



CDC & Linux **versione 2006**

introduzione all'utilizzo delle risorse del centro di calcolo

Università degli studi di Pisa
Dipartimento di Informatica
Gruppo Beatrice



Gruppo Beatrice

Questo documento è stato realizzato dal Gruppo Beatrice, gruppo di studenti di informatica dell'Università di Pisa, volontari, il cui scopo è fornire supporto alla didattica e servizi per tutti gli studenti

La scrittura ed aggiornamento dei contenuti della dispensa è stata curata negli anni da:

- Giuseppe Augiero
- Gianni Giaccherini
- Leonardo Fiorini
- Nicola Petruzzellis
- Vittorio Romano

L'attuale versione (2006) è curata da Leonardo Fiorini (fiorini@cli.di.unipi.it). La versione più aggiornata è consultabile presso il sito del Gruppo beatrice, all'indirizzo <http://beatrice.cli.di.unipi.it>.

Per critiche, suggerimenti, segnalazioni di errori contenuti in questo documento è possibile scrivere all'indirizzo g_beatrice@cli.di.unipi.it.

Quest'opera è rilasciata sotto la licenza Creative Commons Attribuzione - Non commerciale. Una copia della licenza è consultabile presso il sito web di Creative Commons, all'indirizzo <http://creativecommons.org/licenses/publicdomain/> o spedendo una lettera a Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Macintosh® è un marchio registrato della Apple Computer Inc.

Microsoft®, Visual Studio®, Windows®, Microsoft Word®, sono marchi registrati della Microsoft Corporation.

KDE™ e K Desktop Environment™ sono marchi della KDE e.V.

Linux® è un marchio registrato di Linus Torvalds e del Linux Mark Institute.

UNIX® è un marchio registrato di The Open Group.

Tutti gli altri marchi citati nel presente documento appartengono ai rispettivi proprietari

Indice

PARTE PRIMA: IL CENTRO DI CALCOLO (CDC).....	5
Introduzione.....	5
Il centro di calcolo.....	5
Le risorse utilizzabili.....	6
Stampanti del centro di calcolo.....	6
Laboratorio portatili.....	7
Account per l'accesso alle risorse.....	7
Sessione di lavoro.....	8
Logon: collegarsi.....	8
Logoff: scollegarsi.....	9
Dove sono i miei dati?.....	9
Regolamento del CDC.....	10
Sempre aggiornati!.....	10
PARTE SECONDA: SISTEMI UNIX E LINUX.....	11
Distribuzioni.....	11
La gestione dei dati in ambiente Unix.....	12
File.....	12
Filesystem.....	12
Percorsi di file e abbreviazioni.....	14
La gestione degli utenti in Unix.....	15
Utenti, gruppi e diritti di accesso alle risorse.....	16
Lavorare con Linux tramite la shell.....	17
Il prompt.....	17
Comandi, parametri ed opzioni.....	18
Gestione dei dati tramite la shell.....	18
Elencare i file del sistema.....	18
Navigazione nel filesystem.....	19
Creazione ed eliminazione di directory.....	19
Eliminazione di file.....	20
Spostamento e rinomina di file o directory.....	20
Copia di file o directory.....	21
Ricerca nel filesystem.....	21
Modifica dei permessi di un elemento.....	21
Esaminare un file.....	22
Ricerca del contenuto di un file.....	23
Spazio su disco.....	24
Altri comandi di base.....	24
Alias.....	25
Abbreviazioni di percorsi.....	26
Metacaratteri.....	26
Mount di dispositivi rimovibili.....	27
Input, Output e redirectione.....	28
Documentazione.....	30
Linux how-to.....	31
Interfaccia grafica: il sistema X Window.....	31
Comandi e funzioni generiche dell'X-server.....	33
Window manager ed ambienti desktop.....	33
K Desktop Environment - KDE.....	34
GNU's Network Object Model Environment - GNOME.....	35
Programmi comuni per il lavoro in Unix.....	36
GNU Emacs.....	36
Vi e Vim.....	37
Nano/Pico.....	38
Programmi di compressione dati.....	38

Uso dei floppy disk con gli Mtools.....	39
Comandi di stampa.....	39
Processi.....	40
PARTE TERZA: LA RETE INTERNET.....	42
Servizi di rete al CDC e limitazioni.....	42
Servizi di connessione remota.....	42
Shell sicura SSH.....	42
Trasferimento file con SCP.....	43
Trasferimento file con FTP.....	44
Navigazione del web.....	44
Posta elettronica.....	46
Programmi di lettura email.....	46
Utilizzo di Pine.....	47
Composizione ed invio di email.....	47
Gestione della posta in arrivo.....	48
Creare una rubrica indirizzi.....	49
Utilizzo del servizio Web Mail.....	49
Configurazione di altri programmi di posta.....	49
PARTE QUARTA: AIUTO ED INDIRIZZI UTILI.....	50
APPENDICE A: ACCEPTABLE USER POLICY RETE GARR.....	51

Parte prima: il centro di calcolo (CDC)

Introduzione

Questa guida si propone come una introduzione ai sistemi del centro di calcolo del Dipartimento di Informatica dell'Università di Pisa. In essa sono contenute le nozioni di base necessarie per iniziare da subito a lavorare utilizzando le risorse che il Dipartimento di Informatica mette a disposizione.

Anche se hai già una buona conoscenza di base dell'informatica, ti consigliamo di leggere questo documento prima di metterti al lavoro; dopodiché potrai iniziare a sperimentare quanto hai letto presso uno dei computer del centro di calcolo. È possibile che tu commetta degli errori, ma questi si limiteranno al massimo a danneggiare solo i tuoi dati, senza conseguenze nefaste per altri; ovviamente un po' di cautela è sempre consigliata.

Durante la lettura, specie all'inizio, potresti incontrare alcuni termini tecnici di cui non conosci il significato, ma che verranno via via riproposti e spiegati. Nella prima parte del documento verrà presentato il centro di calcolo, descrivendoti le diverse apparecchiature che è possibile usare. La seconda parte è invece dedicata a Unix e Linux, il sistema operativo utilizzato principalmente nei laboratori; per favorire la comprensione ti verranno forniti alcuni confronti con Windows, un sistema operativo che dovresti già conoscere.

Speriamo che questo documento possa rispondere a buona parte delle domande che riguardano l'argomento. Se così non fosse, puoi sempre richiedere maggiori informazioni mandando una email all'indirizzo g_beatrice@cli.di.unipi.it. Buon lavoro!

Il centro di calcolo

Il centro di calcolo del Dipartimento di Informatica si trova in largo Bruno Pontecorvo 3, nell'edificio B del polo Fibonacci; l'area è accessibile da via Buonarroti, da largo Pontecorvo e da via Emanuele Filiberto. All'interno della struttura, si trovano cinque laboratori attrezzati, un laboratorio per l'utilizzo di portatili e tre aule didattiche, tutti accessibili dagli studenti. Nello stesso stabile si trovano i locali di servizio, ovvero la sala macchine, la sala sistemisti e quella degli operatori.

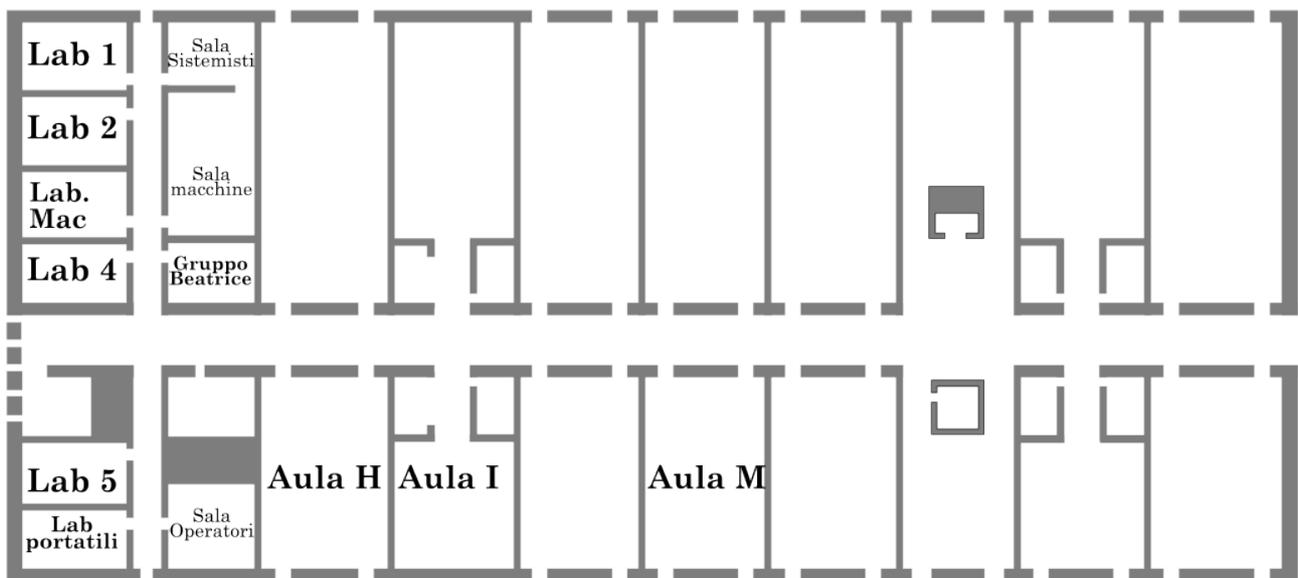


Figura 1: mappa del Centro di Calcolo

Il centro di calcolo è aperto dal lunedì al venerdì dalle 8,30 alle 19,00. In alcuni periodi dell'anno l'apertura viene prolungata fino alle 23,00 dal lunedì al giovedì. L'uso delle macchine è consentito agli studenti iscritti al corso di laurea o diploma in informatica ed agli studenti di altri corsi di laurea che debbano seguire almeno un corso di informatica. L'accesso ai laboratori ed alle aule è

consentito sempre ed indistintamente, salvo nei casi in cui questi siano occupati per lo svolgimento di lezioni, esami o quant'altro.

È importante ricordare che durante la permanenza nei laboratori possono essere effettuati controlli sulla identità delle persone; è quindi necessario avere con se il libretto universitario o un certificato di iscrizione.

All'interno del centro sono presenti alcune figure di riferimento, responsabili per quanto riguarda la gestione e la manutenzione della struttura, quali i sistemisti e gli operatori, che si occupano prevalentemente della gestione tecnica della macchine, ed il personale ausiliario di servizio.

Le risorse utilizzabili

Le macchine a disposizione degli studenti si dividono in due categorie:

- *personal computer/macintosh*, ovvero le macchine IBM compatibili e Mini Mac presenti nei laboratori e nelle aule ed accessibili fisicamente; costituiscono le stazioni di lavoro principali,
- *server*, cioè i computer situati nella sala macchine e non accessibili, se non da remoto; tali server permettono l'accesso a studenti e docenti alle risorse del centro, e mantengono attivi alcuni servizi particolari.

Tutti i computer sono connessi tra loro (oltre che ad Internet) tramite una *rete* interna al centro che permette la condivisione di dati e risorse nel modo più efficiente possibile.

I personal computer sono equipaggiati con sistemi operativi Windows XP e/o Debian GNU/Linux Etch/Sarge, oltre a contenere una vasta gamma di prodotti software per lo sviluppo (GCC, Eclipse, Visual Studio), suite per l'ufficio (OpenOffice, Microsoft Office), programmi per la navigazione in internet (Internet Explorer, Mozilla/Firefox) e molti altri applicativi. Le macchine presenti nelle aule H, I e M hanno una configurazione hardware simile, al contrario di altre situate nei laboratori; poiché non tutti i calcolatori offrono la stessa potenza, non è sempre possibile eseguire gli applicativi più pesanti, come ad esempio alcuni ambienti di sviluppo¹. I Mini Mac sono equipaggiati con il sistema operativo Mac OSX ed il corredo software standard delle macchine, oltre al set di programmi di sviluppo.

I server del centro di calcolo sono macchine dedicate a specifiche applicazioni e che permettono l'accesso alle risorse del CDC. Alcuni server mantengono i dati appartenenti ad ogni utente, altri permettono la navigazione attraverso la rete Internet, ed altri ancora sono utilizzabili anche dagli studenti per lavorare su specifiche applicazioni. Il loro ruolo è fondamentale per il centro di calcolo, ed è per questo che essi sono posti all'interno della sala macchine, il cui ingresso è riservato al personale tecnico. L'accesso a tali macchine è permesso solo tramite connessione remota, ovvero accedendo ad essi tramite un'altra postazione, presumibilmente una macchina dei laboratori. I server per usi generici, accessibili dagli studenti tramite connessione sicura *ssh*², sono due:

- **olivia**: AMD Athlon(tm) XP 2600+, Red Hat Linux 9, kernel 2.4.20-24.9,
- **trudy**: Intel(R) Pentium(R) 4 CPU 2.80GHz, Debian GNU/Linux sarge, kernel 2.6.10-1-i686.

Altri server, non accessibili direttamente dagli studenti, mantengono attivi servizi generali quali, ad esempio, il *web server*, il *mailserver*, il *proxy* ed altri.

Stampanti del centro di calcolo

Il CDC mette a disposizione degli utenti una serie di stampanti comuni, utilizzabili per scopi didattici. La **stampante laser** è utilizzabile per qualsiasi documento; il numero di pagine stampabili è limitato ad una determinata quantità per ogni utente, aggiornata ogni semestre; se alla fine di un

1 In generale, la maggioranza dei computer è abbastanza potente da eseguire senza problemi tutti i programmi necessari alla didattica.

2 Si veda nei capitoli successivi quali sono i metodi per la connessione a macchine remote.

semestre è rimasto un quantitativo di stampe inutilizzato, esso viene aggiunto al totale delle pagine stampabili dall'utente.

Come i server, le stampanti non sono direttamente accessibili; le stampe inviate dai computer del CDC vengono consegnate ad orari stabiliti presso una rastrelliera situata davanti ai laboratori. Per qualsiasi informazione riguardante le stampanti del CDC è sufficiente recarsi nella sezione apposita del sito del centro di calcolo, all'indirizzo <http://www.cli.di.unipi.it/stampe.php>.

Laboratorio portatili

Gli utenti del CDC hanno la possibilità, qualora sia necessario, di utilizzare il proprio computer portatile presso il laboratorio specifico. Per poter fare ciò, è necessario recarsi dai sistemisti per effettuare la registrazione del proprio laptop, dopodiché sarà possibile connetterlo alla rete del centro.

Sebbene le macchine del CDC siano controllate tramite antivirus e la connessione ad Internet sia filtrata, è opportuno verificare lo stato della propria macchina prima e dopo il collegamento con la rete interna, al fine di evitare spiacevoli sorprese, osservando almeno queste semplici regole:

- effettuare una scansione del proprio sistema con un antivirus aggiornato,
- mantenere attivo il proprio antivirus,
- se si hanno cartelle condivise in rete, proteggerle mediante password o, meglio ancora, eliminare la condivisione,
- controllare che non siano attivi sulla propria macchina servizi che possano permettere accessi non voluti da altri utenti (come telnet, web server o altro).

È **severamente vietato** recarsi in un laboratorio diverso da quello portatili, scollegare dalla rete uno dei computer fissi ed usare la postazione come punto di connessione per il proprio laptop. Il mancato rispetto di questa regola può comportare la sospensione dall'utilizzo dei servizi del Centro di Calcolo.

Account per l'accesso alle risorse

Per poter usufruire dei servizi del CDC è necessario effettuare una registrazione per ottenere un *account*. Un account è uno spazio sulle macchine appartenente ad un utente, un'area di lavoro unica alla quale si accede da qualsiasi computer. Tutti i dati e le informazioni di un utente sono inclusi nel suo account, il quale è personale e non utilizzabile da altri.

La richiesta di un account va fatta compilando un apposito modulo, chiamato MAS, situato all'ingresso dei laboratori; tale modulo va poi imbucato nella cassetta apposita. Dopo circa una settimana l'account è pronto e può essere ritirato presso la stanza degli operatori agli orari fissati sulla porta, presentandosi con il libretto universitario o un certificato di iscrizione. Al momento della consegna dell'account, viene fatta firmare una dichiarazione di responsabilità che è bene leggere.

Alla consegna dell'account si ricevono un *nome utente* ed una *password* alfanumerica. Il nome utente (detto anche *login name* o *login*) è identificativo di un unico studente. La password ad esso correlata va custodita scrupolosamente e non va comunicata a nessuno, al fine di evitare che qualcuno possa accedere alle macchine del centro servendosi di altri account ed avendo quindi la possibilità di arrecare danni restando anonimo. In casi come questo la responsabilità cade sul proprietario dell'account piuttosto che sull'autore materiale di danneggiamenti. Un uso improprio delle risorse è **penalmente perseguibile**: occorre quindi prestare attenzione a come vengono utilizzate, oltre che evitare l'uso del proprio account da parte di altri.

Il modulo MAS va compilato **solo una volta** nel corso della propria carriera accademica. La consegna del modulo e la relativa creazione dell'account danno diritto ad una configurazione minima per ogni utente:

- 15MB di spazio disco utilizzabile in ambiente Linux,

- 50 pagine utilizzabili per la stampa laser ogni semestre,
- utilizzo dei servizi di rete quali Web, posta elettronica, accesso ftp,
- utilizzo del software presente sulle macchine.

Gli studenti che seguono corsi di laboratorio, oppure devono realizzare un progetto per un corso o per preparare la tesi, hanno diritto ad un insieme aggiuntivo di risorse, che possono essere ottenute fornendo il proprio nome utente al docente responsabile del corso di laboratorio. In questo caso non è necessaria alcuna firma. Ad operazione effettuata lo studente riceve una email di conferma. Le risorse aggiuntive vengono tolte dopo un periodo più o meno breve, dipendente dalle risorse stesse. Ad eccezione delle risorse necessarie per la tesi, solitamente vengono tolte dopo un semestre.

Sessione di lavoro

Login e password si utilizzano per l'accesso al proprio account presso tutte le macchine del centro di calcolo. Generalmente l'account è accessibile da ogni computer situato nei laboratori e nelle aule. Quando nessuno è connesso ad un PC, dopo pochi minuti di attività il suo monitor si spegne per risparmiare energia; è sufficiente muovere il mouse o premere un tasto per riattivarlo.

Logon: collegarsi

Generalmente, una postazione vuota mostrerà una *schermata di login*, in cui vengono richiesti il login name e la password per accedere al sistema. Tipicamente si possono presentare due situazioni differenti: una schermata di *login grafico* (in ambiente Linux, Windows e Mac) oppure una di *login testuale* (in ambiente Linux).

Il **login grafico** si presenta come una finestra con almeno due campi, uno per il login e l'altro per la password. È sufficiente riempire i due campi e premere il tasto **Invio** per accedere al sistema. Se tale finestra non è presente in ambiente Windows, è sufficiente premere la combinazione di tasti **CTRL ALT** e **CANC** per farla apparire.



Immagine 1: login grafico di Linux, Windows e Mac

Se l'autenticazione va a buon fine si entrerà nella *sessione grafica* dell'account, dalla quale interagire tramite l'utilizzo di mouse e tastiera con il computer.

Il **login testuale** si presenta generalmente mostrando semplicemente una scritta bianca su fondo nero, che invita a digitare il proprio nome utente,

```
Debian GNU/Linux testing/unstable
login:
```

dopodiché viene richiesta la propria password

```
Debian GNU/Linux testing/unstable
login: pippo
password:
```

```
pippo@olivia:~$
```

Se i dati immessi sono corretti apparirà il *prompt della shell* di Linux, simile a quella mostrata qui sopra; da qui sarà possibile interagire con il computer impartendo comandi tramite la tastiera. Per motivi di sicurezza, la password non verrà mostrata durante la digitazione; occorre quindi fare attenzione a cosa si scrive.

È importante ricordare che sia il nome utente che la password sono *case sensitive*, ciò significa che il computer, al momento dell'autenticazione, farà distinzione tra lettere maiuscole e minuscole; se pippo è la login assegnata, sarà ben diverso scrivere combinazioni come Pippo, o pIPPO e così via. Similmente tale discorso vale per la password.

Logoff: scollegarsi

Al termine di una sessione di lavoro, occorre scollegarsi, ovvero uscire dal proprio account, liberando così la macchina per altri utenti. È importante chiarire che il *logoff* (o *logout*) non si effettua spegnendo la macchina sul quale si è collegati, in nessun modo. Similmente alla procedura di login, il logoff può essere fatto sia se ci si trova in ambiente grafico che in un ambiente testuale.

