Introduzione Sistema Operativo UNIX

Antonio Capani

Genova, 9 Marzo 1994

1 Introduzione

Cosa è un sistema operativo?

Si può vedere un sistema operativo come la collezione dei *servizi* offerti all'utente da un calcolatore, più un'interfaccia per accedere a questi servizi.

Sono esempi di sistemi operativi: DOS, OSMac, SunOS, HP-UX, VAX/VMS.

Uno stesso calcolatore può ospitare diversi sistemi operativi, apparendo differente a seconda del sistema operativo in uso. Ad esempio si può installare su un PC sia il sistema WindowsNT che il DOS.

Interfaccia

L'accesso ai servizi offerti dal sistema operativo avviene attraverso un'interfaccia che può essere a testo (DOS) oppure grafica (WindowsNT). La linea di tendenza attuale consiste in una miscelazione di testo e grafica (SunOS, HP-UX, Linux).

Personal o Workstation

Il sistema operativo si adatta alle esigenze dell'utente o degli utenti del calcolatore. Il Macintosh, ad esempio, è concepito come *personal*, adoperato cioè da un solo utente. Il sistema operativo SunOS prevede invece l'utilizzo del calcolatore (*workstation*) da parte di più utenti. Naturalmente questa scelta comporta la progettazione di un sistema di accounting (per definire e limitare l'accesso al calcolatore) e la definizione di meccanismi di protezione dell'ambiente: ciascun utente può modificare solo il proprio ambiente di lavoro.

Condivisione del microprocessore

Le operazioni di Input/Output sono tipicamente molto costose, in confronto alle operazioni interne, che prevedono solo l'utilizzo della cpu. Così è particolarmente utile la condivisione del microprocessore da parte di più *processi*:

mentre il processo P aspetta la lettura di un certo dato dal disco, un altro processo Q può accedere alla cpu per calcolare $\sqrt{49}$.

Inoltre il meccanismo del *multi-processing* consente l'accesso contemporaneo al calcolatore da parte di più utenti collegati al sistema da diversi terminali.

Il sistema operativo Unix è *multi-utente* e *multi-processo*.

Quanti sono gli Unix?

Differenti case produttrici hanno esteso Unix in base alle proprie esigenze: nascono così, ad esempio, il sistema HP-UX (della Hewlett-Packard) e il sistema SunOS (della SUN).

Nel seguito verranno proposte caratteristiche prettamente Unix, indipendentemente dall'estensione.

Convenzione

Nel seguito si considererà l'esempio di una workstation Unix *Cartesio* a cui è collegato il Macintosh *MyMac*.

2 Accounting

Il sistema operativo Unix è *multi-utente*: prevede cioè l'utilizzo del calcolatore da parte di più utenti. Per accedere al sistema è necessario possedere un *account*, consistente in un nome (*user-name*) e in una *password*, cioè una parola segreta che garantisce l'esclusività dell'utilizzo dell'account.

Il gestore del sistema

Data la sua complessità, e considerati i problemi di protezione già accennati, il sistema operativo Unix prevede la figura del *System Manager*, un utente che, essendo dotato di particolari privilegi, può amministrare le risorse offerte dal sistema. Il System Manager ha la responsabilità del buon funzionamento del sistema: per questo motivo alcune operazioni, particolarmente pericolose, quali ad esempio la formattazione di un disco, sono consentite solo a lui.

Come accedere al sistema

Si può accedere al sistema direttamente dalla sua *console* (in parole povere: la tastiera e il video di Cartesio) oppure da un *terminale* connesso fisicamente al sistema (MyMac). Una volta collegati al sistema, appare una schermata del tipo:

```
SunOS UNIX (cartesio)
```

```
login:
```

L'utente digita il suo user-name (e preme RETURN). Poi inserisce la sua password (che non viene visualizzata).

```
SunOS UNIX (cartesio)

login: cap
Password:
```

Se la password è corretta l'utente entra nel sistema. Subito riceve dei messaggi introduttivi:

```
Last login: Tue Mar  1 15:04:37 from perceval
SunOS Release 4.1.3C (GENERIC+) #1: Thu Nov 25 17:18:48 MET 1993
You have new mail.
```

Nell'esempio di sopra il sistema ricorda l'ultimo accesso al sistema (in questo caso dal terminale "perceval") e la versione del sistema operativo. Inoltre rende noto all'utente che nel suo ambiente è presente della posta non ancora letta.

