

Da Windows a Linux – (C) 1999–2003 Paolo Attivissimo e Roberto Odoardi.  
Questo documento è liberamente distribuibile purché intatto.

## 20. Linux mette il turbo

Non è indispensabile che completiate tutte le attività che descrivo in questo capitolo: anzi, a dire il vero potete lavorare benissimo con Linux anche se non fate niente di quello che sto per raccontarvi. Tuttavia se eseguite queste operazioni migliorerete il funzionamento di Linux sul vostro computer.

Cosa forse più importante, la disinvoltura (o lo smarrimento più abietto) con il quale affronterete questo capitolo vi darà una misura precisa del livello di dimestichezza e competenza che avete raggiunto e dei progressi che avete fatto dall'inizio della vostra avventura linuxiana.

Se non superate questo capitolo, non demoralizzatevi. Io ci ho sbattuto la testa prima di voi, e sono sopravvissuto. Vivrete lo stesso.

Il riordino che vi propongo agisce su tre livelli fondamentali:

- **Libera un bel po' di memoria RAM.** La memoria non basta mai, ed è certamente inutile affollarla con cose che non vengono utilizzate.
- **Recupera spazio sul disco rigido.** Linux contiene un sacco di programmi e librerie che sicuramente non utilizzerete mai. Tanto vale eliminarli.
- **Toglie carico di lavoro al processore.** Se il processore ha meno programmi a cui badare, fa girare più velocemente quei pochi che restano.

Dato che si tratta di operazioni di manutenzione, dovete eseguirle tutte accedendo a Linux con l'utente *root*. Di conseguenza, fate molta attenzione a quello che combinate: siete temporaneamente onnipotenti, e uno sbaglio potrebbe rendere inservibile Linux. Lo dico qui all'inizio, ma vale per tutto il capitolo:

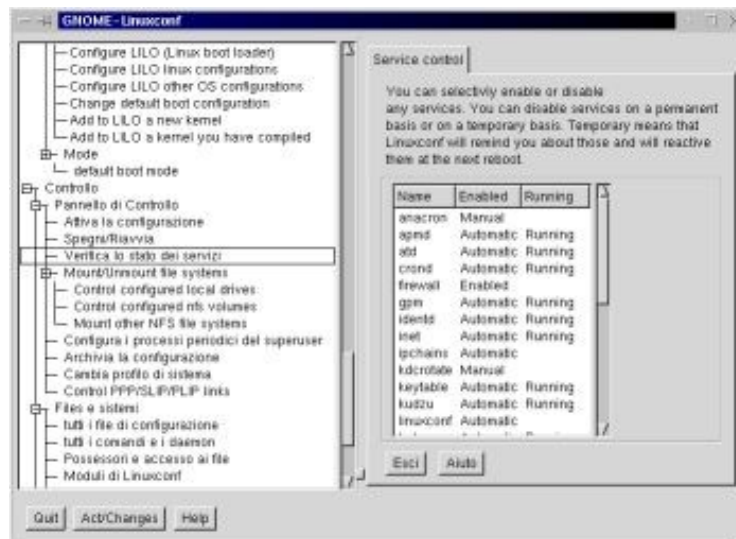
- **Fate un backup** prima di procedere.
- **Prendete nota** (su carta!) di tutto quello che fate, così saprete dove avete sbagliato.

### Pulire i servizi

Non temete, non occorre che tiriate fuori la spugna e il WC Net. I "servizi" ai quali mi riferisco sono i programmi che Linux lancia automaticamente quando lo avviate. Molti di questi servizi sono orientati all'uso di Linux come sistema operativo per macchine collegate in rete e sono perciò inutili per chi vuole usare il computer soltanto per applicazioni personali, ad esempio per scrivere documenti o lavorare con StarOffice e simili.

Odo puntualizza che il termine *servizi* viene solitamente utilizzato con Windows NT, mentre il termine corretto per UNIX è *daemon*, di cui spesso incontrerete l'equivalente italiano *demone*. Si tratta di programmi che svolgono il loro lavoro di supporto silenziosamente e di nascosto. Potete riconoscerli dal fatto che quasi sempre il loro nome termina con la lettera *d* (ad esempio *httpd*, *ftpd*, *lpd*, *inetd*).

Lanciate `linuxconf`. Nella sezione *Controllo*, scegliete *Pannello di controllo* e trovate la voce *Verifica lo stato dei servizi*.



**Figura 20–1. Verificare lo stato dei servizi con linuxconf.**

Disattivare e attivare un servizio è semplice: lo si seleziona dalla lista, in modo da far comparire la sua scheda. Una volta visualizzata la scheda, e se siete ben consapevoli di quello che state facendo:

- per disattivare il servizio, fermatelo immediatamente cliccando sul pulsante *Stop* e disattivatelo permanentemente cliccando sulla casella *Automatic*. In questo modo il servizio non ripartirà più automaticamente quando riavviate Linux;
- per attivare un servizio inattivo, cliccate sul pulsante *Start* per attivarlo immediatamente e cliccate sulla casella *Automatic* per farlo ripartire automaticamente ad ogni avvio di Linux.

Cliccate poi sul pulsante *Accetta* per tornare alla schermata che elenca lo stato dei servizi. Se ci sono altre modifiche da fare, ripetete i passi che ho appena descritto; altrimenti uscite da linuxconf confermando le modifiche che avete immesso.

Piuttosto facile, non vi pare? La vera difficoltà consiste nel sapere cosa disattivare e cosa no. Per aiutarvi nella scelta ho preparato una breve descrizione di ciascun servizio, corredandola di un consiglio non vincolante sulla sua attivazione o disattivazione. Il consiglio presuppone che vogliate usare Linux per fare cose simili a quelle che facevate con Windows e non per cose tipicamente linuxiane come gestire una macchina *firewall* o un server di rete.

Servizio	A cosa serve	Condizione consigliata
anacron	Esegue operazioni di manutenzione a intervalli/orari prefissati dall'utente.	Attivato.
apmd	Spegnimento automatico, monitoraggio batterie e/o ibernazione del computer quando chiudete Linux. Richiede un BIOS compatibile.	Attivato: il computer si comporterà come con Windows, spegnendosi da solo quando chiudete il sistema operativo. Se lo attivate, ricordatevi di impostare correttamente il BIOS.
atd	Esegue operazioni di manutenzione a intervalli/orari prefissati dall'utente.	Attivato.
crond	Esegue operazioni di manutenzione a intervalli/orari prefissati dall'utente.	Attivato.

