

Accesso alle risorse del laboratorio informatico del Dipartimento di Matematica

Marco Liverani*

26 settembre 2012

Il laboratorio informatico del Dipartimento di Matematica mette a disposizione degli studenti del nostro corso di laurea una serie di postazioni di lavoro costituite da personal computer di ultima generazione dotati di doppio sistema operativo: Microsoft Windows XP e Linux (distribuzione “Red Hat Fedora”). Durante le esercitazioni e gli incontri di tutorato del corso di Informatica 1 (IN110) si utilizzerà esclusivamente il sistema operativo Linux.

Per maggiori informazioni sull’uso dei comandi della shell UNIX disponibile sul sistema operativo Linux, si può fare riferimento alle dispense disponibili in rete all’indirizzo <http://www.mat.uniroma3.it/users/liverani/doc/manuale-unix.pdf>.

Nelle pagine seguenti sono descritte in modo sintetico alcune procedure pratiche per l’utilizzo delle infrastrutture informatiche del Laboratorio per lo svolgimento di esercizi ed altre attività pratiche inerenti il corso di Informatica 1. In particolare viene descritto il modo con cui è possibile collegarsi in modalità di emulazione di terminale al server del laboratorio per effettuare esercitazioni pratiche di programmazione in linguaggio C da una postazione di lavoro esterna al Dipartimento, connessa alla rete Internet (es.: un personal computer collegato ad Internet con una linea ADSL dalla propria abitazione).

1 Account utente

Per poter utilizzare le postazioni di lavoro presenti in laboratorio è necessario richiedere l’attivazione di un *account* Linux ai responsabili del laboratorio didattico (Dott.ssa Tiziana Manfroni, Sig. Andrea Angelini, presso la stanza attigua al laboratorio informatico).

L’*account* è costituito da uno *username* pubblico e da una *password* da mantenere rigorosamente segreta. Tutte le attività svolte sulle macchine del laboratorio sono registrate su file di log e quindi è possibile risalire all’identità dell’utente che ha compiuto eventuali operazioni improprie o fraudolente mediante le apparecchiature del laboratorio. È bene quindi che ogni studente mantenga riservata la propria password al fine di evitare che altri possano usare il suo account sulle macchine del laboratorio.

La password deve essere modificata da ciascun utente utilizzando il comando UNIX `yppasswd` su una qualsiasi postazione Linux presente in laboratorio; il sistema chiederà di inserire prima la vecchia password e poi, per due volte, la nuova password. Da quel momento in poi la password dell’utente risulterà modificata su tutte le macchine del laboratorio. Attenzione: la password in ambiente UNIX è *case sensitive*, il che significa che il sistema considera diverse fra loro le lettere maiuscole e quelle minuscole.

Dopo aver effettuato il *login* sulla postazione di lavoro in ambiente Linux, utilizzando il proprio *username* e la propria password segreta, è necessario aprire un programma in cui sia eseguita la *shell* dei comandi UNIX, in modo da operare in modalità interattiva

* Università Roma Tre, Corso di Laurea in Matematica, Corso di Informatica Generale 1 (IN110); e-mail liverani@mat.uniroma3.it; ultima modifica: 26 settembre 2012

