

Alcune notizie di base su Linux e laboratori dipartimento

Paolo Marinelli, Antonio Messina

Dipartimento di Informatica – Scienza e Ingegneria

Università degli Studi di Bologna

Sommario

1. Introduzione.....	1
2. Il sistema operativo Linux.....	1
3. Organizzazione del sistema di file in Linux.....	3
4. Shell e comandi comunemente usati.....	4
4.1 I permessi sui file Linux.....	8
4.2 Comandi comuni per gestione file.....	9
4.3 La parola d'ordine.....	10
5. I laboratori.....	11
6. Account laboratori.....	11
6.1 Come ottenere account.....	12
6.2 Come cambiare la password.....	12
7. Regole generali dei laboratori.....	12
8. Organizzazione del file system nelle macchine dei laboratori.....	13
8.1 Le home: permessi e quota.....	13
8.2 La cartella public.....	14
9. Stampare in laboratorio.....	15
10. Collegarsi da remoto.....	16
11. Montare chiavette USB.....	16
12. Newsgroup.....	17
13. Librerie digitali.....	17

1. Introduzione

In queste brevi note si illustrano alcuni concetti di base utili per introdurre gli studenti all'uso del sistema operativo Linux (in seguito, brevemente, Linux) e delle macchine del laboratorio con questo sistema operativo.

2. Il sistema operativo Linux

Il sistema operativo Linux è di fatto un insieme di software a basso livello che gestisce i compiti del calcolatore, la sua memoria e le periferiche (mouse, tastiera, schermo, stampanti, scheda di rete, ...) di cui il calcolatore è dotato.

Ci sono due componenti di base del sistema operativo:

- il nucleo (kernel);
- e l'insieme dei programmi di sistema.

Il nucleo rende disponibili a tutti gli altri programmi le risorse del calcolatore; in particolare,

- la memoria permanente, ovvero lo spazio disco (disk space),
- la memoria volatile (memory)
- e le operazioni della CPU (Computer Processing Unit).

L'utente non vede mai direttamente questa parte di software, per evitare che il sistema possa essere corrotto.

L'insieme dei programmi di sistema comprende:

- i programmi di utilità (utility program);
- librerie (library) dei linguaggi per lo sviluppo di applicativi da parte dell'utente;
- il software di controllo delle periferiche (device driver);
- le istruzioni di configurazione (configuration script);
- programmi per svolgere dei servizi (server), come, per esempio, la posta;
- la documentazione;
- i programmi degli applicativi;
- gli shell, ovvero gli interpreti dei programmi di utilità, attraverso cui, per esempio, un utente dà comandi al kernel del sistema operativo.

L'insieme delle istruzioni e dei dati del software del sistema operativo può essere considerato, in generale, informazione. Informazioni collegate vengono raccolte in uno stesso file. Se il file è leggibile dall'utente costituisce un file di testo, se invece è leggibile non dall'utente ma dalla macchina il file è binario, perché scritto con simboli binari, 1 e 0. I file possono contenere liste di altri file; in questo caso si chiamano cartelle (directory).

Gli script o i file di configurazione sono file di testo, i file di utilità, come quello che permette di elencare i file in una cartella, sono binari.

Ogni file ha un nome (filename). All'interno della stessa cartella non si possono avere due file con lo stesso nome. Per dare un nome ai file si possono usare

- lettere maiuscole (A-Z);
- lettere minuscole (a-z);
- le cifre di base dei numeri decimali (0-9)
- il segno di sottolineatura (underscore) (_);
- il punto (.).

Se il nome di un file è composto da più parole, p.e. nome e file si usano due notazioni:

- nome_file;
- nomeFile, se non è una cartella;
- NomeFile, se è una cartella.

Si potrebbero usare anche lo spazio

- nome file;

- Nome File;

ma è sconsigliato perché alcuni applicativi non riconoscono lo spazio e considerano i due nomi come distinti.

L'insieme dei file e delle cartelle costituisce una struttura dati (data structure) secondo cui è organizzata l'informazione sul calcolatore. Questa struttura dati costituisce il sistema di file (filesystem) del calcolatore.

3. Organizzazione del sistema di file in Linux

Linux è organizzato in file: un file è un contenitore di informazioni che può contenere altri file (in questo caso diciamo che è una cartella, una directory).

Nel sistema operativo Linux i file sono organizzati gerarchicamente: una cartella principale ne contiene altre che a loro volta ne possono contenere altre.

La Figura 1 mostra le cartelle principali in Linux quando si utilizza l'interfaccia grafica

La directory principale, la radice (root), è **/**, da non confondere con la cartella **/root** riservata al gestore del sistema (root) e in cui gli utenti comuni non possono accedere.

