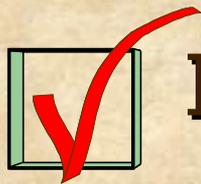


Ing. Gabriele MONTI



"Il Sistema Operativo Linux"

ITI B.Pascal, Cesena 2006



Ing. Gabriele MONTI
prof@ingmonti.it

Fondamenti sul
Sistema Operativo
Linux e suo utilizzo
in rete

Modalità di distribuzione del software

La licenza è un contratto

- un contratto è la "legge" che regola i rapporti fra i contraenti
- Tipi di licenza
 - Software commerciale (proprietario)  
 - Shareware e trialware
 - Freeware  
 - Free software ("libre software")
 - Altri software open source[®]     
 - Software di pubblico dominio (es. BSD)  



Open source

Software il cui codice sorgente è "aperto", visibile a chi ne ha il diritto

Science is ultimately an Open Source enterprise. [...] The scientific method rests on a process of discovery and [...] Replication [...] that [...] is not possible unless the source is **shared**: the hypothesis, the test conditions, and the results [...] that lead to [...] discovery must be served by sharing information [...] pollinating the ideas of others

Where scientists talk of replication, Open Source programmers talk of **debugging**. Where scientists talk of discovering, Open Source programmers talk of creating. Ultimately, the Open Source movement is an extension of the scientific method [...] .Computer science has only one means of enabling peers to replicate results: share the source code

da "Open Sources: Voices from the Open Source Revolution",
Introduction by DiBona, Ockman, Stone



Sorgente "visibile" e open source[®]

- Licenza restrittiva commerciale
 - Vedi contenzioso MS - EU
- Licenza GPL (free software)
 - GNU, Linux, OpenOffice
- Licenze "BSD like" (pubblico dominio)
 - FreeBSD, NetBSD,
- Altre licenze "open source[®]"
 - Mozilla



Richard Stallman (RMS); Movimento "free software":



Stallman, lavori prima di GNU (al MIT):

- EMACS, compilatori

FSF (**Free Software Foundation**, www.gnu.org)

- (GNU manifesto 1985)
- compilatori GNU



Newsgroups: net.unix-wizards,net.usoft
Subject: new UNIX implementation
Date: Tue, 27-Sep-83 12:35:59 EST
Organization: MIT AI Lab, Cambridge, MA

Starting this Thanksgiving I am going to write a complete **Unix-compatible** software system called **GNU** (for Gnu's Not Unix), and give it away **free to everyone** who can use it. **Contributions** of time, money, programs and equipment are greatly needed.

(..) GNU will be a **kernel** plus all the utilities needed to write and run C programs: editor, shell, **C compiler**, linker, assembler, and a few other things (..)

Why I Must Write GNU

I consider that the golden rule requires that **if I like a program I must share it with other people** who like it. (..)

So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to **get along without any software that is not free**. (..)

How **You** Can Contribute (..)



The GNU General Public License (GPL or copyleft)

the GNU General Public License is intended to guarantee your freedom to share and change free software (..)

When we speak of free software, we are referring to **freedom, not price**. (..)

make restrictions that forbid anyone to deny you these rights (..) if you distribute copies of such a program, whether gratis or for a fee, you must **give the recipients all the rights that you have** (..)

If the software is modified by someone else (..) know that what they have is not the original (..) so that any problems (..) will not reflect on the original **authors' reputations**. (..)

any **patent** must be licensed for everyone's free use or not licensed at all (..)

La GPL è attualmente è alla versione 2 (vers. 3 in arrivo)

LGPL Lesser (o **Library**) General Public License

Permette di usare il software libero anche in applicazioni "chiusa", con limitazioni



GPL

- Chi ottiene un software libero ha il diritto di:
 - Ottenerne il sorgente
 - Usarlo
 - Modificarlo
 - Distribuirlo come meglio crede, assieme al sorgente, che devono comprendere anche le proprie modifiche, o con indicazione su come trovare il sorgente
 - Venderlo al prezzo che vuole
- Programmi chiusi scritti interamente da un soggetto, ma che usano componenti di software libero, come per esempio un Sistema Operativo, possono essere distribuiti commercialmente, anche senza sorgente.



