If no name is
specified, root is assumed. To exit the temporary shell, type
CTRL-D. When memory is tight, it is better to become superuser
by logging out and then
Standard-input, 1-23 (Top)
```

A gestire la visualizzazione del manuale in Linux ci pensa `less` o qualche suo parente prossimo (a proposito, vi sarebbe mai venuto in mente che less è una specie di figlio naturale dell'ostico `vi`?), per cui si torna indietro e si va avanti esattamente come avete imparato a fare esercitandovi con la Divina Commedia.

Potrà tornarvi utile sapere che premendo uno / potete inserire una parola da ricercare all'interno del testo visualizzato.

Un'altra cosa utile riguarda invece la riga di comando per man. Digitando *man -k parola_chiave*, otterrete l'elenco delle pagine di manuale che contengono la parola che cercate. Il programma man l'andrà a ricercare tra i testi dei documenti che conosce, riferendovi dove l'ha trovata. Scegliete una buona parola chiave che non vi restituisca cento titoli diversi e poco azzeccati. Imparate a saper leggere nel manuale e tenete presente che ogni documento è spesso diviso negli otto paragrafi illustrati nella tabella seguente:

1	Comandi che si danno al prompt della shell
2	Suggerimenti per la programmazione in C (ostico!)
3	Chiamate di sistema da conoscere durante la programmazione in C (ancora più ostico!)
4	Le periferiche (tutti i file che stanno nella /dev)
5	Configurazione dei programmi
6	Giochi
7	Cose che se non fossero piazzate qua non si saprebbe proprio dove metterle
8	I comandi della shell per lamministratore di sistema
9	I trucchi del kernel (illeggibile per i comuni mortali!)

Quando non viene specificato a man in quale sezione andare a guardare lui comincia dalla prima in progressione verso la nona e vi restituisce la prima pagina che trova (ovviamente tra quelle che hanno lo stesso titolo che state cercando). Per andare a cercare informazioni su come funziona socket (anche se a voi probabilmente non interessa sapere neanche cosa sia) direttamente alla seconda sezione, lanciate man in questa forma:

```
$ man 2 socket
```

12.2.1 Favorisca documenti

Se si vogliono maggiori informazioni di quelle trovate nelle pagine del manuale bisogna, ahimé, andarseli a cercare da soli. Arrampicandovi sull'albero di Linux, dovrete ridiscendere fino alla directory */usr/doc*. Da lì partono altre directory e solitamente ce n'è una per ognuno dei programmi più importanti. Ecco alcune tra le più comuni.

```
$ cd /usr/doc
$ ls
BIND-8.1.2-REL/
Linux-HOWTOs/
Linux-mini-HOWTOs/
MAKEDEV-C-1.6.1/
SoundStudio/
WorkBone-2.31/
autoconf-2.13/
automake-1.4/
bash/
bootutils/
bzip2/
diffutils/
eject-1.5/
elvis-2.1@
faq/
fdutils-5.2/
fetchmail-4.6.3/
freefont/
fvwm2-2.0.46-BETA/
gettext-0.10.35/
ghostscript/
gimp-1.0.4/
Gnuchess-4.0.pl79/
grep-2.3/
groff/
ircii-4.4/
isapnptools-1.18/
joe2.8/
koules/
less-332/
libmpeg-1.2/
libtool-1.2/
lilo@ mailx/
mc-4.1.35/
minicom-1.82-3/
net-tools-1.52/
netwatch/
nfs-server-2.2beta40/
pciutils-1.10/
perl5.005_03/
php3/
pine4.10/
portmap_5beta/
ppp-2.3.7/
pppsetup/
```

```
procinfo-16/
seejpeg/
sendmail/
shadow-19990307/
sox-11gamma-cb3/
sudo/
svgalib-1.3.1/
tar/
tcsh/
termcap-2.0.8/
unzip-5.40/
util-linux-2.9i/
vi/
wget-1.5.3/
wu-ftpd-2.4.2-vr16/
x3270/
xboard-4.0.2/
xfractint/
zip-2.2/
ziptool-1.0/
```

A questo punto basta entrare nella dir che ha un nome vagamente evocativo di quello che stiamo cercando e dare un'occhiata ai documenti che vi si trovano. Di solito questi documenti contengono puro testo, ma in alcuni casi sono file con estensione *.html*, cioè leggibili tramite un browser (lo stesso tipo di programma con cui si va in giro per web su Internet), e in altri casi hanno l'estensione *.ps* e si possono solo stampare (ps in questo caso sta per PostScript, un linguaggio che capiscono solo le stampanti).

12.2.2 Il Che fare e il Che fare in edizione tascabile

Tra le directory che partono da */usr/doc* ce ne sono due di particolare importanza: *Linux-HOWTOs* e *Linux-mini-HOWTOs*.

Queste directory contengono file di solo testo che non parlano di un particolare programma, ma di una situazione precisa. Per esempio trovate HowTo su come configurare la rete o su come visualizzare caratteri cirillici o ebraici.

La differenza tra gli HowTo e i MiniHowTo è solo la dimensione del documento. Entrambi sono (solitamente) pieni di esempi utili che vi fanno capire meglio delle asettiche descrizioni del manuale.

Spesso, per risolvere un problema, basta ricopiare al posto giusto gli esempi trovati negli HowTo e nei Mini e riadattarli con poche modifiche alla macchina sulla quale state lavorando.

L'HowTo viene scritto da qualcuno che si è trovato nello stesso guaio prima di voi e ora vi spiega come uscirne rapidamente e senza impazzire.

Che bel vantaggio arrivare secondi!

12.3 Soluzioni esogene

Se proprio non siete riusciti a risolvere il problema non vi resta che giocarvi l'ultima carta: chiedere aiuto a un tecnico in carne ed ossa.

Di solito questa richiesta avviene in uno di questi tre modi:

- Per telefono;
- Per posta elettronica;

- Di persona.

Ognuno di questi metodi ha delle regole che vanno rispettate.

12.3.1 Hot line Unix

Quando telefonate a qualcuno è bene essere quanto più precisi possibile. Considerate la difficoltà che ha chi vi ascolta di dovere immaginare le cose che state descrivendo.

Espressioni del tipo: «Il computer non funziona» sono di utilità nulla per chi vuole darvi una mano.

Anche «Quando lancio il programma scrive qualcosa e poi si chiude» non è di alcun aiuto: in questo caso dovreste leggere e riferire attentamente il messaggio di errore incontrato.

Non siate generici, ma non affrontate il linguaggio tecnico se non lo conoscete: un vostro errore nella descrizione del problema può sviare facilmente dalla diagnosi e quindi dalla soluzione.

12.3.1.1 Scrivere tecnicamente

Vi servono ancora un paio dei nostri incontri prima di essere in grado di scrivere lettere elettroniche. Ma è bene cominciare subito a farvi un'idea di come scriverle, almeno per quanto riguarda le richieste d'aiuto. Un HowTo o un programma è sempre firmato dal suo autore con l'indirizzo di posta. Quindi consideratelo la fonte autentica di informazione. Le regole sono le stesse delle telefonate, solo che in più stavolta avete la possibilità di riportare esattamente con un copia e incolla l'eventuale messaggio di errore che incontrate. Ricordate sempre due cose: descrivete con attenzione e ricchezza di particolari la versione di Unix sulla quale vorreste far funzionare le cose; leggete sempre in maniera attenta la documentazione relativa al vostro problema: se nei documenti c'è già la soluzione, probabilmente non riceverete mai una lettera di risposta.

12.3.1.2 L'incontro col guru

Le persone vicino a voi che vi aiutano (il mago di turno o persino l'amministratore di sistema) ascolteranno più volentieri le vostre suppliche se per loro ne vale la pena.

I contanti sono un po' volgari, ma potete esprimere la vostra gratitudine in molti altri modi:

- *il cibo è la bustarella più affidabile; questo argomento è trattato con maggiori dettagli più avanti in questo capitolo;*
- *fiori, stupidi piccoli soprammobili per la scrivania, adesivi, o altri oggetti del genere ricordano al guru che utente grato e gentile siete;*
- *se siete una persona attraente del sesso opposto, un bell'abbraccio o bacio sulla guancia qualche volta è appropriato, sebbene i più goffi possano reagire in modo eccessivo e rispondervi con una sincera proposta di matrimonio.*

Siate discreti. È sempre bene scrivere una nota entusiasta al capo della persona in questione e dire quale aiuto meraviglioso avete ricevuto. Questo metodo non va bene se aiutarvi non è ciò che questa persona dovrebbe fare. (Lei è troppo intelligente per perdere il suo tempo ad aiutare semplici utenti.)

12.3.1.3 I tecnici e il cibo

Uno stereotipo ampiamente sostenuto dice che i tecnici mangiano solo cibi che si trovano nei distributori automatici, in particolare le bibite. In alcuni casi questo è ancora vero, ma non contateci. Con l'invecchiamento dei baby boom, le persone che vivevano di Turbo Cola (la marca che è tutto zucchero e doppia caffeina) scoprono che causa spesso una varietà di sintomi fisici che non è necessario descrivere in dettaglio; sono passati alla Perrier perché è meglio per il fegato. Una intera nuova generazione di tecnici Unix è passata ai cibi macrobiotici. Cercate di conoscere il vostro tecnico. Un regalo di ringraziamento con un tocco

personale è sempre apprezzato (per esempio, pasticcini o biscotti). Alcune persone non capiscono la differenza tra i biscotti fatti in casa e quelli che provengono da un tubo di pasta pronta che comprate in negozio, tagliate a fette e cuocete (che è molto più semplice che farli da zero). Se utilizzate pasta pronta, ricordatevi di togliere la plastica prima di tagliarla; altrimenti vi fate scoprire. Alcune persone non lo noterebbero, ma non contateci troppo.

Pizza

Il cibo più apprezzato dai tecnici Unix è la pizza, specialmente quando arriva all'ora di pranzo in un giorno in cui il guru sarebbe dovuto uscire a mangiare con una temperatura di 34 gradi o sotto una pioggia mista a neve. Al giorno d'oggi, con la consegna della pizza a domicilio, nemmeno voi dovete uscire sotto la pioggia mista a neve; ricordatevi di fare un paio di telefonate alle 11:30 del mattino. La scelta della farcitura della pizza è molto importante. Alcuni tipi di maghi Unix hanno come regola generale il non mangiare mai niente di verde, il che esclude ingredienti famosi come gli spinaci e i broccoli. Alcuni tipi molto istruiti, specialmente i laureati di scuole tecniche, hanno una coscienza superiore e non mangiano nessuna pizza a meno che il formaggio provenga da una cooperativa casearia volontaria che appartiene alle mucche stesse. D'altro canto, abbiamo conosciuto dei severi vegetariani che, per il progetto della pizza, hanno dichiarato gli spinaci come farcitura onoraria. La cosa migliore da fare, almeno fino a quando non conoscete i gusti dei vostri amanti di pizza, è di ordinare diversi tipi di pizza. Il concetto di troppa pizza è sconosciuto nell'ambiente dei tecnici, quindi lasciatevi guidare dal portafoglio.

Cibo cinese

Una possibile alternativa alla pizza è il cibo cinese, un'altra categoria di cibi che può essere consegnata con una temperatura di 34 gradi. Un tipico laureato conosce quasi tutto della cucina orientale e non ha molta pazienza con i cibi cinesi di qualità scadente. Il cibo cinese come ringraziamento è una buona idea solo se è veramente buono; in questo la pizza è molto più affidabile. È possibile rovinare una pizza mettendoci sopra dei crauti, qualcosa che abbiamo scoperto in una pizzeria, su in Val Badia, gestita da qualcuno di nome Heidi (un nome poco propizio per un pizzaiolo); però a parte questo, la pizza è sicurissima.

(Tratto da John R. Levine e Margaret Levin Young,
Unix for Dummies, IDG Books Worldwide, San Mateo, 1993)

Parte V

Il computer e la società

Capitolo 13

Non siamo soli

Forse sarà la centesima volta che ce lo sentite dire, ma ripeterlo non guasta, specie perché per noi è un motivo d'orgoglio: ogni Unix è un computer multiutente.

Cioè sullo stesso computer possono essere presenti simultaneamente più operatori.

13.1 Cose per le quali ci bastiamo

13.1.1 Facciamo l'appello

Se vi viene voglia di sapere chi sta usando il computer oltre a voi lanciate il comando *who*:

```
$ who
luther   tty1 Tue Nov 30 15:15
shining  tty2 Tue Nov 30 14:22
luther  ttyp0 Tue Nov 30 15:15 (192.168.3.3)
quest   ttyp1 Tue Nov 30 15:19 (192.168.3.1)
asbesto ttyp2 Tue Nov 30 12:20 (192.168.3.3)
$
```

Cinque utenti sono presenti in questo momento sul sistema.

Due hanno proprio il computer davanti (il primo luther e shining): si riconosce sulla seconda colonna dalla tty (letteralmente telescrivente, TeleTYper, perché in origine si usavano proprio queste come terminali). Ogni Unix chiama a modo suo ogni diversa tty, ma state pur sicuri di due regolette:

- Se dopo tty cè solo un numero, ci si sta riferendo quasi sicuramente alla tastiera e al monitor del computer;
- Se dopo tty, numero a parte, c'è solo una p, allora si tratta di una pseudo-tty, cioè un collegamento proveniente da un altro computer attraverso la rete o da un terminale all'interno dell'ambiente grafico.

13.1.2 I bigliettini sotto il banco

Potreste avere riconosciuto qualcuno e essere presi dall'irresistibile voglia di dirgli qualcosa al volo. Esiste un programma che si chiama *write* utile a questi scopi:

```
$ whoami
shining
```

```
$ write luther ttyp0
ciao!
^D
EOT
$
```

Il comando *whoami* è stato usato solo per far vedere che questo frammento di schermata appartiene a shining (*whoami* può sempre farvi uscire da una crisi di identità). Per scrivere a luther è bastato digitare il comando *write* seguito dal nome dell'utente a cui mandare il messaggio. Poiché luther era presente ben due volte, è stato necessario anche specificare su quale console (userete questo termine come sinonimo di tty) farglielo arrivare.

Dopo aver digitato il comando e premuto il solito invio, shining ha cominciato a scrivere il testo. Per fermarsi ha premuto ^*D* (si vede bene sullo schermo) e il programma gli ha risposto con un *EOT (End of Text)* prima di restituirgli il prompt della shell. Quello che segue è ciò che è apparso sulla console di luther:

```
$ Message from shining (tty2) Tue Nov 30 15:38:02 1999...
Ciao!

EOT
```

Anche luther è stato avvisato che shining ha smesso di scrivere.

A sua volta luther potrà rifare le stesse operazioni.

13.1.3 I muri bianchi non fanno pensare

C'è anche una tecnica per far arrivare a tutti i vostri messaggi. Basta digitare il comando *wall* e cominciare a scrivere. Quando si è fatto, il ^*D* vi ridarà il prompt.

```
$ wall
unix è bello
Broadcast Message from luther@frankenstein (/dev/tty2) at 16:17
...
unix è bello
$
```

Luther, ha fatto arrivare il messaggio a tutti e, come si vede nella schermata, anche a se stesso.

13.2 Cose per le quali serve il resto del mondo

13.2.1 Il protocollo diplomatico

Nella babele dei sistemi operativi e dei linguaggi diversi, più di trenta anni fa è nata l'idea di una rete di computer capace di superare le differenze e fare comunicare tutti con una lingua comune.

Le lingue con cui i computer parlano tra loro si chiama protocolli di comunicazione. Manco a dirlo, tutto cominciò per esigenze militari (gli americani volevano che i loro computer restassero collegati anche in caso di attacco nucleare e quindi di distruzione di alcuni nodi della rete): fu così che nacque *ARPAnet*.

Le università che contribuirono alla sua nascita si tennero il giocattolo e i militari si rifecero la rete per i fatti loro. Oggi il giocattolo si chiama internet ed è andata ben oltre le università. Unix è coetaneo di internet e, di fatto, sono cresciuti assieme.

13.2.1.1 TCP/IP e altre cose impronunciabili

Il protocollo su cui, oggi come ieri, si regge internet si chiama *TCP/IP (Transfer Communication Protocol / Internet Protocol)*. I calcolatori Unix offrono pieno supporto a questo standard ed è la loro lingua madre, anche quando non sono collegati a internet. Dal punto di vista funzionale non c'è alcuna differenza tra la rete del FreakNet MediaLab e il resto di internet: ecco perché quando uno degli host (computer facente parte di una rete) è connesso, automaticamente tutto il laboratorio è su internet. Quando non lo è, i computer continuano a parlarsi tra loro con le stesse identiche regole.

13.2.1.2 Non solo web

Attraverso il TCP/IP si possono fare passare informazioni e dati di tutti i tipi: posta elettronica, pagine ipertestuali, articoli di gruppi di discussione, ecc. Tutti questi diversi tipi di traffico si chiamano *servizi*. Di solito i servizi sono pensati secondo la tecnologia client-server. Niente di spaventoso: vuol dire semplicemente che per accedere a un servizio offerto da un host (che in questo caso svolge il ruolo di server), dovete avere il programma client adatto.

13.2.2 Il dito che punta

Non avrete mai accesso a tutti i computer dell'universo, né c'è da aspettarsi che troverete dei disponibili account per luther. Ma c'è un modo per conoscere chi siano gli utenti presenti su un host: usare il servizio *finger*. Il client sul vostro computer per fortuna si chiama finger pure lui (uno dei pochi casi in cui si può non fare confusione!). Il programma finger si usa seguito da una chiocciola a cui attaccare il nome del computer del quale volete informazioni.

```
$ finger @frankenstein
[frankenstein.freaknet.org] Please supply a username
$ finger shininf@frankenstein
[frankenstein.freaknet.org]
Login: shining       Name: Shining      Directory: /home/shining
      Shell: /bin/bash
On since Tue Nov 30 16:08 (CET) on tty3 1 minute 20 seconds idle
On since Tue Nov 30 16:07 (CET) on tty4 1 minute 36 seconds idle
No mail.
Project: My project is yours. Our project concerns the FreakNet
MediaLab
Plan: Hack the planet!
$
```

Sorpresa. Alcuni amministratori di sistema ligi (forse un po' troppo) alla privacy impediscono che possiate ottenere informazioni sugli utenti presenti sul sistema. Per questo vi chiedono di indicare un nome utente esistente. Questo username si mette prima della chiocciola (tutto l'argomento finisce per assomigliare a un indirizzo di e-mail). Tutto quello che vedete nel campo project è contenuto nel file .project che sta nella home directory di shining sull'host frankenstein.

Se volete fare apparire qualcosa in quel campo al posto del laconico no project dovrete modificare o creare un file .project nella nostra home. Discorso identico per il campo plan che è contenuto nel file *~/.plan.*

Ormai siete abbastanza esperti di vi, emacs e joe da potervi divertire a sparare le vostre c@zz@t& in rete, così come ha fatto shining.

13.2.3 La chiacchierata in interurbana

Ora che avete scoperto chi c'è su quel computer laggiù in Australia (contiamo presto di estendere la rete del FreakNet MediaLab) potete tentare un approccio diretto chiamandolo in *talk*.