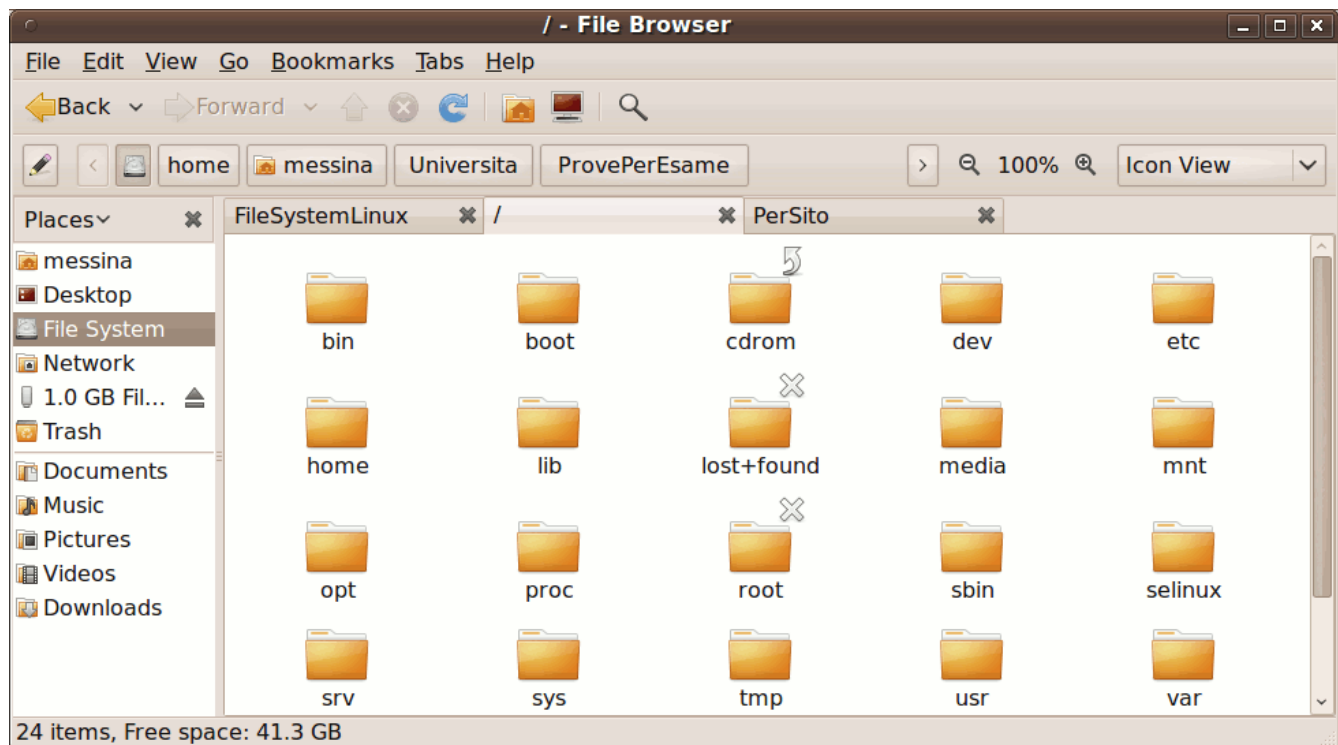


Fig. 1. Il sistema di file del sistema operativo Linux visto dall'interfaccia grafica

Il contenuto di alcune cartelle

/: la directory principale che contiene tutte le altre

/root: home dell'utente root

/boot: contiene tutte le immagini del kernel e file indispensabili al bootstrap del sistema

/etc: contiene i file di configurazione del sistema e dei programmi installati

/home: contiene le home directory degli utenti normali (tutti, tranne l'utente root)
/usr: contiene comandi, documentazione, librerie e sorgenti della maggior parte dei programmi (e i sorgenti del kernel)
/sbin: contiene comandi e programmi riservati a **root**
/bin: contiene comandi e programmi base per tutti gli utenti

4. Shell e comandi comunemente usati

L'utente quando entra su di un calcolatore, dando un nome utente (username) ed una parola d'ordine (password), entra in una sua cartella, con nome uguale al nome utente, che il sistemista ha creato nella cartella **home**. In questa cartella l'utente può esaminare i file, crearli o cancellarli.

Dal momento dell'ingresso tramite nome utente (username) e parola d'accesso (password) l'utente può parlare con il sistema operativo tramite uno shell, utilizzando un terminale.

La Figura 2 mostra un tipico terminale; l'utente non ha strumenti grafici nel terminale ma può utilizzare tastiera e mouse, oppure tastiera e touchpad

Lo shell è un programma che accetta comandi da parte dell'utente, li interpreta e li esegue. Lo shell, per l'esecuzione dei comandi, comunica con il kernel (nucleo), che è il programma che ha l'esclusivo accesso allo hardware del calcolatore (processore, memoria, dischi, ecc.).

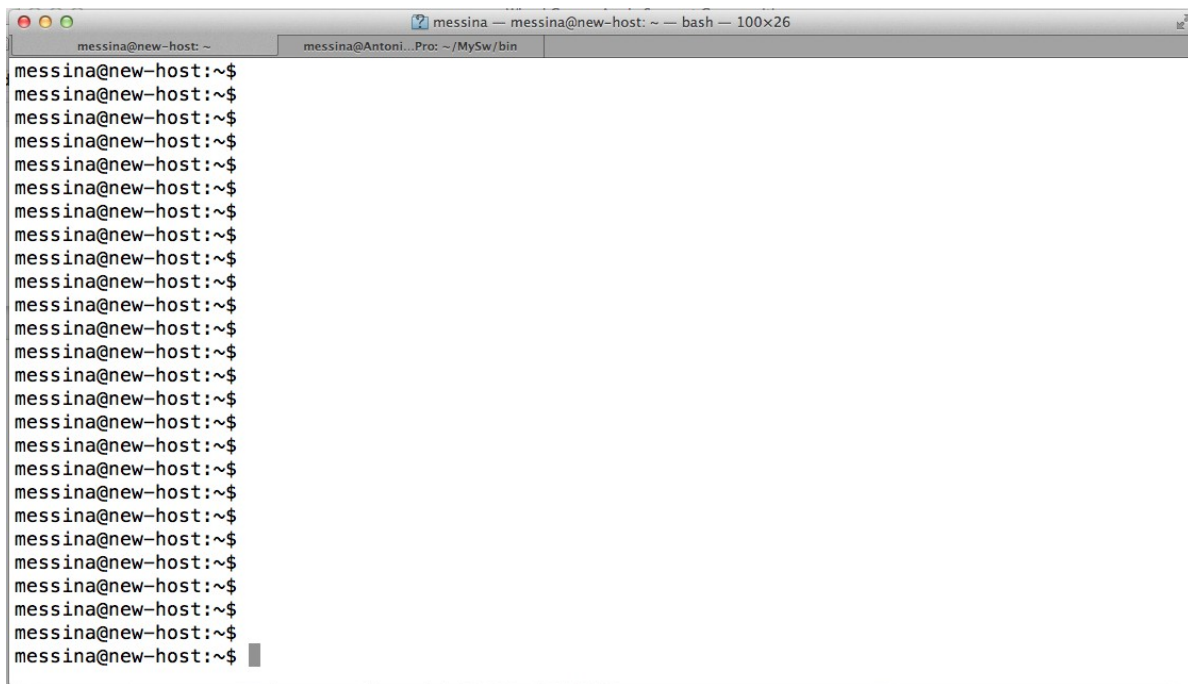


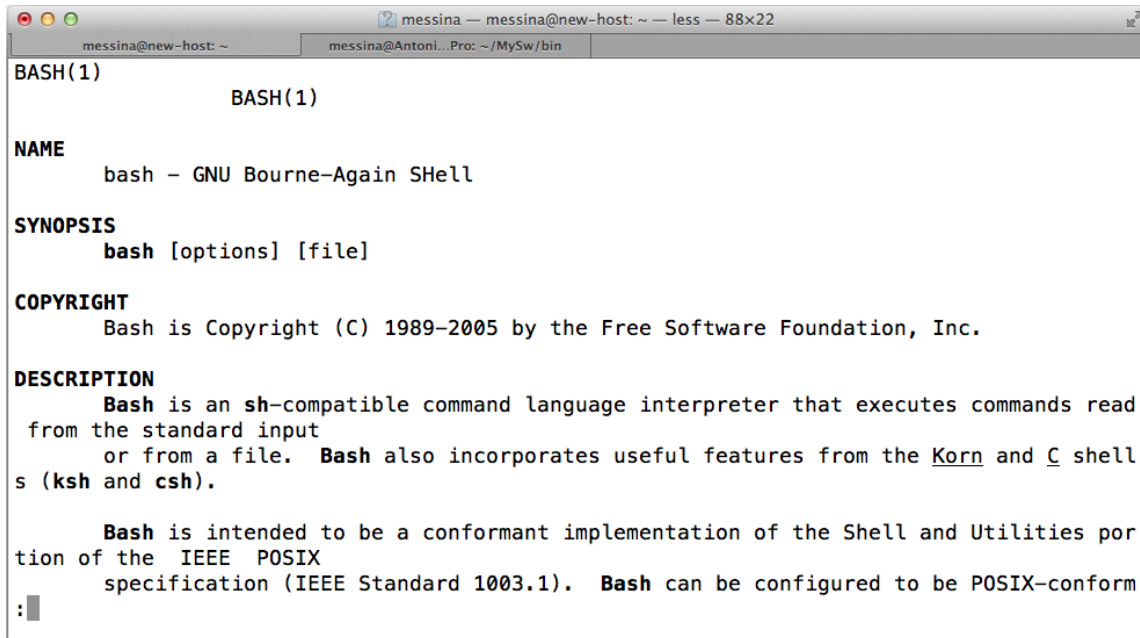
Fig. 2. Un terminale per dare comandi al sistema operativo