GPL

- Chi ottiene un software libero **non** ha il diritto di:
 - renderlo "chiuso", cioè incorporarlo in un proprio programma senza rendere pubblici i sorgenti delle proprie modifiche
 - includere **porzioni** di free software in programmi commerciali che non includano anche il sorgente
 - Restringere verso gli altri i diritti ottenuti con il software
 - Apporre sul programma brevetti che ne limitino la libertà di circolazione
 - Attribuire all'autore originario le proprie modifiche
 - Spacciare per proprie le modifiche altrui



Open Source

- Vantaggi
 - "peer review"
 - velocità di evoluzione
 - prezzo di acquisto
- Svantaggi
 - \$\$\$ (quelli guadagnati!)
 - Riservatezza
- Aspetti dibattuti
 - **sicurezza** (security by obscurity vs. security by review)



Free software != Freeware

Free in Free software = **libero**

- "free as in freedom"
- il free software è sempre open source
- il free software non è sempre gratis
(es. distribuzioni Linux)

Free in Freeware = gratis

- il freeware di solito è closed source
- il freeware è sempre gratis
(es. Internet Explorer è freeware per chi ha Windows
licenziato)





Linus Torvalds, autore del **Kernel Linux:**

Al tempo studente Univ. Helsinki (Finlandia)
Oggi al consorzio Open Source Development Lab



Festa per il rilascio del kernel 1.0



Giorno dopo



Linux

Messaggio con cui Linus cerca info su Unix (Posix)

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: **comp.os.minix**

Subject: Gcc-1.40 and a posix-question

Message-ID:

<1991Jul3.100050.9886@klaava.Helsinki.FI>

Date: **3 Jul 91** 10:00:50 GMT

Hello netlanders, Due to a project I'm working on (in minix), I'm interested in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice.

Minix = S.O. NON libero, Unix like, allegato in sorgente in un libro di testo (autore Tanenbaum)

The Linux kernel is © 1993 by Linus Torvalds, **GPL**



Messaggio che annuncia il rilascio del kernel vers. 0.02: 5 Ottobre **1991**,
Linus scrisse nel newsgroup comp.os.minix:

"Do you pine for the nice days of Minix-1.1, **when men were men and wrote their own device drivers**? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs?

Are you finding it frustrating when everything works on Minix? (..) I'm working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution.

Oggi

Nel Febbraio 2006 la versione stabile del kernel è la **2.6.15** sono mantenute anche la versione 2.0, 2.2, 2.4 (vedi <http://kernel.org/>)



Polemica con Tanenbaum

From: ast@cs.vu.nl (Andy Tanenbaum)

Newsgroups: comp.os.minix

Subject: **LINUX is obsolete**

Date: 29 Jan 92 12:12:50 GMT

1. **MICROKERNEL** VS MONOLITHIC SYSTEM

I still maintain the point that designing a monolithic kernel in 1991 is a fundamental error. Be thankful you are not my student. You would not get a high grade for such a design :-)

2. **PORTABILITY**

I think it is a gross error to design an OS for any specific architecture, since that is not going to be around all that long.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Subject: Re: LINUX is obsolete