In ambiente grafico, sia su Linux che su Windows, è sufficiente selezionare il pulsante di logout situato nel menù principale del sistema. Il pulsante è riconoscibile tra gli altri poiché solitamente situato in fondo al menù, insieme con altri pulsanti dalle funzioni simili, come quello per l'arresto del PC. In ambiente Mac, il logout si effettua selezionando il menu Mela in alto a sinistra e poi l'opzione Log Out.

L'uscita dall'ambiente testuale, ovvero dalla shell di comando, è effettuabile in due modi. Si può uscire digitando il comando `exit`,

```
[pippo]olivia: [~]-> exit
```

```
Debian GNU/Linux testing/unstable
login:
```

che termina l'esecuzione della shell, effettuando un reset della schermata, la quale ritorna nello stato di attesa di login; alternativamente è possibile uscire dalla sessione premendo la combinazione di tasti CTRL e D; il sistema rappresenta in questo modo il carattere di fine di un file.

È bene ricordarsi di non spegnere il computer, né riavviarlo se non si è sicuri che altri utenti non stanno lavorando sulla stessa macchina, collegati ad esempio da remoto; le macchine del CDC sono utilizzate ogni giorno da molti studenti, per lavori e progetti che possono richiedere l'utilizzo di svariati elaboratori anche contemporaneamente.

Dove sono i miei dati?

Nel momento in cui si accede ad un computer del Centro di Calcolo, vengono automaticamente caricati i dati contenuti nello spazio di lavoro dell'utente. Tali dati non risiedono fisicamente su nessuna delle macchine dei laboratori, ma vengono richiamate da un medesimo server centrale, detto *homeserver*, nel quale sono contenuti tutti i dati relativi agli spazi di lavoro di tutti gli utenti

registrati. Al momento del login su una determinata macchina, questa si prodiga di preparare lo spazio di lavoro in modo adeguato; in questo modo, è possibile accedere ai propri file sia che si stia utilizzando una macchina Linux, Windows o Mac. In particolare:

- i dati dello spazio principale di lavoro vengono caricati automaticamente in ambiente Linux,
- in ambiente Windows, è possibile utilizzare il desktop e la cartella Documenti, mentre lo spazio di lavoro principale è accessibile tramite il disco Z : / ,
- in ambiente Mac, è possibile utilizzare le varie cartelle documenti del sistema, mentre lo spazio di lavoro è accessibile dal Finder seguendo il percorso `Macintosh HD, home, linux`.

Regolamento del CDC

Non esiste un vero e proprio regolamento del centro di calcolo. In generale, è vietato sottrarre le risorse del centro agli altri utenti, svolgendo attività non propriamente legate alla didattica. Esistono poi alcune norme che di fatto vietano alcuni comportamenti volti al danneggiamento di uomini o mezzi, quali:

- cercare di conseguire un accesso non autorizzato presso risorse connesse alla rete,
- influenzare negativamente la normale attività della rete o limitarne l'utilizzabilità o le prestazioni per gli altri utenti,
- distrarre e/o distruggere persone, capacità, elaboratori del CDC e più in generale attraverso la rete Internet,
- provocare trasferimenti non autorizzati di dati, quali ad esempio software o database o altre informazioni,
- usare dispositivi esterni, ossia non appartenenti al CDC, senza previa autorizzazione,
- allontanarsi dalla macchina su cui si sta lavorando, lasciandola occupata, per troppo tempo,
- stampare documenti non legati alla didattica,
- stampare più copie di uno stesso documento,
- consentire a persone estranee l'accesso alle risorse del CDC.

Il centro di calcolo, per i propri servizi di rete, è collegato con il GARR, la rete scientifica di ricerca italiana³. A questo proposito, gli utenti che utilizzano i servizi del CDC devono sottostare alle regole decise dal GARR, che per comodità vengono riportate in appendice a questo documento.

Sempre aggiornati!

Per sapere sempre le ultime novità riguardanti il CDC ed i suoi servizi, è utile dare uno sguardo al sito del centro, all'indirizzo <http://www.cli.di.unipi.it>. In esso sono presenti tutte le informazioni indispensabili per l'uso e la configurazione del proprio account. Va ricordato che la situazione del parco macchine può variare nel tempo anche in modo molto drastico, per cui **tenersi informati è fondamentale**.

Il Gruppo Beatrice mette a disposizione un servizio di *feed RSS* per tutti gli studenti, tramite i quali consultare le ultime notizie del Centro di Calcolo e del Dipartimento di Informatica dell'Università. Per informazioni sull'utilizzo è possibile consultare la sezione apposita nel sito del Gruppo beatrice all'indirizzo <http://beatrice.cli.di.unipi.it/web/servizi/rss>.

Consigliamo anche di controllare, quando necessario, i cambiamenti degli orari di apertura del centro di calcolo e della consegna di account e MAS, sui volantini appesi nelle bacheche a ridosso dei laboratori.

³ Per informazioni sull'attività del GARR è possibile consultare il sito <http://www.garr.net>.

Parte seconda: sistemi Unix e Linux

UNIX⁴ è il nome di un sistema operativo nato alla fine degli anni sessanta presso i laboratori di ricerca Bell. Le caratteristiche principali sono l'adattabilità a tutti gli scopi di lavoro (*general purpose*), la capacità di eseguire più programmi in parallelo (*multitasking*) e la possibilità di gestire contemporaneamente il lavoro di più utenti dividendo tra di essi le risorse della macchina (*multiutente*). La società AT&T, proprietaria dei laboratori Bell, permise la distribuzione libera di UNIX e del relativo codice sorgente presso le università americane, fino a quando con la versione 7 il codice venne considerato segreto commerciale. Tuttavia, molti ricercatori avevano già messo mano al codice, studiandolo ed apportando modifiche e migliorie; il passo fu breve nella realizzazione di nuovi sistemi Unix non vincolati alla AT&T, primo fra tutti la Berkeley System Distribution, BSD. Con il passare del tempo molti hanno realizzato sistemi operativi simili a UNIX nelle idee e nelle funzionalità, creando una sorta di “famiglia” di sistemi, operanti su molte architetture diverse; per permettere compatibilità tra le varie piattaforme sono stati sviluppati diversi standard, ancora oggi utilizzati.

Il sistema Linux è, in ordine di tempo, uno degli ultimi arrivati nella famiglia Unix. È stato sviluppato a partire dal 1995 dall'idea dello studente finlandese Linus Torvalds. Il codice sorgente di Linux è sempre stato ed è tuttora di pubblico dominio; questo ha fatto sì che nel tempo migliaia di sviluppatori da ogni parte del mondo si interessassero al progetto contribuendone al miglioramento. Al giorno d'oggi Linux è un sistema solido, utilizzabile sia su personal computer Intel compatibili che su molte altre famiglie di processori (tra le più importanti: Alpha, 680x0, PowerPC, Sparc e MIPS).

Distribuzioni

Linux (e più in generale un sistema Unix) viene rilasciato attraverso delle *distribuzioni*. Una distribuzione non è altro che una raccolta più o meno vasta di software necessario per ottenere un sistema funzionante.

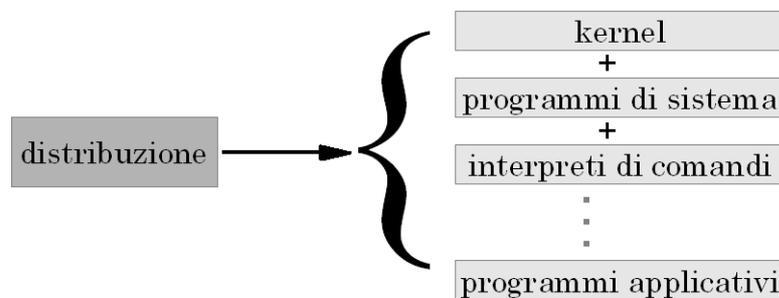


Figura 2: schema di una distribuzione

Non esiste quindi un elenco standard di programmi che devono essere inseriti all'interno di una distribuzione, anche se taluni sono indispensabili per poter lavorare con il calcolatore: è necessario anzitutto un *kernel*, un programma che gestisce il lavoro della macchina gestendone le risorse; una serie di *programmi di sistema*, che sovrintendono la gestione di funzioni specifiche come l'avvio e l'arresto della macchina e di servizi correlati ad essa; uno o più *interpreti dei comandi*, programmi che ricevono in ingresso istruzioni date dall'utente – principalmente invocazioni di altri programmi – e le eseguono; uno o più *programmi applicativi* che, a vari livelli, l'utente utilizza durante una sessione di lavoro; e così via. L'insieme di un numero sufficiente di programmi adeguati costituisce il sistema vero e proprio.

4 Da questo capitolo si fa distinzione tra UNIX, che è il nome proprio del software capostipite di proprietà della AT&T, e Unix, termine con il quale si intende generalmente la famiglia di programmi nati dopo UNIX.

In questa ottica, il software Linux è in realtà solo la componente kernel: in genere per avere un sistema completo vengono aggiunti dei programmi creati dal progetto GNU; il sistema risultante da questa fusione viene chiamato GNU/Linux.

La gestione dei dati in ambiente Unix

File

Generalmente, i dati di un sistema vengono organizzati in *file*. Un file è un archivio corrispondente ad sequenza di byte contenenti informazioni, al quale sono associate alcune proprietà come un nome identificativo, una data di creazione etc. Le informazioni contenute all'interno di un file possono essere di qualunque tipo conosciuto (un filmato video, una foto o un disegno, una registrazione audio o un documento di testo), ovviamente se opportunamente codificate. L'insieme dei file è memorizzato su dispositivi chiamati *memorie di massa*; esempi possono essere il disco rigido (*hard disk*), il disco floppy, il CD-ROM e la penna USB.

Il nome di un file è indispensabile, e pertanto non può essere omissivo; può essere composto da una combinazione di caratteri alfanumerici, cioè lettere e numeri, oltre ai caratteri *.* (*punto*) e *_* (*sottolineatura* o *underscore*). Similmente a quanto accade in altri sistemi, per identificare il tipo di file si utilizza la convenzione di aggiungere al suo nome un carattere di punto seguito da un codice di tre o quattro caratteri, detto *estensione del file*. Esempi di estensione possono essere *.txt* per un documento di solo testo, *.c* per un file sorgente scritto in linguaggio C, *.htm* o *.html* per un documento HTML. Contrariamente a quanto accade usualmente in Windows, l'estensione del file non determina forzatamente il suo tipo, ma rappresenta solo una forma di comodo per l'utente che deve riconoscere il tipo; un file che si chiama *elenco.doc* potrebbe essere tranquillamente un programma che elenca tutti i documenti presenti in una cartella specifica, mentre di norma in Windows può denotare solamente un file creato con il programma Microsoft Word.

Filesystem

In Unix i file vengono organizzati attraverso una struttura logica *ad albero*, chiamata *filesystem virtuale* o VFS. Come succede in molti altri sistemi, tipo Windows, tale struttura è formata da una o più *directory* che a loro volta possono contenere file oppure altre *directory*, creando così una sorta di gerarchia che viene rappresentata tipicamente tramite uno schema nel quale, partendo dalla *directory* principale (detta *radice*) si diramano le varie *sottodirectory* con il rispettivo contenuto. Contrariamente a quanto succede in Windows, non esiste una distinzione tra diversi dischi e dispositivi; il filesystem fa capo ad una sola *directory*, chiamata *root* e contrassegnata con il carattere */*.

I vari dispositivi come il lettore CD-ROM o il lettore floppy vengono inseriti all'interno del filesystem alla stregua di altre *directory*. Facendo riferimento all'immagine, si può vedere come ad esempio all'interno di Windows esistano diversi dispositivi, non vincolati l'uno all'altro, come una unità floppy A: e un lettore CD-ROM E: e così via; gli stessi dispositivi, in un filesystem Unix, sono raggiungibili rispettivamente attraverso i *percorsi* */mnt/floppy* e */mnt/cdrom*.

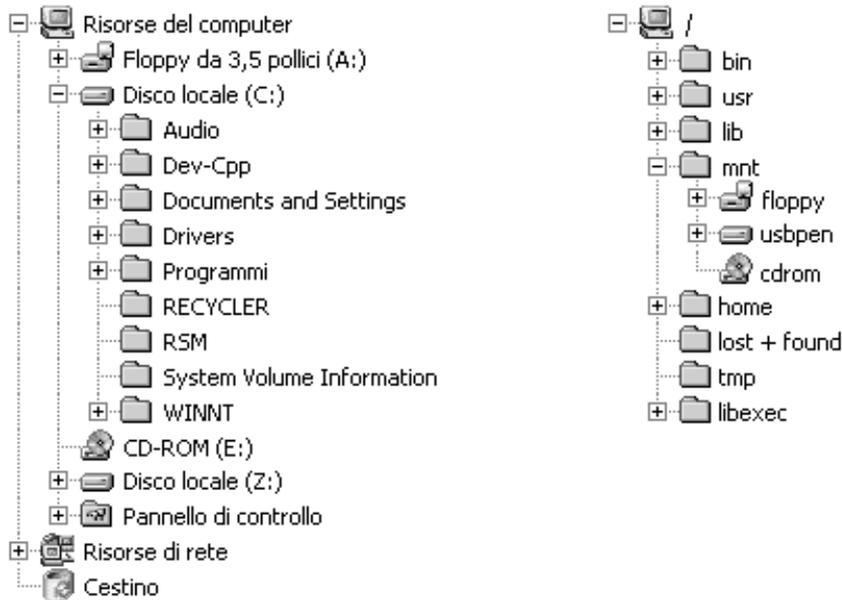


Immagine 2: organizzazione dei dati in Windows e Unix

Una caratteristica fondamentale di Unix è che tutto viene considerato come un file, compresi directory e dispositivi, che vengono trattati come file speciali.

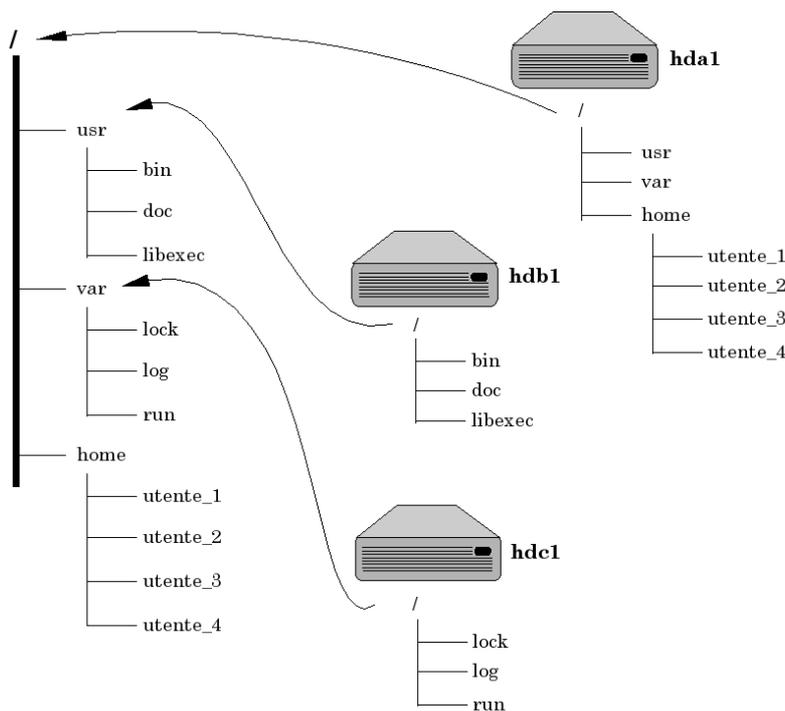


Figura 3: mount di dischi nel filesystem Unix

Poiché il filesystem di Unix è virtuale, esso non è rappresentativo di un qualsiasi dispositivo fisico presente nella macchina, ma bensì viene creato assemblandone il contenuto tramite una operazione chiamata *mounting* del filesystem. L'esempio mostrato in Figura 3 può dare l'idea di quanto accade:

- all'avvio del sistema, il filesystem virtuale non esiste. Il programma che carica il kernel del sistema gli indica, tra gli altri parametri, quale sarà il dispositivo da inserire per primo nel filesystem e dal quale recuperare i dati necessari per continuare;

- tutti i dischi del sistema sono organizzati, ognuno a modo proprio, con dei filesystem propri, ognuno dei quali è quindi visto dall'utente come un'albero di directory a sé stante che si dirama a partire da una propria `root`;
- appena creato, il filesystem virtuale comprende solo la directory `root`, la quale è vuota; il sistema legge le istruzioni precedentemente acquisite ed effettua una prima operazione di `mount`, leggendo in questo caso le informazioni contenute nel disco denominato `hda1` e sostituendo la directory `root` del VFS con quella del disco;
- il sistema ora può leggere altre informazioni dal nuovo filesystem popolato. Una seconda operazione di `mount` viene effettuata per caricare il filesystem del disco `hdb1`, che in questo esempio va a sostituire il contenuto della directory `usr`;
- allo stesso modo, un terzo disco `hdc1` viene letto ed il suo contenuto va a riempire la directory `var`.

Le operazioni di `mount` possono procedere un numero arbitrario di volte, avendo a disposizione un numero di dispositivi di memoria sufficiente. Effettuato il `mount` del disco, è possibile utilizzarne le informazioni contenute fino a quando non si procederà con l'operazione inversa, detta `umount`. L'`umount` di un dispositivo è possibile solo nel momento in cui nessun utente o programma sta utilizzando in alcun modo i suoi dati. Questa condizione, necessaria per mantenere l'integrità dei dati, si ripercuote anche sul funzionamento dei dispositivi rimovibili, come il floppy disk, il CD-ROM o la penna USB. Sempre per ragioni di sicurezza, il `mount` dei nuovi filesystem non è permesso a tutti gli utenti del sistema e comunque non con gli stessi privilegi: di solito l'amministratore configura il sistema indicando una specifica directory per ogni dispositivo, di modo che più operazioni di `mount` non possano sovrapporsi modificando più volte la stessa porzione di filesystem, azione questa che potrebbe potenzialmente danneggiare alcuni dati essenziali.

Percorsi di file e abbreviazioni

All'interno di una directory del filesystem non sono consentite ambiguità di nome, ovvero, ogni file deve avere un nome distinto. Possono però esistere più file con lo stesso nome, se dislocati in directory differenti. Per permettere di effettuare una disambiguazione in ogni caso, è sufficiente specificare il percorso, o *path*, che permette di raggiungere il file attraversando le directory dell'albero del filesystem. Si possono distinguere due tipi di *path*:

- **path assoluto**: a partire dalla directory radice si elencano tutte le directory da attraversare fino a raggiungere il file, separate dallo slash `/`; lo stesso carattere è fondamentale all'inizio del percorso, poiché indica la directory radice;
- **path relativo**: similmente al path assoluto, ma in questo caso il carattere slash all'inizio viene omissso, e si elencano tutte le directory a partire da quella in cui si è posizionati al momento.

La Figura 4 illustra un possibile scenario: all'interno del filesystem si trovano più file aventi lo stesso nome, `documento.doc`. I due path `/var/documento.doc` e `/tmp/var/documento.doc` vengono interpretati sempre alla stessa maniera, poiché partendo dalla directory radice definiscono uno ed un solo file ciascuno; il path `var/documento.doc` invece si può riferire a tre documenti, in relazione alla directory in cui l'utente è entrato: quando questa è `/` il file è `/var/documento.doc`, quando è `/var` si riferisce a `/var/var/documento.doc` e quando è `/tmp` allora il file riferito è `/tmp/var/documento.doc`. Discorso simile per il path `tmp/documento.doc`. Per quanto l'utilizzo di path relativi sembri a prima vista solo una complicazione, questi sono utilizzati efficacemente in specie per l'abbreviazione di percorsi che altrimenti dovrebbero essere costantemente ripetuti.