Ci sono diversi shell possibili. Prodotti in tempi successivi, gli shell disponibili sono

- **sh**: lo shell Bourne
- **csh**: lo shell C
- **tcsh**: lo shell T

- **ksh**: lo shell Korn
- **bash**: lo shell Bourne-Again

Informazioni sullo shell possono essere ottenute da terminale con il comando **man** seguito dal nome dello shell: **man bash** (vedi Figura 3).



```

messina@new-host: ~
messina@Antoni...Pro: ~/MySw/bin

BASH(1)
          BASH(1)

NAME
    bash - GNU Bourne-Again SHell

SYNOPSIS
    bash [options] [file]

COPYRIGHT
    Bash is Copyright (C) 1989-2005 by the Free Software Foundation, Inc.

DESCRIPTION
    Bash is an sh-compatible command language interpreter that executes commands read
    from the standard input
    or from a file. Bash also incorporates useful features from the Korn and C shell
    s (ksh and csh).

    Bash is intended to be a conformant implementation of the Shell and Utilities por
    tion of the IEEE POSIX
    specification (IEEE Standard 1003.1). Bash can be configured to be POSIX-conform
    :
  
```

*Fig. 3. Cosa appare nel terminale dopo aver lanciato il comando **man bash***

Quando i sistemisti attivano uno username per un utente su di un sistema Linux, viene creata una directory, con nome corrispondente al nome utente, nella directory di sistema **/home** e viene definito uno shell di default. Nelle distribuzioni Linux lo shell assegnato di default è lo shell **bash**.

Il sistema utilizza alcuni file presenti nella directory **/etc** per definire la configurazione di default dello shell. Per lo shell **bash** i file sono **/etc/profile** e **/etc/bash.bashrc**, mentre per altri shell, come, per esempio, lo shell **tcsh** questi file di sistema sono **/etc/csh.cshrc** e **/etc/csh.login**. Si consiglia di aprire questi file, che sono di testo e quindi sono leggibili, con un editor per avere un'idea delle risorse di default su di una macchina con sistema operativo Linux.

Il sistema permette all'utente di modificare, se necessario, le definizioni presenti nei file di configurazione di default. Quando si entra su di un sistema su cui si ha uno username, il sistema verifica nella home directory dell'utente se ci sono delle modifiche o delle aggiunte da parte dell'utente alla definizione della configurazione di default leggendo dei file nascosti (ovvero file il cui nome è preceduto da un punto e identificabili con il comando **ls -a**) per la personalizzazione delle risorse.

Per lo shell **bash** i file cercati nella home dell'utente e, se esistenti, letti ed eseguiti dal sistema sono **~/.profile**, **~/.bashrc**, quest'ultimo viene eseguito da **~/.profile**. L'utente dovrebbe verificare innanzitutto quale è lo shell di default (comando: **echo \$SHELL**) e quali

sono i file attivati nel suo shell di default. Si dovrebbe dare il comando `ls -a ~/` per individuare l'esistenza dei file sulla propria home (`~/` indica automaticamente la directory dell'utente nella cartella `/home`) e, se presenti, aprirli con un editor di testo per verificare quello che il sistema fornisce.

Quando si attiva lo shell, vengono definiti anche numerosi parametri dell'ambiente di shell (shell environment) che permettono agli applicativi o ai comandi di conoscere come devono essere eseguiti.

Per avere un'idea di quanto siano numerosi questi parametri di ambiente, basta lanciare, da una finestra di shell, il comando `env`. Uno spezzone tipico dell'uscita del comando `env` è il seguente

```
ciccio@astragalo:~> env
```

```
...
```

```
LESSKEY=/etc/lesskey.bin
```

```
INFODIR=/usr/local/info:/usr/share/info:/usr/info
```

```
MANPATH=/usr/local/man:/usr/share/man:/opt/cross/share/man
```

```
NNTPSERVER=news
```

```
KDE_MULTIHEAD=false
```

```
SSH_AGENT_PID=5338
```

```
HOSTNAME=eg
```

```
DM_CONTROL=/var/run/xdmctl
```

```
XKEYSYMDB=/usr/share/X11/XKeysymDB
```

```
GPG_AGENT_INFO=/tmp/gpg-jjdIRT/S.gpg-agent:5337:1
```

```
HOST=eg
```

```
PATH=/opt/kde3/bin:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/X11  
R6/bin:/opt/cross/bin:/usr/lib/mit/bin:/usr/lib/mit/sbin:/usr/lib/qt3  
/bin
```

```
...
```

Informazioni specifiche sui singoli parametri si possono avere come mostrato, per esempio, per `SHELL`:

```
ciccio@astragalo:~> echo $SHELL
```

```
/bin/bash
```

Si noti che per ottenere il valore assegnato a `SHELL` occorre premettere il simbolo `$`. Senza si avrebbe:

```
ciccio@astragalo:~> echo SHELL
```

```
SHELL
```

La variabile **PATH** è una variabile importante che indica dove andare a cercare quando si esegue un comando. Le cartelle vengono esaminate dal sistema in sequenza, a partire dalla prima cartella (nell'esempio precedente, che mostra il valore di **PATH**, viene esaminato per prima la cartella **/opt/kde3/bin**, se il comando non è trovato si passa alla seconda cartella, e così via).

Nel caso si installino pacchetti software in una directory specifica, diversa da quelle standard, occorre aggiungere in **PATH** la cartella in cui il sistema dovrà cercare i comandi relativi.