Date: 29 Jan 92 23:14:26 GMT

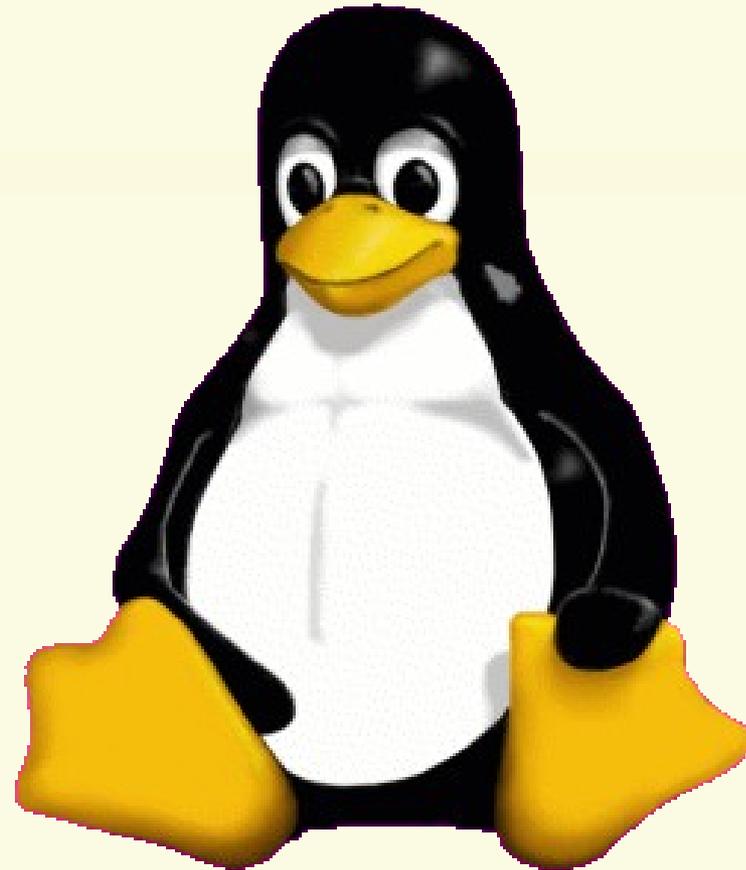
Organization: University of Helsinki

[..] linux still beats the pants of minix in almost all areas

[..] From a theoretical (and aesthetical) standpoint linux loses



Il logo scelto da Linus (chiamato "Tux") per la versione 2.0 del kernel Linux , disegnato da Larry Ewing, è questo pinguino.



Disegnato interamente con "The GIMP" su un 486 con Linux

Linux è solo il kernel del S.O. GNU/Linux



Linux ovunque



PicoTux: Linux nel
connettore di rete



Nokia 770: telefono, PDA



Axis: Web server
e telecamera



Linux Documentation Project

- LDP: documentazione in Inglese
 - <http://www.linuxdoc.org>
 - <http://ldp.iol.it/> (mirror sito LDP)
- ILPD: documentazione in italiano
 - <http://www.pluto.linux.it/ildp/>
- Daniele Giacomini. Appunti di informatica libera (già Appunti Linux)
 - <http://www.pluto.linux.it/ildp/AppuntiLinux/index.html>



Unix

Ken Thompson, Dennis Ritchie, AT&T "Bell Labs", **1969**

– Ritchie: compilatore C

- Scritto in C
- Portabile
- Versione "Berkley" (oggi Free BSD, OpenBSD e NetBSD)
- "Frammentazione" (molti "dialetti" commerciali)



CARATTERISTICHE DI UNIX

- filesystem "montabili"
- multiprogrammato
- multiutente
- networking TCP/IP
- dispositivi visti come file
- Specifiche POSIX
- ambienti grafici X Window.
- Ambienti a finestre Motif, OpenLook



UNIX DI OGGI

- SCO Unix
 - Generale, è l'"erede" di Unix originale
- System X
 - Apple su Mac (CPU PowerPC, da poco anche Intel)
- AIX
 - IBM su mainframe
- Solaris
 - Sun su workstation Sun (CPU Sparc)
- HP UX
 - Su workstation HP (CPU PArisc e Intel Itanium)
- Irix
 - Su workstation Silicon Graphics (CPU Mips e Intel)



Caratteristiche di GNU/Linux

- **Unix - Posix compatibile**
- **multiprogrammazione:** processi, thread
- **multiutenza:** account diversi, sicurezza nell'accesso al computer, accounting, logging, sicurezza nell'accesso ai file
- **multipiattaforma** (Intel PC, Sparc, Alpha, PowerPC, ARM, , PlayStation ..)
- **multiprocessore** (SMP, ovvero per computer che contengono fino a 16 processori (CPU))
- **sistema distribuito** (clustering: Beowulf, Open Mosix ed altri)



- **networking**: suite di protocolli TCP/IP, IPv6
- **Server di rete**:
 - NFS per collegarsi ad altri sistemi Unix
 - SMB (**Samba**) per collegarsi a Win9X, Win NT
 - Protocolli Internet (**HTTP, FTP, Telnet, NNTP, ..**)
- Assegnazione di "**quote**" di disco
- Contabilizzazione utenti (accounting)
- Librerie condivise, linkate dinamicamente
- **Memoria virtuale**: tecnica demand - page per gli eseguibili: caricate in memoria solo le parti di un programma usate al momento
- supporto dei terminali su RS232 e rete (telnet)



- terminali virtuali (diversi terminali sullo stesso computer Alt-F1, Alt-F2, .. oppure con Alt – freccette) (ora spesso disabilitati)
- Ambiente grafico a finestre X Window (XOrg86)
- **"Desktop environment"**
(oggetti grafici, standard di comunicazione fra oggetti, drag & drop, "skins", programmi di sistema, browser..)
 - I principali: **gnome** e **KDE**
- Journaling file system (ext3, ReiserFS e JFS IBM)



Distribuzioni GNU/Linux

Aziende o team di programmatori o singoli compilano, provano ed "impacchettano" il S.O. ed i programmi applicativi.

