Amministrazione del Sistema Operativo UNIX

Giuseppe Vitillaro

Consiglio Nazionale delle Ricerche

Centro di Studio per il Calcolo Intensivo in Scienze Molecolari

Dipartimento di Chimica

Università degli Studi di Perugia

e-mail: <peppe@unipg.it>

Perugia, 27/11/1999

Software di Pubblico Dominio

- Gli archivi anonymous FTP e i serventi World Wide Web mettono a disposizione un enorme patrimonio di software per le più diverse architetture e finalità.

- La maggior parte di questi pacchetti software vengono distribuiti utilizzando diversi tipi di politica e di licensing:

  - In alcuni casi non è necessario acquistarlo. È gratuito. L'autore mantiene la proprietà intellettuale e/o si riserva di accampare diritti solo in caso di uso "commerciale". Si parla di **freeware**.

  - In altri casi viene fornito in *prova* (con limitazioni più o meno importanti). Si parla di **shareware**.

Software di Pubblico Dominio

- Nel Public Domain sono reperibili pacchetti software di grande valore:
  - archiviatori zip, unzip, zoo, arj, gzip, gtar, ...
  - shells bash, zsh, pdksh, tcsh,...
  - editors emacs, axe, jove, vile, vim,...
  - linguaggi fortran, C, C++, basic, perl, tcltk,...
  - condivisione NFS, NIS (YP), samba
  - WWW softw. Apache httpd, Squid Cache, ...
  - Usenet News INND, NNTP, B-News, C-News, ...
  - FTP, gopher wu-ftpd, gopher, ncftp, ...
  - e-mail sendmail, poppers, ...
  - form. testi TeX, groff, ghostscript, ghostview,...
  - grafica xv, xpaint, lib. TIFF, JPEG, MPEG,...

Software di Pubblico Dominio

- Il distribution di X-Window, mantenuto all'MIT (ftp://export.lcs.mit.edu - ftp://ftp.x.org).

- Il software del progetto **GNU** (**GNU** is **N**ot **U**nix) della Free Software Foundation (ftp://prep.ai.mit.edu, ftp://ftp.gnu.org, http://www.gnu.org): gcc, gsed, gawk, gmake, ...

- Da alcuni anni sono disponibili anche distributions di Sistemi Operativi completi:
  - Minix
  - Plan9
  - FreeBSD
  - Linux.

Software di Pubblico Dominio

- È possibile costruire “ambienti” interessanti sia per gli utenti che per i developers.

  Nell’ambito della Ricerca Scientifica e dello Sviluppo Software hanno avuto grande successo: promuovono lo sviluppo tecnico e culturale.

- Il panorama del software disponibile è abbastanza vasto dal permettere un’ampia libertà di scelta e soddisfare ogni tipo di gusto personale.

- Il software Public Domain per l’ambiente UNIX è particolarmente adatto alla costruzione di serventi di rete personalizzati e poco costosi.

Software di Pubblico Dominio

Svantaggi di una soluzione Public Domain

- Non esiste un **vendor** a cui fare riferimento: il supporto è spesso affidato all'impegno personale degli autori.
- Si può *sperare* di ottenere supporto tecnico adeguato solo se si è connessi ad Internet (WWW, Usenet News, e-mail).
- In alcuni casi la documentatione è carente se non completamente assente.
- Richiede conoscenze tecniche nettamente al di sopra del livello *utente finale*.
- Può essere necessario conoscere almeno i *rudimenti* del C language, la struttura di un Makefile e i *trucchi* sistemistici basilari dell'ambiente UNIX.
- La qualità del software è molto variabile: la capacità di giudizio gioca un ruolo fondamentale.

Software di Pubblico Dominio

Vantaggi di una soluzione Public Domain

- Nella maggior parte dei casi il software è gratuito o comunque di bassissimo costo e le condizione di licensing sono molto permissive.
- È quasi sempre disponibile il ***sorgente***: ciò rende possibili adattamenti e personalizzazioni molto spinte.
- È molto facile da ottenere: è sufficiente essere connessi alla Rete Internet o acquistare una delle distribuzioni “stampate” su CDROM (un catalogo interessante alla Walnut Creek http://www.cdrom.com).
- Molti autori sono disponibili ad accettare richieste di fixes o addirittura di personalizzazione.
- Su alcni packages è possibile “costruire” soluzioni tecnologicamente avanzate.

Software di Pubblico Dominio

- Le stesse osservazioni sono valide per i sistemi Operativi UNIX (o Unix-Like) di pubblico dominio.

- La scelta è più delicata: l'investimento di risorse (macchina, tempo uomo, etc.) è più **oneroso**.

- La disponibilità del sorgente è un fattore importante: UNIX (ed in particolare la versione Berkeley) nasce con l'idea che i *sorgenti* del sistema siano sempre disponibili per la consultazione.

  È generalmente molto costoso (in alcuni casi addirittura impossibile) ottenere il sorgente di un sistema UNIX *marchiato*.

FreeBSD

- Tra il 1991 ed il 1992, William Jolitz pubblica una serie di articoli sul Dr. Dobb's Journal che riguardano il porting del BSD 4.3 su architettura Intel 386: "Porting UNIX to the 386".

- Nel Marzo 1992 il 386BSD 0.0 viene rilasciato nel dominio pubblico. È basato sulla distribuzione Net/2 U.C. Berkeley, del 4.3 BSD e contiene alcuni componenti della Free Software Foundation.

- Dal lavoro di William Jolitz nasce, nel 1993, la prima versione del FreeBSD: il progetto si propone la relalizzazione di un derivato della piattaforma Berkeley per architettura Intel 386/486.

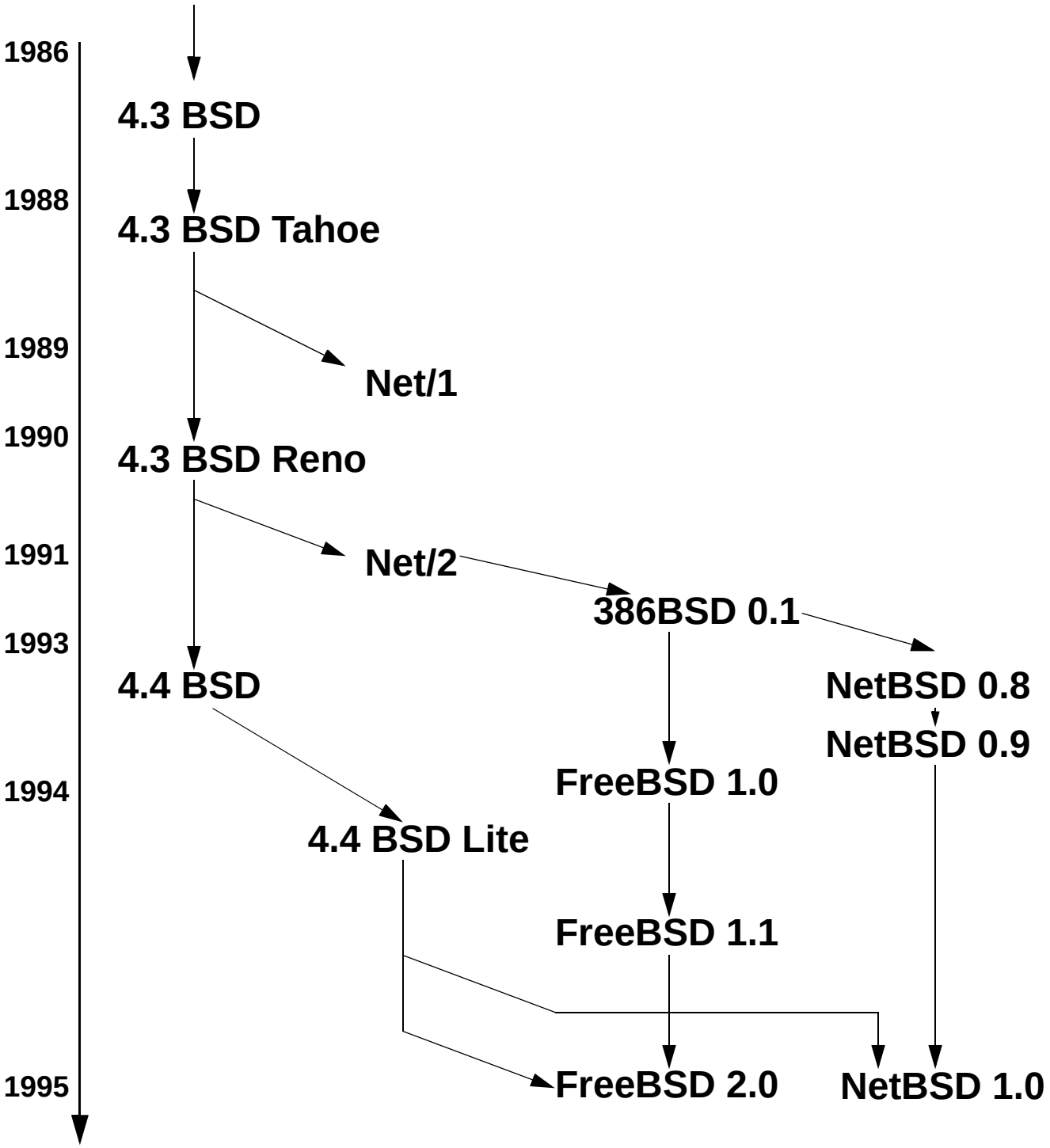
FreeBSD

- Il FreeBSD viene proposto come un sistema Unix like per l'utente finale.
- Un altro parto del 386BSD è il NetBSD: supporta un set più esteso di piattaforme hardware, ma è rivolto essenzialmente all'ambiente della ricerca.
- Il nome FreeBSD è stato coniato da David Greenman.
- Viene distribuito via Internet (http://www.freebsd.org) e su CD dalla Walnut Creek (http://www.cdrom.com).

FreeBSD

- La prima versione è stata distribuita su CD nel Dicembre 1993: FreeBSD 1.0.
- Nel 1994 la contesa legale tra Berkeley e AT&T/Novell viene risolta con il rilascio del 4.4 BSD Lite: il codice del 4.4 BSD Lite viene dichiarato libero dal codice ora di proprietà Novell.
- Da questa versione, nel Gennaio 1995, viene rilasciato il FreeBSD 2.0 e quindi il FreeBSD 2.1 (Novembre 1995).
- La versione 2.2 ha raggiunto il culmine della sua evoluzione con il rilascio 2.2.8 del Dicembre 1998. Il 1999 è iniziato con il rilascio della versione 3.1.

FreeBSD


1986
4.3 BSD
1988
4.3 BSD Tahoe
1989
Net/1
1990
4.3 BSD Reno
1991
Net/2
386BSD 0.1
1993
4.4 BSD
NetBSD 0.8
NetBSD 0.9
FreeBSD 1.0
1994
4.4 BSD Lite
FreeBSD 1.1
FreeBSD 2.0
NetBSD 1.0
1995

FreeBSD

- FreeBSD è un sistema operativo a 32 bit per CPU Intel 386, 486, Pentium:
  - Preemptive multitasking
  - Multiuser
  - TCP/IP networking
  - Memory Protection
  - X Window System (XFree86)
  - Binary compatible con SCO, BSDI, Linux
  - Demang Paged Virtual Memory
  - Shared Libraries
  - Centinaia di packages precompilati
  - Unica distribution centralizzata, source code per l’intero sistema.

FreeBSD

- Si propone per una vasta gamma di applicazioni:
  - la **robusta** implementazione TCP/IP ereditata dela BSD lo rende un candidato idela per i servizi Internet:
    - serventi FTP
    - serventi World Wide Web
    - serventi di e-mail
    - Usenet News
  - nel campo del Networking può svolgere egregiamente i ruoli di:
    - router (PPP user mode, dial on demand)
    - firewall
    - terminal server tty/slip/ppp
    - DNS name server.
- La disponibilità di X11R6 (sia PD che commerciale) ne permette l'uso come X Window workstation (anche diskless).

Linux

- Linux è stato sviluppato da **Linus Torvalds** (all'epoca studente della University of Helsinki, Finlandia) tra il 1991 ed il 1994.

  Nasce dall'idea di realizzare, da *scratch*, un sistema operativo per Personal Computers con architettura Intel IAPX86 (386/486) ed hardware IBM AT compatible.

- È un **kernel**, Unix-like, a 32 bit, compatibile con lo standard **POSIX**.

Linux

- Linux ha iniziato la sua vita, nell'aprile 1991, come un piccolo programma, sviluppato sotto il sistema operativo Minix, che eseguiva lo *switching* tra due task che stampavano, rispettivamente, AAAA e BBBB sulla console di un Personal Computer con architettura 386.

- L'idea originale era quella di estendere il sistema operativo Unix-like Minix (sviluppato dal Prof. Tanenbaum, un teorico dei sistemi operativi) dall'architettura 8086 all'architettura 80386.

- Sebbene l'idea originale prevedesse solo il microprocessore Intel 80386 ed hardware AT compatibile, oggi il kernel Linux è in grado di girare su tutti i microprocessori Intel (fino

Linux

all’ultimo nato Pentium-III) ed esistono versioni (a vari livelli di sviluppo) per:

- Motorola 68000: Amiga ed Ataris
- PowerPC: Apple PowerMac
- DEC AXP: DEC Alpha
- Sparc: SUN
- MIPS.

- Il kernel Linux si è evoluto nel tempo aggiungendo la maggior parte delle funzioni che ci si possono aspettare da un moderno sistema operativo:
  - Preemptive multitasking
  - Virtual Memory

Linux

- Shared Libraries
- Memory Management
- TCP/IP Networking
- Kernel Loader

- Il kernel Linux non contiene codice sviluppato dalla AT&T o comunque di proprietà di un vendor. Il codice sorgente è nel dominio pubblico sotto il licensing GPL (GNU Library General Public License, http://www.gnu.org).

- Parlando in senso *strettamente tecnico*, Linux **non è un sistema operativo**: è il kernel, il nucleo, di un sistema operativo. Controlla e gestisce l'hardware e le risorse della macchina.

Linux

Evoluzione del kernel Linux

anno	utenti	versione	linee di codice
1991	1	0.01	10.000
1992	1.000	0.96	40.000
1993	20.000	0.99	100.000
1994	100.000	1.0	170.000
1995	500.000	1.2	250.000
1996	1.500.000	2.0	1.500.000

- Il **Sistema Operativo** oggi noto con il nome **Linux** è in effetti costituito da:
  - il kernel Linux
  - compilatori e utilità GNU
  - software BSD
  - X Window System
  - software applicativo.

Linux

- Chiunque possieda le conoscenze necessarie può utilizzare il kernel Linux ed il software di pubblico dominio per realizzare un “Linux System”.

- È un compito complesso che richiede un notevole know-how e la disponibilità dell’hardware necessario: consiste nel ricompilare il kernel Linux ed il software applicativo ed impacchettarlo in un formato addatto alla sua installazione e distribuzione.

- Il risultato di questa elaborazione è noto come **Linux Distribution**. Nel corso del tempo organizzazioni senza fini di lucro (Debian) o vendors (RedHat) si sono assunti questo

Linux

compito ed hanno reso disponibili un gran numero di Linux Distributions:

- **Caldera Open Linux** (1.3)

  www.caldera.com

- **Debian GNU/Linux** (2.1)

  www.debian.org

- **Red Hat Linux** (5.2)

  www.redhat.com

- **Slackware Linux** (3.6)

  www.slackware.com

- **S.u.S.E. Linux** (6.0)

  www.suse.com

e la lista potrebbe continuare ...

Linux

- Ogni distribuzione possiede le sue peculari caratteristiche:
  - documentazione
  - installazione
  - configurazione
  - supporto
- Hanno in comune l'uso del kernel Linux, tuttora sviluppato e manutenuto da Linus Torvalds (http://www.kernel.org), e spesso l'uso massiccio del software GNU.

  In effetti sono sistemi operativi ***differenti*** !

  Vengono spesso installati, manutenuti e amministrati attraverso tools e logiche diverse.

Linux

Distributions Rates by www.32BitsOnline.com

Version	Doc	Features	Installa-tion	Configu-ration	Ease of Use	Support
Caldera 1.3	4	4	3	3	4	4
Debian 2.0	3	4	3	3	3	4
RedHat 5.2	4	4	5	5	4	4
Slackw. 3.6	3	3	3	2	2	4
SuSE 5.3	4	5	4	4	4	4

- La tabella mostra una comparazione tra cinque tra le più diffuse distributions Linux.

  La scelta della distribuzione più adatta alle proprie necessità può essere molto complessa e dipendere da fattori non strettamente tecnici.

  Un developer potrebbe installare Slackware sul suo home system ed utilizzare RedHat per scopi professionali.

  Molto dipende dagli obiettivi che si intendono perseguire, dalla sensibilità personale e da vincoli esterni come la disponibilità di software fornito da terze parti.

Linux

- Oggi Linux è un ***clone*** UNIX pressochè completo in grado di far girare X Windows, TCP/IP, NFS, NIS, Emacs, UUCP e di fornire servizi di news e posta elettronica.

- Alcune distribuzioni hanno attirato l'attenzione di vendors importanti come IBM, Microsoft, Digital, Oracle, SyBase.

Red Hat: installazione

- Ci occuperemo della procedura di installazione della distribuzione Linux Redhat 5.1.

  L'installazione del RedHat non è complessa. ma alcune semplici regole possono renderla più facile:

  - utilizzare hardware supportato e ben conosciuto; la lista dell'hardware supportato da RedHat può essere consultata a:

    http://www.redhat.com/hardware

  - installare per la prima volta il sistema su un hard disk vuoto o dopo un completo backup delle partizioni già esistenti

  - Leggere la guida di installazione (in una delle sue forme) **prima** di iniziare l'installazione.

  - in caso di problemi consultare il RedHat **Support Center** a **http://www.redhat.com/support**.

RedHat: installazione

- RedHat può essere installato da diversi supporti:
  - CDROM — SCSI o ATAPI
  - NETWORK — da un server FTP o NFS
  - DISCO — da una partizione MSDOS
  - SMB Shared Volume — cartella Windows condivisa
- Su un Personal Computers di classe Pentium o Pentium-II, un CDROM 24X ed un moderno disco EIDE o SCSI non sono necessari più di 30 minuti (salvo incidenti) per ottenere un sistema “up and running”.
- La distribuzione standard di Linux Redhat 5.1 consiste di un floppy e tre CD-ROM.

RedHat: installazione

- L’installazione da **CDROM** può iniziare in due modi:
  - bootstrap direttamente dal CD di installazione del RedHat Linux: è disponibile sui moderni PC che prevedono il boot da un CDROM (formato El Torito);
  - bootstrap da dischetto ed installazione da CD.
- Nel caso che il dischetto di bootstrap del RedHat non fosse disponibile è possibile *generarlo* su una macchina MSDOS, dal CD di installazione:

```
C:\> d:
D:\> cd \dosutils
D:\dosutils> rawrite
Enter disk image source file name: ..\images\boot.img
Enter target diskette drive: a:
Please insert a formatted diskette into drive A: and
press --ENTER-- : [Enter]
D:\dosutils>
```

RedHat: installazione

- Si esegue il **boot** del PC dal floppy A: (o dal CDROM).

