

Gestione di base di un sistema RedHat Linux

- ◆ *L'interfaccia testo ed il sistema X Window*
- ◆ *Configurazione del sistema (utenti, rete, stampa)*
- ◆ *Personalizzazione del kernel*
- ◆ *Controllo della sequenza di avvio*
- ◆ *Strumenti per l'amministrazione di sistema*

L'interfaccia testo

All'avvio del sistema l'interfaccia di base è il *terminale* in modo testo. Inserendo il proprio username (in un sistema appena installato ci sarà solo l'utente *root*) al prompt *login:* e successivamente la relativa password si accede alla shell.

```
ingbo2 login: root
password:

Last login: Thu May  6 19:31:35 1999 from inpclott.ing.unibo.it
You have new mail.
[root@ingbo2 /root]#
```

L'interfaccia testo (2)

La shell più usata con Linux è la *bash*, un'evoluzione della classica Bourne Shell di Unix. Bash accetta tutti i comandi standard (job control, strutture di controllo, definizione dell'environment, ...) e permette di invocare gli eseguibili installati nel sistema.

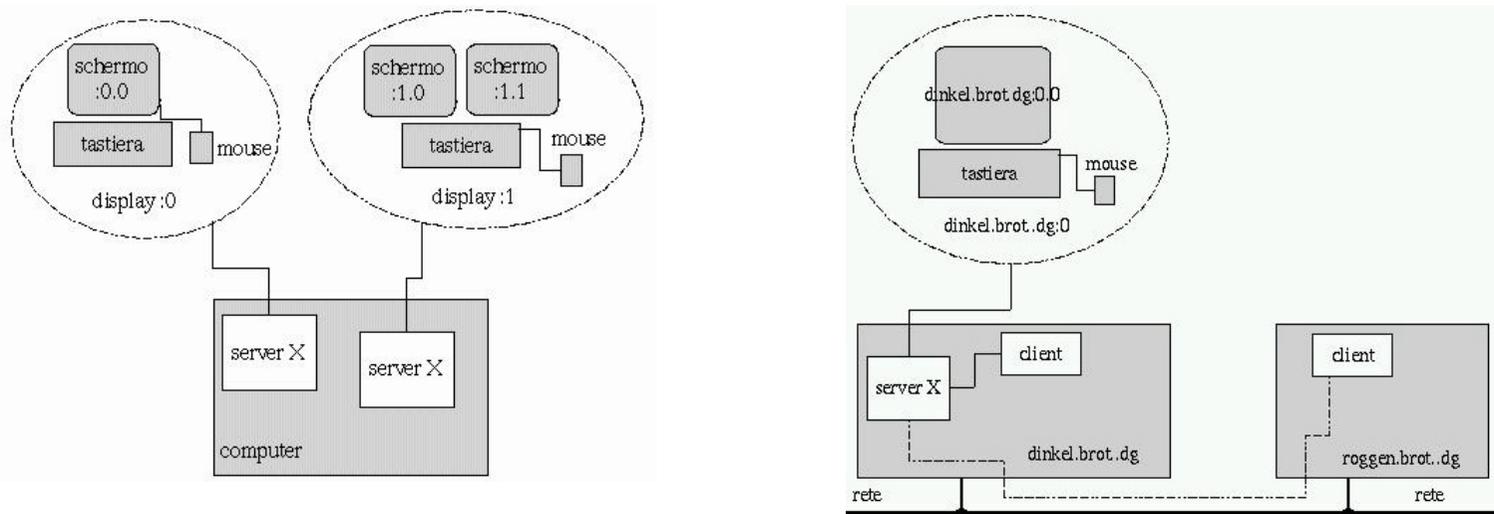
Per maggior comodità, Linux avvia contemporaneamente 6 terminali virtuali, a cui si accede con la pressione dei tasti CTRL-ALT-F1 ... CTRL-ALT-F6, e che possono essere utilizzati indipendentemente l'uno dall'altro da utenti diversi per processi diversi.

Ovviamente l'input/output non può essere contemporaneo perché utilizza la stessa tastiera e lo stesso schermo...

E' possibile definire fino a 24 terminali virtuali, (da CTRL-ALT-F1 a CTRL-ALT-SHIFT-F12).

X Window System

The X Window System (X) è un sistema client/server. I client sono applicazioni che, usando risorse di alto livello (bottoni, scrollbar, aree di testo, bitmaps, menu, ...), vogliono ottenere un certo “aspetto grafico”. Il server gestisce l'hardware grafico e si occupa quindi di trasformare le richieste del client in immagini vere e proprie.



X è uno standard multiplatforma (POSIX) → esistono server per tutti le varianti di Unix, per Windows, per OS/2 e per Macintosh. Il server gratuito per Linux è chiamato Xfree86.

Tutte le informazioni relative a X si possono trovare sui siti:

<http://www.xfree86.org>, <http://www.x.org/>, <http://www.x11.org/>

Configurazione di X

La configurazione di X viene eseguita attraverso il programma *Xconfigurator*.

Verranno chieste informazioni su:

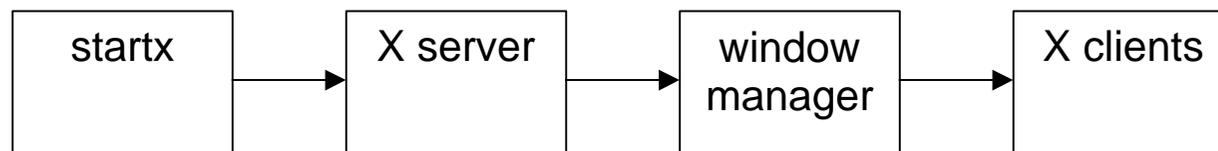
- ◆ marca e modello della scheda video (in alternativa: il *chipset*)
- ◆ marca e modello del monitor (in alternativa: le frequenze di *refresh*)
- ◆ Modi grafici desiderati

Il risultato è la generazione del file */etc/X11/XF86config*

Nota: deve essere installato il server adatto alla propria scheda grafica

Server	Chipset
8514	IBM 8514/A Boards and true clones
AGX	All XGA graphics boards
I128	#9 Imagine 128 (including Series II) boards
Mach32	ATI boards using the Mach32 chipset
Mach64	ATI boards using the Mach64 chipset
Mach8	ATI boards using the Mach8 chipset
Mono	VGA boards in monochrome
P9000	Diamond Viper (but not the 9100) and Others
S3	#9 Boards, most Diamonds, some Orchids, Others
S3V	Boards using the S3 ViRGE (including DX, GX, VX)
SVGA	Trident 8900 & 9400, Cirrus Logic, C & T, ET4000, S3 ViRGE, Others
VGA16	All VGA boards (16 color only)
W32	All ET4000/W32 cards, but not standard ET4000's

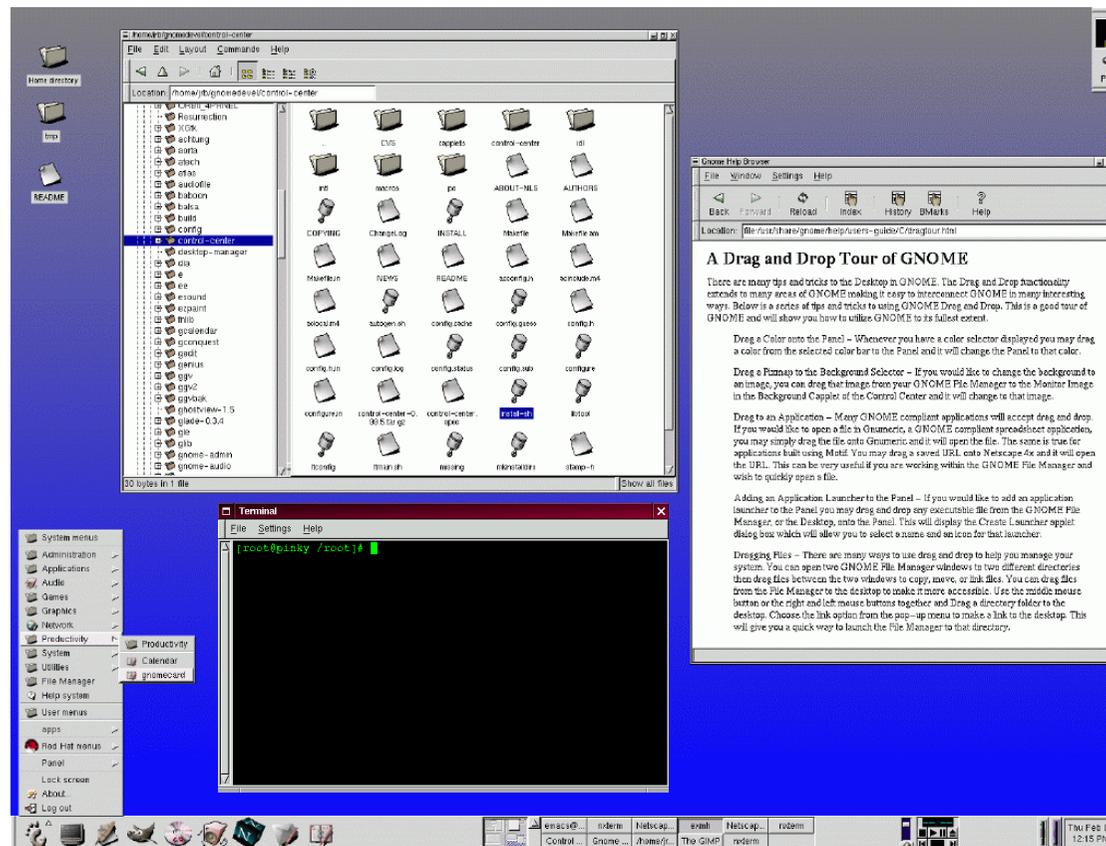
Per far partire manualmente X si usa il comando *startx*



È possibile sostituire interamente l'interfaccia testo con *xdm*, che si occupa anche del login.

Utilizzo dell'interfaccia grafica

L'interfaccia grafica fornita con Linux non si limita a X, ma comprende due sofisticati ambienti di gestione del desktop, KDE e Gnome. Questi ambienti mettono a disposizione strumenti grafici per la gestione dell'intero sistema, e ovviamente permettono di visualizzare in finestre anche tanti classici terminali testo quanti se ne desiderano.



Amministrazione di sistema

L'amministrazione di un sistema Linux può procedere in tre modi distinti.

1. Classico – uno dei punti di forza dei sistemi Unix è che il loro comportamento è completamente governato da un insieme, inevitabilmente articolato, di file di configurazione in formato testo
 - massimo controllo e trasparenza
 - possibilità di accesso da un terminale remoto
2. GUI – per semplificare il compito dell'amministratore sono stati scritti strumenti grafici interattivi, sotto X, che permettono di maneggiare i file senza doversi preoccupare di conoscere la loro posizione né la loro sintassi
 - maggiore accessibilità
 - controllo di consistenza
 - minore libertà e “consapevolezza”
3. Web – l'approccio è identico al precedente, ma anziché basarsi su di un'applicazione X si basa su di un programma collegato al server web della macchina da amministrare, rendendo più facile l'accesso remoto.

Documentazione

Le fonti di documentazione per Linux sono molteplici.

- ◆ man pages – ogni applicazione installa “pagine di manuale” relative al suo utilizzo e configurazione. Le man pages si leggono con il comando *man*
- ◆ info files – a metà strada tra la man page e l’ipertesto, si leggono con il comando *info*, che invoca l’editor emacs appositamente esteso per gestire questi file
- ◆ HOWTO – documenti specifici per la risoluzione dei più svariati problemi pratici, sono raccolti in un pacchetto e vengono tutti installati in */usr/doc/HOWTO*
Inoltre, sotto */usr/doc/HOWTO/translations/it* si possono trovare la maggior parte degli HOWTO tradotti in italiano.
- ◆ documentazione specifica – la maggior parte delle applicazioni più importanti comprende una vasta documentazione. Per sapere quali documenti vengono forniti con un pacchetto, è sufficiente il comando *rpm -qd <pacchetto>*
- ◆ aiuto on-line – possono essere utili le mailing list ed i newsgroup. All’indirizzo <http://www.redhat.com/support/mailling-lists/> sono archiviate tutte le liste gestite da RedHat, e sotto le gerarchie news:/comp.os.linux.*, news:/linux.*, news:/it.comp.linux.* si trovano i gruppi di discussione su Linux.
- ◆ PLUTO è il maggior gruppo italiano di utenti Linux: <http://pluto.psy.unipd.it/>

Man pages

Le man pages sono raggruppate in sezioni:

- (1) User commands
- (2) Chiamate al sistema operativo (programmazione di basso livello)
- (3) Funzioni di libreria, ed in particolare
 - (3C) *libc*
 - (3S) *stdio*
 - (3M) *libm*
 - (3F) *libF77*
 - (3X) varie librerie speciali
- (4) File speciali (/dev/*)
- (5) Formati dei file, dei protocolli, e delle strutture C usate per accederli
- (6) Giochi
- (7) Varie: macro, header, gerarchia del filesystem, concetti generali
- (8) Comandi di amministrazione (richiedono l'utente privilegiato)
- (n) Comandi predefiniti del linguaggio Tcl/Tk

Per avere informazioni sul comando *man* è ovviamente sufficiente usare *man man*.

<i>man -a <comando></i>	cercherà in tutte le sezioni
<i>man <sezione> <comando></i>	cercherà nella sezione specificata
<i>man -k <parola></i>	cercherà tutte le pagine attinenti alla parola specificata

Gestione dei pacchetti software

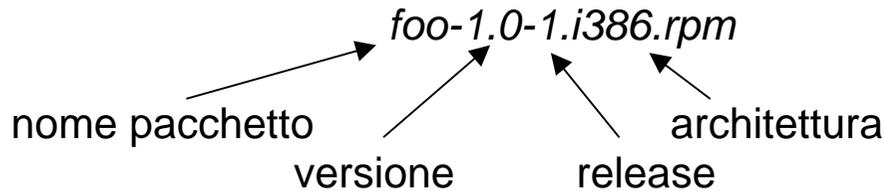
I pacchetti vengono gestiti con il programma *rpm* (RedHat Package Manager). Una lunga serie di istruzioni può essere visualizzata con il comando *rpm -help*. Informazioni approfondite sono disponibili sul sito <http://www.rpm.org>. La mailing list di discussione su rpm è archiviata al sito <http://www.redhat.com/support/mailling-lists>. Per iscriversi basta inviare un messaggio con la sola parola “subscribe” nel soggetto all’indirizzo rpm-list-request@redhat.com

RPM ha sei modi fondamentali di operazione:

1. installazione
2. disinstallazione
3. aggiornamento
4. rinfresco
5. interrogazione
6. verifica

Installazione dei pacchetti

I pacchetti RPM hanno tipicamente nomi come



Installare un pacchetto richiede l'uso dello switch *-i* (install)

```
# rpm -ivh foo-1.0-1.i386.rpm
foo #####
#
```

Possibili errori

package foo-1.0-1 is already installed
 /usr/bin/foo conflicts with file from bar-1.0-1
 bar is needed by foo-1.0-1

Possibili soluzioni

--replacepkgs
 --replacefiles
 installare bar / --nodeps

Disinstallazione dei pacchetti

Installare un pacchetto richiede l'uso dello switch `-e` (erase)

```
# rpm -e foo  
#
```

Si noti che bisogna specificare il nome del pacchetto e non il nome del file originale

L'unico possibile errore è la rottura delle dipendenze:

```
# rpm -e foo  
removing these packages would break dependencies:  
  foo is needed by bar-1.0-1  
#
```

Si può forzare l'operazione con l'opzione `--nodeps`

Aggiornamento dei pacchetti

9.2.3 Upgrading

Aggiornare un pacchetto richiede l'uso dello switch `-U` (upgrade)

```
# rpm -Uvh foo-2.0-1.i386.rpm
foo #####
#
```

- ◆ gestisce “silenziosamente” la disinstallazione della vecchia versione
- ◆ può sostituire `-i`
- ◆ salva i file di configurazione `...saving /etc/foo.conf as /etc/foo.conf.rpmsave`
→ probabilmente il nuovo formato è incompatibile con il vecchio, va risolto a mano

I tentativi di “aggiornamento” con una versione in realtà più vecchia vengono bloccati:

```
# rpm -Uvh foo-1.0-1.i386.rpm
foo package foo-2.0-1 (which is newer) is already installed
error: foo-1.0-1.i386.rpm cannot be installed
#
```

Si può forzare l'operazione con l'opzione `--oldpackage`

Rinfresco dei pacchetti

Rinfrescare un pacchetto è quasi come aggiornarlo e richiede l'uso dello switch *-F* (freshen)

```
# rpm -Fvh foo-1.2-1.i386.rpm
foo #####
#
```

La differenza rispetto all'aggiornamento consiste nel fatto che se una versione precedente del pacchetto che si vuole rinfrescare non è già installata sul sistema, non verrà installata nemmeno la nuova → utile per aggiornare un sistema avendo una nuova distribuzione

Interrogazione del database dei pacchetti

L'interrogazione del database dei pacchetti richiede l'uso dello switch `-q` (query)

Modalità base: interrogazione di un singolo pacchetto

```
# rpm -q foo
foo-2.0-1      oppure   packege foo is not installed
#
```

Opzioni per specificare i pacchetti da interrogare:

<code>-a</code>	interroga tutti i pacchetti installati
<code>-f <file></code>	interroga il pacchetto che fornisce <file> (il suo "proprietario")
<code>-p <packagefile></code>	interroga il pacchetto contenuto in <packagefile>

Opzioni per scegliere le informazioni da visualizzare:

<code>-i</code>	visualizza le informazioni generali (nome, descrizione, versione, data di creazione, stato di installazione, autore, ...)
<code>-l</code>	elenca tutti i file di cui il pacchetto è proprietario
<code>-s</code>	visualizza lo stato di tutti i file di cui il pacchetto è proprietario
<code>-d</code>	elenca tutti i file del pacchetto marcati come "documentazione"
<code>-c</code>	elenca tutti i file del pacchetto marcati come "configurazione"
<code>-v</code>	dà alle liste di file l'aspetto di un "ls -l"

Verifica dei pacchetti

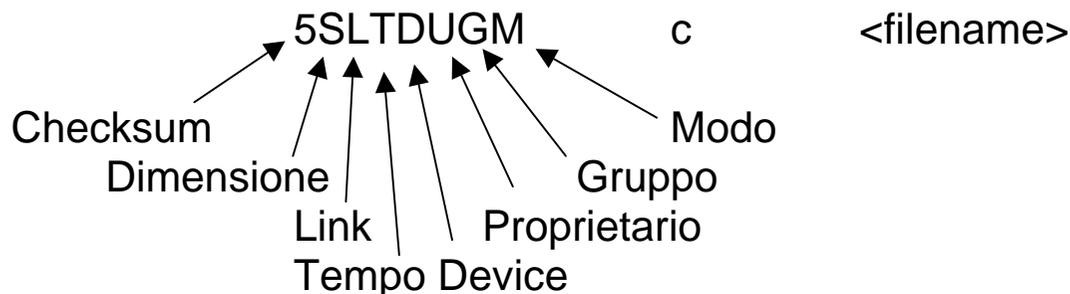
L'operazione di verifica confronta tutti i file installati da un pacchetto con gli originali, controllando dimensione, checksum MD5, permessi, tipo, proprietario e gruppo.

La verifica dei pacchetti richiede l'uso dello switch `-V` (verify)

Valgono le stesse opzioni illustrate per specificare i pacchetti da interrogare.

Nota: normalmente la verifica avviene sul database dei pacchetti installati. Se si sospetta che proprio il database sia corrotto, con `rpm -Vp foo-1.0-1.i386.rpm` si può confrontare il pacchetto installato `foo` con il file di installazione `foo-1.0-1.i386.rpm`.

L'output è in questo formato:



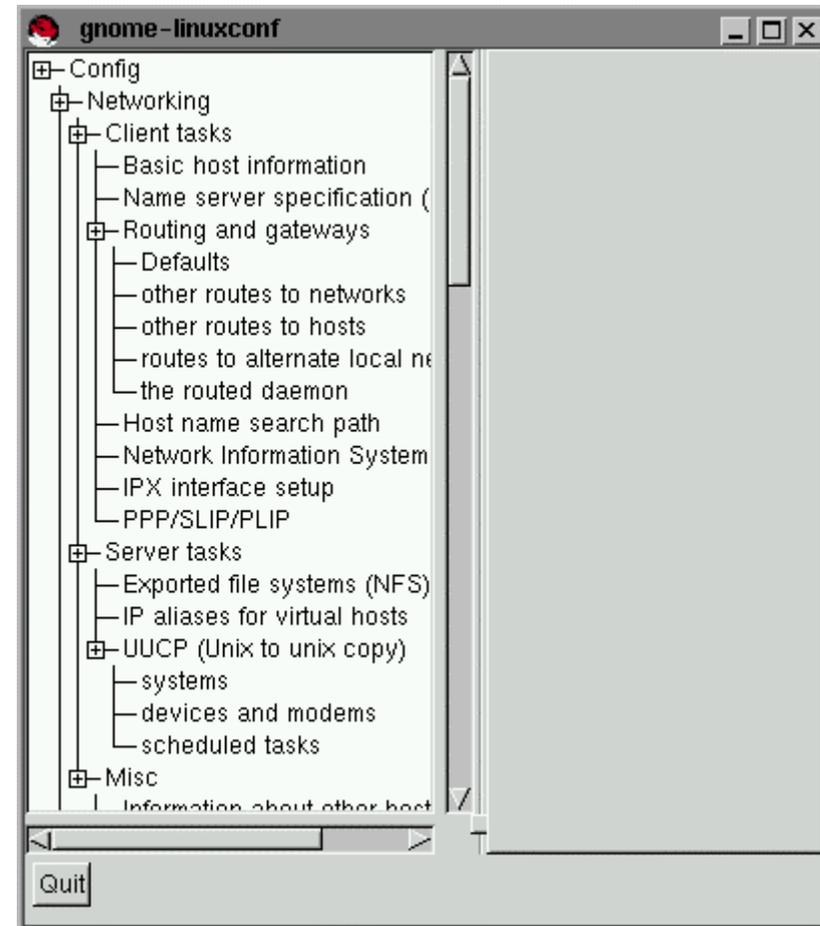
Compare la lettera in caso di discordanza, un puntino in caso di successo del test. La "c", se presente, indica che il file è di configurazione.

Linuxconf

Linuxconf è un sistema di gestione per Linux che fornisce 4 tipi di interfacce:

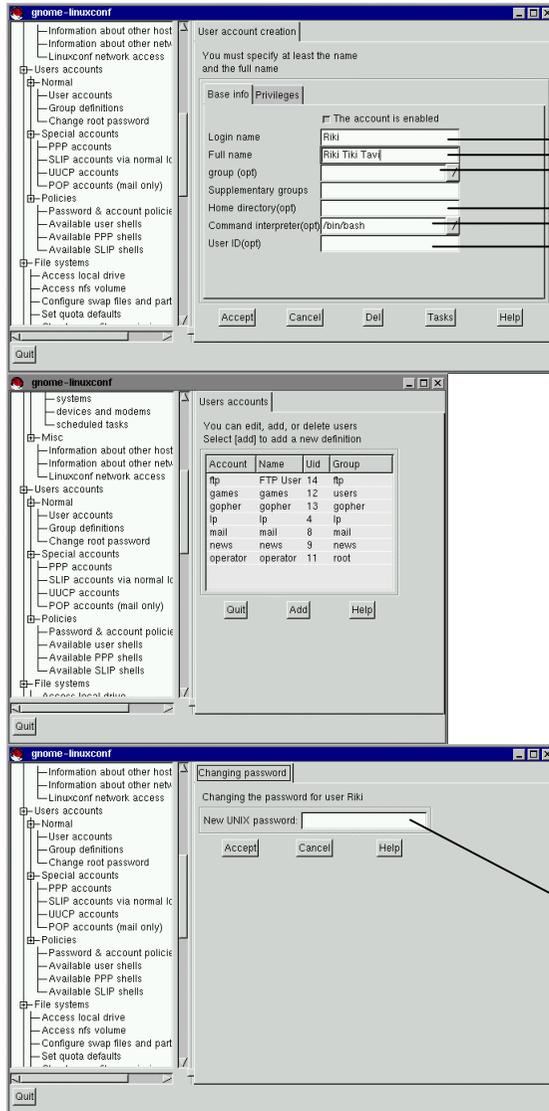
- ◆ a riga di comando (per accedere manualmente ai file e usarlo per il controllo di correttezza)
linuxconf --archive, --diff, --extract, --status, ...
- ◆ a schermate di testo → *linuxconf --text*
- ◆ X Windows → *linuxconf --gui*
- ◆ Web → accesso via <http://<host>:98/>

Nel seguito faremo riferimento all'interfaccia X, cercando di mantenere il parallelo con i file effettivamente manipolati



Gestione utenti

File: `/etc/passwd` + area dati utente

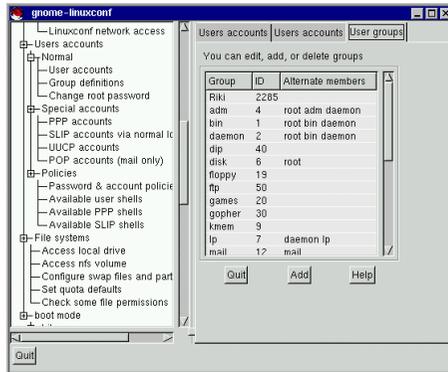


```

root:4KA.itsjWwFZQ:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
mail:*:8:12:mail:/var/spool/mail:
news:*:9:13:news:/var/spool/news:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
nobody:*:99:99:Nobody:/:
    
```

Comandi: `passwd`

Gestione gruppi



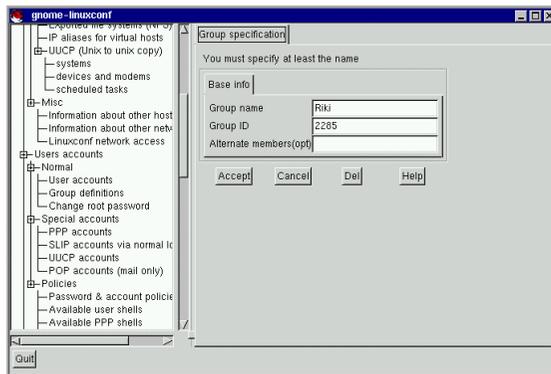
File: `/etc/group`

```

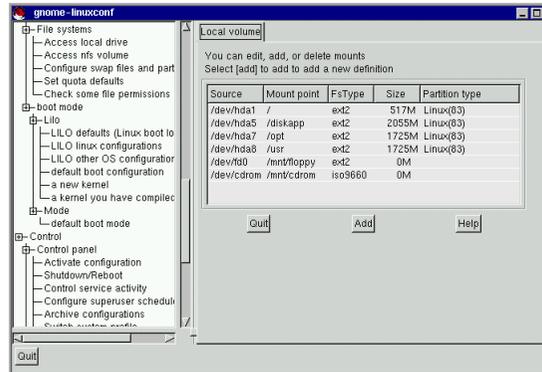
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root
lp::7:daemon,lp
mem::8:
kmem::9:
wheel::10:root
mail::12:mail
news::13:news
man::15:
games::20:
gopher::30:
ftp::50:
nobody::99:
users::100:
floppy:x:19:
pppusers:x:230:
    
```

da [user accounts]
→ supplementary groups

o [group specification]
→ alternate members



Configurazione del filesystem



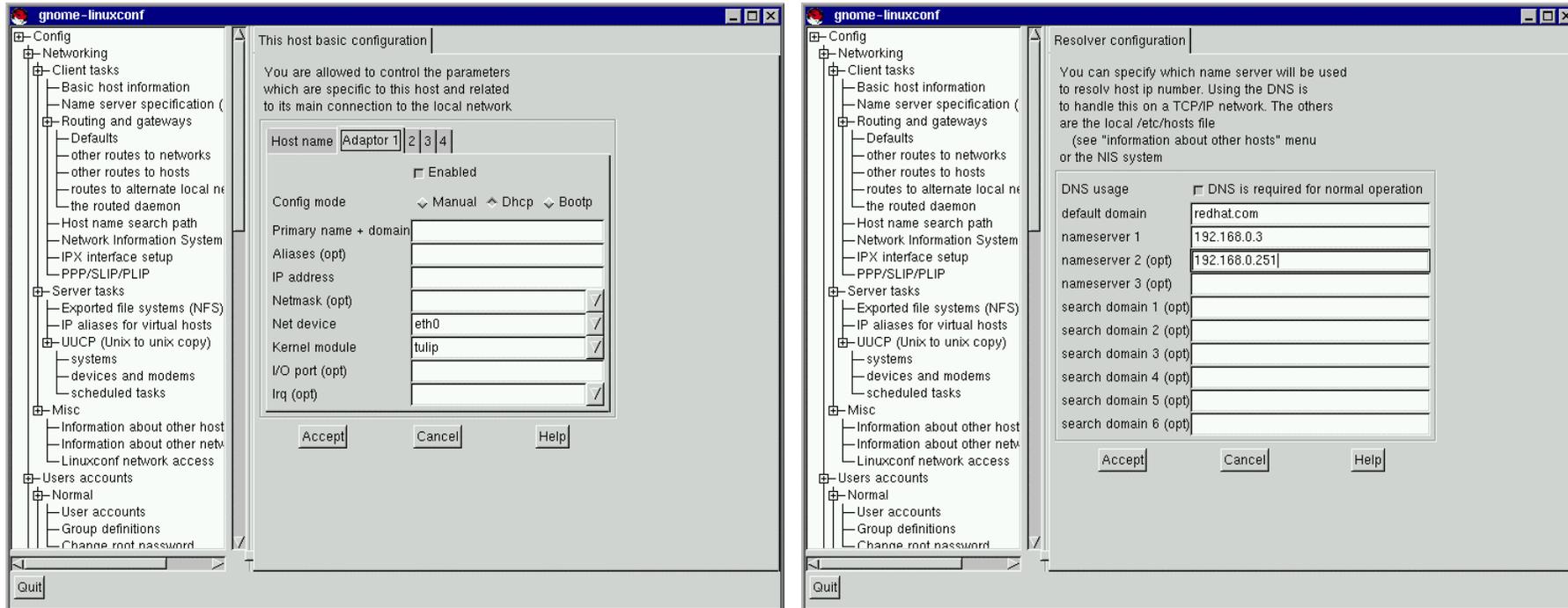
File: */etc/fstab*, */etc/mtab*

<i>/dev/sda2</i>	<i>/</i>	<i>ext2</i>	<i>defaults</i>	<i>1 1</i>
<i>/dev/sda1</i>	<i>swap</i>	<i>swap</i>	<i>defaults</i>	<i>0 0</i>
<i>/dev/fd0</i>	<i>/mnt/floppy</i>	<i>ext2</i>	<i>noauto</i>	<i>0 0</i>
<i>/dev/cdrom</i>	<i>/mnt/cdrom</i>	<i>iso9660</i>	<i>noauto,ro</i>	<i>0 0</i>
<i>none</i>	<i>/proc</i>	<i>proc</i>	<i>defaults</i>	<i>0 0</i>

Comandi: *mount*, *umount*, *fdisk*, *df*

Device	Boot	Start	End	Blocks	Id	System
<i>/dev/sda1</i>		<i>1</i>	<i>17</i>	<i>136521</i>	<i>82</i>	<i>Linux swap</i>
<i>/dev/sda2</i>		<i>18</i>	<i>393</i>	<i>3020220</i>	<i>83</i>	<i>Linux native</i>

Configurazione della rete



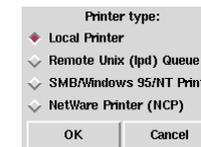
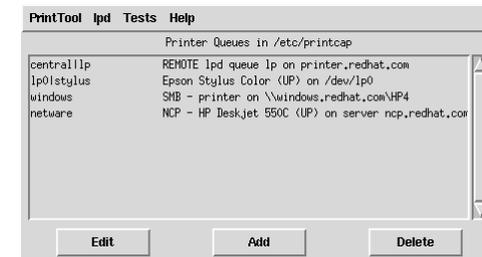
File: */etc/HOSTNAME*
/etc/sysconfig/network
*/etc/sysconfig/network-scripts/**

/etc/resolv.conf
/etc/hosts

Configurazione della stampa

Questa operazione è svolta in modo migliore da una seconda utility di configurazione di Linux, il control-panel. Lo strumento printtool si occupa di compilare con la giusta sintassi (veramente criptica e rigida!) il file /etc/printcap, e di mantenere in modo coerente le directory di spool ed i filtri, ovvero i programmi che sono in grado di distinguere il formato del documento da stampare (GIF, Postscript, testo, ...) e di convertirlo in modo adeguato alla stampante utilizzata.

Con printtool è possibile installare stampanti locali o stampanti accessibili via rete con i protocolli Unix (lpd), Windows (SMB) o Novell Netware (NCP).



Configurazione di una stampante locale

- ◆ possibilità di alias
- ◆ attenzione a non assegnare più stampanti ad una stessa spool directory
- ◆ modalità di connessione

Names (name1|name2|...): lp0|stylus
 Spool Directory: /var/spool/lpd/lp0
 File Limit in Kb (0 = no limit): 0
 Printer Device: /dev/lp0
 Input Filter: Select
 Suppress Headers
 OK Cancel

Printer Type
 Epson AP3250 & ESC/P 2 printers
 Epson Color Dot Matrix, 24 pin
 Epson Color Dot Matrix, 9 pin
 Epson Dot Matrix, 24 pin
 Epson Dot Matrix, 9 pin
 Epson Dot Matrix, 9 pin, hi-res
 Epson Dot Matrix, 9 pin, med-res
 Epson Stylus Color (UP)
 HP DesignJet 650C
 HP DeskJet 400/500C/520/540C
 HP DeskJet 500
 HP DeskJet 550C/560C/6xxC series
 HP DeskJet/DeskJet Plus
 HP DeskJet 550C (UP)
 HP LaserJet
 HP LaserJet 4/5/6 series
 HP LaserJet III* with Delta Row Compression
 HP LaserJet III* with duplex capability
 HP LaserJet IIi/II* with TIFF compression
 HP PaintJet

Driver Description
 Epson Stylus Color uniprint driver settings.

Resolution: NoxNA
Paper Size: Letter, legal, ledger

Color Depth / Uniprint Mode
 stcany, Stylus Color any type, 360x360dpi
 stc, Stylus Color I / PRO Series, 360x360dpi, Plain Paper
 stc.h, Stylus Color I / PRO Series, 720x720dpi, Special Paper
 stc.l, Stylus Color I / PRO Series, 360x360dpi, noleave

Printing Options
 Send EOF after job to eject page?
 Fix stair-stepping text?
 Fast text printing (non-PS printers only)?
 8 4 2 1 pages per output page.

Margins (in pts=1/72 of inch)
 Right/Left: 18 Top/Bottom: 18
 Extra GS options:

OK Cancel HELP

Configurazione di una stampante remota

- ◆ stampante remota via lpd

- ◆ stampante remota su server Novell

- ◆ stampante remota su server Windows

Names (name1 name2 ...)	central1lp
Spool Directory	/var/spool/lpd/centr
File Limit in Kb (0 = no limit)	0
Remote Host	printer.redhat.com
Remote Queue	lp
Input Filter	Select *auto* - PostScript
<input type="checkbox"/> Suppress Headers	
OK Cancel	

Names (name1 name2 ...)	netware
Spool Directory	/var/spool/lpd/netwa
File Limit in Kb (0 = no limit)	0
Printer Server Name	ncp.redhat.com
Print Queue Name	deskjet
User	nwguest
Password	*****
Input Filter	Select
<input checked="" type="checkbox"/> Suppress Headers	
OK Cancel	

Names (name1 name2 ...)	windows
Spool Directory	/var/spool/lpd/windoi
File Limit in Kb (0 = no limit)	0
Hostname of Printer Server	windows.redhat.com
IP number of Server (optional)	
Printer Name	HP4
User	guest
Password	*****
Workgroup	USERS
Input Filter	Select
<input checked="" type="checkbox"/> Suppress Headers	
OK Cancel	

Struttura del filesystem

Red Hat aderisce al Linux File System Standard (FSSTND), un documento collaborativo che definisce i nomi e le collocazioni di molti file e directory, al fine di

- ◆ “orientarsi” in qualunque sistema segua FSSTND
- ◆ poter fare alcune scelte coerenti (es. /usr read-only → protetto, montabile cd CD)

Lo standard lascia ovviamente diverse aree scoperte. Il testo integrale è reperibile sul sito

<http://www.pathname.com/fhs/>

Struttura del filesystem

/dev

La directory /dev contiene i file che rappresentano gli apparati connessi al sistema. Questi file sono essenziali per il funzionamento del sistema stesso.

/etc

La directory /etc è riservata ai file di configurazione locali del sistema. In /etc non devono essere messi eseguibili binari. I binari che in passato erano collocati in /etc devono andare in /sbin o /bin

X11 e **skel** devono essere subdirectories di /etc/

/etc

|- X11

+-- skel

La directory X11 è per i file di configurazione del sistema X Window, come XF86Config. La directory skel è per i prototipi dei file di configurazione delle aree utente.

Struttura del filesystem

/lib

La directory /lib deve contenere solo le librerie richieste per il funzionamento dei programmi che si trovano in /bin e /sbin.

/proc

La directory /proc contiene file speciali che permettono di ottenere informazioni dal kernel o di inviare run-time informazioni al kernel, e merita di essere esplorata con attenzione.

/sbin

La directory /sbin è riservata agli eseguibili utilizzati solo dall'amministratore di sistema, possibilmente solo quelli necessari al boot ed al mount dei filesystem. Qualunque cosa eseguita dopo che /usr sia stato montato correttamente dovrebbe risiedere in /usr/sbin o in /usr/local/sbin

Come minimo devono essere presenti in /sbin i seguenti programmi:

clock, getty, init, update, mkswap, swapon, swapoff, halt, reboot, shutdown, fdisk, fsck. , mkfs.* , lilo, arp, ifconfig, route*

Struttura del filesystem

/usr

La directory /usr è per i file condivisibili in un intero sito. Normalmente risiede su di una propria partizione, e dovrebbe essere montata read-only. Dovrebbe contenere le seguenti subdirs:

/usr

-	X11R6	X Window System
-	bin	eseguibili
-	dict	
-	doc,	documentazione diversa dalle man pages
-	etc	file di configurazione validi per il sito
-	games	
-	include	C header files, , , (those that do not belong in /sbin), and
-	info	GNU info files
-	lib	librerie
-	local	
-	man	man pages
-	sbin	programmi per l'amministrazione di sistema
-	share	
+-	src	codice sorgente

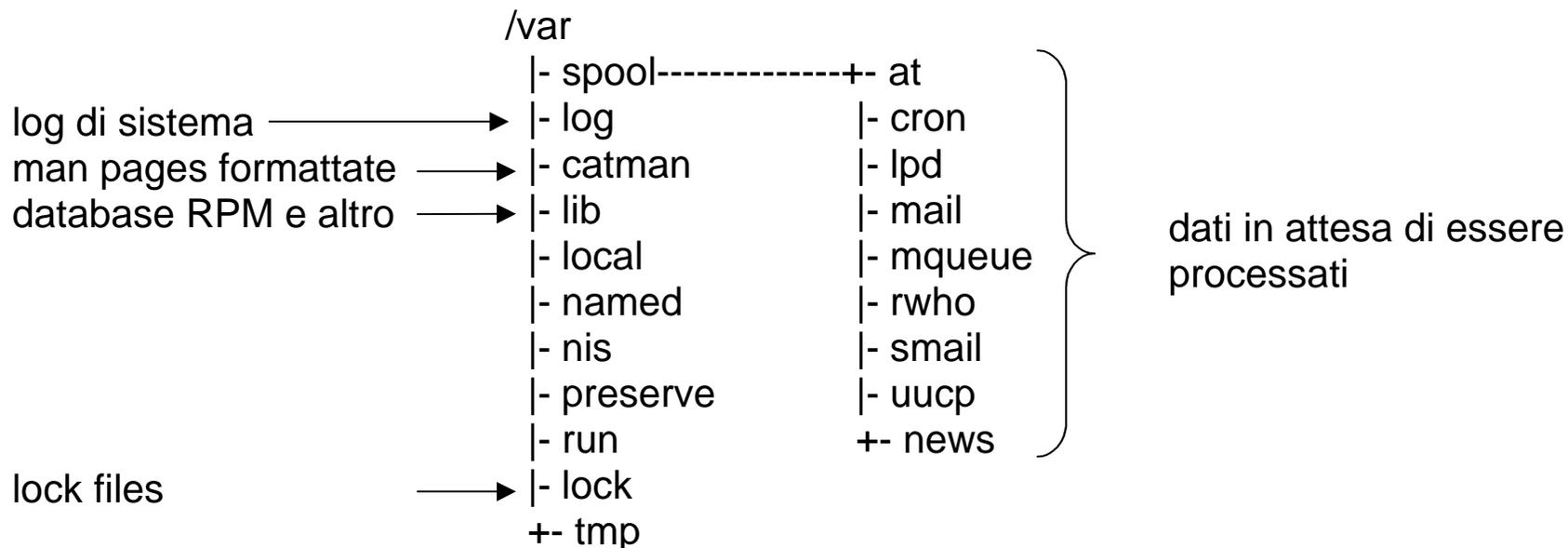
Struttura del filesystem

/usr/local

La gerarchia sotto /usr/local contiene il software installato localmente. Dovrebbe essere salvaguardata in caso di aggiornamenti del software di sistema, ed ha una struttura simile a quella di /usr.

/var

La directory /var è riservata ai file “variabili”, ad esempio i file di log, di spool, di lock, di amministrazione e temporanei. Dovrebbe contenere le seguenti subdirs:



Collocazioni specifiche di Red Hat

- ◆ ***/var/lib/rpm*** informazioni usate dal packet manager
- ◆ ***/usr/lib/rhs*** script, immagini e testi usati da control-panel e strumenti collegati
- ◆ ***/etc/sysconfig*** configurazione del sistema: script di boot, parametri dei moduli e della rete

Personalizzazione del kernel

Il kernel gestisce tutte le risorse hardware, la loro assegnazione ai processi, e la comunicazione tra i processi stessi. Una caratteristica di fondamentale utilità è la modularità:

kernel monolitico (contiene il supporto per tutte le funzionalità necessarie)

- dimensioni critiche
- scarsa portabilità
- teoricamente più veloce e semplice

kernel modulare (può caricare a richiesta un *modulo* deputato alla gestione di una risorsa)

- dimensioni ridotte
- facilità di adeguamento all'hardware
- miglior sfruttamento delle risorse

Attenzione! Fare caricare dei moduli essenziali al boot può essere complicato, quindi è bene costruire kernel “misti” con le parti fondamentali monolitiche e il resto modulare (es. SCSI).

Personalizzazione del kernel

1. sorgenti

- ◆ contenuti in ***/usr/src/linux***
- ◆ fare pulizia con *make mrproper*

2. configurazione

- ◆ *make config* (evitare!) – presenta tutte le opzioni in sequenza
- ◆ *make menuconfig* – interfaccia grafica basata su terminale
- ◆ *make xconfig* – interfaccia grafica sotto X

In qualunque caso, per ogni componente si può scegliere se includerlo (Y), escluderlo (N), o se è possibile includerlo come modulo (M).

3. compilazione

- ◆ *make dep* (costruisce le dipendenze per la compilazione)
- ◆ *make boot* (o *bzImage*, costruisce il kernel)
- ◆ *make modules* (costruisce i moduli)
- ◆ *mv -f /lib/modules/<versione> /lib/modules/<versione>.old* (backup dei vecchi moduli)
- ◆ *make modules_install* (installa i nuovi moduli)

Multiboot con LILO

/etc/lilo.conf

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=100
image=/boot/bzImage ← nuovo kernel
    label=linux          (scelta di default)
    root=/dev/hda3
    read-only
image=/boot/vmlinuz ← vecchio kernel
    label=old
    root=/dev/hda3
    read-only
other=/dev/hda1 ← altro S.O.
    label=dos
    table=/dev/hda
```

→ */sbin/lilo*

Utilizzo dei moduli

I moduli possono essere caricati e rimossi manualmente o automaticamente.

Operazioni manuali:

- ◆ con i comandi *insmod* <modulo>, *lsmod*, *rmmmod* <modulo> rispettivamente si carica un modulo, si elencano i moduli presenti in memoria, e si rimuove un modulo.
- ◆ Se sono necessari parametri (irq, i/o, ...) bisogna specificarli manualmente a *insmod* dopo il nome del modulo, es: *insmod 3c589_cs irq=3*

Operazioni semi-automatiche

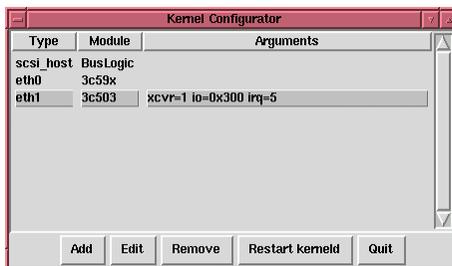
- ◆ con *depmod -a <versione>* viene creato un file */lib/modules/<versione>/modules.dep* che contiene le informazioni di mutua dipendenza tra i moduli
- ◆ con *modprobe <modulo>* è possibile richiedere il caricamento di un modulo e di tutti quelli a lui necessari, o anche di tentare il caricamento di una lista di moduli alla ricerca di quello giusto (*modprobe -l pattern* o *modprobe -t tag*)
- ◆ il comportamento di questi comandi è configurabile con il file */etc/conf.modules*

Il kernel daemon

- ◆ automatizza completamente la gestione dei moduli, caricando quelli necessari e scaricando quelli non utilizzati da un certo tempo
- ◆ è configurato da */etc/conf.modules*

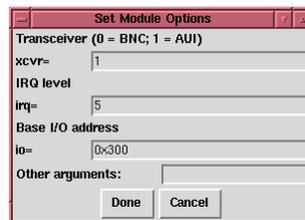
Utilizzo dei moduli

Il comportamento di kerneld (ovvero il file /etc/conf.modules) può essere controllato dal control-panel:



/etc/conf.modules

```
alias scsi_host BusLogic
alias eth0 3c59x
alias eth1 3c503
options eth1 xcvr=1 io=0x300 irq=5
```



La procedura di avvio e arresto del sistema

La procedura di avvio di Linux segue lo standard Unix System V. Il kernel, appena terminato il caricamento e il riconoscimento dell'hardware, lancia il programma *init*, che si occupa di lanciare tutti i processi che è necessario far partire al boot. Init si occupa anche delle transizioni tra i diversi *runlevel*:

0	Halt
1	Single user mode
2	Multiuser mode, senza NFS
3	Multiuser mode, completo
4	Non usato
5	Multiuser mode completo con login sotto X
6	Reboot

Nota: se il PC subisce un danno al file di configurazione di init o al file delle password che impedisce il boot standard, battendo **1** al prompt di LILO si entrerà in single user mode.

La procedura di avvio e arresto del sistema

Tutti i file di configurazione che permettono di controllare il funzionamento di *init* risiedono nella directory ***/etc/rc.d***. In particolare vi si trovano:

- ◆ il file *rc.sysinit*
- ◆ le directory *init.d*, *rc0.d*, *rc1.d*, *rc2.d*, *rc3.d*, *rc4.d*, *rc5.d*, *rc6.d* [,*rc.local*, *rc.serial*]

init.d contiene uno script per il lancio e l'arresto di ciascuno dei servizi gestiti da *init*, esclusi quelli che devono essere eseguiti una volta sola indipendentemente dal runlevel. Questi script devono essere messi in *rc.local* o *rc.serial*.

La catena degli eventi si svolge in questo modo:

1. Il kernel cerca in una serie di directory predefinite e lo lancia
2. *init* lancia */etc/rc.d/rc.sysinit*
3. *rc.sysinit* compie una serie di operazioni preliminari e lancia *rc.serial* (se esiste)
4. *init* lancia tutti gli script per il runlevel di default
5. *init* lancia *rc.local*

Il runlevel di default è configurato in ***/etc/inittab***, dove viene definita la configurazione di *init* ed il suo comportamento in seguito ad eventi esterni segnalati dal kernel o da applicazioni.

La procedura di avvio e arresto del sistema

Con che criterio init sceglie gli script adatti ad ogni runlevel? Se ad esempio si esamina il contenuto della directory *rc3.d*, si troverà qualcosa di simile a questo:

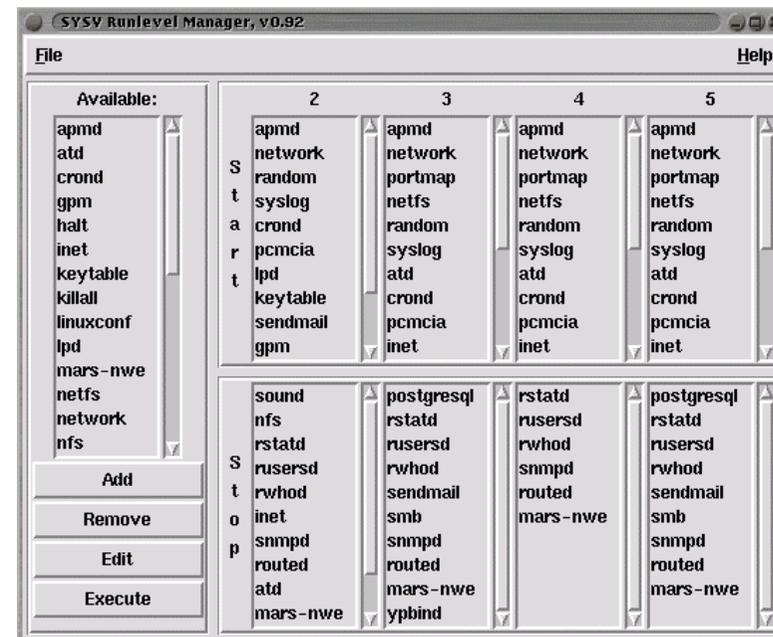
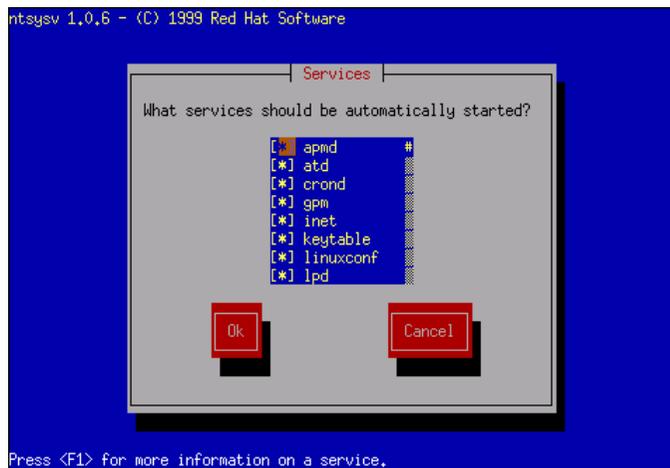
```
lrwxrwxrwx 1 root root 17 3:11 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 16 3:11 S30syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 14 3:32 S40cron -> ../init.d/cron
lrwxrwxrwx 1 root root 14 3:11 S50inet -> ../init.d/inet
lrwxrwxrwx 1 root root 13 3:11 S60nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 15 3:11 S70nfsfs -> ../init.d/nfsfs
lrwxrwxrwx 1 root root 18 3:11 S90lpd -> ../init.d/lpd.init
lrwxrwxrwx 1 root root 11 3:11 S99local -> ../rc.local
```



Non ci sono file “reali”, ma solo link agli script in *init.d*, che sono fatti in modo da accettare un parametro che può essere **start** o **stop**. Di fatto gli script in *init.d* sono il modo migliore per controllare anche a mano i processi. Es: `/etc/rc.d/init.d/httpd start`

La procedura di avvio e arresto del sistema

L'utility **chkconfig** fornisce una semplice interfaccia a linea di comando per mantenere la gerarchia `/etc/rc.d` senza doversi preoccupare di tutti i link. L'utility **ntsysv** mette a disposizione un'interfaccia grafica testuale con la stessa funzionalità. Il runlevel editor del control-panel, infine, mette a disposizione l'interfaccia X.



Per arrestare Linux si utilizza il comando **shutdown**. Le sintassi più comunemente utilizzate sono `shutdown -h now` (reboot immediato) e `shutdown -r now` (reboot immediato).

Esecuzione programmata

Nell'amministrazione di un sistema ci sono due esigenze abbastanza comuni:

- ◆ eseguire un programma con cadenza periodica
- ◆ eseguire un programma occasionalmente ma in un momento ben preciso

Il primo problema è risolto dal ***cron daemon***, il secondo dall'***at daemon***.

Cron

crond è un demone (programma residente in memoria in attesa di eventi che lo riguardino) che esamina una serie di file di configurazione ogni minuto, e determina quali compiti specificati nei file debbano essere eseguiti.

Amministrazione di sistema

/etc/crontab → */etc/cron.hourly/*
/etc/cron.daily/
/etc/cron.weekly/
/etc/cron.monthly/

/etc/cron.d/

Compiti definiti dagli utenti

/var/spool/cron/<utente>

→ *crontab -e*

MI OR GM ME GS <comando>

es.

```
* * 27 * * $HOME/bin/pagatemi
30 8-18/2 * * 1-5 $HOME/bin/lavora
30 4 1,15 * 6 /bin/false
```

At

atd è un demone che gestisce code di compiti da svolgere in momenti prefissati. L'interfaccia ad **atd** consiste di 4 comandi:

at [-V] [-q queue] [-f file] [-mldbv] TIME	← accoda un comando al tempo TIME
atq [-V] [-q queue] [-v]	← elenca i comandi in coda
atrm [-V] job [job...]	← rimuove comandi dalla coda
batch [-V] [-q queue] [-f file] [-mv] [TIME]	← esecuzione condizionata al carico

- ◆ Se non viene specificato un file comandi per *at* o *batch*, verrà usato lo standard input.
- ◆ La specifica dell'ora è flessibile e complessa. Per una definizione completa si veda la documentazione in */usr/doc/at-<versione>/timespec*. Alcuni esempi:

```
echo 'wall "sveglia"' | at 08:00
echo "$HOME/bin/pulisci" | at now + 2 weeks
echo "$HOME/bin/auguri" | at midnight 25.12.99
```

Diagnostica di sistema

La maggior parte dei servizi gira in *background*, ovvero senza essere collegata ad un terminale da cui ricevere input o su cui scrivere output. Per evitare di doversi “inventare” per ogni servizio un gestore dell’output diagnostico e di registrazione delle azioni importanti, e per avere un metodo uniforme di classificazione dei messaggi prodotti dal sistema e dagli applicativi, si utilizza **syslogd**.

/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                                 *
*.emerg                                 @loghost
```

Diagnostica di sistema

Qualunque programma può sfruttare *syslogd*, per mezzo di una chiamata di sistema in C alla funzione ***syslog*** o per mezzo dell'utility ***logger***.

logger -p <priorità> messaggio

È possibile definire facility personalizzate nel file */etc/syslog.conf*, chiamate ***local0...local7***

Nota importante: qualunque metodo usi un'applicazione per produrre dati diagnostici e di controllo, bisogna prevedere un metodo **automatico** per tenere sotto controllo la dimensione dei file che ne derivano.