3 La shell

Una volta entrato nel sistema, l'utente comunica con questo tramite una interfaccia minimale, detta *shell*. La shell è un esempio di interfaccia a caratteri, molto utile quando si accede ad un sistema da un terminale non grafico (ad esempio un terminale remoto). Esistono diversi tipi di shell: Bourne Shell (sh), C Shell (csh, dalla sintassi molto simile al linguaggio di programmazione C), Korn Shell (ksh), ecc.

Il prompt

La shell presenta all'utente un *prompt*, cioè un simbolo che caratterizza lo stato di attesa di un comando.

Ad esempio cartesio offre all'utente cap il seguente prompt:

```
cartesio%
```

Struttura generale dei comandi

In corrispondenza del prompt, l'utente digita dei comandi (con parametri), la cui struttura generale è la seguente:

$$comando\ par_1 \ldots par_n$$

Ad esempio il seguente comando (senza parametri) visualizza i files contenuti nella *directory* corrente.

```
cartesio% ls
program    program.c  testo
```

Quest'altro esempio visualizza la lista dei soli files che iniziano con la lettera "p".

```
cartesio% ls p*
program    program.c
```

Alcuni parametri contengono delle opzioni; tipicamente le opzioni si distinguono dagli altri parametri del comando perché iniziano con il carattere "-".

L'esempio che segue visualizza i files che iniziano con "p", in formato "lungo" (più avanti sarà descritto in maniera specifica il comando "ls").

```
cartesio% ls -l p*
-rwxr-xr-x  1 cap         24576 Mar  3 12:33 program
-rw-r--r--  1 cap            31 Mar  3 12:33 program.c
```

Come uscire dalla shell

Per terminare una sessione si deve digitare il comando *exit*.

4 Il Filesystem

Struttura gerarchica del Filesystem

Il *Filesystem* di Unix presenta la classica struttura gerarchica a *directory* e *subdirectory*. Una directory corrisponde a quella che, secondo la terminologia a icone (alla Macintosh), viene detta *cartellina*. Il fatto che ogni directory possa a sua volta contenere altre directory produce la struttura gerarchica.

In alcuni sistemi operativi, come DOS e OSMac, l'utente accede ad un documento del Filesystem specificando l'unità (*device*) che contiene il documento. Se il file è contenuto in un dischetto, l'utente DOS deve tipicamente specificare l'unita A:, mentre l'utente Macintosh apre l'icona corrispondente al dischetto.

Il sistema operativo Unix, invece, prescinde dall'unità fisica che contiene il file, permettendo così una maggiore astrazione ed una maggiore flessibilità. Mediante il comando *mount* il sistema "monta" un'unità (es: dischetto) sul filesystem; in questo modo l'utente vede il dischetto come una qualunque cartellina.

Tutto il filesystem è contenuto in una speciale directory chiamata *root*.

Il path

Ogni documento è contenuto in una directory. Ciascuna directory, esclusa la *root*, è a sua volta contenuta in un'altra directory. La sequenza delle cartelline, dalla radice a quella che contiene il documento, unitamente al nome del documento stesso, individua unicamente il documento stesso. Questa sequenza è detta *path*: cioé il "percorso" da seguire, partendo dalla radice, per raggiungere il documento.

Mode, Owner, Group

In Unix una directory non è altro che un particolare file contenente la lista dei documenti presenti nella directory stessa, più alcune informazioni su ciascuno di essi.

Ad ogni file è associato lo user-name del *proprietario* e il nome del *gruppo* a cui questi appartiene [1].

Ad ogni file è associato un *mode*, cioè l'elenco dei diritti associati al file. Questi diritti sono divisi in tre categorie:

- Diritti del proprietario
- Diritti del gruppo
- Diritti degli altri utenti (non appartenenti al gruppo del proprietario)

I diritti sono di tre tipi: *lettura* (r), *scrittura* (w) ed *esecuzione* (x).