  Il PC caricherà in memoria il kernel Linux che procederà ad una fase di “probing” dell’hardware presente sul PC ed infine eseguirà il programma di installazione.

```
                    Welcome to Red Hat Linux

 o  To install or upgrade a system running Red Hat Linux 2.0
    or later, press the <ENTER> key.

 o  To enable expert mode, type: expert <ENTER>.  Press <F3> for
    more information about expert mode.

 o  This disk can no longer be used as a rescue disk.  Press <F4> for
    information on the new rescue disks.

 o  Use the function keys listed below for more information.
...

[F1-Main] [F2-General] [F3-Expert] [F4-Rescue] [F5-Kickstart] [F6-Kernel]

boot:
```

  Le informazione emesse a video in questa fase possono essere importanti per determinare il livello di supporto dell’hardware.

RedHat: installazione

```
Console: 16 point font, 400 scans
Console: colour VGA+ 80x25, 1 virtual console (max 63)
pcibios_init : BIOS32 Service Directory structure at 0x000f7820
pcibios_init : BIOS32 Service Directory entry at 0xfd834
pcibios_init : PCI BIOS revision 2.10 entry at 0xfdb06
Probing PCI hardware.
Calibrating delay loop.. ok - 396.49 BogoMIPS
Memory: 254848k/260096k available (740k kernel code, 384k reserved,
3932k data)
Swansea University Computer Society NET3.035 for Linux 2.0
NET3: Unix domain sockets 0.13 for Linux NET3.035.
Swansea University Computer Society TCP/IP for NET3.034
IP Protocols: IGMP, ICMP, UDP, TCP
VFS: Diskquotas version dquot_5.6.0 initialized^M
Checking 386/387 coupling... Ok, fpu using exception 16 error
reporting.
Checking 'hlt' instruction... Ok.
Linux version 2.0.35 (root@bw01) (gcc version 2.7.2.3) #1 Thu Dec 17
17:30:50 CET
1998
Starting kswapd v 1.4.2
Serial driver version 4.13 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16550A
tty01 at 0x02f8 (irq = 3) is a 16550A
PS/2 auxiliary pointing device detected -- driver installed.
Real Time Clock Driver v1.09
Ramdisk driver initialized : 16 ramdisks of 4096K size
ide: i82371 PIIX (Triton) on PCI bus 0 function 145
    ide0: BM-DMA at 0xfcb0-0xfcb7
    ide1: BM-DMA at 0xfcb8-0xfcbf
hda: IBM-DTTA-350840, 8063MB w/467kB Cache, CHS=1024/255/63, UDMA
hdc: Pioneer CD-ROM ATAPI Model DR-A14S 0104, ATAPI CDROM drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
ide1 at 0x170-0x177,0x376 on irq 15
Floppy drive(s): fd0 is 1.44M
FDC 0 is a National Semiconductor PC87306
md driver 0.36.3 MAX_MD_DEV=4, MAX_REAL=8
scsi : 0 hosts.
scsi : detected total.
Partition check:
```

RedHat: installazione

```
 hda: hda1 hda2 < hda5 hda6 hda7 hda8 hda9 hda10 >
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem) readonly.
Trying to unmount old root ... okay
Adding Swap: 128484k swap-space (priority -1)
sysctl: ip forwarding off
Swansea University Computer Society IPX 0.34 for NET3.035
IPX Portions Copyright (c) 1995 Caldera, Inc
Appletalk 0.17 for Linux NET3.035
eepro100.c:v0.99B 4/7/98 Donald Becker
linux-eepro100@cesdis.gsfc.nasa.gov
eepro100.c:v0.99B 4/7/98 Donald Becker
linux-eepro100@cesdis.gsfc.nasa.gov
eth0: Intel EtherExpress Pro 10/100 at 0xfce0, 00:A0:C9:AC:18:5E, IRQ
5.
  Board assembly 677173-001, Physical connectors present: RJ45
  Primary interface chip i82555 PHY #1.
  General self-test: passed.
  Serial sub-system self-test: passed.
  Internal registers self-test: passed.
  ROM checksum self-test: passed (0x24c9f043).
  Receiver lock-up workaround activated.
```

- Il kernel Linux presente sul floppy (o sul CD) contiene il supporto hardware per una configurazione minimale che può essere estesa (caricando dei moduli) durante l'installazione (per esempio controllers e periferiche SCSI).

RedHat: installazione

- L’installazione di RedHat Linux richiede che una certa quantità di spazio disco sia disponibile sul vostro hard disk.

  L’esatto ammontare di questo spazio dipenderà dai packages della distribuzione che deciderete di installare e puo’ variare da un centinaio di megabyes ad un gigabytes.

  In ogni caso l’installazione richiederà la creazione di ***almeno*** una nuova partizione.

- MSDOS, Win95, WinNT, Linux, Freebsd: i sistemi operativi installabili su un PC, in genere, richiedono l’allocazione e l’attivazione di una o più partizioni affinchè sia possbile bootstrappare direttamente dall’hard disk.

RedHat: installazione

Master Boot Record
Partition Table
Partition 1 /dev/hda1
Partition 2 /dev/hda2
Partition 3 /dev/hda3
Partition 4 /dev/hda4

Il BIOS divide il disco in quattro partizioni definite nella Partition Table.

Linux numera queste partizioni da 1 a 4

hda, **hdb**, **hdc**, ... dischi **EIDE**

sda, sdb, **sdc**, ... dischi **SCSI**

- Il disco fisico viene suddiviso in partizioni identiche a quelle utilizzate da MSDOS o Windows (con un **Id** differente).

- Nella *terminologia* Linux (congruente con quella MSDOS) queste partizioni possono essere definite come **primarie** o come **estesa**.

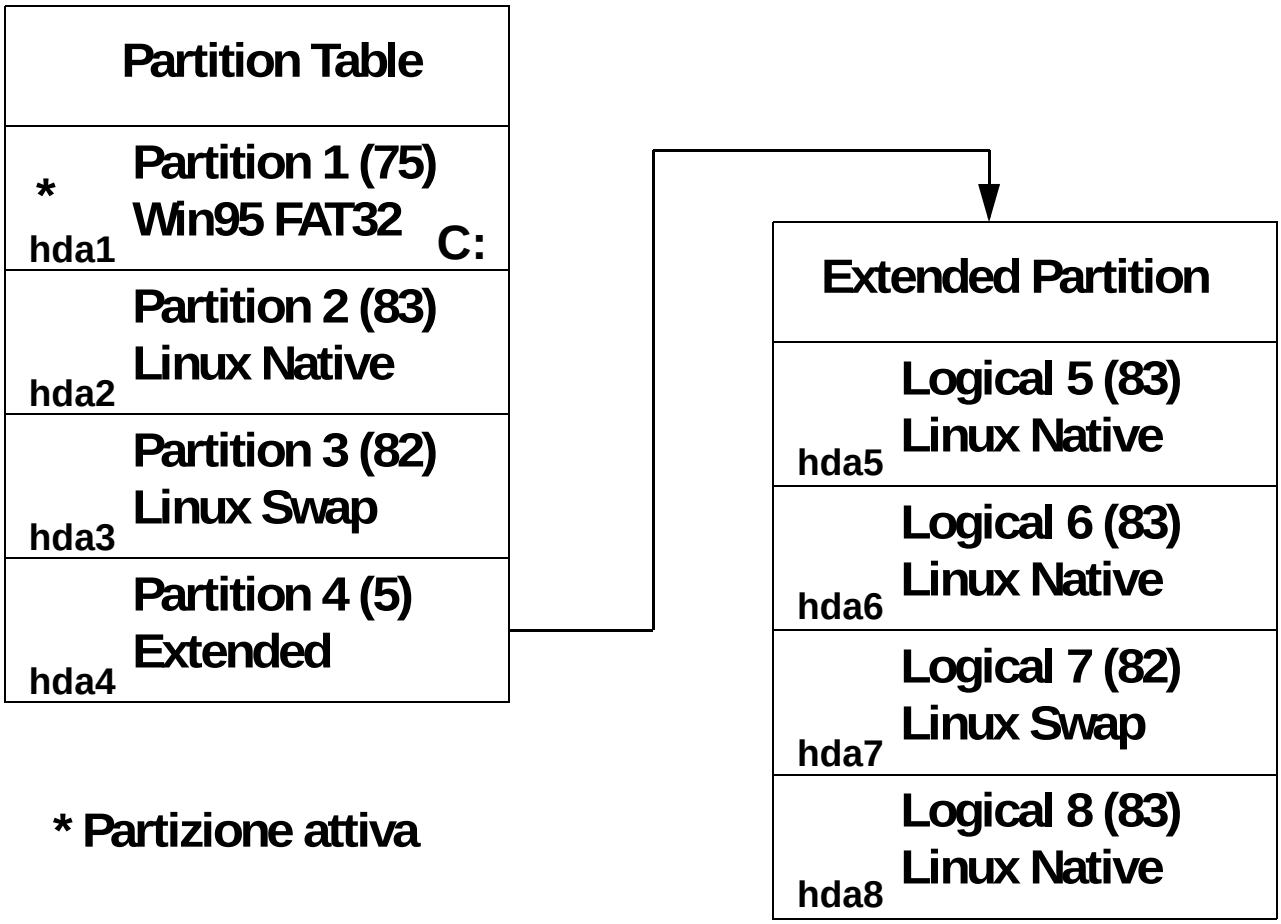
RedHat: installazione

- Le partizioni primarie (Linux può gestirne fino a quattro, il massimo consentito dal BIOS, numerate da **1** a **4**) consentono la costruzione di uno ed un solo file system (da cui il BIOS può eseguire il bootstrap).

- La partizione **estesa** (può esisterne al più una) può essere a sua volta suddivisa in **partizioni logiche** (dalle quali il BIOS non può bootstrappare) che vengono numerate partendo da **5** (fino a 64 per un disco EIDE e 16 per un disco SCSI).

- Ad ogni partizione corrisponde un device Linux sul quale è possibile costruire un file system (uno ed uno solo).

RedHat: installazione



- Nel Master Boot Record risiede il **Bootstrap Manager**: il suo compito è quello di caricare in memoria un sistema operativo da una delle partizioni. Viene caricato in memoria dal BIOS e può essere più o meno evoluto.

RedHat: installazione

- Creare le partizioni ed installare un boot manager può essere una operazione delicata, soprattutto se sono presenti altri sistemi (un backup preliminare è una scelta *prudente*).

- Attenzione: alcuni sistemi operativi (per esempio Windows 95) riscrivono il Master Boot Record ed installano il proprio codice di boot.

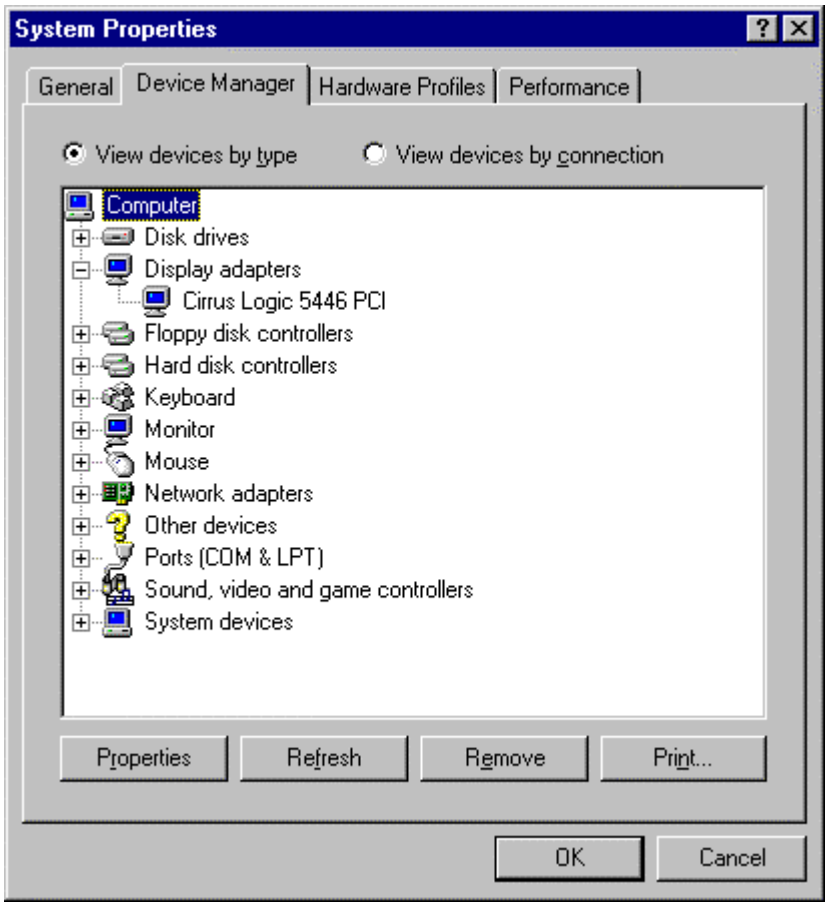
- Nel caso tutto lo spazio disco sia già stato allocato ad altri sistemi operativi esistono due alternative:

  - distruggere le partioni e ricrearle con un nuovo schema;

  - usare una utilità per deallocare lo spazio libero di una partizione esistente (vedi **fips**).

RedHat: installazione

- Se sul Personal è già installato Windows 95 un passo utile può essere quello di utilizzare il “**Device Manager**” (Gestione Periferiche) per ottenere ulteriori informazioni sulla configurazione hardware della macchina.

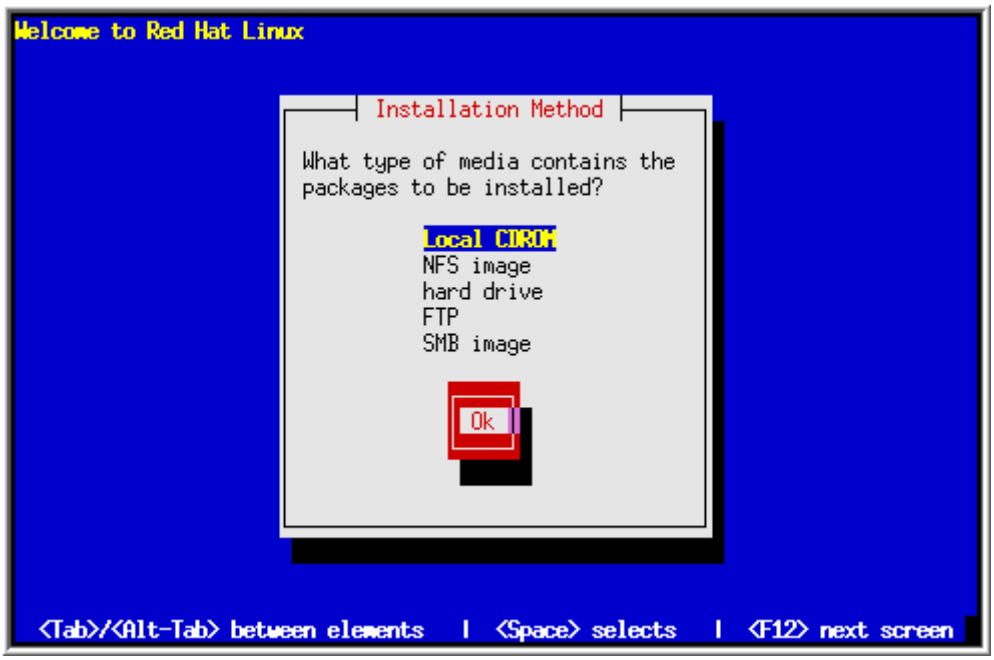


RedHat: installazione

- Al termine della fase di boot il programma di installazione (l'equivalente di un setup Windows) prende il controllo della macchina e propone la scelta della *lingua* utilizzata durante l'installazione,

il tipo di tastiera connessa allla macchina, ed il metodo di installazione da utilizzare

RedHat: installazione



RedHat: installazione

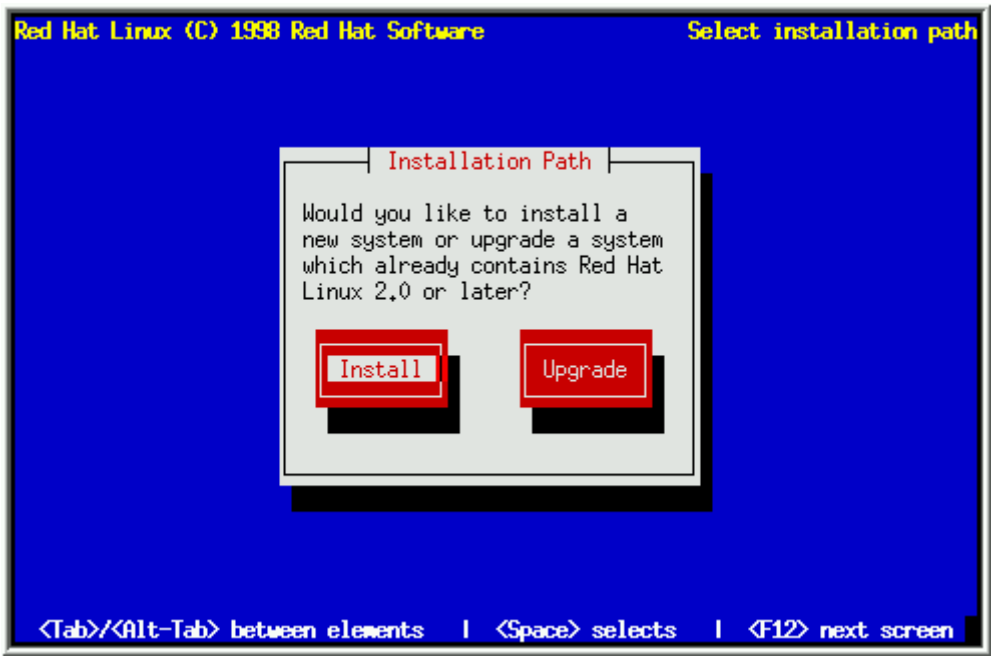
- Esamineremo il metodo di installazione da CDROM: è sicuramente uno dei più comuni, ma, in ambiente di rete, potrebbe essere piuttosto frequente la necessità di installare il sistema da un CDROM o da un file system condiviso sulla rete locale.

- Dopo aver selezionato l'installazione da CDROM il sistema eseguirà una fase di *probing* cercando di determinare che tipo di CDROM è presente sulla macchina.

  Le periferiche IDE/ATAPI vengono individuate automanticamente, ma è possibile utilizzare anche devices di altro tipo (SCSI).

RedHat: installazione

- A questo punto il programma di installazione richiederà se si intende installare un nuovo sistema da scratch o eseguire un aggiornamento (un *upgrade*) di una versione già presente sull'hard disk:



a cui si risponderà **Install** nel caso di una **nuova** installazione.

RedHat: installazione

- Il programma di installazione procederà quindi a collezionare dall'utente le informazioni necessarie a *partizionare* l'hard disk e a definire i file systems necessari al sistema operativo.

- Definire appropriatamente i file system di un sistema UNIX è un'operazione delicata che richiede esperienza e in alcuni casi comporta anche scelte *soggettive,* a volte difficili da giustificare in modo razionale.

  Cercheremo di individuare delle linee guida, ma occorre tener presente, che fermi restando alcuni concetti fondamentali, è praticamente impossibile definire in modo completo il problema.

RedHat: installazione

- Una prima domanda:

  "**quanti e quali file systems**"

  - Ricordando che ogni sistema UNIX deve avere almento un file system da montare come **radice** è evidente che occorrerà definire almeno il file system **root**;

  - Per permettere a Linux di paginare la memoria virtuale è necessario definire ***almeno*** una partizione (attenzione su di essa non verrà creato un file system) che il sistema utilizzerà come **raw device** per eseguire lo **swap** delle pagine di memoria che non è possibile mantenere in **RAM**.

