



# **CdC & Linux**

## **Versione 2004**

Introduzione all'uso delle risorse del  
Centro di Calcolo

Università degli Studi di Pisa  
Dipartimento di Informatica  
Gruppo Beatrice

Questa dispensa è stata realizzata dai membri del Gruppo Beatrice, un gruppo di studenti di informatica dell'Università degli Studi di Pisa.

Scopo di questa dispensa è fornire agli studenti del primo anno dei corsi di Laurea e Diploma in Informatica le conoscenze minime necessarie per usare le macchine Linux del Centro di Calcolo

Varie persone, nel corso di diversi anni, hanno contribuito alla realizzazione, espansione e aggiornamento di queste pagine. Eccoli elencati in ordine alfabetico:

- Giuseppe Augiero (augiero@cli.di.unipi.it)
- Gianni Giaccherini (giaccher@cli.di.unipi.it)
- Nicola Petruzzellis (petruzze@cli.di.unipi.it)
- Vittorio Romano (romano@cli.di.unipi.it)
- Leonardo Fiorini (fiorini@cli.di.unipi.it)

La versione più aggiornata della presente dispensa è disponibile all'indirizzo:

<http://beatrice.cli.di.unipi.it>

Per segnalare eventuali errori, per inviare critiche o commenti è possibile scrivere al seguente indirizzo:

[g\\_beatrice@cli.di.unipi.it](mailto:g_beatrice@cli.di.unipi.it)

## Indice

<i>Indice</i> .....	3
<i>Introduzione</i> .....	5
<i>Descrizione del Centro di Calcolo</i> .....	5
<i>Account</i> .....	6
<i>Regolamento del Cdc</i> .....	6
<i>I servers</i> .....	6
<i>I terminali</i> .....	7
<i>Stampanti</i> .....	7
<i>Sempre aggiornati!</i> .....	7
<i>I sistemi Unix</i> .....	8
<i>Breve storia di Unix</i> .....	8
<i>Logon - ovvero come collegarsi</i> .....	8
<i>Il prompt</i> .....	8
<i>Logoff - ovvero come scollegarsi</i> .....	9
<i>La shell e i comandi Unix</i> .....	9
<i>Le opzioni della shell</i> .....	9
<i>Cambiare la password</i> .....	9
<i>I file</i> .....	9
<i>Le directory e la gerarchia del filesystem</i> .....	10
<i>Gestione dei file</i> .....	10
<i>La home directory</i> .....	10
<i>Directory corrente</i> .....	11
<i>Navigazione nel file system</i> .....	11
<i>Creare una directory</i> .....	11
<i>Cancellare una directory</i> .....	11
<i>Spostare e rinominare le directory</i> .....	11
<i>Spostare e copiare file tra directory</i> .....	12
<i>Cancellazione di un file</i> .....	12
<i>Copia di un file</i> .....	12
<i>Rinomina di un file</i> .....	13
<i>Abbreviazioni dei percorsi</i> .....	13
<i>Diritti di accesso ai file e file nascosti</i> .....	13
<i>Esaminare i file</i> .....	14
<i>Modificare i file</i> .....	14
<i>Emacs</i> .....	14
<i>Pico</i> .....	15
<i>Spazio disco e quota</i> .....	15
<i>Compressione dei file</i> .....	16
<i>Stampa</i> .....	17
<i>Metacaratteri</i> .....	17
<i>Uso dei floppy con gli Mtools</i> .....	18
<i>Redirezione</i> .....	18
<i>Pipeline</i> .....	19
<i>Ricerca nei file</i> .....	19
<i>Trovare un file</i> .....	19
<i>Gli alias</i> .....	19
<i>Manuali</i> .....	20
<i>I processi</i> .....	20
<i>Cos'è la posta elettronica</i> .....	21

Come funziona	21
Gli indirizzi Email	22
Uso di pine	22
Scrivere e spedire un messaggio	22
Gestire la posta con pine	23
Come inviare un file	25
La rubrica degli indirizzi	25
X Window	25
Un po' di storia di Xwindows	25
X e il modello Client-Server	26
La struttura	26
Displays & Screens	27
Windows managers	28
Avviare X Window	28
File di configurazione	28
Shell e programmi sotto X	28
Browser web	28
Servizio Masterizzazione	30
Limitazione Internet	30
Help	31
Appendice A	32
Unix: il Sistema Operativo	32
Il fratello minore, ma solo cronologicamente ("a better Minix, than Minix")	32
Software Libero	33
La struttura di Linux: il KERNEL	33
La Shell	34
Appendice B	35
LOGON	35
CAMBIO PASSWORD	35
SINCRONIZZAZIONE LINUX/WINDOWS	35
LE STAMPANTI	35
GESTIONE PROFILO	36
Appendice C	37
Acceptable Use Policy della rete GARR	37

## Introduzione

Lo scopo di queste pagine è quello di introdurti ai sistemi del Centro di Calcolo del Dipartimento di Informatica dell'Università di Pisa e darti le informazioni utili per poter iniziare a lavorare. Non importa quanto sia la tua esperienza e abilità con i computer, ti renderai presto conto che c'è sempre qualcosa di nuovo da imparare.

In caso di richiesta di maggiori informazioni o segnalazione di errori, contattaci via email all'indirizzo:

`g_beatrice@cli.di.unipi.it`

Ti consigliamo di leggere queste pagine prima di iniziare a lavorare su un terminale. Come ti sentirai in grado di muovere i primi passi, cerca un computer e inizia a sperimentare... Non avere paura di commettere errori, al massimo eventuali sbagli si ripercuoteranno perlopiù sui tuoi dati. Un po' di cautela e di buon senso, comunque, non guastano mai.

Speriamo che tutto ciò risulti per te una esperienza interessante e gratificante. Buon lavoro!

## Descrizione del Centro di Calcolo

Il Centro di Calcolo del Dipartimento di Informatica dell'Università di Pisa ubicato nei locali del Polo Fibonacci (ex Marzotto), sito in via Buonarroti 2, si sviluppa in sei laboratori (Lab1... Lab 5 e Lab portatili) e da tre aule didattiche (Aula H, Aula I, Aula M), una sala macchine e due locali ad uso del personale tecnico (sistemisti e operatori).

I locali del Centro di Calcolo sono aperti all'utenza ogni giorno, dal lunedì al giovedì dalle ore 8,30 alle ore 23,00 ed il venerdì dall'8,30 alle 19,00.

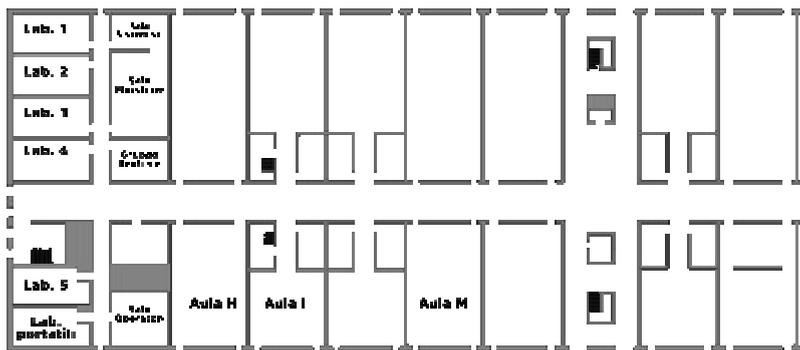
L'accesso è consentito a tutti gli iscritti al corso di Laurea o di Diploma in Informatica, indistintamente sia nei laboratori che nelle aule, salvo i casi in cui questi siano occupati per lo svolgimento di lezioni, esercitazioni, esami, o chiusi per manutenzione.

E' importante ricordare che durante la permanenza nei laboratori possono essere effettuati controlli sulla identità delle persone. E' necessario avere sempre con se il proprio libretto universitario, o un certificato di iscrizione.

I computer a disposizione degli studenti si dividono in due categorie: i PC IBM compatibili con Linux e/o Windows, fisicamente accessibili agli studenti, e i Server (Linux/Solaris/Win2000) che risiedono in sala macchine (non accessibile per gli studenti). Questi ultimi sono macchine dedicate alle applicazioni specifiche della didattica e garantiscono l'accesso a studenti e docenti alle risorse del Centro.

Tutte le macchine sono collegate tra loro (e con Internet) attraverso una rete dati, in modo da consentire la condivisione del maggior numero di risorse presenti all'interno del centro di calcolo come, ad esempio, le stampanti e di dischi in cui risiedono le home degli studenti.

Nella figura 1 si nota una pianta del Centro di Calcolo con le aule H, I, M, i Laboratori 1, 2, 3, 4, 5 e portatili (accessibili agli studenti), la sala macchine (non accessibile) e le stanze dei sistemisti e degli operatori.



## Account

Lo studente iscritto al corso di laurea in Informatica ha diritto ad un account al Cdc. Attraverso l'account e' possibile accedere alle macchine del Centro di Calcolo.

Per ottenere l'account è necessario compilare il modulo MAS, imbucarlo nell'apposita cassetta e, dopo una settimana, presentarsi, muniti di libretto universitario o certificato di iscrizione, dagli operatori per ritirarlo. Al momento della consegna dell'account, viene fatta firmare una dichiarazione di responsabilità che è bene leggere.

Si riceverà, così, il *login-name* e la *password* che permetteranno l'accesso ai computer.

Il *login-name* identifica univocamente lo studente. La *password* deve essere custodita scrupolosamente (come il pin del bancomat) e non deve essere comunicata a nessuno, in modo da evitare che qualcuno possa accedere, a tua insaputa, alle macchine del Centro di Calcolo e commettere danni a tuo nome.

Il Mas va compilato una volta sola durante l'intero percorso accademico. Esso darà diritto alla seguente configurazione minima:

- 15 Mb. di quota disco, e accesso alle macchine di servizio;
- quota di 50 pagine di stampa, a semestre, per la stampante laser;
- numero illimitato di pagine di stampa per la stampante ad aghi;
- possibilità di usare tutti i servizi di rete (es.: e-mail, browser internet, ftp, etc.);
- possibilità di usare il software che e' installato sulla macchina a loro dedicata;

Inoltre, lo studente che deve realizzare un progetto per un corso o preparare la tesi, ha diritto a risorse di laboratorio aggiuntive. Per ottenerle, deve fornire il suo *login-name* al docente responsabile del corso di laboratorio. In questo caso non è necessaria alcuna firma, ed a operazione avvenuta lo studente riceverà una mail che confermerà l'avvenuta assegnazione delle risorse.

**LE SINGOLE RISORSE HANNO SCADENZE DIPENDENTI DALLE RISORSE STESSE (AD ESEMPIO LA RISORSA TESI)**

## Regolamento del Cdc

Non esiste un vero regolamento del Cdc, tuttavia sono attive alcune norme, che vietano di:

1. cercare di conseguire accesso non autorizzato alle risorse connesse alla Rete;
2. influenzare negativamente la regolare operatività della Rete o limitarne l'utilizzabilità e le prestazioni per gli altri utenti;
3. distrarre e/o distruggere risorse (persone, capacità, elaboratori) del Cdc e più in generale sulla rete Internet;
4. violare la riservatezza degli utenti;
5. provocare trasferimenti non autorizzati di informazioni (software, basi dati);
6. usare dispositivi esterni senza preventiva autorizzazione;
7. allontanarsi dai terminali (lasciandoli occupati) per troppo tempo;
8. stampare documenti non legati alla didattica;
9. stampare più di una copia dello stesso documento;

10. consentire a persone estranee l'accesso al Cdc;  
In generale è comunque vietato sottrarre risorse agli altri utenti per attività non strettamente legate alla didattica.

## I server

Il server per l'utenza è una macchina dedicata alle applicazioni specifiche della didattica e garantisce l'accesso a studenti e docenti alle risorse del Centro. Non è fisicamente accessibile ed è posto nella sala macchine. Per via del ruolo che gioca all'interno della rete, viene spesso chiamato col nome di *server*, in italiano "servente" o "chi serve". Un "server" attende che arrivino richieste dai programmi lanciati dagli utenti (tali programmi vengono definiti *client*, in italiano "clienti"). In particolare vedremo come funziona una sessione di lavoro in cui, attraverso un terminale, sarà possibile far eseguire sul server i comandi e i programmi desiderati.

Il server, come del resto tutte le macchine del Centro di Calcolo, ha un proprio hostname, necessario per individuare la singola macchina. Il server disponibile agli utenti è:

- **Olivia** (AMD Athlon XP 2600+ – RedHat Linux 9 – Kernel 2.4.20-24.9) [accesso con account minimo]

È possibile collegarsi attraverso il servizio *ssh* da un qualsiasi terminale (sotto i sistemi Windows si devono usare i programmi *putty* e *winscp*) oppure da un qualsiasi punto di Internet, nel caso in cui si abbia l'esigenza di accedere alle proprie risorse del cdc pur non essendo presenti fisicamente all'interno del Centro di Calcolo.

L'onere della gestione dei servizi generali è invece affidato ad un altro gruppo di host che non è direttamente accessibile dagli studenti. Di questo pool fanno parte il proxy, il mailserver, il webserver, l'home server, il print server e il Nis Server.

## I terminali

Nei laboratori in cui hanno accesso gli studenti sono ubicati diverse tipologie di computer: PC su cui è installato Debian Linux Sarge e PC su cui è installato Windows XP. Il parco macchine è completamente eterogeneo data la diversa composizione hardware delle macchine. A seconda delle caratteristiche del singolo computer è possibile eseguire un determinato numero di programmi (per fini didattici) disponibili all'interno del centro di calcolo.