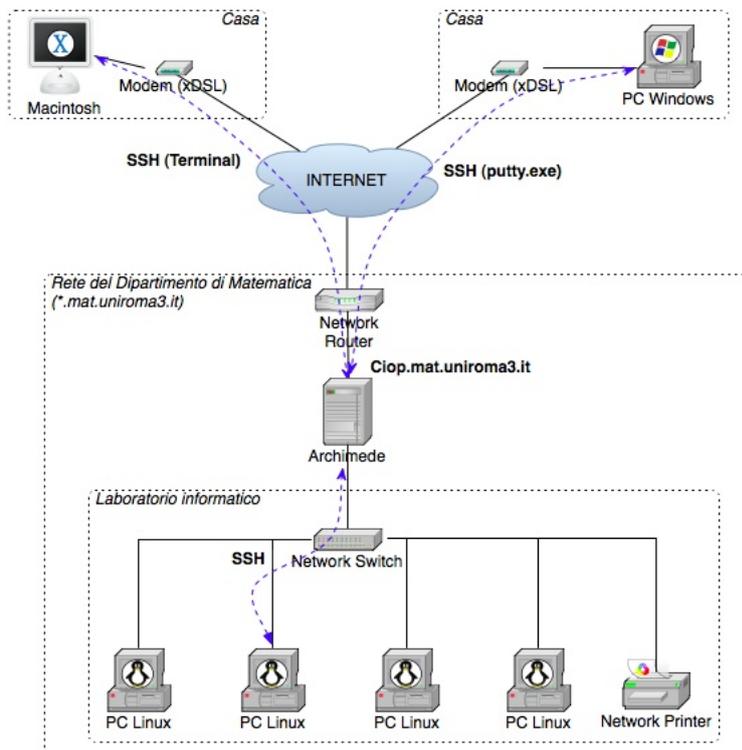


Figura 1: Struttura macroscopica della rete del laboratorio informatico

con il sistema. Tramite la *shell* è possibile impartire i comandi al sistema operativo digitandoli sulla tastiera. L'output di tali comandi sarà presentato dal sistema nella finestra in cui è eseguita la *shell*. In ambiente grafico X11/KDE i programmi più diffusi con cui eseguire una *shell* di comandi sono *xterm*, *kterm* e *konsole*.

2 Collegamento al server del dipartimento

Le macchine Linux del laboratorio informatico sono collegate ad un server centrale denominato *ciop*. Sul server sono presenti le *home directory* di tutti gli utenti del laboratorio, condivise in rete mediante il protocollo NFS (*network file system*); in questo modo ciascuno studente può utilizzare indifferente una qualunque postazione di lavoro in laboratorio, disponendo sul filesystem la propria home directory con tutti i file di lavoro; la memorizzazione dei file non avviene quindi "localmente", ma su un filesystem del server *ciop* condiviso in rete: non è necessario quindi copiare i file da una postazione di lavoro ad un'altra.

Lavorando sulle postazioni del laboratorio si utilizzano le risorse di calcolo "locali" (la CPU e la memoria RAM della postazione di lavoro) ed il filesystem (il disco) del server "remoto". È anche possibile, tuttavia, collegarsi direttamente al server per sfruttare direttamente le sue risorse di calcolo (CPU e memoria, oltre al filesystem). In questo caso la postazione di lavoro viene utilizzata soltanto come "terminale" per l'utilizzo di una macchina remota. Si usa dire che la sessione di lavoro viene effettuata "da remoto" utilizzando un protocollo ed un programma di emulazione di terminale alfanumerico.

È possibile connettersi da una delle postazioni di lavoro allo stesso server *ciop* utilizzando il comando UNIX *ssh* dalla *shell* dei comandi; vediamo di seguito un esempio di sessione di lavoro:

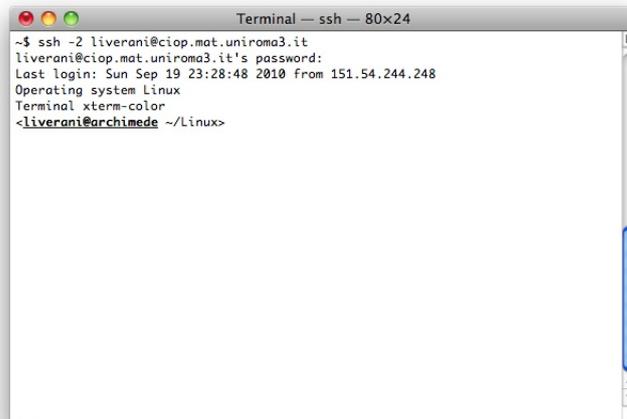


Figura 2: Connessione al server del dipartimento mediante il programma ssh attivato in una finestra di terminale su un computer Apple Macintosh

```

~$ ssh -2 mrossi@ciop.mat.uniroma3.it
mrossi@ciop.mat.uniroma3.it's password:
Last login: Sun Sep 30 12:14:27 2007 from adsl97.fastweb.it
ATTENZIONE:
**1**. le mail possono essere lette esclusivamente da ciop.
**2**. il cambiamento della password puo' essere effettuato
con il comando: ypasswd "nomeutente"
**3**. gli utenti possono connettersi a ciop.mat.uniroma3.it da casa
con il protocollo ssh
<mrossi@archimede ~>

