

G. Carboni  
Aprile 1999

## Introduzione a LINUX

- cos'è Linux?
- cosa sono le distribuzioni di Linux?
- come si installa
- le partizioni del disco rigido
- la configurazione
- ricompilare il Kernel
- X-windows
- sicurezza

## Cos'è Linux?

- Linux è un sistema operativo “tipo” UNIX che è stato sviluppato per processori INTEL x86, e ora esiste per molte altre piattaforme
- Linux è nato come progetto di uno studente finlandese, Linus Torvalds, ed è sviluppato insieme a lui da moltissimi programmatori sparsi per il mondo
- tutto lo sviluppo si basa su uno sforzo puramente volontario. Il software relativo è soggetto alla licenza GPL (General Public License) formulata dalla Free Software Foundation (FSF) che porta avanti anche il progetto GNU
- il software sotto GPL dev'essere distribuibile a chiunque, inclusi i codici sorgente, e nelle distribuzioni dev'essere fatto chiaro che l'utente ha il diritto di trasmettere il software a chiunque lui voglia, ma sempre alle stesse condizioni

Il modello FSF/GNU è l' antitesi del modello di sviluppo commerciale, in cui il software è gelosamente protetto da copyright e per funzionare legalmente ha bisogno di una licenza ecc.

I sostenitori del modello commerciale ritengono che solo grazie ad esso si può avere del S/W di qualità, perchè la qualità si paga

Linux è una palese contraddizione di questa teoria, dato che il S/W prodotto è di altissima qualità. Tutto questo nonostante ci sia pochissimo coordinamento nella direzione del progetto

Linux è molto ben supportato (in maniera ovviamente gratuita). Se avete un problema di S/W o H/W rivolgendovi ai newsgroups avrete la risposta nel giro di poche ore, e quel che più importa, sarà una risposta competente

Negli ultimi anni Linux ha ricevuto un' attenzione crescente da parte degli operatori commerciali. La diffusione è in crescita esponenziale. **IBM offre ora Linux come alternativa a Windows NT sui server Netfinity.**

Linux compete molto bene con Windows NT nel campo dei servers, per i quali ha indubbi vantaggi

Vantaggi per l' utente medio:

- a parità di H/W Linux è più veloce di Windows 95 e NT
- È più stabile di Windows
- Convive senza problemi sullo stesso disco con Windows 95, 98, NT, DOS, ecc.
- Può leggere i files DOS e Windows e può salvarli negli stessi formati, sia su dischi che su floppy.
- La rete è configurabile più facilmente
- Il contatto coll' H/W è più diretto

**Linux non è il solo UNIX per INTEL: esistono anche FreeBSD (Berkeley), SCO UNIX (Santa Cruz Operation) e Minix**

**Linux è però il più diffuso e il meno caro.**

Per noi, “fisici da workstation”, i vantaggi sono:

- Chi è già abituato a UNIX, può continuare ad usarlo anche a casa su H/W di prezzo molto basso
- In genere gli applicativi offrono una scelta sufficiente per noi e sono gratuiti:
  - Ottimo supporto di rete
  - TeX/LaTeX
  - PAW, Root e tutte le librerie CERN (inclusi i MC, GEANT ecc.)
  - Compilatori C, C++, Fortran
  - Perl
  - ecc. ecc.
- Cominciano a uscire applicativi Windows-like (office suites) di buona qualità per chi ama l' interfaccia WYSIWYG
  - Applixware
  - StarOffice
  - WordPerfect

## Documentazione

Se conoscete già UNIX avrete pochi problemi. Per i problemi specifici di Linux potete consultare:

- Manuale Red Hat
- Manuale SuSE
- Running Linux

I manuali Red Hat e SuSE sono anche in italiano.

Consigliabile:

[Linux for Dummies Espresso \(in italiano, L.19.000\)](#)

Un'ottima rivista è:

[Linux Journal \(abbonamento \\$34 /anno in Italia\)](#)

Infine, i vari **HOWTO** contengono moltissime informazioni su vari argomenti. [Insieme ai files in /usr/doc e /usr/info e alle man pages si trovano sempre in ogni distribuzione di Linux.](#)

## Terminologia

A rigor di termini, Linux è il Kernel

Il resto è software che in gran parte viene compilato coi compilatori gcc.

Identificare un sistema Linux equivale a conoscere la versione del suo Kernel

```
$ uname -r  
2.0.35
```

Versione attuale 2.0.36

Non si deve identificare Linux con la “distribuzione”, che è una cosa diversa.

## Cos'è la distribuzione?

La distribuzione è un insieme di programmi che viene raccolto in un pacchetto, in generale sotto forma di CD-ROM, che contiene tipicamente:

- Il Kernel
- Tutti i comandi fondamentali
- Le librerie
- I compilatori
- Applicativi vari, editori, ecc. (a volte software commerciale)
- X-Window

Oltre a questo la distribuzione comprende sempre un suo programma di installazione (che parte in genere con un dischetto) e serve anche ad amministrare il sistema.

Le distribuzioni si pagano (poco) e in genere includono un manuale cartaceo.

**Il pregio di una distribuzione è spesso proprio nel suo programma di installazione e in quanto esso facilita gli aggiornamenti del sistema**

## Quante distribuzioni esistono?

Non tantissime, tuttavia il loro numero è in crescita.

C'è il rischio che si arrivi a problemi di incompatibilità (esempio: versioni diverse delle librerie libc, glibc,...)