## Stampanti

Le stampanti a disposizione degli utenti del Centro di Calcolo sono quattro:

- 1 stampante laser HP LaserJet 55Xi Mopier
- 1 stampante laser Xerox Docuprint N24
- 1 stampante laser HP
- 1 stampante ad aghi Olivetti

## Sempre aggiornati!

Per essere sempre aggiornati su cosa accade all'interno del Centro di Calcolo vi consigliamo caldamente di consultare periodicamente l'url <http://www.cli.di.unipi.it/news.php> dove troverete tutte le news e le informazioni inerenti alle attività svolte dal Cdc.

## I sistemi Unix

### Breve storia di Unix

Il sistema operativo Unix è nato alla fine degli anni sessanta presso i laboratori Bell. All'epoca rappresentò un notevole passo avanti, anche in virtù del fatto che fu il primo S.O. scritto quasi completamente in un linguaggio di alto livello, il C, il cui sviluppo fu a sua volta condizionato da Unix e dalle esigenze che i creatori di quest'ultimo avevano.

I laboratori Bell concessero una licenza di libero uso alle università le quali a centinaia adottarono il nuovo Sistema Operativo (S.O.). Agli inizi, addirittura, era disponibile anche il relativo sorgente, ma purtroppo a partire dalla versione 7 AT&T considerò il codice un segreto commerciale. Unix è definito come un S.O. *general purpose* (adatto a tutti gli scopi), *multiutente* (le risorse di un computer su cui esso gira possono essere sfruttate da più persone contemporaneamente) e *multitasking* (gli utenti possono far eseguire più programmi in parallelo, grazie a una tecnica chiamata *time-sharing* che ripartisce in modo ottimale il tempo macchina tra i vari processi).

Attualmente esistono numerosi dialetti derivati dalla versione originaria che funzionano su macchine diverse: dal personal computer sino ai grandi calcolatori (mainframe), ma la struttura è sostanzialmente la stessa, tutte hanno le stesse funzionalità di base e garantiscono una più che buona compatibilità a livello di programmi.

Un notevole e recente contributo alla sua diffusione è stato Linux, una implementazione di pubblico dominio realizzata inizialmente da uno studente finlandese, Linus Torvalds, per PC compatibili con processore Intel 386. Rapidamente si è diffuso grazie al fatto che è liberamente distribuibile, così come lo sono i suoi stessi sorgenti, al cui miglioramento può contribuire chiunque. Attualmente vengono supportate tutte le piattaforme e tutte le famiglie di processori più diffuse (tra le altre, oltre a Intel x86 e compatibili, anche Alpha, 680x0, PowerPC, Sparc e MIPS).

### Logon - ovvero come collegarsi

In genere tutte le macchine dei laboratori restano accese 24 ore al giorno. Per risparmiare elettricità e preservare i fosfori dei monitor, dopo alcuni minuti di inattività da parte dell'utente, gran parte dei computer mostrano uno schermo nero o addirittura spengono il monitor. Per "risvegliare" il pc basta premere un qualsiasi tasto o muovere il mouse.

Un terminale pronto per effettuare un collegamento in genere mostra a schermo la scritta

```
login:
```

Nel caso essa non compaia, molto verosimilmente il terminale è già in uso da parte di un utente allontanatosi momentaneamente dal posto di lavoro. Non resta altro che cercare un altro terminale libero.

Dopo aver inserito il proprio login-name la macchina vi chiederà di inserire la vostra password

```
Password:
```

La password viene consegnata al momento del ritiro dell'account ed è da considerarsi strettamente personale. Nel caso si abbia il timore che qualcuno abbia letto in qualche modo la propria password, è possibile in qualsiasi momento cambiarla.

Un uso improprio delle risorse è **penalmente perseguibile**: occorre quindi fare attenzione a quello che si fa e assicurarsi di essere gli unici a usare il proprio account.

La procedura di login è importante non solo ai fini della sicurezza e cioè per impedire accessi non autorizzati alle macchine, ma permette, anche, di attribuire a un utente determinate risorse del centro di calcolo.

La password viene immessa "al buio" per motivi di sicurezza, cioè non sono visibili i caratteri che vengono digitati sulla tastiera.

Occorre, quindi, prestare attenzione a ciò che si digita. Vi ricordiamo che i sistemi Unix sono "case-sensitive", vale a dire che fanno differenza tra i caratteri maiuscoli e minuscoli. Se la vostra password è PippòBaudo, dovrete digitarla tutta in minuscolo con solo la prima P e la B in maiuscolo. Pippobaudo o pippobaudo, ad esempio, non saranno password valide. Analogo discorso per la login, che è sempre composta da caratteri minuscoli.

## Il prompt

Se login e password digitate sono entrambe corrette, la macchina ci permetterà di accedere alle sue risorse e sarà pronta a ricevere i nostri comandi attraverso il *prompt di comando*.

```
olivia>
```

A questo punto è possibile iniziare a lavorare impartendo i comandi alla shell.

## Logoff - ovvero come scollegarsi

Al termine della sessione di lavoro, dovete eseguire la procedura di chiusura. Una volta che uscite, Linux libera il terminale e lo rende disponibile ad altri utenti, o a un vostro riutilizzo. Scollegandovi dal sistema si evita che persone non abilitate possano utilizzare il vostro terminale.

Linux definisce due procedimenti differenti per scollegarsi. Il primo utilizza il comando **exit**

```
olivia> exit
```

```
Login:
```

Il sistema interrompe l'esecuzione della shell, si resetta e visualizza il prompt iniziale login.

Il secondo procedimento permette di uscire dal sistema premendo CTRL-D, che rappresenta il carattere di fine file e che la shell interpreta come un segnale di disallocazione.

E' fondamentale non scollegarsi dalla shell attraverso un reset o lo spegnimento della macchina. Tale operazione potrebbe compromettere il buon funzionamento dell'intero sistema.

## La shell e i comandi Unix

Tutti i sistemi Linux/Unix sono dotati di interfaccia grafica (X-Window) che su alcuni sistemi è avviata automaticamente, consentendo di fare il login in ambiente grafico, mentre in altri deve essere avviata dall'utente.

In ogni caso, una volta inseriti login e password, il sistema è pronto per ricevere i comandi dall'utente.

La shell è il programma che sta in attesa dei comandi introdotti dal terminale, li riceve e li traduce in istruzioni nella sintassi interna del sistema. Il termine shell indica effettivamente la funzione che svolge, cioè una funzione di interfaccia che si colloca tra la parte più interna del sistema operativo e il mondo esterno.

I programmi sono generalmente programmi che lo shell ricerca ed esegue in risposta a istruzioni che vengono dettate da tastiera. La shell vi aiuta a utilizzare al meglio la macchina, essendo un interprete dei comandi che può espandere o modificare il comando prima di eseguirlo. In quest'ottica, particolare importanza assumono i meta caratteri e gli operatori di connessione dei comandi, che possono rendere una linea di comandi più generalizzata e flessibile (parleremo nelle prossime pagine dei wildcard).

## Le opzioni della shell

Una opzione (o *switch* o *flag*) indica a un comando di comportarsi in un certo modo. Quasi sempre le opzioni consistono in una lettera preceduta da un trattino. Purtroppo il modo in cui più opzioni vanno specificate non è lo stesso per tutti i comandi. Alcuni si aspettano che esse vengano raggruppate e precedute da un solo trattino (es. -auxv), mentre altri le

vogliono ognuna con il proprio trattino (es. -a -u -x -v). Grazie al comando `man` (descritto nella sezione Manuali) è possibile conoscere quali opzioni sono supportate da un comando e il loro scopo.

## Cambiare la password

Questa operazione, che andrebbe effettuata periodicamente, può essere svolta solo da una shell di un sistema Linux/Unix. Nello scegliere la nuova password, si consiglia di scartare qualsiasi parola facilmente associabile con se stessi, ossia nome proprio, soprannome, numero di telefono o nomi di parenti/amici, anche se digitati al contrario. Sono da evitarsi infine parole comunque presenti in un dizionario. Una buona password la si ottiene combinando numeri, lettere (sia maiuscole che minuscole) e segni di punteggiatura (per esempio H5.dw2i! ).

La password di un account del cdc si modifica connettendosi su sul server alice. Dopo aver effettuato il login in una macchina del cdc, tramite la procedura mostrata sopra, il comando da usare è `ssh alice`. Durante l'iterazione il comando chiederà, L'inserimento della vecchia password, quindi, per due volte, l'inserimento della nuova password. Di norma la nuova password diventa valida da mezzanotte del giorno successivo.

## I file

Un *file* (archivio) è un contenitore di informazioni che corrisponde a una sequenza di byte. Un file può essere quindi un programma eseguibile, una parte di un testo, un database, un insieme di bit che memorizza una immagine su video. I file degli utenti sono memorizzati sui dischi rigidi (hard disk) del server RAID. Ad ogni utente è riservata una parte del disco (quota) da usare come spazio di lavoro. Tale spazio prende il nome di *home directory* (o più semplicemente home).

Qualunque combinazione di lettere e numeri, più i caratteri `.` (punto) e `_` (sottolineatura o underscore), può essere usata come nome di file. In teoria è possibile creare nomi di file contenenti spazi vuoti, ma ciò introduce poi spesso notevoli complicazioni, quindi è buona regola farne a meno.

Molto spesso si incontrano file con nomi del tipo `text.c`, contenenti cioè un punto. La parte finale del nome, dopo l'ultimo punto, è chiamata in genere *suffixo o estensione*. L'estensione viene usata dagli utenti per raggruppare facilmente file contenenti dati omogenei (cioè dello stesso tipo). Ad esempio ci si può aspettare che `text.c` e `gui.c` contengano entrambi listati di programmi scritti in C, mentre `leggimi.txt` conterrà molto probabilmente del normale testo.

## Le directory e la gerarchia del filesystem

Tutti i file di un sistema Unix fanno parte di un filesystem. Questo è organizzato attraverso una gerarchia di directory. Una directory è un raccoglitore di file e può fare parte a sua volta di altre directory, per cui si viene a creare una struttura complessa e ramificata a forma di albero. Alla sommità dell'albero è presente la cosiddetta *directory radice (root)*, rappresentata dal simbolo `/` (slash, da non confondere con il backslash, di uso comune nel mondo MSDOS). In pratica all'interno di una directory possono essere collocati i file e altre directory (sottodirectory), le quali a loro volta contengono file o altre sottodirectory, e così via. Rappresentando su carta i contenuti di una directory e le relative directory "figlie", si ottiene qualcosa di simile al disegno di un albero capovolto. Da ciò quindi il termine radice.

```
/ --- File1
|
| - File2
|
| - File3
|
| - Directory1 --- FileA
|                   |
|                   |- FileB
|
| - Home --- File4
|           |
|           |- Directory3 --- File5
```

Partendo dalla directory radice, spostandosi di sottodirectory in sottodirectory, è possibile alla fine raggiungere qualsiasi file

facente parte del filesystem. Può capitare che esistano due file con nome identico, ma posti in parti diverse dell'albero. Per evitare confusione, Unix identifica il file mediante i cosiddetti *path* (percorsi), formati dai nomi delle varie directory che "conducono" al file, separate dal simbolo /. Ad esempio, il path `/pippo/pluto/dati` identifica il file `dati`, contenuto nella directory `pluto`, contenuta a sua volta in `pippo`, che, infine, è una directory figlia di /, ossia la directory radice. Nell'accedere ai dati contenuti in tale file, quindi, Unix risalirà prima alla sommità dell'albero, scenderà all'interno di `pippo`, poi all'interno di `pluto` e infine accederà a `dati`.

## Gestione dei file

Per mostrare la lista dei file contenuti in una directory è possibile usare il comando **ls**.

```
olivia> ls
a.out* text.c text.h
olivia>
```

Il file il cui nome termina con l'asterisco si chiama in realtà `a.out`. L'asterisco è usato da `ls` per indicare all'utente in modo immediato che si tratta di un file contenente un programma che può essere lanciato digitandone il nome. Anche la prima volta che ci si collega, si troveranno dei file già esistenti. Essi sono necessari all'avvio di ogni sessione e vengono creati automaticamente dal sistema al momento della creazione dell'account. Finché non si è perfettamente consci di quello a cui servono, è buona idea non modificarli.

## La home directory

Quando entrate nel sistema, iniziate a operare da un punto particolare del file system che corrisponde alla vostra home directory, in cui potete liberamente creare file e subdirectory senza che altri utenti possano accedervi. L'home directory è rappresentata dal simbolo `~`. All'interno del Centro di calcolo la propria home directory si colloca in un ben preciso punto del file system: `/home/PrimaLetteraDelLogin/LoginName`. Ad esempio `/home/p/pippobaudou`.

Non preoccupatevi di dove vi trovate nel file system in quanto potete ritornare nella vostra home directory attraverso il comando **cd**.

## Directory corrente

Quando vi trovate allocati in una directory, questa directory prende il nome di *directory corrente*. Il comando **pwd** ci informa su quale sia la directory corrente.

```
olivia> pwd
/usr/bin
olivia>
```

## Navigazione nel file system

Potete percorrere tutto l'albero delle directory attraverso il comando **cd** seguito dal pathname.