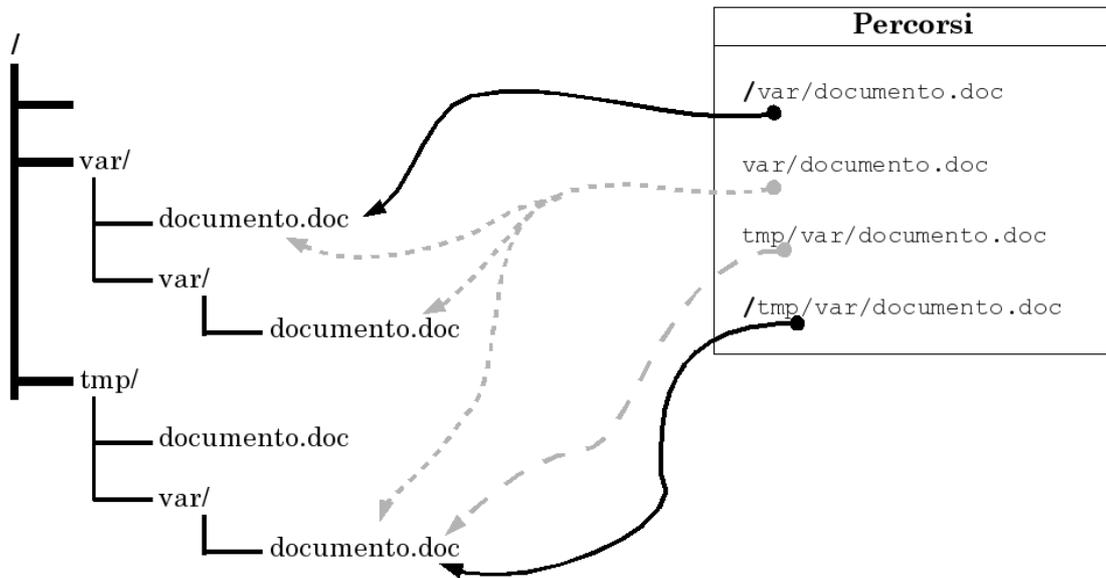


Figura 4: path relativi e assoluti a confronto

La gestione degli utenti in Unix

I sistemi Unix permettono da sempre una separazione tra gli utenti che usufruiscono della stessa macchina, tramite l'utilizzo degli account. Quando questo viene creato, l'utente ha la possibilità di accedere al sistema e di utilizzare un insieme dei programmi disponibili, lavorando all'interno di una speciale directory di cui è proprietario, denominata *home directory*. All'interno della propria home l'utente può fare di tutto, creando, modificando e cancellando file e sottodirectory.

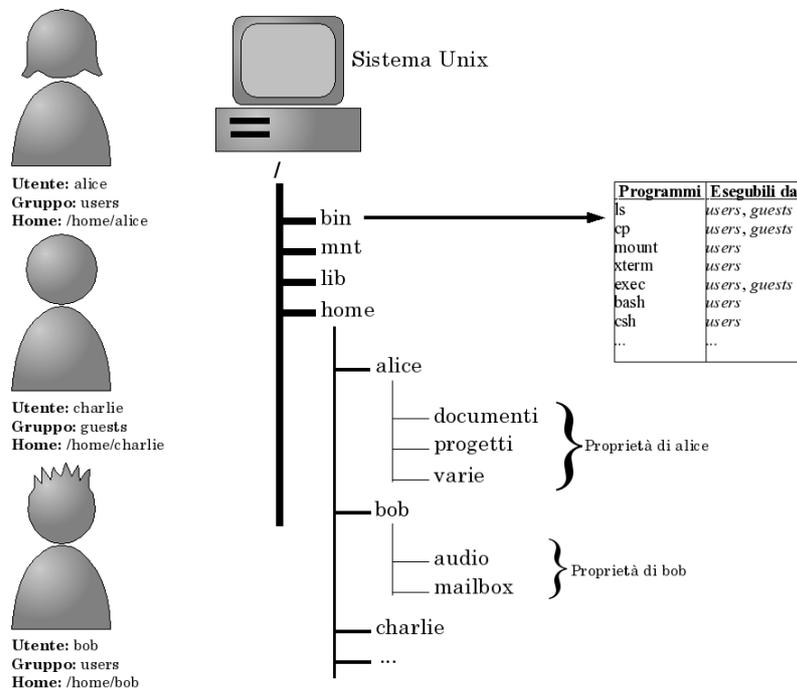


Figura 5: utenti, gruppi e risorse usabili

Gli utenti di un sistema vengono associati ad uno o più *gruppi*. Gli utenti di uno stesso gruppo hanno gli stessi diritti di accesso alle medesime risorse. Ad esempio, potrebbe esistere un file chiamato `istruzioni_per_users.txt` il quale contiene informazioni utili solo per gli utenti appartenenti al gruppo `users`, e che quindi deve essere letto esclusivamente dai suoi appartenenti.

Tra tutti gli utenti di un sistema ne esiste uno, chiamato `root`, che ha pieno controllo di tutte le risorse. È l'utente `root` che gestisce la creazione e l'eliminazione degli account, dei gruppi e dell'accesso degli utenti alle risorse. Nei sistemi di grandi dimensioni come ad esempio il centro di calcolo, la figura che detiene l'account di `root` è il sistemista.

Utenti, gruppi e diritti di accesso alle risorse

Come detto prima, in Unix ogni risorsa viene gestita come un file appartenente al filesystem. Tra le informazioni riguardanti ogni file, oltre al nome, sono presenti anche il nome dell'utente e del gruppo al quale appartiene il file. Per garantire la sicurezza dei dati, il sistema associa ad ogni elemento del filesystem una serie di informazioni riguardanti l'accesso e la modifica dell'elemento stesso, chiamate *permessi*:

- il permesso di **lettura**, ovvero l'accesso al file senza possibilità di modifica, contrassegnato dalla lettera `r` (*read*);
- il permesso di **scrittura**, cioè la modifica totale o parziale dei dati e delle proprietà del file, inclusa quindi la sua eliminazione, contrassegnato dalla lettera `w` (*write*);
- il permesso di **esecuzione**, che associato al file di un programma determina se questo può essere eseguito o meno, mentre per una directory indica che è possibile accedervi, contrassegnato dalla lettera `x` (*execution*).

I tre permessi `r`, `w` e `x` sono rappresentati ognuno da un bit, per cui possono assumere il valore di vero (1) o falso (0), con il relativo significato che ne consegue.

L'insieme dei permessi non viene applicato a tutti allo stesso modo, bensì è operata una distinzione, per ogni file, che associa determinati diritti a

- l'utente che possiede il file,
- il gruppo di utenti che possiede il file,
- gli utenti che non appartengono al gruppo sopracitato.

Complessivamente le informazioni di accesso di un elemento sono date da una *maschera* di nove bit, che indica l'insieme dei diritti di accesso sul file stesso; tale maschera viene presentata nella forma `<uuu><ggg><ooo>`, mostrando in sequenza i permessi per l'utente (*user*), per il gruppo (*group*) e per gli tutti gli altri (*other*). L'output di esempio riportato qui di seguito, fornito dal comando `ls`,

```
pippo@olivia:~$ ls -l
total 24
 4 -rw-r----- 1 pippo studenti 515 Feb 18 2003 domanda.tex
12 -rw-rw-rw- 1 pippo studenti 11319 Feb 7 2004 noam-1.0.0.tar.gz
 4 -r--r--r-- 1 pippo studenti 116 Mar 1 2003 rgb.txt
 4 -rwxr-xr-- 1 pippo studenti 4096 Mar 3 2005 conquer_world
```

mostra in ogni riga alcune proprietà del file, tra le quali – partendo da destra – il nome del file, la data di creazione, il gruppo e l'utente di appartenenza, e la maschera di bit associata⁵, la cui interpretazione è mostrata nella Tabella 1: quando il bit di un diritto di accesso è settato come vero, ne viene mostrata la lettera corrispondente tra lettura, scrittura ed esecuzione; viceversa, per un bit settato come falso, viene mostrata un carattere linea `-`.

Nome del file	Utente di appartenenza	Gruppo di appartenenza	diritti		
			utente	gruppo	altri
domanda.tex	pippo	studenti	lettura, scrittura	lettura	-

⁵ La maschera dell'esempio è formata da 10 caratteri invece di 9; questo perché il primo carattere da sinistra è utilizzato per usi speciali. I caratteri interessati per i diritti d'accesso in questo esempio sono tutti escluso il primo da sinistra.

Nome del file	Utente di appartenenza	Gruppo di appartenenza	diritti		
			utente	gruppo	altri
noam-1.0.0.tar.gz	pippo	studenti	lettura, scrittura	lettura, scrittura	lettura, scrittura
rgb.txt	pippo	studenti	lettura	lettura	lettura
conquer_world	pippo	studenti	lettura, scrittura, esecuzione	lettura, esecuzione	lettura

Tabella 1: esempi di file e relativi diritti di accesso

I permessi di ogni file possono essere cambiati in ogni momento dal suo utente possessore e dall'utente `root`, che ha sempre accesso a tutti i file del sistema pur non essendo né proprietario né appartenente ad un gruppo proprietario di alcuni file.

Lavorare con Linux tramite la shell

La shell, o interprete dei comandi, è un programma che si fa carico di eseguire i vari ordini impartiti dall'utente tramite la tastiera. Quando la shell è avviata, stampa a video una riga contenente alcune informazioni di base (il *prompt*) e poi resta in attesa dei comandi utente. Le interazioni tra la shell e l'utente si svolgono attraverso il *terminale*, ovvero una interfaccia di solo testo, senza cioè l'ausilio della grafica come ad esempio succede in ambiente Windows, oppure all'interno di una apposita finestra detta *terminale virtuale*. Pur sembrando una scelta anacronistica, l'utilizzo di shell e terminali risulta ancora molto vantaggiosa, in merito al fatto che la tastiera rappresenta ancora il metodo di input più veloce, specie nella programmazione. È quindi consigliabile prendere confidenza con questa metodologia di lavoro, che può presentarsi molto utile. In modalità non grafica Linux lancia al suo avvio almeno sei terminali, chiamati *tty*⁶ ed identificati da un numero progressivo a partire da 1, dei quali solo uno alla volta viene visualizzato. Per cambiare la visualizzazione tra un terminale e l'altro, è sufficiente premere una combinazione di tasti formata da `ALT` ed uno tra i tasti funzione, il cui numero identifica il medesimo terminale; ad esempio per passare da `tty1` a `tty3` vanno premuti `ALT` ed `F3`.

Esistono diversi programmi differenti che implementano le funzionalità di una shell; la più utilizzata comunemente nei sistemi GNU/Linux è la shell Bash, ed ogni nuovo account creato per il centro di calcolo è impostato per l'utilizzo di questo programma. Una shell alternativa utilizzata in passato, chiama Tcsh, è ancora disponibile per l'utilizzo.

Il prompt

Per tutta la durata della sua esecuzione, la shell mostra, nella parte sinistra della riga in cui è posizionato il cursore in attesa di comandi, il suo prompt. Le informazioni contenute all'interno del prompt variano a seconda della configurazione del sistema; in genere, sono sempre presenti alcune informazioni utili:

- il nome dell'utente attualmente collegato alla macchina,
- il nome della macchina dalla quale è stata avviata la shell,
- la directory in cui si è posizionati, dove cioè è ora possibile lavorare.

```
pippo@olivia:~$
```

Ad esempio, nella riga di prompt qui sopra, l'utente pippo è connesso alla macchina olivia, nella sua home directory, indicata con il carattere tilde `~`.

6 Abbreviazione di *teletype*, ovvero telescrivente.

Comandi, parametri ed opzioni

L'approccio più semplice e generico all'utilizzo della shell è l'esecuzione di comandi: l'utente digita il nome del programma che desidera; alla pressione del tasto `Invio`, la shell cerca il file del programma richiesto, quindi lo esegue. Generalmente di seguito al nome del programma vengono aggiunte altre informazioni che prendono il nome di *parametri*, separati l'uno dall'altro da uno o più spazi. I parametri di un programma vengono a loro volta divisi in due categorie, le *opzioni* e gli *argomenti*.

Una **opzione** (detta anche *switch* o *flag*) è un comando specifico che modifica il comportamento del programma stesso; è facilmente riconoscibile poiché in genere è composta da un carattere linea - seguito da uno o più caratteri alfanumerici, come ad esempio `-i`, `-l`, `-s`, `-9`. Ogni programma può avere le proprie opzioni, che non hanno necessariamente lo stesso significato per altri programmi.

Un **argomento** è invece un parametro variabile, che permette all'utente di indirizzare il programma per operare specificatamente con i valori passati dall'argomento stesso. Un esempio di argomento per un programma di editing testi potrebbe essere il nome del file che si vuole modificare.

Gestione dei dati tramite la shell

Esiste, in ambiente Unix/Linux, un insieme di comandi comune in tutte le implementazioni che permettono la manipolazione di file e directory in modo semplice e veloce. Molti di questi rispecchiano la filosofia Unix, secondo cui un programma è progettato per fare una sola operazione e nel migliore dei modi.

Elencare i file del sistema

Il comando `ls` (*list*) elenca i file presenti nella directory di lavoro corrente.

```

pippo@olivia:~$ ls
 4 articolo.txt          4 Desktop/          328 guide.pdf
 4 conquer_world*       4 dir.txt           4 noam-1.0.0/
16 Concurrent.java     4 etaxi-1.0.2/      4 rgb.txt
 4 cpp/                 4 file_sync/        4 src/

```

La lista dei file viene mostrata su più colonne per evitare all'utente di sprecare tempo nello scorrimento dell'output. A sinistra del nome del file è presente la sua dimensione in numero di blocchi; a destra, la presenza di un carattere asterisco `*` denota che il file ha diritti di esecuzione, mentre uno slash `/` indica che il nome corrisponde ad una directory. Il comando `ls` per default non mostra il contenuto completo della directory, omettendo quei file il cui nome inizia con il carattere punto; per visualizzare questi file nascosti, si utilizza l'opzione `-a`.

```

pippo@olivia:~$ ls -a
 4 ./                   4 articolo.txt      4 etaxi-1.0.2/
 8 ../                  4 conquer_world*   4 file_sync/
 8 .bash_history        16 Concurrent.java  328 guide.pdf
 4 .emacs               4 cpp/             4 noam-1.0.0/
 4 .mailboxlist         4 Desktop/         4 rgb.txt
 4 .xinitrc             4 dir.txt          4 src/

```

I primi due elementi della lista si riferiscono rispettivamente alla directory corrente (`.`) e a quella superiore (`..`): questi possono essere riferiti all'interno di path come qualsiasi altro nome di directory.

Un'altra utile opzione è `-l`, che mostra un maggiore numero di informazioni riguardo ai file non nascosti.

```

pippo@olivia:~$ ls -l
 4 -rw-r--r--  1 pippo  studenti    2291 Nov  7  2003 articolo.txt
 4 -rwxr--r--  1 pippo  studenti     803 Oct  4  2004 conquer_world*
16 -rw-r--r--  1 pippo  studenti   16343 Mar  1  2004 Concurrent.java
 4 drwxr-xr-x  2 pippo  studenti    4096 Jul 12  2003 cpp/
 4 drwxr-xr-x  2 pippo  studenti    4096 Feb  3  2005 Desktop/
 4 -rw-r--r--  1 pippo  studenti     535 Jan 22  2003 dir.txt
 4 drwxr-xr-x  4 pippo  studenti    4096 Feb  9 15:24 etaxi-1.0.2/
 4 drwxr-xr-x  2 pippo  studenti    4096 Jul  6  2004 file_sync/
328 -rw-----  1 pippo  studenti  327813 Feb 11 14:39 guide.pdf
 4 drwxr-xr-x  4 pippo  studenti    4096 Feb 11 12:14 noam-1.0.0/
 4 -rw-r--r--  1 pippo  studenti     116 Mar  1  2003 rgb.txt
 4 drwxr-xr-x  3 pippo  studenti    4096 Jan 27  2004 src/

```

Navigazione nel filesystem

In ambiente Windows il passaggio da una directory (cartella) ad un'altra si effettua tramite le finestre di esplorazione: cliccando col mouse sull'icona di una directory la si “apre”, ovvero ci si posiziona all'interno di essa. Allo stesso modo, da terminale si utilizza il comando `cd` (*change directory*):

```

pippo@olivia:~$ ls
 4 documenti/          4 file_test.c
pippo@olivia:~$ cd documenti
pippo@olivia:~/documenti$ ls
 4 contatti.txt       4 racconti.txt
 4 dati_importati.txt 4 ricette.txt
pippo@olivia:~/documenti$

```

Quando la directory di lavoro si sposta, il prompt viene aggiornato, mostrandone il percorso. In caso ci si trovi in una sottodirectory della propria home, il path mostrato dal prompt viene abbreviato con l'utilizzo della tilde `~`, rappresentante il percorso dalla directory radice fino alla home stessa. Ovunque ci si trovi nell'albero del filesystem, è sufficiente indicare la tilde come argomento di `cd` per tornare alla propria home.

```

pippo@olivia:/usr$ ls
 4 bin/                4 lib/
pippo@olivia:/usr$ cd ~
pippo@olivia:~$

```

Nel caso in cui il prompt della shell non sia impostato per la visualizzazione della directory corrente, il comando `pwd` (*print working directory*) fornisce tale informazione.

```

pippo@olivia:$ pwd
/usr/bin
pippo@olivia:$ cd ~
pippo@olivia:$ pwd
~
pippo@olivia:$

```

Creazione ed eliminazione di directory

Dove consentito, come ad esempio nella propria home, il comando `mkdir` (*make directory*) permette la creazione di una nuova directory vuota. La sua eliminazione avviene invece tramite il comando `rmdir` (*remove directory*), ma solo se la directory da rimuovere è vuota.

```

pippo@olivia:~$ ls
 4 documenti/          4 file_test.c
pippo@olivia:~$ mkdir nuova_directory
pippo@olivia:~$ ls
 4 documenti/          4 file_test.c          4 nuova_directory/
pippo@olivia:~$ rmdir documenti
rmdir: `documenti/': Directory not empty
pippo@olivia:~$ rmdir nuova_directory
pippo@olivia:~$

```

Eliminazione di file

La cancellazione di uno o più file contemporaneamente avviene utilizzando il comando `rm` (*remove*). La sua azione elimina in modo permanente i file specificati, ed occorre quindi cautela nel suo utilizzo. Per gli account del centro di calcolo, `rm` è configurato per essere utilizzato automaticamente con l'opzione `-i`, che chiede conferma all'utente per procedere nell'eliminazione. Per evitare qualsiasi domanda, si usa l'opzione `-f`.

```

pippo@olivia:~$ ls
 4 file_test.c          4 note_sviluppo.txt    4 test.txt
pippo@olivia:~$ rm file_test.c
rm: remove file `file_test.c'? y
pippo@olivia:~$ ls
 4 note_sviluppo.txt    4 test.txt
pippo@olivia:~$ rm -f note_sviluppo.txt
pippo@olivia:~$ ls
 4 test.txt
pippo@olivia:~$

```

Ovviamente l'eliminazione è consentita solo se l'utente ha il permesso di scrittura sugli elementi interessati.

Spostamento e rinomina di file o directory

Lo spostamento di un elemento da un punto all'altro del filesystem si effettua tramite il comando `mv` (*move*). Comunemente si indicano ad `mv` uno o più file o directory da spostare, ed una destinazione nel filesystem, inserita sempre come ultimo parametro. Se vengono specificati più file, deve essere indicata una directory di destinazione come parametro finale.

```

pippo@olivia:~$ ls
 8 applicazione/        4 file_test.c          4 test.txt
24 database/           4 note_sviluppo.txt    8 test_2.txt
pippo@olivia:~$ mv file_test.c applicazione
pippo@olivia:~$ ls
 8 applicazione/        4 note_sviluppo.txt    8 test_2.txt
24 database/           4 test.txt
pippo@olivia:~$ mv test.txt test_2.txt database
pippo@olivia:~$ ls
 8 applicazione/        24 database/           4 note_sviluppo.txt
pippo@olivia:~$ mv database applicazione
pippo@olivia:~$ ls
 8 applicazione/        4 note_sviluppo.txt
pippo@olivia:~$ ls applicazione
24 database/           4 test.txt             8 test_2.txt
 4 file_test.c

```

Il comando `mv` è utilizzato anche per rinominare un file o una directory; in questo caso vengono forniti solo due parametri, il vecchio ed il nuovo nome che verrà assunto dall'elemento: se si vuole rinominare un file, `mv` lo “muove” nel percorso indicato dal nuovo nome, e se questo è preesistente viene sovrascritto; allo stesso modo, una directory viene spostata verso un'altra, se questa non esiste verrà creata, altrimenti la directory sorgente verrà inserita all'interno di quella di destinazione. Anche per `mv` l'opzione `-i` chiede all'utente se si vuole procedere lo stesso in caso di sovrascrittura.

```

pippo@olivia:~$ ls
 8 applicazione/          4 file_test.c          4 test.txt
24 database/             4 note_sviluppo.txt    8 test_2.txt
pippo@olivia:~$ mv test_2.txt prova_finale.txt
pippo@olivia:~$ ls
 8 applicazione/          4 file_test.c          8 prova_finale.txt
24 database/             4 note_sviluppo.txt    4 test.txt
pippo@olivia:~$ mv test.txt prova_finale.txt
mv: overwrite file 'prova_finale.txt'? y
pippo@olivia:~$ ls
 8 applicazione/          4 file_test.c          4 prova_finale.txt
24 database/             4 note_sviluppo.txt

```

Copia di file o directory

Il comando `cp` (*copy*) funziona in modo simile ad `mv`, anche se invece di “spostare” il file ne crea una copia. In ingresso `cp` prende il nome del file da copiare e la sua destinazione: se va copiato nella stessa directory in cui il file si trova, va specificato un nome differente; se i file da copiare sono più di uno, va indicata come destinazione una directory preesistente. Il comando di copia per default non agisce sulle directory; per modificare questo comportamento viene utilizzata l'opzione `-r`.

```

pippo@olivia:~$ ls
 8 applicazione/          4 file_test.c
pippo@olivia:~$ cp file_test.c applicazione
pippo@olivia:~$ ls applicazione
 4 file_test.c
pippo@olivia:~$ cp file_test.c programma.c
pippo@olivia:~$ ls
 8 applicazione/          4 file_test.c          4 programma

```

Ricerca nel filesystem

Esistono diversi comandi che permettono la ricerca di un file o una directory nel filesystem; il più semplice ed immediato di questi è `locate`, che ricerca una qualsiasi espressione all'interno di un database contenente l'immagine del filesystem.

```

pippo@olivia:~$ locate test
/bin/test
/tmp/test
/usr/bin/test
/home/p/pippo/file_test.c
pippo@olivia:~$

```

Una ricerca effettuata con `locate` è molto veloce, tuttavia non è sempre accurata, poiché il database viene ricostruito di massima una volta al giorno, per cui un elemento la cui presenza nel filesystem è di poche ore non viene registrato e quindi sarà difficilmente rintracciabile. Un metodo più sicuro di ricerca consiste nell'utilizzo del programma `find`, ma data la vastità di opzioni del comando stesso non verrà trattato in questo documento.