Per quanto detto, **PATH** è quindi una variabile di ambiente essenziale per chi programma. Se si sono costruite delle librerie con file di intestazione, come accade con i linguaggi C e C++, occorre aggiungere alle directory di ricerca le proprie.

In generale, conviene creare tre directory nella propria home directory:

- **lib**, in cui copiare librerie proprie
- **include**, in cui copiare, per esempio, i file di intestazione ***.h** del linguaggio C++, oppure i file archivio di Java ***.jar**
- **src**, in cui copiare i file di programmi.

Ciascuna di queste cartelle può eventualmente essere organizzata in sottocartelle (p.e. una gerarchia per programmi in C++, una per programmi in Java, ecc.)

Per creare le cartelle si esegue (**ciccio@astragalo:~>** è il prompt e **~** indica la directory dell'utente nella cartella **/home**) il comando **mkdir** (make directory):

```
ciccio@astragalo:~> mkdir include
```

```
ciccio@astragalo:~> mkdir src
```

```
ciccio@astragalo:~> mkdir lib
```

Per inserire le directory **include**, **lib** e **src** nella ricerca si può modificare la variabile **PATH**. Questo può essere realizzato modificando i file di risorse personali che si trovano nella propria home. Le modifiche minime da effettuare per lavorare con applicativi sono le seguenti per shell **bash**

```
PATH=$PATH:/home/ciccio/lib:/home/ciccio/include
```

```
export PATH
```

Queste due istruzioni, su due righe separate, possono essere inserite alla fine del file **~/.bashrc**. Se si lavora intensamente con propri applicativi converrà inserire le directory con il seguente ordine (nell'esempio si utilizza la macro **\$HOME** per individuare tramite il sistema la propria home):

```
PATH=/ $HOME/lib:$HOME/include:$PATH
```

Come si può notare dall'esempio il simbolo **:** permette l'aggiunta di nuove directory a variabili preesistenti e il simbolo **=** permette l'assegnazione ad una variabile.

Introdotte queste modifiche nei file di risorse per lo shell, basterà uscire dalla sessione e rientrare per attivarle. Se non lo si fa, il sistema operativo conserverà l'informazione acquisita, dai file di configurazione locali, all'ingresso dell'utente nella sua cartella di base

(home), che naturalmente non riportava ancora le modifiche introdotte nella sessione di lavoro.

Come abbiamo già detto, in Linux lo shell comunemente utilizzato come default è lo shell **bash**. A volte gli utenti, o i sistemisti stessi, possono attivare altri shell come, per esempio, **tcsh** per un qualche loro motivo esistenziale. Questo può essere un problema se chi adotta il nuovo shell non modifica coerentemente tutti gli applicativi della distribuzione Linux che sono stati compilati e inseriti nella distribuzione nell'ipotesi dello shell di default e che quindi andranno a cercare alcune macro di sistema nei file dello shell di default. Se questo succede, in genere, l'utente deve effettuare degli interventi per modificare la configurazione negli applicativi delle macro di sistema collegate ai file caratteristici dello shell.

Lo shell può essere cambiato digitando il nome dello shell su cui si vuole passare dal prompt di un terminale. Ma questo ha un effetto solo locale e i suoi effetti cessano di essere validi quando si esce dalla sessione. Per modificare il proprio shell per tutte le finestre si può inserire la definizione del parametro di sistema **SHELL** con istruzione **SHELL=/bin/bash** nei file indicati in precedenza che permettono la modifica dei valori di shell di default.

4.1 I permessi sui file Linux

Tutti i sistemi Unix hanno una gestione standard dei permessi sui file, che tiene conto del fatto che diversi utenti possono accedere alla stessa macchina e che possono esserci diversi gruppi di utenti, per esempio quelli che partecipano ad uno stesso progetto.

I permessi possono essere di lettura, scrittura ed esecuzione e vengono differenziati sulla base del tipo di utente rispetto al file o directory:

- utente proprietario, **owner** del file (sinteticamente **u**)
- gruppo proprietario, **owner group** del file (sinteticamente **g**)
- tutti gli altri utenti, **others** (**o**), che non sono **owner** e non appartengono allo **owner group**.

Il permesso di esecuzione è necessario per poter accedere a delle directory e, ovviamente, permette l'esecuzione di file come, per esempio i programmi binari compilati.

Per visualizzare i permessi di un file basta usare il comando **ls -l** che per ogni file genera un output del tipo:

```
-rwxr-xr-- 1 ciccio fluidi 1707094 2010-02-04 21:10 14SIP4017.pdf
drwxr-xr-x 17 ciccio fluidi 4096 2010-02-17 21:36 Attivita
```

Nella prima riga delle due mostrate, la prima colonna, composta da 10 caratteri, descrive i permessi sul file **14SIP4017.pdf**, dove:

- il primo carattere (nell'esempio: **-**) identifica il tipo di file, nel nostro caso un file in formato pdf.
- i successivi 3 caratteri identificano i permessi in lettura/scrittura/esecuzione (**r**ead/**w**rite/**e**xecute) dell'owner di **14SIP4017.pdf** (in questo caso lo owner ciccio ha tutti i permessi sul file: **rw**x);
- I successivi 3 identificano i permessi del gruppo **fluidi** di **14SIP4017.pdf** (in questo caso il gruppo ha permessi solo di lettura ed esecuzione: **r-x**);

- I successivi 3 identificano i permessi di tutti gli altri utenti del sistema (in questo caso hanno solo il permesso di lettura: `r--`).

Le successive colonne nell'output di `ls -l` indicano l'owner (`ciccio`), il gruppo (`fluidi`), la dimensione in Byte¹ (`1707094`), la data (`2010-02-04`) e l'ora (`21:10`) di ultima modifica e, infine, il nome del file (`14SIP4017.pdf`).

La seconda riga mostrata differisce nel primo carattere (`d`), che indica che il file è una directory, e per i privilegi del gruppo e degli altri utenti, che in questo caso possono non solo leggere ma eseguire tutti (`r-x`).

I permessi dei file possono essere modificati. Per modificarli si utilizza il comando `chmod`.

Per modificare i permessi dei file si usa il comando `chmod` come mostrato negli esempi.

Per dare privilegi di scrittura ed esecuzione a `others`:

```
chmod o+wx 14SIP4017.pdf
```

Per dare privilegi di lettura, scrittura ed esecuzione a tutti, si possono identificare sinteticamente tutti gli utenti con `a` (all)

```
chmod a+rwX 14SIP4017.pdf
```

Per togliere privilegi di scrittura ed esecuzione a `others`:

```
chmod o-wX 14SIP4017.pdf
```

Si può anche cambiare proprietario e gruppo. Si verifichi la sintassi dei comandi dando `man chown` e `man chgrp -help`.