Al comando `cd` potete passare un path di due diversi tipi :

- *path assoluto*: rappresenta il percorso completo partendo dalla root directory (`/usr/local/bin/`)
- *path relativo*: rappresenta il path partendo dalla directory corrente (`bin`)

Il carattere `.` sostituisce il nome della directory corrente. Potete utilizzare il pathname `./note` per referenziare il file `note` presente nella directory corrente.

## Creare una directory

Il comando **mkdir** permette di creare una nuova directory.

```

olivia> ls
primofile
olivia> mkdir primadirectory
olivia> ls
primadirectory/ primofile
olivia> mkdir primadirectory/secondadirectory
olivia> ls
primadirectory/ primofile
olivia> ls primadirectory
secondadirectory/

```

In quest'ultimo esempio sono state create due directory, una figlia dell'altra.

## Cancellare una directory

Il comando **rm** rimuove una directory, purché vuota. Ciò per evitare la cancellazione accidentale di un gran numero di file. Specificando come argomento un file, una directory inesistente o non vuota, verrà restituito un messaggio di errore.

```

olivia> ls primadir
secondadirectory/
olivia> rm primadir/secondadirectory
olivia> ls primadir
olivia>

```

## Spostare e rinominare le directory

Le directory possono essere rinominate come un qualsiasi file, col comando **mv**. Possono anche essere spostate in un'altra directory, formando al suo interno una o più sottodirectory.

```

olivia> ls
primadirectory/ primofile
olivia> mv primadirectory primadir
olivia> ls
primadir/ primofile
olivia>

```

## Spostare e copiare file tra directory

Specificando come secondo argomento di **cp** o **mv** il nome di una directory, il file sorgente verrà rispettivamente copiato/mosso in essa.

```

olivia> ls
primadir/ primofile
olivia> cp primofile secondofile
olivia> ls
primadir/ primofile secondofile
olivia> mv secondofile primadir
olivia> ls
primadir/ primofile
olivia> ls primadir
secondofile
olivia>

```

## Cancellazione di un file

Il comando `rm` consente di eliminare file di cui non si ha più bisogno. Volendo rimuovere il file appena creato, basterà impartire `rm` seguito dal nome del (o dei) file.

A differenza di quanto avviene con altri sistemi operativi, in genere i comandi Unix sono “poco loquaci”, ossia stampano messaggi sullo schermo solo se necessario (per informazioni) o in caso di errore. La regola generale è che un comando è andato a buon fine, se non è apparso alcun messaggio di errore. In questo `rm` non fa eccezione. Se come parametri vengono passati più nomi di file ed ognuno di questi era valido, `rm` non stamperà comunque nulla.

Per ognuno dei file specificati, il comando chiederà se si è sicuri di volerli eliminare. Solo rispondendo `y` (yes, sì) il file verrà effettivamente cancellato. Ciò è possibile grazie a un *alias* (vedasi la sezione relativa) che ridefinisce il comando `rm`, aggiungendo automaticamente l'opzione `-i` (interactive, interattivo) che appunto indica che per ogni file va chiesta prima la conferma dell'operazione.

Nell'usare `rm` va posta molta attenzione. I file, una volta eliminati, non possono essere recuperati senza l'aiuto dei sistemisti. In alcuni casi il ripristino è addirittura impossibile. Benché la copia di sicurezza (*backup*) dei dati di ogni utente venga effettuata regolarmente, essa avviene durante la notte. Se ad esempio un file creato alle ore 11, cioè dopo il backup, viene cancellato nel pomeriggio, ovviamente non vi è alcuna copia su nastro e quindi non sarà possibile recuperarlo. In ogni caso, il ripristino dei file è una operazione lunga e tediosa. È quindi consigliabile farne richiesta solo in caso di perdita di file grandi e/o di estrema importanza: per piccoli file è molto più facile e veloce crearli da capo.

## Copia di un file

Capita spesso di dover creare una copia di un file. Può ad esempio servire una copia di sicurezza di un file oppure avere l'esigenza di copiare un file in un'altra directory del file system. Il comando da usare è `cp`. La sintassi da utilizzare è *cp file* *file* *nuovofile*:

```
olivia> ls
primofile
olivia> cp primofile secondofile
olivia> ls
primofile secondofile
olivia>
```

Nel caso in cui il file di *destinazione* esistesse già, questo verrà sovrascritto. Non è possibile usare come primo argomento una directory. Se intendete copiare il contenuto di una directory in uno nuovo, dovete indicare il nome di ogni file che volete copiare o esplicitamente o tramite un metacarattere (`cp dir1/* .`)

Se il comando `cp` termina l'esecuzione correttamente, la shell riassume il controllo del sistema, in caso contrario `cp` invia un messaggio di errore.

## Rinomina di un file

Il comando `mv` permette di cambiare il nome di un file. Come per `cp`, la sua sintassi prevede il nome del file di origine e il nome del file di destinazione.

```
olivia> ls
primofile
olivia> mv primofile secondofile
olivia> ls
secondofile
olivia>
```

Il comando `mv` viene usato anche per lo spostamento dei file

## Abbreviazioni dei percorsi

Per rendere le cose più semplici, è possibile usare il carattere ~ (tilde), per identificare la home directory di un utente. Usando ad esempio come nome di file `~pippo`, esso verrà sostituito con il percorso completo della home directory dell'utente pippo. Il simbolo ~ da solo, invece, identifica la propria home directory. Il file `~pippo/dati/ricette` è quindi in realtà il file `ricette` contenuto in `dati`, la quale a sua volta è posta all'interno della home directory dell'utente pippo. A `~/dati/indirizzi`, invece, corrisponde il file `indirizzi`, contenuto in `dati`, posta all'interno della propria home directory.

Per identificare la directory corrente si può usare il simbolo `.` (punto). Per copiare nella directory corrente il file `ricette` che l'utente pippo tiene nella sua sottodirectory `dati`, useremo quindi il comando

```
olivia> cp ~pippo/dati/ricette .
```

Allo stesso modo `..` identifica la directory padre della directory corrente

## Diritti di accesso ai file e file nascosti

Unix associa a ogni file un numero grazie a cui poter risalire in modo univoco all'utente che l'ha creato. Questi ha la possibilità di scegliere a proprio piacimento come rendere accessibile il file per sé e per gli altri.

Esistono tre tipi diversi di accesso (*r*, lettura; *w*, scrittura; *x*, esecuzione), settabili a tre livelli (*u*, utente; *g*, gruppo a cui l'utente appartiene; *o*, qualsiasi altro utente). Per le directory i tre tipi diversi di accesso corrispondono rispettivamente alla possibilità di leggere la lista dei file presenti, di creare nuovi file e di poter entrare nella directory stessa.

Un utente potrebbe volere che in una sua directory gli utenti del suo gruppo non possano né creare file né richiedere la lista di quelli già esistenti, ma possano in compenso entrarvi lo stesso (quasi "al buio") per poter accedere a una sua sottodirectory aperta ad es. in lettura e scrittura. Ciò può essere fatto azzerando, a livello di gruppo, i diritti di lettura/scrittura e attivando solo quello di esecuzione. Grazie all'opzione `-l` del comando `ls`, è possibile sapere di chi è un file e quali diritti esso ha, presentati nell'ordine utente-gruppo-altri, oltre ad altre utili informazioni come dimensioni e data dell'ultima modifica.

```
olivia> ls -l
-rwx----- 1 UTENTE studenti 15298 May 12 13:21 ciao*
-rw-rw-r-- 1 UTENTE studenti   387 Sep 10 10:43 note
-rw-r--r-- 1 UTENTE studenti 78204 Sep 21 16:05 ricette
olivia>
```

Nel precedente esempio, `ciao` è un programma che può essere modificato e eseguito solo dal proprietario, mentre `ricette` è aperto in lettura a tutti, ma modificabile solo dal possessore.

Infine `note` è aperto in scrittura anche agli altri utenti del gruppo studenti.

Per modificare i diritti di accesso a un file, si usa il comando `chmod` <livello> <azione> <diritto> file, in cui livello è uno o più caratteri tra *u*, *g*, *o* e *a*, azione è uguale al simbolo `+` oppure `-`, mentre diritto è una combinazione di *r*, *w* e *x*. Per impedire agli altri utenti del gruppo studenti di scrivere nel file `note`, useremo il seguente comando

```
olivia> chmod g-w note
olivia> ls -l
-rwx----- 1 UTENTE studenti 15298 May 12 13:21 ciao*
-rw-r--r-- 1 UTENTE studenti   387 Sep 10 10:43 note
-rw-r--r-- 1 UTENTE studenti 78204 Sep 21 16:05 ricette
olivia>
```

Esiste una categoria di file *nascosti* che non viene visualizzata con il comando `ls`, a meno di specificare l'opzione `-a`. Sono i file il cui nome inizia con un punto (per esempio `.emacs`), in genere si tratta di file di configurazione di alcuni programmi.

## Esaminare i file

Il modo più semplice per esaminare il contenuto di uno o più file è usare il comando *less*. Se per esempio si vuole dare un'occhiata al contenuto del file `pippo`.

```
olivia> less pippo
Pippo e topolino uscirono con pluto...
```

Se il file è abbastanza piccolo da riempire meno di una schermata, il comando ritorna immediatamente al prompt della shell. Altrimenti, nell'angolo sinistro in basso comparirà un messaggio evidenziato del tipo

```
pippo lines 1-12/110 1%
```

dove viene indicato il nome del file, il numero le linee visualizzate nella schermata attuale sul numero di linee totali e la porzione di file mostrata nella schermata in percentuale. Per poter visualizzare il resto del file, occorre utilizzare i tasti cursore, per spostarsi indietro e in avanti di una riga, o *Page Up* e *Page Down*, per spostarsi di una pagina alla volta. Se si vuole terminare l'esecuzione senza vedere il resto del file, basta premere il tasto `q`.

## Modificare i file

Un altro modo per esaminare i file è usare un cosiddetto *editor* o editore di file. In aggiunta alla possibilità di vederne il contenuto, un editor ha l'innegabile vantaggio di permetterne la modifica, oltre a consentire di crearne di nuovi.

Esistono tanti editor diversi per Unix, ma quelli più usati sono

- **emacs** (completissimo e relativamente semplice da usare per compiti poco complessi)
- **pico** (veloce, pratico e semplice, ma poco adatto per usi avanzati; è l'editor usato dal programma di EMail pine nella stesura dei messaggi)
- **vi** (potente, ma per nulla intuitivo per chi è alle prime armi; è il preferito da chiunque professi Unix come una religione)

Descriviamo ora le procedure da seguire per creare un nuovo file (che nell'esempio sarà chiamato `primofile`) usando Emacs e Pico.

## Emacs

Dopo aver digitato il nome del comando seguito dal file scelto, ossia

```
olivia> emacs nomefile
```

lo schermo viene cancellato e appare il contenuto del file.

Se il file non esiste, lo schermo sarà vuoto, ad eccezione delle due righe inferiori, usate dal programma per dare informazioni all'utente. A questo punto si potrà inserire il testo e modificarlo usando i tasti di movimento e di cancellazione (vedi figura).



Una volta che le operazioni di modifica sono finite occorre salvare su disco il nuovo contenuto del file. Per far ciò basta premere il tasto Control assieme a X, seguito da Control-S. Per uscire da Emacs, la combinazione di tasti è Control-X Control-C. In alternativa, invece di salvare e poi uscire, si può uscire direttamente. Il programma chiederà se si vuole salvare il file appena modificato, domanda alla quale ovviamente si risponderà di sì. Ecco alcune combinazioni di tasti più usate:

Combinazione tasti	Funzione
Control-x Control-f	Carica un altro file in memoria
Control-x Control-s	Salva il file corrente
Control-x i	Inserisce nel documento il contenuto di un altro file
Control-x Control-w	Salva il testo corrente con un altro nome
Control-s	Cerca una stringa di caratteri
Control-v	Va alla schermata successiva
Alt-v	Va alla schermata precedente
Control-a	Sposta il cursore a inizio riga
Control-e	Sposta il cursore a fine riga
Control-Spazio	Delimita blocco di testo
Control-w	Taglia il blocco di testo delimitato
Control-y	Incolla il blocco di testo tagliato
Control-h	Sistema di aiuto in linea

## Pico

Dopo aver digitato

```
olivia> pico nomefile
```

comparirà la maschera di Pico. Per inserire e modificare il testo si possono usare i tasti di movimento e cancellazione. Una volta esaurite le modifiche al file, basterà premere contemporaneamente i tasti Control e X per uscire da Pico. Il programma chiederà prima se si vuole memorizzare il contenuto attuale del file, domanda a cui si risponderà y (yes) e quindi il nome del file stesso. Premendo il tasto Invio (Enter) si confermerà il nome già inserito.

## Spazio disco e quota

I file occupano spazio. Senza un periodico controllo del loro numero e delle loro dimensioni, un qualsiasi disco si riempie in poco tempo. L'utente ha l'obbligo morale di tenere il proprio consumo di spazio sul disco al minimo possibile, rimuovendo

eventuali file non usati.