firewall	Blocca gli accessi non autorizzati al computer effettuati via Internet e reti locali.	Attivato.
gpm	Gestione del mouse nelle console.	Attivato, a meno che non usiate il mouse.
identd	Verifica dell'identità degli utenti tramite TCP/IP.	Attivato.
inetd (6.2) xinetd (7.0)	Il super-demone di Linux che fa partire gli altri demoni che gestiscono servizi di rete opzionali come telnet e ftp.	Attivato.
ipchains	Sorveglianza degli accessi da Internet al vostro computer e viceversa.	Attivato sempre e comunque; è un elemento fondamentale della sicurezza.
irda	Gestione comunicazione a infrarossi.	Disattivato, a meno che abbiate una porta a infrarossi (improbabile su un computer fisso, probabile su un portatile).
kdcrotate	Gestione della sicurezza tramite Kerberos.	Disattivato, se non avete un'ottima ragione per usare Kerberos (ad esempio perché sapete che cos'è).
keytable	Impostazione della tastiera in modo che adotti la disposizione dei tasti nazionale.	Attivato sempre e comunque.
kudzu	Rilevamento automatico del nuovo hardware.	Attivato sempre e comunque, a meno che non installiate mai nuovo hardware (improbabile). Il vero linuxiano, però, lo disattiva perché, per usare la metafora di Odo, vuole tenere il cacciavite dalla parte del manico (cioè decidere lui come e quando far vedere l'hardware al sistema operativo).
linuxconf	Il programma che state usando.	Attivato in ogni caso.
lpd	Gestione delle stampanti.	Attivatelo se avete collegata una stampante.
netfs	Gestione dei file system accessibili tramite rete: ad esempio Samba e NFS.	Attivato soltanto se il vostro computer è collegato in rete con altri computer con i quali volete condividere file.
network	Controllo delle interfacce di rete (schede di rete e simili).	Attivato anche se non avete una scheda di rete.

nfs	Controllo della condivisione di file in rete.	Attivato soltanto se sapete cos'è NFS e se lo usate.
nfslock	Controllo della condivisione di file in rete.	Come sopra.
pcmcia	Gestione delle schede PC Card o PCMCIA.	Disattivato, se non avete schede di questo tipo.
portmap	Gestione avanzata della condivisione di file tramite NFS o NIS.	Disattivato se non usate NFS o NIS per la condivisione di file. Sicuramente disattivato se il vostro computer non è collegato in rete.
random	Gestione evoluta della generazione di numeri casuali.	Attivato, anche non non avete idea di quale sia lo scopo di generare numeri casuali (indizio: sicurezza).
rwhod	Elencazione dall'esterno degli utenti che stanno accedendo al computer.	Disattivato.
sendmail	Gestione avanzata dell'e-mail da ritrasmettere ad altri computer. Non serve per il normale scambio di e-mail.	Disattivato.
smb	Condivisione di risorse (file e stampanti) in una rete Windows.	Attivato soltanto se usate Samba per condividere file con macchine Windows.
sshd (non presente nell'installazione standard)	Connessione protetta e cifrata di utenti esterni al vostro computer.	Attivato. Questo servizio non è incluso nella normale installazione; compare nella lista soltanto se avete installato la manutenzione a distanza ( <i>ssh</i> ) descritta nel Capitolo 19.
syslog	Archiviazione dei messaggi di sistema.	Attivato.
xf86-config	Gestione dei font dell'interfaccia grafica.	Attivato, a meno che non usiate mai l'interfaccia grafica.
yppbind	Connessione di rete tramite NIS.	Disattivato.

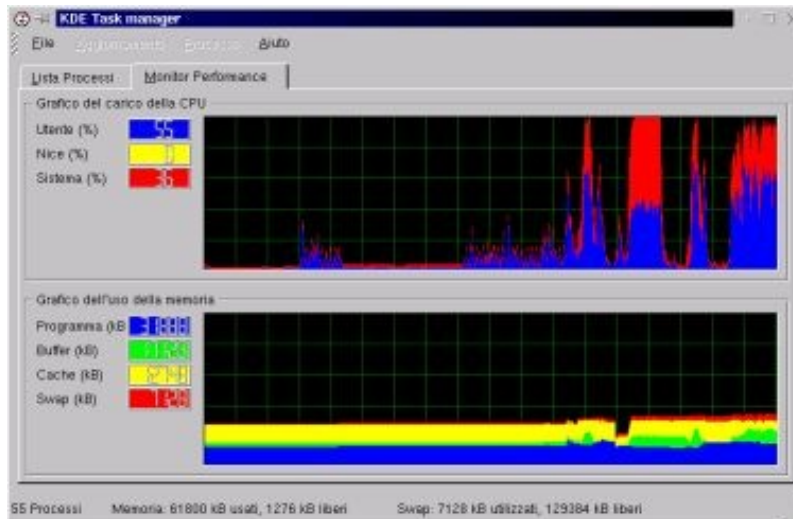
Quelli elencati in tabella sono i servizi installati sia da Red Hat 7.0, sia da Red Hat 6.2. La distribuzione 7.0, però, ne installa anche degli altri che riassumo qui:

- **arpwatch**: verifica incrociata degli indirizzi di rete (MAC e IP). Disattivatelo.
- **isdn**: se non avete una linea ISDN, disattivatelo.
- **linuxconf-web**: consente di comandare linuxconf tramite un browser da remoto. Disattivatelo.
- **nfslock**: utile soltanto se avete a che fare con filesystem NFS. Disattivatelo.
- **pppoe**: realizza una rete privata virtuale su Ethernet. Disattivatelo.

- **rhnsd**: aggiornamento automatico del software (simile a Windows Update). A Odo fa venire i brividi da Grande Fratello (il personaggio di Orwell, non la patetica trasmissione televisiva). Disattivatelo.

## Misurare i risultati

Il programma *ktop*, accessibile tramite il Pannello in Sistema > Task Manager, vi offre un modo per verificare il risultato delle vostre modifiche. La scheda *Monitor performance* indica quanto carico di lavoro affidate al processore e quanta memoria viene utilizzata.



**Figura 20–2. Uso del processore e della memoria: ktop.**

L'ideale è che a riposo, cioè quando Linux è attivo ma non state adoperando il computer, il carico dal processore sia a livelli minimi e il consumo di memoria sia stabile (preferibilmente a un livello modesto). Se non è così, vuol dire che ci sono servizi troppo pesanti che divorano risorse, ed è il caso di guardare cosa si può disattivare.

Odo adora il suo screensaver di *The Matrix*, ma questo non gli impedisce di ricordarmi che ogni fronzolo ha il suo prezzo in termini di consumo di risorse del computer, e lo screensaver è un fronzolo. Se volete velocizzare il vostro computer, mettete in cima alla lista delle cose da fare la disattivazione degli screensaver.

La scheda *Lista processi* di *ktop* elenca tutti i processi attivi (se selezionate la voce di menu *Tutti i processi*). Non sorprendetevi se molti dei loro nomi non vi sono familiari. Cliccando sull'intestazione della colonna *CPU* o della colonna *Memoria*, la lista dei processi viene riordinata in base al carico che ciascun processo impone al processore o, rispettivamente, in base alla quantità di memoria che utilizza, così potete identificare con facilità i processi troppo avidi.

Attivando la casella *Mostra albero*, tra l'altro, i processi vengono visualizzati in ordine gerarchico. Infatti capita spesso in Linux che un processo lanci un altro processo che a sua volta ne lancia un terzo, e così via. Quest'informazione è utile quando c'è qualche processo che fa le bizze e dovete scoprire chi (o cosa) l'ha attivato.

## Mettiamo Linux a dieta

Con poche eccezioni, tipo tenori stramiliardari residenti a Montecarlo e lottatori di sumo, la ciccia non fa bene a nessuno: appesantisce e rallenta. Lo stesso vale per Linux; e di ciccia, nell'installazione che vi ho proposto nel Capitolo 10, ce n'è tanta.