Modifica dei permessi di un elemento

Modificare i diritti di accesso ad un file o directory richiede che l'utente ne sia il possessore, o quantomeno appartenga al gruppo possessore ed abbia il permesso di scrittura. Il comando che permette la modifica dei diritti è `chmod`. Il parametro principale è costituito dalle informazioni sui diritti sul quale mettere mano, codificate nella forma `<livello><azione><diritto>`:

- il **livello** indica se si vuole agire sui permessi di utente, gruppo, altri o tutti indistintamente (rispettivamente `u`, `g`, `o`, `a`);
- l'**azione** può essere l'assegnazione del diritto, indicata dal segno `+`, la sua privazione indicata con il `-`;

- il **diritto** è uno tra lettura, scrittura o esecuzione (r, w, x).

```

pippo@olivia:~$ ls -l
 4 -r--r--r--  1 pippo  studenti      2291 Nov  7  2005 adesioni.txt
 4 drwxr-xr-x  2 pippo  studenti      4096 Jul 12  2005 cpp/
 4 -rw-r--rw-  1 pippo  studenti       535 Jan 22  2005 demo.c
pippo@olivia:~$ chmod u+w adesioni.txt
pippo@olivia:~$ ls -l
 4 -rw-r--r--  1 pippo  studenti      2291 Nov  7  2005 adesioni.txt
 4 drwxr-xr-x  2 pippo  studenti      4096 Jul 12  2005 cpp/
 4 -rw-r--rw-  1 pippo  studenti       535 Jan 22  2005 demo.c
pippo@olivia:~$ chmod a-w demo.c
 4 -rw-r--r--  1 pippo  studenti      2291 Nov  7  2005 adesioni.txt
 4 drwxr-xr-x  2 pippo  studenti      4096 Jul 12  2005 cpp/
 4 -r--r--r--  1 pippo  studenti       535 Jan 22  2005 demo.c

```

In genere, il permesso di esecuzione è necessario solo nel caso in cui si stia considerando il file di un programma oppure si voglia dare accesso ad una directory.

Esaminare un file

Alcuni programmi permettono direttamente dal terminale di visualizzare il contenuto di file *non binari*, cioè codificati come testo semplice utilizzando codici alfanumerici come ASCII o UNICODE; alcuni esempi sono i file HTML, o i file sorgente codificati in un linguaggio di programmazione. Tra i vari programmi, i più utilizzati sono `more` e `less`. I due comandi svolgono la stessa funzione, stampano cioè a video il contenuto del file, ma in modo differente. `more` scorre il documento dall'inizio alla fine mostrandone una schermata alla volta, senza possibilità di tornare a schermate già visualizzate; premendo la barra spaziatrice si avanza di una schermata, mentre con il tasto Invio si avanza di una riga. `less` invece permette la *navigazione* del documento, tramite l'utilizzo dei tasti freccia su e giù e dei tasti Page Up e Page Down è possibile muoversi in avanti ed indietro di una riga o si una schermata.

```

pippo@olivia:~$ ls
 4 leopardi.txt          4 leopardi2.txt
pippo@olivia:~$ more leopardi.txt
La donzelletta vien dalla campagna,
in sul calar del sole,
col suo fascio d'erba; e reca in mano
un mazzolin di rose e di viole,
onde, siccome suole,
ornare ella si appresta,
dimani, al dì di festa, il petto e il crine.
pippo@olivia:~$

```

Quando si utilizza `less` per esaminare un file, l'ultima riga del terminale mostra sempre un carattere due punti : sempre presente qualunque sia la lunghezza del testo. Quando il file supera le dimensioni di una schermata, `more` mostra nella riga in fondo al terminale alcune informazioni, quali il nome del file, il numero di linee visualizzate rispetto al totale e la porzione del file mostrata, considerata in percentuale.

```

pippo@olivia:~$ ls
 4 leopardi.txt          4 leopardi2.txt
pippo@olivia:~$ more leopardi2.txt
Siede con le vicine
su la scala a filar la vecchierella,
incontro là dove si perde il giorno;
e novellando vien del suo buon tempo,
quando ai dì della festa ella si ornava,
ed ancor sana e snella
solea danzar la sera intra di quei
ch'ebbe compagni dell'età più bella.
--More-- (19%)

```

Per effettuare una ricerca all'interno del file controllando dall'inizio alla fine, è sufficiente digitare il carattere slash / seguito dall'espressione che si vuole ricercare; usando less, è anche possibile effettuare una ricerca all'indietro – cioè dalla fine all'inizio del file – digitando il carattere di punto interrogativo ? . Le occorrenze dell'espressione digitata, se presenti, verranno evidenziate rispetto al testo, siano esse parole intere o meno; per continuare nella ricerca della medesima espressione è sufficiente digitare il carattere n.

```

pippo@olivia:~$ less leopardi2.txt
Siede con le vicine
su la scala a filar la vecchierella,
incontro là dove si perde il giorno;
e novellando vien del suo buon tempo,
quando ai dì della festa ella si ornava,
ed ancor sana e snella
solea danzar la sera intra di quei
ch'ebbe compagni dell'età più bella
/ella

```

Raggiunta la fine del file, more termina uscendo dal programma, al contrario di less; in entrambi i programmi è comunque sufficiente premere il tasto q per terminare immediatamente la visualizzazione, in qualunque parte del file ci si trovi; la pressione del tasto ESC non ha nessun effetto riguardo all'uscita dai due programmi.

```

Questo di sette è il più gradito giorno,
pien di speme e di gioia:
diman tristezza e noia
recheran l'ore, ed al travaglio usato
ciascuno in suo pensier farà ritorno.
:q
pippo@olivia:~$

```

Ricerca del contenuto di un file

Sempre a proposito di file di testo, può accadere di voler controllare velocemente la presenza di una determinata espressione all'interno di alcuni file. Il comando grep esamina tutti gli elementi passatigli come argomenti alla ricerca della frase interessata; quando questa viene trovata, stampa a video l'intera riga interessata ed il nome del file in cui è memorizzata.

```

pippo@olivia:~$ ls
 4 cantoquinto.txt      4 veglia.txt
pippo@olivia:~$ grep mani cantoquinto.txt veglia.txt
cantoquinto.txt:Le tue mani si fanno come un soffio
veglia.txt:delle sue mani
pippo@olivia:~$

```

Tramite l'opzione -n è inoltre possibile conoscere il numero esatto della riga in cui si trova l'espressione ricercata.

Spazio su disco

Diversi comandi permettono di conoscere informazioni riguardanti l'occupazione del disco rigido fornendo dati statistici al riguardo. Il comando `df` (*disk free*) ad esempio mostra la quantità di spazio libero nelle varie unità di memoria di sistema, come l'hard disk o periferiche come il floppy o la penna USB, se inserite.

```
pippo@olivia:~$ df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/hda6              20161172    7050536 12086496 37% /
/dev/hda1                77749         9312     64423 13% /boot
/dev/hdb1              9803940     731276  8574652  8% /disc1
none                   256892         0     256892  0% /dev/shm
homeserver:/RAID/5    140853204   77766980 55931280 59%
/.automount/homeserver/RAID/5
pippo@olivia:~$
```

Il comando `du` (*disk usage*) mostra invece lo spazio occupato dai vari file presenti nella directory corrente. Con l'opzione `-s`, `du` mostra la dimensione totale, ovvero la somma delle dimensioni dei vari file e sottodirectory della directory corrente.

```
pippo@olivia:~$ du
12  ./classes
24  ./c files
20  ./ssh
7232 ./mail
16  ./links
pippo@olivia:~$ du -s
7304
pippo@olivia:~$
```

Un terzo comando, `quota`, permette di conoscere lo spazio che un utente ha a disposizione nel sistema, riferendosi alla propria cartella home. La *quota* di un utente, ovvero la quantità di spazio a lui riservata nel filesystem, è affiancata da un valore che indica il periodo, detto di *grazia*, entro il quale è possibile superare il valore di quota pur mantenendo intatti i propri dati; entro questo lasso di tempo l'utente deve necessariamente liberare spazio nella propria directory home, in modo da diminuire l'uso dello spazio concesso fin al di sotto della quota a lui assegnata.

```
pippo@olivia:~$ quota
Disk quotas for user pippo (uid 22058):
    Filesystem blocks quota limit grace files quota limit grace
homeserver:/RAID/5
    33544 40000 42000          1125 0 0
pippo@olivia:~$
```

Nel controllare la propria home per cercare eventuali file da eliminare, va ricordato anche di controllare i vari file nascosti presenti, creati solitamente dopo l'esecuzione di programmi che necessitano file di configurazione specifici per l'utente e che talvolta creano elementi temporanei tranquillamente eliminabili.

Altri comandi di base

Molti altri comandi fanno parte dell'insieme di base di un sistema Unix; in sostanza, la conoscenza di un buon numero di questi si acquisisce facendo pratica con il sistema. Di seguito riportiamo la descrizione di altri programmi d'uso comune nell'ambiente shell.

Il programma `file` determina il tipo di un elemento fornendone una breve descrizione. Per fare ciò utilizza un suo database in cui sono memorizzate le informazioni su svariate centinaia di formati file, dai più comuni come documenti RTF o HTML ai più particolari; questo implica che in una ristretta percentuale `file` non riesca a riconoscere alcuni elementi, descrivendoli solo come file contenenti generiche informazioni.

```

pippo@olivia:~$ ls
 4 guida.pdf          4 sconosciuto.dat
pippo@olivia:~$ file guida.pdf
guida.pdf: PDF document, version 1.2
pippo@olivia:~$ file sconosciuto.dat
sconosciuto.dat: data
pippo@olivia:~$

```

Il comando `cat` permette la concatenazione di due o più file dati come parametri di ingresso; se questi vengono omessi, `cat` attende dall'utente informazioni in ingresso immesse da tastiera. Se non specificato diversamente il risultato della concatenazione viene mostrato direttamente a video.

```

pippo@olivia:~$ ls
 4 dispari.txt        4 pari.txt
pippo@olivia:~$ cat pari.txt dispari.txt
2
4
6
1
3
5
pippo@olivia:~$

```

Il programma `sort` fa un passo ulteriore, concatenando il contenuto dei file dati in ingresso e stampandone poi le linee risultanti in ordine lessicografico; se non vi sono file in ingresso, `sort` attende le informazioni da tastiera. Per terminare l'immissione di dati per `cat` e `sort`, è sufficiente premere la combinazione dei tasti CTRL e D.

I comandi `head` e `tail` mostrano, rispettivamente la “testa” e la “coda” di un documento, ovvero la parte iniziale e quella finale. Il programma `wc` invece conta il numero di linee di testo contenute in un file.

```

pippo@olivia:~$ ls
 4 divina_commedia.txt
pippo@olivia:~$ head divina_commedia.txt
Inferno - Canto I

```

```

    Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura
ché la diritta via era smarrita.
pippo@olivia:~$ tail divina_commedia.txt
    A l'alta fantasia qui mancò possa;
ma già volgeva il mio disio e 'l velle,
sì come rota ch'igualmente è mossa,

```

```

    l'amor che move il sole e l'altre stelle.
pippo@olivia:~$

```

Alias

Un comando frequentemente utilizzato con certi parametri, può essere richiamato istantaneamente utilizzando un suo alias. Un alias può essere considerato come l'abbreviazione di un qualsiasi comando impartito alla shell, definito essenzialmente per comodità; è naturale, ad esempio, che piuttosto che scrivere ripetutamente `ls -Fs --color -k` ogni volta che si vuole leggere il contenuto di una directory mostrando la dimensione di ogni elemento in kilobyte e colorando i nomi dei file diversamente a seconda del proprio tipo, è molto più semplice e desiderabile digitare un breve comando `ls`, ed ottenere lo stesso risultato. Il nome di un alias può essere definito come una qualsiasi parola formata da caratteri alfanumerici, includendo in questo anche i nomi di comandi preesistenti nel sistema.

```

pippo@olivia:~$ alias la='ls -a -Fs --color'
pippo@olivia:~$ la
 4 ./                4 .mailboxlist      16 Concurrent.java
 8 ../              4 .xinitrc          4 cpp/
 8 .bash_history    4 articolo.txt      4 Desktop/
 4 .emacs           4 conquer_world*    4 dir.txt
pippo@olivia:~$

```

Ovviamente la creazione di un alias avviene solo nell'*ambiente* dell'utente che lo crea e invoca; utenti che vogliono un alias devono definirlo; il comando `alias`, oltre a questa funzione, mostra tutti gli alias caricati al momento dalla shell sul quale si sta lavorando. Viceversa, il comando `unalias` permette l'eliminazione dell'alias.

```

pippo@olivia:~$ alias
alias bye='exit'
alias cp='cp -i'
alias la='ls -a'
alias lj='lpr -Plj0'
alias ll='ls -gl'
alias lla='ll -a'
alias ls='ls -Fs --color'
alias mv='mv -i'
alias rm='rm -i'
pippo@olivia:~$

```

Abbreviazioni di percorsi

Come già visto sopra, nello specificare il path di un elemento, sia esso relativo o assoluto, è possibile utilizzare alcune convenzioni per permettere l'abbreviazione della digitazione del percorso. La Tabella 2 le riassume mostrandone significato e indicando se possono essere concatenate, ovvero sono utilizzabili più di una volta nella scrittura di un path.

Abbreviazione	Utilizzo	Concatenazione?	Esempio
/	Directory radice (root). Ha questo significato solo se posizionata all'inizio del path	no	/usr/bin /home/p/pippo
.	Directory corrente	si	/home/pippo/./dati ./dati
..	Directory superiore alla directory corrente	si	.././pippo /home/l/./p/pippo
~	Directory home dell'utente corrente	no	~/dati
~utente	Directory home di un utente qualsiasi	no	~pippo/dati ~local/mail

Tabella 2: abbreviazioni di percorsi

Metacaratteri

Nella manipolazione di file e directory attraverso la shell può capitare di non essere a conoscenza del nome degli elementi di cui si ha bisogno; oppure si deve elencare un numero enorme di file da passare come argomento per un determinato programma. Per ovviare a questi ed altri problemi la shell mette a disposizione un insieme di *metacaratteri*, ovvero di caratteri speciali che permettono di individuare gruppi di elementi, avendo a disposizione solo una espressione comune. Questi caratteri

jolly, o *wildcards*, possono essere utilizzati singolarmente o concatenati tra essi all'interno di una stringa. Le wildcards più usate sono il caratteri * e ? .

Il **punto interrogativo** indica uno ed un solo carattere qualsiasi appartenente ad una stringa; può essere utilizzato quando i nomi degli elementi da considerare hanno la stessa lunghezza e differiscono per uno o più caratteri particolari.

```

pippo@olivia:~$ ls
 8 gatto          4 scarto
24 ratto          4 scatto
 4 parto
pippo@olivia:~$ ls ?atto
 8 gatto          24 ratto
pippo@olivia:~$ ls ?a?to
 8 gatto          24 ratto          4 parto
pippo@olivia:~$

```

Avendo ad esempio due file chiamati rispettivamente *gatto* e *ratto*, l'espressione che li accomuna è *?atto*; per includere tra questi un nome come ad esempio *parto*, è sufficiente modificare l'espressione in *?at?o*; la stessa esclude invece nomi di file terminanti con uguale desinenza ma più lunghi, come ad esempio *scatto* o *scarto*.

L'**asterisco** indica invece un gruppo non precisato di caratteri; nella fattispecie viene usato quando non si conosce che una minima informazione riguardo agli elementi considerati, come ad esempio una estensione. Usato da solo, senza cioè altri caratteri accostati, viene interpretato come una inclusione totale, elencando quindi tutti gli elementi che riesce a trovare all'interno di una directory. Una espressione del tipo **to* includerà tutti gli elementi il cui nome termina con quelle due lettere, mentre *s*to* richiede che ogni nome abbia anche inizio con la lettera *s*.

```

pippo@olivia:~$ ls *to
 8 gatto          4 scarto
24 ratto          4 scatto
 4 parto
pippo@olivia:~$ ls s*to
 4 scarto          4 scatto
pippo@olivia:~$

```

L'uso di metacaratteri porta con sé, oltre ad ovvi vantaggi, alcuni rischi riguardo la manipolazione dei dati; è importante che nella definizione di espressioni contenenti wildcards non si incappi in distrazioni che, in taluni casi, possono portare alla perdita accidentale dei dati. Ad esempio, se si vuole eliminare da una directory una grande quantità di file il cui nome inizia con la *c* senza doverne confermare la cancellazione, è possibile utilizzare un comando del tipo `rm -f c*`, dove tra la *c* e l'asterisco formano una stringa unica, priva cioè di spazi al suo interno; se i due caratteri fossero stati divisi, digitando ad esempio `rm -f c *`, il comando `rm` avrebbe agito molto diversamente, cercando anzitutto un file da eliminare chiamato *c*, e poi cancellando, ad uno ad uno, tutti gli elementi contenuti nella directory. Onde evitare questo genere di situazioni, negli account del centro di calcolo vari comandi di base sono stati ridefiniti utilizzando degli alias che includono l'opzione `-i`, nel quale si richiede all'utente di confermare le proprie azioni. In ogni caso è **buona norma fare attenzione a come vengono utilizzate le wildcards**, specie se si è alle prime armi con questo strumento.

Mount di dispositivi rimovibili

Come detto in precedenza l'utilizzo dei dispositivi rimovibili si effettua collegandoli al filesystem virtuale. I moderni sistemi Unix prevedono la possibilità di configurare alcuni parametri dei dispositivi di memoria di massa all'interno di un file di testo chiamato *fstab*, solitamente contenuto nella directory */etc*. Le informazioni ivi contenute sono utilizzate principalmente dai programmi che permettono le operazioni di mount di dispositivi. Il file *fstab* è accessibile a tutti in lettura, e solo agli utenti aventi privilegi di *root* è permessa la modifica.

```

pippo@olivia:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/hda1 /boot ext2 defaults 0 1
/dev/hda6 / ext3 defaults,errors=remount-ro 0 1
/dev/hda5 none swap sw 0 0

# removable devices
/dev/fd0 /floppy auto rw,user,noauto 0 0
/dev/sda1 /flash auto rw,user,noauto 0 0
/dev/hdc /cdrom auto ro,user,noauto 0 0

# special filesystems
proc /proc proc defaults 0 0
pippo@olivia:~$

```

In questa schermata di esempio è mostrata una possibile definizione di `fstab`. Ogni riga del file descrive una operazione di `mount`, ad eccezione di quelle che iniziano con il carattere `#` che vengono ignorate; nella prima colonna a sinistra sono elencati i dispositivi fisici interessati, a seguire è descritto il *mount point*, ovvero il punto del VFS a partire dal quale saranno caricati i dati. Le altre opzioni si riferiscono al tipo di filesystem utilizzato all'interno del dispositivi ed ai diritti di accesso per gli utenti, che possono essere limitati alla lettura (`ro`) o comprendere anche la modifica dei dati (`rw`). Il secondo gruppo di tre linee descrive le informazioni riguardanti i supporti rimovibili, il floppy disk, la penna USB ed il CD-ROM.