In aiuto di sistemisti e utenti, esiste un meccanismo automatico di controllo. Esso prevede che ogni utente abbia un limite fissato di spazio su disco a propria disposizione, chiamato *quota*. Una volta superata la quota, il sistema avvertirà ad ogni collegamento che in un certo lasso di tempo (in genere una settimana) bisogna rimuovere l'eccedenza in modo da rientrare nella quota assegnata. Nel caso in cui non si eliminasse tale eccedenza, al termine di tale scadenza, non sarà più possibile creare alcun file.

Tale situazione può essere evitata tenendo costantemente sotto controllo lo spazio occupato e la quota assegnata. Il comando da usare è *quota -v*.

```
olivia> quota -v
Disk quotas for UTENTE (uid 29345):
Filesystem usage quota limit timeleft files quota limit timeleft
/a/isotta/local/disc2
         460 1000 1200                0    0    0
olivia>
```

Le cifre da considerare sono le prime tre: 460, 1000 e 1200. Esse rappresentano rispettivamente in Kilobyte l'uso attuale, la quota assegnata e il limite massimo di spazio a disposizione. Il limite massimo non può MAI essere superato, pena l'impossibilità di creare file nuovi, superando, invece, la quota si riceverà semplicemente messaggi di errore all'inizio di ogni sessione. Dopo il tempo indicato nella colonna *timeleft*, se vi è ancora troppo spazio in uso su disco, sarà possibile soltanto la lettura e la cancellazione dei file. Resta comunque preferibile restare il più possibile al di sotto della quota.

Con i comandi *df* e *du* è possibile vedere rispettivamente lo spazio disco occupato sui vari dischi e l'occupazione della directory corrente. Ecco un esempio del loro uso

```
croma20> df
Filesystem          1k-blocks      Used Available Use% Mounted on
/dev/hda6            202220        73226   118554   38% /
/dev/hda5            3842376       820412  2826776   22% /l/disc1
/dev/hda1            2016016       1182840  730764   62% /usr
homeserver:/RAID/5  17287332     11464884  4930052   70%
/.automount/homeserver/RAID/5
croma20> du
4      ./cache
4      ./archive
4      ./xover-cache
1092  .
croma20>
```

Nel caso ci si accorga con controlli regolari che ci si sta avvicinando alla soglia, è consigliabile dare una ripulita alla home directory cancellando i file inutili o poco usati.

La quota base per tutti è di 15 Mb. Frequentando i corsi a cui è associato un laboratorio si ottiene in genere un aumento della quota disco e delle risorse utilizzabili.

## Compressione dei file

Per ridurre l'occupazione su disco dei file o per consentirne un invio per posta più rapido spesso si usa comprimere uno o più file grazie ad alcuni programmi appositamente studiati.

Nella tabella che segue c'è l'elenco delle estensioni dei file e il nome del programma che lo gestisce. Per maggiori informazioni sull'uso di tali programmi si rinvia al manuale on-line (vedasi l'apposita sezione).

.tar	tar	crea un unico file che contiene vari file e directory, senza effettuare compressione
------	-----	--

<code>.gz</code>	<code>gzip (gunzip)</code>	comprime un unico file
<code>.tar.gz</code> <code>.tgz</code>	tar e gzip	file creato con tar e compresso con gzip
<code>.zip</code>	zip (unzip)	uno o più file compressi
<code>.Z</code>	compress (uncompress)	comprime un file
<code>.bzip2</code>	bzip2 (bunzip2)	simile a zip, ma con un migliore rapporto di compressione

## Stampa

Dal momento che la stampante è un dispositivo relativamente lento e che spesso più utenti tentano di accedervi, il sistema operativo mantiene una lista (chiamata *coda*, dall'inglese *queue*) di tutto il materiale inviato e, uno alla volta, manda in stampa i file (chiamati in questo caso anche *job*, "lavori") degli utenti. Il comando da usare per la stampa di un file è `lpr`.

```
olivia> lpr -Plp sched/sched.h
```

Per stampare invece sulla stampante laser (da usarsi solo per tesi e relazioni) si usa uno dei due comandi seguenti:

```
olivia> lpr -Plj0 sched/sched.h
olivia> lj sched/sched.h
```

Il parametro `-P<nomestampante>` indica su quale stampante inviare la stampa (la stampante ad aghi `lp` o la laser `lj0`). Ogni stampa inserita nella coda ha un numero, chiamato *id* (perché identifica in modo univoco il file/job). Grazie al comando `lpq` è possibile in qualsiasi momento sapere se vi sono eventuali file ancora da stampare, con tanto di posizione all'interno della coda, dimensione (in byte) e il numero del Job.

```
olivia> lpq -Plp
lp is ready and printing
Rank  Owner      Job  Files                Total Size
1st   UTENTE     530  sched/sched.h       212 bytes
olivia> lpr -Plp sched/fifo.h
olivia> lpq -Plp
lp is ready and printing
Rank  Owner      Job  Files                Total Size
1st   UTENTE     530  sched/sched.h       212 bytes
2nd   UTENTE     531  sched/fifo.h        194 bytes
olivia>
```

Nel caso si voglia annullare una stampa ancora in coda, il comando da usare è `lprm`, seguito dall'id della stampa.

```
olivia> lprm 530
athena: dfA530carlotta dequeued
athena: cfA530carlotta dequeued
olivia> lpq -Plp
lp is ready and printing
Rank  Owner      Job  Files                Total Size
1st   UTENTE     531  sched/fifo.h        194 bytes
olivia>
```

Anche per le stampe è stata istituita una quota per evitare abusi. Tale quota riguarda solo le stampe inviate sulla stampante laser e può essere ampliato con l'attivazione di alcuni laboratori. Non esiste quindi una quota di massima per la stampante ad aghi.

## Metacaratteri

Spesso risulta comodo poter accorciare il nome di un file quando lo si usa come argomento di un comando. Unix permette, tramite alcuni caratteri speciali, di indicare in maniera compatta un numero arbitrario di file. Tali caratteri sono chiamati *wildcard* (o caratteri *jolly*).

I più comuni sono ? e \*. Il punto interrogativo indica un qualsiasi carattere. Ad esempio, per riferirsi ai file *matto*, *ratto* e *gatto*, si può usare l'espressione *?atto*. Un file chiamato *scatto* non verrà preso in considerazione. L'asterisco, invece, è ancora più generico e può essere specificato al posto di una qualsiasi sequenza di caratteri (zero, uno o più di uno). L'espressione *p\*to*, ad esempio, può essere usata per sostituire tutti i nomi che iniziano con *p* e finiscono in *to* (*pianto*, *partito*, *presunto*, *provato* ecc.). Si possono usare i caratteri jolly più volte all'interno di una espressione o combinarli assieme. Ad esempio, è possibile usare *?re\*r?* per indicare nomi come *credere*, *tremare* o *preliminari*.

Sebbene le wildcard siano utilissime quando si ha a che fare con un gran numero di file, bisogna sempre porre estrema attenzione nel farne uso. Si potrebbe per errore includere anche file che non si desidera specificare. In particolare i caratteri jolly e il comando *rm* sono una combinazione micidiale: eventuali espressioni digitate o pensate in modo errato possono far perdere il lavoro di ore, giorni o settimane. Può capitare di voler cancellare i file che iniziano per *c* e digitare per errore *rm c \** (con uno spazio tra la *c* e l'asterisco) invece che *rm c\**. In tal caso *rm* riterrebbe che l'utente vuole eliminare *c* e i file che corrispondono a *\** (tutti i nomi che contengono una qualsiasi stringa, cioè tutto il contenuto della directory corrente!).

## Uso dei floppy con gli Mtools

L'uso dei floppy disk è semplificato dagli Mtools, comandi il cui uso è simile ai corrispondenti comandi DOS per la gestione dei file.

Esistono vari comandi tipo *mdir*, *mcopy*, *mformat* e altri (in generale il nome di un comando DOS con una *m* davanti) il cui uso è del tutto analogo ai corrispondenti comandi DOS.

Anche in questo caso, per brevità, ci limiteremo a un esempio, rimandando al manuale on-line per ulteriori dettagli.

```
olivia> mdir a:
Volume in drive A has no label
Directory for A:/

No files
                                1 457 664 bytes free

olivia> mcopy stat a:
olivia> mdir a:
Volume in drive A has no label
Directory for A:/

stat          1567 09-19-2000   1:13
      1 file                1 567 bytes
                                1 455 616 bytes free

olivia>
```

## Redirezione

Tramite il carattere *>*, seguito da un nome di un file, si può far sì che l'output di un comando (il testo che normalmente il programma stampa a schermo) venga scritto in tale file, piuttosto che inviato al terminale. Questa tecnica è chiamata *redirezione*. Nel caso tale file esista già, esso viene rimpiazzato con il nuovo contenuto. Usando invece *>>* (due simboli di maggiore), l'output viene accodato a ciò che era già presente nel file: se esso non esiste o è vuoto, il risultato sarà identico a

quello prodotto da >.

Il carattere < invece permette di specificare il nome di un file dal quale prelevare l'input per un comando. In pratica, il comando leggerà ciò che è contenuto nel file come se fosse stato immesso da tastiera.

Il comando **cat** permette di sfruttare in modo efficace la redirectione. Esso in pratica restituisce in output il contenuto di tutti i file specificati. Se nessun file viene specificato, allora verrà usato ciò che l'utente immette tramite il terminale. Il comando `cat > file` permette di scrivere su un file ciò che si digita da tastiera, finché non si preme Control-D, combinazione di tasti che indica che si è arrivati alla fine del file. Infatti, non essendo stato specificato nessun file come suo argomento, legge i caratteri in arrivo dal terminale (cioè ciò che l'utente digita) e li rispedisce in uscita. Dal momento che tramite il simbolo > l'output è stato rediretto, i caratteri verranno scritti sul file piuttosto che essere inviati di nuovo al terminale.

```
olivia> cat numeridellotto
12 27 42 71 83
olivia> cat <numeridellotto >numeri
olivia> cat numeri
12 27 42 71 83
olivia> cat numeridellotto numeri
12 27 42 71 83
12 27 42 71 83
olivia> cat numeridellotto numeri >ancoranumeri
olivia> cat ancoranumeri
12 27 42 71 83
12 27 42 71 83
```

## Pipeline

Il meccanismo della redirectione è ottimo quando si ha a che fare con i file, ma può capitare di dover usare l'output di un comando come input di un altro. La shell riserva a tale scopo il carattere | (*pipe*, in italiano tubatura o condutture). Basterà digitare i comandi da eseguire nell'ordine, separati dalla pipe. La shell eseguirà il primo comando, per poi passare il suo output come input del secondo, dopo la cui esecuzione verrà eseguito il terzo e così via.

Il comando **sort** restituisce in output le linee che legge dall'input ordinandole. Diventa quindi semplicissimo riordinare i contenuti di un file e mostrare il risultato a schermo, senza dover creare nessun file temporaneo di appoggio:

```
olivia> cat miofile | sort | less
```

In realtà si sarebbe potuto immettere anche solo `sort miofile | less`, ma l'esempio voleva mostrare l'uso di più pipe.

## Ricerca nei file

Per agevolare la ricerca di parti di testo all'interno di un gran numero di file, Unix fornisce un comando, **grep**, che evita all'utente di dover cercare da sé analizzando ogni singolo file con un editor o `less`. La sintassi base del comando è `grep stringa listafile`.

Il comando cercherà all'interno di ognuno dei file indicati la stringa specificata e, per ogni occorrenza stamperà il nome del file seguito dalla riga di testo contenente la parola.

```
olivia> grep terr *
racconto: con sommo terrore vide che la guardia non dava segni di vita
ricette: uova, quindi agitare il tutto nella terrina.
olivia>
```

Nell'esempio abbiamo cercato "terr" all'interno di tutti i file (usando la wildcard \*) nella directory corrente e il comando `grep` ha trovato due occorrenze nei file `racconto` e `ricette`.

Notare che non è necessario indicare una parola intera ma anche solo parti di essa. Per un uso avanzato delle opzioni di `grep`, consultare il manuale in linea (come spiegato nella sezione Manuali).

## Trovare un file

Può capitare di cercare un file di cui si conosce il nome ma non ci si ricorda in quale directory sia contenuto. In tal caso si può usare il comando `locate`:

```
olivia> locate hosts.allow
/etc/hosts.allow
olivia> cd /etc
olivia> ls hosts.allow
 4 hosts.allow
olivia>
```

Tale comando, essendo basato su un database, potrebbe essere non aggiornato e non funzionare su file remoti. In alternativa si può usare il comando `find`, dalla sintassi molto più complessa. Per maggiori informazioni consultare il manuale on-line (vedere sezione relativa).

## Gli alias

Il comando `alias` permette di creare abbreviazioni per comandi usati frequentemente o dalla sintassi complessa. Se invocato senza argomenti, esso stampa la lista delle sostituzioni (o appunto alias) che l'utente ha finora definito. Altrimenti, si indicherà come primo argomento l'abbreviazione e come secondo il comando che essa sostituisce.

Un comando che può capitare di usare spesso è `ls -l`. Si può allora usare `alias ll ls -l` per far sì che ogni volta che si digita `ll`, il sistema si comporti come se in realtà si fosse immesso `ls -l`.

Di solito, quando un nuovo account viene creato, all'interno dei file di configurazione di sistema sono presenti alcuni alias predefiniti. In particolare, i file di avvio leggono (se esistente) il file `.aliases` nella home directory dell'utente. Se quindi si vuole aggiungere qualche alias in maniera permanente, basterà editare tale file.