```

Il collegamento al server del laboratorio informatico del dipartimento può essere effettuato anche da casa, sfruttando una connessione Internet (modem su linea analogica, ADSL, ecc.) ed il protocollo SSH (*secure shell*) che consente di operare da remoto sulla shell del sistema operativo UNIX in modalità sicura (il traffico di rete durante la sessione di lavoro viene criptato). In ambiente Linux o Mac OS X è presente il comando ssh, che può essere eseguito da una finestra di terminale. Su computer Apple Macintosh in ambiente Mac OS X il programma da utilizzare si trova tra le Utility e si chiama “Terminal” (vedi figura 2).

In ambiente Microsoft Windows è necessario installare un programma come ad esempio putty, distribuito gratuitamente su Internet al seguente indirizzo:

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Per connettersi via ssh ad un server remoto (come il server del laboratorio) è necessario specificare lo *username* con cui vogliamo aprire la sessione di lavoro sul server e l'indirizzo di rete del server stesso. Dall'interno della rete del laboratorio il server può essere raggiunto con l'indirizzo archimede (non ciop). In figura 1 si riassume la struttura della rete del laboratorio informatico del dipartimento.

Il programma Putty in ambiente Microsoft Windows può essere salvato in una qualsiasi directory del PC (es.: c:\Programmi\putty.exe) e può essere eseguito effettuando un “doppio click” sull'icona del programma. Una volta avviato il programma viene presentata una finestra di dialogo in cui è possibile specificare l'indirizzo del server a cui ci si intende collegare (nel nostro caso ciop.mat.uniroma3.it – vedi Figura 3. Per

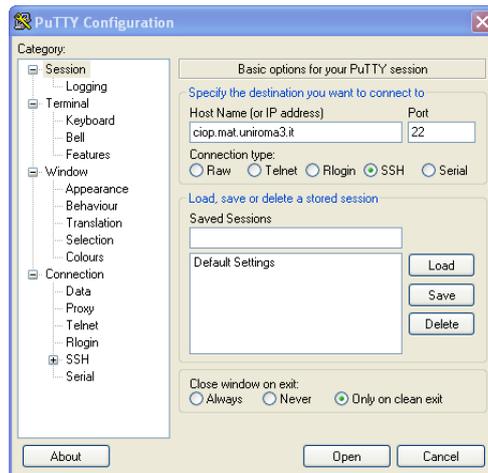


Figura 3: La finestra di dialogo del programma Putty con cui specificare l'indirizzo del server a cui si intende collegarsi in emulazione di terminale attraverso il protocollo SSH



Figura 4: La finestra di dialogo del programma Putty con cui viene chiesta conferma della connessione con il server remoto

attivare la sessione di lavoro in emulazione di terminale bisogna selezionare il bottone "Open".

La prima volta che viene eseguita una connessione devono essere scambiate delle informazioni tra il client e il server per definire le chiavi con cui saranno criptate le comunicazioni successive. Per questo motivo il programma Putty chiede di confermare la connessione con il server specificato nella finestra di dialogo (vedi Figura 4); tale richiesta di conferma avviene solo alla prima connessione con il server.

Infine viene aperta una finestra del programma di emulazione di terminale e viene stabilita la connessione con il server (vedi Figura 5). L'utente potrà accedere al sistema remoto specificando il proprio *username* e la propria *password* segreta.

Operando con una *shell* remota direttamente sul server *ciop/archimede*, ogni utente può utilizzare un servizio di posta elettronica mediante il programma *pine* che può essere lanciato anche nell'ambito di una connessione in emulazione di terminale via *ssh*; a ciascun utente è assegnato un indirizzo di posta elettronica Internet del tipo "*username@ciop.mat.uniroma3.it*".

Per terminare la sessione di lavoro e chiudere la shell dei comandi si può utilizzare il comando *logout*. Prima di lasciare la postazione di lavoro in laboratorio è necessario effettuare la procedura di *logout* riportando il sistema alla maschera di *login* per consentire l'accesso ad un altro utente.