```
$ talk asbesto@kangaroo.freaknet.org
```

La sintassi è la stessa del finger: *talk username@host*

Questo è invece il messaggio che apparirà sullo schermo di asbesto:

```
$
Message from Talk_Daemon@kangaroo at 16:20 ...
talk: connection requested by luther@smarterm.freaknet.org.
talk: respond with: talk luther@smarterm.freaknet.org
```

Il talk ha la gentile abitudine di spiegarci sempre come fare ad accettare la conversazione. Bisogna semplicemente digitare quello che ci dice dopo il *respond with:*. Quindi in questo caso asbesto dovrà immettere al prompt:

```
$ talk luther@smarterm.freaknet.org
```

A questo punto lo schermo si spaccherà in due e si potrà cominciare a digitare in diretta. Cioè ognuno vedrà quello che sta scrivendo l'altro in quel momento (linee permettendo).

13.2.4 Il mondo è isolato

Leggete con più attenzione del solito questo paragrafo. Su alcuni sistemi è attiva un'opzione che impedisce alla shell di farvi arrivare messaggi dal mondo esterno (e anche interno) mandati da write, wall e talk. Per restituire la giusta dimensione sociale al vostro colloquio con Unix, dovrete dare il comando che segue.

```
$ mesg y
```

Come avete già visto, l'interprete dei comandi legge come prima cosa il file *.profile* della vostra ~ e esegue quello che vi trova (il file *.profile* è un cosiddetto script di shell) e solo dopo vi dà il prompt. La cosa migliore da fare è aggiungere alla fine del vostro *.profile* il comando *mesg y* in maniera tale che venga automaticamente eseguito a ogni inizio di colloquio e voi non dobbiate ridigitarlo ogni volta (in questo modo non dimenticherete di farlo).

Capitolo 14

Il postino elettronico

Ormai siete abituati alla cortesia di Unix, che all'avvio del colloquio vi informa sull'eventuale presenza di posta. La posta può esservi stata spedita da un altro utente dello stesso computer, della stessa rete o può arrivarvi da molto lontano attraverso internet.

14.1 I regolamenti postali

Ovunque mandiate la posta le regole sono sempre le stesse. La prima regola è quella di scrivere correttamente l'indirizzo. Tutti gli indirizzi sono nel formato *nomeutente@nomecomputer* Di solito *nomecomputer* viene indicato con tutto il percorso. È lo stesso concetto di identificare un certo file all'interno dell'albero delle directory.

14.1.1 L'indirizzo pesante

Ecco un indirizzo e-mail reale (non abusatene!): t*opo@mail1.dba.unict.it*

Il peso di un indirizzo e-mail decresce da destra verso sinistra dopo ogni . (*punto, dot*):

- *it*: identifica la zona di internet amministrata dall'autorità italiana; tecnicamente è il *dominio di primo livello* ed esiste sempre;
- *unict*: la rete dell'università di Catania, che è sottoposta alle regole dell'autorità italiana per internet; in termini tecnici è il *dominio di secondo livello* e anche questo esiste sempre; quando sentite qualcuno dire una cosa tipo voglio «registrare il dominio», si sta riferendo proprio al dominio di secondo livello;
- *dba*: all'interno della rete universitaria di Catania c'è la sottorete del Dipartimento di Biologia Animale; ovviamente questo è chiamato *dominio di terzo livello*, ma può non esistere: non è detto che ci siano sempre delle sottoreti;
- *mail1*: sulla sottorete del Dipartimento di Biologia Animale cè *l'host* mail1; questo non è un dominio, ma il nome vero e proprio del computer;
- @: l'italiana "chiocciola" si chiama *at* nella lingua madre dell'informatica; ha più o meno il significato di *"presso"* e separa il nome del computer dal nome dell'utente;
- *topo*: è il nome dell'utente; spesso coincide con il nome usato per il login, ma non sempre. Per questa parte dell'indirizzo è molto importante utilizzare gli stessi caratteri minuscoli e MAIUSCOLI con cui vi è stato dato (minuscolo e MAIUSCOLO è del tutto indifferente a destra della @).

Nessuna delle parti di un indirizzo di posta elettronica può contenere spazi. Fate caso che un indirizzo di posta elettronica non è molto differente da un indirizzo di *snail mail* (posta lumaca, quella di carta). Shining, 7332, via Etnea, Catania, Italia è simile a shining@7332.etnea.catania.italia.

14.1.2 Lo scambiatore di posta

Il vostro indirizzo *luther@voyanet.org* sembra non riferirsi a nessun computer. In realtà dietro c'è un trucco: alcuni computer su internet sanno che la posta spedita come *@voyanet.org* deve essere reindirizzata a un vero host come *mail.voyanet.org*. Quindi la regola nomeutente@nomecomputer rimane sempre valida.

14.2 Errare è umano ed anche digitale

Leggete sempre con molta attenzione i messaggi che provengono da *postmaster*, *mail daemon*, *mail server* e simili. Un computer sta cercando di segnalarvi che una vostra e-mail non è giunta a destinazione. Cercherà anche di aiutarvi spiegandovi perché, ma sempre a modo suo!

14.2.1 Avviso gratuito. Attenzione: l'utente è sconosciuto!

Ricevere un messaggio che vi dica una cosa del tipo *user unknown* vuol dire che avete spedito un messaggio sbagliando qualcosa nella parte a sinistra dell'indirizzo. Guardate con attenzione a chi avete spedito il messaggio (lo troverete scritto all'interno della lettera che avete ricevuto dal server di posta del vostro amico), trovate l'errore e rispedite la lettera al destinatario corretto. Se avevate scritto l'indirizzo così come lo avevate segnato allora vi è stato dato male. Ricontattate con altri mezzi chi ve lo ha fornito e chiedetegli di darvelo giusto.

14.2.2 Avviso gratuito. Attenzione: il computer selezionato è inesistente!

Se invece ricevete un messaggio che contenga qualcosa come *host unknown* avete sbagliato qualcosa nella parte a destra della chiocciola. Stavolta è stato il vostro server di posta a dirvi che non è riuscito a trovare il suo collega. Fate le solite verifiche e rispedite all'indirizzo corretto.

14.2.3 Avviso gratuito. Attenzione: il computer da voi chiamato potrebbe essere spento!

Un messaggio del tipo *"Your message was not delivered because the destination computer was not reachable within the allowed queue period"* non vuol dire che abbiate sbagliato voi qualcosa, ma che per qualche ragione non è stato possibile contattare il computer che avrebbe dovuto ricevere la lettera. Non potete farci niente. State attenti alle intenzioni del vostro server di posta: se è passato poco tempo vi dirà qualcosa che significa più o meno «attento, io non ho ancora contattato il tuo amico dopo più di 4 ore. Ci riproverò per cinque giorni. NON CÈ BISOGNO DI RISPEDIRE LA LETTERA». Altrimenti potreste ricevere un avviso del tipo «Mi arrendo. Dopo 5 giorni non sono riuscito a consegnare il messaggio. Se ne hai proprio voglia, PROVA A RISPEDIRLO». Ovviamente ve lo dirà in inglese.

14.3 I mailer

Non è molto utile scendere nei dettagli del funzionamento di un programma per la posta elettronica (*mailer*). Ne troverete tanti, diversi, ma tutti vi chiederanno di inserire un destinatario e vi permetteranno di scrivere il messaggio in se e il suo *subject* (letteralmente *oggetto*, una specie di titolo). Il più diffuso è *mail* e un altro molto popolare è *elm*. C'è chi riesce a usare emacs per questo scopo (non chiedeteci come si fa: noi non ci siamo mai riusciti), ma al FreakNet MediaLab lo standard è usare *pine* (una versione molto aggiornata del vecchio *elm*). Imparerete ad adattarvi a tutti i programmi. Per comodità abbiamo riportato una guida sintetica all'uso delle principali funzioni di pine).

14.3.1 Breve manuale d'uso del programma di posta elettronica PINE

realizzato da Vinx <vinx@freaknet.org>
leggermente riadattato da Asbesto <asbesto@freaknet.org>

Pine è il programma che permette di leggere e scrivere posta elettronica, sia in locale (cioè all'interno della nostra rete di computer) sia su internet (cioè in tutto il mondo!)

Vedremo passo passo come è possibile fare le seguenti operazioni :

1. Scrivere una email;
2. Leggere le email ricevute.

Allora, iniziamo dalle basi. Una volta che si è fatto il login su un computer, con il proprio nome utente e la propria password, sarà sufficiente digitare *pine* e premere invio, per avviare il programma di posta.

Dopo l'avvio di Pine verrà mostrata una schermata con varie scelte possibili. Quella a cui siamo interessati per prima cosa è la *C, COMPOSE MESSAGE.*

14.3.1.1 Comporre un messaggio

Muovendosi su e giù con le frecce si va su COMPOSE e si batte invio (o si preme direttamente C) per passare alla fase di scrittura di una email. A questo punto ci si trova di fronte una nuova pagina in cui si devono fare in ordine le seguenti operazioni (muovetevi con i tasti cursore su e giù) :

From: immettere il nome che si vuole visualizzato come MITTENTE nel messaggio, ad esempio luther@medialab. Se dovete invece mandare un messaggio su Internet, potete lasciare il nome che vi viene proposto accanto a "From:" che è già il vostro indirizzo internet;

To: è l'indirizzo internet della persona a cui volete scrivere. Basta scrivere la sua email nel caso di un messaggio che va ad internet, oppure l'indirizzo LOCALE della persona a cui scrivete se questa è un account della rete locate (ad esempio, se volete scrivere a "caio" della rete locale scriverete quindi "cario@medialab");

Cc: nel terzo rigo (identificato con Cc) si possono inserire altri indirizzi a cui mandare contemporaneamente lo stesso messaggio, separando tali indirizzi con una virgola. Esempio: vinx@freaknet.org, asbesto@freaknet.org, shining@medialab

Se gli indirizzi sono stati correttamente inseriti (mettendo solo un nome viene automaticamente aggiunto @freaknet.org), con la virgola a separarli, il programma li metterà uno in ogni riga quando si passa al campo successivo. Scrivendo solo il nome, automaticamente questo verrà "completato" con l'aggiunta di "freaknet.org": questo perché Pine è programmato in modo che sia automatico il suo uso per la posta Internet.

Attchmnt: serve se vogliamo mandare un file o un documento insieme al nostro messaggio (ad esempio, una immagine, una foto, un programma, un virus); basterà scrivere il nome del file che si vuole mandare insieme alla email. Non esageriamo con questa opzione, per due semplici motivi:

1. nella situazione in cui questo documento è scritto il servizio ci viene attualmente regalato e ogni carattere che si manda lo paghiamo con la bolletta e lo paga chi ci aiuta a fornirvi questo servizio;
2. la banda non va sprecata!

Subject: Nell'ultimo campo, subject, si inserirà l'equivalente del titolo dell'email che si sta scrivendo. È buona abitudine mettere qualche parola significativa, dato che chi riceve molte email per prima cosa legge i subject e da essi decide quali messaggi leggere prima e quali dopo. Si tratta in definitiva del "titolo" del messaggio.

A questo punto, sempre muovendosi con la freccia in basso o in alto, si arriva nel riquadro dove sarà possibile scrivere il testo dell'email.

Una volta finito il messaggio e comunque durante la scrittura, si hanno varie opzioni. Fondamentalmente a noi servirà mandare il messaggio o non mandare il messaggio. Per mandare il messaggio appena concluso si deve premere *CTRL e X* (tenendo premuto il tasto CTRL, premere una volta X). Il programma chiama questa cosa *Send*, e lo spiega nella barra sul fondo dello schermo, che dà svariati comandi. Quello che stiamo usando è appunto questo.

Per abortire il messaggio scritto (se avete cambiato idea e non volete più spedirlo) si devono premere *CTRL e C* contemporaneamente (CTRL+C). Dopo aver premuto una di queste combinazioni viene chiesto di digitare *Y* (yes) o *N* (no). (spesso i comandi dati a Pine, come CTRL-X e CTRL-C, viene chiesta una conferma: per rispondere basta premere *Y* (yes) o *N* (no)).

14.3.1.2 Leggere i messaggi

Per leggere i messaggi ricevuti l'opzione che ci interessa è la *I*, *MESSAGE INDEX*. Sempre muovendosi con i tasti cursore o digitando semplicemente I si entrerà in una schermata in cui si vedrà la lista dei messaggi ricevuti. Se sono presenti dei messaggi ci si potrà muovere tra essi con le freccette e premendo *invio* sarà possibile leggerlo per esteso. A questo punto, mentre leggete il messaggio, potete decidere di rispondere ad esso, cancellarlo, o rimandarlo a qualcun altro:

1. **Rispondere**: si preme *R*, viene così chiesto se rispondere a tutti coloro inclusi nel messaggio, di solito è normale dire di no. Premuta la N (o la Y) ci troveremo nella fase di scrittura già analizzata prima. Da notare che le frasi scritte dal mittente sono evidenziate con un carattere all'inizio, di solito il segno di maggiore, >. Questo serve ad avere un riferimento per ciò di cui si sta parlando. Con il tempo e l'esperienza capirete.

2. **Cancellare** il messaggio, premendo *D* ed eventualmente ripristinarlo O con la *U* se cambiate idea e non volete più cancellarlo;

3. **Mandare a qualcun altro** il messaggio (forwardare), premendo la *F*. In questo caso sarà come scrivere un messaggio ex novo ma con il testo già scritto.

Capitolo 15

Controllo a distanza

15.1 Se luther non va al computer, è il computer che va a luther!

15.1.1 Come abbiamo pensato di semplificarvi la vita

Non è assolutamente detto che il vostro lavoro sia sul computer che vi sta davanti. Al FreakNet MediaLab non succede quasi mai! Avete presente la parolina magica *"medialab"* che immettete prima del vostro username? Essa non fa altro che evitarvi una noiosa sequenza di comandi che altrimenti dovreste digitare manualmente. Facciamo il caso che non ci sia la possibilità di evitarli.