Vediamo un esempio:

```
cartesio% ls -lg
total 26
-rwxr-xr--  1 corsi    other       24576 Mar  5 12:26 program
-rw-r-----  1 corsi    other          31 Mar  5 12:26 program.c
-rw-r--r--  1 corsi    other         101 Mar  5 12:26 testo
```

La stringa -rwxr-xr– indica che il documento program è:

- leggibile, scrivibile ed eseguibile da parte del proprietario corsi
- leggibile ed eseguibile da parte del gruppo other
- solo leggibile da parte degli altri utenti

La directory corrente e le directory . e ..

Durante una sessione è sempre definita la cosiddetta *Current Working Directory, cwd*. Questa particolare directory si indica con ".". È possibile fare riferimento anche alla directory immediatamente precedente nel path (che contiene la directory corrente) con il simbolo "..".

[1] Ogni utente appartiene ad almeno un gruppo di utenti

5 Alcuni tra i comandi più usati

In questa sezione vengono presentati brevemente alcuni tra i comandi più utilizzati durante una sessione Unix.

pwd

Il comando *pwd* (Path Working Directory) visualizza il path della directory corrente.

```
cartesio% pwd
/home/cartesio/corsi
```

ls

Il comando *ls* (LiSt) visualizza il contenuto di una directory. Si tratta di uno dei comandi più utilizzati; presenta numerose opzioni. I prossimi esempi presentano utilizzi tipici del comando ls.

- lista semplice directory corrente

  ```
  cartesio% ls
  demo      textfile
  ```

- lista "lunga" directory corrente

  ```
  cartesio% ls -l
  total 1
  drwxr-xr-x  2 corsi         512 Mar  5 12:26 demo
  -rw-r--r--  1 corsi           0 Mar  5 12:25 textfile
  ```

  Si noti la prima lettera "d" nel modo del file demo. Indica che demo è una directory.

- lista di una directory (non corrente)

  ```
  cartesio% ls -l demo
  total 26
  -rwxr-xr--  1 corsi       24576 Mar  5 12:26 program
  -rw-r-----  1 corsi          31 Mar  5 12:26 program.c
  -rw-r--r--  1 corsi         101 Mar  5 12:26 testo
  ```

chmod

Il comando *chmod* consente di cambiare i diritti (*mode*) associati ad un file. Gli esempi che seguono dovrebbero chiarirne l'uso:

- Aggiunta del permesso di scrittura al gruppo (g=group)

  ```
  cartesio% ls -l program.c
  -rw-r-----  1 corsi          31 Mar  5 12:26 program.c
  cartesio% chmod g+w program.c
  cartesio% ls -l program.c
  -rw-rw----  1 corsi          31 Mar  5 12:26 program.c
  ```

- Rimozione del permesso di lettura al gruppo

  ```
  cartesio% chmod g-r program.c
  cartesio% ls -l program.c
  -rw--w----  1 corsi          31 Mar  5 12:26 program.c
  ```

- Aggiunta del permesso di lettura a tutti (a=all)

  ```
  cartesio% chmod a+r program.c
  cartesio% ls -l program.c
  -rw-rw-r--  1 corsi          31 Mar  5 12:26 program.c
  ```

Per cambiare i diritti del proprietario si usa la lettera "u" (user) (prima del + o del -), mentre per cambiare i diritti degli altri (terza terna) si usa la lettera "o" (other).

cd

Il comando *cd* (Change working Directory) cambia la directory corrente.

- ```
  cartesio% pwd
  /home/cartesio/corsi
  cartesio% cd demo
  cartesio% pwd
  /home/cartesio/corsi/demo
  ```

- ```
  cartesio% pwd
  /home/cartesio/corsi/demo
  cartesio% cd ..
  cartesio% pwd
  /home/cartesio/corsi
  ```

  Si noti l'uso della directory "..".

- ritorno alla propria home-directory

```
cartesio% pwd
/home/cartesio/corsi/demo
cartesio% cd
cartesio% pwd
/home/cartesio/corsi
```

cp

Il comando *cp* (CoPy) serve a copiare files.