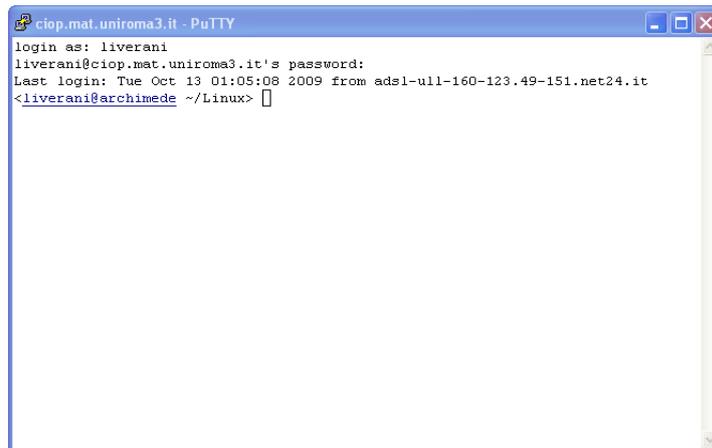


Figura 5: La finestra di emulazione di terminale alfanumerico all'interno della quale viene eseguita la sessione di lavoro sul server remoto

3 Editing, compilazione ed esecuzione di programmi C

Nell'ambito del corso di Informatica 1 (IN110) il laboratorio viene utilizzato prevalentemente per esercitarsi nella codifica in linguaggio C di algoritmi risolutivi per alcuni problemi piuttosto elementari su array, matrici, alberi e grafi. È necessario quindi disporre di un ambiente di sviluppo dotato di un editor di testo per la scrittura dei programmi sorgente in linguaggio C, di un compilatore C per la traduzione del programma da codice sorgente a codice eseguibile (in linguaggio macchina) ed infine di una *shell* per l'esecuzione dei programmi compilati.

In ambiente Linux, sulle postazioni di lavoro del laboratorio, potranno essere utilizzati, tra gli altri, i seguenti programmi per digitare il codice sorgente in linguaggio C e salvare il programma su un file in formato *plain text*:

Emacs È un potente editor di testo disponibile in ambiente UNIX sia in modalità alfanumerica (“a carattere”) che in modalità grafica; per utilizzare questo programma per creare da zero o modificare il file “programma.c” è sufficiente utilizzare il comando “emacs programma.c”; per salvare il file modificato sul filesystem si deve utilizzare la sequenza di tasti Ctrl-x, Ctrl-s; per uscire dall'editor si deve utilizzare la sequenza di tasti Ctrl-x, Ctrl-c.

Kedit È un editor di testo disponibile nell'ambito dell'ambiente grafico KDE; per lanciare kedit per modificare o creare il file “programma.c” si deve utilizzare una *shell* di comandi in ambiente grafico KDE, come xterm, kterm o konsole e si deve utilizzare il comando “kedit programma.c &”. Per uscire dal programma basta chiudere la finestra o scegliere la voce “Quit” dal menù “File”.

Pico È un editor di testo molto semplice che può essere utilizzato da un terminale alfanumerico; è lo stesso editor utilizzato dal programma pine per consentire la scrittura di messaggi di posta elettronica. Per lanciare il programma per modificare o creare il file “programma.c” si deve utilizzare il comando “pico programma.c”. Per salvare il file modificato si deve utilizzare la sequenza di tasti Ctrl-o, mentre per uscire dal programma basta digitare Ctrl-x.

Per la traduzione del programma da codice sorgente (in linguaggio C) a codice eseguibile (in linguaggio macchina), in ambiente Linux è disponibile il compilatore GCC (*GNU C Compiler*). Per compilare il programma sorgente “esercizio.c” producendo in output il programma eseguibile “esercizio” si deve utilizzare il seguente comando:

```
gcc -Wall esercizio.c -o esercizio
```

L'opzione “-Wall” consente di visualizzare tutti (all) i messaggi di *warning* (-W) prodotti dal compilatore e non soltanto i messaggi di errore; questo è utile, ad esempio, per segnalare eventuali variabili inutilizzate nel programma.

L'opzione “-o” consente di specificare il nome del file eseguibile su cui verrà registrato il programma in linguaggio macchina; nel caso in cui venga omessa l'opzione “-o” ed il nome del file di *output* il programma eseguibile verrà registrato nel file “a.out”.