    Una **swap partition** può avere dimensioni fino a 127Mb. La dimensione minima suggerita è almeno la dimensione della RAM o 16Mb se la macchina monta meno di 16Mb. È praticamente inutile superare di due/tre volte la dimensione della RAM.

RedHat: installazione

- Questi sono i requisiti ***minimi*** necessari all'installazione di praticamente ogni sistema operativo Unix o Unix-like.

  L'esperienza suggerisce che nella pratica sono necessari un numero maggiore di file system: solo in caso di sistemi di test o di condizioni estreme ci si riduce alla sola radice.

- Un solo file system espone in maniera **seria** a situazioni nelle quali un danno alle strutture dati della radice comporta la perdita **dell'intero sistema**.

  Inoltre la suddivisione in più file system permette una gestione più razionale dello spazio disco e una suddivisione logica delle sue componenti.

RedHat: installazione

- Un possibile schema di suddivisione potrebbe consistere dei seguenti file systems:
  - **/ (root) partition**
    file system radice con dimensioni variabili tra 50 e 100Mb
  - **/usr partition**
    nel file system /usr viene installato la maggior parte del software del sistema RedHat. Può avere dimensioni variabile tra 300 e 700Mb
  - **/var partition**
    la maggior parte dei **log** di sistem (in /var/log) e delle aree di **spool** (per la posta, la stampa in /var/spool) si trovano in questo file system
  - **/tmp partition**
    conterrà i file *temporanei* di lavoro degli utenti. Le sua esistenza e dimensione dipendono dagli scopi del sistema che si sta installando.

RedHat: installazione

- Altri possibili file systems:
  - **/boot partition**
    tutto ciò che è necessario al kernel Linux per eseguire il bootstrap si trova in questa directory (dimensione tra i 5 e i 10Mb).
    Definire un partizione di boot può tornare utile nel caso di dischi con più di 1024 cilindri o di spazio disco molto limitato sul primo disco EIDE/SCSI.
  - **/home partition**
    il file system che conterrà le home directories degli utenti.
  - **/usr/local partition**
    software utente o di pubblico dominio comunque non parte integrate del sistema RedHat
  - **/usr/src partition**
    sorgenti del kernel Linux (circa 30Mb)
    sorgenti del sistema RedHat

RedHat: installazione

- Le informazioni richieste per configurare i file systems riguardano le partizioni da definire ed i *mount point* che l'utente intende utilizzare.

- La distribuzione RedHat fornisce due strumenti distinti che permettono di compiere questa operazione:

  - **Disk Druid**

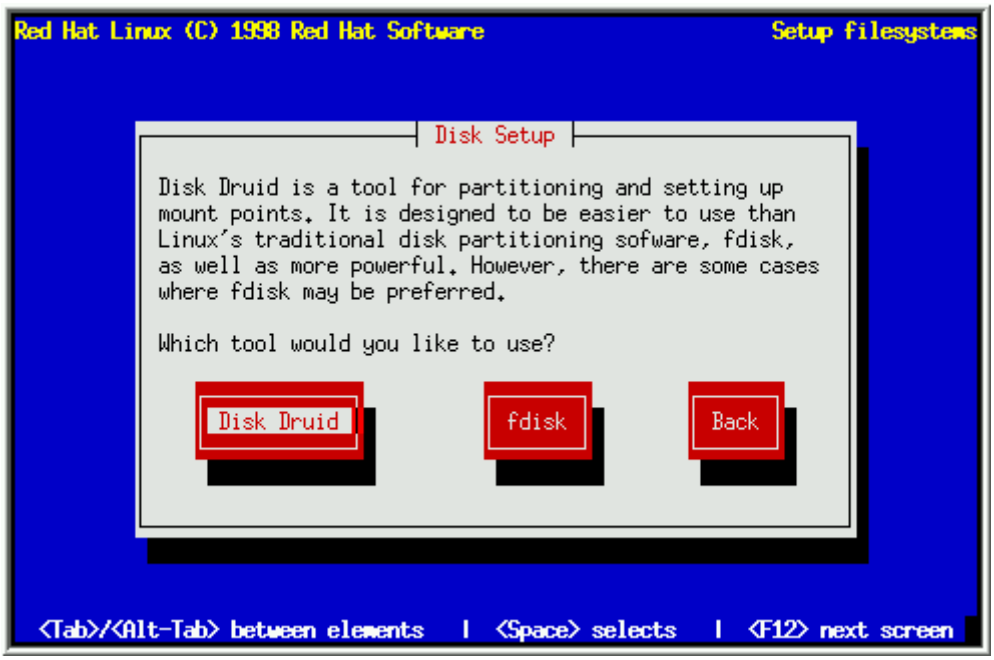
    è un tools particolare della distribuzione RedHat con una interfaccia *accattivante;*

  - **fdisk**

    è lo strumento tradizionale Linux (funzionalmente simile all'fdisk MSDOS) con una interfaccia a linea di comando.

RedHat: installazione

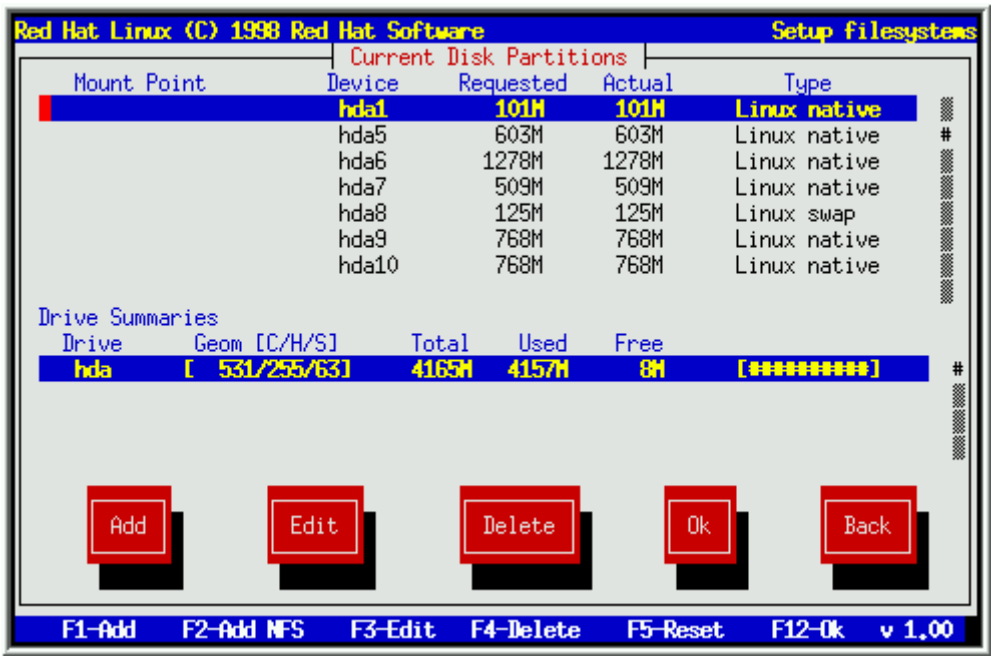
- La scelta fra uno dei due strumenti viene proposta dal programma di installazione



- Il consiglio migliore è quello di scegliere lo strumento sul quale si possiede più controllo. fdisk è più flessibile del Disk Druid.

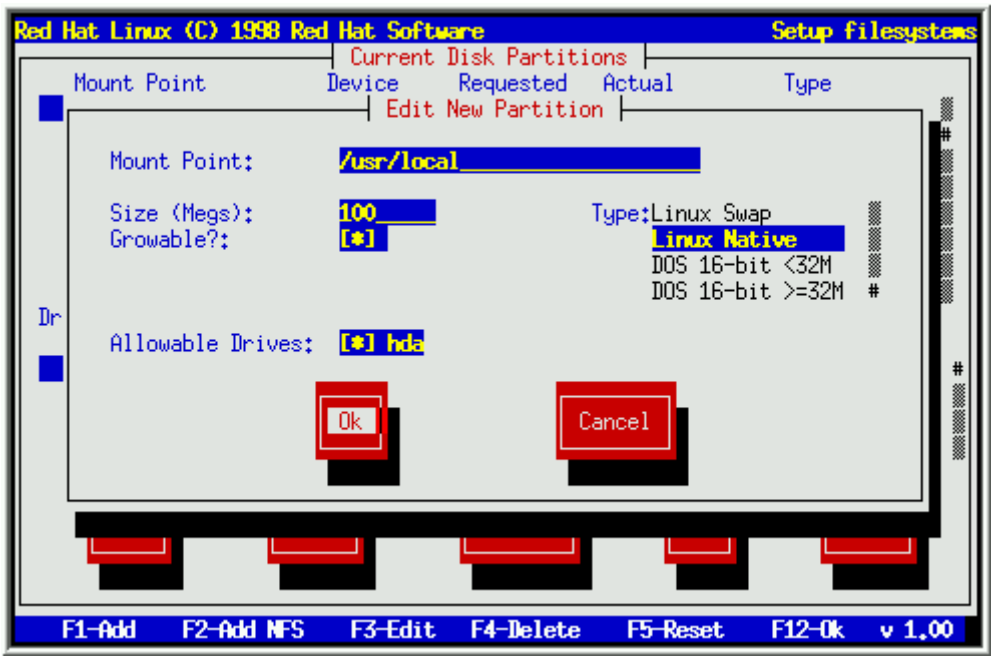
RedHat: installazione

- Le informazioni presentate e richieste dal Disk Druid o da fdisk sono praticamente le stesse:
  - l'elenco delle partizioni presenti sul disco con le informazioni relative al nome del **device** Linux, al **tipo** ed alle **dimensioni**;
  - il ***mount point*** dove la partizione verrà montata alla partenza del sistema.



RedHat: installazione

- Richiedendo al Druid di aggiungere una partizione (su uno dei dischi presentati nella sezione Drive Summaries) verrà presentato un pannello di questo tipo



che collezionerà le informazioni relative al mount point, al tipo ed alla dimensione della partizione che state creando.

RedHat: installazione

- Altre funzioni rese disponibili dal Disk Druid sono:
  - **Edit**

    permette di modificare le caratteristiche di una partizione che è già stata definita
  - **Delete**

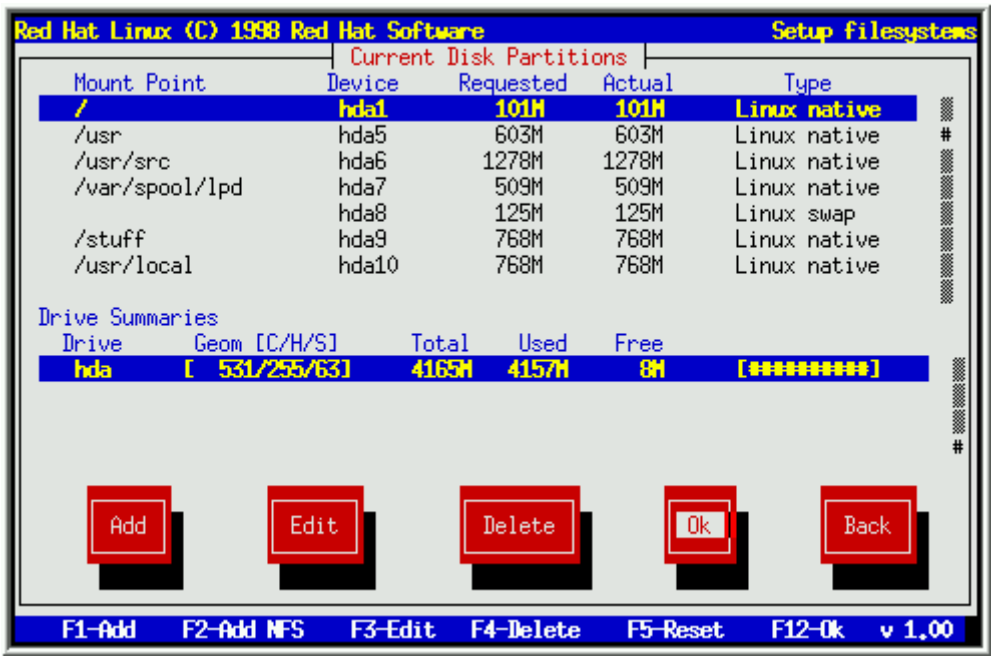
    permette di eliminare una partizione esistente
  - **Ok**

    conferma che le informazioni inserite sono quelle desiderate e richiede al Druid di riscrivere la nuova *partition table* sull'hard disk

Altre funzionalità sono disponibili mediante l'uso di vari tasti funzione in genere auto esplicativi.

RedHat: installazione

- Il compito di definizione delle partizioni può considerarsi terminato nel momento in cui avrete definito tutti le partizioni necessarie al funzionamento del vostro nuovo sistema



confermando con Ok renderete permanenti le modifiche (attenzione!) e potrete proseguire con la successiva fase dell'installazione.

RedHat: installazione

- Occorre tener presente che il Disk Druid è disponibile solo nella distribuzione RedHat e solo per la fase di installazione. La RedHat non lo rende *ufficialmente* disponibile come tool standalone.

- Imparare ad usare il più *tradizionale* **fdisk** fornisce comunque uno strumento più veloce, flessibile e sempre disponibile. Inoltre fdisk può essere utilizzato anche per definire nuove partizioni e file systems una volta che il sistema operativo sarà installato e funzionante.

- Nel caso scegliate di utilizzare fdisk durante l'installazione, il programma di installazione presenterà un pannello che richiederà quale disco desiderate partizionare

RedHat: installazione

e quindi procederà a lanciare per voi il comando fdisk con il quale potrete eseguire il partizionamento del disco.

fdisk svolge solo ed unicamente questo compito. Sarà quindi necessaria una successiva fase per configurare i mount point delle partizioni che sono state definite.

RedHat: installazione

- L’interfaccia utente del comando fdisk è molto semplice (qualche maligno potrebbe sostenere che è *rudimentale*). Occorre ricordare che spesso la facilità d’uso di una interfaccia utente è direttamente proporzionale alla complessità ed alle dimensioni del codice che la implementano.

- Nel caso di un comando ***fondamentale*** come questo è molto più importante che possa funzionare in condizioni **estreme**: potrebbe essere necessario definire o modificare la partition table di una macchina che si trova all’altro capo del mondo ed ha eseguito il boot da un dischetto, mentre voi state utlizzando un handheld computer ed un cellulare per connettervi.

RedHat: installazione

- Il comando fdisk accetta in genere comandi molto semplici composti da un solo carattere:

```
# fdisk
Using /dev/hda as default device!

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 1024 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1         4     32098+   4  DOS 16-bit <32M
/dev/hda2             5      1024   8193150    5  Extended
/dev/hda5             5        11     56196   83  Linux native
/dev/hda6            12       142   1052226   83  Linux native
/dev/hda7           143       158    128488+  82  Linux swap
/dev/hda8           159       167     72261   83  Linux native
/dev/hda9           168       176     72261   83  Linux native
/dev/hda10          177       307   1052226   83  Linux native
/dev/hda11          308       830   4200966   83  Linux native

Command (m for help):
```

il comando per l'help è ***m*** e fornisce l'elenco dei comandi principali di fdisk.

RedHat: installazione

```
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
   o   create a new empty DOS partition table
   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)
```

- Per creare una nuova partizione è possibile utilizzare il comando ***n***.

```
Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
p
Partition number (1-4): 3
No free sectors available

Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (831-1024): 831
Last cylinder or +size or +sizeM or +sizeK ([831]-1024): +20M
```

RedHat: installazione

E questo sarà il risultato che potrete visualizzare con il comando ***p***:

```
Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 1024 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1         4     32098+   4  DOS 16-bit <32M
/dev/hda2             5      1024   8193150    5  Extended
/dev/hda5             5        11     56196   83  Linux native
/dev/hda6            12       142   1052226   83  Linux native
/dev/hda7           143       158    128488+  82  Linux swap
/dev/hda8           159       167     72261   83  Linux native
/dev/hda9           168       176     72261   83  Linux native
/dev/hda10          177       307   1052226   83  Linux native
/dev/hda11          308       830   4200966   83  Linux native
/dev/hda12          831       833     24066   83  Linux native
```

L'asterisco "*****" indica la partizione attiva (quella da cui il BIOS esegue il boot se non è attivo un boot manager) che può essere modificata con il comando ***a***.

- È sicuramente una buona idea annotare nomi, dimensioni e mount point delle partizioni.

RedHat: installazione

- Un altro comando che vi sarà sicuramente utile e ***t***. Permette di modificare il **tipo** della partizione. L'elenco dei tipi noti ad fdisk può essere ottenuto in ogni momento con il comando ***l***.

```
Command (m for help): l

 0  Empty            a  OS/2 Boot Manag 65  Novell Netware  a6  OpenBSD
 1  DOS 12-bit FAT   b  Win95 FAT32     75  PC/IX           a7  NEXTSTEP
 2  XENIX root       c  Win95 FAT32 (LB 80  Old MINIX       b7  BSDI fs
 3  XENIX usr        e  Win95 FAT16 (LB 81  Linux/MINIX     b8  BSDI swap
 4  DOS 16-bit <32M  f  Win95 Extended  82  Linux swap      c7  Syrinx
 5  Extended        40  Venix 80286     83  Linux native    db  CP/M
 6  DOS 16-bit >=32 51  Novell?         85  Linux extended  e1  DOS access
 7  OS/2 HPFS       52  Microport       93  Amoeba          e3  DOS R/O
 8  AIX             63  GNU HURD        94  Amoeba BBT      f2  DOS secondary
 9  AIX bootable    64  Novell Netware  a5  BSD/386         ff  BBT
```

Potrete notare che fdisk può essere utlizzato anche per partizionare il disco per sistemi operativi diversi da Linux. Esistono anche utilità MSDOS (come **pfdisk**) che permettono di eseguire lo stesso compito da un semplice dischetto di boot MSDOS.

RedHat: installazione

- Il comando ***t*** dovrà sicuramente essere utilizzato per modificare il tipo di almeno una partizione: la partizione di swap. Infatti il comando ***n*** crea partizioni “**Linux native**” (**ID 83**) mentre la partizione di swap è di tipo “**Linux swap**” (**ID 82**)

```
Command (m for help): t
Partition number (1-11): 11
Hex code (type L to list codes): 82
Changed system type of partition 11 to 82 (Linux swap)

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 1024 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1         4     32098+   4  DOS 16-bit <32M
/dev/hda2             5      1024   8193150    5  Extended
/dev/hda5             5        11     56196   83  Linux native
/dev/hda6            12       142   1052226   83  Linux native
/dev/hda7           143       158    128488+  82  Linux swap
/dev/hda8           159       167     72261   83  Linux native
/dev/hda9           168       176     72261   83  Linux native
/dev/hda10          177       307   1052226   83  Linux native
/dev/hda11          308       830   4200966   82  Linux swap
```

RedHat: installazione

- Potete uscire in ogni momento con il comando ***q*** (attenzione fdisk non chiede conferma anche se avete cambiato la tabella delle partizioni) senza modificare in alcun modo la tabella delle partizioni sull'hard disk.