## Manuali

La fonte primaria di documentazione in linea è il comando `man`, che mostra a schermo il manuale del comando richiesto. Il manuale contiene una breve descrizione del comando e la relativa sintassi, seguiti poi da una descrizione dettagliata dei parametri e delle opzioni che il comando prevede. Per esempio, per sapere quali sono i parametri accettati dal comando `cp`, basterà usare

```
olivia> man cp
Reformatting page.  Wait... done

CP(1)                                USER COMMANDS                                CP(1)

NAME
  cp - copy files

SYNOPSIS
  cp [ -ip ] filename1 filename2
  cp -rR [ -ip ] directory1 directory2
  cp [ -iprR ] filename ... directory

DESCRIPTION
  cp copies the contents of filename1 onto filename2.  The
  mode and owner of filename2 are preserved if it already
```

existed; the mode of the source file is used otherwise. If filename1 is a symbolic link, or a duplicate hard link, the contents of the file that the link refers to are copied; links are not preserved.

In the second form, cp recursively copies directory1, along with its contents and subdirectories, to directory2. If directory2 does not exist, cp creates it and duplicates the

--More--(26%)

Se non si conosce il nome esatto del comando, si può provare a usare l'opzione -k di man, seguita da una parola chiave. Per cercare tutti i manuali attinenti alla copia, si digiterà il comando

```
olivia> man -k copy
```

a cui il sistema operativo risponderà elencando i manuali e una breve descrizione di ognuno.

## I processi

Ogni programma in esecuzione è detto *processo*. Processi particolarmente lunghi (e noiosi) possono bloccare la shell per lunghi periodi di tempo prima di restituire il controllo all'utente (prompt).

In certi casi può essere utile l'esecuzione del programma in background che consiste nell'avvio del processo richiesto e nella immediata restituzione del controllo all'utente che può impartire nuovi comandi mentre il PC continua l'esecuzione del programma.

Per eseguire un comando in background basta aggiungere il simbolo & al termine della riga di comando.

Come esempio possiamo comprimere una directory con il seguente comando:

```
olivia> zip -q progetto.zip progetto/* &
[1] 31217
olivia> ls
progetto/  doc/
olivia>
[1]+  Done                  zip -q progetto.zip progetto/*
olivia>
```

È facile notare come abbiamo avviato il comando zip e il prompt (apparso subito) ci indica che il sistema è pronto per accettare nuovi comandi. Noi chiediamo il listato della directory e, durante l'esecuzione di ls, contemporaneamente il sistema esegue la compressione della directory progetto. Quando il comando eseguito in background termina, il sistema scrive un messaggio indicando il numero del processo (*pid*) e la riga di comando.

In generale su una macchina esistono più processi in esecuzione che si alternano nell'uso del processore. Per visualizzare i processi in esecuzione si usano i comandi *ps* e *top* (che visualizza i processi ordinandoli secondo la percentuale di uso del processore). A causa della grande quantità di opzioni e parametri mostriamo solo un piccolo esempio, rimandando al manuale on-line (vedasi sezione apposita) per un approfondimento sul loro funzionamento.

```
olivia> ps
  PID TTY          TIME CMD
 6088 pts/1    00:00:00 bash
 31262 pts/2    00:00:00 jove
 31263 pts/2    00:00:00 ps
olivia>
```

È anche possibile *uccidere* un processo, ovvero terminare (prematuramente) l'esecuzione di un programma mediante l'uso del comando *kill* e indicando il numero del processo da uccidere (pid)

```
olivia> kill 31262
[1]+  Killed                  jove
olivia>
```

## Cos'è la posta elettronica

Uno dei servizi più importanti disponibili attraverso l'accesso ad Internet è di sicuro la *posta elettronica*. Con tale termine si designa la possibilità di spedire, attraverso la rete, del testo (ma anche grafica o suoni) ad un altro utente, che troverà nella sua *mailbox* (casella postale) i dati da noi inviati. Il punto di forza della email sta nella rapidità con cui è possibile spedire o ricevere mail: per esempio, un utente negli USA in genere si ritroverebbe una nostra lettera nella mailbox dopo pochi secondi dall'invio. Concettualmente si tratta di una vera e propria rivoluzione, ormai in atto da diversi anni e che probabilmente è destinata a prendere sempre più piede fino a soppiantare il tradizionale servizio postale, a cui in gergo ci si riferisce col termine *snail mail* (in inglese, snail vuol dire lumaca).

### Come funziona

Vediamo di capire a grandi linee come funziona il servizio di posta elettronica. Quando inviamo una email, essa raggiunge prima di tutto il cosiddetto *mailserver*, una macchina destinata cioè a ricevere e smistare la posta, sia in partenza che in arrivo. Da qui viene stabilito un percorso fino al mailserver del destinatario, e da quel momento il nostro messaggio passerà temporaneamente attraverso diversi sistemi dedicati al recapito, fino ad arrivare a destinazione, dove il messaggio verrà salvato nella mailbox dell'utente.

Molto spesso la fase dei vari passaggi intermedi viene saltata, ove sia possibile una connessione diretta fra i server origine e destinazione.

Per leggere i nuovi messaggi bisogna utilizzare un programma apposito, che nel caso dei sistemi Unix del nostro Centro di Calcolo può essere *pine* (vedasi a tal proposito la sezione apposita) oppure *netscape* (vedasi la sezione apposita).

La lettura della posta non presuppone la necessità di utilizzare Unix ed il suo software: chi si collega da casa, infatti, può leggere e inviare posta elettronica direttamente dal suo computer. I software generalmente più usati in tal caso sono Outlook (Windows) o Mozilla/Thunderbird (Windows, Linux o MacOS).

Il nostro mailserver implementa i protocolli POP3, SMTP e IMAP per la gestione remota della posta elettronica. Esso permette a programmi come Thunderbird di collegarsi direttamente al mailserver per prelevare la posta in arrivo e spedire quella in uscita. Tali applicativi quindi consentono la lettura e la stesura di posta elettronica anche quando non si è collegati, memorizzando il tutto sull'harddisk del nostro computer in attesa del collegamento successivo.

Per chiudere questa sezione, accenniamo ad un problema molto dibattuto: quello della sicurezza dei dati trasmessi per posta elettronica. Teoricamente chiunque abbia accesso al mailserver o alle macchine poste lungo il tragitto che la posta elettronica compie per giungere a destinazione può leggere la nostra posta. Se ciò non comporta particolari problemi ad un normale utente della rete, è invece molto importante per ditte e aziende che volessero scambiare velocemente dei documenti (anche riservati) con terzi enti o persone.

### Gli indirizzi Email

Ovviamente per sapere a chi inviare posta elettronica è necessario conoscerne l'indirizzo, così come dobbiamo conoscere il nostro. Una volta assegnato l'account, per ottenere il nostro indirizzo di email basterà unire il nome dato al nostro account alla stringa `@cli.di.unipi.it`. Così per esempio, lo studente Luca Viola a cui è stato assegnato l'account `viola` avrà come indirizzo email `viola@cli.di.unipi.it`, a cui dovranno fare riferimento tutti coloro i quali volessero contattarlo via posta elettronica.

Ricordiamo che i nostri docenti hanno un indirizzo email sulle macchine del Dipartimento di Informatica, il cui uso viene consigliato. In genere tale indirizzo si può determinare dal cognome del docente unito alla stringa `@di.unipi.it`.

Mediante posta elettronica sono disponibili servizi come le *mailing list* che sono delle liste di utenti ai quali viene inviato contemporaneamente lo stesso messaggio, organizzando così una sorta di forum per le discussioni su argomenti di comune interesse.

## Uso di pine

Pine è un programma di posta elettronica disponibile sui sistemi Unix (ma non solo) che permette una gestione completa della posta elettronica e dei gruppi di discussione Usenet.  
Per avviare Pine, digitare al prompt della shell

```
olivia> pine
```

e premere Invio. La prima volta (e solo quella) apparirà una schermata di presentazione. Premere tranquillamente N e apparirà la schermata principale.

A questo punto inserite username e password per la connessione al mailserver.

```
PINE 3.91      MAIN MENU                               Folder: INBOX  1 Message
?      HELP                -  Get help using Pine
C      COMPOSE MESSAGE     -  Compose and send/post a message
I      FOLDER INDEX        -  View messages in current folder
L      FOLDER LIST         -  Select a folder OR news group to view
A      ADDRESS BOOK        -  Update address book
S      SETUP               -  Configure or update Pine
Q      QUIT                -  Exit the Pine program

Copyright 1989-1994.  PINE is a trademark of the University of Washington.
                        [Folder "INBOX" opened with 3 messages]
? Help                P PrevCmd                R RelNotes
O OTHER CMDS L [ListFldrs] N NextCmd          K KBlock
```

Per eseguire una opzione occorre selezionarla con i tasti cursore e premere Invio o, più velocemente, premere il tasto indicato nella prima colonna.

Pine mostra sempre nelle ultime due righe dello schermo la lista completa dei comandi disponibili e i relativi tasti da premere. Quando lo spazio non è sufficiente per contenerli tutti, il tasto O (Other Commands) permette di vedere i rimanenti. L'accento circonflesso indica che il tasto deve essere premuto assieme al tasto Control (ad esempio ^C sta per Control-C).

Ora faremo una breve introduzione alle funzionalità di pine; per gli incontentabili è a disposizione una guida molto chiara e completa (in inglese) premendo ? dalla schermata principale.

L'aiuto è sensibile al contesto: premendo quindi ? da una schermata che non sia quella principale, pine intelligentemente fornisce informazioni sul comando che si sta eseguendo in quel momento.

### Scrivere e spedire un messaggio

Il comando Compose Message permette di scrivere un messaggio da inviare a uno o più utenti.

```
PINE 3.91      COMPOSE MESSAGE                               Folder: INBOX  1 Message
To          :
Cc          :
Attchmnt   :
Subject    :
----- Message Text -----
```

```

^G Get Help ^X Send ^R Rich Hdr ^Y PrvPg/Top ^K Cut Line ^O Postpone
^C Cancel ^D Del Char ^J Attach ^V NxtPg/End ^U UnDel Line ^T To AddrBk

```

Le prime righe (intestazione o *header*) del messaggio devono essere completate come spiegato in tabella.

Campo	Significato
To	Indirizzo del o dei destinatari
Cc	Indirizzi ai quali inviare copia del messaggio
Attachmnt	File da allegare al messaggio
Subject	Oggetto (breve descrizione) del messaggio

In un campo possono essere inseriti anche più indirizzi, separati da virgole.

Il campo Attachmnt serve solo quando si vuole inviare un qualsiasi file mediante posta elettronica (vedasi la sezione apposita) e deve quindi essere ignorato per semplici messaggi di solo testo.

A questo punto si può iniziare a scrivere il messaggio vero e proprio, sotto la riga

```

----- Message Text -----

```

Quando il testo è pronto, premendo Control-X si può scegliere se spedire effettivamente la mail (Y), o continuare a scrivere (N). Se, invece, si vuole rinunciare a spedire la mail e tornare alla schermata principale, premere Control-C seguito da Y.

### Gestire la posta con pine

Pine utilizza dei contenitori, detti *folder*, nei quali ordina la posta in arrivo e quella conservata dall'utente. Il folder contenente la posta in arrivo ha nome INBOX; gli altri folder vengono creati mano a mano che l'utente chiede di memorizzare la posta.

La prima riga dello schermo mostra in ogni momento il nome del folder in uso; all'avvio esso è INBOX, la nostra "cassetta della posta". Con il comando Folder Index (I) si può vedere l'elenco dei messaggi presenti.

```

PINE 3.91 FOLDER INDEX Folder: INBOX Message 2 of 2 NEW
+ 1 Nov 26 Bove Ercole, Luca (3,966) Nuovo www sex-site!
  N 2 Nov 30 Luca Barbarossa (1,036) Luca si buca ancora
  N 3 Nov 27 Marco Masini (1,771) Perché lo fa?
  N 4 Dec 23 Umberto Pone (1,890) Sistemi Equivoci

```

```
? Help      M Main Menu P PrevMsg    - PrevPage  D Delete    R Reply
O OTHER CMDS V [ViewMsg] N NextMsg  Spc NextPage U Undelete  F Forward
```

Di ogni messaggio vengono indicati numero progressivo, data di invio, mittente, lunghezza in byte e l'oggetto. Ogni riga può essere preceduta da caratteri indicano lo "stato" del messaggio, come indicato in tabella.

Carattere	Significato
N	New (nuova): la mail non è stata ancora letta
A	Answered (risposta): è stata inviata una risposta all'autore
D	Deleted (cancellata)
+	La mail è indirizzata all'utente

Il simbolo + indica che siamo il destinatario principale del messaggio; non compare se il messaggio è stato inviato a più persone, oppure se quello a noi diretto è una copia.

Per leggere un messaggio occorre selezionarlo usando i tasti cursore e premere Invio.

Per rispondere a un messaggio si usa il comando Reply (R). Alla domanda "Include original message in Reply?" rispondere Y per inserire nella risposta una copia dell'originale, N altrimenti. Sullo schermo appare ora la schermata di composizione, con la riga del destinatario già completata. Basta immettere il testo della risposta, poi spedire con Control-X seguito da Y per confermare.