Se il compilatore GCC termina la traduzione del programma senza visualizzare alcun messaggio sul terminale dell'utente, allora vorrà dire che la traduzione è avvenuta correttamente, senza alcun errore; in caso contrario saranno visualizzati sullo schermo del terminale una serie di messaggi di errore; è sempre conveniente correggere gli errori a cominciare dal primo, ricompilando il programma dopo aver corretto ciascun errore.

Per lanciare in esecuzione un programma compilato correttamente e senza errori mediante gcc, è sufficiente digitare il nome del file eseguibile seguito dal tasto Enter; il nome dell'eseguibile deve essere preceduto dal *path* completo del file, espresso in forma assoluta o relativa; ad esempio per lanciare il programma eseguibile “esercizio” salvato nella directory “/studenti/mrossi” si deve digitare il comando:

```
/studenti/mrossi/esercizio
```

oppure, se la directory corrente è proprio “/studenti/mrossi”, si può digitare il seguente comando, esprimendo il *path* in forma relativa:

```
./esercizio
```

Per interrompere l'esecuzione di un programma all'interno di una *shell* di comandi è sufficiente battere i tasti Ctrl-c.

4 Principali comandi UNIX

Di seguito viene proposta una sintesi dei principali comandi della *shell* del sistema operativo UNIX.

4.1 Gestione di file e directory

Con i comandi riportati nella tabella seguente è possibile visualizzare il contenuto delle directory del filesystem del sistema e muoversi all'interno della struttura di directory e sottodirectory; è possibile anche creare e cancellare file e directory e visualizzare il contenuto di file di testo.

Comando	Sintassi	Descrizione	Esempio
ls	ls [-latF] [<i>file</i>]	Visualizza la lista dei file (ls = <i>list</i>).	ls -lF
cp	cp [-r] <i>origine</i> <i>destinazione</i>	Copia (cp = <i>copy</i>) uno o più file in un altro file o directory.	cp pippo.c pluto.c
mv	mv <i>origine</i> <i>destinazione</i>	Sposta (mv = <i>move</i>) uno o più file da una posizione ad un'altra; può essere usato anche per cambiare nome ad un file.	mv *.c sorgenti
rm	rm [-r] <i>file</i>	Cancella uno o più file (rm = <i>remove</i>).	rm pippo.*

Comando	Sintassi	Descrizione	Esempio
mkdir	mkdir <i>directory</i>	Crea una nuova directory (mkdir = <i>make directory</i>).	mkdir sorgenti
rmdir	rmdir [-r] <i>directory</i>	Cancella una directory (rmdir = <i>remove directory</i>).	rmdir sorgenti
cd	cd [<i>directory</i>]	Cambia la directory corrente (cd = <i>change directory</i>); senza argomento torna nella <i>home directory</i> .	cd ../sorgenti
pwd	pwd	Visualizza il <i>path</i> della directory corrente (pwd = <i>print work directory</i>).	pwd
cat	cat <i>file</i>	Visualizza il contenuto di un file di testo.	cat pippo.c
more	more <i>file</i>	Visualizza (impaginandolo) il contenuto di un file di testo. Si usa il tasto Return per visualizzare la riga successiva ed il tasto Spazio per visualizzare la pagina successiva e il tasto Q per interrompere la visualizzazione (<i>quit</i>).	more pippo.c
chmod	chmod <i>modalità file</i>	Cambia i permessi di accesso al file (chmod = <i>change mode</i>).	chmod 640 file.c
chown	chown <i>username:group file</i>	Cambia il proprietario di un file (chown = <i>change owner</i>).	chown bianchi:user file.c
pico	pico <i>file</i>	Esegue il programma di videoscrittura pico per creare o modificare il contenuto del file di testo.	pico pippo.c

4.2 Collegamento in rete e trasferimento file

I comandi riportati nella tabella seguente servono ad attivare sessioni di lavoro da remoto su un sistema UNIX e trasferire file da un sistema ad un altro (o da un sistema UNIX alla propria postazione di lavoro) utilizzando una rete TCP/IP.