Ci sono infatti due approcci fondamentali all'installazione di Linux: scegliere soltanto i pezzi che servono, oppure installare tutto e poi sfrondare. Il primo metodo produce sicuramente un Linux ultrasnello, ma richiede conoscenze che non avevate dieci capitoli fa. Il secondo, benché occupi molto spazio su disco, ha il vantaggio di darvi subito un Linux funzionante (magari un po' appesantito, ma funzionante), ed è per questo che l'ho scelto per questo libro.

Lo svantaggio è che ora sul vostro disco rigido c'è un enorme bouquet di programmi, la maggior parte dei quali serve a cose che non c'entrano nulla con la vostra attività e sta consumando inutilmente spazio su disco. Se volete sbarazzarvene, seguitemi.

### Approcci alternativi

Adesso che siete un po' più pratici di Linux, potreste anche esplorare un'altra strada: reinstallare Linux scegliendo l'opzione di personalizzazione. I risultati di una personalizzazione ben fatta sono strepitosi, ma definire con precisione (e soprattutto senza sbagliare) tutte le scelte necessarie richiede ore di lavoro e di studio. In più, perdereste tutto il lavoro di adattamento fatto fin qui. A voi la scelta.

Se volete misurare quanto spazio liberate al termine della pulizia, prima di cominciare aprite una finestra di terminale e digitate **df -m**. Ottenete l'elencazione, in megabyte, dello spazio libero su tutte le unità disponibili a Linux. Ripetete l'operazione dopo che avete terminato la pulizia e fate i debiti conticini. A me, ad esempio, la pulizia di base (grosso modo quella che vedete elencata nelle prossime pagine) ha liberato 220 megabyte di spazio, riducendo il totale a 470 megabyte occupati. E si può fare di meglio.

Con i dischi rigidi immensi che ci sono in giro oggi, potreste pensare che non ci sia troppo da esaltarsi per aver liberato "soltanto" duecento megabyte: dopotutto, che differenza c'è fra avere liberi 19,5 gigabyte e averne liberi 19,3? Non posso darvi torto, ma riflettete un momento. Non è soltanto una questione di spazio su disco: dovete tenere conto dello spazio occupato dai backup.

L'installazione standard di Linux, infatti, ammonta a circa 700 megabyte: troppo per copiare tutto a un CD-ROM da usare come backup. Togliendo 220 megabyte o giù di lì, lo spazio occupato scende al di sotto della soglia di capienza di un CD-ROM (circa 640 megabyte) e potete salvare tutto il vostro Linux su un solo CD. Grazie alla dieta, il backup di Linux ingombra (e quindi costa) la metà.

Per rimuovere i pezzi inutili dell'installazione di Red Hat Linux, che tecnicamente si chiamano *pacchetti*, lanciate *kpackage* (dal Pannello, scegliete *Accessori > kpackage*), che elenca tutti i programmi e tutte le librerie installate. Ricordate che dovete essere *root* quando lanciate *kpackage*, altrimenti vi verrà chiesta la sua password. Dall'elenco offerto da *kpackage*, selezionate l'elemento che volete disinstallare e cliccate su *Uninstall*.

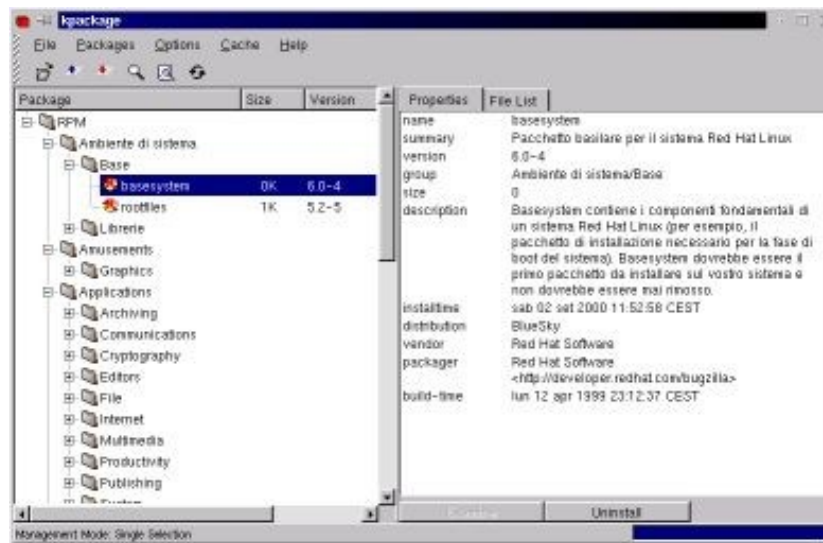


Figura 20–3. La finestra di kpackage.

Come prima, il problema è sapere cosa rimuovere e cosa no. La regola generale da seguire è questa: se non sapete a cosa serve un pacchetto di Linux, non toccatelo. Rimuovetelo soltanto se siete ben sicuri che non vi serva. Il modo migliore per saperlo è leggere la descrizione dell'elemento visualizzata nella finestra di kpackage.

L'unico, piccolo problema di questo approccio è che la descrizione è quasi sempre in inglese, e per giunta anche in gergo tecnico. Siccome non tutti masticano pane e inglese dalla mattina alla sera, Odo e io abbiamo preparato una lista di rimozioni consigliate. Ecco come usarla:

- Mi raccomando, non prendete il nostro elenco come oro colato. È soltanto una lista di suggerimenti che, come prima, si basa sul presupposto che Linux venga usato per le normali attività di Windows, quindi non come server di rete o per programmazione in linguaggi esoterici.
- Sicuramente ci sono altri pacchetti sacrificabili che non abbiamo citato qui. Nello stilare l'elenco abbiamo preferito eccedere in prudenza piuttosto che dirvi di gettar via qualcosa che potrebbe servirvi.
- Per non generare una lista chilometrica, abbiamo omesso i pacchetti che sono sicuramente da mantenere e abbiamo elencato soltanto i casi dubbi, cioè quelli in cui la rimozione è possibile ma non sempre consigliabile.
- Facendo scorrere la lista di kpackage vi imbatteverete sicuramente in molti programmi che non sapevate neppure di avere sul computer. Se volete provarli prima di decidere di rimuoverli o meno, fate pure: lanciateli digitando **Alt-F2** per far comparire la "mini-console" in cui digitare il nome del programma.
- La lista delle rimozioni consigliate risparmia molti programmi, come *vi*, che non servono a voi ma probabilmente serviranno a chi deve aiutarvi nella manutenzione della vostra installazione di Linux.
- Se vi pentite di una rimozione, potete sempre reinstallare: trovate tutto nel CD di installazione, nella directory *RPMS*.
- a di quello che rimuovete dal computer. Se qualcosa non funziona più, basterà consultare gli appunti per ripristinare la situazione originale.
- L'elenco presenta i pacchetti nello stesso ordine in cui li presenta kpackage, ripartiti in cartelle tematiche.
- Alcuni pacchetti dipendono da altri, nel senso che non potete disinstallarli se prima non disinstallate il pacchetto da cui dipendono. Se compare il messaggio *Dependency problem*, vuol dire che siete incappati in una dipendenza di questo tipo. Per risolverla, prendete nota del nome del pacchetto citato nel messaggio dopo le parole *is needed by* e disinstallate prima quel pacchetto. Poi ritentate.
- Prima di disinstallare un pacchetto, fate una disinstallazione simulata attivando la casella *Test (do not uninstall)* nella finestra di dialogo che compare dopo che avete cliccato su *Uninstall* nella schermata principale di kpackage. Questo vi permetterà di verificare eventuali dipendenze e di scoprire se il pacchetto che volete disinstallare è

inaspettatamente richiesto da un altro pacchetto che invece volete mantenere nel computer.