- copia di un file nella directory corrente

```
cartesio% ls
demo      textfile
cartesio% cp textfile textfile_copy
cartesio% ls -F
demo/          textfile       textfile_copy
```

  ([2])

- copia di files in una diversa directory, ultimo parametro del comando

```
cartesio% ls demo
program    program.c  testo
cartesio% cp textfile textfile_copy demo
cartesio% ls demo
program     program.c   testo      textfile    textfile_copy
```

- copia di files da una directory a quella corrente

```
cartesio% ls
demo           textfile       textfile_copy
cartesio% cp demo/testo .
cartesio% ls
demo        testo       textfile    textfile_copy
```

  Si noti che è necessario specificare la directory corrente "." (a differenza di altri sistemi operativi, quali il DOS).

[2]L'opzione -F del comando ls, dà indicazioni sul tipo di file appendendo al nome il carattere "/" se è una directory, "*" se è un file eseguibile.

rm

Il comando *rm* (ReMove) rimuove uno o più files.

```
cartesio% ls demo
program      program.c    testo        textfile     textfile_copy
cartesio% rm textfile_copy
cartesio% rm -i demo/*
rm: remove demo/program? n
rm: remove demo/program.c? n
rm: remove demo/testo? n
rm: remove demo/textfile? y
rm: remove demo/textfile_copy? y
cartesio% ls demo
program    program.c  testo
```

L'opzione -i (Interactive) provoca da parte del sistema una richiesta di conferma della volontà di cancellare.

mkdir

Il comando *mkdir* (MaKe DIRectories) crea una o più directories.

```
cartesio% mkdir d
cartesio% ls -l
total 3
drwxr-xr-x  2 corsi         512 Mar  5 16:47 d
drwxr-xr-x  2 corsi         512 Mar  5 16:37 demo
-rw-r--r--  1 corsi         101 Mar  5 16:33 testo
-rw-r--r--  1 corsi           0 Mar  5 12:25 textfile
```

mv

Il comando *mv* (MoVe files) muove uno o più files.

- ```
  cartesio% mv testo d
  cartesio% ls . d
  .:
  d         demo      textfile

  d:
  testo
  ```
- Cambio nome ad un file
  ```

```
cartesio% mv d/testo d/TESTO
cartesio% ls d
TESTO
```

Si noti che Unix distingue le lettere maiuscole da quelle minuscole (a differenza del DOS).

rmdir

Il comando *rmdir* (ReMove DIRectories) viene utilizzato per cancellare delle directory.

```
cartesio% rmdir d
rmdir: d: Directory not empty
cartesio% rm d/TESTO
cartesio% rmdir d
cartesio% ls
demo      textfile
```

Si noti il messaggio di errore.

cat

Il comando *cat* (conCATenate and display files) applicato ad una lista di files provoca la visualizzazione di questi sullo schermo, nella sequenza della lista.

```
cartesio% cat textfile demo/program.c
Questo e' un file di testo...
... ma forse si era capito.

Antonio
main()
{
  printf("Ciao\n");
}
```

more

Quando il testo è troppo lungo per essere visualizzato in una sola schermata è opportuno utilizzare il comando *more*.

```
cartesio% more demo/testo
Questo testo contiene le seguenti parole:

Questo
```

```
testo
contiene
le
seguenti
parole
Questo
testo
contiene
le
seguenti
parole
Questo
testo
contiene
--More--(48%)
```

Al prompt "More" è indicata la percentuale del documento, visualizzata fino ad ora. Premendo il tasto "ENTER" si avanza di una riga. Premendo lo spazio si avanza di una pagina. Premendo il tasto "q" (oppure control-C) si interrompe la visualizzazione.

lpr

Il comando *lpr* provoca la stampa dei documenti elencati. È possibile specificare la stampante su cui operare.

```
cartesio% lpr textfile demo/testo
cartesio% lpr -Plw textfile
```

Il secondo comando opera sulla stampante "lw".

ps

Il comando *ps* (Process Status) visualizza lo stato dei processi correnti.