I comandi `mount` e `umount` effettuano le operazioni di caricamento del filesystem. L'utente `root` non ha limitazioni nell'uso di questi comandi, a differenza degli utenti generici, ai quali è permesso operare solo su alcuni dispositivi descritti in `fstab`, nei quali è presente l'opzione `user`; stando al file descritto nell'esempio, ogni utente può effettuare il `mount` di supporti rimovibili. È sufficiente specificare al comando `mount` il dispositivo fisico oppure il suo `mount point` per effettuare l'operazione. Viceversa, il comando `umount` richiede come parametro il `mount point`.

```

pippo@olivia:~$ mount /cdrom
pippo@olivia:~$ mount /dev/hdc
mount: according to mtab, /dev/hdc is already mounted
pippo@olivia:~$ umount /cdrom
pippo@olivia:~$

```

È possibile effettuare il `mount` di un dispositivo solo una volta nello stesso `mount point`. Per sapere quali dispositivi sono montati nel filesystem, è sufficiente invocare `mount` senza parametri in ingresso. È fondamentale per ogni dispositivo effettuare l'`unmount`, senza il quale il sistema non garantisce la preservazione dei dati.

Input, Output e redirectione

È già stato fatto notare come in un sistema Unix esista una grande varietà di piccoli programmi “specializzati”, che svolgono ognuno una unica e semplice funzione ben precisa: nella stragrande maggioranza dei casi il risultato di una operazione effettuata tramite tali comandi viene indirizzato verso un *canale di output* generico, del quale poi altri programmi decideranno cosa fare. Per fare un esempio concreto, si può supporre che dalla shell sia stato eseguito il programma `more` su un file di testo; se l'utente non specifica cosa fare dell'output, la shell prepara il canale generico in modo tale che il tutto venga poi mandato verso il terminale video, così `more` stamperà sullo schermo tutto ciò che il comando passerà come risultato.

Poiché il risultato di una operazione non sempre è fine a sé stesso può ritenersi utile voler riutilizzare da subito tale output, ad esempio salvandolo all'interno di un file oppure “inviandolo” ad un altro programma come parametro di ingresso. A questo scopo la shell permette di utilizzare degli operatori di *redirezione*, speciali comandi utilizzati nella riga di comando interessata che si dividono in *filtri* e *pipe*.

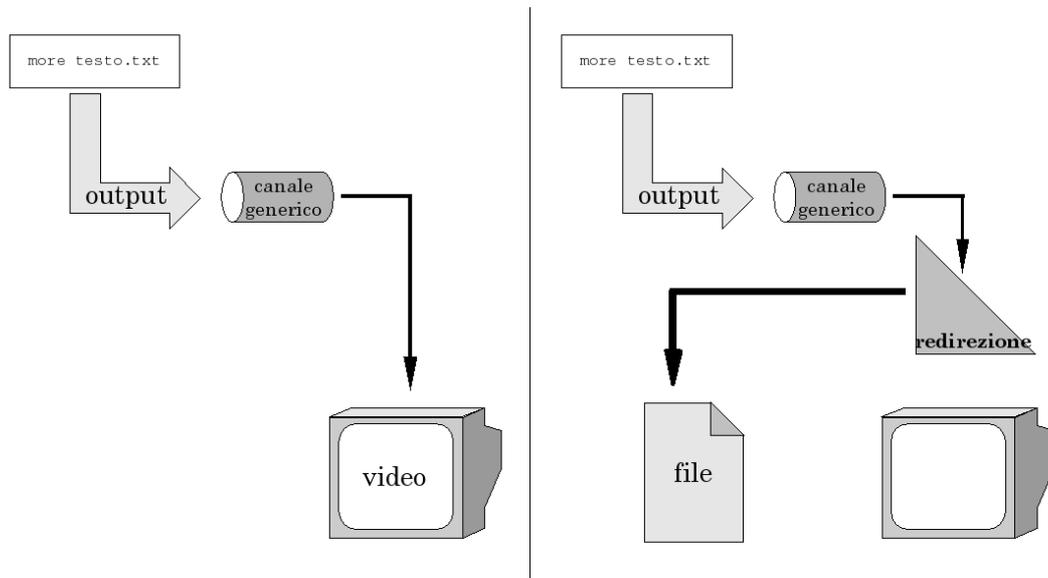


Figura 6: redirezione dell'output di un comando

Un **filtro** permette la redirezione di input e output da e verso un file. È rappresentato da uno o due segni di maggiore o minore (< o >) con significati opposti. Il segno maggiore indica che l'output di un comando verrà scritto all'interno di un file, che se esiste verrà sovrascritto, altrimenti sarà creato sul momento; per evitare la sovrascrittura si utilizzano due segni maggiore consecutivi (>>), col significato che l'output dovrà essere “appeso” al file, cioè verrà scritto nella posizione successiva ai dati già presenti in esso. Viceversa il segno minore permette di specificare il nome di un file dal quale il programma dovrà prendere dati in ingresso; praticamente invece di elencare parametri sulla riga di comando, questi possono essere scritti in un file e passati al programma tramite la redirezione dell'input, evitando all'utente di elencare a volte grandi quantità di dati, risparmiando tempo. Nell'esempio che segue, il contenuto del file `mattina.txt` viene passato al comando `cat` una prima volta, che lo stampa a video, successivamente viene effettuata la stessa operazione effettuando la redirezione dell'output verso il file `ungaretti.txt`.

```
pippo@olivia:~$ cat < mattina.txt
m'illumino
d'immenso
pippo@olivia:~$ cat < mattina.txt > ungarretti.txt
pippo@olivia:~$ cat < ungarretti.txt
m'illumino
d'immenso
pippo@olivia:~$
```

La **pipe** permette invece di riutilizzare l'output di un comando come input di un altro. La pipe, indicata proprio con il carattere omonimo `|`, permette in pratica di effettuare una concatenazione di programmi che, uno alla volta eseguono il loro lavoro su dati che altri programmi hanno elaborato precedentemente. In questo esempio vengono concatenati i comandi `cat`, `grep` e `sort`, utilizzati per cercare all'interno di un file `ricette.txt` i nomi di piatti contenenti la parola `Torta`, ordinati alfabeticamente.

```
pippo@olivia:~$ cat ricette.txt | grep Torta | sort
Torta di lamponi
Torta di noci
Torta di noci al cacao
pippo@olivia:~$
```

Documentazione

La maggior parte del software distribuito in Unix mette a disposizione dell'utente una completa documentazione; i documenti in questione, chiamati *manuali*, sono consultabili utilizzando il comando `man`. Le pagine di manuale vengono esplorate in modo simile a quanto accade con il comando `less`, e contengono svariati tipi di informazioni su tutto ciò che riguarda la parola chiave data; per un generico comando di terminale, anzitutto viene fornita una breve descrizione, quindi una completa sinossi del metodo di utilizzo ed infine la spiegazione dettagliata di come funziona e del significato dei vari parametri di ingresso ed opzioni.

```
pippo@olivia:~$ man cp
CP(1)                                FSF                                CP(1)

NAME
    cp - copy files and directories
SYNOPSIS
    cp [OPTION]... SOURCE DEST
    cp [OPTION]... SOURCE... DIRECTORY
DESCRIPTION
    Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --archive
        same as -dpR
:
```

Le pagine di manuale sono il punto di riferimento principale per l'utilizzo del sistema ed anche per la programmazione, spaziando su diversi argomenti e non di rado una stessa parola chiave, utilizzata per identificare una funzione, un programma o quant'altro, può essere ritrovata in diversi contesti ed avere molteplici descrizioni; per questo motivo le varie pagine di manuale del sistema sono catalogate in sezioni, ognuna riguardante uno specifico aspetto. In particolare, nel sistema Linux, sono presenti nove sezioni:

- la sezione 1 comprende le descrizioni di tutti i comandi ed i programmi invocabili dalla shell,
- le sezioni 2 e 3 contengono documentazione per lo sviluppo di programmi, riguardante cioè l'insieme delle funzioni messe a disposizione da Linux stesso e le varie funzioni offerte dalle librerie software installate nella macchina,
- la sezione 4 documenta alcuni file speciali presenti nel sistema, come i file `null` o `random`, la cui presenza non è determinata da altri programmi se non Linux,
- la sezione 5 descrive vari formati di file di configurazione di programmi presenti nel sistema,
- la sezione 6 include la descrizione dei giochi (se presenti nel sistema),
- la sezione 7 comprende le descrizioni di macro e di convenzioni tipiche di Unix,
- la sezione 8 si occupa dei comandi preposti alla gestione del sistema operativo,
- la sezione 9 include altre informazioni riguardanti la programmazione specificamente in ambiente Linux,

- altre sezioni, indicate con le lettere n, o, l, si riferiscono a documentazione nuova, vecchia e specifica del sistema considerato; queste ultime tre definizioni sono deprecate in favore delle altre, anche se alcuni manuali fanno parte di questi insiemi.

Per conoscere nel dettaglio in quali sezioni si possono trovare informazioni su una determinata parola chiave, vengono utilizzati indistintamente i comandi `man -k` o `apropos`; come risultato mostrano un elenco di pagine manuale riferite alla parola data ed il numero di sezione al quale appartengono. Ovviamente poi, per avere informazioni dettagliate sul comando `man`, è sufficiente consultare la sua pagina di manuale, digitando `man man`.

Linux how-to

Insieme alla documentazione standard delle pagine di manuale, molti sviluppatori ed utilizzatori di sistemi Linux hanno realizzato decine di documenti, chiamati in gergo *how-to*, che descrivono la configurazione e l'utilizzo di varie parti del sistema e programmi. Gli *how-to* non fanno parte della documentazione di riferimento del sistema, ma bensì costituiscono un approccio pratico, spiegando passo dopo passo delle metodologie già sperimentate da altri e che possono essere riprese come esempio, se non seguite alla lettera, per la risoluzione di diversi problemi. Generalmente gran parte delle distribuzioni esistenti fornisce una raccolta di *how-to* consultabili direttamente all'interno del proprio sistema⁷; sono comunque reperibili in innumerevoli siti internet, primo fra tutti quello del Linux Documentation Project, il cui indirizzo è <http://www.tldp.org/> e la sua controparte italiana, <http://it.tldp.org/>.

Interfaccia grafica: il sistema X Window

Accanto all'interfaccia a riga di comando, i sistemi Unix offrono un sottosistema grafico altrettanto versatile, genericamente riferito come *sistema X Window*. Come succede spesso in Unix, i programmi facenti parte di questo insieme non sono indispensabili per il funzionamento e l'amministrazione del sistema, al contrario di quanto succede nei sistemi Windows, dove per scelta progettuale l'utente interagisce quasi esclusivamente a mezzo dell'interfaccia grafica, implementata per larga parte nel cuore del sistema operativo e denominata GDI.

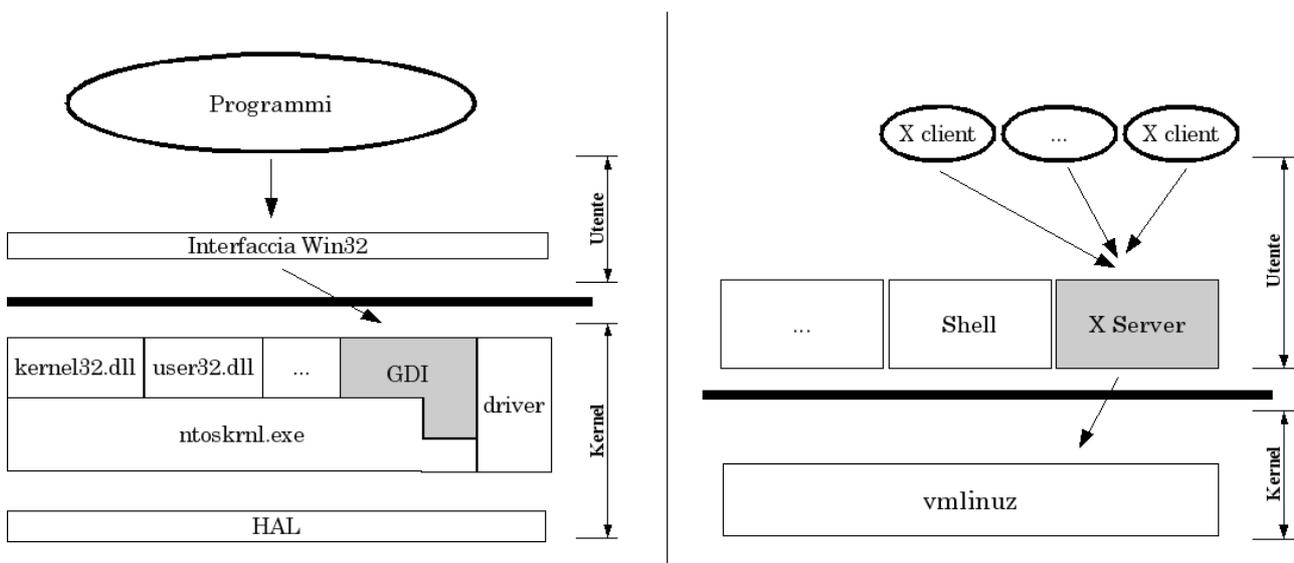


Figura 7: implementazione della GUI a finestre in Windows e Linux

Il sistema X è stato disegnato fin dal principio come un insieme di moduli cooperanti indifferentemente sia all'interno di una stessa macchina che su calcolatori differenti. Similmente a quanto accade per Internet, X funziona secondo il modello *client/server*.

⁷ La directory *how-to* è situata solitamente all'interno di `/usr/doc` oppure `/usr/share/doc`, ed il suo nome può variare a seconda delle distribuzioni.

- Il programma principale, denominato *X-server*, viene avviato sulla macchina in cui l'utente vuole lavorare; l'*X-server* prepara il sistema per l'interfaccia grafica, caricando alcuni driver appositi per la scheda video ed i dispositivi di input come mouse e tastiera, configura la risoluzione e la frequenza di aggiornamento del monitor, dopodiché si mette in attesa di istruzioni da altri programmi; finché questo non accade il server mostra una schermata vuota.
- Quando un qualsiasi programma, concepito per la modalità grafica, detto *X-client*, viene avviato dall'utente, si connette all'*X-server* e, ogni volta che deve mostrare qualcosa sul video, gli invia una serie di comandi che descrivono le operazioni volute.
- Server e client comunicano attraverso un protocollo standard, l'*X-protocol*. Quando il server riceve una operazione descritta tramite l'*X-protocol*, sulla base dei dati ricevuti disegna sul video quanto richiesto, gestendo anche le modifiche che si ripercuotono sui “disegni” di eventuali altri programmi client in esecuzione al momento.

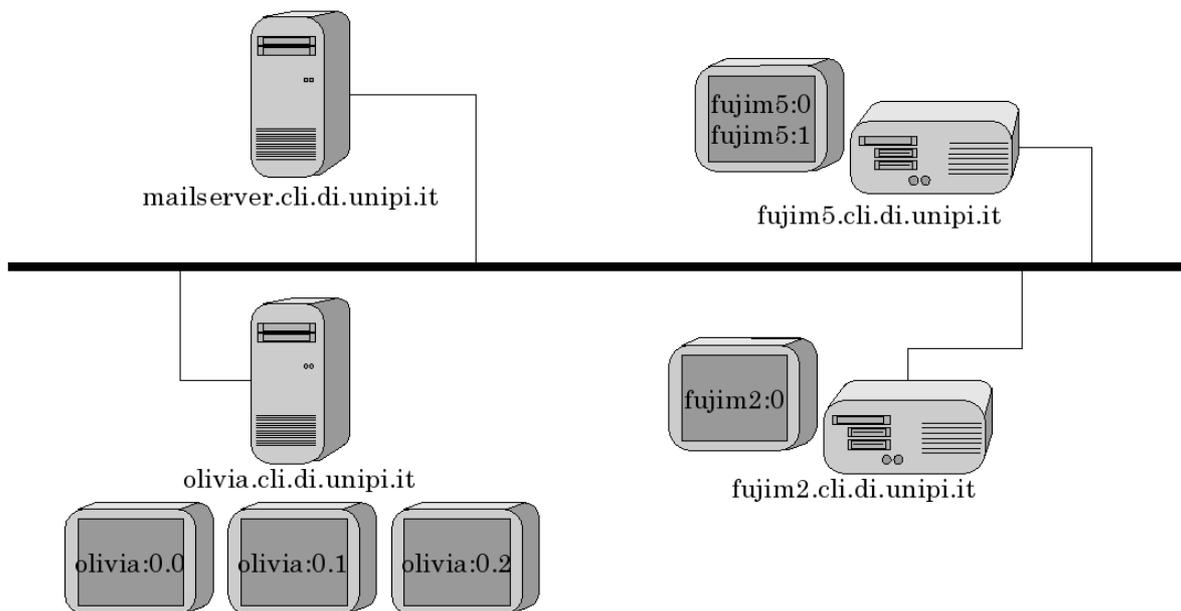


Figura 8: display e screen generati associati agli X-server

Questo schema di comunicazione fa sì che programmi avviati su una macchina possano essere visualizzati su un'altra, permettendo l'interazione dell'utente da remoto rispetto al computer su cui risiede il programma. Per questo motivo è necessario poter indirizzare univocamente tutte le possibili combinazioni di programmi server e di monitor utilizzabili; a questo proposito vengono introdotti i concetti di *display* e *screen*. Il nome **display** identifica un X-server avviato su una macchina; poiché in essa possono coesistere più istanze di server, vengono identificate tramite una numerazione progressiva a partire da 0. Lo **screen** invece identifica uno o più dispositivi fisici di output, tipicamente il monitor del computer. All'avvio di un programma client, è possibile indicare dove dovrà essere visualizzato, indicando uno speciale indirizzo di destinazione descritto nella forma *<nome della macchina>:<numero display>.<numero screen>*. La Figura 8 mostra una possibile situazione: su tre delle quattro ipotetiche macchine sono avviati diversi display (X-server): olivia:0, fujim2:0, fujim5:0 e fujim5:1. Il display olivia:0 controlla tre screen, i cui indirizzi sono olivia:0.0, olivia:0.1, olivia:0.2. A sua volta, sul computer fujim5 sono presenti due display, ma solo uno screen, per cui gli indirizzi possono essere ridotti al semplice elenco dei display, fujim5:0 e fujim5:1.

Questa elasticità di funzionamento del sistema viene pagata con una maggiore complessità, che si traduce spesso volte in un calo di prestazioni, percepito in relazione ai tempi di risposta rispetto alle interazioni con l'utente. Chi sta davanti al monitor può avere la sensazione che il sistema sia rallentato, soprattutto per quanto riguarda piccole modifiche nel disegno dello schermo, come la semplice pressione di un tasto.

Comandi e funzioni generiche dell'X-server

X-server e terminali possono convivere ed essere utilizzati nella stessa macchina; se dal terminale si vuole passare alla interfaccia grafica, è sufficiente utilizzare la solita combinazione del tasto ALT con uno dei tasti funzione che indicano terminali non allocati: se ad esempio nel sistema sono caricati sei terminali, ai quali si ha accesso tramite di ALT con i tasti da F1 a F6, il server grafico viene configurato per essere attivato da ALT e F7 ed allo stesso modo ulteriori X-server utilizzeranno combinazioni dei tasti funzione superiori. Viceversa, per passare dalla sessione grafica ad un terminale, alla solita combinazione viene aggiunto il tasto CTRL.

La terminazione di un X-server può avvenire in modo “brutale”, attraverso la pressione dei tasti CTRL, ALT e BACKSPACE. L'ordine impartito è di terminare immediatamente il server, il che significa che tutte i programmi client ad esso collegati verranno terminati: l'utilizzo di questa funzione è consigliato solo in casi di estrema necessità.

Una importante funzione offerta dal server è la copia automatica di testo. Ogni volta che con il mouse si seleziona un gruppo di caratteri da una qualsiasi applicazione, questi vengono direttamente copiati in una apposita area di memoria, in modo simile a quanto succede per l'area appunti di Windows. Per riutilizzare tali dati è sufficiente premere il tasto centrale del mouse nel punto in cui si vuole che siano immessi. In caso non si disponga di un mouse a tre tasti, il server X permette l'emulazione del tasto centrale tramite la pressione contemporanea dei due tasti presenti.