- Se volete disinstallare nonostante siano segnalate delle dipendenze, disattivate la casella *Check dependencies*.
- Spesso la disinstallazione segnala *cannot remove* seguito da un nome di directory. Questo significa che la disinstallazione non è riuscita a cancellare la directory indicata perché contiene ancora dei file appartenenti a qualche altra applicazione. Va bene così: accettate cliccando su *OK* e proseguite tranquilli.

## Pacchetti probabilmente disinstallabili

### Ambiente di sistema

- **Librerie:** *libc* (salvo che usate vecchie applicazioni Linux).

### Amusements

- **Games:** *fortune-mod* (viene usato dal programma di gestione dello screensaver, ma se non adoperate lo screensaver potete eliminare questo pacchetto).
- **Graphics:** *xmorphism*, *xscreensaver* (se non usate gli screensaver).

### Applications

- **Communications:** *kpilot* e *pilot-link* (sicuramente da eliminare, salvo abbiate un Palm Pilot), *uucp* (sicuramente da eliminare), *minicom*.
- **Editors:** *emacs*, *emacs-X11*, *emacs-nox*, *psmg*.
- **Engineering:** *bc*.
- **Internet:** *elm*, *exmh*, *nmh*, *pine*, *slrn*, *rsync*, *tin*, *trn*, *xmailbox*, *xrn*.
- **Multimedia:** scegliete uno dei vari programmi per l'audio e la grafica e rimuovete i rimanenti: difficilmente ve ne servirà più di uno.
- **Productivity:** qui va a gusti, se non usate i programmi-organizer potete buttare via tutto.
- **Publishing:** se non usate TeX, sicuramente potete buttare *jadetex*, tutti i pacchetti che iniziano con *tetex*, nonché *texinfo* e *sgml-tools*. Questa rimozione, da sola, libera una cinquantina di megabyte.
- **System:** *arpwatch*, *autorun*, *irda-utils* (se non avete porte a infrarossi), *mt-st*, *samba-client* e *samba-common* (se non siete collegati in rete con macchine Windows), *timetool*.
- **Text:** *docbook*, *stylesheets*, *ed*, *jade* (ben 45 megabyte), *sgml-common*, *indent*.

### Development

- **Debuggers:** *gdb*, *xxgdb*, *ltrace*, *memprof*, *strace*.
- **Libraries:** *pilot-link-devel* (a meno che abbiate un Palm Pilot), *linuxconf-devel*.
- **Tools:** *ElectricFence*, *bison*, *byacc*, *cvs*, *flex*, *gettext*, *rcs*.



## User Interface

- **Desktops:** *fvwm2, fvwm2-icons.*

### Anema e core

Ci sono altri due trucchetti per liberare spazio su disco. Il primo consiste nell'eliminare i *core dump*, ossia le immagini della memoria che Linux salva su disco quando un'applicazione va in tilt. A meno che la vostra passione sia analizzare i motivi dei *crash* delle applicazioni, questi file di core dump sono del tutto inutili.

Solitamente i file di core dump si chiamano *core* (abbastanza ovviamente) e risiedono nelle home directory degli utenti che hanno lanciato le applicazioni andate in *crash*. Il metodo migliore per localizzarli tutti è essere *root* e usare *kfind* per trovare qualsiasi file di nome *core* ovunque sotto la root directory. Non cancellate eventuali directory di nome *core* (ad esempio */proc/sys/net/core/*).

In alternativa, Odo coglie l'occasione di mostrare la potenza della riga di comando di Linux:

```
find / -name core -type f -print
```

elenca tutti i file di core dump, ma si può fare anche di meglio:

```
rm -f `find / -name core -type f -print`
```

li scova e cancella direttamente, ma fate *molta* attenzione a quello che fate, prima di cancellare cose di cui poi vi pentirete.

Il secondo trucchetto è cancellare i vecchi file di log, che risiedono in */var/log*. Quelli vecchi si riconoscono dalla data del loro ultimo aggiornamento e dal fatto che sono numerati.

## Ricompilare il kernel

Questa, fra tutte le operazioni che potete effettuare con Linux, è senz'altro la più mistica. Per gli utenti Linux, ricompilare il kernel non è una semplice operazione di manutenzione, come cambiare l'olio all'auto o spolverare; è un gesto altamente spirituale, una sorta di rito di purificazione dell'anima. Abbiate rispetto per il luogo sacro nel quale state per entrare.

Misticismo a parte, ci sono delle ottime ragioni per ricompilare il kernel. La prima, ovviamente, è dimostrare che siete veri linuxiani (un vero utente Linux non si accontenta del pacchetto preconfezionato; ci *deve* smanettare). Poi ci sono delle motivazioni pratiche:

- **Snellire.** Il kernel standard fornito con i CD di installazione è un modello universale, tipo temperino svizzero, adatto un po' per tutti i computer e tutti i componenti hardware, ma contenente una sovrabbondanza di accessori inutili per il vostro caso specifico. Ricompilandolo, si eliminano i driver relativi a dispositivi che non sono presenti nel vostro computer e quindi il kernel occupa meno spazio in memoria. Lo spazio liberato diventa disponibile per eseguire altri programmi.
- **Ottimizzare.** Il kernel può essere impostato in modo da sfruttare le caratteristiche specifiche del vostro particolare modello di processore, consentendo elaborazioni più rapide ed efficienti.
- **Aggiornare.** Periodicamente il kernel di Linux viene aggiornato per perfezionarlo o consentirgli di gestire nuovi componenti hardware. La ricompilazione serve per includere questi aggiornamenti nella vostra installazione senza dover reinstallare tutto il sistema operativo. Gli aggiornamenti di Linux, ovviamente, sono gratuiti; in Windows, invece, in genere è necessario comperare la nuova versione dell'intero sistema operativo o installare delle "pezze" (*patch*), e in caso di reinstallazione occorre reinstallare anche tutte le pezze.

Quello descritto qui non è senz'altro l'unico modo per compilare un kernel di Linux. Come capita spesso in questo sistema operativo, ci sono tantissimi modi diversi per ottenere lo stesso risultato. Non stupitevi, quindi, se trovate altrove una descrizione molto diversa della procedura di compilazione.