```
cartesio% ps
  PID TT STAT  TIME COMMAND
 2204 p0 S     0:00 -csh (csh)
 2294 p0 R     0:00 ps
```

PID (Process IDentifier) è il numero che identifica il processo. Ad ogni nuovo processo il sistema assegna un nuovo PID (in progressivo). *TT* indica il terminale che controlla il processo. *STAT* indica lo stato del processo. *TIME* indica il tempo di *cpu* utilizzato, finora, dal processo. *COMMAND* dà indicazioni sul comando che ha creato il processo.

passwd

Il comando *passwd* consente di cambiare la propria password.

```
cartesio% passwd
Changing password for corsi on cartesio.
Old password:
New password:
Retype new password:
```

Come si può notare la password non viene mai visualizzata (per ragioni di sicurezza).

Il sistema tiene nota delle password di ciascun utente, registrandole criptate in un opportuno file. Spesso il file delle password è condiviso tra più macchine, attraverso uno speciale servizio di rete, detto Network Information Service (NIS). In questo caso, al posto del comando *passwd* si deve usare *yppasswd*.

6 Ambiente di lavoro

Ciascun utente può personalizzare il proprio ambiente di lavoro a seconda delle proprie esigenze.

Variabili di ambiente

L'ambiente di lavoro è caratterizzato da alcune variabili di ambiente, visualizzabili con il comando *env*

```
cartesio% env
HOME=/home/cartesio/corsi
SHELL=/usr/bin/csh
TERM=sun-cmd
USER=corsi
PATH=/bin:/usr/bin:/usr/ucb:/etc:.
LOGNAME=corsi
PWD=/home/cartesio/corsi/demo
```

La variabile di ambiente PATH indica i percorsi (separati da ":") dove cercare file eseguibili. Ad esempio se ci troviamo in /home/cartesio/corsi e vogliamo eseguire il programma "program" presente nella directory demo, abbiamo bisogno di specificarne il path, a meno che non modifichiamo in maniera opportuna la variabile di ambiente PATH.

```
cartesio% program
program: Command not found.
cartesio% demo/program
```

```
Ciao
cartesio% setenv PATH /home/cartesio/corsi/demo:$PATH
cartesio% echo $PATH
/home/cartesio/corsi/demo:bin:/usr/bin:/usr/ucb:/etc:.
cartesio% program
Ciao
```

Il "$" serve per accedere al contenuto della variabile. Il comando *echo* provoca la stampa di una stringa.

Alias

In alcune *shell* (ad esempio la *C shell*) è possibile definire degli *alias*, cioè dei comandi personali basati sul proprio utilizzo del calcolatore.

```
cartesio% alias rm rm -i
cartesio% rm textfile
rm: remove textfile? n
cartesio% alias md mkdir
cartesio% alias
md      mkdir
rm      (rm -i)
```

Nel esempio il comando *rm* viene ridefinito forzando l'opzione "-i": questo alias è particolarmente utile per gli utenti "distratti". Con gli alias è possibile definire delle abbreviazioni per i comandi più utilizzati. Chiamando *alias* senza parametri si ottiene la lista degli alias definiti.

Per rimuovere un alias si usa il comando *unalias*.

```
cartesio% unalias md
cartesio% alias
cancella        rm
rm      (rm -i)
```

File di Startup

Una volta terminata la sessione si perdono le variabili di ambiente e gli alias. In questo caso è opportuno registrare le proprie preferenze nel file **.login**, presente nella propria *home-directory*. I comandi contenuti in questo file vengono eseguiti ogniqualvolta si apre una sessione.

```
cartesio% more .login
# File di startup per il seminario su Unix

/usr/5bin/banner Benvenuti al seminario
```

```
stty -istrip
stty erase ^H

alias rm rm -i
alias md mkdir
```

7 Come arrangiarsi

Data la vastità del sistema Unix, non è conveniente o sensato memorizzare tutti i comandi forniti. Né, tantomeno, è necessario imparare la moltitudine di opzioni offerte da alcuni comandi.

Manuale on-line

Unix fornisce all'utente un manuale in linea. Questo manuale si può richiamare per mezzo del comando *man*, seguito dall'argomento che si vuole esaminare (tipicamente il nome di un comando).