Comando	Sintassi	Descrizione	Esempio
ssh	ssh [-2] [-X] <i>user@hostname</i>	Attiva una sessione di lavoro in modalità sicura su un sistema remoto.	ssh -2 rossi@ciop
ftp	ftp <i>hostname</i>	Attiva un collegamento su protocollo FTP (<i>file transfer protocol</i>) per trasferire file da un host remoto al sistema locale o viceversa.	ftp ftp.mat.uniroma1.it
scp	scp <i>user@host:file filedest</i>	Copia in modalità sicura (scp = <i>secure copy</i>) un file dall'host remoto al sistema locale, o viceversa.	scp programma.c rossi@ciop:prog.c

4.3 Compilazione di programmi C

Con i seguenti comandi è possibile “compilare” un file sorgente scritto in linguaggio C, ottenendo in questo modo un programma eseguibile.

Comando	Sintassi	Descrizione	Esempio
cc	cc <i>sorgenti</i> [-Wall] [-o <i>eseguibile</i>] [-lm]	Compila il programma sorgente (in linguaggio C) generando un programma eseguibile (cc = C Compiler).	cc prova.c -o prova -lm
gcc	gcc <i>sorgenti</i> [-Wall] [-o <i>eseguibile</i>] [-lm]	Compila il programma sorgente (in linguaggio C) generando un programma eseguibile. Viene usato il programma GNU C Compiler (gcc)	gcc prova.c -Wall -o prova

4.4 Comunicazione con altri utenti

I seguenti sono alcuni dei comandi messi a disposizione dell'utente dalla shell del sistema operativo UNIX, per verificare quali altri utenti stanno utilizzando il medesimo sistema e comunicare con loro attraverso strumenti di tipo “chat interattiva” o posta elettronica.

Comando	Sintassi	Descrizione	Esempio
who	who	Visualizza l'elenco degli utenti collegati sul sistema.	who
who am i	who am i	Visualizza i dati relativi al proprio account.	who am i
w	w	Visualizza l'elenco degli utenti collegati sul sistema	w
tty	tty	Visualizza il nome del terminale da cui è connesso l'utente.	tty
finger	finger [-l] [<i>user</i>] [<i>@host</i>]	Visualizza le informazioni di dettaglio relative agli utenti collegati sull'host specificato o su un particolare utente.	finger
write	write <i>username</i> [<i>term</i>]	Visualizza un messaggio sul terminale dell'utente indicato. Terminare il messaggio con “.” o Ctrl-d.	write rossi tty3
talk	talk <i>username</i> [<i>@host</i>]	Attiva una sessione di dialogo interattivo (<i>chat</i>) con l'utente specificato.	write bianchi
pine	pine	Attiva il programma per la lettura e la scrittura di messaggi di posta elettronica.	pine

4.5 Altri comandi utili

Nella seguente tabella sono raccolti alcuni comandi utili messi a disposizione dalla shell del sistema operativo UNIX.

Comando	Sintassi	Descrizione	Esempio
set	set <i>variabile=valore</i>	Imposta il valore di una variabile d'ambiente; ad esempio imposta la variabile con cui viene definita la modalità di funzionamento del terminale secondo lo standard Digital VT100 (24 righe ed 80 colonne).	set term=vt100
yppasswd	yppasswd	Cambia la password di accesso dell'utente sul sistema nel sistema distribuito NIS (<i>Network Information Service</i>). Viene richiesta prima la vecchia password e poi per due volte la nuova.	yppasswd
ps	ps [-ef]	Visualizza l'elenco dei processi attivi; con l'opzione "-ef" visualizza tutti i processi attivi sul sistema (<i>ps = process</i>).	ps -ef more
jobs	jobs	Visualizza l'elenco dei job attivi nella sessione corrente (in <i>foreground</i> ed in <i>background</i>).	jobs
kill	kill [-KILL -TERM] <i>pid</i>	Termina il processo identificato dal numero <i>pid</i> .	kill %1
date	date [+modalità]	Visualizza la data e l'ora corrente.	date
cal	cal [mese [anno]]	Visualizza il calendario del mese e dell'anno specificati.	cal 1492
man	man [comando]	Visualizza la pagina del manuale UNIX relativa al comando specificato.	man ls