Per inoltrare ad altri un messaggio si usa il comando Forward (F). Pine presenta la schermata di composizione, con il cursore sulla riga del nuovo destinatario. Basta immettere l'indirizzo, poi spedire con Control-X seguito da Y per confermare.

Il comando Delete (D) serve per cancellare dalla cassetta della posta il messaggio selezionato: esso viene marcato con la lettera D e non sarà più visibile nelle successive sessioni di Pine.

Undelete (U) serve a recuperare il messaggio. Il comando Expunge (X) cancella immediatamente i messaggi marcati con D (attenzione!).

È possibile salvare un messaggio con il comando Save (S): pine mostra un nome di folder (ricavato dal nome del mittente); è possibile adottare quello proposto oppure inserirne un altro. Il messaggio verrà cancellato da INBOX e salvato nel folder specificato.

Per accedere ai messaggi salvati in un folder, occorre utilizzare il comando Folder List (L).

```
PINE 3.91    FOLDER LIST                                Folder: INBOX  1 Message
```

```
-----
Folder-collection <mail/[]>  ** Default for Saves **          (Local)
-----
```

[ Select Here to See Expanded List ]

```
-----
News-collection <News on newsserver.unipi.it>                  (Remote)
-----
```

[ Select Here to See Expanded List ]

```
? Help      M Main Menu P PrevFldr - PrevPage  D Delete    R Rename
O OTHER CMDS V [Select] N NextFldr Spc NextPage  A Add
```

Da questa schermata è possibile selezionare non solo i folder della posta, ma anche i gruppi news di Usenet. Adesso ci interessano i primi, quindi selezioniamo con i tasti cursore la prima delle due righe

[ Select Here to See Expanded List ]

e premiamo Invio. Pine mostra la lista dei folder, dalla quale è possibile finalmente scegliere quale consultare.

```
PINE 3.91  FOLDER LIST                                     Folder: INBOX  1 Message
```

```
-----
Folder-collection <mail/[]>  ** Default for Saves **          (Local)
-----
```

```
INBOX      masini      barbaros
```

```
-----
News-collection <News on newsserver.unipi.it>                  (Remote)
-----
```

[ Select Here to See Expanded List ]

```
? Help      M Main Menu P PrevFldr - PrevPage  D Delete    R Rename
O OTHER CMDS V [ViewFldr] N NextFldr Spc NextPage  A Add
```

## **Come inviare un file**

Pine permette di allegare a un messaggio di testo un qualsiasi file e spedire il tutto in un'unica mail. Per sfruttare questa caratteristica basta posizionare il cursore sul campo Atchmnt e inserire il nome del file (o più nomi separati da virgole). Se non si ricorda il nome, premendo Control-T pine mostra il contenuto della home directory; basta selezionare con i tasti cursore il file che si intende allegare e premere Invio.

## **La rubrica degli indirizzi**

Gli indirizzi di posta elettronica sono difficili da ricordare. Per evitare di commettere errori nella loro digitazione, pine mette a disposizione una rubrica elettronica: ad ogni indirizzo può essere associato un soprannome o *nickname*, facile da ricordare, da usare poi in luogo del vero indirizzo.

Per modificare la rubrica, scegliere Address Book dalla schermata principale. Con il comando Add (A) si aggiunge un nuovo indirizzo: pine richiede prima cognome e nome, poi il nickname e infine l'indirizzo di posta elettronica. Il comando Delete (D) cancella un indirizzo dalla rubrica; Edit (E) permette di modificare i dati inseriti; Main (M) riporta alla schermata principale.

Se si sta leggendo una mail e si vuole inserire nella rubrica l'indirizzo del mittente, basta usare il comando TakeAddress (T): pine chiede solo il nickname da associare e provvede da sé a impostare cognome, nome e indirizzo nella rubrica.

## **X Window**

Chiunque di voi avvia nella sua vita utilizzato almeno per una volta Microsoft Windows o un Macintosh conosce benissimo l'utilità di un sistema grafico a finestre (**GUI: Graphical User Interface**). Anche nel mondo UNIX esiste un ambiente grafico che però non fa parte del sistema operativo ma si basa su un sistema di finestre X-Windows che sfrutta la tecnologia Client-Server.

### **Un po' di storia di Xwindows**

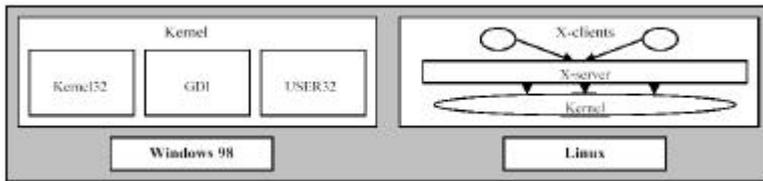
X Windows fu sviluppato nel 1984, da una cooperazione tra il MIT Massachusetts Institute of Technology (<http://www.mit.edu>) e la Digital Equipment Corporation (<http://www.digital.com>), con il nome in codice "**Athena**", da Robert Scheifler e il suo gruppo di collaboratori, basato sul software "W", sviluppato da Paul Asente, nella Università di Stanford. In quel periodo esisteva un vasto assortimento di workstation di differenti produttori che erano incompatibili tra loro, con il Sistema X-Windows si voleva creare un sistema di finestre indipendente dal sistema operativo e dal hardware.

Le prime versioni di X furono usate esclusivamente presso il MIT e alla Digital, ma dalla versione X 10.3, molte case costruttrici di hardware mostrarono interesse a tale sistema. Poi nel 1988, venne creato il MIT Consortium, tra IBM, HP, SUN, DIGITAL e SGI, per supportare X11 R3, sulle loro piattaforme, come piattaforma grafica nativa. Al consorzio si affiancarono in seguito, SCO e altre case. Oggi il MIT Consortium si è trasformato nel X-Consortium dove appartengono circa sessanta, firme tra software e hardware e' arrivato alla versione X11 R6.4, ed è disponibile per VAX/VMS della Digital, per tutte le piattaforme hardware UNIX, e per PC è disponibile per SCO, Solaris, Linux (Caldera OpenLinux, RedHat, Debian ecc.).

Una delle principali differenze tra X e gli altri sistemi a finestra è che X, non supporta un particolare e definito stile di interfaccia. Alcuni sistemi GUI, quali Macintosh della Apple e Microsoft Windows, supportano un particolare sistema di interfaccia. X invece permette di avere la massima flessibilità, evitando di fissare la veste o la così chiamata "*look and feel*" di qualsiasi tipo di interfaccia grafica.

### **X e il modello Client-Server**

A differenza del sistema operativo della Microsoft, in cui l'interfaccia grafica (GDI) e' parte integrante del kernel (per kernel si intende l'interfaccia tra hardware e processi che gestisce ogni periferica della macchina), nel mondo Unix tale modulo è rappresentato da un processo esterno dal nucleo che mette a disposizione di altre applicazioni le primitive grafiche .



Con il termine X si intende un server grafico in grado di controllare monitor, mouse e tastiera. Le varie applicazioni che devono fornire l'output sotto forma grafica dialogano con il Server X con un protocollo predefinito.

## La struttura

Gli elementi fondamentali del X Windows System sono:

- *X server.* Il programma che gestisce e mette a disposizione degli altri programmi le funzionalità grafiche.
- *X clients.* I programmi che sfruttano le funzionalità grafiche dell'X server, comunicando con quest'ultimo.
- *X protocol.* Il protocollo di comunicazione utilizzato per le transazioni tra Server e applicazioni.
- *X lib.* La libreria condivisa dai programmi che utilizzano i servizi del Server X.
- *Window Manager.* Il programma "speciale" che gestisce le finestre e quindi la veste grafica delle varie applicazioni.

Come abbiamo detto l'architettura del sistema grafico X-Windows, è basata sul modello *client-server*. Un singolo processo, noto come *server* (X-Server), controlla completamente tutti i dispositivi input e hardware. Il server crea e gestisce le finestre (windows), sullo schermo, produce testo e grafica gestendo gli *eventi* ricevuti da tastiera e mouse. Il server provvede a creare un interfacciamento tra l'hardware del video, display hardware, (X-Terminals, PC, Workstations) e le applicazioni, sviluppate.

Un'applicazione che sfrutta le facilitazioni offerte dal server viene chiamata *client*. Un client, comunica con il server, tramite la rete utilizzando protocolli asincroni (*x-protocol*), come TCP/IP, DECnet, UDS (Unix Domain Sockets), e altri.

Molti client, possono collegarsi contemporaneamente a un singolo server e un singolo client può collegarsi a diversi server, allo stesso tempo.

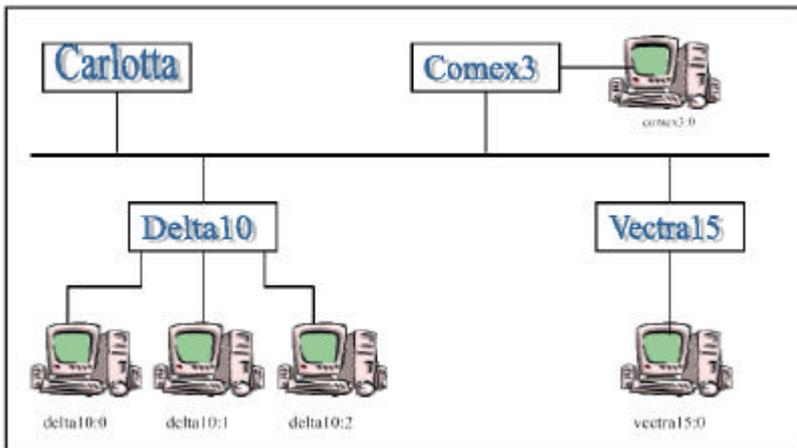
L'architettura X nasconde molti dei dettagli della sua implementazione device-independent del server e del hardware che controllano i clients. Qualsiasi client può comunicare con il server, provvedendo a entrambi: server e client, di ubbidire al X protocol.

## Displays & Screens

Displays e Screens sono concetti associati al server. Il termine display, nel momento in cui si parla di X Windows e' sinonimo del termine server e si riferisce alla periferica su cui sta girando il processo servente.

Il termine screens e' usualmente riferito a un device hardware di output (il monitor), sotto il controllo del display. Spesso un singolo display può avere accesso a più di un singolo screens, permettendo, così, di inviare applicazioni, non soltanto a stazioni grafiche remote, ma anche a **displays** (schermi) remoti.

Nella terminologia di X Windows il display viene identificato da un numero a partire da zero, nella forma **:n**. Se una stazione grafica è dotata di più di uno schermo, quando occorre far riferimento a uno di questi occorre aggiungere all'indicazione del numero del display quello dello schermo attraverso la seguente forma **:n.m** (dove n è il display e m lo schermo).



Nella figura precedente, ci sono 4 macchine nella rete: **Carlotta, Comex3, Delta10, Vectra15**

Tre di queste macchine hanno schermi grafici: **Comex3, Delta10, Vectra15**.

Ci sono 5 server in questa rete: **comex3:0, delta10:0, delta10:1, delta10:2, vectra15:0**.

I Client (inclusendo i loro windows manager, che possono essere diversi), risiedono su qualsiasi macchina e possono essere usati da qualsiasi utente. Supponiamo che ad esempio, comex3 desidera *inviare*, il suo programma **emacs**, sullo schermo numero 1 di delta10. Allora, comex3 eseguirà il comando:

**“emacs -display delta10:0.1”**

Se invece voleva inviarlo sul primo schermo, allora avrebbe scritto 0.0 al posto di 0.1. L'invio e la ricezioni di applicazioni X, viene gestita sfruttando la potenzialità del protocollo X, e del programma **xauth**, che permette di accedere o meno un programma sul display remoto.

Per una connessione remota occorre stabilite un collegamento. Il server X resta normalmente in ascolto sulla porta (socket) numero  $6000 + n$ , dove  $n$  rappresenta il numero della stazione grafica, ovvero del server X. Nel caso del server *delta10:1* la porta su cui dovrebbe trovarsi in ascolto il server relativo è la numero 6001.

X inoltre, come è il caso dell'utente vectra15, permette di gestire diversi monitor, (otto ad esempio), con un solo server, ma in questo caso il server è sempre lo stesso: **vectra15:0.0**

E' possibile sfruttando questa architettura, lanciare applicazioni dal Centro di Calcolo e, o visualizzare l'applicazione su un display al Dipartimento, sfruttando il protocollo TCP/IP, via Internet.

### **Windows managers**

Nell'architettura X né il client conoscono l'aspetto e il comportamento del sistema di finestre. I bordi delle finestre, i meccanismi per la creazione e lo spostamento delle finestre, i menu sullo schermo e molte altre caratteristiche generali, sono realizzate da un terzo componente dell'architettura. Il gestore delle finestre o window manager (WM) è quel speciale programma client, che si occupa di incorniciare le superfici degli altri programmi client, di gestire la messa a fuoco, e cioè del passaggio da un programma all'altro e di altre funzioni di contorno. In altre parole il window manager può cambiare la faccia e il funzionamento operativo del sistema X-Windows e quindi ne rappresenta la veste grafica. Con X, si è liberi di utilizzare l'interfaccia piu' appropriata per lavorare, e può essere cambiata a piacere con un'altra, senza nessun costo aggiuntivo permettendo così di avere più interfacce su una sola macchina, in modo che l'importanza passi dall'interfaccia, al sistema operativo e al Back-End.