```
cartesio% man man
Reformatting page.  Wait... done

MAN(1)                  USER COMMANDS                  MAN(1)

NAME
     man - display reference manual pages; find  reference  pages
     by keyword

SYNOPSIS
     man [-] [-t] [-M path] [-T macro-package]  [[section]  title
     ...] ...
     man [-M path] -k keyword ...
     man [-M path] -f filename ...

DESCRIPTION
     man displays information from the reference manuals.  It can
     display  complete  manual pages that you select by title, or
     one-line summaries selected either by keyword  (-k),  or  by
     the name of an associated file (-f).

...
```

Le pagine di manuale contengono spesso dei riferimenti ad altri argomenti e alle sezioni del manuale in cui sono descritti. Tipicamente viene visualizzato un opportuno paragrafo denominato "SEE ALSO" nel quale vengono elencati gli argomenti esaminabili con man;

Ecco il "SEE ALSO" che appare con man man.

```
SEE ALSO
     apropos(1),  cat(1V),  col(1V),  eqn(1),  lpr(1),   more(1),
     nroff(1),  refer(1), tbl(1), troff(1), vgrind(1), vtroff(1),
     whatis(1), eqnchar(7), man(7), catman(8)
```

Come si può notare c'è un riferimento a man(7). Questo significa che nella sezione 7 delle pagine di manuale esiste un argomento denominato man.

Per esaminare man(7) è necessario indicare esplicitamente la sezione 7.

```
cartesio% man 7 man
Reformatting page.  Wait... done

MAN(7)       ENVIRONMENTS, TABLES, AND TROFF MACROS       MAN(7)

NAME
     man - macros to format Reference Manual pages

SYNOPSIS
     nroff -man filename...

     troff -man filename...
...
```

whatis

Il comando *whatis* visualizza solo una breve descrizione del comando.

```
cartesio% whatis whatis
whatis (1)       - display a one-line summary about a keyword
```

man -k

A volte capita di sapere che esiste un comando per accedere ad un certo servizio, ma non ci si ricorda quale è questo comando. In questo caso viene in aiuto l'opzione "-k" del comando man; questa opzione consente di effettuare ricerche su tutte le pagine di manuale, sulla base di una parola che si ritiene attinente con il problema che vogliamo risolvere.

L'esempio che segue dovrebbe chiarirne l'uso.

```
cartesio% man -k printer
banner (6)       - print large banner on printer
```

```
ext_ports (5)      - external ports file for network printers,
terminals, and modems
lp, cancel (1)     - send/cancel requests to a printer
lpc (8)            - line printer control program
lpd (8)            - printer daemon
lpq (1)            - display the queue of printer jobs
lpr (1)            - send a job to the printer
lprm (1)           - remove jobs from the printer queue
lpstat (1)         - display the printer status information
lptest (1)         - generate lineprinter ripple pattern
nroff (1)          - format documents for display or line-printer
pac (8)            - printer/plotter accounting information
pp (4)             - Centronics-compatible parallel printer port
printcap (5)       - printer capability data base
vpc (4S)           - Systech VPC-2200 Versatec printer/plotter and
Centronics printer interface
vplot (1)          - plot graphics for a Versatec printer
ypprintcap (5)     - NIS printer capability database
```

8 Reti di calcolatori

Più calcolatori possono colloquiare tra di loro attraverso una connessione fisica. Si dice che questi calcolatori sono “in rete”. Le reti si distinguono a seconda delle modalità con cui avviene la comunicazione. La più importante rete conosciuta è la Internet.

Indirizzi

Ciascuna macchina (*Host*) connessa alla rete Internet possiede un indirizzo che la caratterizza univocamente. Questo indirizzo, detto *IP Address*, consiste in una sequenza di quattro numeri separati da “.”, come 147.31.254.130. All’indirizzo numerico corrisponde una sequenza alfanumerica detta *FQDN, Full Qualified Domain Name*. E’ possibile utilizzare sia l’indirizzo numerico, sia quello alfanumerico.

Ad esempio la macchina *cartesio* ha l’indirizzo numerico **130.251.60.3** e l’indirizzo alfanumerico **cartesio.dima.unige.it**.

Più macchine possono essere raggruppate in sottoreti, costituendo un *dominio*. Ad esempio tutte le macchine del DIMA appartengono al dominio **dima.unige.it**, il quale a sua volta è un sottodominio della rete universitaria genovese **unige.it** e della rete nazionale **it**.

9 Il servizio della Posta

Uno dei più interessanti servizi offerti dalla connessione in rete è sicuramente quello della Posta. Ogni utente che abbia un account su una macchina connessa in rete può spedire delle lettere ad un altro utente (anch'esso sulla rete).

Indirizzo

Per spedire una lettera è necessario specificare l'indirizzo del destinatario. L'indirizzo si compone di due parti: lo *user-name* del destinatario e l'indirizzo Internet della macchina su cui questi opera.

L'utente *corsi* della macchina **cartesio.dima.unige.it** viene identificato con l'indirizzo **corsi@cartesio.dima.unige.it**.

Spesso si vuole prescindere dalla specifica macchina sulla quale opera l'utente: in questo caso si può indirizzare la lettera specificando solo il nome dell'utente ed dominio della macchina su cui lavora (es: **corsi@dima.unige.it**). In questo caso una delle macchine del dominio **dima.unige.it** (detta *MAILHOST*), si occuperà di consegnare la lettera alla macchina opportuna.

Lettura della posta

Per leggere la posta da una workstation Unix si usa il comando *Mail*.