Window manager ed ambienti desktop

L'approccio modulare del sistema X ha fatto sì che con il tempo siano state sviluppate decine e decine di suite di programmi, aventi tutte lo scopo di gestire l'ambiente grafico in modo funzionale e semplice; per quanto possano essere diverse si compongono più o meno di un insieme simile di elementi:

- un *gestore di finestre (window manager)*, preposto ad organizzare ogni applicazione in una apposita cornice alla quale sono associate diverse proprietà, come il ridimensionamento della finestra stessa o la chiusura dell'applicazione;
- un *gestore del desktop (desktop manager)*, che si occupa dell'interazione tra l'utente e la superficie dello schermo non appartenente a nessuna finestra, chiamata desktop, e che di solito è replicata in diverse istanze chiamate *desktop virtuali*,
- un *gestore di sessione (session manager)*, che recupera le informazioni riguardanti il profilo dell'utente correntemente collegato, tra cui le preferenze grafiche di tutto il sistema, inclusi i colori delle finestre, lo stile dei caratteri, eventuali immagini di sfondo del desktop, ecc,
- uno o più *menù delle applicazioni (application menu)*, che permettono l'accesso ai programmi, catalogati di solito in base alla loro funzione primaria.

L'insieme di questi elementi, che possono essere divisi oppure appartenere ad uno stesso programma, viene solitamente riferito come window manager, poiché è questa la caratteristica al quale viene dato maggior risalto nella progettazione; sono quindi programmi scarni, il cui scopo è fornire una semplice interfaccia utente che dia accesso alle funzioni più elementari. Innumerevoli window manager sono stati sviluppati, ed alcuni di essi vengono ancora utilizzati in primo luogo per la loro leggerezza e velocità; alcuni dei più famosi sono Fvwm, BlackBox, FluxBox, IceWM ed XFCE. Da questo tipo di applicazioni si differenziano altri progetti, i quali, oltre ai canonici elementi di un window manager, aggiungono funzionalità avanzate: un gestore per barre dei menù, un supporto multilingua per le applicazioni, una interfaccia per l'accesso facilitato ai disabili, un gestore generico per i contenuti multimediali, diverse applicazioni di utilità e via discorrendo. Il risultato viene definito *ambiente desktop (desktop environment)*, il cui scopo prefissato è dare all'utente un grado di usabilità del sistema più elevata possibile e nel modo più semplice ed intuitivo, fornendo tutti gli strumenti necessari e uniformando il tutto con un aspetto gradevole. Tra i più famosi ed utilizzati ambienti desktop sono KDE e GNOME.

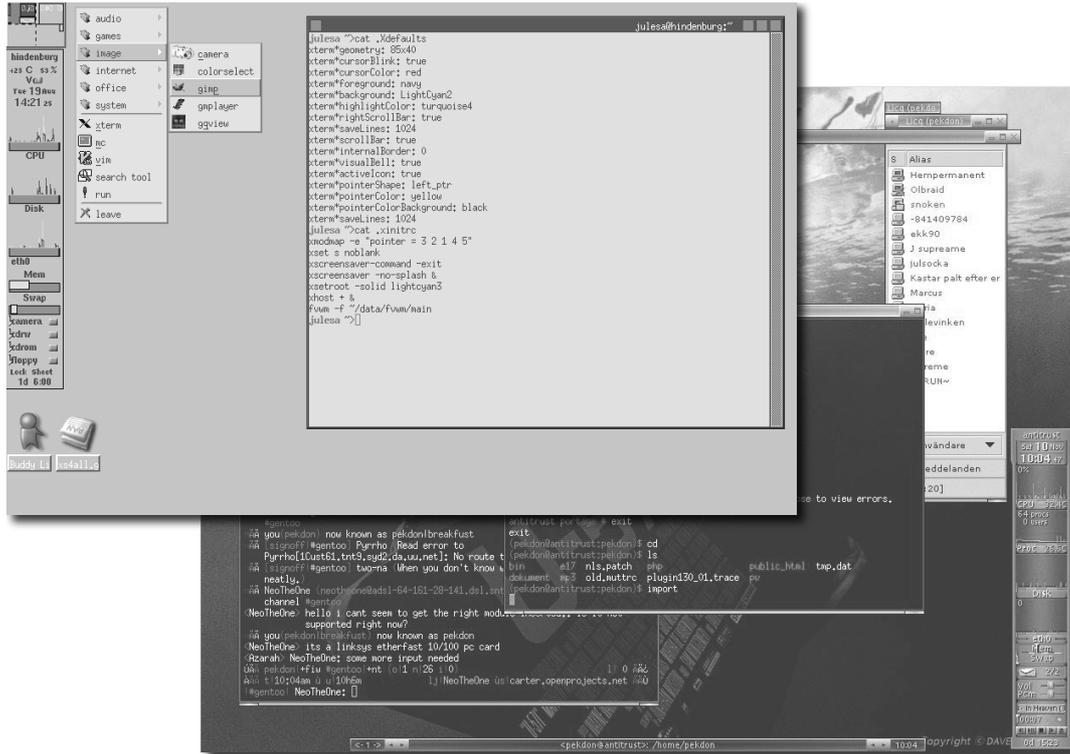


Immagine 3: i window manager Fvwm e FluxBox

K Desktop Environment - KDE

KDE è un progetto nato in Germania nel 1996, da una idea di Matthias Ettrich, il cui scopo era fornire i sistemi Unix di un vero ambiente desktop completo, al pari e migliore delle interfacce allora già esistenti per sistemi Windows e Macintosh.

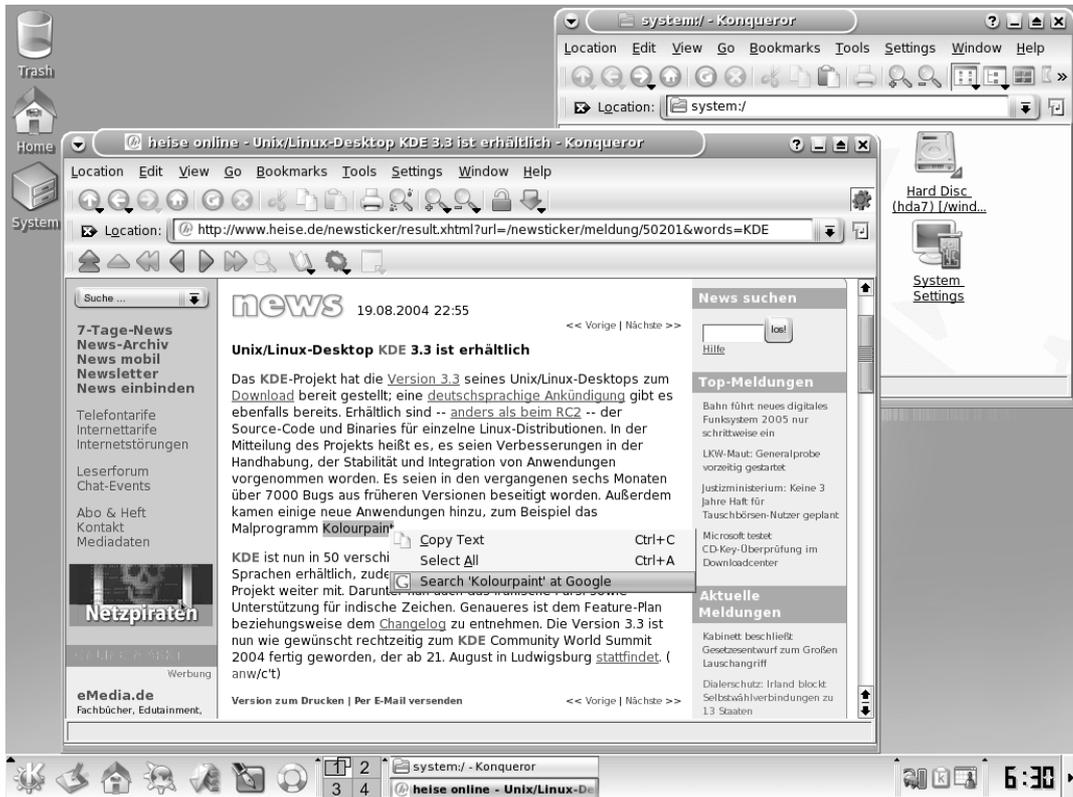


Immagine 4: il desktop KDE

Pur essendo relativamente giovane rispetto a tali concorrenti, le recenti versioni di questo sistema hanno colmato il divario temporale, integrando soluzioni già esistenti ad idee innovative, come ad esempio la possibilità di modificare in ogni parte lo stile grafico. Col passare del tempo lo scopo di KDE è stato allargato al fine di creare una piattaforma di sviluppo che rendesse la programmazione di nuove applicazioni veloce e con una interfaccia omogenea al resto del sistema; ad oggi KDE include svariate decine di applicazioni che coprono le esigenze di una utenza diversificata, come programmi multimediali, suite di sviluppo e per l'ufficio, giochi, programmi educativi e quant'altro, mentre molti altri progetti sono in via di sviluppo.

Il desktop di KDE si presenta in modo simile a quello di una comune versione di Windows, con una ampia barra delle applicazioni in fondo allo schermo. In fondo a sinistra della barra è situato il pulsante del menù principale, dove sono elencate le applicazioni disponibili classificate secondo il tipo di impiego. Il programma principale offerto da KDE è Konqueror, la cui funzione primaria è quella di browser generico: con Konqueror è possibile esplorare il contenuto del filesystem e delle condivisioni di rete di altri computer, visualizzare ogni tipo documento riconosciuto dal sistema, utilizzarlo come browser per Internet e molto altro. La configurazione di KDE si effettua tramite il Control Center, che raccoglie tutte le opzioni disponibili catalogate in modo opportuno; alcune di queste sono modificabili solo se si dispone dei privilegi dell'utente `root`.

GNU's Network Object Model Environment - GNOME

Il progetto GNOME nasce all'incirca un anno dopo KDE, del quale si presenta come antagonista, dall'iniziativa di Miguel de Icaza e Federico Mena. A differenza del suo predecessore, GNOME viene elaborato da subito come un progetto orientato alla creazione di una piattaforma di sviluppo per programmi facilmente integrabili nel sistema.

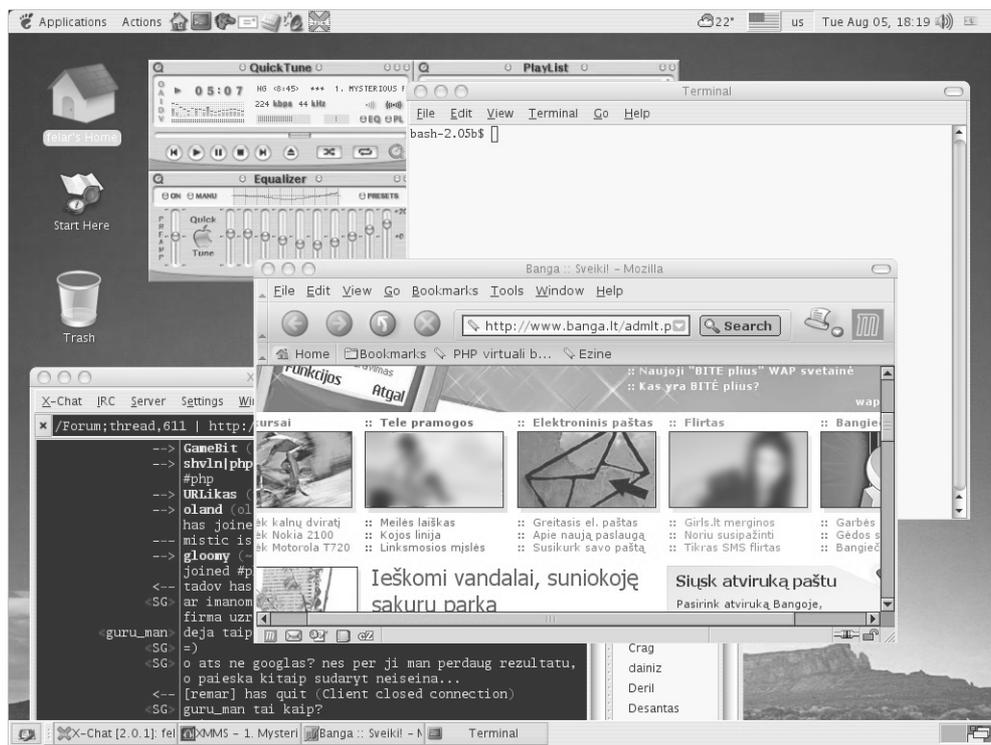


Immagine 5: il desktop GNOME

Una seconda differenza è data dal fatto che lo stesso progetto è diviso in molti sotto-progetti, i quali solitamente sono realizzati in modo da essere indipendenti e quindi riutilizzabili per altri scopi. Lo sviluppo di GNOME è oggi orientato alla creazione di uno standard per la massima usabilità dell'interfaccia ed all'accesso per utenti diversamente abili. Pur non essendo ancora al pari del suo antagonista, GNOME si propone come un desktop per tutti gli utilizzi, decisamente più leggero di KDE anche se ancora poco improntato all'interoperabilità tra le applicazioni.

Il desktop di GNOME presenta due barre delle applicazioni; la barra in alto mostra i menù delle applicazioni e delle azioni a sinistra, mentre la barra in basso è libera e di volta in volta è occupata dai programmi avviati. Il programma di esplorazione del filesystem e della rete è Nautilus, che gestisce anche le proprietà del desktop. Nautilus fornisce una visione globale del computer simile a quanto accade con il programma Risorse Del Computer di Windows, raggruppando in una finestra i diversi dispositivi ai quali è possibile accedere. Qualsiasi altra funzione è demandata a programmi specifici non correlati a Nautilus. GNOME non offre un programma unico per la configurazione del sistema; le opzioni configurabili si trovano all'interno del menù delle applicazioni raggruppate in due sezioni differenti, per gli utenti standard e per gli amministratori di sistema.

Programmi comuni per il lavoro in Unix

I comandi presentati nei capitoli precedenti sono un insieme minimo indispensabile per utilizzare al meglio le risorse del sistema. Ci occupiamo adesso di altri programmi ampiamente diffusi per il lavoro in Unix.

GNU Emacs

Emacs è l'editor per la programmazione più diffuso in ambiente Unix. È un programma concepito per esser estensibile in modo incredibilmente potente, al punto da poterne ridefinire gli stessi comandi e da riuscire ad integrare programmi di qualsiasi genere come un browser web o un programma di posta elettronica. Emacs nella sua versione attuale, 21.4, permette di lavorare sia da terminale che da X.

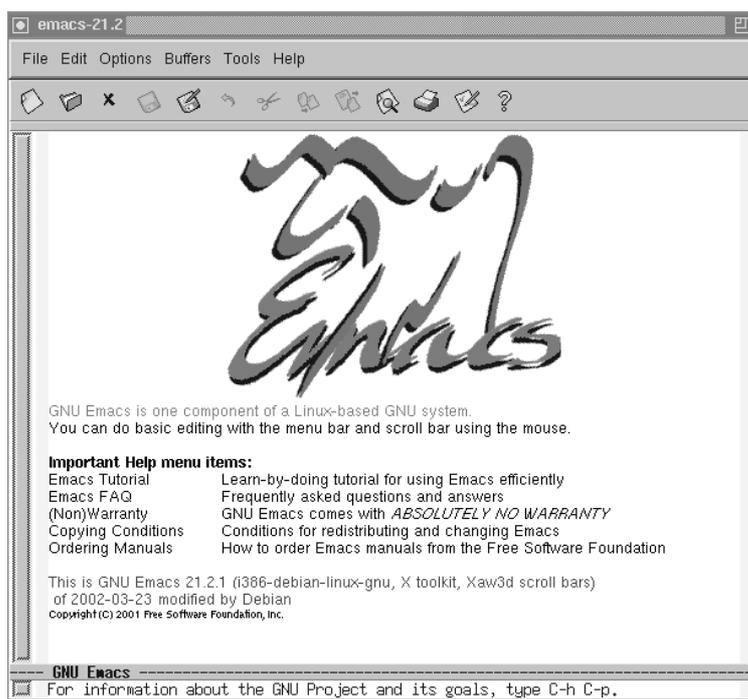


Immagine 6: GNU Emacs

Il principio di lavoro in Emacs è l'utilizzo di *buffer*. Un buffer è tipicamente un'area di memoria all'interno del computer, dove vengono caricati dati da utilizzare nel breve termine; in questo caso ogni buffer corrisponde ad una zona di lavoro a sé, dove solitamente viene caricato un file da modificare. In base al contenuto del buffer attivo – dove cioè è situato il cursore – il contenuto della barra principale dei menù cambia, aggiungendo alle voci generiche alcune opzioni specifiche.

All'avvio la finestra (o la schermata, a seconda dell'ambiente di lavoro) è suddivisa in due parti: l'area più grande è quella di lavoro, mentre la parte più piccola in basso è il *minibuffer di comando*, dove cioè vengono mostrati i comandi eseguiti dall'utente; sebbene l'utilizzo della grafica permetta

di lavorare per gran parte tramite il mouse, l'editor è stato sviluppato in precedenza per essere usato solo con la tastiera, utilizzando un insieme di combinazioni di tasti molto vasto, formato dai tasti CTRL o ALT seguiti da una lettera o un numero. Non è inusuale che le combinazioni richiedano la pressione consecutiva di più tasti lettera, come ad esempio per permettere il salvataggio di un file: la prima combinazione richiede la pressione di CTRL e x e consecutivamente CTRL e s; nella notazione di Emacs viene scritta nella forma CTRL-x CTRL-s. Nella Tabella 3 sono raccolte alcune delle principali combinazioni di tasti utilizzabili.

<i>Combinazione tasti</i>	<i>Funzione</i>
CTRL-x CTRL-f	Aprire il file dal nome specificato e carica il contenuto in un buffer; se il file non esiste crea un buffer vuoto
CTRL-x CTRL-s	Salva il buffer corrente nel file dello stesso nome; se il buffer non ha nome, ne va indicato uno
CTRL-x CTRL-w	Salva il buffer corrente con un altro nome
CTRL-x i	Inserisce nel buffer corrente il contenuto di un file specificato
CTRL-x k	Chiude il buffer dal nome specificato. Se nessun nome viene dato, il buffer chiuso è quello corrente
CTRL-s	Cerca una stringa di caratteri
CTRL-e	Sposta il cursore alla fine della riga corrente
CTRL-a	Sposta il cursore all'inizio della riga corrente
CTRL-v	Sposta la visuale del buffer corrente di una schermata in basso
ALT-v	Sposta la visuale del buffer corrente di una schermata in alto
CTRL-<spazio>	Delimita l'inizio di un blocco di testo
CTRL-w	Delimita la fine di un blocco di testo e lo taglia
ALT-w	Delimita la fine di un blocco di testo e lo copia
CTRL-y	Incolla il blocco di testo tagliato o copiato
CTRL-g	Interrompe l'immissione di qualsiasi combinazione di tasti
CTRL- <u> </u>	Annulla l'ultima operazione
CTRL-h	Aiuto in linea

Tabella 3: principali combinazioni di tasti di Emacs

Vi e Vim

Vi è un altro editor largamente diffuso. Nonostante esista una sua versione per X, viene utilizzato maggiormente da terminale. Attualmente esiste un programma clone, chiamato Vim, il quale implementa tutte le funzionalità dell'originale offrendone di nuove. Nonostante Vim introduca anche un comando di aiuto in linea per l'utente, l'utilizzo di questi programmi non è intuitivo e specialmente per quanti non hanno dimestichezza può risultare ostico e non è quindi consigliato come punto di partenza.

Nano/Pico

L'editor Pico è stato realizzato presso l'Università di Washington, allo scopo di ottenere un programma dal numero minimo di funzionalità in modo da risultare compatto e leggero; per questo motivo Pico non si presta molto bene come editor per lo sviluppo applicazioni, quanto alla semplice modifica di piccoli file di testo. In molti sistemi Linux Pico è stato sostituito da Nano, un clone che ne implementa le stesse funzionalità e modalità di utilizzo.

Nano/Pico è utilizzabile solo tramite terminale. Caratteristica principale di questo editor è quella di mostrare in fondo alla schermata tutte le azioni disponibili, ognuna delle quali è attivabili tramite la combinazione del tasto CTRL con un'altra lettera. Per convenzione le combinazioni di tasti disponibili sono mostrate utilizzando per primo il carattere ^, che per l'appunto descrive la pressione del tasto CTRL. È possibile lavorare solo con un documento alla volta. La Tabella 4 elenca i comandi della riga di fondo di Pico; Nano mette a disposizione anche altre funzioni, anche se molto raramente vengono utilizzate.