```
Welcome to Linux 2.2.6.

vostok login: luther
Password:

Linux 2.2.6.
1 failure since last login.
Last was 11:19:27 on ttyp0.
Last login: Wed Dec 01 22:12:52 1999 on tty1.
No mail.

A handful of friends is worth more than a wagon of gold.

luther@vostok:~$ rlogin medialab
Password:
Linux 2.2.10.
Last login: Wed Dec 1 23:11:37 on ttyp1 from viking.freaknet..
You have mail.
Never eat anything bigger than your head.
luther@medialab:~$ exit
logout
rlogin: connection closed.
vostok:~$ exit
logout
```

Ecco i cinque passi seguiti per ottenere questo risultato:

1. Accedere con lo username luther (che è presente su tutti i calcolatori del laboratorio) dalla console fisica di un qualsiasi computer (in questo esempio vostok);

2. Dare il comando *rlogin* (che sta per *remote login*) seguito dal nome del computer a cui si vuole accedere (medialab, giusto?). La forma che è stata usata è *rlogin nomehost*. Questa forma sottoindente che il vostro username sul computer sul quale state cercando di collegarvi sia lo stesso di quello usato sul computer che avete sotto le mani;

3. Inserire la password di luther su medialab (che in questo caso è sempre la stessa). Su alcune reti poco sicure potrebbe non esservi richiesta. Il computer di destinazione potrebbe prendervi per un tipo fidato per la semplice ragione che avete già un accesso su un suo compare;

4. Visto che in effetti non avete niente da fare, uscire con il solito *exit*, o in alternativa con *logout* (c'è una sottile differenza tra le due parole chiave, ma è troppo sottile per questo corso). A questo punto bisogna chiudere anche la shell che avete aperto all'inizio, quella su vostok.

Non vi sarà molto utile entrare su medialab nuovamente come luther. La cosa migliore è specificare il vostro nome al momento di collegarvi. In questo caso bisognerà usare rlogin con l'opzione *-l* seguita dal nome dell'utente che vorreste essere sul computer di destinazione. Tenete presente che al laboratorio la vostra utenza esiste solo su medialab. Guardate l'esempio.

```
luther@vostok:~$ rlogin -l shining medialab
Password:
Linux 2.2.10.
Last login: Thu Dec 2 10:39:02 on ttyp1 from 192.168.1.200.
You have new mail.
You have a truly strong individuality.
shining@medialab:~$ exit
logout

rlogin: connection closed.
vostok:~$
```

15.1.2 Pronto? Cè un vax in casa?

Il programma rlogin è rapido, efficace, occupa poca memoria ma ha un piccolo difetto: sa parlare solo tra Unix. Se avete un accesso su una macchina appartenente a un'altra famiglia, rlogin non va bene e al suo posto si deve usare il telnet. Anche questo programma si usa nella forma *telnet nomehost*. È più pesante di rlogin, occupa più risorse sulla macchina dal quale lo lanciate, ma è universale.

```
$ telnet venus.calstate.edu
Trying...Open
Connected to venus.calstate.edu.
Escape character is '^]'.
Welcome to VAX/VMS V5.3-1
Username: luther
Password:
Welcome to the California State University
VAX 6230 VAX/VMS 5.3-1
Last interactive login on Wednesday, 10-OCT-1990 09:35
$ logout
LUTHER logged out at 16-OCT-1990 13:19:38.82
Connection closed to venus.calstate.edu
$
```

15.2 Facciamolo al telefono

Trasformare il vostro computer in una console remota di un altro calcolatore è possibile non solo attraverso la rete ma anche per mezzo di comuni linee telefoniche. C'è un programma che trovate praticamente dappertutto che si chiama *cu* (sta per call Unix). Se attraverso *cu* fate il numero di un computer (un computer qualsiasi, non per forza uno Unix) che aspetta le chiamate dall'altra parte della linea telefonica, questo programma emulatore di terminale vi permetterà di trasformare il vostro computer nella tastiera-e-monitor di quello chiamato. Sotto Linux *cu* non è molto usato. Ha preso il suo posto *minicom*, un clone del vecchio *Telix* per Ms-Dos (i nostalgici comprenderanno). Una volta, in epoche lontane quanto belle (ai tempi dell'Auro per intendeci), si poteva chiamare lo 095320959, cioè la nostra *BBS* (*Bullettin Board System*, bacheca elettronica).

15.3 Il saccheggio degli archivi

Non solo è possibile lavorare da un capo all'altro della rete come se niente fosse, ma è possibile anche far circolare dei file. Si può prenderne e mandarne con pochi comandi. Il programma client (ricordate che tutto quello di cui parliamo in questa lezione sono servizi TCP/IP) si lancia con *ftp* (sta per *file transfer protocol*).

```
luther@frankenstein:~$ ftp
```

Una volta invocato, ftp mostra il suo prompt. A questo punto dovrete specificare il server al quale collegarvi e fare una procedura di ingresso simile a quella abituale dell'inizio colloquio.