4.2 Comandi comuni per gestione file

I comandi di gestione dei file su Unix sono paragonabili agli equivalenti di altri sistemi operativi a riga di comando.

Tutti questi comandi hanno varie opzioni che possono ampliare notevolmente le potenzialità del comando stesso.

Significato dei comandi comunemente usati:

`ls`: visualizza il contenuto di una directory;

`cd`: cambia la directory corrente. Corrisponde all'omonimo `cd` su DOS;

`cp`: Esegue la copia di uno o più file. Corrisponde al `copy` di DOS;

`mv`: Esegue lo spostamento, p.e. in una directory superiore (`mv file.txt ../file.txt`) o il cambio del nome di un file (`mv testo.txt mioTesto.txt`);

`rm`: cancella uno o più file (fare attenzione: su Linux non è previsto l'equivalente di un cestino temporaneo e quindi se un file viene cancellato lo si perde, a meno che non si disponga dei dispositivi opportuni per recuperarla);

`mkdir` e `rmdir`: sono usati rispettivamente per creare e cancellare una directory.

Le opzioni per i vari comandi si ottengono dando il comando `man <nome comando>`.

1 bit = **b**inary **u**nit, 1 Byte = 8 bit

Comandi con opzioni di uso frequente possono essere memorizzati per il proprio uso, ridefinendo un nuovo nome di comando. Per esempio, per richiedere un prompt di conferma per la cancellazione si può scrivere

```
alias ri='rm -i'
```

nel file `.bashrc`, oppure nel file `.bash_aliases` che viene attivato da `.bashrc` all'ingresso nella propria home directory.

Si noti che tutte le modifiche effettuate al file `.bashrc` diventano operative non con l'uscita dal file ma all'ingresso successivo sulla macchina.

4.3 La parola d'ordine

Nella maggior parte dei calcolatori che si acquistano la configurazione predefinita prevede che, all'accensione della macchina l'utente entri senza parola d'ordine nella sua cartella collocata nella cartella home.

Questa è una pessima abitudine: chiunque si impadronisca del calcolatore può utilizzarlo senza problemi. Non solo, chi si è impadronito del calcolatore può operare a nome del vero proprietario secondo i suoi fini, non necessariamente nobili.

Si deve quindi configurare la propria macchina in modo che vengano richiesti un nome utente (username) ed una parola d'ordine (password).

Lo scopo della password è quello di limitare l'accesso al calcolatore solo a quelli che la conoscono. Una buona password deve avere due proprietà

- non deve essere facile da identificare
- deve essere facile da ricordare

Un utente poco accorto di solito si concentra sulla seconda proprietà, trascurando l'altra. Tipicamente questo tipo di utente usa nomi di famigliari, cuccioli, date di nascita e così via. In questo modo chiunque conosca abbastanza bene l'utente riesce ad indovinare la password facilmente.

Quando si digita la password sulla apposita finestra, il calcolatore legge la stringa e la trasforma con una opportuna procedura (un algoritmo) in un'altra stringa che viene poi immagazzinata sul disco. Questa nuova stringa, ottenuta da un processo unidirezionale, è la forma crittata della password originale. Quindi neanche il calcolatore conosce la password originaria. Tutte le volte che l'utente la fornisce, il calcolatore la critta nuovamente con lo stesso algoritmo e la confronta con quella che ha in memoria, prima di autorizzare l'ingresso dell'utente.

Più una password è lunga, più difficile è la decodifica. Lavorando in binario, ovvero con due soli simboli, se si utilizzano n caratteri per la password, ci saranno 2^n possibilità.

Si possono usare diverse tecniche per definire una buona password. Una è quella di scegliere una frase da cui estrarre la password e poi modificarla, comprimerla e mescolarla. A titolo di esempio, ovviamente non utilizzabile d'ora in poi per una password, possiamo mostrare come ottenere una buona password partendo da: La Trilogia di Matrix.

In sequenza, si ottiene:

- Trilogia Matrix
- 3logiaMa3x
- Ma3x3logia
- Ma9logia
- mA9LoGIA

Si tenga presente che una buona password non è “per sempre”. La password va cambiata periodicamente, almeno ogni sei mesi. Per il cambio si possono utilizzare gli strumenti grafici, oppure il comando **passwd** da terminale.

5. I laboratori

Gli studenti hanno a disposizione due laboratori GNU/Linux:

- Laboratorio Ercolani, Mura Anteo Zamboni 2, con 17 postazioni GNU/Linux. Aperto dalle 09.00 alle 19.30, da lunedì al venerdì. Sabato aperto dalle 09.00 alle 14.30.
- Via Ranzani 14/C, con 40 postazioni in due sale. Aperto dalle 09.00 alle 18.00, dal lunedì al venerdì.

Nelle postazioni del laboratorio Ercolani in tutte le postazioni è installato il sistema operativo **Debian Wheezy**. La maggior parte delle macchine montano un processore Intel Core-i5 a 3,20GHz e 4GB di RAM. Le macchine rimanenti montano un processore Intel Core 2 Duo a 2,93GHz e una RAM di 2GB.

Sulle macchine del laboratorio Ranzani, alla data corrente, è installato il sistema operativo **Ubuntu 12.04 LTS**. Le macchine montano un processore Intel Core 2 Duo a 2,93GHz ed hanno una RAM di 2GB. In futuro, anche queste macchine avranno il sistema operativo Debian.

Per accedere alle macchine è necessario essere in possesso di un account specifico (vedi sezione successiva).

Inoltre, il **laboratorio Ercolani** è sottoposto a controllo accessi. Questo vuol dire che troverete la porta chiusa e per aprirla dovrete usare il vostro badge da studenti, avvicinandolo all'apposito lettore situato accanto alla porta di ingresso.

I badge sono rilasciati dalla segreteria studenti (via Zamboni 33). Prima che veniate riconosciuti potrebbero essere necessari alcuni giorni (tipicamente 5). Se dopo una decina di giorni non riuscite ancora ad entrare, potete passare dall'ufficio tecnici di Mura Anteo Zamboni 7 (in orario di ricevimento) o scrivere ad accessi@cs.unibo.it.

Infine, sia il laboratorio Ercolani che il laboratorio Ranzani sono sorvegliati da apposito personale. Prima di accomodarvi, dovrete lasciare un documento al sorvegliante di turno e firmare gli orari di ingresso e uscita dal laboratorio.

6. Account laboratori

Per accedere alla macchine dei laboratori è necessario possedere un **account di Ateneo** (conosciuto anche come **account di DSA** e che è quello che si userà per accedere alla posta @studio.unibo.it e ad AlmaEsami). Questo account è stato comunicato al momento

dell'iscrizione. Per renderlo valido occorre seguire le istruzioni comunicate all'iscrizione. Le credenziali hanno la forma nome.cognome@studio.unibo.it.