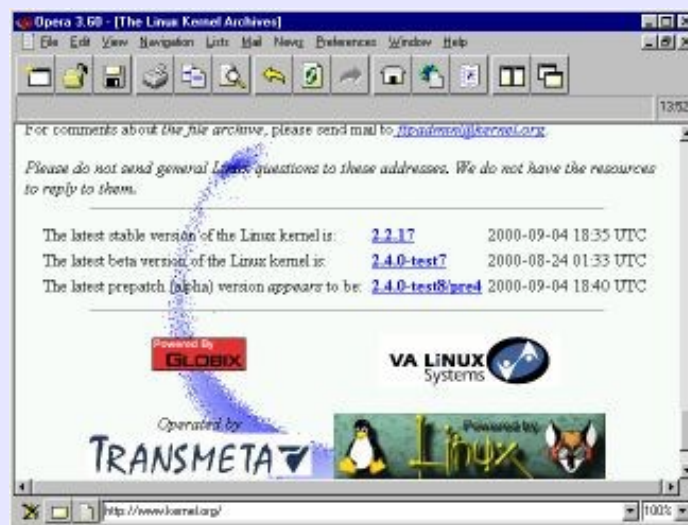
## Pari e dispari

Il posto migliore dove sapere se è disponibile un kernel aggiornato è <http://www.kernel.org>. Questo sito elenca sempre l'attuale situazione dei kernel.

*Dei kernel? Perché, quanti ce ne sono? Chiarisco subito. Su Internet sono disponibili sia il kernel più aggiornato, sia i prototipi degli aggiornamenti successivi. I prototipi sono più sofisticati delle versioni normali del kernel e contengono nuove caratteristiche di funzionamento; tuttavia non sono ancora considerati stabili dalla comunità di Linux, per cui vanno usati soltanto da chi sta collaborando allo sviluppo del sistema operativo o da chi ama il rischio.*

Distinguere un kernel normale da un prototipo è facile. Tutte le versioni del kernel sono etichettate da una serie di tre numeri (ad esempio *linux-2.2.14*). Se il numero di mezzo è pari, si tratta di un kernel normale; se è dispari, è un prototipo.

I kernel normali, a loro volta, si distinguono in stabili (*stable*, in inglese), cioè tranquillamente utilizzabili dall'utente comune, e *beta*, cioè kernel la cui affidabilità è una via di mezzo fra quella dei kernel stabili e quella dei kernel dispari. Ci sono anche kernel normali *alpha*, ancora meno stabili dei beta ma comunque migliori di quelli prototipo. Gli utenti normali dovrebbero usare soltanto i kernel normali classificati come stabili.



**Figura 20–4.** La situazione aggiornata dei kernel è presso <http://www.kernel.org>.

Se volete sapere quale versione di kernel state usando sul vostro computer, aprite una console o una finestra di terminale e digitate **uname -a**. Ottenete una risposta di questo tipo:

```
Linux deepspace9 2.2.14-6.1.1 #1 Thu Apr 13 20:01:58 EDT 2000 i686 unknown
```

Da sinistra a destra, trovate il nome del sistema operativo (*Linux*, ovviamente), il nome di rete del computer, la versione del kernel e la data di compilazione; infine viene indicato il tipo di processore.

## Materiale occorrente

Cominciamo con un caso abbastanza semplice: una ricompilazione della versione già installata del kernel per rimuoverne le parti non necessarie per la vostra specifica configurazione hardware.

Innanzitutto occorre che siano presenti sul disco i cosiddetti *sorgenti* del kernel: gli "ingredienti" di base che lo compongono e che la ricompilazione "impasta" per ottenere il kernel cotto a puntino. Trovate tutti i file necessari sul CD di Red Hat, nella directory */RedHat/RPMS*.

I file più importanti da installare sono:

- *kernel-doc\*.rpm*
- *kernel-headers\*.rpm*
- *kernel-pcmcia\*.rpm* (soltanto se avete hardware PC Card o PCMCIA)
- *kernel-source\*.rpm*
- *kernelcfg\*.rpm*

Al posto dell'asterisco trovate il numero di versione del kernel e/o il tipo di processore a cui il file si riferisce (ad esempio *kernel-doc-2.2.14-6.1.1.i386.rpm*). A proposito del tipo di processore, non fatevi ingannare dalla dicitura *i386*, che non vuol dire "per processori Intel 386", ma "per processori Intel e compatibili dal 386 in su". Per cui prelevate i file contenenti la dicitura *i386* anche se avete un Pentium 4.

Per contro, non prelevate file contenenti diciture tipo *i586*, *i686* eccetera, come ad esempio *kernel-2.2.14-12.i686.rpm*: Sono kernel precompilati, ottimizzati per i vari tipi di processore ma non personalizzati per la vostra specifica installazione: roba da lazzaroni.

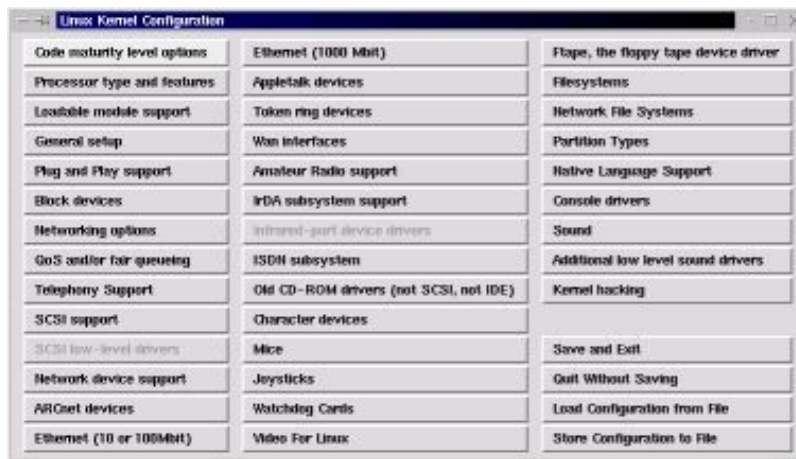
È probabile che molti di questi file siano già installati, ma fate prima a ricaricarli dal CD piuttosto che fare la conta per vedere chi c'è e chi non c'è. Inoltre ci sono vari altri file di supporto che non cito qui perché sono sicuramente già installati e non vengono toccati dalla cura dimagrante della sezione precedente.

L'installazione è semplice: cliccate sul nome del file RPM. Questo fa comparire il programma *kpackage*, nel quale vi basta cliccare su *Install* per installare il tutto. Le directory di installazione sono predefinite e non modificabili, per cui non c'è da preoccuparsi di questo dettaglio.

Terminata l'installazione dei pacchetti necessari, siete pronti per la fase successiva. Aprite una finestra di terminale e digitate **cd /usr/src/linux** per andare nella directory che contiene i sorgenti. Tecnicamente, la directory */usr/src/linux* non esiste: è in realtà un link simbolico che porta alla vera directory che effettivamente contiene i sorgenti, che è in genere */usr/src/linux-x.x.x* (dove *x.x.x* è il numero di versione del kernel). Ma per ora non preoccupatevi di questo dettaglio.

## Scelta dei componenti

Dalla directory */usr/src/linux*, digitate **make xconfig**: questo lancia un questionario grafico, nel quale dovete scegliere quali parti del kernel volete includere e quali no. Ahimè, il programma-questionario è tutto in inglese: attendiamo con fiducia e impazienza una traduzione italiana.