### **Avviare X Window**

A seconda del computer che si utilizza la procedura necessaria per l'avvio di X può variare. Nel caso dei PC più nuovi non è necessaria alcuna operazione, visto che il sistema parte già in modalità grafico.

Le macchine meno recenti, invece, normalmente attivano una semplice sessione in modo testo. Con il comando `startx` è possibile lanciare il server X.

```
comex17> startx
```

## File di configurazione

Uno dei pregi più rilevanti di X è la sua estrema configurabilità.

Al suo avvio, il server si occupa di eseguire i comandi contenuti nel file `~/.xinitrc`, permettendo quindi all'utente di far caricare automaticamente programmi di uso comune quali `xclock` (l'orologio). Attenzione a non esagerare, caricando programmi di dimensioni consistenti e/o non sempre necessari! In tali casi, per evitare di sprecare inutilmente risorse che si stanno condividendo con gli altri utenti del Centro di Calcolo, è preferibile attivarli esplicitamente da una qualsiasi finestra `xterm` solo quando necessario.

Il window manager legge, se presente, un apposito file di configurazione (`~/.fvwmrc` per `fvwm`). In esso è possibile stabilire i caratteri e i colori da usare per menu e bordi delle finestre, le voci dei menu relative alle finestre o alla finestra di sfondo (il desktop, o nella terminologia X *root window*).

Window manager più recenti (Kde e Gnome) hanno una serie di file e directory di configurazione e, data la complessità delle configurazioni, in questa guida non approfondiremo ulteriormente l'argomento.

Gran parte dei programmi X consente di configurare sia il proprio aspetto che il comportamento mediante il file `/.Xdefaults`, il quale è composto da righe contenenti una coppia variabile-contenuto.

## Shell e programmi sotto X

Anche sotto X Window è possibile aprire una shell e lavorare come descritto in precedenza in modo testuale. In aggiunta vi sono molti programmi che semplificano molti compiti mediante l'uso dell'interfaccia grafica. Essendoci però molti programmi disponibili, sulle diverse installazioni, noi descriveremo solo uno di questi, forse il più importante e usato, in quanto ingloba un *browser* Web, un lettore di posta elettronica e di newsgroup.

## Browser web

Su tutte le macchine del cdc sono disponibili diversi browser web, i quali richiedono l'impostazione di alcuni parametri per effettuare correttamente la navigazione. I browser più comuni sono Mozilla, Konqueror e Galeon.

La configurazione richiesta da tutti i browser è l'impostazione dell'indirizzo del server proxy del cdc, il quale permette la navigazione anche all'esterno del cdc.

I browser più comuni permettono l'impostazione del server proxy tramite la scheda delle preferenze del browser usato. da qui vanno settati i campi riferiti a protocolli HTTP ed FTP con l'indirizzo

```
proxy.cli.di.unipi.it:8080
```

Se è presente un campo in cui specificare la porta da utilizzare, esso va settato con il numero 8080.

Per permettere la navigazione all'interno del cdc senza l'uso del proxy, va settato il campo "*non usare proxy per*" con la stringa

```
cli.di.unipi.it
```

Una volta effettuata questa configurazione è possibile usare il browser per visualizzare i documenti HTML presenti su Internet semplicemente digitando l'indirizzo nella casella Location presente sulla finestra principale.

## Servizio Masterizzazione

Un servizio attivato dal Gruppo Beatrice è il Servizio Masterizzazione, per venire incontro alle esigenze degli studenti che

devono portare a casa grandi moli di dati e dei docenti che vogliono distribuire software e documentazione.  
Per maggiori informazioni in merito, visita la pagina:

<http://beatrice.cli.di.unipi.it/>

## Limitazione Internet

A causa della maleducazione di alcuni studenti che usavano il Centro di Calcolo per fini non didattici, impedendo a chi doveva realizzare tesi e progetti di lavorare, è stato deciso di limitare l'accesso a internet, bloccando l'accesso a determinati siti. Questa soluzione ovviamente crea dei problemi a chi usa Internet a scopo didattico, ma si è resa necessaria per risolvere il problema della carenza di terminali disponibili.

## Help

L'ultima parte di questa breve guida riguarda quella che forse è una delle componenti più importanti per l'uso del Centro di Calcolo: il sistema di *help*, ovvero a chi rivolgere le richieste di aiuto.

Per consentire una gestione più efficiente del Centro di Calcolo il servizio di help è stato strutturato in questo modo:

<a href="mailto:bug@cli.di.unipi.it">bug@cli.di.unipi.it</a>	segnalazioni relative a malfunzionamenti software o hardware e per richieste di informazioni <b>per i servizi esistenti</b> ed è gestito direttamente dai sistemisti
<a href="mailto:ga@cli.di.unipi.it">ga@cli.di.unipi.it</a>	gestione degli account e l'attivazione dei laboratori
<a href="mailto:backup@cli.di.unipi.it">backup@cli.di.unipi.it</a>	richieste per il ripristino di file erroneamente cancellati (da usare solo per file particolarmente importanti!)
<a href="mailto:wwwadm@cli.di.unipi.it">wwwadm@cli.di.unipi.it</a>	assistenza per il servizio web
<a href="mailto:cdcdire@cli.di.unipi.it">cdcdire@cli.di.unipi.it</a>	indirizzo del direttore del Centro di Calcolo
<a href="mailto:abuse@cli.di.unipi.it">abuse@cli.di.unipi.it</a>	segnalazioni di usi impropri delle risorse (per esempio la gente che gioca!)

I primi quattro servizi sono gestiti dai sistemisti e quindi la risposta è **garantita**. Può però anche capitare che una mail non riceva risposta perché cancellata per errore, o dimenticata: è buona norma quindi, nel caso non si riceva risposta entro un paio di giorni, di inviare nuovamente il messaggio.

Le mail indirizzate al supporto di aiuto devono essere scritte in maniera semplice cercando di indicare quanti più particolari possibili, specificando sempre i seguenti punti

- Macchina su cui si è verificato il problema
- Sistema operativo in uso
- I messaggi d'errore visualizzati
- Il malfunzionamento riscontrato
- Qualsiasi altra informazione utile alla diagnosi e risoluzione del problema
- Come riprodurre l'errore

I membri del Gruppo Beatrice sono sempre disponibili per informazioni e/o chiarimenti. Siamo raggiungibili su web all'indirizzo:

<http://www.cli.di.unipi.it/Beatrice>

per Email all'indirizzo:

[g\\_beatrice@cli.di.unipi.it](mailto:g_beatrice@cli.di.unipi.it)

oppure potete venire a trovarci nella nostra sede, indicata nella cartina che trovate nelle prime pagine.

## Appendice A

Linux è un sistema operativo Unix-like (molto simile a Unix) che supporta varie piattaforme (Intel, Amiga, Atari, Sun Sparc, Digital Alpha, Apple Macintosh, PowerPc, MIPS) distribuito gratuitamente e sviluppato grazie al contributo di sviluppatori e utilizzatori sparsi nel mondo intero. Linus Torvalds lo definisce in questo modo: "Linux has evolved from a hacker's toy to a serious competitor to other operating systems" ma andiamo con calma e iniziamo a parlare del modello su cui si basa: Unix.

### **Unix: il Sistema Operativo**

Il sistema UNIX sviluppato dalla AT&T Bell Laboratories, è nato essenzialmente come reazione ai sistemi operativi di grandi dimensioni, complessi e poco flessibili disponibili alla fine degli anni Sessanta. Il principale obiettivo di progettazione è stato quello di ottenere un sistema operativo d'elevata potenzialità, definendo comandi che non cerchino di risolvere tutti i problemi possibili. Viceversa, un aspetto fondamentale tipico delle tendenze meno recenti riguarda, infatti, la necessità di disporre di comandi che eseguono una singola azione, ma con semplicità e nel modo migliore. Originariamente (anni '70) il sistema operativo fu sviluppato da un gruppo di esperti di computer come uno strumento di sviluppo per uso personale, ignorando così le esigenze dei principianti a vantaggi della velocità e della precisione.

A causa delle leggi antitrust il software non poteva essere venduto e prese la strada delle Università Americane che a loro volta iniziarono sviluppare varie versioni di Unix facendolo maturare velocemente ma creando un po' di disordine e confusione.

In questo modo iniziarono a formarsi una casta di persone che una volta laureate e pronte per il mondo del lavoro iniziarono a divulgare e far conoscere Unix nelle aziende. Pian piano Unix poté diventare un prodotto commerciale e si iniziarono a vedere sul mercato varie versioni proprietarie e quindi a pagamento. Oltre alle versioni commerciali esistono anche versioni "free" (troverete ulteriori spiegazione del concetto di software libero più avanti), la più importante è quella che prende il nome di Linux.

### **Il fratello minore, ma solo cronologicamente ("a better Minix, than Minix")**

Linux è nato come estensione del Minix (sistema operativo Unix per processori i86 sviluppato ad uso didattico dal professor Andrew S. Tanenbaum) e come clone di Unix, dallo studente finlandese dell'Università di Helsinki **Linus Benedict Torvalds**, per un progetto universitario sullo studio delle funzionalità multitasking dei microprocessori i386. Per essere più precisi, Linux nacque, anche se non esiste una data esatta perché ancora non si poteva parlare di un vero sistema operativo, il **5 ottobre 1991** quando Torvalds postò sul news-group comp.os.minix il seguente messaggio:

*``Hello netlanders. Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.*

*As I mentioned a month ago, I'm working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02...but I've successfully run bash, gcc, gnu-make, gnu-sed, compress, etc. under it."*

Il lavoro procedette a buoni ritmi e Linux iniziò a prendere la forma di un vero sistema operativo, ma Linus aveva bisogno di una mano per finirlo e grazie all'aiuto di moltissimi ed eterogenei gruppi di sviluppatori Unix ed esperti di Sistemi Operativi che misero a disposizione, in modo gratuito, il codice da loro prodotto, poté ben presto annunciare una versione del kernel stabile: la 1.00 (14 marzo 1994).

Il cuore di Linux è POSIX con estensioni BSD e SYSV (cioè simile a Unix ma con codice sorgente diverso). Esso presenta tutte le caratteristiche di un moderno sistema operativo e cioè al suo interno sono state sviluppate funzionalità come multitasking, multi-threading, multi-user, multipiattaforma, shared libraries, memoria virtuale ed eseguibili copy-on-write con un memory management adeguato. Supporta completamente il TCP/IP (che in quest'ultimo periodo è in fase di revisione per adeguarlo ai nuovi standard IPv6). A tutto questo si deve aggiungere un X Windows System (X-Free86) che rappresenta il porting di una parte della versione ufficiale X11 R6.3 distribuita da X Consortium Inc. compatibile a pieno con questo standard.

Oggi Linux è il sistema operativo tecnicamente più evoluto dell'intera categoria, conta milioni di utenti sparpagliati in tutto il mondo e grazie alla presenza del codice sorgente è il punto di riferimento per lo sviluppo del kernel. E' utilizzato sia dai Guru dell'informatica che da utenti che hanno bisogno di un sistema robusto ma allo stesso tempo molto snello, sia in ambito universitario e sia in grandi aziende che da Internet Service Provider e rappresenta, in ultima analisi, una possibile minaccia per altri sistemi operativi commerciali. L'implementazione più particolare che ho visto, è stata la gestione di un distributore di benzina attraverso un complesso sistema gestito da Linux.

Sono disponibili diverse distribuzioni di Linux (S.u.S.E., Debian, RedHat, Slackware) che contengono il sistema operativo completo delle procedure di installazione e configurazione, dell'interfaccia grafica X-11 e di moltissime applicazioni di svariato genere.

Infine, dopo sette anni dalla nascita di Linux, Linus Torvalds si è laureato e dopo aver collaborato per diversi anni come ricercatore all'interno dell'Università di Helsinki si è trasferito in America dove continua a dirigere lo sviluppo del sistema operativo e a lavorare su una versione Real-Time di Linux.

## **Software Libero**

Nel campo informatico, e più specificatamente nel settore del software, si è spesso sentito il bisogno e la necessità di libertà ma per raggiungere un tale obiettivo occorre lavorare sul piano legale.

Nel 1985 Richard Stallman, fondò la F.S.F. (Free Software Foundation) con il preciso fine di creare e diffondere la filosofia del free-software e cioè il concetto di poter distribuire e modificare il software secondo le proprie esigenze e ridistribuire anche le modifiche apportate.

Nacque, così la GPL (General Public License) in modo da proteggere il software libero da coloro che avrebbero voluto renderlo commerciale e il copyright che proteggeva il free-software prese il nome di **copyleft**. Il software in questo modo protetto non è quindi né shareware e né di pubblico dominio.

Stallman si volle prefiggere come obiettivo la realizzazione un sistema operativo completo, magari realizzato con l'aiuto di volontari. In questo modo prese vita il progetto chiama **GNU: Gnu is Not Unix** con il quale dopo la realizzazione di un compilatore C si iniziò a creare una serie di applicazioni necessarie nel momento in cui il cuore del sistema operativo fosse stato ultimato.