```
ftp> open smarterm
Connected to smarterm.freaknet.org.
220 FTP service ready on smarterm at Thu, 02 Dec 2001 14:35:42 GMT
Name (smarterm:luther):
331 Password required for luther. Password:
230 User luther logged in, directory /usr/luther.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Non è stato necessario specificare il nome utente perché il client ha proposto lo stesso nome di chi ha lanciato il programma (luther) anche per l'accesso sul computer remoto e in questo caso andava bene. È stato necessario comunque inserire la password. La directory remota da cui si comincia a lavorare è la solita home directory.

```
ftp> cd foto
250 CWD command okay.
ftp> pwd
251 "/usr/luther/foto" is current directory.
ftp>
```

I comandi per cambiare directory e per ritrovare la posizione sono i soliti *cd* e *pwd*. Per la lista dei file nel 90% dei casi si potrà usare il solito ls. Ma il comando standard è il seguente:

```
ftp> dir
200 Port command okay. 150 File LIST okay. Opening data connection.
Total 0
```

```
-rw-r--r-- 1 luther other 0 Nov 18 15:43 carmelo.jpg
-rw-r--r-- 1 luther other 0 Nov 18 15:43 iano.jpg
-rw-r--r-- 1 luther other 0 Dec 2 14:39 leggimi.txt
-rw-r--r-- 1 luther other 0 Nov 18 15:44 mara.jpg
-rw-r--r-- 1 luther other 0 Nov 18 14:47 pippo.jpg
-rw-r--r-- 1 luther other 0 Nov 18 14:47 turi.jpg 226
Transfer finished successfully. 0.32 KB/s
ftp>
```

Se volete prelevare il file leggimi.txt dovrete prima, con il comando *ascii*, specificare che i file con i quali volete lavorare contengono solo testo. Poi richiamarlo con il comando *get.*

```
ftp> ascii
200 Type set to A.
ftp> get leggimi.txt
local: leggimi.txt remote: leggimi.txt
200 Port command okay.
150 File leggimi.txt okay. Opening data connection.
226 Transfer finished successfully. 0.00 KB/s
ftp>
```

Per tutti gli altri tipi di file invece bisognerà specificare il tipo *binary.*

E per prelevare più di un file alla volta il comando è *mget*, seguito da un espressione solitamente con i caratteri jolly.

```
ftp> binary
200 Type set to I.
ftp> mget *.jpg
mget carmelo.jpg? y
200 Port command okay.
150 File carmelo.jpg okay.
Opening data connection.
226 Transfer finished successfully. 0.00 KB/s
mget iano.jpg?
```

Per mettere un file, invece di prelevarlo, il comando è *put*. Attenzione a quale è la modalità attiva tra ascii e binary: se viene spedito in maniera scorretta, il file può risultare inutilizzabile. Se siete in dubbio ridate ascii o binary, a seconda dei casi.

```
ftp> put lab.tif
local: lab.tif remote: lab.tif
200 Port command okay.
150 File lab.tif okay.
Opening data connection.
226 Transfer finished successfully. 0.00 KB/s
ftp>
```

Esiste anche il comando *mput*, omologo di *mget*. Se volete evitare che vi venga chiesta conferma prima della spedizione di ogni file, usate il comando *prompt*. Ogni volta che lo si immette viene attivata o spenta la modalità interattiva.

```
ftp> prompt
Interactive mode off.
ftp> mput *.gif
local: napoli.gif remote: napoli.gif
200 Port command okay.
150 File napoli.gif okay.
Opening data connection.
226 Transfer finished successfully. 0.00 KB/s
local: roma.gif remote: roma.gif
200 Port command okay.
150 File roma.gif okay.
Opening data connection.
226 Transfer finished successfully. 0.00 KB/s
```

Per uscire, il comando è *quit*.

```
ftp> quit
221 Service closing, don't be a stranger.
luther@frankenstein:~$
```

15.3.1 L'anonima trasferimenti colpisce ancora

Se non avete un account su un computer non potete accedere alla shell. Ma via ftp, solitamente qualche opportunità in più è data. Molti computer su internet offrono un servizio di *ftp anonimo*. In pratica, quando vi verrà richiesto di presentarvi, la vostra risposta dovrà essere *anonymous*. L'anonymous ftp richiede che voi immettiate come password il vostro indirizzo e-mail e per rispetto verso la Netiquette (parola composta da Net, rete, e Etiquette, buon comportamento) dovreste mettere quello vero!

```
luther@medialab:~$ ftp vostok
Connected to vostok.freaknet.org.
220 vostok.freaknet.org FTP server
(Version wu-2.4.2-VR16(1) Sun May 9 20:10:03 CDT 1999) ready.
Name (vostok:luther): anonymous
331 Guest login ok, send your complete e-mail address as
password.
Password:
230-Server FTP ufficiale della rete interna del FreakNet Medialab
230-Per eventuali problemi, scrivete una email a :
230-root@vostok.freaknet.org
230-Se avete problemi, provate ad usare il segno meno "-" come
30-primocarattere della vostra password -- questo disattiverà il
30-messaggio di continuazione che potrebbe confondere il vostro
30-client ftp.
230- 230 Guest login ok, access restrictions apply.
Remote system type is UNIX. Using binary mode to transfer files.

ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
Total 10
drwxrwxr-x 8 root wheel 1024 Oct 13 20:07 .
```

```
drwxrwxr-x 8 root wheel 1024 Oct 13 20:07 ..
drwxrwxr-x 2 root wheel 1024 Apr 7 1998 bin
drwxrwxr-x 2 root wheel 1024 Aug 30 1993 etc
drwxrwxrwx 2 root wheel 1024 Dec 1 15:15 incoming
drwxrwxr-x 2 root wheel 1024 Nov 17 1993 lib
drwxr-xr-x 11 root wheel 1024 Nov 30 19:58 pub
drwxrwxr-x 3 root wheel 1024 Aug 30 1993 usr
-rw-r--r-- 1 root wheel 317 Oct 13 20:07 welcome.msg
226 Transfer complete.
ftp>
```

Una volta dentro non potrete saltare da un ramo all'altro dell'albero delle directory. Potrete muovervi solo tra le directory disponibili per questo servizio. Di solito sono due quelle che sulle quali potete operare:

- ***pub***: da questa si dipartono i successivi rami che contengono i file che potete liberamente prelevare. A meno di una grande svista dell'amministratore, vi sarà impossibilitato scriverci dentro.
- ***incoming***: questa directory è riservata ai vostri upload. Cioè, qui potete spedire i file che volete far arrivare all'amministratore o a qualche utente che vi ha accesso.

Parte VI

Il mondo dalla finestra, le interfaccie grafiche

Capitolo 16

Aprite le finestre, le G.U.I.

16.1 Gli infissi e i topi

GUI è il solito acronimo da tre lettere tanto amato dai programmatori. Sta per *Graphical User Interface* (interfaccia grafica utente) e si riferisce all'ambiente di lavoro grafico.

Fino a ora avete lavorato su console rigorosamente testuali, incapaci di visualizzare alcunché di grafico e quello che avete visto sullo schermo era composto solo da lettere e numeri tutti della stessa grandezza; per comunicare con l'interprete di comandi l'unico mezzo è stata la tastiera. La GUI invece vi fa lavorare con programmi dall'aspetto più grazioso, che occupano lo schermo solo per la dimensione che volete e fa un forte uso di una graziosa periferica chiamata *mouse*.

Se lavorate con più programmi, vi permette di visualizzarli tutti contemporaneamente. Ogni programma gira nella sua *finestra*, cioè in una porzione dello schermo nel quale visualizza il suo funzionamento. Le finestre possono essere spostate, rimpicciolite, allargate, nascoste, persino mandate su un altro *schermo virtuale*. Avendo però un solo monitor e una sola tastiera, i vostri comandi possono riferirsi a una sola finestra per volta. Anche se tutte le finestre aperte continuano a far funzionare il programma che ospitano, quella sulla quale potete operare è la *finestra attiva*.

16.2 Il gestore della scrivania

I modi con cui vi viene segnalata dal computer la finestra attiva in un dato momento cambiano da un *desktop manager* all'altro. Quando avete di fronte un Macintosh o un altro elettrodomestico equipaggiato con Windows 9x, le finestre hanno tutte lo stesso aspetto (almeno sui bordi). Sotto Unix regna invece la confusione (e quando mai?).

Se è vero che alla base di tutto cè *X Windows* (si chiama così la base dell'interfaccia grafica per oltre il 90% degli Unix), il modo con cui tutto viene visualizzato dipende dalle vostre scelte o da quelle fatte dall'amministratore di sistema per vostro conto (che di solito potete comunque modificare). Il vostro ambiente grafico può avere un aspetto spartano o barocco a seconda del programma che avete scelto come gestore della scrivania.

Di solito quelli spartani sono molto veloci, non hanno un aspetto particolarmente attraente e chiedono poche risorse al computer;

Quelli barocchi sono belli da vedere, sono lenti e assorbono buona parte dell'impegno della memoria di lavoro e del microprocessore.

Alla prima categoria appartengono, tra gli altri, *fvwm2 e blackbox*; alla seconda *KDE, Window Maker e Gnome*.

Innumerevoli altre scrivanie appartengono alle categorie intermedie. Il pregio è che si può scegliere in questo modo il gestore più adatto nel rapporto comodità d'uso/potenza del computer. Anche macchine abbastanza vecchie possono avere un ambiente grafico abbastanza veloce rinunciando a un po' di appeal.

16.3 Ma il programma non è qui

Qui al FreakNet MediaLab riusciamo a usare l'ambiente grafico anche sugli 80486 grazie a un'importante caratteristica della GUI. X Windows (per gli amici semplicemente X) è un'applicazione che usa il protocollo TCP/IP e la tecnologia client-server (ve l'avevamo detto che per Unix è la lingua madre). In pratica succede che mentre la parte base (X Windows vero e proprio) gira sul computer dal quale avete lanciato la GUI (il computer davanti alla quale state fisicamente: per esempio votsok, mir, mariner etc.), i programmi che aprite viaggiano attraverso la rete e vengono eseguiti da medialab. Cioè, pioneer o vostok o l'altro computer sul quale vi trovate si occupa di visualizzare le finestre (ed è il server), mentre su medialab gira effettivamente Netscape o le altre cose che avete aperto. Al solito, questa cosa è possibile farla anche su lunghe e lunghissime distanze e anche via internet, se le linee della rete lo consentono.