DISTRIBUZIONE LINUX = il kernel del Sistema Operativo (**Linux!**) + un programma di installazione e configurazione + programmi di sistema GNU + programmi applicativi già compilati e provati insieme



Distribuzioni GNU/Linux

✓ Non commerciali

– Non hanno una versione "in vendita"

– Debian

- Solo programmi GPL

 - In più diversi "repository" con programmi non GPL

- Knoppix

 - **Live**, Germania, Klaus Knoppler

 - Molti programmi, anche non GPL

 - GameKnoppix, eduKnoppix

- Ubuntu



kubuntu



edubuntu

 - Sudafrica, Shuttleworth



Distribuzioni GNU/Linux

✓ Non commerciali

– Dynebolic

- Multimediale



– Gentoo

- Distribuita **in sorgente**

- Distribuito un "nucleo" piccolo con un kernel precompilato ed accesso ad Internet
- Scaricati i sorgenti del grosso del sistema, compilati in automatico sul computer del destinatario
- "Personalizzazione" automatica
- Aggiornamento continuo via Internet



Distribuzioni GNU/Linux

✓ Commerciali

- Possono essere vendute commercialmente, possono contenere software chiuso.
- Di solito hanno una versione meno "impegnativa" che comprende solo software libero e che si può scaricare liberamente.



Distribuzioni GNU/Linux

✓ Commerciali

– RedHat



- Per server

- solo versioni "professional", per server
- Solo a pagamento

- Fedora Core



- Sperimentale, seguita ed indirizzata da Red Hat
- Gratuita

– Mandriva (ex Mandrake (Francia) + ex Conectiva (Brasile))



- Anche versioni per desktop
- La "limited edition" ha solo Sw libero, è gratuita e scaricabile



Distribuzioni GNU/Linux

✓ Commerciali

– SuSe (Novell)



- Ottima per desktop, in sostituzione di Windows
- Scaricabile la versione desktop, che ha pochi server

– Slackware

- USA, William Wolkerding
- "One man distribution", costo simbolico
- Per uomini duri

■ Molte altre

- Whoppix, Netkit, ITIX, morphix, yellow dog(per Mac), red flag

(cinese).....



Strumenti Open Source (in Linux ed altri S.O.)

✓ GPL (free software)

- Tool per lo sviluppo GNU (gcc, gdb, make ..)
- Linux (kernel)
- gnome, KDE (ambienti grafici per Unix)
- mySQL, Postgres (database)
- OpenOffice (già StarOffice)

✓ non GPL

- Mozilla (Netscape)
- Interbase (database)
- Strumenti per Java (IBM)
- StarOffice (Sun, alcune parti sono closed source)



Strumenti Open Source (in Linux ed altri S.O.)

- GPL (free software)
 - Linux (kernel)
 - Tool per lo sviluppo GNU (gcc, gdb, make ..)
 - gnome, KDE (ambienti grafici per Unix)
 - mySQL, Postgres (database)
 - OpenOffice (già StarOffice)
- NON GPL
 - Interbase (database)
 - Strumenti per Java (IBM)



Strumenti non Open Source in Linux

- Licenze "miste"
 - Es. gratis solo per sviluppo di software GPL, da pagare se usato per sviluppare software commerciale o sotto Windows
- Licenze commerciali chiuse
 - Oracle (database)
 - DB2 (database IBM)
 - Lotus Notes – Domino (gestione documentale e workgroup)



Installazione

- Installazione automatica delle distribuzioni
 - Fedora (Red Hat), Mandriva: fanno uso di file eseguibili in formato **RPM** (è indicata la dipendenza dei programmi da altri file (es. librerie))
 - SuSE: programma yast (**y**et **a**nother **s**etup **p**rogram)
 - Debian: file .deb
 - Slackware: suo programma di installazione; per gestire le dipendenze dei file: **NIENTE**



Aggiornamento

- Aggiornamento automatico
 - Tutte le distribuzioni hanno qualche forma di aggiornamento automatico via Internet
 - Fedora: yum
 - Debian: apt
 - SuSE; yast
 - Mandriva: ?
 - Slackware: upgradepkg



Preparazione degli hard disk

- Partition table, nel primo settore dell'HD
- Creare il "posto" per una partizione
 - Se il disco non è vuoto:
 - Compattare (**defragment**)
 - Modificare le partizioni esistenti
 - **fips** (programma MS DOS, the First nondestructive Interactive Partition Splitting: crea una nuova part. primaria)
 - **Partition resizer** (può anche ingrandire o ridurre una partizione, non solo suddividere)
 - **Partition magic** (programma commerciale, più completo)