**Figura 20–5. Configurazione grafica del kernel.**

Questa fase della compilazione del kernel consiste semplicemente nel cliccare un pulsante dopo l'altro di questa schermata di configurazione, procedendo per colonne partendo dall'angolo superiore sinistro. Ad ogni pulsante corrisponde una serie di scelte. Nel complesso, queste scelte definiscono la configurazione del nostro kernel "su misura" e la registrano in un file che verrà poi usato come traccia per generare il kernel.

Per ciascuna scelta ci sono sempre almeno due risposte possibili: *y* (cioè "sì") e *n* ("no"). Fin qui non sembra troppo difficile, vero? Il guaio è che molte scelte presentano anche una terza risposta: una misteriosa "m". Visto la difficoltà di certe scelte a cui dovrete rispondere, verrebbe spontaneo pensare che stia per "Mah!".

Purtroppo non è così. La M sta per "modulo", e richiede una spiegazione di come funziona il kernel di Linux. Non è una materia molto stimolante, ma che ci volete fare, non l'ho inventata io. Cercherò di renderla meno soporifera con un esempio.

Immaginate di dover preparare le valigie per un viaggio. Se non sapete che clima vi attende a destinazione, dovrete portare un po' di tutto, dal costume da bagno alle mutande termiche, dalla crema solare ai guanti imbottiti. Risultato: siete sicuramente ben attrezzati per qualsiasi evenienza, ma i vostri bagagli sono mostruosamente pesanti e c'è il rischio che vi facciano pagare il sovrappeso al *check-in*. Le vostre valigie sono come il kernel standard di Linux: contengono un po' di tutto in modo da coprire sicuramente ogni eventualità, ma di certo includono molte cose ingombranti che non userete.

Ovviamente, se sapete che tempo fa nel vostro luogo di arrivo, potete preparare una valigia molto più leggera che contiene soltanto quello che davvero vi serve: se andate ad Acapulco, infilate in valigia costume e crema, ma lasciate a casa mutande termiche e guanti. Questo equivale a compilare un kernel su misura per il vostro computer: sfrondate il superfluo per viaggiare più leggeri.

C'è anche una terza via: saltare in aeroplano senza bagagli, portando soltanto documenti e carta di credito per comperare tutto il necessario quando arrivate a destinazione, e soltanto se e quando vi serve. Così siete sicuri di non portarvi appresso roba inutile e se diluvia non avrete comperato invano la crema solare; tuttavia perderete un po' di tempo a procurarvela in caso di improvviso solleone. Questo approccio è l'equivalente di rispondere *m* a una scelta del programma di configurazione: la parte di codice corrispondente non viene integrata direttamente nel file di kernel ma viene memorizzata separatamente sul disco rigido, con istruzioni per richiamarla e caricarla soltanto se e quando serve.

In altre parole, la compilazione permette di assemblare i vari pezzi del kernel in due modi diversi: integrandoli in un unico file oppure mantenendoli in più file separati. Potete anche mescolare le due tecniche e ottenere un kernel un po' monolitico e un po' modulare: per tornare all'esempio vacanziero, è un po' come partire con lo spazzolino da denti e un cambio di biancheria, le cose più indispensabili, ma comperare il resto sul posto.

Le tre scelte hanno rispettivi vantaggi e svantaggi:

- Se un pezzo di kernel dedicato a un dispositivo non viene incluso del tutto (rispondendo  $n$ ), il kernel diventa più piccolo e quindi occupa meno memoria, e lo spazio occupato su disco da Linux si riduce; tuttavia, se il computer è dotato di quel dispositivo, Linux non potrà gestirlo.
- Se includete una parte di kernel direttamente nel file primario del kernel medesimo (rispondendo  $y$ ), il dispositivo gestito da quella parte di kernel verrà gestito da Linux nel modo migliore e più veloce possibile, ma al prezzo di un'occupazione supplementare di spazio in memoria.
- Se rispondete  $m$  per rendere disponibile a Linux una parte di kernel (un *modulo*, appunto) senza integrarla direttamente nel file primario del kernel stesso, non aumenterete l'occupazione di memoria del kernel e potrete comunque gestire, all'occorrenza, il dispositivo a cui quella parte di kernel è dedicata; per contro, dato che Linux deve leggere quella parte di kernel dal disco rigido invece che dalla memoria, i suoi tempi di reazione saranno più lunghi.

La regola generale, quindi, è rispondere  $y$  a tutte le scelte che riguardano funzioni o dispositivi che usate molto spesso,  $m$  per quelli che vi possono servire saltuariamente, e  $n$  per quelli che sicuramente non vi serviranno mai. Se non avete idea di che cosa tratti una determinata scelta, non modificate le sue impostazioni.

Tutto chiaro? Allora cominciamo il questionario. Nelle pagine che seguono, Odo e io abbiamo raccolto un po' di consigli su cosa includere e cosa escludere nella compilazione di un kernel su misura. Ovviamente, siccome le vostre esigenze possono essere diverse dalle nostre, quello che è su misura per noi non è detto che lo sia anche per voi, ma è probabile che lo sia, visto che siamo partiti dal presupposto che abbiate un computer abbastanza generico e privo di accessori troppo stravaganti.

Non è necessario che completiate tutto il questionario in un'unica sessione. In qualsiasi momento potete tornare al menu principale del programma di configurazione, cliccare su *Save and Exit* per salvare le impostazioni fatte fino a quel punto e uscire dal programma di configurazione. Potete proseguire il questionario in un secondo momento: basta rilanciare il programma, che ricaricherà automaticamente le impostazioni che avete già fatto.

Voce di menu	Consiglio
Code maturity level options (opzioni per il livello di maturità del codice)	Se non fate parte del team di sviluppo di Linux, rispondete $n$ . Questa scelta, infatti, consente al programma di configurazione di includere (con richiesta di conferma) parti di kernel non ancora considerate del tutto stabili che però consentono di gestire funzioni e dispositivi che il kernel normale non contempla.
Processor type and features (tipo e caratteristiche del processore)	Per un corretto funzionamento del kernel è indispensabile scegliere il tipo di processore giusto in questa schermata. Se avete un processore superiore al 486sx, potete impostare a $n$ il parametro <i>Math emulation</i> (emulazione del coprocessore matematico). Per il resto, accettate pure le impostazioni proposte.
Loadable module support (supporto per modul caricabili)	Lasciate tutto attivato, altrimenti non potrete usare i moduli per il kernel.
General setup (impostazione generale)	Attivate con $y$ l' <i>Advanced Power Management BIOS support</i> se avete un BIOS che gestisce l'APM: questo consentirà a Linux di spegnere fisicamente il computer quando chiudete il sistema operativo, esattamente come fa Windows.

Plug and Play support (supporto per le funzioni Plug and Play)	Nulla da modificare.
Block devices (dispositivi a blocchi)	Nulla da modificare, ma ricordate che se avete unità a disco esterne (ad esempio collegate alla porta parallela), è qui che dovete venire a regolare le impostazioni del kernel. Se avete un drive ZIP su porta parallela o volete usare un masterizzatore IDE con programmi concepiti per drive SCSI, attivate (almeno come modulo) <i>SCSI emulation support</i> .
Networking options (opzioni di rete)	Disabilitate tutto quel che riguarda <i>IPX</i> (a meno che dobbiate collegarvi a reti Novell), <i>Appletalk</i> (a meno che dobbiate collegarvi a macchine Apple) e tutti i riferimenti a <i>X.25</i> .
QoS and/or fair queuing (qualità del servizio e/o criteri di gestione equa delle code)	Tutto disattivato.
Telephony support (supporto per servizi di telefonia)	Tutto disattivato.
SCSI support (supporto per dispositivi SCSI)	Da attivare soltanto se avete dispositivi SCSI, un drive ZIP su porta parallela o masterizzatori IDE da usare tramite programmi per apparecchi SCSI; se li avete, attivate l'opzione <i>SCSI generic support</i> come modulo.
SCSI low-level drivers (driver a basso livello per dispositivi SCSI)	Tutto disattivato, salvo abbiate dispositivi SCSI o drive ZIP su porta parallela.
Network device support (supporto per dispositivi di rete)	Lasciate tutto com'è, salvo probabilmente disattivare <i>General instruments Surfboard 1000</i> , <i>FDDI driver support</i> e <i>SBN112-xx support</i> .
ARCnet devices (dispositivi ARCnet)	Da attivare soltanto se avete una scheda di rete ARCnet.
Ethernet (10 or 100 Mbit) (Ethernet a 10 o 100 megabit)	Se avete una scheda di rete Ethernet, scegliete <i>y</i> nella prima opzione della lista e poi attivate l'opzione corrispondente alla marca e al modello della vostra scheda. Tenete presente che molte schede di rete non citate nella lista sono cloni della NE2000 e quindi funzionano se attivate <i>NE2000/NE1000 support</i> .
Ethernet 1000 Mbit (Ethernet a 1000 megabit)	Da attivare soltanto se avete una connessione di rete a mille megabit (altamente improbabile).
Appletalk devices (dispositivi Appletalk)	Se non avete dispositivi Appletalk, non attivate niente.
Token ring driver support (supporto per driver Token Ring)	Da attivare soltanto se il vostro computer è collegato a una rete Token Ring (altamente improbabile).
Wan interfaces (interfacce Wan)	Lasciate tutto disattivato.
Amateur Radio support (supporto radioamatori)	Lasciate tutto disattivato, salvo che vogliate collegare il vostro computer a un impianto di trasmissione radioamatoriale.
IrDA subsystem support (supporto per sottosistema IrDA)	Da attivare soltanto se avete porte a infrarossi.

Infrared-port devices (dispositivi con porta a infrarossi)	Da attivare soltanto se avete porte a infrarossi.
ISDN subsystem (sottosistema ISDN)	Da attivare soltanto se avete una connessione ISDN.
Old CD-ROM drivers (not SCSI, not IDE) (driver per vecchi drive CD-ROM non-SCSI e non-IDE)	Da attivare soltanto se avete un lettore di CD-ROM che non è né SCSI né IDE.
Character devices (dispositivi a caratteri)	Lasciare tutto invariato. Disattivate <i>QIC-02 tape support</i> , a meno che abbiate un drive a nastro di questo tipo.
Mice (tipi di mouse)	Lasciare tutto invariato, a meno che il vostro mouse sia specificato nell'elenco che viene presentato (improbabile).
Joysticks (tipi di joystick)	Se avete un joystick, attivate <i>Joystick support</i> (con <i>y o m</i> ) e la voce corrispondente al vostro modello di joystick.
Watchdog cards (schede watchdog)	Da attivare soltanto se avete una scheda <i>watchdog</i> (molto improbabile).
Video for Linux (video per Linux)	Da attivare soltanto se avete schede TV/radio o Webcam indicate nella lista presentata dalla schermata.
Ftape	Da attivare soltanto se avete un dispositivo a nastro QIC-80/Travan.
Filesystems (tipi di filesystem supportati)	Attivate come moduli tutti i filesystem che prevedete di utilizzare e disattivate gli altri. Ad esempio, se avete Linux su un portatile, è probabilmente inutile attivare il supporto per OS/2.
Network file systems (filesystem di rete supportati)	Da attivare soltanto se intendete condividere attraverso la rete dischi installati su altri computer.
Partition types (tipi di partizione)	Tutto disattivato.
Native language support (supporto lingue locali)	Attivare come modulo le codepage contrassegnate <i>United States, Europe, Latin-1e Latin 9</i> (supporto euro). Le altre codepage sono disattivabili, salvo che vogliate gestire documenti redatti nelle lingue alle quali si riferiscono.
Console drivers (driver per le console)	Attivate le prime due opzioni, <i>VGA text console</i> e <i>Video mode selection support</i> , se volete usare formati diversi da 80 x 25 caratteri nelle console.
Sound (Audio)	Se avete una scheda audio, attivate (con <i>y o m</i> ) la prima voce, <i>Sound card support</i> , e le voci riguardanti il vostro specifico modello di scheda audio.
Additional low level sound drivers (altri driver audio a basso livello)	Se avete una scheda audio, attivate (con <i>y</i> ) la prima voce, <i>Additional low level sound drivers</i> , e le voci riguardanti il vostro specifico modello di scheda audio.
Kernel hacking	Disattivato, sempre e comunque.

Terminate le domande, tornate alla schermata principale del programma di configurazione e cliccate su *Save and Exit* se volete salvare la configurazione nel file standard, sovrascrivendo quella esistente, oppure cliccate su *Store configuration to file* per salvare la nuova configurazione altrove e mantenere quella originale.





ma per fortuna non richiede la vostra presenza: potete farvi una buona pausa caffè, o dedicarvi ad altro lavoro, intanto che il nuovo kernel viene generato e poi scritto sul floppy.

Anche se il dischetto viene scritto alla fine di questa fase, il lavoro non è ancora finito: digitate **make modules**, poi preparatevi a un'altra attesa di diversi minuti.

Quando ricompare il prompt dei comandi, digitate **make modules\_install**. L'elaborazione risultante è brevissima.

Infine, digitate **depmod -a**. Questo comando crea un file (*/usr/lib/modules/versione\_kernel/modules.dep*) contenente i nomi dei moduli che possono essere caricati. Attenzione a eventuali messaggi d'errore: sottoponeteli al vostro linuxiano di fiducia.

Bene! In teoria a questo punto il vostro nuovo kernel ottimizzato su misura è pronto per l'uso. Per provarlo, chiudete Linux e riavviate, tenendo il floppy inserito nel drive. Prestate molta attenzione ai messaggi anomali che potrebbero comparire durante il riavvio e prendetene nota.

Non è detto che eventuali messaggi di avvertimento siano sintomo di fallimento: Linux probabilmente sta semplicemente facendo il pignolo e ricordandovi che c'è ancora qualche perfezionamento da fare. Affidatevi comunque alle mani esperte di un linuxiano per decifrare il senso dei messaggi all'avvio.

Se Linux non parte del tutto o si pianta a metà strada, rimuovete il floppy e riavviate il computer: in teoria ricomparirà il Linux di sempre, come se nulla fosse accaduto. Se non dovesse succedere, ricorrete al backup che avete fatto prima di cominciare e rileggete gli appunti in cui avete preso nota di quello che avete fatto, controllando di non aver dimenticato nulla.

Se invece Linux parte, fate un giretto qua e là per vedere se funzionano tutte le periferiche (dischi, stampanti, mouse, scheda di rete). Tutto a posto? Datevi un pizzicotto. Per verificare che non state sognando, ma che state davvero eseguendo un kernel che avete creato con le vostre mani, aprite una finestra di terminale e digitate **uname -a**. Come già accennato, compare un messaggio che specifica la versione di kernel in esecuzione e, cosa più importante, la sua data di compilazione. Se è quella di oggi, state davvero eseguendo il vostro kernel "fai da te".

## Sostituire il kernel

Una volta che vi siete sincerati che il nuovo kernel funziona bene, potete decidere di sostituirlo al kernel presente sul disco rigido, in modo da fare a meno del dischetto di avvio contenente il nuovo kernel e quindi ottenere un avvio più veloce.

Mi raccomando, non prendete alla leggera questa decisione. Se cancellate il kernel vecchio e lo sostituite con uno nuovo che non funziona correttamente, potreste paralizzare completamente Linux e renderlo inservibile. Potreste addirittura non essere più in grado di riavviarlo, perdendo quindi l'accesso a eventuali dati presenti nella partizione Linux. Certo, ci sono procedure d'emergenza per risolvere questi problemi, ma vi assicuro che sono lunghe e dolorose. Se siete un tipo prudente, restate fedeli al vostro dischetto di kernel.

Se invece amate la sperimentazione e il rischio, ecco come procedere, preferibilmente con un linuxiano a portata di mano per i momenti difficili.

- Fate un po' di copie di sicurezza: copiate */usr/src/linux-x.x.x* a un'altra directory, ad esempio */usr/src/linux-x.x.x\_originale*, e copiate la directory */boot* a un'altra directory, ad esempio */boot\_originale*.
- Rieseguite la compilazione partendo dal comando **make dep**, ma invece di dare il comando **make bzdisk**, digitate **make bzImage**. Attenzione a digitare correttamente la I in maiuscolo. Questo comando crea il file del kernel sul disco rigido invece che sul floppy. Il kernel nuovo non sovrascrive quello vecchio, per cui se qualcosa va storto

potrete comunque avviare Linux senza problemi.

- Il risultato del passo precedente è un file di nome *bzImage* che risiede nella directory */usr/src/linux-x.x.x/arch/i386/boot/*. Siete al punto di non ritorno: copiate il file alla directory */boot* e dategli il nome del kernel originale compresso, che è *vmlinuz* seguito dai numeri di versione. Attenzione a digitare *vmlinuz* con la zeta, non con la X.
- Eseguite il comando **lilo**. Questo dice a LILO di "agganciare" il nuovo kernel e riscrive il settore di boot del disco rigido. Se preferite mettere LILO su un floppy per avere un dischetto che fa da "chiave di avviamento", seguite le istruzioni del Capitolo 13 (*Messa a punto di Linux*).
- Riavviate Linux e tenete le dita incrociate. Se Linux parte correttamente, digitate **uname -a** in una finestra di terminale per verificare che state davvero eseguendo il nuovo kernel dal disco rigido (lo si vede dalla data indicata nella risposta a questo comando).

## Aggiornare il kernel

Il kernel di Linux è soggetto ad aggiornamenti molto frequenti, a volte addirittura quotidiani. Non è indispensabile procurarsi sempre ogni nuova versione del kernel: se quella che avete funziona soddisfacentemente, non sostituirla.

Ogni tanto, però, esce una nuova versione di kernel che contiene cambiamenti piuttosto importanti o consente finalmente di gestire qualche dispositivo di cui avevate bisogno sotto Linux. In casi come questi conviene installare l'aggiornamento del kernel.

Come tutte le modifiche al cuore del sistema operativo, anche questa è molto pericolosa. Se fallisce, potreste rendere totalmente inservibile il vostro Linux. Se possibile, eseguite l'aggiornamento in presenza di un linuxiano.

## Procurarsi il nuovo kernel

Potete prelevare i file del nuovo kernel direttamente da Internet, ad esempio presso <http://www.redhat.it>, oppure da un CD di distribuzione. Se avete una normale connessione via modem, il prelievo da Internet richiede un po' di tempo, dato che i file contenenti il codice sorgente di Linux occupano una ventina di megabyte.

I file del kernel sono solitamente disponibili in due formati: quello con l'estensione *.rpm*, ottimizzato per Red Hat Linux, e quello con l'estensione *.tar.gz*, valido per qualsiasi distribuzione. La versione RPM è più facile da usare, per cui preferitela all'altra (il contenuto è comunque identico). La descrizione che segue presuppone che abbiate la versione RPM.

I file che dovete procurarvi sono gli stessi elencati nella sezione *Materiale occorrente* di questo capitolo. Copiateli a una directory temporanea della partizione Linux ed installateli uno per uno cliccando sul nome del file e cliccando sul pulsante *Install* nella finestra di kpackage che compare.

## Generare il nuovo kernel

Non temete di sovrascrivere il kernel precedente: i file del nuovo kernel vengono installati in directory separate. Il pericolo, semmai, è di intasare il disco rigido con troppe versioni di kernel. Ricordatevi di fare pulizia delle versioni che non usate più.

Ad esempio, se il vostro kernel attuale è la versione 2.2.14 e volete installare la versione 2.2.16, il kernel attuale risiede in */usr/src/linux-2.2.14*, mentre la versione nuova verrà installata in */usr/src/linux-2.2.16*, e così via. I kernel dormono quindi in stanze separate. Fate attenzione alla possibilità che venga alterato il link simbolico che collega la directory fittizia */usr/src/linux* alla directory effettiva che contiene i file del kernel: installando i file di un nuovo kernel, questo link potrebbe puntare alla directory del nuovo kernel invece che a quella del kernel preesistente.

Non vi preoccupate troppo di questo problema: l'importante è che sappiate se è stata apportata questa modifica e che vi ricordiate di ripristinare la situazione originale se fate marcia indietro e ritornate al kernel preesistente.

Cosa più importante, se durante l'installazione del nuovo kernel compaiono messaggi che segnalano problemi di dipendenza, lasciate stare e parlatene con il vostro linuxiano di fiducia.

Nessun problema? Allora andate nella directory in cui si è installato il nuovo kernel e usatela al posto di `/usr/src/linux` per eseguire la sequenza già vista per la ricompilazione del kernel esistente, partendo dal comando **make xconfig**. Al termine della sequenza dovreste ottenere un floppy contenente il nuovo kernel. Tutto qui.

Una volta assodato che il nuovo kernel funziona, potete decidere di sostituirlo a quello precedente anche sul disco rigido: basta rieseguire le operazioni descritte nella sezione *Sostituire il kernel*.

## Sostenete *Da Windows a Linux!*

Questo libro è distribuito **gratuitamente**, ma le **donazioni** sono sempre ben accette, sia tramite **PayPal**, sia tramite il collaudato sistema della **banconota in una busta**. Se volete dettagli e istruzioni su come procedere, le trovate presso <http://www.attivissimo.net/donazioni/donazioni.htm>.

**Grazie!**

Da Windows a Linux – (C) 1999–2003 Paolo Attivissimo e Roberto Odoardi.  
Questo documento è liberamente distribuibile purché intatto.