Le più popolari:

- Slackware (la più spartana)
- Red Hat (ottima per gli aggiornamenti)
- SuSE (“tedesca”)

### Consiglio

Se non volete divertirvi a sperimentare, usate quelle che usano già i vostri colleghi. Risparmierete tempo.

## Come si installa LINUX?

Prima di tutto diciamo che è consigliabile non installare solo Linux su un PC.

È meglio riservare una parte del disco a DOS/Windows

Perchè?

- È più comodo avere due sistemi
- In certi casi particolari può essere utile avere il DOS per configurare, ad esempio, le schede di rete.
- Se avete un nastro tipo TRAVAN, i nastri si formattano solo su DOS (ma francamente non consiglio a nessuno di formattarsi i nastri!!)

Dopo essersi procurata una distribuzione si decide quale spazio dedicare a Linux.

Se sul PC è già installato Windows, occorre allora ridimensionare la/le partizioni già esistenti per far posto a Linux

## Partizioni del disco

- Un disco di PC può essere suddiviso in max 4 partizioni primarie
- 1 sola di queste può essere una partizione estesa; al suo interno si possono creare tante partizioni logiche quante se ne vogliono

**Attenzione** Windows 95 e Windows NT devono essere avviati da una partizione primaria.

**Attenzione** Normalmente DOS e Windows 95 tollerano una sola partizione primaria visibile su ogni disco (disco C:), le altre, se ci sono, devono essere logiche.

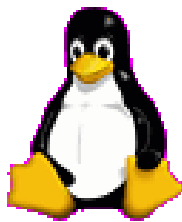
**Attenzione** Le partizioni avviabili (di boot) devono trovarsi sotto il cilindro 1024 del disco

**Attenzione** La partizione per un dato sistema operativo deve essere creata solo con l' FDISK di quel sistema operativo (non potete creare partizioni Dos da Linux ecc.)

Quindi da DOS userete **FDISK.EXE** mentre da Linux userete **fdisk**.

## Ridimensionare le partizioni

- Metodo brutale: cancellare le partizioni Windows e crearle nuovamente usando **FDISK.EXE** da un dischetto di boot DOS  
**oppure**
- Non distruttivo: ridimensionare le partizioni DOS/Windows utilizzando un programma:
  - FIPS.EXE (gratis, si trova sempre nelle distribuzioni)
  - PARTITION MAGIC (commerciale, v3.0...v4.0)  
La versione 3.0 può ridimensionare solo partizioni FAT, HPFS (OS/2), NTFS ma non Linux.



## Installazione

Una volta fatto lo spazio, si può cominciare l'installazione, in genere partendo da un floppy di boot. All'inizio verrà chiesto di creare le partizioni. Questo si fa con il programma **fdisk** di Linux. Ci vuole anche 1 partizione per lo swap (dimensioni  $\approx$  RAM).

Ecco la schermata e il menu di fdisk (o /sbin/fdisk)

```
xterm
Disk /dev/hdb: 16 heads, 63 sectors, 1652 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start    End  Blocks  Id System
/dev/hdb1             1         1    204  102784+  6 DOS 16-bit >=32M
/dev/hdb2            205        205    270   33264  82 Linux swap
/dev/hdb3            271        271   1462  600768  83 Linux native
/dev/hdb4           1024       1463   1652   95760  83 Linux native

Command (m for help): m
Command action
 a toggle a bootable flag
 b edit bsd disklabel
 c toggle the dos compatibility flag
 d delete a partition
 l list known partition types
 m print this menu
 n add a new partition
 p print the partition table
 q quit without saving changes
 t change a partition's system id
 u change display/entry units
 v verify the partition table
 w write table to disk and exit
 x extra functionality (experts only)

Command (m for help):
```

## Quante partizioni?

Qui sotto vedete un tipico esempio per un PC con 2 dischi (/dev/hda e /dev/hdb) sul quale convivono

- Windows (/dev/hda1, /dev/hda5 e /dev/hdb1)
- Linux Red Hat 5.1 (/dev/hda6, /dev/hda4 e /dev/hdb4)
- Linux SuSE 6.0 (/dev/hda7)
- una partizione swap (/dev/hdb2) in comune ai due Linux
- il Boot Manager OS/2 (/dev/hda2)

```
Color xterm
Device Boot   Begin    Start    End    Blocks  Id System
/dev/hda1     1         1        261   2096451  6 DOS 16-bit >=32M
/dev/hda2     *        262     262     262     8032+   a OS/2 Boot Manager
/dev/hda3     263     263     775   4120672+  5 Extended
/dev/hda4     776     776    1027   2024190   83 Linux native
/dev/hda5     263     263     522   2088418+  6 DOS 16-bit >=32M
/dev/hda6     523     523     650   1028128+  83 Linux native
/dev/hda7     651     651     775   1004031   83 Linux native
Command (m for help):
```

```
Color xterm
Device Boot   Begin    Start    End    Blocks  Id System
/dev/hdb1     1         1        204   102784+   6 DOS 16-bit >=32M
/dev/hdb2     205     205     270    33264    82 Linux swap
/dev/hdb3     271     271    1462   600768   83 Linux native
/dev/hdb4    1024    1463    1652    95760   83 Linux native
Command (m for help):
```

Sul Linux Red Hat i file system sono montati come:

```
Color xterm
[root@Linux /root]# df
Filesystem      1024-blocks  Used Available Capacity Mounted on
/dev/hda6        995115    606744    336965     64% /
/dev/hdb4         92711     38120     49803     43% /home
/dev/hda4       1957973    409735   1447029     22% /disc2
/dev/hda7         971681    736651    184829     80% /disc3
[root@Linux /root]#
```

# Configurare il sistema già installato

## Creare nuovi utenti

Lo deve fare root:

- con `adduser` (`useradd`)
- tramite **YAST** (SuSE Linux)
- da X windows con `linuxconf` (Red Hat).

Questi sistemi creano la directory `/home/username` e modificano i files `/etc/passwd` e `/etc/group`

## Cambiare la password

Root può cambiare le password degli utenti col comando `passwd username`

Un utente usa per sé il comando `passwd`. Solo i primi 8 caratteri della password sono significativi.

La password (opportunamente criptata) si trova nel file `/etc/passwd` che dev'essere leggibile da tutti gli utenti. Questo può essere un punto debole per la sicurezza del sistema.

## Ho dimenticato la password di root!

Si ricorre al dischetto di boot di emergenza. Una volta avviato il sistema occorre montare il filesystem root del disco rigido ed editare il file `/etc/passwd`. La prima riga del file

```
root:Lhzgxn2ZiYGz2:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
ecc. ecc.
```

contiene la password criptata in mezzo a una coppia di 2 punti (`Lhzgxn2ZiYGz2`). Cancellarla (lasciare `::`, salvare il file e al prossimo avvio si potrà loggarsi come root senza password.

**Attenzione** Il dischetto di avvio contiene come editore `vi`. Se non sapete come salvare un file in `vi` potete avere qualche problema!

## Come si formatta un dischetto

- normalmente i dischetti formattati DOS sono OK (questo è il low-level formatting)
- se ci fosse ragione di formattarli, da root fare  
`# fdformat /dev/fd0H1440`

Dopo occorre creare un filesystem tipo ext2:

```
# mkfs -t ext2 /dev/fd0 e il dischetto è pronto
```

## Come si verifica un filesystem

Questa verifica viene sempre fatta automaticamente da Linux al boot della macchina.

A volte può esser necessario farla manualmente, p.es. dopo un crash serio. Occorre che la partizione del disco su cui si trova il filesystem da verificare non sia montata. Ad esempio col comando:

```
# /sbin/fsck /dev/hda1
```

si verifica la partizione /dev/hda1 ed eventualmente la si “ripara”.

## Come montare un filesystem

Per leggere i files su un disco (partizione) occorre prima *montarlo/a*: questo normalmente lo fa solo root. Esempi:

```
# mount -t ext2 /dev/hda5 /disc0 Monta la partizione /dev/hda5 su /disc0. La directory /disc0 deve esistere e -t ext2 identifica il tipo di FS
```

```
# mount -t msdos -o conv=auto /dev/fd0 /mnt/floppy Monta il floppy DOS e fa una conversione automatica dei files
```

```
# mount -t iso9660 /dev/hdc /mnt/cdrom Monta il CD IDE /dev/hdc.
```

## Il file /etc/fstab

Le informazioni sui filesystem da montare si trovano in */etc/fstab*. Ecco un esempio:

/dev/hda6	/	ext2	defaults	1
/dev/hdb4	/home	ext2	defaults	1
/dev/hdb3	/disc0	ext2	defaults	1
/dev/hda4	/disc2	ext2	defaults	1
/dev/hda7	/disc3	ext2	ro,defaults	1
/dev/hdb2	swap	swap	defaults	0
/dev/fd0	/floppy	ext2	user,noauto	0
/dev/hdc	/mnt/cdrom	iso9660	user,noauto,ro	0
none	/proc	proc	defaults	0

L'opzione **user** indica che il floppy e il CD possono esser montati da chiunque

In questo caso basta il comando `$ mount /floppy` per montare il dischetto.

## Posta elettronica

Se usate **Pine** questo normalmente è già configurato alla fine dell'installazione di Linux. Se il vostro PC è in rete potete già mandare e ricevere posta.

Se volete guardare la posta da casa, dovete essere collegati con PPP e potete utilizzare con Pine 2 metodi:

- Usate **fetchmail** per ricevere la posta dal mail server:

```
$ fetchmail -p protocollo -u utente mailserver
```

Il protocollo è in genere pop3 o imap. Con questo sistema la posta viene scaricata sulla vostra macchina, e poi potete leggerla con Pine.

Con pop3 la posta scaricata viene cancellata dal server.

- Configurate Pine per leggere la posta remotamente. Questo funziona bene solo con imap (ad esempio col mail server del CERN) dato che la posta non viene cancellata dal server.

Per far questo dovete modificare il file di configurazione **.pinerc**

Se siete collegati con PPP potete spedire posta senza problemi da Pine: tuttavia dovete editare a mano il nome del dominio in **.pinerc**. Inoltre se sul PC di casa avete come utente un nome "illegittimo" per il dominio, l'indirizzo **From:** del mittente nei vostri mail risulterà errato. In tal caso dovete anche dire a Pine di inserire la riga **Reply-To:** col corretto indirizzo per le risposte. Esempio

```
customized-hdrs=Reply-To: Giovanni.Carboni@cern.ch
```

## Esempio di .pinerc

```
# Updated by Pine(tm) 3.96, copyright 1989-1996 University of Washington.
#
##### Essential Parameters #####
# personal-name specifies your full name as it should appear on outgoing mail.
personal-name=Giovanni Carboni
# user-domain specifies the domain part of your return address on outgoing
user-domain=mail.cern.ch
nntp-server=news.utoverm.it
# List of SMTP servers for sending mail. You must have an SMTP server
# If blank on Unix, sendmail will be used.
smtp-server=
# inbox-path specifies the name/path/location of your INBOX.
# Normal Unix default is the local INBOX (usually /user/spool/mail/$USER)
inbox-path=
##### Collections, Folders, and Files #####
# incoming-folders are those other than INBOX that receive new messages.
# Folder syntax: optnl-label {optnl-imap-hostname}folder-path
# Use only if you filter incoming email into multiple files or receive
# email on several different machines.
#
incoming-folders="CERN" {carboni.mailbox.cern.ch/user=carboni}inbox
# folder-collections specifies a list of folder collections wherein saved
# messages are stored. The first collection is the default for saves.
# Collection syntax: optnl-label {optnl-imap-hostname}optnl-directory-path[]
# Example:
# folder-collections=Saved-Email {foo.bar.edu}mail/[],
# Widget-Project widget/[], <-- Valid only in Unix Pine
# Local-PC mail\[] <-- Valid only in PC-Pine
folder-collections=Mail/[],
CERN Server Folders {carboni.mailbox.cern.ch}mail/[]
```

## Rimettere l'orologio

In genere l'orologio hardware del PC (CMOS clock) viene regolato sul tempo universale UTC. L'ora di sistema (corretta anche per l'ora legale) viene determinata in base al fuso orario e mostrata dal comando `$ date`.

Windows invece usa il tempo locale nel CMOS clock, e quando cambia l'ora legale lo sposta (opzione disabilitabile).

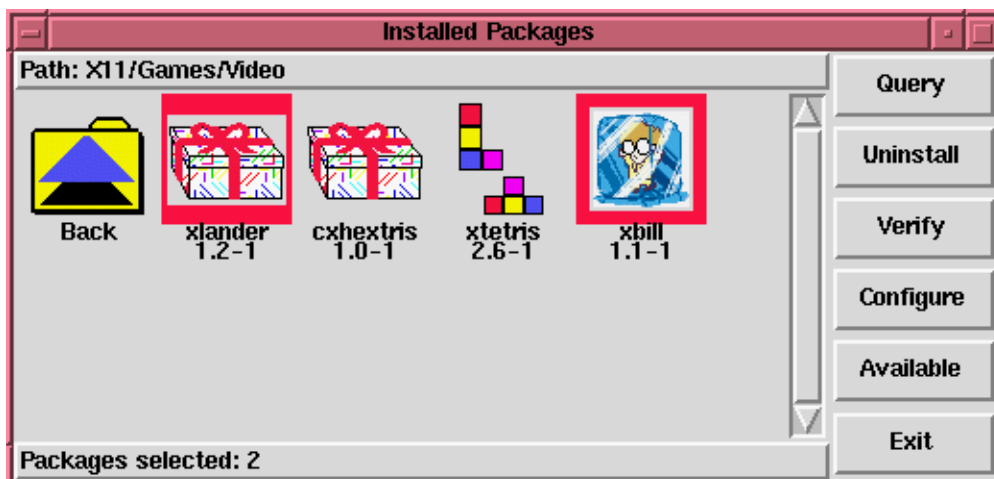
Per regolare l'orologio, se non volete usare `Linuxconf`, scrivete:

```
# clock --set --utc --date=hh:mm
```

Se non usate il tempo UTC omettete il `--utc`. La man page di `clock` sta a volte sotto `hwclock`. Con lo stesso comando potete anche cambiare la data.

## Installare/disinstallare programmi

- “a mano” dai files sorgente .tgz o .tar.gz. Il problema è che può risultare difficile ricordarsi cosa si è installato.
- tramite il Red Hat Package Manager **RPM**. RPM ha un’infinità di comandi:
  - **rpm -qa** elenca tutti i *pacchetti* installati
  - **rpm -ivh *cucco-2.0.45-1.rpm*** installa il pacchetto *cucco*
  - **rpm -e *CUCCO*** disinstalla il pacchetto *cucco* (il nome è in genere abbreviato)
  - **rpm -qil *CUCCO*** fa un sommario di tutti i files installati con *cucco*
  - **rpm -qp1 *cucco-2.0.45-1.rpm*** (prima dell’installazione) fa un sommario di tutti i files che saranno installati con *cucco-2.0.45-1.rpm*
- in Red Hat si può usare **glint**:



- in SuSE si usa **YAST**

Il vantaggio di RPM è che esso controlla tutte le interdipendenze e le compatibilità col software già installato. Nel caso rifiuta di installare o disinstallare (a meno che non lo si forzi). Lo svantaggio è che i pacchetti installati senza usare RPM non risultano nel database di quest'ultimo e questo può creare problemi. A volte si creano inoltre dei loops fastidiosi dai quali è difficile uscire.

RPM si può usare con qualunque installazione di LINUX (non solo Red Hat).

## Ricompilare il Kernel

E' bene saperlo fare. Ma perchè non va bene il Kernel della distribuzione?

- Fino a Linux 1.2 era necessario configurare il Kernel per il tipo di hardware dell'utente: esempio, schede di rete, unità nastro, CD-ROM con interfaccia non IDE-ATAPI (es. il Mitsumi), scheda sonora ecc. Si creava un *Kernel monolitico*. Attualmente con l'uso dei *moduli* ciò non è necessario.
- I moduli si caricano quando necessario (es. per accedere al CD) e addirittura si possono caricare automaticamente. Quindi le distribuzioni forniscono alcuni Kernel generici precompilati che in genere sono OK e non ci sono problemi.
- Può ancora esser necessario ricompilare il Kernel se abbiamo qualche pezzo di hardware speciale che non è supportato coi moduli, se vogliamo ottimizzare il Kernel per la nostra CPU (386, 486, 586, 686). Oppure se non riusciamo a spegnere il notebook.

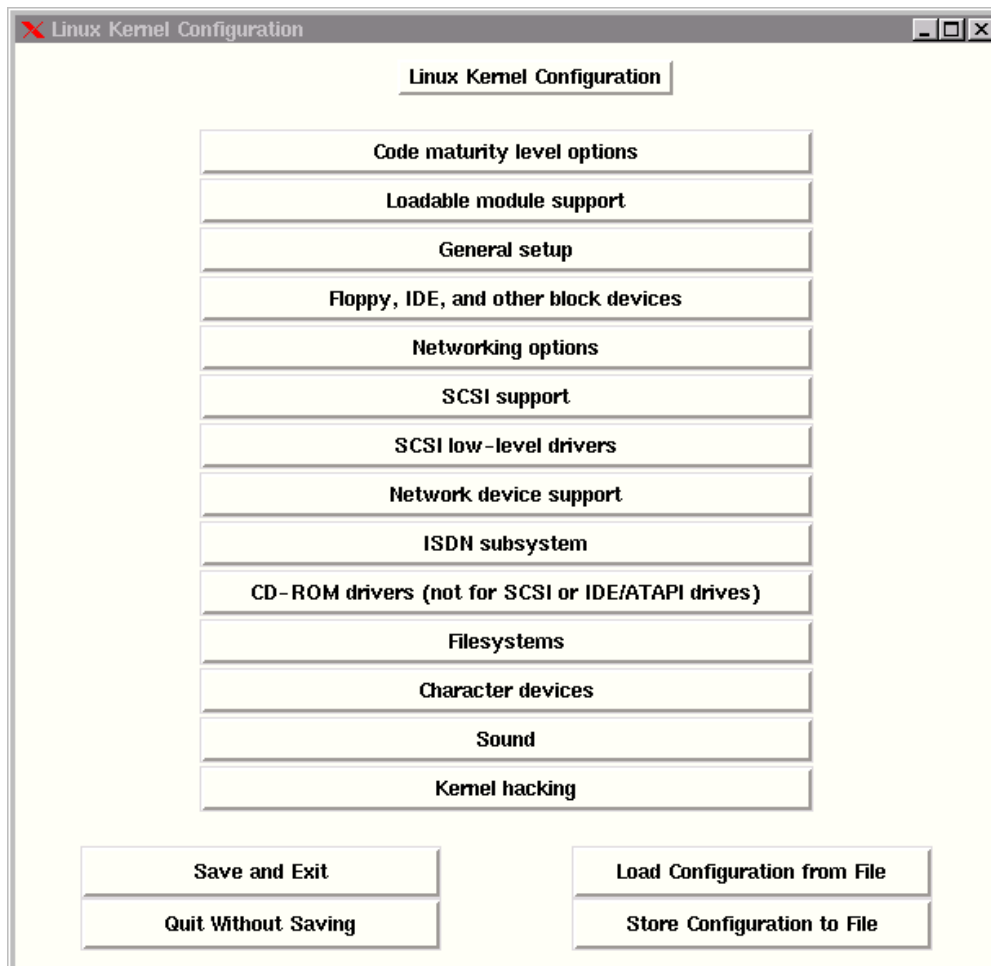
## Aiuto, non riesco a spegnere il notebook!

In tal caso occorre compilare nel Kernel l'opzione APM (Advanced Power Management).

Per compilare il Kernel occorre avere la sorgente (si può p.es. installare con RPM), che va in /usr/src/linux. La sorgente si trova sempre nella distribuzione ma a volte occorre richiedere esplicitamente la sua installazione.

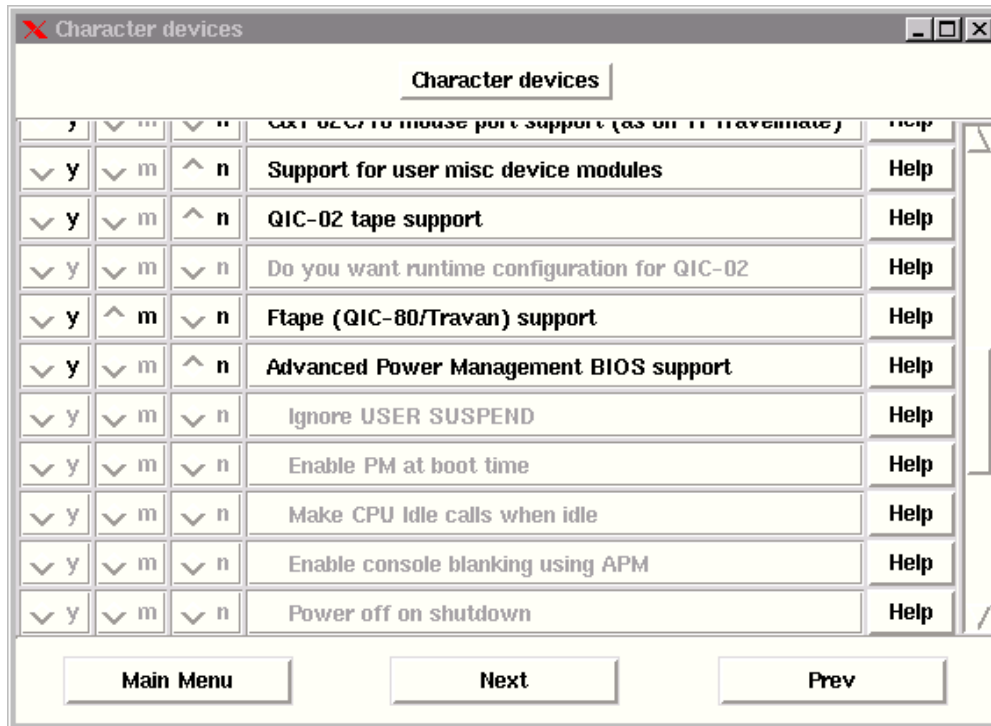
In tale directory si fa

`make config` o `make menuconfig` o `make xconfig` (quest'ultimo si fa da X)



e si risponde alle varie domande partendo dall'alto.

Il menu *Character Devices* contiene anche le opzioni per APM:



Alla fine salvare la configurazione (che va in `/usr/src/linux/.config`)

Poi fare

```
# make dep
# make clean
# make zImage
```

Il tempo per la compilazione varia da 45 minuti (486 66 MHz) a qualche minuto (AMD K6-2 300 MHz). Il Kernel compresso si trova in `/boot` (per Red Hat) e si chiama appunto **zImage**.

La compilazione del Kernel è molto stressante per l'hardware del vostro PC. Se il programma si pianta con un messaggio di errore **fatal signal 11** con tutta probabilità avete delle DIMM e SIMM (chip di memoria) difettose e non vi resta che cambiarle.

Informazioni su questo problema le trovate su <http://www.bitwizard.nl/sig11>.

## Come provo il nuovo Kernel? (Configurare LILO)

Per far partire Linux conviene usare il Boot Loader: LILO. Con LILO è facile provare diversi kernel. LILO si configura col file `/etc/lilo.conf`. Esempio:

```
# LILO configuration file
# Start LILO global section
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
# prompt
timeout=0
image=/boot/vmlinuz-2.0.35-2.CERN
    label=Linux
    root=/dev/hda6
    read-only
# Linux bootable partition config ends
# Windows95 boot
other=/dev/hda1
    label=Win
    table=/dev/hda
```

- la riga `boot=/dev/hda` dice che Lilo si andrà ad installare nel MBR del disco `/dev/hda`
- la riga `image=/boot/vmlinuz-2.0.35-2.CERN` dice dove si trova il Kernel.
- le 2 righe `label=Linux` e `label=Win` identificano le scelte possibili per l'utente (qui sono 2) all'avvio di LILO
- la riga `root=/dev/hda6` dice che la partizione root di Linux è la `/dev/hda6`
- la riga `other=/dev/hda1` dice che l'altro sistema (Win95) è installato su `/dev/hda1`

Alla fine fare (da root): `# lilo` per rendere LILO operativo. Si possono aggiungere altre righe per selezionare, volendo, Kernel differenti.

## X-Windows

X-Windows è l'interfaccia grafica di UNIX. Per Linux esiste l'implementazione gratuita XFree86 (versione attuale 3.3.3) XFree86 supporta un gran numero di schede grafiche (tra cui anche le ultime RIVA TNT con bus AGP)

Qualche problema si può avere coi notebook (chip video molto variabili), ma in genere si tratta di problemi sormontabili. Comunque, prima di acquistare un notebook, conviene verificare se è supportato (vedi pagine WEB sotto <http://www.linux.org>).

X-Windows si può configurare prima o dopo la prima installazione di Linux. Con l'installazione si copiano le varie librerie che vanno in `/usr/X11R6/lib`.

L'altro elemento fondamentale che viene installato è il **server X**. Questo è legato alla scheda video:

- XF86\_SVGA : Super VGA, supporta numerosi chip video
- XF86\_S3 : Per le schede con il chip S3
- .....

e in genere viene installato quello corretto in base alla scheda video montata. Il server X in genere è un link logico:

```
lrwxrwxrwx 1 root root 24 Jul 20 1997 X -> /usr/X11R6/bin/XF86S_VGA
```

Come tutto UNIX, anche X-Windows è altamente configurabile. Il file di configurazione si chiama **XF86Config**. In genere si trova come `/etc/XF86Config`. I vari programmi di configurazione per X-Windows modificano tale file.

## Configurare X-Windows

L'interfaccia di configurazione standard di X in Linux è alquanto spartana. Il programma si chiama `xf86config`.

`xf86config` vi chiederà notizie sul tipo di tastiera, mouse e monitor, per cui è bene che abbiate il suo manuale, in particolare per le varie frequenze di scansione. Se il vostro monitor non è fra quelli proposti e non ne avete il manuale, potete tentare anche con qualche tipo standard.

Altre domande riguardano la scheda video (comunque la funzione `autoprobe` funziona abbastanza bene.) Infine vi verranno proposte alcune soluzioni per il numero di bit colore, `bpp` e per la risoluzione (800x600, 1024x768 ecc.)

Potete scegliere più di una risoluzione HxV, dato che potete cambiarle ciclicamente da tastiera ( `Ctrl alt +`). Potete anche avere un desktop più grande del vostro schermo (`virtual resolution`) e navigarci dentro col mouse. Se la scheda video ha sufficiente memoria, sarà possibile anche avere 8 o 16 `bpp`. Questi però si scelgono all'avvio di X. Alla fine il file `XF86Config` viene salvato sul disco.

Alla fine, X-Windows viene fatto partire con il comando `$ startx`.

Se avete possibilità di scelta fra 8 e 16 `bpp` tenete presente che il server parte col default (sempre 8 `bpp`) ma potete usare:

```
$ startx -- -bpp 16
```

Se volete automatizzare questo comando una volta per tutte, usate il file `/etc/X11/xinit/xserverrc` dove scriverete:

```
exec X :0 -bpp 16
```

## Installazioni alternative

Esistono programmi più amichevoli di xf86config:

- Xconfigurator (Red Hat)
- SaX (SuSE)
- Xconfig (Metro-X)

Metro-X è un server commerciale. Un altro è Accelerated X. SaX e Xconfig permettono di centrare lo schermo in maniera interattiva (un vantaggio notevole nel caso di problemi di questo tipo)



## Window Managers

X-Windows ammette vari window managers. startx fa partire xinit che sceglie il window manager. La scelta avviene col file

`/etc/X11/xinit/xinitrc`

che verso la fine contiene una riga tipo

`exec /usr/local/bin/X11/fvwm95-2`

È possibile avere un proprio file `$HOME/.xinitrc`. Il file `xinitrc` (`.xinitrc`) legge i files

`$HOME/.Xresources`

`/usr/X11R6/lib/X11/xinit/.Xresources`

Esempio:

```
xterm*cursorColor: Orchid \\  
color_xterm*cursorColor: Orchid  
emacs.cursorColor: blue
```

Il Window Manager `fvwm95-2` rassomiglia esteticamente a Windows 95. Al solito è altamente configurabile, tramite i files:

`$HOME/.fvwm2rc95`

`/usr/X11R6/lib/X11/fvwm/.fvwm2rc95`

## Esempio di .fvwm2rc95

```
#
# .fvwm2rc95 example configuration file for fvwm95-2.0.41
#
#=====
#
#.....
```

### Cose da fare all'inizio

```
#
# Stuff to do at start-up
#
AddToFunc "InitFunction" "I" Exec xsetroot -solid cyan4 &

#+
+ "I" Module FvwmBanner
+ "I" Module FvwmButtons
+ "I" Module FvwmTaskBar
#+
+ "I" Module FvwmPager 0 3
+ "I" Exec xterm -ls -sb -sl 500 -bg black -fg white -fn 10x20 -title "Color xterm"
AddToFunc "RestartFunction" "I" Module FvwmButtons
+ "I" Module FvwmTaskBar
#+
+ "I" Module FvwmPager 0 3
```

## Menu pop-up

```
AddToMenu "Shells" "Shells" Title
+ "Xterm (9x15 font)%mini-term.xpm%" Exec xterm -sb -sl 500 -j -ls -fn 9x15 &
+ "Color Rxvt (VT100)%mini-term.xpm%" Exec rxvt -font 7x14 -ls -fat &
+ "White Xterm (9x15 font)%mini-term.xpm%" Exec xterm -sb -sl 500 -j -ls \
-fn 9x15 -fb 9x15bold -bg white -fg black -title "White xterm" &
+ "Color Xterm (8x13 font)%mini-term.xpm%" Exec xterm -sb -sl 500 -j -ls \
-fn 8x13 -title "Color xterm" &
+ "" Nop
+ "Large Xterm (10x20 font)%mini-term.xpm%" Exec xterm -sb -sl 500 -j -ls -fn 10x
+ "Large Rxvt (10x20 font)%mini-term.xpm%" Exec rxvt -font 10x20 -ls -fat &
+ "Large Color Xterm (10x20 font)%mini-term.xpm%" Exec xterm -sb -sl 500 \
-j -ls -fn 10x20 &
```

.....

## Barra dei pulsanti

```
#----- FvwmButtons
.....

# Define the buttons to use.....
*FvwmButtons (2x1) load NULL Swallow "xload" Exec nice -16 xload -nolabel \
-bg grey60 -update 5 -geometry -1500-1500
*FvwmButtons Xterm xtermblu.xpm exec "Xterm" xterm -geometry 80x23 -sb -sl \
500 -j -ls -fn 10x20 -bg black -fg white &
*FvwmButtons Wxterm xtermblk01.xpm Exec "XTerm" xterm -ls -sb -sl 500 \
-fn 9x15 -bg White -fg Black -title "White xterm" &
*FvwmButtons xterm xtermblk03.xpm Exec "XTerm" xterm -ls -sb -sl 500 \
-fn 8x13 -title "Color xterm" &
*FvwmButtons Emacs emacs.xpm Exec "Emacs" emacs &
*FvwmButtons Gv ghostview.xpm Exec "Gv" gv -geometry 900x740+10+0 &
*FvwmButtons xv xv.xpm Exec "xv" xv &
*FvwmButtons netscape netscape.xpm Exec "Netscape" /usr/bin/netscape &
*FvwmButtons Xfig xfig.xpm exec "Xfig" xfig &
*FvwmButtons kill rbomb.xpm Destroy
*FvwmButtons(2x1, Swallow(UseOld) "Desktop" 'FvwmPager 0 0')
```

## Taskbar stile Win95

```
#----- FvwmTaskBar
Style "FvwmTaskBar" NoTitle,BorderWidth 4,HandleWidth 4,Sticky,StaysOnTop, \
WindowListSkip,CirculateSkip

*FvwmTaskBarGeometry +0-0
*FvwmTaskBarFore Black
..... ecc
```

## Sicurezza

### Sostituirsi a root

A volte sarebbe comodo, come utente normale, poter eseguire certi comandi riservati a root, ad es. `/sbin/shutdown`. Esistono vari metodi, il più rischioso è di dare a tutti il permesso di esecuzione del comando, che fra l'altro in certi casi non funziona. Ad esempio, anche se digitate

```
# chmod 755 shutdown per cui  
-rwxr-xr-x  1 root  root 12932 Jul 26  1998 shutdown*
```

tutti dovrebbero poter fare shutdown, invece il programma stesso non accetta che root come esecutore.

In tal caso potete usare due metodi:

- Settare da root il *suid bit*. # `chmod 4755 shutdown`  
`-rwsr-xr-x 1 root root 12932 Jul 26 1998 shutdown*`
- Usare il programma `sudo` che richiede di inserire in una lista nel file `/etc/sudoers` il nome degli utenti abilitati a eseguire shutdown (o qualunque altro comando). Gli utenti abilitati danno il comando `$ sudo /sbin/shutdown` (verrà loro richiesta la password).

## SSH

rlogin, telnet, ftp, ecc. hanno seri problemi di sicurezza in quanto le password sono trasmesse via rete in chiaro.

È sufficiente un programma *sniffatore di pacchetti ethernet* installato su una macchina in rete per scoprire tutte le password delle macchine sulla stessa sotto-rete.

Fortunatamente per UNIX (e Linux) esiste **SSH: Secure Shell**. Il pacchetto di programmi è disponibile gratis per uso non commerciale. SSH stabilisce una comunicazione completamente criptata (col metodo della chiave pubblica) con 128 bit.

SSH contiene essenzialmente vari clienti e **un server, sshd:**

- ssh: secure shell, esegue comandi di shell su un host remoto
- slogin: analogo a rlogin, permette il login remoto
- scp: secure-copy, serve a copiare file da e su altre macchine
- ssh-keygen: serve a generare le chiavi per permettere accesso remoto su altre macchine senza dover fornire la password
- sshd: il server

L'installazione di SSH è molto semplice se avete il file `*.rpm`, altrimenti fate da root

```
# configure && make && make install
```

 e il tutto finisce in `/usr/local`.

Se volete potervi loggare dall'esterno sul vostro PC, bisogna che il server sia attivo. Per far questo dovrete farlo partire all'avvio del PC.

La sintassi di `scp` con cui Carlo copia il file `cucco` dalla macchina `dotto.roma2.infn.it` sulla propria è:

```
$ scp alice@dotto.roma2.infn.it:~/cucco .
```

(Alice su dotto è la proprietaria del file in questione)

## Limitare gli accessi via rete

Se avete installato SSH, potreste voler limitare gli accessi via ftp e telnet alla vostra macchina. Un metodo è **disabilitare totalmente ftp, telnet, rlogin**

Per far questo dovete modificare il file `/etc/inetd.conf`, commentando le righe corrispondenti:

```
#
ftp      stream  tcp      nowait  root    /usr/sbin/tcpd  wu.ftpd -d
telnet   stream  tcp      nowait  root    /usr/sbin/tcpd  in.telnetd
#
```

Col comando `# ps aux | grep inetd` trovate il PID di inetd, Quindi lo fermate e lo fate ripartire:

```
# kill -HUP pid
```

Una limitazione selettiva si fa editando i files:

- `/etc/hosts.allow`
- `/etc/hosts.deny`

Queste limitazioni valgono per tutti gli utenti del PC. Si possono personalizzare con i files

- `$HOME/.rhosts` (per rlogin)
- `$HOME/.shosts` (per slogin)

## Esempio di /etc/hosts.allow/deny

```
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided by
#               the '/usr/sbin/tcpd' server.
#
in.rlogind     : localhost  galileo.pi.infn.it   .unipi.it hpplus09.ce
rn.ch
in.telnetd    : localhost  .unipi.it   .ccd.utovrm2.it  .cern.ch
in.ftpd       : localhost  .unipi.it   .cern.ch  dotto.roma2.infn.it
# End of hosts.allow.

#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
ALL : ALL
# End of hosts.deny.
```

## Esempio di \$HOME/.shosts

```
galileo.pi.infn.it
hpplus.cern.ch
slip18.unipi.it
brontolo.roma2.infn.it
pisolo.roma2.infn.it
```

# Interagire con Windows



*Virtual Network Computing*



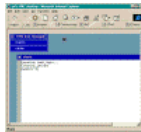
- Home
- Screenshots
- Why is it free?
- Getting started
- Documentation
- FAQ
- Download
- Keep in touch
- Contributed
- Want to help?
- VNC people
- Search
- About Us

## VNC screenshots

These shots show very simple desktops being accessed from a number of different platforms. Click on the thumbnails for larger images.



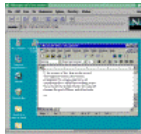
An X desktop being viewed from a native PC viewer.



An X desktop being viewed from Microsoft Internet Explorer on a PC.



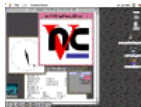
A Windows desktop being viewed from a native X viewer.



A Windows desktop being used from within Netscape on a Unix machine.



A Unix desktop being accessed from a Macintosh using Java.



A Unix desktop being accessed from a native Macintosh viewer.

A Windows desktop being accessed from a Macintosh using

<http://www.uk.research.att.com/vnc/>