■ Creare le partizioni

- Per le partizioni DOS-Win usare **fdisk** di DOS (FORMAT funziona bene solo se si usa questo fdisk)
- Per le partizioni Linux usare **cfdisk**, **fdisk**, **disk druid**, **qtparted ..**

■ "Formattare"

- Partizioni DOS: A:> FORMAT C:
- Partizioni Linux: # mkfs -t ext2 -c /dev/hdb2

!! per creare una partizione usare l'fdisk del Sistema Operativo con cui la si formatta !!



I "nomi" delle partizioni

- Linux

- /dev/hda è il primo disco fisico IDE/ATA, /dev/hdb il secondo, /dev/hdc il terzo ..
 - /dev/hda1 è la prima partizione primaria
 - /dev/hda1 .. /dev/hda4 sono tutte le partizioni primarie
 - /dev/hda5 è la prima partizione secondaria
 - /dev/hda6 è la seconda partizione secondaria
 - /dev/hda7 .. sono le partizioni secondarie successive



Processo di boot "standard"

- MBR (master boot record)
 - Primo settore del **primo** disco (hda)
 - contiene la tabella delle partizioni ed il primo programma che viene eseguito
- Boot sector
 - Ogni partizione ha un primo settore, di boot, nel quale può essere scritto un programma da far partire al boot (se qualcuno lo fa partire dall'MBR)