<i>Combinazione di tasti</i>	<i>Funzione</i>
^G	Aiuto in linea
^O	Salva le modifiche del file
^R	Legge dal file specificato
^V	Scorre di una schermata in avanti il documento corrente
^Y	Scorre di una schermata indietro il documento corrente
^K	Taglia il testo della linea in cui è posizionato il cursore
^C	Mostra la posizione del cursore all'interno del documento
^X	Esce dal programma
^J	Effettua la giustificazione del testo
^W	Cerca una stringa di caratteri
^U	Incolla il testo precedentemente tagliato
^T	Effettua la correzione ortografica del testo

Tabella 4: comandi di Nano/Pico

Programmi di compressione dati

Esistono diversi programmi che permettono di creare *archivi compressi*, file che contengono al loro interno uno o più elementi la cui dimensione è stata ridotta attraverso l'uso di algoritmi specifici; ogni programma di compressione ne utilizza uno a sé, producendo risultati diversi per lo stesso file. I più comuni sono descritti nella Tabella 5. Tra questi, l'unico che non effettua alcun tipo di compressione è tar, il quale viene utilizzato il più delle volte per raggruppare un insieme di file da passare poi ai programmi bzip2 o gzip; in questo caso viene utilizzata una estensione nata dalla concatenazione dei programmi utilizzati, come `.tar.bz2`, `.tar.gz` o `.tgz`.

<i>Programma di compressione</i>	<i>Estensione</i>	<i>Descrizione</i>	<i>Utilizzo</i>	<i>Programma di decompressione</i>
gzip	.gz	Comprime un unico file	Documenti di testo, piccoli archivi	gunzip
bzip2	.bz2	Comprime un unico file	File di grandi dimensioni	bunzip2
zip	.zip	Comprime una lista di file	Elementi di qualsiasi tipo	unzip
compress	.Z	Comprime un unico file	Elementi di qualsiasi tipo ma non eccessivamente grandi	uncompress
tar	.tar	Crea un file unico da una lista di file, senza comprimere	Backup di dati, creazione di archivi con gzip o bzip2	tar
rar	.rar	Comprime una lista di file	Elementi di qualsiasi tipo	unrar

Tabella 5: formati di archiviazione e compressione dati

Tutti i programmi descritti dalla tabella sono a riga di comando, quindi utilizzabili solo da terminale. I maggiori ambienti desktop offrono tra le altre applicazioni anche utility di compressione utilizzabili attraverso una interfaccia grafica, come ad esempio Kompres per KDE e File Roller per GNOME. Per i dettagli sull'utilizzo dei programmi citati, si faccia riferimento alle relative pagine di manuale.

Uso dei floppy disk con gli Mtools

I programmi dell'insieme Mtools sono stati sviluppati per permettere l'utilizzo di floppy disk da terminale evitando però di effettuarne il mount nel filesystem ed utilizzando una sintassi simile ai comandi del sistema operativo MS-DOS. Questi comandi si rendono necessari nel momento in cui l'amministratore di sistema non abbia permesso il mount di dispositivi rimovibili nella configurazione della macchina sul quale si sta lavorando. La sintassi degli Mtools è identica a quella dei corrispondenti comandi DOS; il nome dei comandi è invariato, con l'aggiunta di una m all'inizio di ogni nome: avremo così i comandi mdir, mformat, mcopy, mdel, mren e così via.

```

pippo@olivia:~$ mcopy lettera.txt a:
Copying LETTERA.TXT
pippo@olivia:~$ mdir a:
Volume in drive A has no label
Directory for A:

LETTERA  TXT          18639    4-29-94   2:57p
      1 File(s)      1438720 bytes free
pippo@olivia:~$

```

Comandi di stampa

La stampa nei sistemi Unix avviene tramite l'invio dei dati ad una *coda di stampa*, che raccoglie tutte le richieste degli utenti e le esegue dalla prima all'ultima. L'invio avviene tramite il comando

lpr, il cui parametro `-P` indica quale stampante riceverà il documento. La stampante del centro di calcolo viene identificata come `lj0`.

```
pippo@olivia:~$ lpr -Plj0 documentazione.ps
```

Ogni richiesta di stampa inviata è identificata da un id, ossia un valore numerico; tramite un id è possibile annullare una richiesta di stampa, ovviamente appartenente all'utente stesso, attraverso il programma `lprm`; attraverso il comando `lpq` si può conoscere lo stato della coda di stampa, visualizzando le informazioni sulle richieste pendenti, incluso il loro id numerico, l'utente che ha inviato la stampa, la dimensione del documento e la sua posizione nella coda di stampa. Anche questi due comandi prevedono l'opzione `-P`.

```
pippo@olivia:~$ lpq -Plj0
lj0 is ready and printing
Rank Owner Job File(s) Total Size
1st pippo 530 documentazione.ps 212 bytes
pippo@olivia:~$ lprm -Plj0 530
pippo@olivia:~$
```

Processi

Un programma correntemente in esecuzione in un sistema viene detto *processo*. Nei sistemi multitasking come Unix, in uno stesso istante sono presenti nel sistema molti processi che vengono eseguiti in modo virtualmente parallelo. L'invocazione di un programma attraverso il terminale coincide con la creazione di un processo, così come la sua attivazione attraverso l'ambiente grafico. Come per altri aspetti del sistema Unix, per ciascun processo vengono mantenute informazioni riguardanti l'utente che lo ha avviato ed il suo gruppo di appartenenza; di conseguenza ogni processo è soggetto a restrizioni più o meno limitanti rispetto all'utilizzo delle risorse, sia hardware (ad es. schede audio, video, dispositivi rimovibili) che software (accesso ad aree del filesystem). Fatta eccezione di `root`, l'utente comune ha il pieno controllo dei soli processi che gli appartengono. Unix assegna ad ogni processo avviato un numero identificativo, detto *process id* (o *pid*), in modo univoco.

L'invocazione di un programma, tramite interfaccia grafica o attraverso l'uso della shell, comporta la creazione di un nuovo processo. Nel caso della shell, questo solitamente implica che il terminale venga occupato dal processo per tutta la durata della sua esecuzione, di modo che l'utente possa interagire con esso; è il caso ad esempio del programma Nano, che visualizza una sua schermata di editing impedendo l'utilizzo della shell. Per evitare che il processo si trovi in questo stato, detto *foreground*, un programma può essere invocato aggiungendo alla fine della riga di comando il carattere `&`; in questo modo, verrà mandato in esecuzione in *background*, ovvero sarà creato un processo che sarà eseguito senza ottenere il controllo della finestra di terminale, permettendo all'utente di continuare il lavoro. L'esecuzione in background permette di utilizzare una unica shell per eseguire molteplici operazioni in contemporanea, siano esse lunghe o brevi. Nell'esempio che segue, viene avviato il programma di archiviazione tar per creare un archivio compresso di dati; supponendo che il procedimento richieda un certo periodo di tempo, viene avviato in background, di modo che il prompt della shell sia subito disponibile per l'utente mentre la compressione avanza.

```
pippo@olivia:~$ tar -zcf progetto.tar.gz progetto/ &
[1] 3618
pippo@olivia:~$ ls
 8 applicazione/          4 file_test.c          8 prova_finale.txt
pippo@olivia:~$
```

Quando un processo viene avviato in background, la shell mostra una serie di coppie di numerici: il primo viene assegnato dalla shell stessa ad ogni processo avviato in background, il secondo è il pid (identificatore di processo) dato dal sistema. Ovviamente può essere avviato più di un processo in background dallo stesso terminale, ogni volta che uno di questi terminerà verrà segnalato automaticamente dalla shell. Per conoscere in qualsiasi momento lo stato dei processi in background, è comunque possibile utilizzare il comando `jobs`.

```

pippo@olivia:~$ jobs
[1]+  Running                  yes >/dev/null &
[2]+  Running                  tar -zxvf documenti.tar.gz
pippo@olivia:~$

```

Di norma, un processo termina non appena ha compiuto il suo lavoro. Quando questo non accade si ha la possibilità di “uccidere”, cioè terminare prematuramente, i processi che non rispondono ai comandi dell'utente. A questo scopo vengono utilizzati i comandi `kill`, `killall` e `xkill`; il primo richiede la conoscenza del pid del processo da terminare, mentre il secondo uccide indistintamente tutti quei processi che eseguono uno stesso programma dal nome specificato. Diversamente, `xkill` permette la terminazione forzata di sole applicazioni in ambiente grafico, selezionabili attraverso un click del mouse. Ovviamente non è possibile per un utente comune uccidere processi non eseguiti da lui.

Per avere informazioni riguardanti i processi correntemente presenti nel sistema, sono utilizzabili i programmi `top` e `ps`. Il primo indica in tempo reale la situazione del sistema, mostrando una lista aggiornata dei processi ordinati secondo la percentuale di utilizzo del processore: `ps` fornisce invece l'elenco dei processi fornendo un gran numero di informazioni; il suo utilizzo è molto complesso e non verrà mostrato negli esempi, per cui rimandiamo alla consultazione della pagina di manuale associata.

```

pippo@olivia:~$ ps
  PID TTY          TIME CMD
 19455 pts/0    00:00:00 bash
 19460 pts/0    00:01:33 mozilla-bin
 19477 pts/0    00:00:00 gconfd-2
 19516 pts/0    00:00:00 run-mozilla.sh
 19529 pts/0    00:00:00 netstat <defunct>
 20141 pts/0    00:00:16 evince
 20424 pts/0    00:00:00 ps
pippo@olivia:~$

```

Parte terza: la rete Internet

Servizi di rete al CDC e limitazioni

L'accesso alla rete Internet tramite le macchine del Centro di Calcolo è soggetto ad alcune limitazioni al fine di evitare che vengano utilizzate per scopi non affini alla didattica. Per questo motivo la connessione verso l'esterno è filtrata attraverso l'uso di un *firewall*, un apparecchio preposto al controllo dei dati che viaggiano attraverso la rete. Di norma un firewall impedisce che utenti malevoli riescano ad entrare all'interno dei computer di una rete, chiudendo quindi l'accesso dall'esterno. Nel nostro caso, il firewall impedisce anche che dall'interno, cioè dalle macchine del Centro di Calcolo, si possa uscire per andare a fare danni all'esterno. Questi provvedimenti si sono resi necessari in quanto più volte sono state violate le norme di utilizzo della rete GARR, alla quale, ricordiamo, il CDC è collegato.

L'accesso alla rete è inoltre filtrato attraverso un *proxy server*, il cui scopo è limitare ulteriormente le possibilità di uscita a malintenzionati. I servizi web ed ftp raggiungono l'esterno passando "attraverso" il proxy, ovvero rivolgendosi ad esso per effettuare una connessione con server esterni.

Servizi di connessione remota

Le macchine del Centro di Calcolo sono sempre accesse, per permettere a chiunque sia abilitato di accedervi e lavorare in ogni momento della giornata. Quando non è possibile accedervi fisicamente, se si dispone di una connessione ad Internet possono essere utilizzate da remoto, ovvero tramite la connessione alle macchine attraverso la rete, similmente a quanto fanno programmi come il browser o il client di posta elettronica. Il Centro di Calcolo è predisposto per la connessione tramite *SSH/SCP* e *FTP*.

Shell sicura SSH

SSH (*secure shell*) è un servizio che permette di eseguire la shell di una macchina remota, permettendo quindi di usarne i programmi ivi presenti come se si avesse effettuato fisicamente il login. La caratteristica di questo servizio è la garanzia di sicurezza dei dati offerta: SSH codifica tutte le informazioni inviate attraverso la rete utilizzando particolari algoritmi di crittografia, impedendo a chiunque voglia intercettarli di utilizzarli in tempo utile. La connessione si effettua indicando semplicemente il nome dell'utente e quello della macchina di destinazione. Se è la prima volta che ci si connette a quel computer, verrà richiesto di confermare che l'host di destinazione è effettivamente quello desiderato.

```
pippo@home:~$ ssh pippo@olivia.cli.di.unipi.it
The authenticity of host 'olivia.cli.di.unipi.it (131.114.11.17)' can't be
established.
RSA key fingerprint is 71:9a:26:63:94:6b:cc:06:af:4b:1f:34:30:76:17:--.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'olivia.cli.di.unipi.it,131.114.11.17' (RSA) to the
list of known hosts.
Password:
pippo@olivia:~$
```

Per specificare il nome dell'utente è sufficiente anteporlo al nome della macchina, seguito dal carattere @, oppure usare l'opzione `-l`. Al termine del proprio lavoro la disconnessione si effettua similmente al logout dalla macchina, utilizzando i comandi `exit` o `logout`, oppure la combinazione di tasti CTRL e D.

L'utilizzo di SSH è possibile solo verso sistemi Unix/Linux. Tuttavia è comunque possibile collegarsi da una postazione con Windows, attraverso l'utilizzo di programmi come PuTTY. Anche in questo caso è sufficiente indicare il nome del computer al quale ci si connette ed indicare il modo di connessione tra le opzioni della sezione *Protocol* (in questo caso SSH): stabilita la connessione

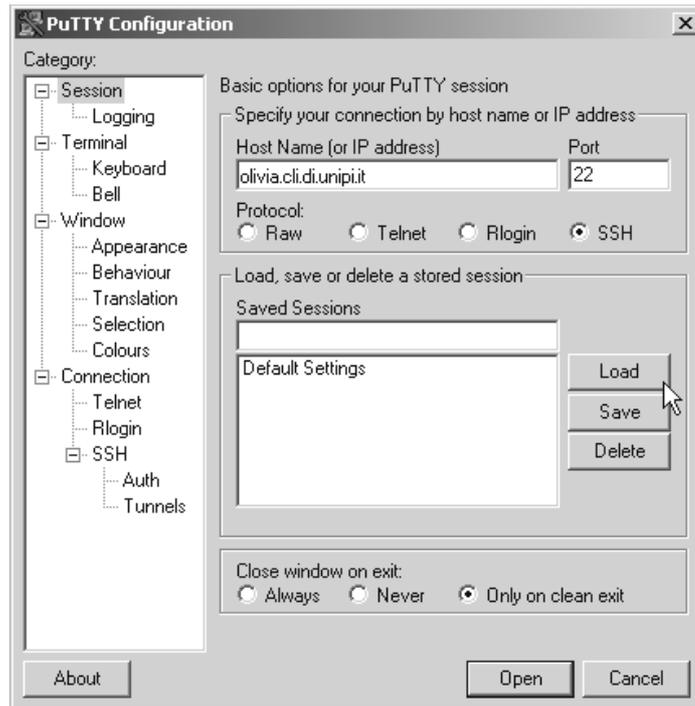


Immagine 7: finestra principale di PuTTY

PuTTY apre una finestra di pseudo-terminale, nella quale verrà mostrato l'ambiente di lavoro della shell.

Se ci si connette da un terminale in ambiente grafico, è possibile utilizzare SSH per effettuare un *forwarding* dell'output del server X della macchina remota: in pratica, le applicazioni per ambiente grafico vengono eseguite presso la macchina remota e visualizzate in locale, passando attraverso il canale protetto della connessione. Se il computer consente questa possibilità è possibile connettersi utilizzando l'opzione `-X` del programma.

Poiché le macchine del Centro di Calcolo sono predisposte per tale servizio, è **fortemente consigliato l'utilizzo di SSH**, in quanto permette di avere accesso praticamente a tutte le macchine del CDC senza doversi spostare fisicamente, sia dall'interno che dall'esterno del CDC.

Trasferimento file con SCP

Il programma SCP (*secure copy*) adotta lo stesso meccanismo di protezione utilizzato da SSH per copiare file da un computer all'altro. La sintassi ed i parametri per la sua invocazione sono a metà strada tra quelli di SSH e del comando `cp`. L'indirizzo remoto va specificato nella forma `<nome_utente>@<indirizzo.computer.remoto>:<path>`; se nessun path viene specificato viene considerata la directory home come punto iniziale. Il segno di due punti tra il nome della macchina ed il path è sempre necessario. Nell'esempio che segue, un file `foto.jpg` viene trasferito dal computer di un utente presso la sua home del Centro di Calcolo, connettendosi ad `olivia`. Successivamente, lo stesso file viene ricopiato dal server alla propria directory home locale.

```
pippo@home:~$ scp immagine.jpg pippo@olivia.cli.di.unipi.it:
pippo@olivia.cli.di.unipi.it's password:
foto.jpg                               100% 69KB 69.0KB/s 00:00
pippo@home:~$ scp pippo@olivia.cli.di.unipi.it:immagine.jpg ./
pippo@olivia.cli.di.unipi.it's password:
foto.jpg                               100% 69KB 69.0KB/s 00:00
pippo@home:~$
```

Come per SSH, esistono per Windows programmi che permettono l'uso di SCP, il nome del più noto è WinSCP, un programma basato su PuTTY e per questo molto simile nelle impostazioni.

Trasferimento file con FTP

Il servizio FTP (*file transfer protocol*) viene utilizzato per la condivisione e lo scambio di file generici, senza fornire però particolari misure di protezione dei dati. Il servizio è utilizzabile sia per effettuare connessioni verso la propria home presente nei server del centro di calcolo, sia per connettersi verso altri server FTP esterni. Anche l'utilizzo di programmi FTP da e verso il centro di calcolo è soggetto alle limitazioni del firewall. Per accedere ad un server esterno è necessario connettersi alla macchina `fire-int.cli.di.unipi.it`, indicando il nome utente desiderato seguito dall'indirizzo della macchina al quale connettersi. In questo esempio l'utente utilizza il programma a riga di comando `ftp` per effettuare una connessione verso il server `ftp.slackware.com`, effettuando il login come `anonymous`.

```
pippo@olivia:~$ ftp fire-int.cli.di.unipi.it
Connected to fire-int.cli.di.unipi.it.
220 fire-int FTP proxy (Version V2.1) ready.
Name (fire-int.cli.di.unipi.it:pippo): anonymous@ftp.slackware.com
331-(---GATEWAY CONNECTED TO ftp.slackware.com---)
331-(220 207.173.172.131 FTP server ready)
331 Anonymous login ok, send your complete email address as your password.
Password:
230-
*****
Welcome to ftp.slackware.com,

You are user 86 of 96.
*****
230 Anonymous access granted, restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

La connessione dall'esterno verso le macchine del centro di calcolo richiede un procedimento simile, se non che la connessione va effettuata presso la macchina `fire-ext.cli.di.unipi.it`; ovviamente l'utente deve avere un account presso il CDC.