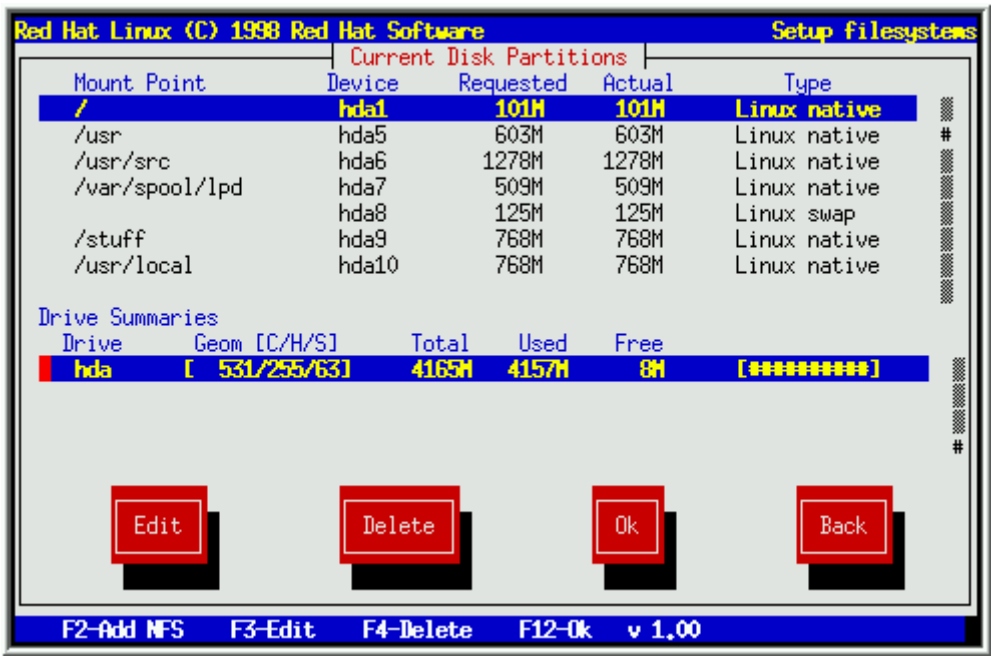
- Il comando ***w*** scrive il nuovo schema nella partition table dell'hard disk.

  Attenzione: a questo punto avrete reso permanenti le modifiche e riscritto la partition table del vostro hard disk.

  Utilizzate il comando ***w*** solo quando siete *assolutamente* sicuri delle modifiche che avete apportato. Un errore in questa fase può danneggiare ***irreparabilmente*** le partizioni già esistenti.

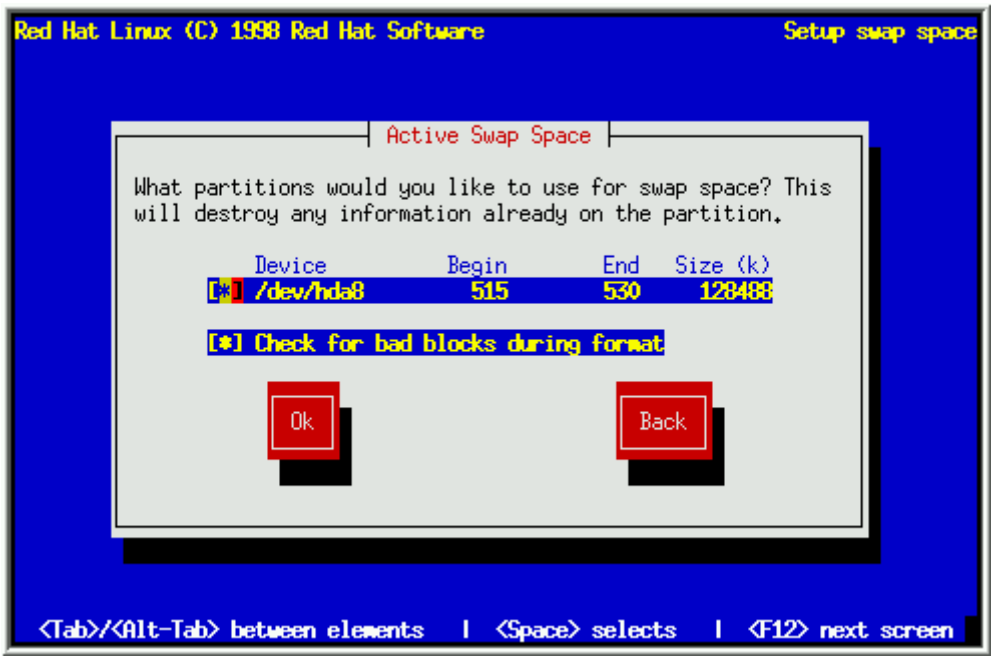
RedHat: installazione

- Una volta salvata la nuova tabella delle partizioni il programma di installazione RedHat procederà a configurare i file systems presentando comunque nuovamente il pannello del Disk Druid



RedHat: installazione

- A meno che non vi siate dimenticati di definire almeno una partizione di **swap** (nel qual caso è sicuramente meglio tornare sui propri passi) il programma di installazione inizializzerà la partiziono di swap



eseguendo eventualmente un controllo per identificare settori difettosi dell'hard disk.

RedHat: installazione

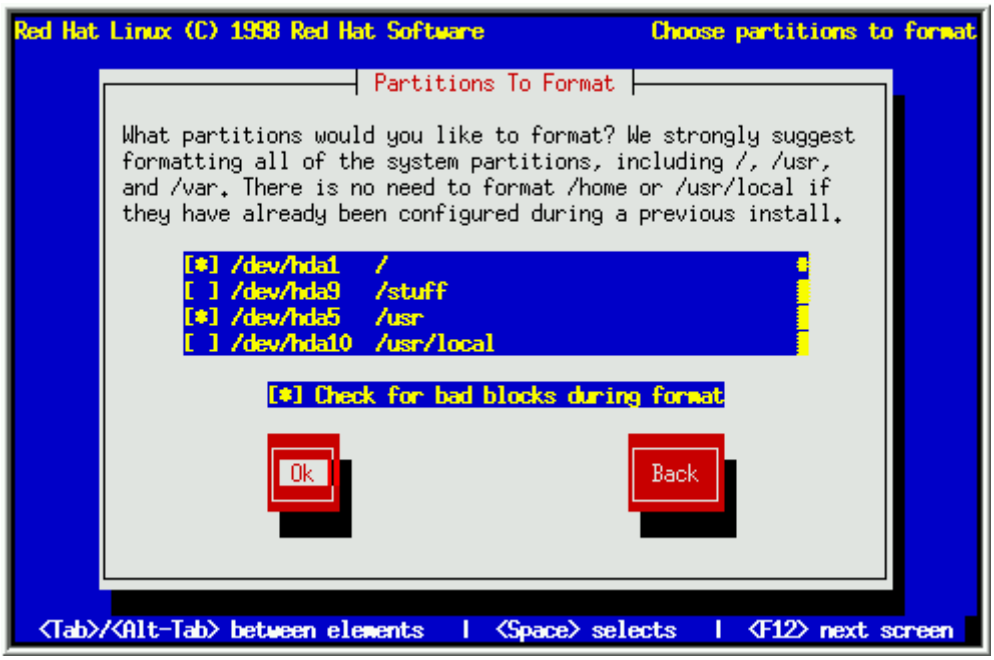
- La verifica dei blocchi delle aree di paginazione è un passo **importante**. Un sistema UNIX che non sia in grado di rileggere una pagina da un'area di swap in genere va in ***panic*** senza fornire ulteriori messaggi di errore.

- A meno che non abbiate la certezza, ottenuta per altre vie, che il disco che state utizzando abbia una superficie perfettamente integra, questo passo è fortemente consigliato.

- Lo stesso tipo di verifica, sebbene sia comunque utile, è sicuramente meno importante per i file systems.

RedHat: installazione

- Il programma di installazione è ora in possesso di tutte le informazioni necessarie alla creazione dei file systems che il nuovo sistema utlizzerà.



Le partizioni che sceglierete di “riformattare” saranno *reinizializzate*. Le altre semplicemente *montate* così come sono.

RedHat: installazione

- Durante tutta l'installazione, oltre ai pannelli di dialogo full screen del programma di setup, sulla console principale, sono disponibili altre quattro consoles *virtuali*:
  - Console 1 - ***programma di installazione***

    Alt-F1
  - Console 2 - ***shell prompt***

    Alt-F2
  - Console 3 - ***log del programma di installazione***

    Alt-F3
  - Console 4 - ***log del kerne***l

    Alt-F4
  - Console 5 - ***altri messaggi***

    Alt-F5

RedHat: installazione

- In ogni momento è possibile *switchare* da una console all'altra premendo il corrispondente tasto.

- Si rivela particolarmente utile nel caso si presenti qualche problema imprevisto: hardware non supportato, errori sull'hard disk o sul CDROM, etc.

- Utilie può rivelarsi anche la disponibilità della shell sulla console 2 e dei messaggi del log di installazione sulla console numero 3.

- È possibile seguire ogni passo del programma di installazione sulla console 3.

RedHat: installazione

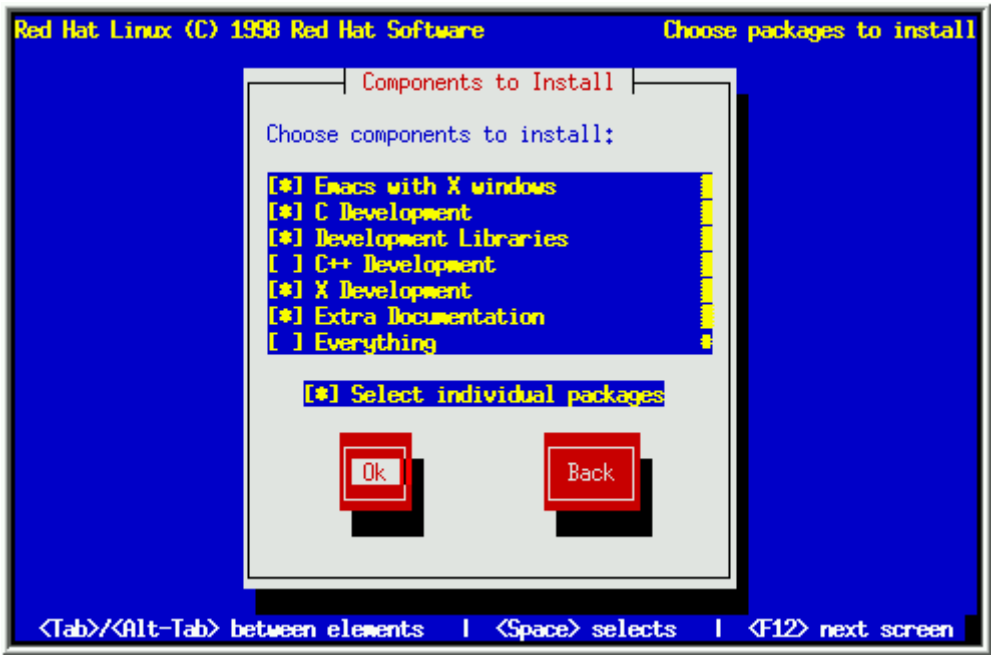
- La distribuzione RedHat è composta da un gran numero di *packages* la cui installazione può essere decisa in qualunque momento.

- Il programma di installazione raggruppa i packages in *componenti* per permettere una installazione *selettiva* che non richieda all'utente una conoscenza approfondita dei singoli pacchetti.

- La guida di installazione del RedHat contiene, in appendice, una descrizione, necessariamente sommaria, dei packages disponibili.

  I singoli pacchetti possono, in alcuni casi, essere applicazioni molto complesse.

RedHat: installazione



- Il sistema di gestione dei packages, il **R**ed **H**at **P**ackage **M**anager (**RPM**), è un sistema *sofisticato* ed *aperto* per la gestione e la installazione di pacchetti software in grado di considerare anche le *dipendenze* tra diversi packages.

RedHat: installazione

- Il problema di mantenere un record del software installato, delle dipendenze e dell'aggiornamento di un certo numero di componenti di un sistema operativo è un problema **complesso**.

- La distribuzione Linux RedHat contiente una utilità che fornisce queste funzionalità: **rpm**.

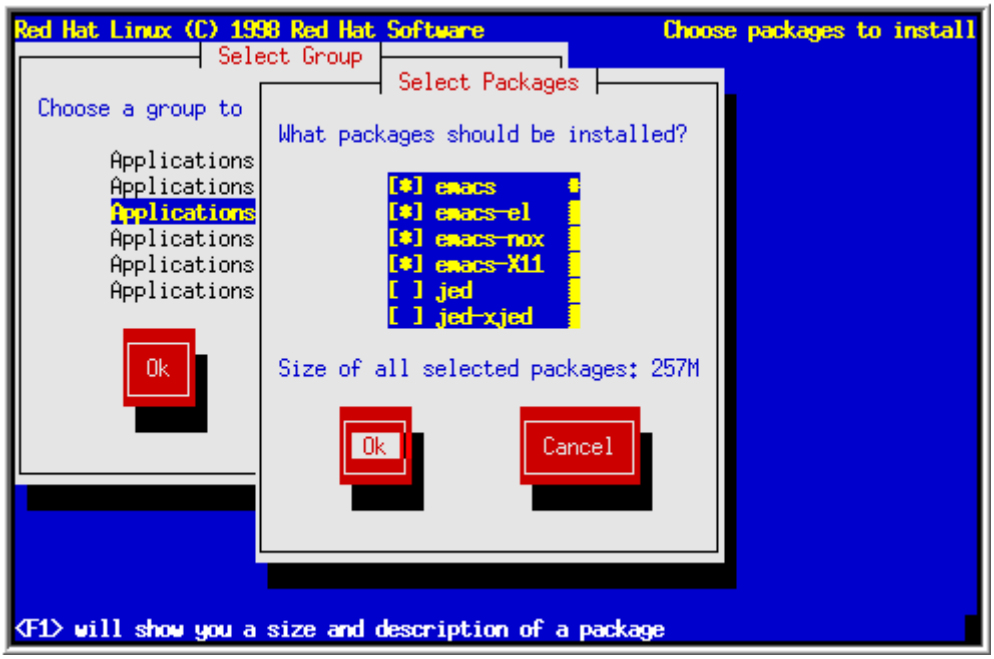
  È fortemente consigliato a chiunque debba installare e/o amministrare un RedHat leggere con attenzione le pagine del manuale di questo tool.

- Il programma di installazione RedHat utilizza il comando rpm dopo aver collezionato le informazioni necessarie.

RedHat: installazione

- RedHat fornisce anche una interfaccia grafica (X Window) per la installazione e gestione dei packages nota come **glint** (Graphical Linux INstallation Tool).

- È possibile scegliere i singoli packages di cui eseguire l'installazione selezionando l'opzione "**Select individual packages**":



RedHat: installazione

- Il tasto funzionale **F1** permette di ottenere una breve descrizione del singolo package in installazione.

  Ciò non può esimire dal conoscere il software che si sta installando. Una buona linea guida può essere quella di installare un package solo quando sono soddisfatti alcuni criteri:

  - il pacchetto è **indispensabile** per il funzionamento del sistema;
  - esiste una esplicita richiesta da parte degli utenti del sistema;
  - si è a conoscenza delle funzionalità del pacchetto software e lo si ritiene importante per l'amministrazione o per l'uso del sistema;
  - il pacchetto è un **prerequisito** per l'installazione di uno o più packages.

RedHat: installazione

- Nel caso il programma di installazione individui una serie di packages che sono *prerequisiti* all'installazione dei pacchetti selezionati potrà presentare un pannello di dialogo:

che potrà essere confermato per dare il via all'installazione del sistema.

RedHat: installazione

- La gestione dell'**orologio** (clock) di una macchina UNIX è differente da quello di una macchina MSDOS/Windows.

- In genere l'orologio **hardware** (CMOS) di una macchina UNIX viene mantenuto sul meridiano di Greenwich (un'ora indietro con l'ora solare e due ore indietro con l'ora legale).

  Si usa dire che il la macchina lavora sul GMT (Greenwich Mean Time) o sull'UTC (Coordinated Universal Time).

- L'orologio **software** del sistema (quello utilizzato dagli utenti) puo' essere adattato al proprio meridiano specificando le caratteristiche del proprio *time zone.*

RedHat: installazione

- Un modo semplice di fare qualche esperimento può consistere nel definire la variabile di environment **TZ**

```
$ date
Wed Apr 21 13:09:29 CEST 1999
$
$ export TZ=MET-1MDT,M3.5.0,M10.5.0
$
$ date
Wed Apr 21 13:09:37 MDT 1999
$
$ export TZ=ORS-2ORL,M3.5.0,M10.5.0
$
$ date
Wed Apr 21 14:09:45 ORL 1999
$ export TZ=
$
$ date
Wed Apr 21 11:10:30 UTC 1999
$
```

ed in effetti il sistema adatterà la data addirittura sulle esigenze del *singolo* utente (che potrebbe star lavorando dall'Australia su un sistema italiano).

RedHat: installazione

- In effetti Linux RedHat utlizza un metodo più complesso per definire il time zone di *default* del sistema. Il *link simbolico* **/etc/localtime** punta ad un file (binario) che contiene tutte le caratteristiche del meridiano locale:
  - distanza dal meridiano di Greenwich (negativa verso Est e positiva verso Ovest, così come la vedono gli Americani);
  - inizio e fine dell'ora legale in tutta la storia del meridiano locale, nel range di date supportate da un sistema UNIX a 32 bit:
    - inizio: Fri Dec 13 1901
    - zero: Thu Jan 1 1970 (**the Epoch**)
    - fine: Tue Jan 19 2038
- Le macchine UNIX mantengono l'orologio come distanza in secondi dall'**Epoch** basata sul GMT.

RedHat: installazione

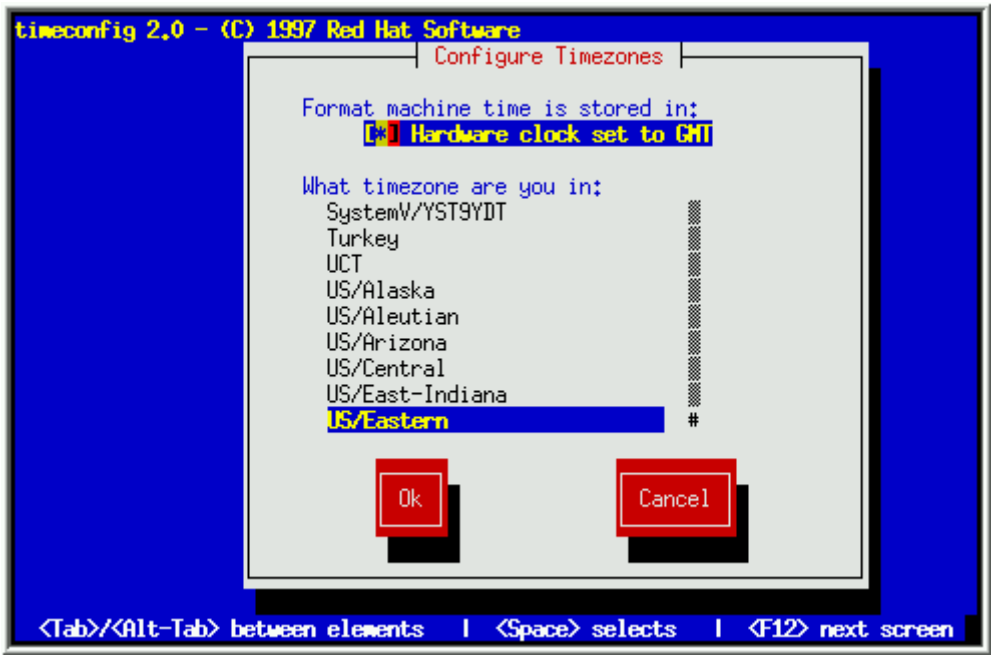
- Questa tecnica è utilizzabile (e fortemente consigliata) solo su una macchina che giri prevalentemente sotto il sistema operativo Linux.

- Se la macchina verrà utlizzata con sistemi che usano l'hardware clock per ottenere direttamente la data del sistema (leggi MSDOS o Windows) è preferibile lasciare la data CMOS sull'ora locale, informando il sistema Linux di questo fatto.

  Occorre tener presente che questo tipo di configurazione può creare problemi nel passaggio automatico da ora solare ad ora legale.

RedHat: installazione

- Il programma di installazione del RedHat Linux richiede all'utente questo tipo di informazione presentando il seguente pannello:



la cui configurazione può essere modificata, una volta che il sistema sarà installato e funzionante, con l'utilità **/usr/sbin/timeconfig**.

RedHat: installazione

- Le ultime informazioni richieste dal programma di installazione riguardano le modalità di bootstrap della macchina.

- Il kernel Linux viene caricato in memoria da una utlità nota come **LILO**: the **L**inux **LO**ader.

  LILO è un Bootstrap Manager molto flessibile che gestisce il caricamento in memoria del kernel.

  Esistono alternative che permettono di eseguire il boot da un dischetto (indispensabili in situazioni di emergenza) o da una partizione MSDOS (LOADLIN, SYSLINUX).

  Installare LILO è sicuramente il modo più flessibile e sicuro per gestire il boot di Linux dall'hard disk.

RedHat: installazione

- LILO può essere installato:
  - nel Master Boot Record

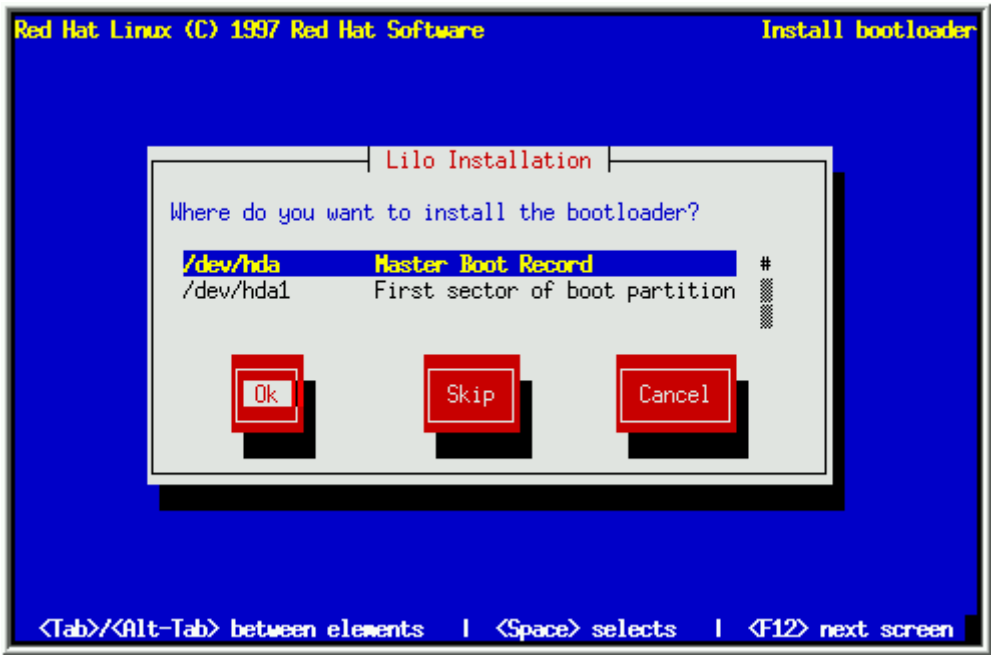
    è la configurazione raccomandata a meno che non si abbia l'intenzione di usare un differente boot manager (OSBS, OS/2 Boot Manager, etc.);

    in questo caso LILO può fungere da boot manager anche per altri sistemi operativi: MSDOS, DR-DOS, OS/2, Windows 95/98, Windows NT, FreeBSD, SCO UNIX, etc.

  - il primo settore della partizione di bootstrap

    in questo caso LILO si entrerà in azione solo nel momento in cui si sceglierà (mediante l'uso di un altro boot manager installato nell'MBR) di far girare Linux sulla macchina.

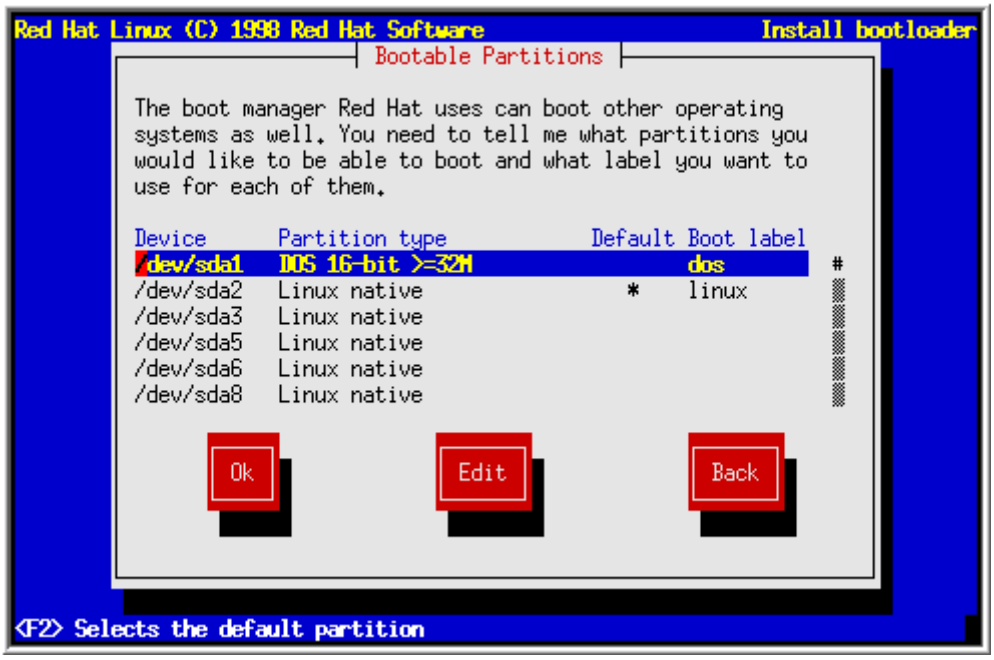
RedHat: installazione



RedHat: installazione

- LILO è soggetto ad alcune limitazioni imposte dal BIOS della maggior parte dei Personal Computers:
  - può accedere solo ed unicamente hard disk che vengono gestiti dal BIOS della macchina; nelle versioni più datate i primi due hard disk IDE/EIDE ed i primi due dischi SCSI;
  - non si possono accedere in modo corretto partizioni che si trovino del tutto o in parte sopra la linea del 1024 cilindri.
- Nelle macchine più moderne questi vincoli sono stati *alleggeriti*: va verificato di volta in volta, almeno fino a quando non sarà emerso un nuovo standard.

RedHat: installazione



- Il passo finale consiste nell'assegnare una label (una etichetta) a ciascuna partizione da cui volete eseguire il boot e definire la partizione da cui LILO eseguirà il boot per default.

RedHat: installazione

- LILO memorizza la configurazione nel file **/etc/lilo.conf**:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.0.36-3
        label=linux
        root=/dev/hda5
        initrd=/boot/initrd-2.0.36-3.img
        read-only
image=/boot/vmlinuz-2.0.36-3.ORG
        label=old
        root=/dev/hda5
        initrd=/boot/initrd-2.0.36-3.img.ORG
        read-only
other=/dev/hda1
        label=diag
        table=/dev/hda
```

ed il comando **/sbin/lilo** permette di modificare in qualsiasi momento il comportamento del LILO, per esempio aggiungendo altri sistemi operativi o altre versioni del kernel.

RedHat: installazione

- Durante l'installazione verrà offerta l'occasione di creare un *dischetto* di bootstrap.

  È una immagine di boot su floppy ***personalizzata***. Contiene tutto il software di sistema (i moduli) necessari per eseguire correttamente il boot.

  Potrebbe essere necessario usarlo:

  - al posto del **LILO**; il boot viene eseguito dal floppy ma la radice viene montata dalla partizione definita sull'hard disk;
  - in caso di **emergenza**; il dischetto di boot può essere usato insieme al dischetto di ***rescue*** (che può essere creato con la stessa tecnica dell'immagine di boot di installazione) per controllare lo stato dei file systems o rimpiazzare file di sistema cancellati o danneggiati;

RedHat: installazione

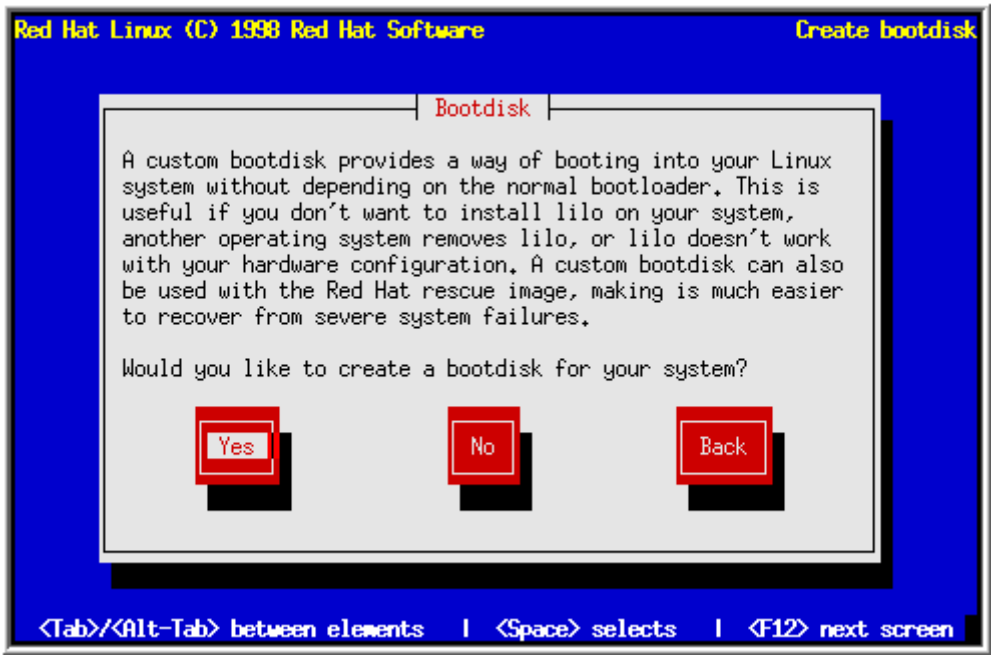
- nel caso un altro sistema operativo (per esempio Windows 95) riscriva la Master Boot Partition (ovvero esegua l'equivalente di un **fdisk /mbr**) cancellando il LILO, il dischetto di boot può essere utilizzato per eseguire il boot e reinstallare LILO.

- Successivamente alla installazione è comunque possibile creare il dischetto di boot usando il comando di sistema (una script molto istruttiva che potete tentare di decifrare)

  /sbin/mkbootdisk

- È fortemente consigliato tenere a portata di mano i dischetti di boot e rescue per risolvere situazioni in cui il sistema non riparte in modo automatico dall'hard disk (per esempio vi siete persi la password di root).

RedHat: installazione



RedHat: installazione

- Al termine dell'installazione un reboot della macchina permetterà di eseguire un ***root login*** sulla console della macchina.

- I tasti **ALT-Fn** con n=1,...,6 consentono l'accesso al login prompt su sei consoles ***virtuali***. La console numero **7** (ALT-F7) è riservata ad X Window.

```
Red Hat Linux release 5.1 (Manhattan)
Kernel 2.0.35 on an i686
login: root
Password:
Last login: Sat Apr 10 16:45:26 on tty1
You have new mail.
[root@bw01 /root]#
```

- Da questo momento è possibile iniziare a personalizzare il sistema per le proprie necessità.

Gestione degli utenti

- Uno dei compiti dell'amministratore di sistema è la gestione degli accounts.

  In un sistema UNIX l'uso dell'account **root** è riservato solo all'amministrazione ed alla gestione del sistema operativo.

  Anche l'amministratore del sistema *normalmente* lavora con un account senza privilegi ed acquisisce i diritti di **root** solo quando è effettivamente necessario.

- Una delle prime attività da eseguire dopo l'installazione consiste proprio nel definire gli utenti del sistema.

- Le caratteristiche degli utenti vengono registrate, fondamentalmente, in due files che

Gestione degli utenti

sono vitali per il corretto funzionamento del sistema:

- **/etc/passwd**

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
postgres:x:100:233:PostgreSQL Server:/u0/pgsql:/bin/bash
peppe:x:201:1001:Giuseppe Vitillaro:/home/peppe:/bin/bash
```

è la tabella delle password che mappa ***username*** su ***userid*** con il formato di record

username:password:uid:gid:gecos:homedir:shell

Gestione degli utenti

- **/etc/group**

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
mem:x:8:
kmem:x:9:
wheel:x:10:root
mail:x:12:mail
news:x:13:news
uucp:x:14:uucp
man:x:15:
games:x:20:
gopher:x:30:
dip:x:40:
ftp:x:50:
nobody:x:99:
users:x:100:
thch:x:1001:
```

è la tabella dei gruppi che mappa ***groupnames*** su ***groupid*** con il formato di record

groupname:passwd:gid:members

Gestione degli utenti

- I campi della tabella **passwd**:
  - **username**

    è una stringa limitata ad una lunghezza di otto caratteri nella maggior parte dei sistemi UNIX; deve comparire una sola volta all'interno del file passwd e permette di autenticare l'utente all'atto del login;

  - **password**

    rappresenta la password *cifrata*; viene definita mediante l'uso del programma ***passwd***; se il primo catattere è un asterisco ***** l'utente non può in alcun modo eseguire il login;

  - **uid**

    lo ***userid***, un numero intero che identifica in modo univoco l'utente al sistema operativo; differenti usernames possono essere mappati sullo stesso userid;

Gestione degli utenti

- **gid**

  è il groupid, un intero che definisce il gruppo di default a cui appartiene l'utente; all'atto del login l'utente acquisirà i diritti di accesso di questo gruppo (definito nella tabella group);

- **gecos**

  deriva storicamente il suo nome dall'acronimo **G**eneral **E**lectric **C**omprehensive **O**perating **S**ystem e contiene una serie di informazioni, separate da una virgola che descrivono il nome *reale* dell'utente, il suo numero di telefono, la sua stanza e che vengono utilizzate dal programma **mail** e da **finger**;

- **homedir**

  il pathname assoluto della directory home ed anche della current working directory subito dopo il login;

Gestione degli utenti

- **shell**

  è il pathname assoluto del programma che prenderà il controllo dopo che login avrà *autenticato* l'utente; nella maggior parte dei casi è una *shell*, ma in generale qualunque comando o applicazione UNIX può comparire in questo campo;

  un modo semplice per impedire il login di utente consiste nel definire questo campo come **/bin/false**.

- Il file /etc/passwd può essere manipolato direttamente utilizzando un editor, ma, in genere, il sistema mette a disposizione delle utilità che ne semplificano e coordinano la gestione (pensate a due amministratori che aggiornano *contemporaneamente* il file).

Gestione degli utenti

- I campi della tabella **group**:
  - **groupname**

    è una stringa, in genere limitata ad otto caratteri, che compare una sola volta nel file group;
  - **password**

    è una password, opzionale, cifrata che viene gestita in Linux RedHat con il comando **gpasswd**; permette agli utenti di modificare il proprio gruppo con il comando **newgrp**;
  - **gid**

    il ***groupid***, un numero intero che identifica il gruppo al sistema operativo, corrisponde al campo gid nella tabella passwd;

Gestione degli utenti

- **members**

  è una lista di *usernames*, separata da virgole, che identifica gli utenti appartenti al gruppo; non è *necessario* elencare i gruppi di default degli utenti; gli utenti elencati acquisiscono i diritti di accesso del gruppo corrispondente.

- Anche il file /etc/group può essere manipolato direttamente con un editor e valgono le stesse osservazioni fatte per il file delle password.

- Una delle tecniche suggerite per la gestione dei gruppi consiste nel definire un nuovo gruppo, con lo stesso nome e lo stesso gid, per ogni nuovo utente del sistema ed eventualmente assegnare l'utente ai gruppi ai

Gestione degli utenti

quali deve appartenere per poter accedere ai files del sistema che gli sono necessari.

- Se in un file system sono presenti files appartenti a *uid* o *gid* non più esistenti nelle tabelle passwd e group il comando “ls -l” visualizzerà le informazioni numeriche:

```
# ls -l
total 0
-rw-r--r--   1 peppe    3011            0 Apr 22 12:10 nogroup
-rw-r--r--   1 7896     811             0 Apr 22 12:10 noinfo
-rw-r--r--   1 4777     thch            0 Apr 22 12:10 nouser

# find . -nouser
./nouser
./noinfo

# find . -nogroup
./nogroup
./noinfo
```

ed il comando **find** permetterà di rintracciate i files *orfani* di un utente e/o di un gruppo.

Gestione degli utenti

- I programmi utilizzati per la gestione degli utenti variano *significativamente* da un sistema UNIX all'altro.

  AIX, Sun Solaris, FreeBSD, SCO UNIX, le distribuzioni Linux, pur condividendo essenzialmente la stessa struttura delle tabelle passwd e group differiscono sostanzialmente nelle interfacce disponibili per la gestione degli accounts.

  È fortemente consigliato individuare e leggere con attenzione la documentazione e le pagine del manuale del sistema in uso prima di avventurarsi nella definizione degli utenti.

  I nostri esempi verteranno essenzialmente sull'interfaccia command line disponibile nel sistema RedHat 5.1.

Gestione degli utenti

- Una nota di attenzione: la presenza della password *cifrata* nella tabella delle password rappresenta in qualche modo una *security exposure*.

  Il diritti di accesso del file passwd

```
-rw-r--r--   1 root     root          847 Mar 30 10:08 /etc/passwd
```

  devono necessariamente permettere la lettura del file a ***tutti*** gli utenti del sistema. Infatti eliminare il privilegio di lettura per gli utenti non privilegiati avrebbe come effetto immediato la indisponibilità della mappa tra usernames ed userid con ovvie conseguenze.

```
root:EHxmN.Myu3qdg:0:0:root:/root:/bin/bash
peppe:YmhRBVKVTaJso:201:1001::/home/peppe:/bin/bash
tuser:2c40.lCbpEDXc:501:501::/home/tuser:/bin/bash
```

  Sebbene la cifratura impedisca una *semplice* lettura della passwords esistono metodi, più o

Gestione degli utenti

meno onerosi dal punto di vista computazionale, che permettono di ***decifrare*** una password.

Si può andare dal confronto della password cifrata con passwords *banali* estratte da dizionari più o meno cospicui ad un attacco diretto all'algoritmo di cifratura (DES per il sistema Linux: un algoritmo *abbastanza* facile da rompere).

In ogni caso la disponibilità delle passwords cifrate potrebbe rappresentare, per alcuni utenti, un invito irresistibile ad un attacco diretto alla sicurezza del sistema.

In effetti la disponibilità della password è necessaria solo ad un ristretto numero di programmi (come ***login***, ***passwd***, etc) coinvolti

Gestione degli utenti

nell'autenticazione e nella gestione del sistema.

- Molti sistemi moderni, tra cui anche Linux, implementano un modello in cui la password cifrata non compare direttamente nel file /etc/passwd, ma viene memorizzata in un file *ausiliario*, accessibile solo ad utenti privilegiati.

  Gli schemi possono essere differenti, ma l'idea è sempre la stessa. Creare un meccanismo di sicurezza che nasconda, che metta in *ombra* (shadowing) le passwords cifrate.

- Molte distribuzioni Linux, tra le quali anche la RedHat (vedi il package ***shadow-utils***) implementano questo meccanismo.

Gestione degli utenti

- Sfortunatamente lo shadowing non è attivo nella configurazione di installazione e va attivato esplicitamente.

- Il comando ***pwconv*** (ed il suo analogo ***grpconv*** per il file dei gruppi) permettono di creare i files **/etc/shadow** e **/etc/gshadow**

```
-r--------   1 root     root          681 Mar 30 10:08 /etc/shadow
-r--------   1 root     root          340 Mar 30 10:07 /etc/gshadow
```

```
root:rtKETd1K1nFpG:10640:0:99999:7:::
bin:*:10640:0:99999:7:::
daemon:*:10640:0:99999:7:::
...
...
peppe:20oEJ1neu/B.U:10640:0:99999:7:::
```

che implementano il meccanismo di shadowing per le tabelle passwd e group.

I programmi di autenticazione e gestione delle password sono strutturati per tener conto *automaticamente* della scelta effettuata.

Gestione degli utenti

- Il comando utilizzato per la creazione di un nuovo utente nella distribuzione RedHat è:

  /usr/sbin/useradd

  il cui uso, descritto nella corrispondete pagina del manuale,

```
useradd [-c comment] [-d home_dir]
        [-e expire_date] [-f inactive_time]
        [-g initial_group] [-G group[,...]]
        [-m [-k skeleton_dir] | -M] [-s shell]
        [-u uid [ -o]] [-n] [-r] login

useradd -D [-g default_group] [-b default_home]
        [-f default_inactive] [-e default_expire_date]
        [-s default_shell]
```

  consente di aggiungere un utente e crearne in modo automatico la home directory.

- Il comando

  /usr/sbin/groupadd

Gestione degli utenti

permette la creazione di un nuovo gruppo

```
groupadd [-g gid [-o]] [-r] [-f] group
```

- L'uso del comando **useradd** può essere molto semplice. Vediamo la definizione del nuovo utente ***norbert*** appartenente al gruppo di default ***norbert***

```
# useradd norbert
#
# grep norbert /etc/passwd
norbert:x:2002:2002::/home/norbert:/bin/bash
#
# grep norbert /etc/shadow
norbert:!!:10703:0:99999:7:::
#
# grep norbert /etc/group
norbert:x:2002:
```

```
# passwd norbert
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
#
# grep norbert /etc/passwd
norbert:x:2002:2002::/home/norbert:/bin/bash
#
# grep norbert /etc/shadow
norbert:EeZsYePOo6ycU:10703:0:99999:7:-1:-1:134529860
```

Gestione degli utenti

```
# id norbert
uid=2002(norbert) gid=2002(norbert)
#
# ls -ld /home/norbert
drwx------   2 norbert  norbert      4096 Apr 22 13:36 /home/norbert
#
# ls -la /home/norbert
total 28
drwx------   2 norbert  norbert      4096 Apr 22 13:36 .
drwxr-xr-x  11 root     root         4096 Apr 22 13:36 ..
-rw-r--r--   1 norbert  norbert      3768 Apr 22 13:36 .Xdefaults
-rw-r--r--   1 norbert  norbert        24 Apr 22 13:36 .bash_logout
-rw-r--r--   1 norbert  norbert       220 Apr 22 13:36 .bash_profile
-rw-r--r--   1 norbert  norbert       124 Apr 22 13:36 .bashrc
-rw-rw-r--   1 norbert  norbert      3336 Apr 22 13:36 .screenrc
```

Il comando useradd ha, *correttamente*, definito un nuovo utente, *norbert*, un nuovo gruppo che ha lo stesso nome e gid e la home directory.

Inoltre ha copiato, da **/etc/skel**, i *templati* dei dot files necessari al corretto funzionamento della shell e di X Window nella nuova home directory **/home/norbert**.

Gestione degli utenti

- Non sempre i default di installazione sono appropriati per il nuovo sistema che si sta personalizzando.

- Di volta in volta potrebbe essere necessario analizzare l'ambiente dell'utente finale e decidere quale schema di userid e groupid implementare.

- Nel caso di *migrazione* da un sistema, magari di architettura differente, già esistente potrebbe essere necessario trasferire utenti, gruppi, password, homes dal vecchio sistema al nuovo.

- I dot files in /etc/skel vanno personalizzati per le esigenze dei propri utenti.

Gestione degli utenti

- I comandi

  /usr/sbin/userdel

  /usr/sbin/usermod

  permettono, rispettivamente, di cancellare e modificare un utente del sistema.

```
# usermod -c "Norberto Bobbio,Room A10,075/555-5531" norbert
#
# grep norbert /etc/passwd
norbert:x:2002:2002:Norberto Bobbio,Room
A10,075/555-5531:/home/norbert:/bin/bash
#
# finger norbert
Login: norbert                          Name: Norberto Bobbio
Directory: /home/norbert                Shell: /bin/bash
Office: Room A10, 075/555-5531
Never logged in.
No mail.
No Plan
```

- La cancellazione di un utente dal sistema può *sembrare* una operazione semplice come l'immissione di un comando:

Gestione degli utenti

```
# userdel norbert
#
# grep norbert /etc/passwd
# grep norbert /etc/shadow
# grep norbert /etc/group
norbert:x:2002:
#
# ls -ld /home/norbert
drwx------   2 2002     norbert      4096 Apr 22 13:36 /home/norbert
#
```

In effetti la cancellazione di un utente si presenta molto più problematica:

- cosa fare del suo gruppo?
- che destinazione avranno i files ad esso appartenti che sono ***ancora*** presenti nel sistema?
- possiamo riutilizzare lo userid che corrispondeva allo username che abbiamo cancellato?
- esistevano programmi setuid o setgid che appartenevano all'utente?
- che destinazione avrà la sua mailbox?

Gestione degli utenti

- Occorre tener presente che in installazioni medio grandi, distribuite su rete, ad un singolo utente possono appartenere migliaia di files su decine di file systems di macchine differenti.

- Spesso la soluzione migliore consiste nel ***disabilitare*** l'account senza cancellarlo effettivamente, continuando a mantenere l'utente attivo almeno nella *storia* del sistema.

- In ogni caso è necessario fornire una risposta ad almeno alcuni dei problemi proposti, prima di cancellare un account dal sistema.

- Il comando ***useradd*** può essere utilizzato in modo più complesso e il suo uso può essere mediato da un shell script opportuna:

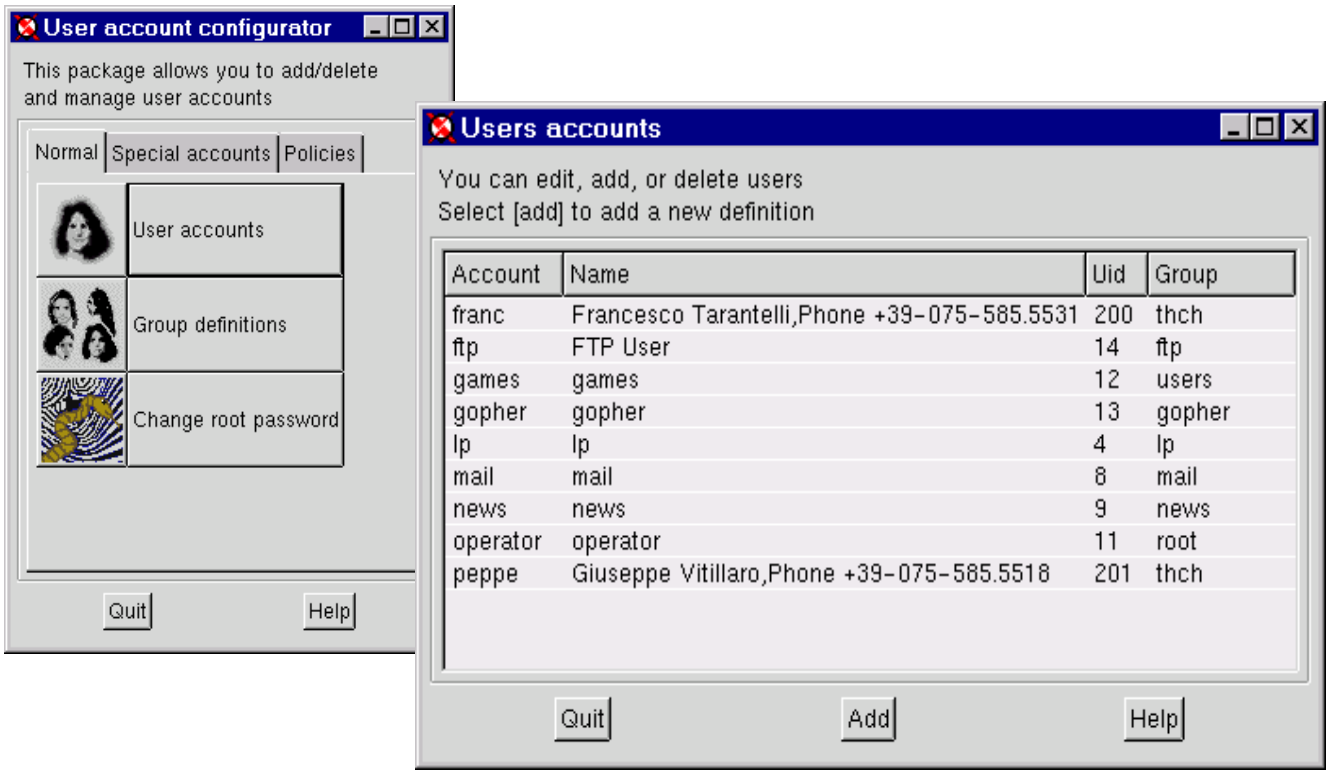
Gestione degli utenti

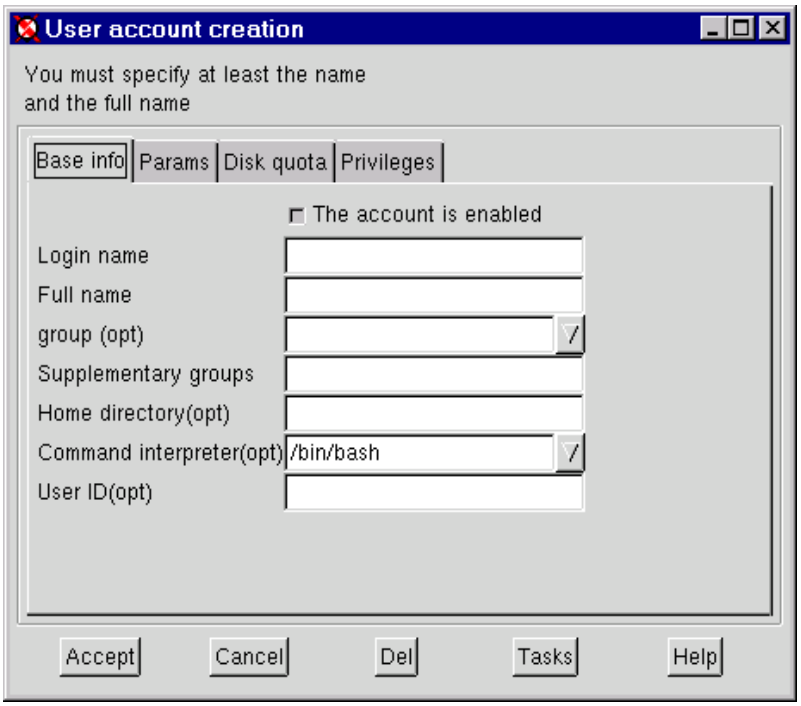
```
# useradd -u 3001 -g users -G thch,tuser -d /u1/blarg -s /bin/csh -p
"9k2n4fIL3DSol" -c "John Blarg,Room A21, Phone 075/555-3331" blarg
#
# grep blarg /etc/passwd
blarg:x:3001:100:John Blarg,Room A21, Phone
075/555-3331:/u1/blarg:/bin/csh
#
# grep blarg /etc/shadow
blarg:9k2n4fIL3DSol:10703:0:99999:7:::
#
# grep blarg /etc/group
thch:x:1001:blarg
tuser:x:2001:blarg
#
# id blarg
uid=3001(blarg) gid=100(users) groups=1001(thch),2001(tuser)
```

- Nella distribuzione RedHat è possibile gestire utenti e gruppi utilizzando una *graziosa* interfaccia grafica disponibile da **control-panel** o dal tool **linuxconf**.

  L'uso di una interfaccia utente *sofisticata* non può esimere un amministratore competente dal conoscere i dettagli relativi ai files fondamentali che definiscono gli utenti.

Gestione degli utenti





Bootstrap ed Inizializzazione

- Lo scopo di questa sezione è quello di guadagnare una più profonda comprensione del meccanismo attraverso il quale un sistema UNIX esegue il bootstrap e viene inizializzato.

  Ancora una volta è necessario puntualizzare che spesso le implementazioni tra le diverse piattaforme Unix differiscono in modo notevole.

  In generale lo schema è fondamentalmente lo stesso, ma sono i dettagli (che è necessario conoscere in modo approfondito) che forniscono un controllo completo su una delle fasi più importanti della vita del sistema.

- I nostri esempi saranno ancora una volta basati sulla distribuzione Linux RedHat 5.1.

Bootstrap ed Inizializzazione

- La sequenza di boot, della quale abbiamo già parlato nella sezione relativa all'installazione, viene iniziata dal *firmware* congelato nella memorie a sola lettura (ROM, PROM, EPROM, Flash EPROM, ROS, CMOS, nvram, etc.) della macchina.

- Nei Personal Computer (per motivi storici relativi alla loro derivazione dai PC IBM XT/AT) il sistema che gestisce la inizializzazione, la autodiagnostica ed il bootstrap (anche noto in ambiente IBM come **IPL**, Initial Program Loading) è denominato **BIOS** (Basic Input Output System).

Bootstrap ed Inizializzazione

- È il **BIOS** della macchina che si occupa della fase iniziale del caricamento in memoria del sistema operativo.

- Il kernel del sistema operativo Linux utilizza il loader LILO per gestire le fasi più avanzate del caricamento in memoria del kernel da una partizione dell'hard disk.

  In effetti il kernel del sistema, contenuto nel file **/boot/vmlinuz** altro non è che l'immagine compressa di un *bootable floppy*:

```
dd if=/boot/vmlinuz of=/dev/fd0 bs=8k
```

  a cui il LILO fornisce il supporto necessario per il boot da una partizione.

Bootstrap ed Inizializzazione

- Dopo il caricamento in memoria, il kernel esegue una fase di probing ed inizializzazione dell'hardware emettendo sulla console una serie di messaggi (che possono essere esaminati con calma, dopo il boot, con il comando **/bin/dmesg**) che informano l'utente sull'hardware disponibile e su eventuali errori.

- Al termine dell'operazione di boot, il kernel crea il primo processo del sistema:

  - ha pid (process-id) uguale ad **1;**
  - è l'unico processo non originato da una ***fork***;
  - viene eseguito il programma **/sbin/init***;*

  e con questo atto il sistema diviene finalmente *pienamente operativo*.

Bootstrap ed Inizializzazione

- Il programma **init**, padre di tutti i processi, può essere implementato in modi più o meno complessi.

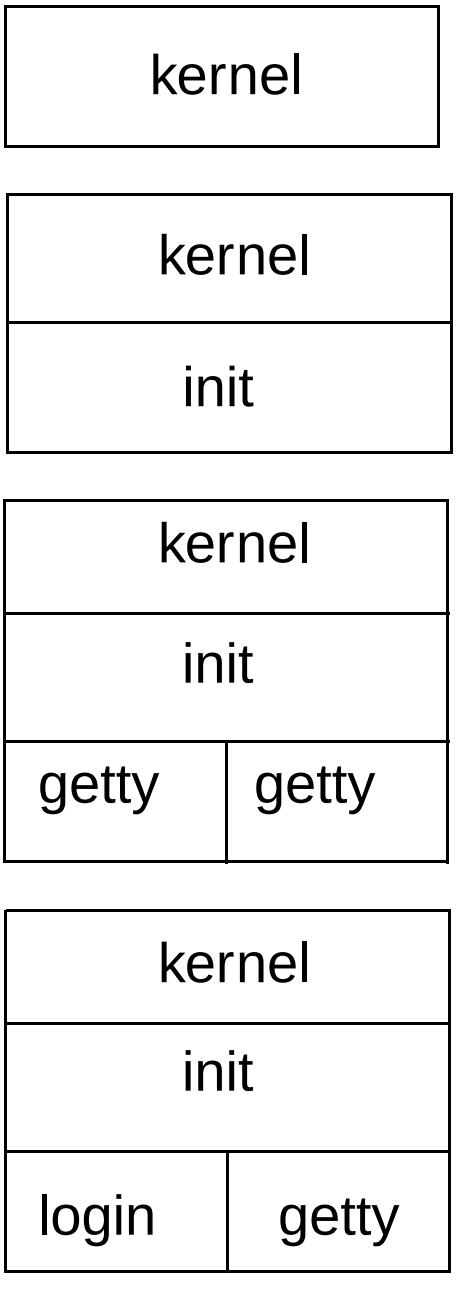
  Ciò può comportare notevoli differenze tra un sistema operativo e l'altro.

  Nello schema più semplice, utilizzato ad esempio dal FreeBSD 2.2, init non fa altro che eseguire la shell script ***/etc/rc*** (**r**untime **c**onfigurations o **r**esource **c**onfiguration files?).

- Negli schemi più complessi, derivati dal System V ed in uso su Linux RedHat, **init** esegue una sequenza di comandi, memorizzati in una serie di scripts di startup.

Bootstrap ed Inizializzazione

Inizializzazione di un sistema FreeBSD 2.2



boot
caricamento del kernel

creazione di init
ed esecuzione di /etc/rc

abilitazione console
e terminali locali

login di un utente
autenticato da userid
e password

Bootstrap ed Inizializzazione

- Il programma **init** del sistema RedHat implementa un meccanismo più complesso e sofisticato.

- Il file di configurazione ***/etc/inittab*** definisce le azioni che devono essere compiute da **init** per una serie di **runlevels**: **0123456Ss**.

- Ad ogni runlevel **init** assegna una shell script che ha il compito di eseguire la sequenza di comandi necessari a porre il sistema nello stato desiderato.

- Le shell scripts di configurazione possono essere personalizzate per essere adeguate alle necessità di un particolare sistema operativo.

Bootstrap ed Inizializzazione

```
#
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud::once:/sbin/update

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Bootstrap ed Inizializzazione

```
# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting
Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown
Cancelled"


# Run gettys in standard runlevels
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

- Nel caso del RedHat la scelta degli implementatori della distribuzione (del tutto arbitraria) è stata quella di utilizzare una sola script ***/etc/rc.d/rc*** che in pratica *pilota* l'ingresso del sistema in ogni runlevel.

Bootstrap ed Inizializzazione

- Il kernel crea il processo **1** che cercherà di caricare, con una primitiva di **exec**, nell'ordine,

  /etc/init /bin/init /sbin/init /etc/rc

  nel proprio spazio di indirizzi di memoria virtuale. In un sistema *standard* ciò si traduce nell'esecuzione del programma **/sbin/init**.

- Come prima azione **/sbin/init** esegue il comando definito dalla entry

```
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
```

  del file inittab: **/etc/rc.d/rc.sysinit** e quindi fa entrare il sistema nel runlevel definito da

```
id:3:initdefault:
```

  ovvero porrà il sistema nello stato "Full Multiuser Mode".

Bootstrap ed Inizializzazione

- In termini più concreti il sistema eseguirà la shell script assegnata dalla entry

```
l3:3:wait:/etc/rc.d/rc 3
```

  al runlevel **3** (in questo caso), ovvero il comando:

  /etc/rc.d/rc 3

- Un esame più attento della script **/etc/rc.d/rc** rivela uno schema abbastanza semplice da decifrare.

  La script tenterà di eseguire, nell'ordine, le scripts presenti nella directory **/etc/rc.d/rc3.d** il cui filename inizia con **Knn** e con **Snn**, dove **nn** è un numero di due cifre.

Bootstrap ed Inizializzazione

- I filenames che iniziano con **K** (Kill Scripts) verranno eseguiti con l'argomento *stop*.

  I filenames che iniziano con **S** (Start Scripts) verranno eseguiti con l'argomento *start*.

- Le scripts che si trovano nelle directories **/etc/rc.d/rc***n***.d** (con *n=0,1,2,3,4,5,6*) sono in genere link simbolici a files che sono memorizzati nella directory **/etc/rc.d/init.d**.

  Questo *trucco* consente di gestire una sola copia delle scripts che vengono così *riutilizzate* per differenti *runlevels*.

- La *numerazione* stabilisce l'ordine di esecuzione.

Bootstrap ed Inizializzazione

- Ogni volta che **init** entra in un runlevel (come conseguenza della configurazione o per un comando ricevuto dal programma **/sbin/telinit**) eseguirà prima le Kill scripts e quindi le Start scripts contenute nella directory **/etc/rc.d/rc*n*.d** corrispondente.

  In pratica un insieme di sottosistemi verranno fermati dalle Kill scripts ed un diverso insieme prenderà il suo posto, lanciati dalle Start scripts.

- Il sistema viene fermato ponendolo nel runlevel **0** (**halt**) e ribootstrappato portandolo al runlevel **6** (**reboot**). È esattamente l'effetto dei comandi "**shutdown -h now**" e "**shutdown -r now**".

Bootstrap ed Inizializzazione

- Una rapida esplorazione della directory /etc/rc.d ci può consentire una migliore comprensione dello schema:

```
bw01 /etc/rc.d(10)-> ls -CF
init.d/         rc.local*       rc0.d/          rc3.d/          rc6.d/
network.local*  rc.local.ORG*   rc1.d/          rc4.d/
rc*             rc.sysinit*     rc2.d/          rc5.d/
```

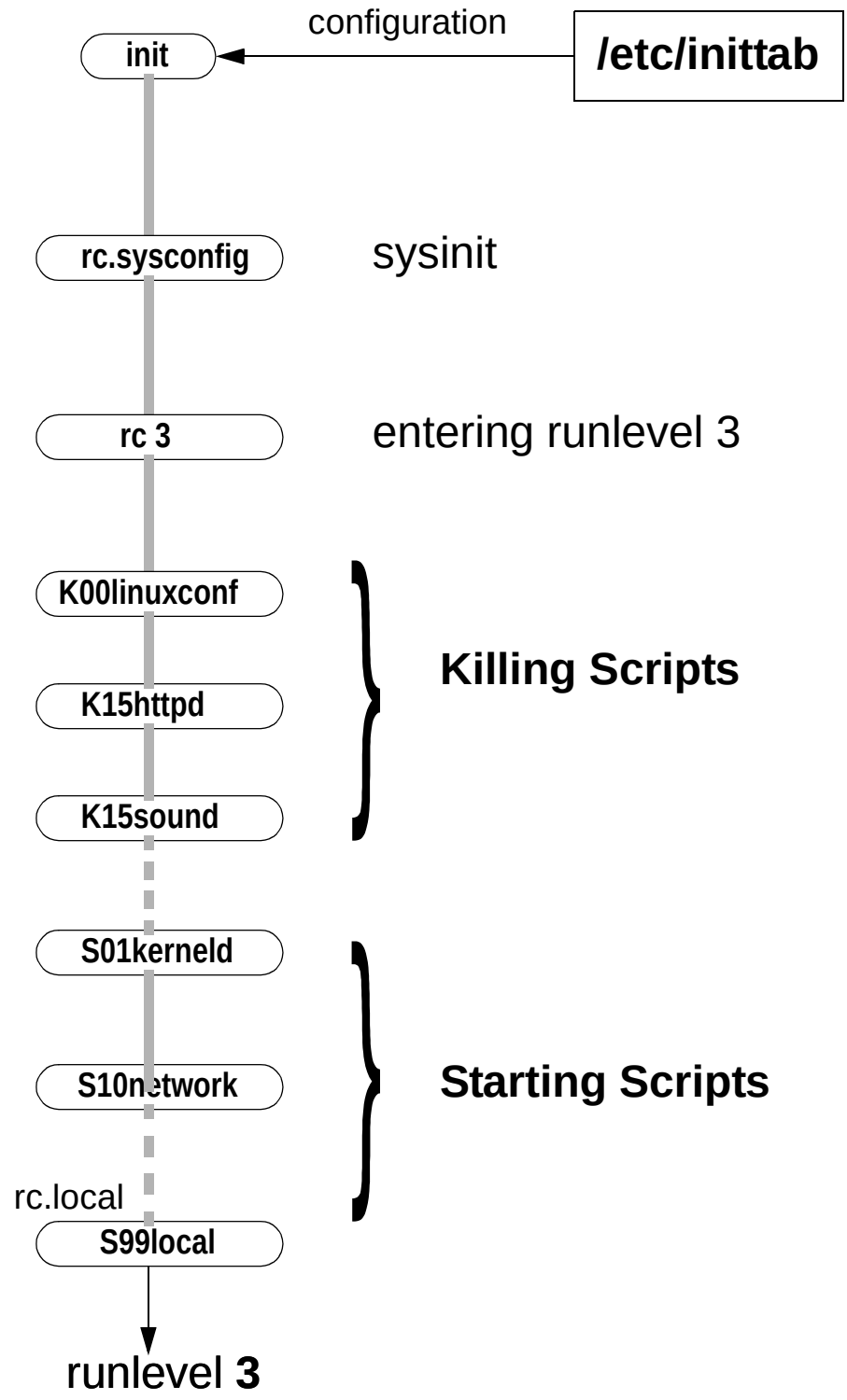
```
bw01 /etc/rc.d/init.d(14)-> ls -CF
apmd*          halt*          mcserv*        random*        sound*
atd*           httpd*         named*         routed*        syslog*
bootparamd*    inet*          network*       rusersd*       xntpd*
crond*         kerneld*       network.ORG*   rwalld*        ypbind*
dhcpd*         keytable*      nfs*           rwhod*         yppasswdd*
functions*     killall*       nfsfs*         sendmail*      ypserv*
gated*         linuxconf@     pcmcia*        single*
gpm*           lpd*           portmap*       smb*
```

```
bw01 /etc/rc.d/rc3.d(16)-> ls -CF
K00linuxconf@   K35smb@         S10network@     S40crond@       S60rwhod@
K15httpd@       K45named@       S11portmap@     S50inet@        S65ypserv@
K15sound@       K55routed@      S13ypbind@      S55xntpd@       S66yppasswdd@
K20bootparamd@  K75gated@       S15nfsfs@       S60lpd@         S75keytable@
K30mcserv@      K92apmd@        S20random@      S60nfs@         S85gpm@
K30sendmail@    K96pcmcia@      S30syslog@      S60rusersd@     S99local@
K35dhcpd@       S01kerneld@     S40atd@         S60rwalld@
```

```
lrwxrwxrwx   1 root     root           18 Apr 18 11:01 K30sendmail ->
../init.d/sendmail
lrwxrwxrwx   1 root     root           11 Apr 14 19:03 S99local ->
../rc.local
```

Si può notare come la script **/etc/rc.d/rc.local** venga eseguita per ultima.

Bootstrap ed Inizializzazione



Bootstrap ed Inizializzazione

- La script **rc.sysinit** esegue una serie di compiti fondamentali nella inizializzazione del sistema:
  - controlla che i file systems siano *integri* mediante il comando **fsck** (paragonabile ai comandi **chkdsk** o **scandisk** di MSDOS/Win95/8);
  - monta la radice del sistema in read/write mode; subito dopo il boot la radice viene montata in read-only mode per permetterne il controllo;
  - attiva le partizioni di swap (**swapon -a**);
  - monta i file systems *locali* elencati come *automatici* nel file **/etc/fstab**;
  - attiva il meccanismo di controllo della **quota**;
  - inizializza i files necessari al caricamento da **/lib/modules** dei **kernel modules**.

Bootstrap ed Inzializzazione

/etc/fstab

```
/dev/hda5               /                       ext2    defaults        1 1
/dev/hda6               /usr                    ext2    defaults        1 2
/dev/hda9               /var                    ext2    defaults        1 2
/dev/hda8               /tmp                    ext2    defaults        1 2
/dev/hda10              /usr/local              ext2    defaults        1 2
/dev/hda11              /home                   ext2    defaults,usrquota 1 2
/dev/hda7               swap                    swap    defaults        0 0
/dev/hda1               /dos                    msdos   noauto          0 0
/dev/fd0                /mnt/floppy             ext2    noauto          0 0
/dev/cdrom              /mnt/cdrom              iso9660 noauto,ro       0 0
none                    /proc                   proc    defaults        0 0
```

quota

```
bw01 /home/peppe(2)-> df
Filesystem         1024-blocks  Used Available Capacity Mounted on
/dev/hda5              54760   26712    25239     51%   /
/dev/hda6            1038035  530501   454923     54%   /usr
/dev/hda9              70415   11226    55576     17%   /var
/dev/hda8              70415     100    66702      0%   /tmp
/dev/hda10           1038035  210026   775398     21%   /usr/local
/dev/hda11           4174289  114850  4017430      3%   /home
/dev/sda1            8945312  602020  7895088      7%   /u0

bw01 /home/peppe(3)-> quota
Disk quotas for user peppe (uid 201):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/hda11   30727  512000  600000            1640   51200   60000
```

- In caso di malfunzionamenti (errori dell'hardware, incoerenza dei file systems) il sistema potrebbe non essere in grado di eseguire il boot in modo ***automatico***.

Bootstrap ed Inizializzazione

- In questo caso è necessario un intervento ***manuale*** per analizzare il problema nel dettaglio ed eventualmente risolverlo.

  Utile può rivelarsi il runlevel ***single user****.*

  In questo caso il sistema modifica il suo comportamento ed invece che portarsi al runlevel configurato in inittab, si porta al runlevel **1**.

- Il single user mode può essere richiesto in fase di boot (al prompt **boot:**) con il comando:

  linux single

  e permette di eseguire il boot di un sistema *limitato* che può almeno consentire l'analisi del problema.

Bootstrap ed Inizializzazione

- Nel caso un intervento di questo tipo si rivelasse *insuffciente* l'unica soluzione può essere rappresentata dall'uso dei dischetti di ***boot*** e ***rescue***.

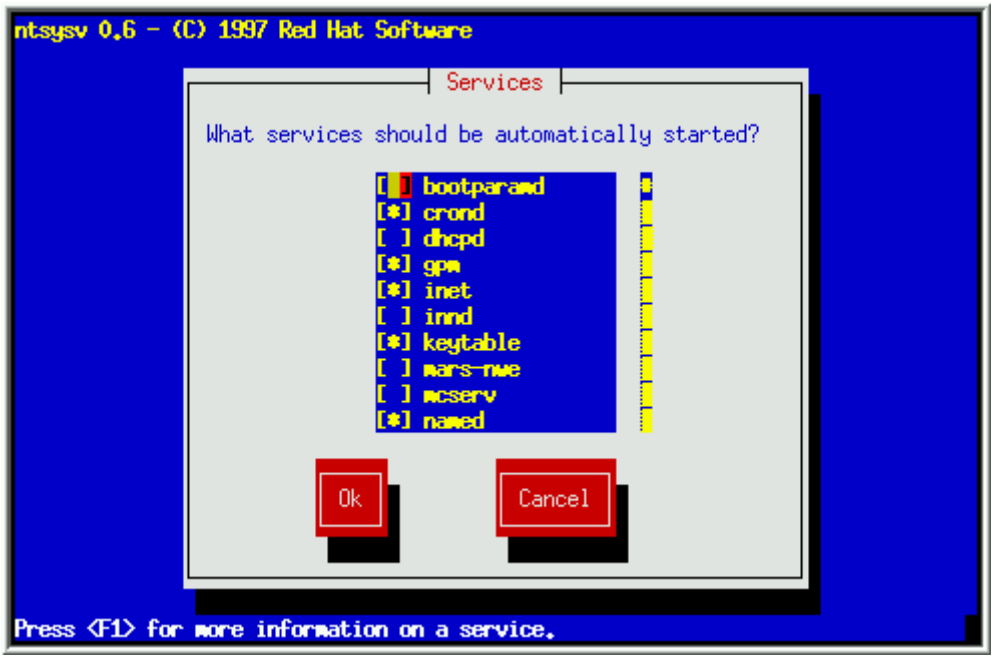
- Molto istruttiva può rivelarsi la lettura dell'***HOWTO***:

  /usr/doc/HOWTO/BootPrompt-HOWTO

  relativo ai dettagli del bootstrap.

- Per quanto riguarda le altre script di Killing/Startup un primo approccio può essere rappresentato dall'uso del control-panel o dell'utilità **/usr/sbin/ntsysv** (la cui interfacccia viene usata anche durante l'installazione).

Bootstrap ed Inizializzazione



L'interfaccia propone un elenco di *servizi* i cui nomi corrispondono alle scripts della directory **/etc/rc.d/init.d**.

L'attivazione di un servizio corrisponde alla creazione degli opportuni link simbolici nelle directories **/etc/rc.d/rc*n*.d**.

Gestione dei log di sistema

- La famiglia dei sistemi UNIX offre una grande varietà di meccanismi che consentono di registrare e monitorare l'attività del sistema operativo, dei sottosistemi applicativi e degli utenti.

- I sistemi di logging permettono di tenere sotto osservazione le principali funzioni dell'hardware e del software.

  In un sistema multitasking, operativo nelle 24 ore, connesso alla rete e con decine/centinaia di utenti attivi, all'amministratore di sistema *devono* essere forniti strumenti *automatici* che registrino *continuamente* le attività e consentano una analisi *dettagliata* dei problemi che si possono presentare.

Gestione dei log di sistema

- Particolarmente *sensibili* sono i problemi che possono verificarsi nelle seguenti *aree*:
  - errori software
  - errori hardware
  - attività di rete
  - sicurezza del sistema
  - attività degli utenti
  - attività del superuser.
- Alcuni meccanismi di logging, come il *syslog*, sono presenti praticamente in tutte le piattaforme UNIX. Altri (vedi l'**errorlog** di AIX/6000 o il **sar** del SysV) esistono solo in alcune implementazioni.

Gestione dei log di sistema

- L'idea dei base, in ogni caso, nasce dalle consoles di sistema dei *"bei tempi andati"* : una telescrivente in grado di *registrare* su un modulo continuo l'attività del sistema.

- Oggi i sistemi di logging registrano su una serie di files, in modo selettivo, le attività dei vari sottosistemi: kernel, posta elettronica, servizi di rete, autenticazione degli utenti, etc.

- La ***syslog*** *message logging facility*, nata nell'ambiente BSD, è uno dei sistemi più diffusi ed utilizzati.

- Il *daemon* (demone) ***syslogd*** si occupa di registrare una serie di messaggi provenienti da programmi opportunamente predisposti.

Gestione dei log di sistema

- Una breve *digressione* a proposito dei *demoni*.

  Nella terminologia UNIX un processo che disconnette stdout/stdin/stderr e gira in *background* allo scopo di fornire un qualche tipo di servizio viene spesso denominato ***daemon***.

  È una denominazione del tutto arbitraria e soggettiva. Sono demoni *sendmail* (servizio di posta elettronica SMTP), *inetd* (il *centralinista* TCP/IP), *lpd* (gestione dello spool di stampa), etc.

- In generale i *daemons* possono comunicare con gli utenti solo attraverso un file di log o attraverso un servizio di *logging* come il syslog.

Gestione dei log di sistema

- Il ***syslog*** nasce proprio con l'idea di centralizzare in qualche modo la gestione dei messaggi provenienti dai demoni che girano in background nel sistema.

- I programmi che intendono servirsi di questa facility devono essere esplicitamente *predisposti* attraverso l'uso della funzioni **openlog()**, **syslog()**, **closelog()**.

- Anche una shell script può utilizzare questo servizio utilizzando il comando **logger***.*

- L'idea è quella di inviare un *messaggio*, opportunamente marcato, al ***syslogd*** che lo *registrerà* in un opportuno file di ***log***.

Gestione dei log di sistema

- I messaggi vengono identificati mediante l'uso della coppia (**facility**, **level**).

- La **facility** definisce il tipo di servizio da cui proviene il messaggio:

  - auth — autenticazione
  - kern — kernel
  - local0-7 — propri dell'installazione.

- Il **level** (anche noto come **priority**) dichiara l'*importanza* del messaggio:

  - crit — critico
  - debug — messaggio di debugging
  - info — messaggio informativo.

Gestione dei log di sistema

- Il file di configurazione **/etc/syslog.conf** assegna ad ogni tipo di messaggio l'azione che dovrà essere corrispondentemente compiuta.

```
*.info                  /var/log/messages
mail.*                  /var/log/maillog
*.emerg                 *
```

I messaggi di livello ***info*** verranno registrati nel file **/var/log/messages**.

I messaggi della facility ***mail*** verranno registrati nel file **/var/log/maillog**.

I messaggi di livello ***emerg*** verranno inviati a tutti gli utenti login nel sistema.

- In questo modo diverse *categorie* di messaggi, per provenienza o importanza, possono condurre ad azion differenti: dalla semplice

Gestione dei log di sistema

registrazione in un file di log locale o remoto, alla stampa su una stampante di sistema o addirittura all'invio automatico di un e-mail o di un messaggio telefonico ad uno degli amministratori o operatori di sistema.

- La configurazione e la gestione dei log può essere complessa, ma, tranne nei casi più semplici di uso personale del sistema, non può essere evitata.

- La distribuzione RedHat installa una configurazione *minimale* che va ovviamente personalizzata per essere adattata alle proprie esigenze.

Gestione dei log di sistema

/etc/syslog.conf

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none                          /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  /var/log/maillog

# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                                 *

# Save mail and news errors of level err and higher in a
# special file.
uucp,news.crit                                          /var/log/spooler
```

- Il programma di gestione della posta, ***sendmail***, registra le operazioni compiute nel file di log **/var/log/maillog**:

```
Apr 26 19:13:41 bw01 sendmail[2428]: TAA02428: from=peppe, size=39, class=0, pri=3
0039, nrcpts=1, msgid=<199904261713.TAA02428@bw01.hpc.thch.unipg.it>, relay=peppe@
localhost
Apr 26 19:13:41 bw01 sendmail[2430]: TAA02428: to=peppe@unipg.it, ctladdr=peppe (2
01/1001), delay=00:00:00, xdelay=00:00:00, mailer=nullclient, relay=rs5.thch.unipg
.it. [141.250.9.2], stat=Sent (TAA28206 Message accepted for delivery)
```

Gestione dei log di sistema

- I messaggi relativi ad errori di autenticazione o che possono influenzare la sicurezza del sistema vengono registrati in **/var/log/secure**:

```
Apr 25 19:10:01 bw01 in.rlogind[1664]: connect from phoenix.thch.unipg.it
Apr 26 10:08:09 bw01 in.rlogind[2142]: connect from bw02
Apr 26 10:25:39 bw01 in.telnetd[2169]: connect from simbad.thch.unipg.it
```

- Una possibile definizione di messaggi *locali*

```
#
local1.info     /var/log/log1
local1.notice   /var/log/log2
```

nel syslog.conf potrà essere attivata inviando il *segnale* di *hangup* (il segnale 1) al demon syslogd:

```
# touch /var/log/log1 /var/log/log2
# ps auxwww | grep syslogd
root       245  0.0  0.2   904   580  ?  S   Apr 24   0:01 syslogd
# kill -1 245
#
# fuser /var/log/log1
/var/log/log1:          245
# fuser /var/log/log2
/var/log/log2:          245
#
```

Gestione dei log di sistema

- Il comando ***logger*** può essere utilizzato per *testare* la configurazione e/o per inviare messaggi di log nei due files log1, log2:

```
# logger -p local1.info "test of local1.info"
#
# logger -p local1.notice "test of local1.notice"
```

che produrrà i seguenti messaggi di log:

```
# cd /var/log
#
# tail -f log1
Apr 26 19:26:31 bw01 peppe: test of local1.info
Apr 26 19:26:48 bw01 peppe: test of local1.notice
# (ctl-C)
#
# tail -f log2
Apr 26 19:26:48 bw01 peppe: test of local1.notice
# (ctl-C)
```

La priorità ***info*** è maggiore della priorità ***notice***. Al file ***log1*** verranno inviati i messaggi con priorità info e notice. Al file ***log2*** verranno inviati solo i messaggi notice.

Gestione dei log di sistema

- Ovviamente i messaggi di ***log*** tendono ad accumularsi nel tempo ed, ovviamente, occupano spazio disco.

  In sistemi molto *attivi* il syslogd può generare quantità molto grandi di informazioni che, dopo un certo periodo di tempo, devono essere eliminate dal sistema.

- Ogni sito UNIX implementa una propria ***politica*** di *rotazione* dei file di logs. Si può semplicemente *troncare* il file di log dopo un certo periodo di tempo, lo si può salvare su un supporto di storage secondario, si possono conservare i file di logs di un certo numero di periodi (giorni, settimane, mesi), etc.

Gestione dei log di sistema

- Il sistema RedHat implementa una ***policy*** di default utilizzando il ***cron daemon***.

  Il ***crond*** è un demone presente nella stragrande maggioranza dei sistemi UNIX che permette di eseguire azioni con scadenza ***periodica***.

  Ogni utente del sistema, purchè abilitato all'uso, può utilizzare cron per far partire un comando (un programma o una shell script) periodico (ogni ora, ogni giorno, etc.).

- Senza addentrarci nel funzionamento di cron, la distribuzione RedHat è configurata per fa partire una serie di scripts con periodicità oraria, giornaliera, settimanale e mensile.

Gestione dei log di sistema

- Il file **/etc/crontab** definisce i dettagli della configurazione:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

  - le scripts nella directory /etc/cron.hourly verranno lanciate il primo minuto di ogni ora;
  - le scripts nella directory /etc/cron.daily verranno lanciate ogni giorno alle 4:02am;
  - le scripts nella directory /etc/cron.weekly verranno lanciate la notte tra sabato e domenica alle 4:22am;
  - le scripts nella directory /etc/cron.monthly verranno lanciate il primo di ogni mese alle 4:42am.

Gestione dei log di sistema

- Una rapida analisi della directory **/etc/cron.daily** permette di comprendere lo schema di rotazione dei log di sistema:

```
#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf
```

  La script **/etc/cron.daily/logrotate** esegue il programma **/usr/sbin/logrotate** ogni giorno alle 4:02am, con i privilegi di root.

- Il programma logrotate legge dal file **/etc/logrotate.conf** una serie di definizioni che ruotano, con diverse scadenze, i file di logs contenuti nella directory **/var/log**.

- Per rotazione si intende che i vecchi log vengono *rinominati* periodicamente per un

Gestione dei log di sistema

numero di ***periodi*** predeterminato dopodichè logrotate riutilizza gli stessi nomi, ricoprendo i files di log più vecchi.

```
# ls -lt pacct*
-rw-r--r--   1 root     root        36712 Apr 26 20:00 pacct
-rw-r--r--   1 root     root         4107 Apr 26 04:02 pacct.1.gz
-rw-r--r--   1 root     root         6562 Apr 25 04:02 pacct.2.gz
-rw-r--r--   1 root     root         5886 Apr 24 04:02 pacct.3.gz
-rw-r--r--   1 root     root         6162 Apr 23 04:02 pacct.4.gz
-rw-r--r--   1 root     root         7119 Apr 22 04:02 pacct.5.gz
```

- Questo schema, relativamente semplice, può funzionare solo a condizione che non si vogliano conservare per lunghi periodi le informazioni di log.

- Nuovi programmi, differenti schemi di rotazione, personalizzazioni richiedono spesso un intervento diretto da parte dell'amministratore di sistema.

Gestione dei log di sistema

- In un sistema UNIX esistono altre fonti importanti di informazioni sulle attività del sistema.

- La distribuzione RedHat registra gli utenti login nel sistema nel file **/var/run/utmp**. Sono le informazioni che vengono elaborate dal prorgramma **who**:

```
# who
peppe    ttyp0    Apr 26 10:25 (simbad.thch.unipg.it)
peppe    ttyp1    Apr 27 11:40 (phoenix.thch.unipg.it)
```

  Su diverse piattaforme il file **utmp** può essere posizionato in differenti directories (/var/adm, /etc, ...), ma, in genere, svolge sempre lo stesso ruolo.

Gestione dei log di sistema

- Il comando **last** elabora le informazioni relative alla *storia* dei login. Nel file **/var/log/wtmp** vengono registrate (e mantenute a meno che non venga ruotato) le attività di login sul sistema

```
# last
peppe    ttyp1        phoenix.thch.uni Tue Apr 27 11:40   still logged in
peppe    ttyp1        phoenix.thch.uni Tue Apr 27 10:25 - 10:26  (00:00)
peppe    ttyp0        simbad.thch.unip Mon Apr 26 10:25   still logged in
peppe    ttyp0        bw02             Mon Apr 26 10:08 - 10:08  (00:00)
peppe    ttyp0        phoenix.thch.uni Sun Apr 25 19:10 - 19:10  (00:00)
```

  Come è facile immaginare le informazioni in esso contenute possono divenire importanti in caso di accesso non autorizzato al sistema o di una qualche violazione delle policy del sistema.

- Nei sistemi che lo prevedono il sistema di ***accounting*** è una fonte importante di informazioni sulle attività degli utenti.

Gestione dei log di sistema

- È stato concepito per registrare l'uso delle risorse del sistema (CPU, Memoria, Disco, etc.) in modo da poter eseguire addebiti agli utenti.

- Può essere utilizzato per ricostruire, almeno in parte, le attività degli utenti. Ogni volta che viene lanciato un comando, il kernel registra un record in un file di log.

  I records contengono il nome del programma, userid e groupid dell'utente, data ed ora a cui il programma è partito, il tempo per cui è rimasto attivo, etc.

Gestione dei log di sistema

- Il comando **lastcomm** permette di visualizzare queste informazioni:

```
# lastcomm peppe
who                    peppe    ttyp1      0.01 secs Tue Apr 27 11:40
bash                   peppe    ttyp1      0.00 secs Tue Apr 27 11:40
fortune                peppe    ttyp1      0.01 secs Tue Apr 27 11:40
bash                   peppe    ttyp1      0.00 secs Tue Apr 27 11:40
hostname               peppe    ttyp1      0.01 secs Tue Apr 27 11:40
bash                   peppe    ttyp1      0.00 secs Tue Apr 27 11:40
hostname               peppe    ttyp1      0.01 secs Tue Apr 27 11:40
bash                   peppe    ttyp1      0.00 secs Tue Apr 27 11:40
id                     peppe    ttyp1      0.00 secs Tue Apr 27 11:40
bash                   peppe    ttyp1      0.00 secs Tue Apr 27 11:40
id                     peppe    ttyp1      0.01 secs Tue Apr 27 11:40
bash                   peppe    ttyp1      0.00 secs Tue Apr 27 11:40
```

Un pò di esperienza permette di interpretare queste informazioni allo scopo di comprendere quali comandi sono stati utilizzati da un particolare utente in un certo periodo di tempo e, in qualche caso, per quale scopo.

In un sistema molto attivo il file di *accounting* può diventare *rapidamente* molto grande.

Gestione dei log di sistema

- Alcune situazioni possono richiedere l'uso dei privilegi di accesso sui file di log allo scopo di impedirne un uso *indiscriminato* da parte degli utenti non privilegiati del sistema.

- Una combinazione *appropriata* di privilegi di accesso e di programmi *setuid/setgid* consentirà ad un *gruppo* di utenti privilegiati (gli amministratori, gli operatori, etc.) l'uso quotidiano di alcuni di questi tools e ne impedirà invece l'uso agli altri utenti.

Backup

- Nella distribuzione Linux RedHat sono compresi un gran numero di tools per l'archiviazione, la compressione ed il salvataggio dei files.

- Il salvataggio dei dati è una delle attività più importanti (ed anche una delle più noiose) della vita di un sistema.

  Non si potrà mai evidenziare abbastanza quanto sia fondamentale eseguire salvataggi regolari e periodici sia del sistema operativo che dei dati utente.

  Problemi hardware ed errori umani sono le cause principali della perdita di informazioni. In molti casi l'unico modo per porre rimedio a

Backup

queste situazioni è la disponibilità di un ***backup*** dei dati.

- In un sistema multitasking e multiutente il problema è ancora più serio. La cancellazione di un file può renderlo immediatamente irrecuperabile. In generale i sistemi UNIX non offrono strumenti per eseguire l'***undelete*** di un file.

  Le attività *concorrenti* degli utenti aumentano la probabilità che i blocchi resi liberi dalla cancellazione di un file vengano *immediatamente* riutilizzati dal sistema per allocare lo spazio necessario ad un file aperto da un altro processo attivo nel sistema.

Backup

- Uno degli strumenti di archiviazione e backup più diffusi in ambiente UNIX è il programma **tar** (***tape archive***).

  Il nome è in qualche modo fuorviante: ***tar*** è una utilità molto versatile che, oltre a consentire il backup ed il salvataggio di un intero sistema operativo, offre funzionalità di archiviazione che permettono ad un utente UNIX di “impachettare” interi alberi di directories in un unico file.

- Esistono molte *implementazioni* del comando tar. Il RedHat contiene la versione GNU 1.12.

  Le funzionalità disponibili possono variare significativamente tra una implementazione ed un’altra.

Backup

- La pagina del manuale dello GNU tar può, ad una prima lettura, intimidire un novizio, ma in effetti il suo uso è relativamente semplice.

  In questo esempio chiediamo a **tar** di archiviare una serie di files nel file ***arch.tar***:

```
$ tar -cvf /tmp/arch.tar qtail clean test
qtail
clean
test
```

  Il flag ***c*** ordina a tar di creare un nuovo archivio; il flag ***v*** pone tar nello stato *verbose*; il flag ***f***, seguito dal filename dell'archivio, indica a tar il file in cui archiviare i files "***qtail, clean, test***".

  tar esegue l'archiviazione ed elenca a video i files man mano che vengono salvati (l'effetto del flag ***v***).

Backup

- Chiediamo ora a tar di fornirci l'elenco dei files contenuti nell'archivio appena creato:

```
$ ls -l /tmp/arch.tar
-rw-r--r--   1 peppe    thch        10240 Apr 27 12:43 /tmp/arch.tar

$ tar -tvf /tmp/arch.tar
-rwxr-xr-x peppe/thch      114 1999-04-20 13:23 qtail
-rwxr-xr-x peppe/thch       66 1999-04-20 13:04 clean
-rwxr-xr-x peppe/thch       72 1999-04-20 12:57 test

$ tar tf /tmp/arch.tar
qtail
clean
test
```

  Possiamo notare che tar accetta gli stessi flags anche se non sono preceduti dal carattere *dash "-"*.

- Il file **arch.tar** può a questo punto essere trasferito su un altra macchina e i files in esso archiviati ripristinati (anche da un'altra implementazione del comando tar).

Backup

```
$ ls -l
total 0

$ tar -xvf /tmp/arch.tar clean test
clean
test

$ ls -l
total 2
-rwxr-xr-x   1 peppe    thch           66 Apr 20 13:04 clean
-rwxr-xr-x   1 peppe    thch           72 Apr 20 12:57 test
```

Il flag ***x*** indica l'operazione di *e**X**stract* dei files **clean, test** che vengono *ripristinati* nella current directory.

Attenzione: se nella directory corrente sono contenuti files con lo stesso filename verranno **ricoperti** (e quindi il loro contenuto andrà perduto) senza ulteriori messaggi di avvertimento.

Backup

- Il flag ***-f*** può essere seguito a sua volta dal trattino “**-**”. In questo caso tar invierà il suo output su standard output

```
$ tar -cvf - clean qtail test > /tmp/arch.tar
clean
qtail
test

$ ls -l /tmp/arch.tar
-rw-r--r--   1 peppe    thch        10240 Apr 27 19:22 /tmp/arch.tar
```

  che ovviamente può essere ridiretto su un file o inviato ad un altro programma attraverso una *pipe*.

- Il comando tar, lo si può notare dalle dimensioni del file **arch.tar**, blocca, per default, il suo output su 10Kb, ovvero 20 blocchi da 512 bytes.

Backup

- È un comportamento derivato dall'uso per cui è stato originariamente concepito, ovvero archiviare su nastro (*tape*) i files.

  L'archiviazione su nastro viene ottenuta semplicemente dirigendo l'output su un device a cui sia connessa una unità a nastro.

```
$ tar -cvf /dev/st0 clean qtail test
clean
qtail
test
```

  Per il resto tar viene utilizzato esattamente con le stesse modalità usate per l'archiviazione su file.

- Nel caso dell'uso di una unità a nastro occorre solo tener presente alcune particolarità di questo tipo di hardware.

Backup

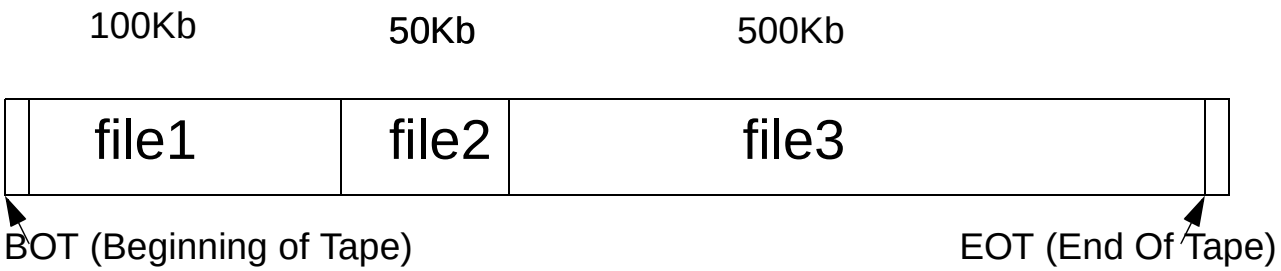
- Il nastro è un supporto su cui le infomazioni possono venir registrate solo *sequenzialmente.*

  Ogni volta che si esegue una *open()* di un device *tape*, concettualmente, viene creato un file sequenziale. Il file possiederà un *BOF* (Beginning Of File) ed un *EOF* (End Of File).

  Sul nastro fisico l'unita tape scriverà una serie di informazioni di *servizio* che le permetteranno di comprendere dove inizia e finisce il singolo file.

  Più files sequenziali possono venire accodati sullo stesso nastro fisico.

Backup



- In generale nella directory /dev sono presenti devices che permettono di scrivere un file eseguendo alla *close()* il riavvolgimento del nastro (***/dev/st0***) e devices che non eseguono il riavvolgimento (***/dev/nts0***) e che quindi permettono di accodare i files.

- I comandi:

```
$ tar -cf /dev/nst0 /usr

$ tar -cf /dev/nst0 /home
```

  salveranno gli *alberi* **/usr** e **/home** sullo stesso nastro, accodandoli.

Backup

- L'utilità **mt** (**magnetic tape**) permette di *controllare* una unità inviando (mediante la primitiva *ioctl()*) comandi all'unità a nastro.

- Esistono comandi che permettono di:
  - eseguire il riavvolgimento **rewind**
  - spostarsi di *n* files avanti **fsf**
  - spostarsi di *n* files indietro **bsf**
  - scaricare la cassetta **offline**
  - rimettere in tensione il tape **retension**
  - scrivere *n* End Of File **eof**
  - cancellare il tape **erase**
  - ottenere lo stato dell'unita **status.**

Backup

- Per esempio il comando:

  mt -f /dev/nst0 fsf 3

  permetterà di spostarsi sul BOF del quarto file sul nastro (ammesso che esista) e quindi il comando

  tar -xvf /dev/nst0

  permetterà di ripristinare il contenuto del tar contenuto nel file archiviato sul tape.

- Alcuni comandi possono essere disponibili su certi tipi di unità (per esempio offload) e non su altre.

  Da tener presente che alcune unità a nastro sono in grado di scrivere solo files *bloccati* su certe dimensioni fisiche.

Backup

- Complementari ai programmi di archiviazione sono i tools che permettono di comprimere i dati.

  Sotto UNIX sono disponibili (ed alcuni sono stati concepiti e scritti appositamente per questa piattaforma) quasi tutti i principali programmi di compressione:

  - zip, unzip **.zip**
  - zoo **.zoo**
  - gzip, gunzip **.gz**
  - compress, uncompress **.Z**

  e la lista potrebbe continuare ...

Backup

- Ovviamente la disponibilità di programmi come tar rende, nell'ambiente UNIX, molto meno importanti i tools che eseguono contemporaneamente le funzioni di archiviazione e di compressione (come **zip/unzip**).

```
$ ls -l arch.tar
-rw-r--r--   1 peppe    thch        10240 Apr 27 19:22 arch.tar
$
$ gzip arch.tar
$
$ ls -l arch.tar.gz
-rw-r--r--   1 peppe    thch          370 Apr 27 19:22 arch.tar.gz
```

- Inoltre il meccanismo di *piping* permette al programma tar di cooperare in modo molto efficiente con i compressori

```
$ tar -cvf - clean qtail test | gzip > /tmp/arch.tar.gz
clean
qtail
test
```

Backup

o con i decompressori

```
$ zcat /tmp/arch.tar.gz | tar -tvf -
-rwxr-xr-x peppe/thch       66 1999-04-20 13:04 clean
-rwxr-xr-x peppe/thch      114 1999-04-20 13:23 qtail
-rwxr-xr-x peppe/thch       72 1999-04-20 12:57 test
```

fino ad arrivare, nel caso dello GNU tar, ad una completa integrazione

```
$ tar -zcvf /tmp/arch.tar.gz clean qtail test
clean
qtail
test
$
$ ls -l /tmp/arch.tar.gz
-rw-r--r--   1 peppe    thch         361 Apr 27 20:04 /tmp/arch.tar.gz
$
$ tar -ztvf /tmp/arch.tar.gz
-rwxr-xr-x peppe/thch       66 1999-04-20 13:04 clean
-rwxr-xr-x peppe/thch      114 1999-04-20 13:23 qtail
-rwxr-xr-x peppe/thch       72 1999-04-20 12:57 test
```

- File del tipo **.tar.Z** (compress/uncompress) e **.tar.gz** (gzip) sono modi comuni per archiviare e distribuire, via anonymous FTP, packages sulla Rete.