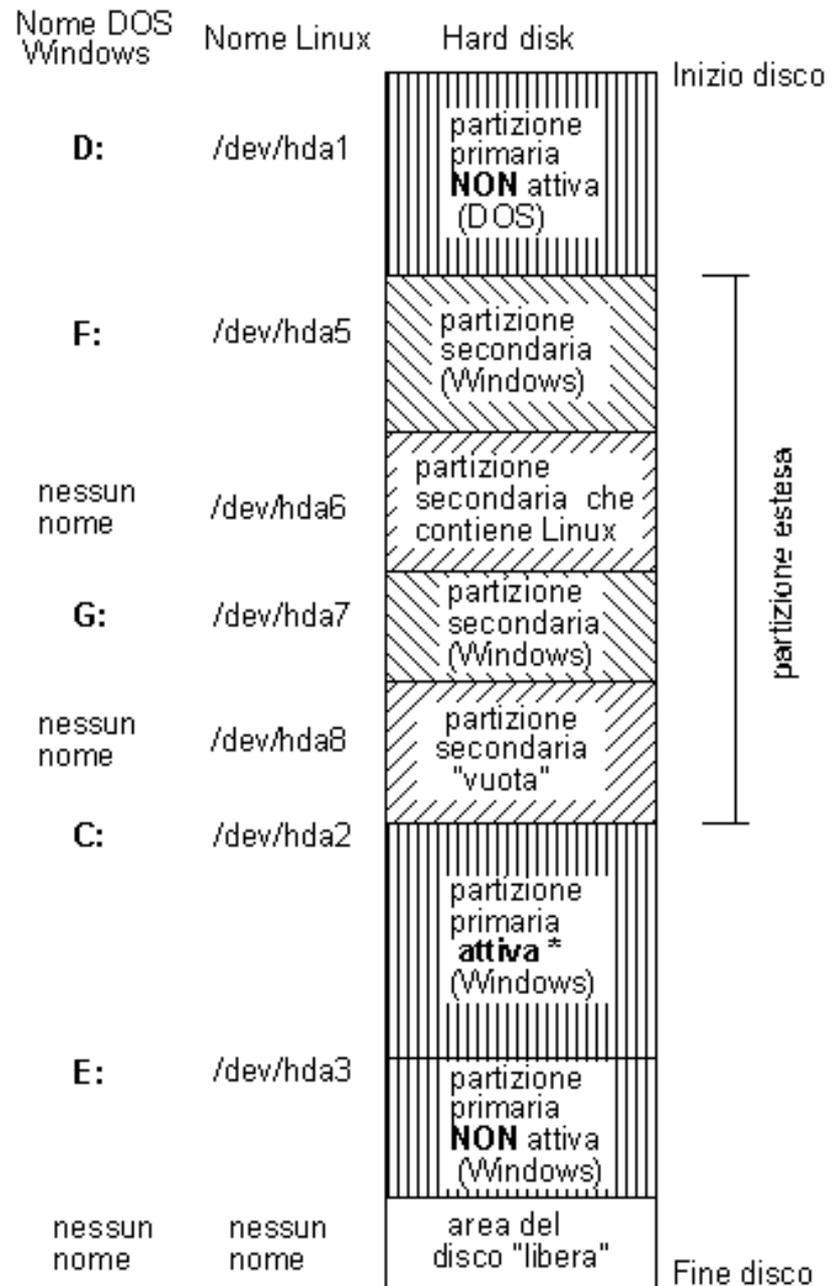
I "nomi" delle partizioni

- Windows

- C è il nome dell'unica partizione primaria attiva
 - I nomi delle partizioni successive sono dati in ordine alfabetico seguendo l'ordine
 - Prima le partizioni primarie non attive, in ordine "fisico"
 - Poi le partizioni secondarie, in ordine "fisico"
- Da Windows XP è possibile assegnare il nome che si vuole ad una partizione (C escluso!)



Convenzioni sui "nomi" delle partizioni



Processo di boot "standard"

- MBR contiene un programma che fa partire il S.O. voluto, lanciandolo dal boot sector dove è stato installato
 - Indipendente dal S.O.
 - Ordine di installazione arbitrario
 - Il contenuto di una partizione riguarda solo UN S.O., che ha il suo boot loader nella stessa partizione dove ci sono anche i suoi programmi e dati



Processo di boot Windows

- MBR contiene la tabella delle partizioni ed un programma che "salta" alla p.p.a.
 - Il processo di boot parte comunque dalla partizione primaria e attiva (p.p.a.)
- Multiboot
 - Dalla prima partizione primaria attiva si può "saltare" alla partizione che contiene effettivamente il S.O. Windows voluto
- Installare in ordine "cronologico"
- Non cancellare la partizione primaria attiva



Boot loader

- Chiede il S.O. voluto e ne lancia il kernel
 - Di solito si installa nell'MBR
 - **lilo** (**linux loader**)
 - Limitazione: non può accedere ai cilindri dell'H.D. oltre il 1024 (? rimossa ?)
 - Si installa da Linux, può partire da floppy, MBR, boot sector
 - **GRUB** (**Grand Unified Boot Loader**)
 - Il più completo e complicato
 - Si installa da Linux, parte da floppy, MBR, boot sector, CD ROM



Boot loader

- **GAG** (indipendente dal S.O.)
 - Lancia fino a 9 sistemi, ognuno con il **su**o bootloader
 - Grafico, facile
 - Carica sempre **altri** boot loader
 - L'altro boot loader fa partire il S.O.
 - Può "nascondere" partizioni primarie a S.O. MS
 - Poco flessibile
 - Si installa da floppy, parte da floppy o MBR
 - Qualcuno lo sa far partire da CDROM



Un mio vecchio file di configurazione di lilo (**/etc/lilo.conf**)

```
boot = /dev/hda
```

```
timeout = 50
```

```
linear
```

```
prompt
```

```
default = win
```

```
vga = normal
```

```
read-only
```

```
map=/boot/map
```

```
install=/boot/boot.b
```

```
message=/boot/message
```

```
image = /boot/vmlinuz-2.2.14-12
```

```
label = linux
```

```
root = /dev/hdb6
```

File immagine kernel



Partizione di boot



(/etc/lilo.conf)

```
image = /AltBoot/gaKernelAMD  
label = gamon  
root = /dev/hdc5
```

Un kernel compilato da me

Che risiede su un altro directory

In un altro h.d. !

```
other = /dev/hda  
label = win
```

E' la partizione primaria attiva!

Fa partire Windows

```
other = /dev/hdb1  
label = dos
```

Partizione primaria attiva di un altro disco

Fa partire MS DOS



per rimettere lilo nell'MBR dopo che è stato cancellato, fare boot da Linux e:

```
# lilo
```

questo serve anche per scrivere nell'MBR dopo le variazioni alla configurazione di lilo (**bisogna** farlo per applicare le modifiche)

per disinstallare lilo dall'MBR:

```
# lilo -u
```

Per toglierlo e far partire Windows se non c'è più Linux: fare il boot con un disco MS DOS di ripristino e lanciare fdisk con opzione MBR:

```
A:> fdisk /MBR
```