Gli account di Ateneo degli studenti del primo anno sono stati abilitati al login sulle macchine dei laboratori. Se però qualche studente dovesse avere dei problemi con il login è pregato di rivolgersi ai tecnici scrivendo ad account@cs.unibo.it.

6.2 Problemi con la password

La password dovrebbe essere cambiata ogni sei mesi. Nel caso sorgano dei problemi, come per esempio l'incapacità di ricordare della password, si può accedere al portale di Ateneo (unibo.it) ed impostare la ricerca di "credenziali studenti unibo". Nell'elenco che appare, selezionando "**Credenziali** istituzionali e accesso ai servizi online per gli **studenti** dell'Università di Bologna — Università di Bologna" si aprirà una pagina con le indicazioni utili a riacquisire la password.

Si sottolinea che la robustezza della password è una cosa che deve essere sentita in prima persona da ogni utente. Se qualcuno riesce ad entrare con le vostre credenziali, potrebbe fare a nome vostro qualsiasi cosa, anche illegale. In questi casi i responsabili del sistema risalgono al possessore dell'account e passano i dati alla polizia postale. Le conseguenze di illeciti informatici sono di tipo **penale**.

7. Regole generali dei laboratori

I laboratori sono una piattaforma su cui sviluppare progetti, stampare documenti, accedere ad Internet ecc. In sostanza i laboratori sono delle risorse messe a disposizione degli studenti per le loro esigenze di didattica e scientifiche.

Ovviamente, queste risorse sono condivise tra gli studenti. Questo vuol dire che ogni studente non deve pensare al laboratorio come al proprio laboratorio personale. Non solo, non deve pensare neanche alla singola postazione, come alla propria postazione personale, né alla stampante come alla propria stampante personale.

Ovvero, le risorse possono essere ovviamente usate (sono lì per quello), ma con criterio, sapendo quindi che possono essere necessarie anche ad altri studenti.

Il principio generale nell'utilizzo dei laboratori è dunque quello di usare le risorse ma non in modo esclusivo. Quindi:

- non è corretto stampare documenti che non siano legati ad esigenze didattiche o scientifiche. Stampare un volantino "Cerco posto letto" spreca carta e toner a discapito di altri studenti.
- Lanciare un processo su una macchina che occupa l'intera CPU o l'intera RAM, vuol dire rendere quella macchina inutilizzabile per altri studenti. In questi casi, i tecnici possono intervenire e terminare ("killare") i processi che consumano troppe risorse.
- Loggarsi su una macchina e poi andarsene non è corretto nei confronti di altri studenti che non possono così usarla.
- Spegnerne una macchina non è corretto, visto che le macchine sono accese 24 al giorno 7 giorni su 7 e accettano connessioni remote: potrebbe esserci un utente

loggato da remoto o che magari ha lanciato un processo che ha bisogno di girare per diverse ore o giorni.

Inoltre ci sono delle regole di buon senso che sono:

- non si possono lasciare sessioni incustodite (senza lock screen)
- non è permesso mangiare o bere nei laboratori
- non è permesso staccare prese di rete dati o elettriche
- non è permesso collegare il proprio portatile nelle prese libere, se non nelle postazioni dedicate
- non è permesso portare amici, fidanzati/e, zii/e, nonni/e, ecc.

8. Organizzazione del file system nelle macchine dei laboratori

Le macchine dei laboratori hanno un disco locale, diviso in varie partizioni, in cui è installato il sistema operativo e i vari programmi. Inoltre, montano via NFS (Network File System) delle partizioni remote da server presenti nella sala server della palazzina di Mura Anteo Zamboni 7.

Nei mount point sotto /home (ad esempio, /home/staff, /home/students, ecc) sono montate le home degli utenti con un account presso il Dipartimento e sono suddivise appunto per categorie di utenti.

Gli studenti hanno la home sotto /home/students e questa viene montata dunque via NFS dal server chiamato dx.

Quindi, uno studente con username pippo avrà la home in /home/students/pippo.

Gli studenti non sono amministratori delle macchine dei laboratori e dunque non hanno accesso completo (lettura e scrittura) all'intero file system, ma soltanto in alcune gerarchie. Di queste ne vedremo due.

8.1 Le home: permessi e quota

Ogni studente è responsabile del contenuto che inserisce nella propria home così come dei *permessi* ad essa associata. Dunque, se si vuole che i propri documenti non siano leggibili a tutti gli altri (studenti e docenti), ci si deve accertare di quali siano i permessi dei file ed eventualmente cambiare i permessi tramite gli opportuni comandi (**chmod**).

Il fatto che ogni macchina monti le home via NFS fa sì che ci si può loggare su qualunque macchina e trovare la stessa home e con gli stessi documenti. Questo fa sì che, se una macchina dovesse essere non utilizzabile, potete trovare i vostri dati collegandovi ad un'altra macchina.

Inoltre, le macchine sono mantenute in modo tale da presentare la stessa gamma di software installati.

Ogni studente ha a disposizione una quota di spazio disco, questo per evitare che un solo studente possa “mangiarsi” l'intero spazio disco, togliendolo quindi dalla disponibilità degli altri studenti.

La quota è settata a 220MB con limite hard a 250MB. Questo vuol dire che se si supera la quota di 220MB si hanno a disposizione 7 giorni di *grace* per poter rientrare in quota. Superati i 7 giorni, viene negata la possibilità di scrittura nella propria home. Inoltre, non è possibile superare in nessun caso i 250MB: il file system negherà la possibilità di scrivere se si supera il limite hard.

Per controllare la propria quota, è possibile utilizzare il comando **quota**.

È possibile richiedere un aumento quota per motivi di studio/tesi. La richiesta va inviata tecnici@cs.unibo.it, mettendo in conoscenza un docente di riferimento e fornendo una motivazione valida (ovvero, verranno respinte motivazioni del tipo “ho bisogno di maggiore spazio”).

Nel caso si dovesse superare la quota, è probabile che non si riesca ad entrare nelle macchine dei laboratori nella normale modalità grafica.

In tal caso, è necessario entrare da una tty (premere Ctrl+Alt+F1 per passare alla tty1, Ctrl+Alt+F2 per passare alla tty2, ecc) oppure remotamente tramite ssh. A quel punto si deve liberare spazio, cancellando file e cartelle. Una cartella che di solito può mandare fuori quota è quella del browser (p.e. .mozilla).

Per capire quanto spazio si sta utilizzando, si può fare riferimento al comando **du**, soprattutto se richiamato con l'opzione **-h** per avere informazioni sull'occupazione in formato leggibile dall'utente (*human readable*).

8.2 La cartella public

220MB non sono molti. Dunque, le macchine dei laboratori montano via NFS una partizione di disco non quotato. Questo spazio disco si trova in /public e viene comunemente detto *la public*.