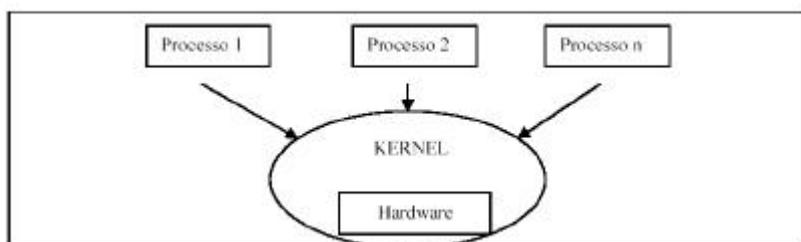
A causa di possibili confusioni, in inglese *free-software* può significare anche *free of charge* (gratuito), nel 1998 il team di sviluppo della Debian ha reso pubblica la definizione di *Open Source* che elimina qualsiasi problema di ambiguità.

Grazie alla cooperazione di svariate persone e della licenza GPL, Linux è divenuto un qualcosa di interessante non soltanto in ambito didattico ma a non pubblico ben più vasto.

## **La struttura di Linux: il KERNEL**

Il sistema operativo è costituito da un sistema di base e da un certo numero di estensioni. Il sistema di base comprende il kernel, le shell e le utility di base. Le estensioni sono ad esempio le funzionalità grafiche, le librerie, i compilatori.

Il kernel di Linux, che consiste nell'interfaccia tra l'hardware e i processi, gestisce il tempo di Cpu (scheduling) e la memoria e si occupa del controllo e della gestione di ciascun device e di tante altre cose. Rappresenta il nocciolo dell'intero sistema operativo ed idealmente è una sorta di astrazione della macchina fisica ed è il livello a cui i processi si rivolgono per qualsiasi operazione.



E' un kernel monolitico molto complesso e altamente configurabile. Dalla versione 2.0.0 è possibile suddividere il kernel in moduli e uno speciale demone (kerneld) permette di filtrare tutte le chiamate ai moduli e di caricarli in tempo di esecuzione nel momento in cui vengono richiesti. Per quanto riguarda la versione del kernel bisogna ricordare che poiché Linux segue il modello a sviluppo aperto e quindi che tutte le nuove release sono rilasciate al pubblico anche se non rappresentano "un prodotto di qualità". Per le release del kernel è utilizzato un specifico schema di rilascio: le versioni V.x.y con x pari rappresentano le versioni stabili mentre al contrario le versioni con x dispari sono le beta release utilizzate normalmente dagli sviluppatori e possono essere instabili portando in crash l'intero sistema; come vengono risolti bug viene incrementato y.

### **La Shell**

La shell rappresenta l'interfaccia tra macchina e utente in modo che quest'ultimo possa accedere ai servizi resi disponibili dal kernel e come indica il nome (shell in anglosassone vuol significare *conchiglia*) protegge il kernel da eventuali gravi errori che l'utente potrebbe commettere gestendo direttamente le risorse del nucleo. Occorre capire che la shell non è un semplice programma ma è un vero e proprio ambiente di sviluppo dotato di un suo linguaggio di programmazione dove le possibilità applicative sono infinite, anche la sua forma può essere della più disparata: attraverso riga di comando, forma più semplice e antica, ma anche in veste di sistema grafico o di riconoscimento di comandi in forma vocale (forma evoluta).

## Appendice B

### Tips & Tricks

#### **LOGON**

Per accedere ai laboratori Linux è necessario frequentare un laboratorio. La risorsa viene assegnata dal docente titolare e comunicata al responsabile della gestione account che provvede ad abilitare l'utente all'uso delle risorse per le quali è necessario.

I motivi per cui Linux può rifiutare l'accesso sono:

La mancanza della risorsa LABORATORIO: in questo caso rivolgersi al proprio docente, o al personale preposto per chiarire la propria posizione;

La tastiera ha il CAPS LOCK attivo: l'autenticazione di Linux è case sensitive sulla password;

L'account può essere stato sospeso: rivolgersi al personale preposto.

Se l'accesso è rifiutato solo in modalità grafica la causa può essere un superamento di quota assegnata: ripulire la propria home

Prima di rivolgersi al personale, è comunque opportuno verificare di non trovarsi in uno degli altri casi.

#### **CAMBIO PASSWORD**

Il cambio password si effettua tramite il comando ssh alice digitato da una shell su una qualsiasi macchina del CdC. Il comando effettua un collegamento con l'host alice. Viene richiesto il logon. A questo punto digitare la password attualmente in uso (Old password:), digitare la nuova password (New password:), digitare nuovamente la nuova password (Re-type new password:).

Il collegamento ad alice si chiude e la nuova password sarà attiva dalla mattina successiva, poichè l'aggiornamento viene effettuato durante la notte.

#### **SINCRONIZZAZIONE LINUX/WINDOWS**

La sincronizzazione delle password tra Linux e Windows viene effettuata durante la notte in automatico. Se non si riesce ad accedere a una macchina Windows pur avendo accesso ad un laboratorio effettuare la procedura di cambio password.

È possibile utilizzare nuovamente la propria password come "New password". In questo modo l'accesso a Windows si sincronizza con la password usata con Linux e viene attivato a partire dalla mattina successiva.

Se la mattina successiva non viene ancora consentito l'accesso su Windows contattare gli operatori addetti.

#### **LE STAMPANTI**

Le stampanti utilizzabili nel CdC sono di due tipi: stampante ad aghi e laser. Per stampare sulla stampante ad aghi digitare il comando 'lpr nome\_file'. Per stampare sulla stampante laser digitare il comando 'lj nome\_file'. Attenzione: il comando 'lj nome\_file' è un alias. Se non dovesse funzionare digitare 'lpr -Plj0 nome\_file'.

La quota stampe che viene assegnata insieme alla quota laboratorio tiene conto esclusivamente delle stampe mandate sulla stampante laser.

È necessario fare attenzione a che cosa viene mandato in stampa, in quanto non tutti i formati sono stampabili sotto Linux e non tutti i formati sono stampabili sulla stampante ad aghi.

#### FORMATI STAMPABILI SOTTO LINUX:

- Tutti i formati visualizzabili da Linux stesso (.ps, .txt, .html, .java, .c, ecc.. ecc..).
- Non mandare in stampa file tipicamente Windows (.doc, .xls ecc.. ecc..).
- Per stampare questi formati è necessario usare il sistema operativo Windows.

#### FORMATI STAMPABILI SULLA STAMPANTE AD AGHI:

- Tutti i formati di testo (es: txt), di codice (java, C e così via).
- Non mandare sulla stampante ad aghi file di tipo postscript (.ps) o PDF.

#### FORMATI STAMPABILI SULLA STAMPANTE LASER:

- Tutti i formati che si possono stampare sulla stampante ad aghi e in più i file postscript.

#### NOTA FILE PDF

I file .pdf si possono stampare in due modi diversi:

da acroread: in questo caso si manda in stampa il file con estensione .pdf, mettendo come stampante la lj;

da shell: in questo caso il file va prima trasformato in .ps. Questo si può fare o facendo la stampa su file da acroread (nomefile.ps) o usando il comando 'pdf2ps -o nomefile.ps nomefile.pdf'. Il file da mandare in stampa è il file .ps

## **GESTIONE PROFILO**

Come utente linux si possono utilizzare sia i vari ambienti grafici delle RedHat (KDE e GNOME) che impostazioni personali. Nella propria home ogni utente ha i file

.cshrc  
.fvwmrc  
.xinitrc

che sono personalizzabili.

Per ripristinare i file di default (cioè quelli assegnati con la creazione dell'account) e' necessario eseguire il comando /share/lib/profile/setprofile.

Durante l'esecuzione del comando viene chiesto se si vuole conservare una copia di backup dei propri file (.cshrc, .xinitrc, .fvwmrc). Se si richiede verranno salvati in una directory .backup.

## Appendice C

### ***Acceptable Use Policy della rete GARR***

1. La Rete Italiana dell'Università e della Ricerca Scientifica, denominata comunemente "la rete del GARR", si fonda su progetti di collaborazione scientifica ed accademica tra le Università e gli Enti di Ricerca pubblici italiani. Di conseguenza il servizio di rete GARR è destinato principalmente alla comunità che afferisce al Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST). Esiste tuttavia la possibilità di estensione del servizio stesso anche ad altre realtà che svolgono attività di ricerca in Italia, specialmente ma non esclusivamente in caso di organismi "no-profit" impegnati in collaborazioni con la comunità afferente al MURST. L'utilizzo della rete è comunque soggetto al rispetto delle Acceptable Use Policy (AUP) da parte di tutti gli utenti GARR.
2. Il "Servizio di rete GARR", definito brevemente in seguito come "Rete GARR", è costituito dall'insieme dei servizi di collegamento telematico, dei servizi di gestione della rete, dei servizi applicativi e di tutti quelli strumenti di interoperabilità (operati direttamente o per conto del GARR) che permettono ai soggetti autorizzati ad accedere alla rete di comunicare tra di loro (rete GARR nazionale).

Costituiscono parte integrante della rete GARR anche i collegamenti e servizi telematici che permettono la interconnessione tra la rete GARR nazionale e le altre reti.

3. Sulla rete GARR non sono ammesse le seguenti attività:
  - fornire a soggetti non autorizzati all'accesso alla rete GARR il servizio di connettività di rete o altri servizi che la includono, quali la fornitura di servizi di housing, di hosting e simili, nonché permettere il transito di dati e/o informazioni sulla rete GARR tra due soggetti entrambi non autorizzati all'accesso sulla rete GARR (third party routing);
  - utilizzare servizi o risorse di rete, collegare apparecchiature o servizi o software alla rete, diffondere virus, hoaxes o altri programmi in un modo che danneggi, molesti o perturbi le attività di altre persone, utenti o i servizi disponibili sulla rete GARR e su quelle ad essa collegate;
  - creare o trasmettere (se non per scopi di ricerca o comunque propriamente in modo controllato e legale) qualunque immagine, dato o altro materiale offensivo, diffamatorio, osceno, indecente, o che attenti alla dignità umana, specialmente se riguarda il sesso, la razza o il credo;
  - trasmettere materiale commerciale e/o pubblicitario non richiesto ("spamming"), nonché permettere che le proprie risorse siano utilizzate da terzi per questa attività;
  - danneggiare, distruggere, cercare di accedere senza autorizzazione ai dati o violare la riservatezza di altri utenti, compresa l'intercettazione o la diffusione di parole di accesso (password) e chiavi crittografiche riservate;
  - svolgere sulla rete GARR ogni altra attività vietata dalla Legge dello Stato, dalla normativa Internazionale, nonché dai regolamenti e dalle consuetudini ("Netiquette") di utilizzo delle reti e dei servizi di rete acceduti.
4. La responsabilità del contenuto dei materiali prodotti e diffusi attraverso la rete è delle persone che li producono e diffondono.
5. I soggetti autorizzati (S.A.) all'accesso alla rete GARR, definiti nel documento "Regole approvate dalla CRCS", possono utilizzare la rete per tutte le proprie attività istituzionali. Si intendono come attività istituzionali tutte

quelle inerenti allo svolgimento dei compiti previsti dallo statuto di un soggetto autorizzato, comprese le attività all'interno di convenzioni o accordi approvati dai rispettivi organi competenti, purchè l'utilizzo sia a fini istituzionali. Rientrano in particolare nelle attività istituzionali, la attività di ricerca, la didattica, le funzioni amministrative dei soggetti e tra i soggetti autorizzati all'accesso e le attività di ricerca per conto terzi, con esclusione di tutti i casi esplicitamente non ammessi dal presente documento.

Altri soggetti, autorizzati ad un accesso temporaneo alla rete (S.A.T.) potranno svolgere solo l'insieme delle attività indicate nell'autorizzazione.

Il giudizio finale sulla ammissibilità di una attività sulla rete GARR resta prerogativa degli Organismi Direttivi del GARR.

6. Tutti gli utenti a cui vengono forniti accessi alla rete GARR devono essere riconosciuti ed identificabili. Devono perciò essere attuate tutte le misure che impediscano l'accesso a utenti non identificati. Di norma gli utenti devono essere dipendenti del soggetto autorizzato, anche temporaneamente, all'accesso alla rete GARR.

Per quanto riguarda i soggetti autorizzati all'accesso alla rete GARR (S.A.) gli utenti possono essere anche persone temporaneamente autorizzati da questi in virtù di un rapporto di lavoro a fini istituzionali. Sono utenti ammessi gli studenti regolarmente iscritti ad un corso presso un soggetto autorizzato con accesso alla rete GARR.

7. E' responsabilità dei soggetti autorizzati all'accesso, anche temporaneo, alla rete GARR di adottare tutte le azioni ragionevoli per assicurare la conformità delle proprie norme con quelle qui espone e per assicurare che non avvengano utilizzi non ammessi della rete GARR. Ogni soggetto con accesso alla rete GARR deve inoltre portare a conoscenza dei propri utenti (con i mezzi che riterrà opportuni) le norme contenute in questo documento.
8. I soggetti autorizzati all'accesso, anche temporaneo, alla rete GARR accettano esplicitamente che i loro nominativi (nome dell'Ente, Ragione Sociale o equivalente) vengano inseriti in un annuario elettronico mantenuto a cura degli Organismi Direttivi GARR.
9. In caso di accertata inosservanza di queste norme di utilizzo della rete, gli Organismi Direttivi GARR prenderanno le opportune misure, necessarie al ripristino del corretto funzionamento della rete, compresa la sospensione temporanea o definitiva dell'accesso alla rete GARR stessa.

L'accesso alla rete GARR è condizionato all'accettazione integrale delle norme contenute in questo documento.