16.4 Anatomia di un topo (mouse)

I mouse che si usano con X Windows devono avere tre tasti. Se ne avete due la pressione contemporanea di entrambi simulerà il tasto centrale, ma il consiglio che vi diamo è di farvelo cambiare o di andarne a comprare un altro voi stessi (con pochi euro potete evitarvi molte frustrazioni).

Usare il mouse non è difficile (esiste qualcosa di umano anche in Unix) ma bisogna familiarizzarvi. Per far scorrere il puntatore sullo schermo (cioè la freccetta nera) bisogna lasciare i mouse ben adagiato su una superficie liscia (meglio ancora se sul suo tappetino al quale è tanto affezionato) e farlo scorrere con il filo (la coda!) rivolta dal lato opposto al vostro. È bene dare subito un nome ad alcune basilari operazioni che si eseguono col mouse:

- ***Cliccare o fare click***: esercitare una breve pressione sul tasto sinistro (LMB, Left Mouse Button);
- ***Doppio click***: premere rapidamente per due volte il tasto sinistro;
- ***Cliccare col destro***: esercitare una breve pressione sul tasto destro (RMB, Right Mouse Button);
- ***Cliccare col centrale***: esercitare una breve pressione sul tasto centrale (CMB, Central Mouse Button) o, nel caso di mouse minorato, esercitare una breve pressione contemporaneamente sui tasti destro e sinistro;
- ***Trascinare***: puntare loggetto da spostare e iniziare a tenere premuto il tasto sinistro. Non rilasciare l'LMB e contemporaneamente far scorrere il mouse fino a che il puntatore arrivi alla posizione di destinazione per l'oggetto. Poi rilasciare il tasto;
- ***Trascinare col destro***: le stesse operazioni del punto precedente, ma impegnando il tasto destro.

16.4.1 Il topo salterino

Se il vostro mouse comincia a far muovere il puntatore a scatti, probabilmente ha bisogno solo di una buona pulizia. Non si tratta delle orecchie o delle unghie, ma dovrete tirargli fuori la palla.

Rovesciando il mouse, vi accorgerete che c'è il modo di smontare il coperchietto di plastica che impedisce alla palla di schizzar via. Una volta estratta, mettete la palla da parte (il vostro mouse ne avrà ancora bisogno!) e concentratevi sugli ingranaggi che vedete all'interno.

Con una pinzetta, un bastoncino cotonato, con un fiammifero, con quello che vi trovate, cominciate ad asportare quello strato di polvere e sudore pietrificato che forma una guarnizione completa sulle rotelline. Abbiate cura di gettare il materiale estratto lontano dal tappetino o dalla superficie sulla quale lavorate, altrimenti presto sarete nella stessa situazione. Quando avete finito reintroducete la palla e incastratela dentro col suo coperchietto.

Se siete ricchi e potete permettervi un mouse ottico allora difficilmente vi ritroverete nella situazione di pulire il vostor mouse.

16.5 Le finestre del FreakNet MediaLab

Un'avvertenza pratica che riguarda il laboratorio. Per lanciare X Windows sui computer che lo supportano, al login dovrete semplicemente digitare x e premere invio. Il server di X partirà e dopo pochi secondi di verrà richiesto l'abituale login su medialab, anche se stavolta con l'aspetto grafico (non fatevi intimorire e fate caso che in alto c'è scritto proprio medialab). A questo punto comportatevi come avete sempre fatto e tutto andrà per il meglio.

Quando X Windows e le sue applicazioni non sono distribuite su più computer (e questo succede nella maggioranza dei casi), la procedura normale per avviare l'ambiente grafico è digitare *startx* al prompt dell'interprete dei comandi.

Ricapitolando: affrontate il login come sempre, presentandovi col vostro account e poi immettete:

```
$ startx
```

16.6 Le finestre rotte

Quando non riuscite a chiudere X secondo la procedura normale del vostro gestore della scrivania, premete contemporaneamente i tasti *alt, control e backspace.*

Capitolo 17

La rete si naviga, non si e$plora!

17.1 Fra le U.R.L. delle sirene

Prima di entrare nel vivo dell'argomento, permetteteci una breve premessa.

17.1.1 Breve premessa

L'enorme espansione della rete internet è cominciata a metà degli anni novanta, quando la potenza dei computer casalinghi aveva già raggiunto livelli ragguardevoli. Potenza necessaria per usare il servizio che ha determinato lo stesso successo di internet: il *World Wide Web* (per gli amici *WWW* e per gli amiconi di lunga data *3W*).

La ragnatela ipertestuale ha reso internet uno strumento alla (presunta) portata di tutti, avendo tolto di mezzo la necessità di imparare a usare rigidi client testuali e avendo introdotto l'idea che con un clic è possibile raggiungere tutto il mondo. Questo ha creato la convinzione errata, ben difficile da estirpare, che identifica internet con il www. Oggi internet gode di linee molto più capaci di quelle che erano in funzione solo cinque anni fa. Ma queste linee sono sempre intasate per via delle ingenti quantità di spazzatura ipermediale (neologismo che non amiamo troppo, formato dalle parole ipertestuale e multimediale) che vi scorre. Prima che gli ipertesti divenissero il servizio più popolare della rete, le linee meno capienti riuscivano a soddisfare più rapidamente le richieste degli utilizzatori. Scusate l'attacco di nostalgia. Ma la frittata ormai è fatta e non vorremmo certo lasciarvi più indietro di un utente medio di un elettrodomestico qualsiasi. Anzi!..

17.1.2 Nomi comuni di cose mai capite

Ipertesto: È vero che se volete potete aprire un libro da pagina 200, leggerne un paio di pagine e poi saltare indietro a pagina 125 per tornare subito dopo a pagina 218. Ma ciò non è il modo normale con cui il libro andrebbe consultato. Il libro infatti è pensato con una sequenza di pagine che parte dalla 1 e che va letta in progressione. Per seguire il ragionamento dell'autore bisogna leggere il libro con lo stesso percorso che ha pensato. Quindi cominciare da pagina 1 per passare alla 2 e poi alla 3 e così via fino alla fine. Un ipertesto ribalta questo modo sequenziale di leggere. Non esiste un percorso predefinito e il lettore può seguirne uno proprio, seguendo il link posto sulla parola chiave dell'argomento che più gli interessa nel momento stesso in cui legge il documento. A differenza dei libri, è facile perdersi.

Link: Il link letteralmente è il collegamento. Avete già esperienza dei link simbolici che avete creato nel file system di Unix. I link dei documenti ipertestuali funzionano secondo un concetto molto simile, ma possono puntare anche a file che si trovano su altri computer.

Browser: Il browser (impossibile tentarne una traduzione decente, forse "sfogliatore") è il programma client che si utilizza per visualizzare i documenti ipertestuali sulle reti internet-like (e anche su internet, ovviamente!). Ha spesso capacità grafiche e può gestire servizi diversi dall'http, come ftp, anonymous ftp, gopher (un castoro padre naturale dell'http), eccetera. Il browser privo di capacità grafiche più famoso si chiama *lynx* (esiste anche per Dos e si fa chiamare doslynx); quello grafico è il *netscape* nelle sue varie versioni (Navigator, Communicator, ecc.). Alcuni file manager si comportano da browser, seppure con caratteristiche inferiori: per esempio il kfm per KDE, l'explorer per Windows 95/98/NT, oppure nautilus di Gnome.

Url: La Url è una convenzione per indicare un unico e preciso file su tutta internet. L'acronimo significa effettivamente *Universal Resource Locator* (localizzatore delle risorse universali può essere una traduzione approssimata e quasi decente), anche se talvolta la U è espansa con Unique o con Uniform. Ecco una tipica url:

http://www.freaknet.org/eventi/index.php

Mentre nella posta elettronica la parte più pesante è a destra, la Url si legge in maniera un po' ingarbugliata.

- ***http:*** è il prefisso del servizio al quale si intende accedere. Http sta per *Hyper Text Transfer Protocol* e si riferisce al fatto che vogliamo vedere un documento ipertestuale (detto gergalmente pagina web). Altri prefissi validi sono ftp, gopher, ways (ovviamente danno accesso ad altri tipi di servizi). Il separatore dopo il prefisso è sempre ***://***
- ***www.freaknet.org***: questa parte è del tutto identica alla parte a destra della @ degli indirizzi di posta elettronica. Quindi www sarà il nome del computer, freaknet il dominio di secondo livello e org il dominio di primo livello. È molto comune che le macchine che funzionano da server http vengano chiamate www, così come il nome della macchina per molti server ftp è proprio ftp. Il separatore che si usa tra nome macchina e domini è il solito . (punto);
- ***eventi***: Il nome della directory all'interno del quale cercare il documento che ci interessa. L'albero delle directory offerte dal server web è solo una porzione di tutto l'albero del computer (cioè potete vedere solo da un certo ramo in poi). Il file che state cercando potrebbe essere a un livello ancora inferiore, cioè in una sottodirectory: in un caso del genere si scriverà il percorso completo delle directory con il solito separatore */ (slash)*. Insomma: questa storia qui delle directory è la stessa che avete sempre saputo. Se la directory viene omessa, si assume che vogliate partire dalla radice */;*
- index.php: il nome del documento da visualizzare. È questo quello che apparirà sul vostro browser. Se il nome del file viene omesso, il server http vi invierà il file predefinito per quella directory.

Sito: Potremmo definire un sito web come l'insieme di documenti ipertestuali che stanno sullo stesso computer e hanno una matrice comune (argomento, autore, grafica).