Sotto questo spazio, ogni studente è libero di creare la propria cartella e di mettere tutti i file che vuole, senza nessun limite di quota. Questo spazio disco però è pensato per appoggiare dati per un periodo di tempo limitato. Infatti, ogni primo del mese (tra poco potremmo cambiare ad ogni prima domenica del mese e ne daremo comunicazione tramite i siti di corsi di studio/di dipartimento), tutto quello che si trova nella public viene cancellato senza possibilità di recupero.

8.3 La cartella 2014_2015

Per gli studenti del corso di Programmazione Internet è a disposizione un'altro spazio: /home/students/COURSES/pi/2014_2015.

Entrando in questa cartella, lo studente potrà creare una cartella con nome **nome.cognome** e potrà depositarvi tutti gli elaborati sviluppati durante le esercitazioni. Questo spazio e i file in esso depositati per il tempo che intercorre dall'inizio dell'Anno Accademico alla fine dello stesso Anno Accademico.

9. Stampare in laboratorio

Le stampanti presenti in laboratorio sono:

- Laboratorio Ercolani

- corriere
- gazzetta
- Laboratorio Ranzani
 - repubblica
 - stampa

Le varie applicazioni installate nelle macchine dei laboratori vedono automaticamente le stampanti. Ad esempio, Acrobat Reader vi permetterà di scegliere la stampante su cui lanciare la stampa.

Le stampanti sono gestite da un server di stampa che gestisce le code di stampa. Potrebbe accadere che una stampa (detta anche *job*) non esca subito, perché ce ne sono altre in attesa.

Per controllare la coda di stampa, ad esempio della stampante repubblica, è possibile utilizzare il seguente comando:

```
pmarinel@elettra:~$ lpq -Prepubblica
repubblica is ready
no entries
```

A volte potrebbe essere comodo lanciare una stampa da linea di comando. Ad esempio, per stampare il file *documento.ps* sulla stampante repubblica, può essere utilizzato il seguente comando:

```
lpr -Prepubblica documento.ps
```

Per fermare un job, è possibile utilizzare il comando *lprm*.

ATTENZIONE. I documenti pdf, NON vanno stampanti con il comando *lpr*. Infatti, uscirebbero pagine e pagine di simboli indecifrabili, sprecando così carta e toner.

Da linea di comando, si possono lanciare in stampa dei documenti di testo oppure dei documenti *PostScript* (PS).

Può dunque essere utile sapere che per convertire un documento PDF in PS, esiste un comando apposito: *pdf2ps*.

Altro comando utile è *psnup*, per la manipolazione di documenti PS (ad esempio, per mettere molteplici pagine per foglio).

NOTA. La carta dei laboratori ha un costo per il Dipartimento e non è infinita. Dunque, lo studente è tenuto a stampare consumando soltanto lo stretto indispensabile. Non si escludono in futuro politiche di gestione del consumo di stampa più strette.

10. Collegarsi da remoto

Le macchine dei laboratori hanno indirizzi IP pubblici e sono accese 24 ore al giorno per 7 giorni alla settimana. Questo vi permette di collegarvi da remoto anche da casa e in qualsiasi momento.

Per farlo, dovete usare un **client ssh**. Se state usando un computer basato su Linux o su un Mac, avrete a disposizione il comando **ssh**. Se invece state usando un Windows, potete usare l'applicazione **putty**, scaricabile gratuitamente dalla rete. In tutti i casi, la connessione che si instaura è cifrata.

Ad esempio, se l'utente **paolo.marinelli5@unibo.it** si vuole collegare alla macchina del laboratorio giovanna, dovrà usare il seguente comando:

```
ssh -l paolo.marinelli5@unibo.it giovanna.cs.unibo.it
```

Potrebbe accadere che una macchina non sia raggiungibile. In tal caso, ne potete provare un'altra. Se però diverse macchine non sono raggiungibili allora si potrebbe essere verificata una delle seguente situazioni:

1. c'è un guasto alla rete di Dipartimento
2. le macchine del laboratorio Ranzani sono usate per esami. Durante gli esami sono isolate e dunque non è possibile collegarvi da remoto. Potete collegarvi al sito <http://calendar.cs.unibo.it> per controllare le disponibilità dei laboratori (così come delle aule Ercolani)
3. i tecnici hanno spento le macchine per manutenzione (ad esempio, perché hanno aggiornato il kernel e quindi devono riavviarle per far usare il nuovo kernel).

La lista delle macchine è pubblicata alla pagina

<http://www.informatica.unibo.it/it/servizi-informatici/accesso-da-remoto>

Il client SSH permette di creare tunnel cifrati. Questo, potrebbe essere utile per fare collegamenti a servizi che accettano connessioni soltanto dalla rete di Dipartimento.

Ad esempio, supponiamo che l'host remoteservice offra un servizio web (porta 80) ma che lo offra soltanto ai client interni alla rete di Dipartimento (macchine dei laboratori). Se l'utente paolo.marinelli5@unibo.it si trova a casa e vuole comunque accedere al servizio, può usare il seguente comando:

```
ssh -L:8000:remoteservice:80 -l paolo.marinelli5@unibo.it giovanna.cs.unibo.it
```

Una volta dato questo comando, l'utente pmarinel accede al servizio remoteservice (porta 80) facendo connessioni in localhost sulla porta 8000. Tutte le informazioni spedite a localhost:8000 passano attraverso la connessione cifrata instaurata verso la macchina giovanna, la quale penserà a girarle verso il servizio remoteservice:80.

11. Montare chiavette USB

È possibile montare chiavette USB sulle macchine dei laboratori. Nelle maggior parte dei casi, non appena si inserisce la chiavetta, questa sarà montata automaticamente. Tuttavia, è opportuno sapere i comandi montare e smontare dei dispositivi:

- **mount /dev/sdb1 ./miachiavetta**
 - monta la chiavetta individuata dal device **sdb1** nel mount point **./miachiavetta**
- **umount ./miachiavetta**
 - smonta il dispositivo montato in **./miachiavetta**

12. Newsgroup

I newsgroup sono piazze virtuali in cui discutere. Il Dipartimento mette a disposizione degli studenti e dei docenti diversi newsgroup, ai quali è possibile accedere tramite il news server news.cs.unibo.it.

Gli studenti possono liberamente comunicare tra loro, tramite il newsgroup:

unibo.cs.students

I vari client di posta (ad esempio, **Thunderbird**) offrono la possibilità di collegarsi ad un **news server**. Per accedere ai newsgroup di cui sopra, è importante dunque indicare il nome del server (news.cs.unibo.it) e quindi sottoscrivere ai gruppi di discussione desiderati.