Interfaccia Unix con l'utente

- CUI (Character User Interface)
 - terminali e terminali virtuali
- GUI (Graphical User Interface)
 - X Window
 - gnome
 - KDE



GRUB

- Flessibile, complicato, "non Linux"
 - Può far partire il S.O. anche dalla rete
 - Può "nascondere" partizioni primarie a S.O. MS
- Nomi
 - (hd0, 1)
 - primo hard disk, seconda partizione primaria
 - (hd2, 4)
 - Terzo hard disk, prima partizione estesa



GRUB

■ Installazione, da Linux

– `grub-install /dev/hda`

- Installa nell'MBR del primo disco, l'immagine di boot è nel directory root (/)

– `grub-install /dev/hda5`

- Installa nel boot sector della prima partizione estesa, l'immagine di boot è nel directory root (/)

– `grub-install --root-directory=/boot /dev/hda4`

- Installa nella prima partizione estesa, l'immagine di boot è nel directory root (/)



GRUB

- Installazione, da un "grub boot floppy"
 - Creare un floppy di boot con grub
 - Boot con il nuovo floppy
 - grub> _
 - » Si va in una linea di comando "grub"
 - Per configurare dare i comandi giusti a linea di comando
 - L'uso di grub-install è considerato meno sicuro



GRUB file di configurazione

- File di configurazione
 - In Fedora è /etc/grub.conf
- ESEMPI:
 - Per far partire Linux dal secondo HD:
 - `# For booting GNU/Linux`
 - `title GNU/Linux`
 - `kernel (hd1,0)/vmlinuz root=/dev/hdb1`
 - Fa partire Linux dal secondo hard disk



GRUB file di configurazione

- File di configurazione

- Per far partire Windows dalla prima partizione:

- ```
For booting Windows NT or Windows95
title Windows NT / Windows 95
boot menu root (hd0,0)
makeactive chainloader +1
For loading DOS if Windows NT is installed

chainload /bootsect.dos
```
  - Con il comando **map** si può scambiare l'hard disk 0 con un altro, in modo da far partire Windows anche dal secondo disco



# Terminali virtuali

---

- Linux ha diversi terminali (console) sullo stesso computer e si può collegare a terminali remoti
- Da ciascuno si può iniziare una sessione (fare il "login") come un diverso utente
- Si passa da uno all'altro con Alt-F1, Alt-F2 o con Alt-Freccia destra o Alt-Freccia sinistra
- i programmi lanciati nei diversi terminali funzionano "contemporaneamente"
- Oggi alcune distribuzioni bloccano queste console



# Le "shell" del S.O.

---

- Interprete dei comandi dell'utente.
- Diverse shell => Diversi linguaggi di comando
- La più usata è bash (Bourne Again Shell)



# Xwindow e interfaccia grafica

---

---

desktop environment    **gnome** , **KDE**

---

window manager        fwm, kwm, Window maker,  
enlightenment, sawfish,  
Motif (Unix)

---

Xwindow                server X (Xorg86)

---

Scheda video            Driver della scheda



Solo il server X dipende dalla scheda video, gli altri servizi che stanno “sopra” no

## **Xwindow**

- un insieme di primitive grafiche, non di "oggetti"

## **window manager**

- gestione di oggetti grafici finestre, bottoni, sfondi (detti "widget")

## **desktop environment** (ambiente integrato):

- drag and drop, barre di controllo, associazioni di nomi di file, utility, file manager, text editor, file viewer, browser



# Runlevel

---

|            |                        |
|------------|------------------------|
| Runlevel 0 | Halt                   |
| Runlevel 1 | Single user            |
| Runlevel 2 | Multiuser, Testo       |
| Runlevel 3 | Multiuser, Testo, rete |
| Runlevel 4 | Non usato              |
| Runlevel 5 | Multiuser, Grafica     |
| Runlevel 6 | Reboot                 |

- Runlevel switch

```
init <runlevel>
```



- Per uscire dalla visualizzazione X senza chiudere la sessione
  - Ctrl - Alt - F1 va al primo terminale modo testo
  - Ctrl - Alt - F2 al secondo ..
- Per rientrare a X quando si è usciti senza chiudere la sessione
  - Shift - F7 per il display X n. :0
  - Shift - F8 per il display X n. :1
- “cambio” di display
  - Ctrl-Alt-F7 entra in display n. :0
  - Ctrl-Alt-F8 entra in display n. :1



# File system

---

- Linux: diversi file system “montabili”