Homepage: L'homepage in senso proprio è la pagina di partenza di un sito, quella dove solitamente si trova la maggior parte dei link alle pagine che compongono il sito stesso.

17.2 Mollate gli ormeggi

Quando lanciate il vostro browser, esso vi porterà automaticamente da qualche parte. La prima volta vi farà vedere la pagina che ha predefinito l'autore del programma (Netscape vi porterà a casa sua e altrettanto Lynx). Quando usate un browser testuale, la cosa più conveniente è lanciarlo seguito direttamente dalla Url da aprire, come nell'esempio:

```
$ lynx http://www.freaknet.org
```

Questi sono i comandi principali di lynx:

Comando	Descrizione
invio o freccia dx	segui il link
freccia sx	torna al documento precedente
freccia giù	giù di una schermata
freccia sù	sù di una schermata
g	apri la url...
C-r	ricarica il documento
Q	esci da lynx

I browser grafici sono ormai così intuitivi che non necessitano di spiegazioni.

17.3 Fare il punto della nave

Ci sono due modi per trovare quello che ci serve tra i milioni di computer presenti su internet.

17.3.1 Gli elenchi

Il primo modo è quello di raggiungere un elenco di siti, organizzato ipertestualmente nella solita struttura ad albero. L'elenco più noto è Yahoo (http://www.yahoo.com), ma anche google.it. Quello che appare della prima pagina di Yahoo è una struttura di questo tipo:

Arts & Humanities
Literature,
Photography...
News & Media
Full Coverage, Newspapers, TV...

Business & Economy
Companies, Finance, Jobs...
Recreation & Sports
Sports, Travel, Autos, Outdoors...
Computers & Internet
Internet, WWW, Software, Games...
Reference
Libraries, Dictionaries, Quotations...
Education
College and University, K-12...
Regional Countries, Regions, US States...
Entertainment
Cool Links, Movies, Humor, Music...
Science
Animals, Astronomy, Engineering...
Government
Elections, Military, Law, Taxes...
Social Science

Archaeology, Economics, Languages...
Health
Medicine, Diseases, Drugs, Fitness...
Society & Culture
People, Environment, Religion...

A questo punto potete scegliere un settore nel quale indirizzare la ricerca. Questo è quello che vi troverete di fronte scegliendo Society & Culture.

Advice (270)
Etiquette (17)
Magazines (218)
Bibliographies (12)
Events (35)
Museums and Exhibits (3924)
Chats and Forums (23)
Families (692)
Mythology and Folklore (484)
Crime (2515)
Firearms (133)
People (47877)
Cultural Policy (13)
Food and Drink (3123)
Relationships (377)
Cultures and Groups (13057)
Gender (33)
Religion and Spirituality (30111)
Death and Dying (432)
Holidays and Observances (1516)
Reunions (301)
Disabilities (1356)
Issues and Causes (3222)
Sexuality (1624)
Environment and Nature (3161)
Journals (5)
Social Organizations (390)

Ogni nuova voce porta a fianco il numero di siti conosciuti dagli autori di Yahoo come appartenenti a quella specifica categoria. Ogni categoria può avere una o più sottocategorie, che a loro volta possono averne delle altre. Avete presente un albero di directory? Ecco!

17.3.2 Search engine

Altre volte può essere utile cercare per parole chiave. Per questo tornano utili i motori di ricerca, che altro non sono che dei computer che hanno raccolto il contenuto dei documenti web di buona parte della rete e sono in grado di dirvi in quale file è presente la parola che cercate. Ci sono molti motori di ricerca con eccellenti prestazioni, tra i quali Google `http://www.google.it`, Hot Bot `http://www.hotbot.com`, Lycos `http://www.lycos.com`,
Excite `http://www.excite.com`.

Finirete per affezionarvi a uno di questi in maniera particolare. Il nostro motore di ricerca è Goggle, nella sua versione linux (`http://www.google.com/linux`). Questo motore (come molti altri del resto) vi permette di fare ricerche molto accurate grazie all'uso degli operatori dell'algebra booleana. Ogni motore vi chiederà probabilmente una sua precisa sintassi per l'immissione della ricerca, ma avendo ben chiaro il funzionamento degli operatori logici (altro nome

per chiamare gli operatori dell'algebra di Boole) vi adatterete a qualunque situazione. Basterà leggere la documentazione del motore stesso.

Oltre all'uso degli operatori booleani, Google usa una serie di parole riservate che vi permettono di specificare su quale host cercare o su quale dominio o che tipo esattamente di file (oltre a una quantità di altre cose). L'aiuto si trova in bella vista sulla home page del motore di ricerca.

17.4 Tutto inizia in Europa e sotto Unix

L'iniziativa per la creazione del www parte nel 1989, quando *Timothy Berners-Lee*, allora collaboratore del *CERN* (Centro Europeo di Ricerca Nucleare), propone un sistema di ipertesto in rete (all'inizio puramente testuale) per trasferire documenti e migliorare la comunicazione scientifica tra i fisici delle alte energie. Il *www*, in quanto sistema in grado di trasmettere, visualizzare e manipolare documenti ipertestuali attraverso l'Internet, esiste dal 1990, quando vengono definite le prime versioni di HTML e HTTP; tra la fine del 1990 e l'estate del 1991 *Berners-Lee*, in collaborazione con *Robert Cailliau*, sviluppa e rende disponibili, prima al CERN poi sull'Internet, il primo server www e il primo client, un browser e editor grafico per il sistema operativo Next-step [1]. Nello stesso periodo viene sviluppato da Nicola Pellow per uso pubblico un browser in modalità solo testuale per Unix, in seguito riscritto per il Macintosh. Segue un periodo di sviluppo latente tra il 1991 e il 1992. La prima efficace interfaccia grafica per il www viene sviluppata negli Stati Uniti, al National Center for Supercomputing Applications (NCSA), con il nome di «*Mosaic*»: si tratta del primo browser in grado di visualizzare le varie componenti del testo con diverse dimensioni e appropriati attributi tipografici, nonché le immagini interne al documento; supporta efficacemente dei programmi ausiliari che permettono di visualizzare o di utilizzare altrimenti documenti di ogni formato cui si punti da un documento HTML, trasferiti mediante uno dei diversi protocolli previsti; sviluppato inizialmente per Unix, poi per Macintosh e Windows, dalla metà del 1993 viene distribuito liberamente attraverso la rete e diventa rapidamente molto popolare tra gli utenti del www, scatenandone (è il caso di dirlo) lo sviluppo. Nel corso del 1993 il traffico sulla dorsale americana attribuito all'uso del www aumenta di quasi 200 volte. Nel giugno di quell'anno il primo programma di esplorazione automatica del www, in grado di seguire sistematicamente la rete dei documenti da link a link, trova un centinaio di siti, con più di 200.000 documenti. Un programma analogo, sulla base del quale è stato poi realizzato uno dei primi sistemi di ricerca libera sui contenuti della rete, ne individua 3.800 a maggio del 1994. A metà del 1996 i cataloghi del contenuto del www, che sono ormai uno dei servizi più frequentati della rete, punto di partenza per ogni ricerca d'informazioni, censiscono decine di milioni di pagine per più di 250.000 server.

(Enrico Pasini, tratto dall'Appendice al Grande Dizionario Enciclopedico,UTET-Torino, 1997)

[1] *NeXTstep non è propriamente un sistema operativo, ma lambiente grafico di una versione di Unix chiamata NeXT. Svolge insomma la stessa funzione di X Windows sotto Linux (nota del FN ML).*

Parte VII

Approfondimenti

Capitolo 18

Appendice A - Approfondimento sui permessi

18.1 Quelli con otto dita ed anche meno

La matematica che vi è stata insegnata fin dalla prima elementare è fatta con dieci cifre che sono:

0	1	2	3	4	5	6	7	8	9

Questo viene dal fatto che avete proprio dieci dita e che i matematici degli esordi, non avendo abachi, regoli e calcolatrici, si appoggiavano su di esse per i loro calcoli. Non saranno stati calcoli molto complicati, ma la scelta di un tale sistema di numerazione ce la portiamo ancora dietro.

Immaginate adesso che di dita ne avete solo otto. Come prima cosa i vostri nonni avrebbero cominciato a contare naturalmente fino a otto. La matematica sarebbe basata su queste cifre.

0	1	2	3	4	5	6	7

Il vostro computer quando regola i permessi si comporta proprio come un tipo a cui mancano due dita. Si dice che conta *in base otto* o *in ottale.* In realtà non conta né in ottale e né in *base dieci* (o *decimale*, in base dieci, proprio come avete sempre fatto). Sembra proprio che di dita ne abbia solo due, seppure non siano documentate nei manuali sull'hardware. Per questa ragione conta solo fino a due e la sua matematica è fatta solo da due cifre: 0 e 1. Quando vi chiede un numero in ottale lo fa solo per la vostra convenienza: il povero illuso pensa che vi venga più facile.

La numerazione a due cifre viene chiamata *binaria* (forse l'inventore era un ferroviere?).

Questo modo di fare matematica è del tutto simile a quello tradizionale. Le cifre vengono scritte da sinistra verso destra perdendo progressivamente «peso». Se scrivete 139 (senza sistemi strani, come vi ha insegnato la maestra) vi rendete subito conto che il 9 è meno importante dell'1. Allo stesso modo, scrivendo 10001011 (che in binario significa 139), l'1 più a sinistra conta molto di più dell'ultimo a destra. Ecco perchè:

Numero = 139_{10}

base	10^2	10^1	10^0
peso in decimale	100	10	1
valore in posizione	1	3	9

Operazione: 1 x 128 + 0 x 64 + 0 x 9 = 139

Numero 10001011_2 = 139_{10}

base	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
peso in decimale	128	64	32	16	8	4	2	1
valore in posizione	1	0	0	0	1	0	1	1

Operazione: 1 x 128 + 0 x 64 + 0 x 32 + 0 x 16 + 1 x 8 + 0 x 4 + 1 x 2 + 1 x 1 = 139

18.2 Quando 1 + 1 non fa 2

Anche le operazioni aritmetiche in binario hanno le stesse regole di quelle che avete eseguito finora.