Tuttavia, l'accesso ai newsgroup è possibile soltanto dalla rete del Dipartimento.

Dunque si può usare la tecnica del **tunnel ssh**:

```
ssh -L1119:news.cs.unibo.it:119 -l username giovanna.cs.unibo.it
```

13. Librerie digitali

Il Dipartimento ha sottoscritto contratti con riviste digitali, in modo tale che dalla rete di Dipartimento sia possibile accedere ad articoli scientifici.

A questo scopo è utile utilizzare il proxy di dipartimento quando ci si trova fuori dal laboratorio. Ecco come si usa

1. si instaura il tunnel **ssh** tramite il comando

```
ssh -L8000:proxy.cs.unibo.it:8080 -l username giovanna.cs.unibo.it (ovviamente si può usare qualunque altra macchina dei laboratori)
```

2. configurare il browser per usare il proxy sulla porta 8000 di localhost

Per il secondo punto, si allega la schermata di configurazione di **icewasel** in Figura 4.

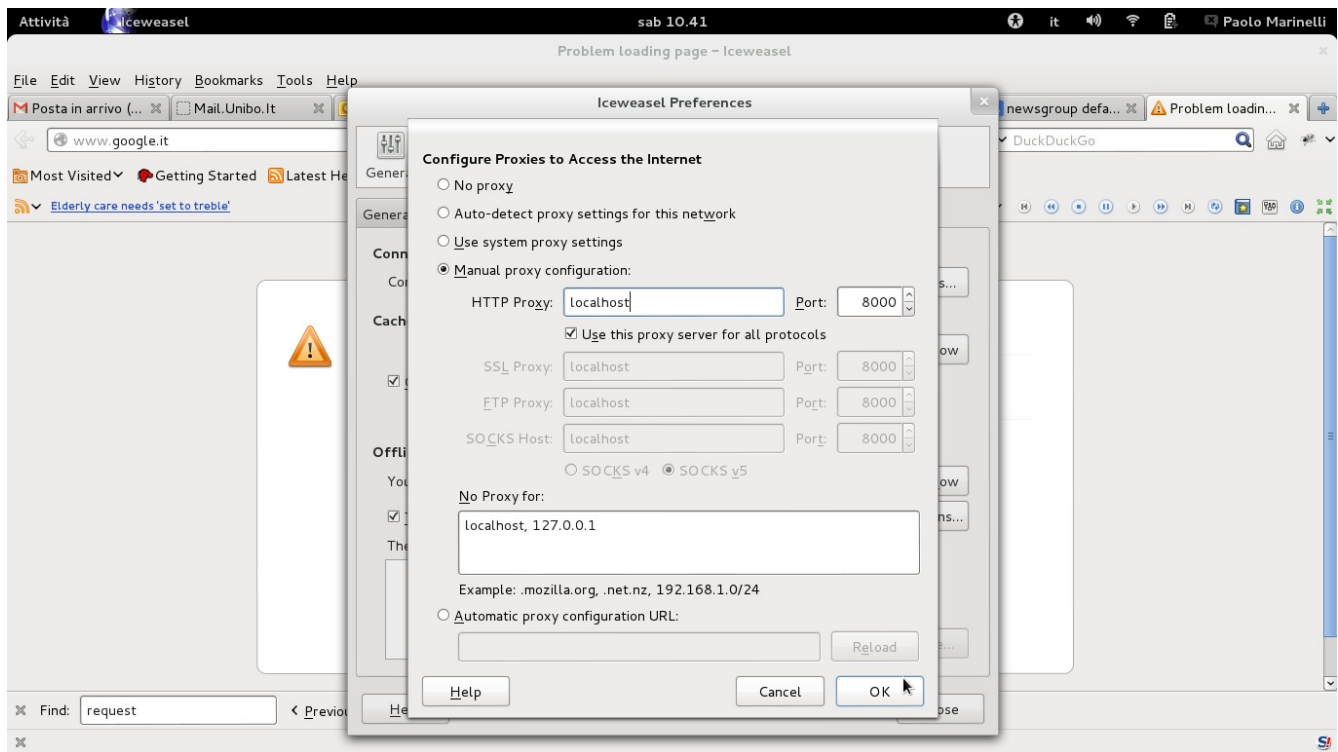


Fig. 4. La configurazione proxy di icewasel per tunneling

La Figura 5 mostra come configurare il proxy per tunneling su Mac

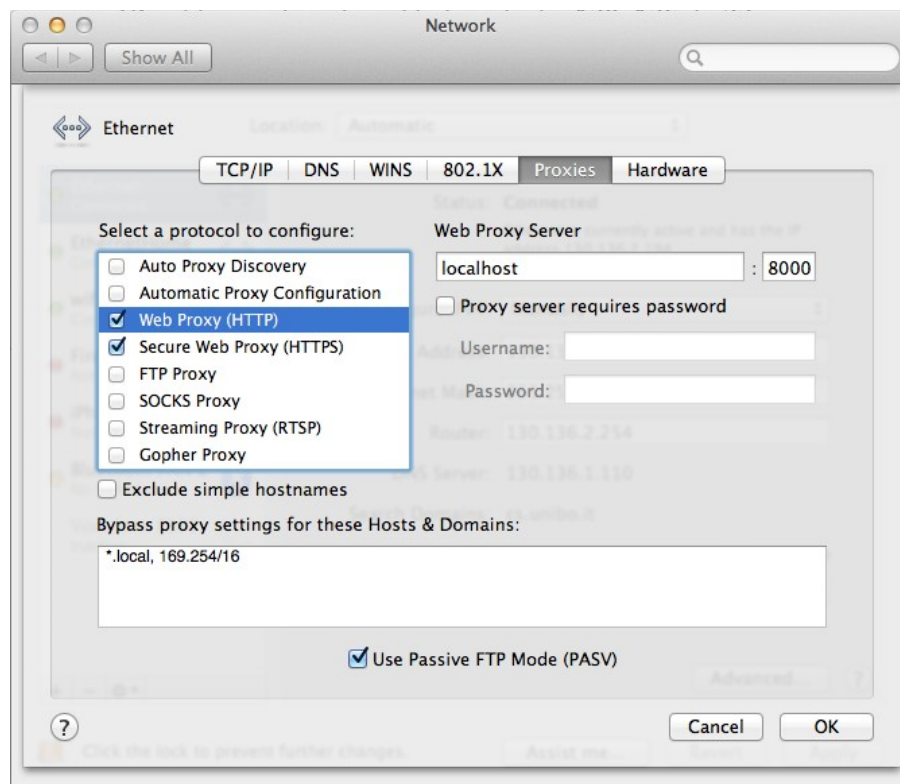


Fig. 5. La configurazione proxy di Safari