```
cartesio% Mail
Mail version SMI 4.0 Fri Jul  2 11:55:02 PDT 1993 Type ? for help
"/usr/spool/mail/corsi": 1 message 1 new
>N  1 cap@lancelot.dima.unige.it Sun Mar  6 19:27  14/560   prova
& 1
Message  1:
From cap@lancelot.dima.unige.it Sun Mar  6 19:27:13 1994
Received: from lancelot.dima.unige.it by dima.unige.it with
SMTP id AA03010
  (5.65c/IDA-1.4.4 for <corsi@dima.unige.it>); Sun, 6 Mar 1994
 19:27:12 +0100
Received: by lancelot.dima.unige.it (5.0/SMI-SVR4)
        id AA13691; Sun, 6 Mar 1994 19:26:53 --100
Date: Sun, 6 Mar 1994 19:26:53 --100
From: cap@lancelot.dima.unige.it (Antonio CAPANI)
Message-Id: <9403061826.AA13691@lancelot.dima.unige.it>
To: corsi@dima.unige.it
Subject: prova
Content-Length: 51
Status: R
```

```
Questa lettera e' una prova per il seminario Unix.

& q
Saved 1 message in /home/cartesio/corsi/mbox
cartesio% Mail
No mail for corsi
```

Dopo avere digitato il comando *Mail* si riceve l'elenco della posta presente nella cassetta.

In questo caso l'utente corsi ha ricevuto una lettera dall'utente cap della macchina **lancelot.dima.unige.it**.

Al prompt "&" l'utente ha battuto "1" per visualizzare il messaggio numero 1 (in questo caso l'unico presente nella cassetta).

Con il comando "q" (che sta per *quit*) ha concluso la visione della posta provocandone il salvataggio nel file /home/cartesio/corsi/mbox.

In questo modo la lettera è stata rimossa dalla cassetta della posta, (come si vede dal successivo tentativo di esecuzione di Mail).

A volte si desidera mantenere la posta nella cassetta, per poterla rileggere più tardi: in questo caso si deve uscire da Mail usando il comando "x" (exit).

Guida rapida di Mail

Mail possiede una piccola guida rapida, richiamabile attraverso il comando *help* (al prompt del Mail).

```
& help
                mailx commands

alias,group user ...   declare alias for user names
alternates user        declare alternate names for your login
cd,chdir [directory]   chdir to directory or home if none given
!command               shell escape
delete [msglist]       delete messages
...
```

Scrittura di una lettera

Il comando *Mail*, seguito dall'indirizzo di un utente, consente l'invio di una lettera a quest'ultimo.