```
pippo@home:~$ ftp fire-ext.cli.di.unipi.it
Connected to fire-ext.cli.di.unipi.it.
220 fire-int FTP proxy (Version V2.1) ready.
Name (fire-ext.cli.di.unipi.it:pippo): pippo@olivia
331-(---GATEWAY CONNECTED TO olivia---)
331-(220 olivia FTP server ready)
331 Password required for pippo.
Password:
230-
ftp>
```

La sintassi dei parametri per la connessione appena mostrata è applicabile anche all'utilizzo di applicazioni FTP a finestre.

Navigazione del web

Uno dei servizi più largamente utilizzati in Internet è sicuramente il World Wide Web, l'esplorazione di innumerevoli siti tramite l'uso di un programma *browser*. I browser principali installati al centro di calcolo sono Internet Explorer e Firefox per Windows, Mozilla, Galeon e Firefox per Linux e Safari per Mac. La navigazione è, come per altri servizi, limitata dal firewall ed utilizzabile impostando il browser.

La configurazione avviene in modo simile per tutti i browser sulle diverse piattaforme, come mostrato nell'Immagine 8. Tra le opzioni del programma vanno impostate quelle relative all'accesso a Internet.

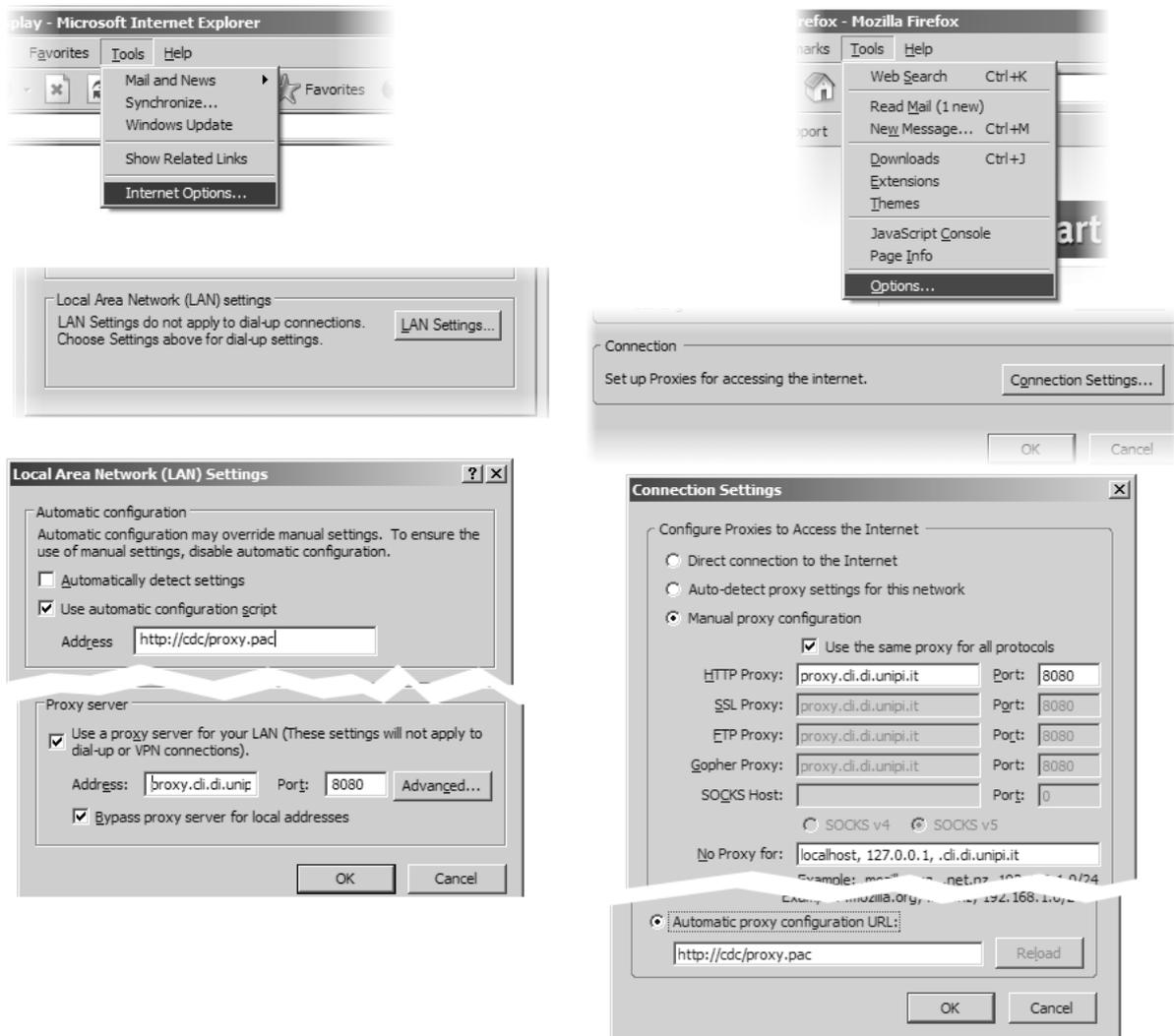


Immagine 8: configurazione di Internet Explorer e Firefox

- Per Internet Explorer, dal menù Tools selezionare Internet Options e la scheda Connections. Da qui, premendo il pulsante LAN Settings si accede configurazione.
- Per Firefox 1.0 (installato sui sistemi Windows), dal menù Tools selezionare Options e premere il pulsante Connection Settings.
- Per Firefox 1.5 (installato sui sistemi Linux), dal menù Edit selezionare Preferences, aprire la scheda General e premere il pulsante Connection Settings.
- Per Mozilla, dal menù Edit selezionare Preferences e, tra le opzioni del menù Advanced le impostazioni sono modificabili alla voce Proxies.

La connessione viene configurata in modo automatico attraverso l'uso di uno script di configurazione, raggiungibile all'indirizzo `http://cdc/proxy.pac`; è sufficiente impostare questo indirizzo nella caselle Automatic proxy configuration URL (o Use automatic configuration script) per fare sì che il browser recuperi le impostazioni in modo corretto. Questa procedura è **fortemente consigliata**, in quanto le impostazioni del browser vengono aggiornate in modo automatico ogni qualvolta la configurazione della rete del CDC debba variare.

Alternativamente alla procedura automatica, la configurazione manuale si imposta attualmente utilizzando i seguenti parametri: l'indirizzo del server proxy è `proxy.cli.di.unipi.it`, la porta da impostare è la 8080. Nella sezione No Proxy for va impostata la stringa `localhost, 127.0.0.1, .cli.di.unipi.it`; alternativamente va selezionata l'opzione Bypass proxy

server for local address; queste due ultime impostazioni permettono di non utilizzare il proxy per raggiungere le macchine interne alla rete del CDC, velocizzando quindi le operazioni.

Posta elettronica

Il funzionamento del servizio di posta elettronica, o email, si basa principalmente sull'utilizzo di *mailserver*, macchine che ricevono, smistano ed inviano i messaggi che gli utenti di tutto il mondo si scambiano. Quando una email viene inviata da un utente essa raggiunge ed attraversa uno o più mailserver, fino a raggiungere quello in cui è situata la *mailbox* del destinatario, ovvero lo spazio riservato per mantenere la posta. Il destinatario, a sua volta, si connette al proprio mailserver con un programma apposito chiamato *email client* e controlla il contenuto della propria mailbox.

I mailserver che mantengono le mailbox degli utenti possono essere acceduti tramite l'utilizzo di due diversi protocolli, chiamati POP e IMAP. Il protocollo POP viene utilizzato solitamente per permettere agli utenti di prelevare la posta in arrivo dal server e scaricarla sul proprio computer di casa. Il protocollo IMAP, più moderno e flessibile, è stato studiato per far sì che l'utente possa connettersi al server ed utilizzare efficacemente la propria mailbox senza il bisogno di scaricare tutti i messaggi, mantenendoli invece sul server ed quindi accedendovi ovunque si abbia la possibilità di utilizzare un client di posta relativamente moderno.

Il Centro di Calcolo mette a disposizione per ogni utente registrato un indirizzo di posta elettronica, assegnato in base al nome utente ottenuto al momento della registrazione. L'utente pippo avrà ad esempio un indirizzo pippo@cli.di.unipi.it, e così gli altri utenti, che avranno a disposizione un indirizzo email del tipo `<nome_utente>@cli.di.unipi.it`.

Programmi di lettura email

Nelle macchine del CDC sono presenti diversi programmi di lettura di posta elettronica, utilizzabili nei diversi sistemi. In ambiente Linux sono presenti i programmi Pine, Mozilla, Mozilla Thunderbird e Evolution; in Windows sono installati Mozilla Thunderbird e Outlook Express.

Utilizzo di Pine

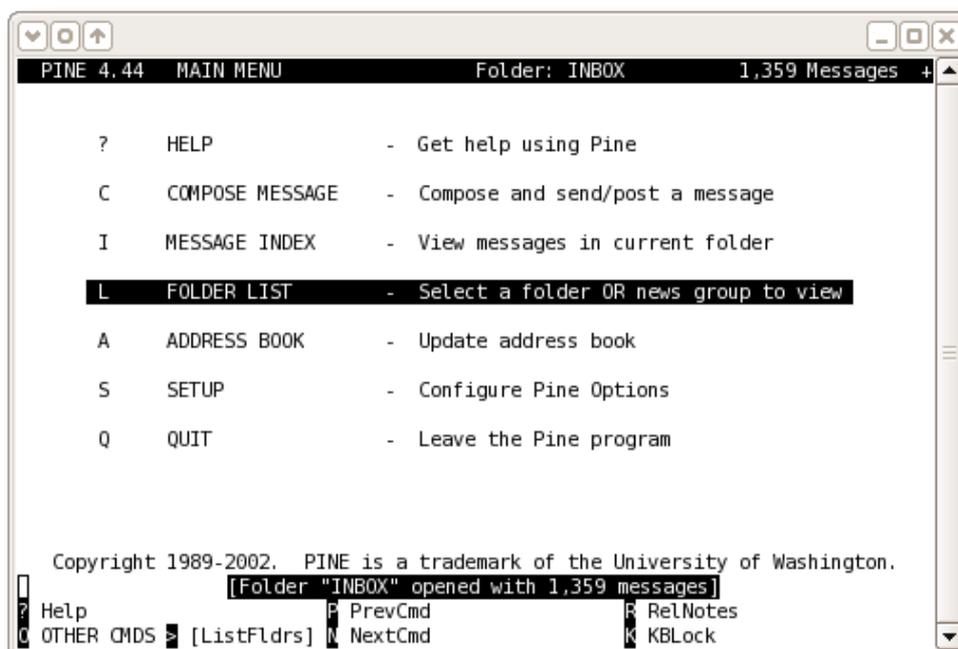


Immagine 9: schermata iniziale di Pine

Pine è un semplice client di posta elettronica, per Linux, utilizzabile da terminale. La versione installata al Centro di Calcolo è configurata e pronta da subito per l'utilizzo tramite il protocollo IMAP, evitando quindi di effettuare il download di messaggi dal server. È stato creato parallelamente all'editor Pico, difatti ne ricorda molto la struttura.

All'avvio dell'applicazione, Pine richiede l'immissione del proprio nome utente e password, quindi si connette al mailserver e, ad autenticazione effettuata, mostra il numero di messaggi presenti nella cartella INBOX della propria mailbox, in modo simile a quanto mostrato nell'Immagine 9. Le opzioni principali sono mostrate al centro della finestra di terminale, e raggiungibili tramite il carattere corrispondente visualizzato ad inizio di ogni opzione. Quando si accede ad una sezione tra quelle proposte, la schermata centrale cambia in funzione della selezione, mentre sono sempre visibili i comandi situati nelle due righe in fondo al terminale: Pine mostra in questo modo i principali comandi disponibili in base al contesto in cui ci si trova, selezionabili sempre attraverso il carattere evidenziato a sinistra; i restanti comandi sono raggiungibili attraverso l'opzione OTHER CMDS, premendo il tasto O. Tutte le funzioni di Pine sono spiegate nella sezione di aiuto, raggiungibile premendo il tasto ? .

Composizione ed invio di email

Tramite l'opzione COMPOSE MESSAGE si accede alla creazione di un nuovo messaggio email. Nella parte centrale della finestra di Pine appaiono i campi di intestazione e del corpo del messaggio. Nella parte dell'intestazione sono presenti i diversi campi che specificano il destinatario (o i destinatari) del messaggio, possibili file da allegare e l'oggetto dell'email, il titolo che viene visualizzato dal programma di posta a che descrive sinteticamente il contenuto del messaggio. I campi degli indirizzi (To: o CC:) possono contenere più di un indirizzo email, separati da una virgola. Il campo allegati (Attachmnt:) mostra tutti i file che sono stati selezionati attraverso l'opzione Attach, raggiungibile tramite la combinazione dei tasti CTRL e J (^J), e che verranno inviati all'interno della email in composizione.



Immagine 10: composizione messaggi con Pine

La parte del corpo del messaggio è la zona bianca situata al di sotto della riga Message Text. In questa zona viene scritto il messaggio vero e proprio, che sarà poi inviato in formato di solo testo. La modalità di scrittura è identica a quella dell'editor Pine. Al termine della composizione, è possibile inviare la mail (^X) o annullare l'operazione (^C).

Gestione della posta in arrivo

Il protocollo IMAP permette di gestire i messaggi contenuti nella propria mailbox in vari modi, come ad esempio catalogandoli utilizzando dei *folder*, dei contenitori simili in tutto e per tutto a delle directory. Il folder principale, presente da subito e contenente tutta la posta in arrivo, è INBOX. Successivamente diversi folder vengono creati automaticamente da Pine se ne ha bisogno, oppure dall'utente che desidera catalogare i messaggi. Il comando FOLDER LIST (L) permette la visualizzazione e l'accesso ai vari folder.

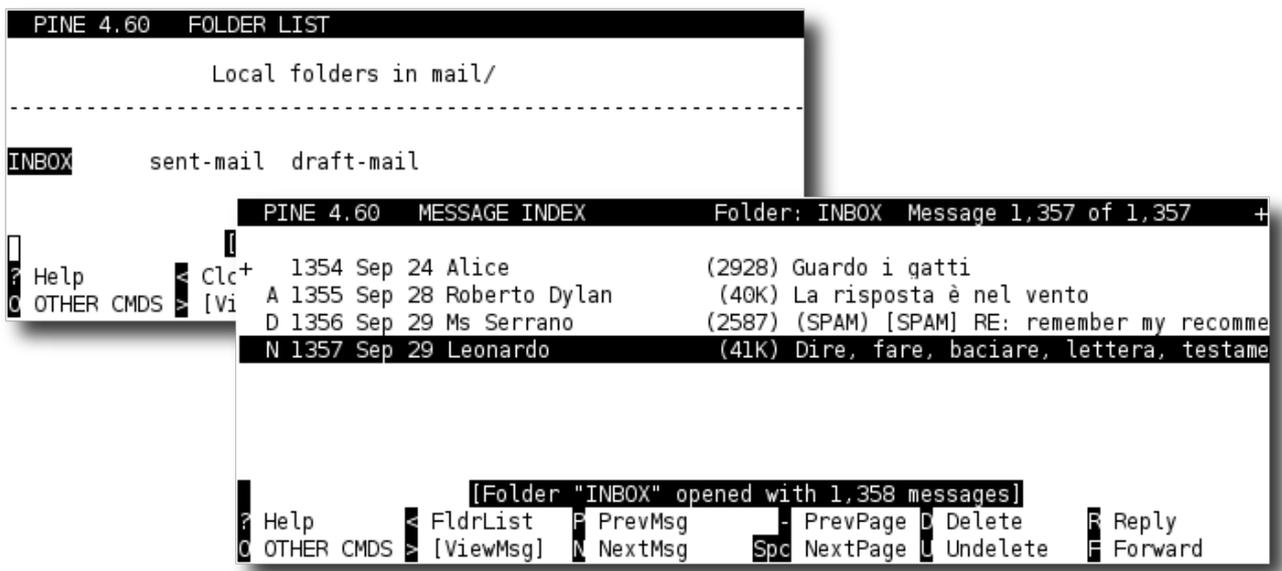


Immagine 11: accesso al folder INBOX ed elenco degli ultimi messaggi

Le email contenute in INBOX vengono elencate in ordine cronologico. Di ogni messaggio vengono specificati la data di invio, il mittente, la dimensione del messaggio e l'oggetto. Ogni email viene contrassegnata con un certo carattere, mostrato sul lato sinistro di ogni riga, che indica che il messaggio è nuovo (N), ha come destinatario principale l'utente in questione (+), è stato segnato come da cancellare (D), oppure è già stato inviato un messaggio di replica al mittente (A). Il messaggio evidenziato in nero come nell'Immagine 11 viene selezionato muovendo i tasti freccia attraverso l'elenco; su questo elemento è possibile operare utilizzando i vari comandi presentati in fondo alla schermata: ad esempio è possibile rispondere all'email (Reply, R), includendo il

messaggio originale nella risposta, Inoltrare (Forward, F) ad altri indirizzi il messaggio, segnare come da eliminare (Delete, D) o annullare l'eliminazione (Undelete, U) e altro ancora.

I messaggi segnati come da eliminare verranno cancellati dalla INBOX solo dopo una operazione chiamata *expunge*, effettuata su richiesta dell'utente tramite l'apposita opzione (X), oppure all'uscita del programma, al momento in cui Pine ne chiede conferma dell'operazione.

Può capitare di trovare nella propria INBOX alcuni messaggi il cui oggetto inizia con la parola SPAM racchiusa tra parentesi quadre o tonde: questo perché tutte le email che arrivano presso gli account del centro di calcolo sono filtrati durante il loro passaggio verso il mailserver, il quale controlla la provenienza di ogni email e segna quelle “sospette” come *spam*, messaggi pubblicitari non autorizzati ed indesiderati⁸. Generalmente i messaggi di spam possono essere tranquillamente eliminati, anche se è buona norma fare lo stesso attenzione sia al mittente che all'oggetto dell'elemento interessato.

Creare una rubrica indirizzi

Attraverso Pine è possibile gestire una rubrica di indirizzi email; in questo modo, al momento della composizione di un nuovo messaggio, è possibile specificare uno o più *nickname* (soprannomi), dei nomi mnemonici associati dall'utente ad alcuni indirizzi. Accedendo alla funzione ADDRESS BOOK (A) possono essere registrati i vari indirizzi che si conoscono. Le voci memorizzate potranno essere richiamate durante la scrittura di un messaggio attraverso la combinazione dei tasti CTRL e T (^T).

Utilizzo del servizio Web Mail

Il centro di calcolo permette la consultazione della propria mailbox anche attraverso una interfaccia web, ovvero una sezione del sito del CDC, consultabile quindi tramite il browser. Per usufruire di questo servizio è sufficiente collegarsi all'indirizzo <https://webmail.cli.di.unipi.it>. Qui verranno richiesti il nome utente e la password per accedere all'interfaccia di consultazione. Il servizio Web Mail è utilizzabile da qualsiasi macchina, esterna o interna al centro di calcolo, connessa ad Internet. Le opzioni viste precedentemente per Pine sono utilizzabili in modo simile anche con Web Mail.

Configurazione di altri programmi di posta

Pine e Web Mail permettono l'accesso alla propria mailbox in modo sufficientemente adeguato. È tuttavia possibile, se si vuole, utilizzare client di posta differenti impostando i parametri di connessione al mailserver del centro di calcolo⁹. Come per il proxy, queste configurazioni possono essere soggette a modifiche nel tempo, per cui i programmi utilizzati potrebbero non funzionare.

Sia per l'utilizzo con il protocollo IMAP che POP, l'indirizzo del server da utilizzare per la connessione è `mailserver.cli.di.unipi.it`. Per il protocollo POP la porta da utilizzare è la 110, mentre per IMAP 143. Se il client di posta supporta questa modalità è possibile utilizzare la connessione sicura SSL

È opportuno ricordare che se si vogliono utilizzare client di posta alternativi da computer esterni al CDC, **vanno prese in ogni caso misure di sicurezza**, come l'adozione di un antivirus o di un personal firewall, che impediscano la proliferazione di messaggi di spam o di worm attraverso la rete Internet utilizzando come punto di partenza il proprio computer.

8 I messaggi di spam non vengono cancellati, ma solo marcati come tali, in quanto il server non ha la certezza esatta di quale sia il contenuto delle email che sta filtrando, salvo per alcuni casi. Per questo motivo la macchina si permette di “suggerire” soltanto, lasciando all'utente la possibilità di verificare.

9 La spiegazione dettagliata delle impostazioni di altri client di posta esula dallo scopo di questo documento, pertanto si assume che il lettore abbia le competenze necessarie per l'utilizzo di tali programmi.

Parte quarta: aiuto ed indirizzi utili

Al termine di questa dispensa, che speriamo sia stata utile, ricordiamo i diversi modi possibili per chiedere aiuto e trovare informazioni per utilizzare nel migliore dei modi le strutture del centro di calcolo.

Anzitutto il sito del centro di calcolo <http://www.cli.di.unipi.it> offre tutte le informazioni possibili sull'utilizzo delle macchine e la configurazione dei programmi, informando costantemente su tutte le novità e le variazioni. Quando i contenuti del sito non sono esaustivi, oppure si vogliono segnalare problemi tecnici riguardanti software o programmi, è possibile contattare direttamente i sistemisti del CDC, inviando una email all'indirizzo sistemisti@cli.di.unipi.it. La segnalazione di problemi avviene indicando il maggiore numero di elementi possibili, quali ad esempio

- il nome della macchina su cui si è verificato il problema, visibile sia su un lato dei calcolatori che sul bordo dei monitor,
- il sistema operativo in uso al momento dell'errore, qualora si trattasse di un problema software,
- i messaggi di errore visualizzati,
- il malfunzionamento riscontrato, hardware o software,
- come riprodurre l'errore, ovvero cosa si stava facendo fino al momento in cui si è verificato il problema,
- qualsiasi altra informazione utile alla diagnosi e risoluzione del problema.

Per dubbi, informazioni e chiarimenti relativi all'utilizzo ed alla configurazione delle macchine, i membri del Gruppo Beatrice sono disponibili; è possibile consultare il sito <http://beatrice.cli.di.unipi.it>, controllando le ultime versioni di questa dispensa oppure partecipando alle discussioni del forum di supporto, oppure inviare una email all'indirizzo g_beatrice@cli.di.unipi.it o infine passare direttamente in sede.

Appendice A: Acceptable User Policy 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 riguardante 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, purché 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. È 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 esposte 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.
10. L'accesso alla rete GARR è condizionato all'accettazione integrale delle norme contenute in questo documento.