**TUTTA** la struttura dei directory parte dal directory root (/).  
Da root partono tutti i dischi del computer, Linux, DOS o di rete, “montati” ad un qualsiasi directory

i nomi dei file e delle path sono CASE SENSITIVE, sono lunghi, senza estensioni "obbligatorie", non ci sono convenzioni sull'estensione

# **mount**

senza parametri mount mostra tutti i dispositivi attualmente montati, ed il relativo directory di “mount”



# Mounting

```
mount -t <tipo> <device> <directory>
```

- <tipo> (sono circa 100 !!)
  - ext2 Linux nativo ("normale")
  - swappartizione swap Linux
  - iso9660 cdrom
  - msdos DOS fat 16
  - vfat FAT 32 windows 95 osr2 e Win 98
  - ntfs NT filesystem
  - smbfs SMB filesystem (rete Windows)
  - nfs NFS: Network File System (rete Unix)



— <device>

- /dev/hdX (hard disk IDE-ATA)
- /dev/fdX (floppy disk)
- /dev/scX (hard disk SCSI)
- /dev/cdromX (cdrom ATA)

X = 1, 2 ..

— <directory>

- path di una directory che parte da root (/)
- è il "**mountpoint**"
- il directory **deve esistere** prima del mount



# Unmount

```
umount <directory>
```

- la stessa <directory> data nel mount

## File /etc/fstab

- file dal quale si leggono i filesystem da montare automaticamente all'accensione del computer
- se un device od un mounpoint sono in fstab si possono montare e smontare in modo semplificato:

```
mount <device> oppure
```

```
mount <mountpoint>
```



## /etc/fstab del mio computer:

```
/dev/hdb6 / ext2 defaults 1 1
/dev/hda1 /C vfat exec,dev,suid,rw,r 0 0
/dev/hda5 /E vfat exec,dev,suid,rw 0 0
/dev/hda6 /F vfat exec,dev,suid,rw 0 0
/dev/hda7 /G vfat exec,dev,suid,rw 0 0
/dev/hda8 /H vfat exec,dev,suid,rw 1 2
/dev/hda9 none swap defaults 0 0
/dev/hda10 /usr ext2 defaults 1 1
/dev/hdb5 /I vfat exec,dev,suid,rw 0 0
/dev/hdb1 /D msdos exec,dev,suid,rw,r 0 0
/dev/cdrom /mnt/cdrom iso9660
ro,user,noauto,unhide
/dev/fd0 /mnt/floppy ext2
user,owner,exec,dev,suid,rw,noauto 0 0
```



# Esecuzione di comandi

- I comandi del S.O. sono programmi

i programmi che partono senza indicare tutta la path sono presenti nella variabile path:

```
echo $path
```

- la directory locale ( ./ ) NON è nella path

se il file eseguibile “programma” è nella directory locale e **non** è in \$path la prima di queste **NON** lo fa partire, l'altra sì:

```
programma
```

```
./programma
```



# Programmi CUI

---

- **vi**

vi è un vecchio editor Unix che, per quanto datato, è utile quando non si usa Xwindow e si lavora solo in linea di comando.

– vi ha tre modi operativi:

- modo di comando
- modo di inserimento
- riga di comando



—vi: comandi di riga di comando:

- :q esce dal programma senza salvare (quit)
- :w <file> scrive il file (write file)
- :x salva ed esce dal programma (exit)
- :e <file> comincia a modificare un altro file (edit file)
- :u undo: elimina le modifiche fatte
- /<stringa> cerca la stringa indicata



# Midnight commander (mc)

---

è un clone per Linux di Norton Commander.  
Permette il browsing fra i directory, il viewing e  
l'editing dei file.



# Installazione di applicazioni

---

- shell script
- RPM (Red Hat e compatibili)
- yast (SuSE)
- programmi per RPM in ambiente grafico (gnorpm, krpm)
- compilazione (make)



# Ing. Gabriele MONTI



- Laureato in Ingegneria Elettronica all'Università di Bologna, nel 1986
- dal 1987 al 1993 ha lavorato presso l'Istituto GIORDANO di Bellaria come responsabile delle Sezioni Informatica e Metrologia
- dal Settembre 1988 al Maggio 1989 ha lavorato negli USA, presso la University of Florida, Gainesville
- dal 1993 è insegnante per la materia "Sistemi per l'elaborazione e la trasmissione dell'informazione", all'ITI di Cesena
- svolge inoltre attività di consulenza nei settori informatica e reti di computer, qualità, metrologia, formazione