```
cartesio% Mail cap@dima.unige.it
Subject: prova scrittura posta
Questa lettera arrivera' (spero) all'utente cap del dominio
dima.unige.it.
.
```

```
EOT
```

Si noti il punto ("*.*") a inizio linea, con il quale l'utente termina la scrittura della lettera.

9.1 Forwarding

A volte uno stesso utente possiede più di un account. Tipicamente, però, possiede un account preferenziale per quanto riguarda la gestione della posta. Le lettere che giungono su una macchina A possono essere "proseguite" su una seconda macchina B, scrivendo il file *.forward* nella propria *home-directory*.

Ad esempio l'utente cap fà proseguire le sue lettere sulla macchina di indirizzo **lancelot.dima.unige.it**.

```
cartesio% more .forward
cap@lancelot.dima.unige.it
```

10 Altri servizi di rete

Quello della posta non è l'unico servizio offerto dal *networking*.

telnet

Il comando *telnet* permette di aprire una sessione con un altra macchina, sulla quale si possiede un account.

```
cartesio% telnet lancelot.dima.unige.it
Trying 130.251.60.16 ...
Connected to lancelot.dima.unige.it.
Escape character is '^]'.


UNIX(r) System V Release 4.0 (lancelot)

login: cap
Password:
Last login: Sun Mar  6 17:53:47 on console
Sun Microsystems Inc.   SunOS 5.3       Generic September 1993
You have mail.
lancelot%
```

ftp

Il comando *ftp* (File Transfer Protocol) consente il trasferimento di files da una macchina da un'altra.

```
cartesio% ftp lancelot.dima.unige.it
Connected to lancelot.dima.unige.it.
220 lancelot FTP server (UNIX(r) System V Release 4.0) ready.
Name (lancelot.dima.unige.it:corsi): cap
331 Password required for cap.
Password:
230 User cap logged in.
ftp> get documento
200 PORT command successful.
150 ASCII data connection for documento (130.251.60.3,1344) (50
bytes).
226 ASCII Transfer complete.
local: documento remote: documento
50 bytes received in 0.0041 seconds (12 Kbytes/s)
ftp> quit
221 Goodbye.
cartesio%
```

In questo esempio l'utente "corsi" si è collegato all'account dell'utente "cap" (del quale conosce la password) ed ha prelevato con il comando *get* il file "documento".

talk

Un comando particolarmente interessante è *talk*. Il comando talk permette a due utenti di colloquiare sulla rete utilizzando il proprio terminale in modo bidirezionale. In sostanza è possibile "telefonare" ad un utente utilizzando video e tastiera.

```
cartesio% talk cap@lancelot.dima.unige.it
```

In questo modo l'utente *corsi* ha inoltrato a lancelot.dima.unige.it una richiesta di talk con l'utente cap. L'utente cap (se collegato) riceve un messaggio del tipo:

```
Message from Talk_Daemon@lancelot at 10:06 ...
talk: connection requested by corsi@cartesio.dima.unige.it
talk: respond with:  talk corsi@cartesio.dima.unige.it
```

Se cap intende rispondere alla chiamata, digiterà a sua volta:

```
lancelot% talk corsi@cartesio.dima.unige.it
```

A questo punto entrambi i terminali degli utenti si dividono in due parti: sulla parte superiore si scrive, su quella inferiore si legge. È utile sottolineare che la lettura e scrittura possono avvenire contemporaneamente.

```
[Connection established]
Ciao cap, sono corsi: sto sperimentando il talk.





-----------------------------------------------------------------
Salve a te.

Buona prosecuzione.
```

Premendo CONTROL-C si termina la "telefonata".

finger

Il comando *finger* viene usato per chiedere informazioni su un utente di una certa macchina.

```
lancelot% finger corsi@cartesio.dima.unige.it
[cartesio.dima.unige.it]
Login name: corsi
Directory: /home/cartesio/corsi      Shell: /usr/bin/csh
On since Mar  7 08:58:42 on ttyp0 from lancelot
20 seconds Idle Time
No unread mail
No Plan.
```