Nell'addizione i numeri si scrivono in colonna e si sommano come sempre. Ovviamente non si risporta quando si supera il 10, ma quando si supera il 2.

```
  10 +
  11 =
-------
 101
```

Come sempre si comincia dalla colonna più a destra. 0 + 1 fa uno (fin qui niente da discutere). Poi ci si sposta di un posto a sinistra e si somma 1 + 1. Visto che non potete usare la cifra 2, azzerate il conto (come faremo normalmente arrivando a 10 con le dita) scrivendo 0 e riportando 1. Non avendo tra gli addendi cifre più a sinistra degli 1 iniziali, il riporto sarà direttamente la prima cifra della somma.

18.3 Scrivere conciso

Avete già visto quanto siano ingombranti i numeri binari. Se per scrivere 139 usate tre cifre decimali, per scrivere la stessa cosa in binario (1001011_2) ve ne servono otto! In più la possibilità di sbagliare a scrivere un numero lungo e noioso (ha solo due simboli) è molto elevata.

Gli stessi scellerati che hanno insegnato al computer a contare con due dita se ne sono resi conto ben presto (l'hanno voluta la bici?). Così hanno trovato un paio di fantasiosi modi per sbagliare un po' di meno. Gli hanno dato dei nomi terrificanti e li hanno pure spacciati come cose amichevoli: *ottale* ed *esadecimale*.

Del primo avete già sentito parlare, ma adesso vediamo più nel dettaglio come funziona.

Prendete un numero binario qualsiasi e dividetelo in gruppi da tre cifre partendo da destra. Se vi restano cifre in più mettetegli degli 0 davanti fino a quando avrete completato il gruppetto da tre.

1001011_2 diventa:

010 (il primo 0 è stato aggiunto da voi)	001	011

Ora è possibile procedere con la conversione dei singoli blocchi:

base	2^2	2^1	2^0
peso in decimale	4	2	1
valore in posizione	0	1	0

Operazione: 0 x 4 + 1 x 2 + 0 x 1 = **2**

base	2^2	2^1	2^0
peso in decimale	4	2	1
valore in posizione	0	0	1

Operazione: 0 x 4 + 0 x2 + 1 x 1 = **1**

base	2^2	2^1	2^0
peso in decimale	4	2	1
valore i posizone	0	1	1

Operazione: 0 x 4 + 1 x 2 + 1 x 2 = **3**

Ricapitolando il nostro 1001011_2 in binario ora vale 213_8in ottale. Volete una prova?

Qui sotto c'è la conversione del nostro 213_8 che, indovinate un po', fa sempre 139_{10} in decimale. Di fatto non c'è niente di nuovo ed i metodi sono uguali a quelli usati prima per convertire tra gli altri sistemi di numerazione.

numero da convertire in decimale: 213_8

base	8^2	8^1	8^0
peso in decimale	64	8	1
valore i posizone	2	1	3

Operazione: 2 x 64 + 1 x 8 + 3 x 1 = 139_{10}

18.4 Arti marziani

L'altro sistema di cui si è accennato è *l'esadecimale*. Quasi sicuramente è stato importato da qualche pianeta alieno, visto che il suo inventore doveva avere sedici dita. Ormai avete capito l'antifona e avete colto al volo che questo è un sistema di numerazione a sedici cifre. L'unica cosa che non sapete ancora è cosa sia stato affiancato alle nostre solite dieci cifre.

esadecimale	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
decimale	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

La conversione dal binario all'esadecimale funziona esattamente come quella all'ottale, con l'unica differenza che i gruppetti di cifre binarie si fanno a quattro a quattro. Provate a fare da soli la solita conversione di 1001011_2(il risultato è $8B_{16}$).

Insomma, 1001011_2, 213_8, $8B_{16}$ è sempre lo stesso numero e fa 139_{10}.

18.5 E che c'entra il chmod?

Ora che avete fatto un bel ripassino di matematica vediamo un po' di applicarla e magari a proposito dell'argomento «permessi» trattato nel capitolo 9. L'ultima cosa che avete fatto al vostro file era questa:

```
$ chmod 600 rubrica
$ ls -l rubrica
-rw------- 1 luther users 209 Nov 20 17:05 rubrica
$
```

Come avete visto i permessi si leggono, a parte il primo carattere della riga, a gruppetti di tre, proprio come si raccolgono le cifre per la conversione in ottale. Basta considerare 0 quello che non desiderate e 1 quello che invece volete ed ecco nella tabella qui sotto svelato l'arcano.

La shell al posto degli «uni» e degli «zeri» vi farà vedere i loro significato (con r,w e x).

permesso	binario	ottale
- - -	000	0
- - x	001	1
- w -	010	2
- w x	011	3
r - -	100	4
r - x	101	5
r w -	110	6
r w x	111	7

18.6 Un argomento appiccicoso

Delle volte guardando i permessi di un file eseguibile, cioè un programma o uno script di shell (un file di testo che contiene una sequenza di istruzioni comprensibili dall'interprete dei comandi) nel posto di una *x* potreste trovare una *s*.

Questa s sta per *sticky (appiccicoso)*. Chi ha i permessi per far eseguire il programma, lo fa spacciandosi automaticamente per il proprietario del programma stesso.

Difficilmente (forse proprio mai) dovrete impostare queste condizioni a un vostro programma o script, ma vi diciamo subito come fare per quel giorno in cui accadrà:

```
$ ls -l script
-rwxr-xr-x 1 luther users 209 Nov 24 15:29 script
$ chmod 4755 script
$ ls -l script
-rwsr-xr-x 1 luther users 209 Nov 24 15:29 script
$
```

Il 4 che anticipa il consueto gruppetto di tre cifre ottali attiva lo sticky bit. Il 755 è stato solo mantenuto, perché script aveva già quei permessi. Per rimuovere lo sticky bit mettete 0 nel posto in cui prima avete messo il 4 (tipo 0755).

Capitolo 19

Appendice B - Il vostro editor preferito

Smanettate un po' con gli editor di cui abbiamo parlato. Spontaneamente comincerete a preferire uno piuttosto che un altro.

Facciamo il caso che abbiate scelto joe e seguite queste istruzioni.

Verificate dove si trova il programma attraverso un comando del tipo:

```
$ whereis joe
```

oppure

```
$ locate joe
```

Se l'interprete non capisce whereis o locate, localizzate il programma con un *find*. In ogni caso appuntatevi il percorso completo, che sarà una cosa del tipo: */usr/bin/joe.*

Ora cercate di capire quale sia la vostra shell.

Il metodo più rapido è quello di vederlo direttamente dall'elenco degli utenti, che è un file che si chiama *passwd* (omonimo del programma che vi permette di cambiare la password) e che sta sotto la directory */etc*. Lanciate un grep su di esso cercando il vostro username come nell'esempio.

```
$ grep luther /etc/passwd
```

Annotate l'ultima parte della risposta. Se dopo il percorso (solitamente */bin* o */usr/bin*) c'è qualcosa come *sh, ash, o bash* avete una *Bourne Shell*. Se invece vi dice *csh o tcsh* allora è una *shell C*.

Tornate nella vostra home directory e lanciate il vostro editor preferito invocando *.profile* per la Bourne shell o *.login* per la shell C. Facciamo il caso che abbiate una bash.

```
$ joe .profile
```

Se il file esiste non toccate niente di quello che c'è già dentro (altrimenti non è detto che la prossima volta che aprirete un colloquio il computer sia felice di rivedervi) e limitatevi ad aggiungere come ultima riga:

```
export EDITOR=/usr/bin/joe
```

Se invece la vostra è una shell C, la riga da inserire sarà:

```
set editor=(/usr/bin/joe)
```

Salvate il file e chiudete il colloquio.

Dalla prossima volta, (quasi) qualsiasi programma debba appoggiarsi a un editor sceglierà automagicamente il vostro.

19.0.1 La modifica dell'ambiente

Quello che avete appena fatto riguarda la modifica di una variabile d'ambiente.

La variabile d'ambiente è un qualcosa che il sistema ricorda e che passa a un programma quando gli viene chiesta. Ci sono tante variabili d'ambiente e ognuna si occupa delle cose più disparate: una, per esempio, riguarda la visualizzazione sul vostro schermo; un'altra vi permette di lanciare programmi che stanno in directory molto lontane dalla vostra home.

Provate a digitare quanto segue se avete la Bourne shell (da ora in poi daremo per scontato che userete sempre questa: del resto è difficile che ve ne diano una diversa):

```
$ echo $PATH
```

Questo vi darà proprio l'elenco delle directory in cui sono contenuti i programmi che potete eseguire.

Avrete già capito che per cambiare il contenuto di questa variabile dovreste scrivere qualcosa come

```
$ export PATH=/altro/percorso/a/bin.
```

Non fatelo, o non riuscirete più a lavorare. Alcune variabili possono essere introdotte da voi (meglio se inserite nel .profile), mentre altre sono state preimpostate dall'amministratore di sistema (come, appunto, la $PATH). State attenti, perché potete modificare anche quelle preimpostate, ma non vi garantiamo un gran risultato. Anzi, potreste cominciare ad accusare non pochi problemi durante i colloqui con Unix.

Elenco delle figure

Parte VIII

GNU Free Documentation License

GNU Free Documentation License Version 1.1, March 2000

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin

distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page: