

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

19. Manutenzione di base

Dunque, dunque. Arrivati a questo punto avete un Linux che funziona (e magari funziona anche bene, grazie a un po' di messa a punto). Per mantenerlo sano e stabile, consiglio caldamente le cure amorevoli che vi descrivo qui. Come al solito, in questo capitolo vi descrivo soltanto le operazioni strettamente indispensabili: in realtà la manutenzione potrebbe essere molto più approfondita. Ma per gli approfondimenti c'è tempo.

Manutenzione a distanza

Nel corso di questo libro potreste aver avuto l'impressione che io sia riuscito a decifrare Linux praticamente tutto da solo e che Odo si sia limitato ad un'occasionale imposizione delle mani. Mi piacerebbe farvelo credere, così da fare bella figura, ma lo sapete che sono una frana a mentire. Le volte in cui Odo è intervenuto per tirarmi fuori dai guai sono innumerevoli.

Come è successo a me, succederà anche a voi di impantanarvi con Linux. Non voglio menar gramo; è un dato di fatto. Probabilmente, però, sarete più fortunati di me: avrete a portata di mano un linuxiano che possa venirvi a trovare e sedersi davanti al computer per risolvervi i problemi. Io, invece, ho avuto un problema in più: per gran parte della lavorazione di questo libro, io e Odo non abbiamo potuto lavorare fianco a fianco (io ero in Inghilterra e lui in Italia).

Se anche voi avete bisogno di assistenza ma il vostro linuxiano non è in grado di sedersi al vostro computer a causa delle distanze (non necessariamente internazionali), potete fare quello che ho fatto io: sfruttare la potenza di Linux per predisporre un canale di comunicazione che consenta la manutenzione a distanza.

In questo modo, se avete un problema, potete chiedere al vostro linuxiano di fiducia di "telecomandare" il vostro computer e correggerne le magagne via Internet, senza dovervi venire a trovare. Questo è un campo in cui Linux è decisamente più avanti di Windows, grazie al fatto che Linux è interamente controllabile tramite la sua interfaccia testuale, che viaggia molto veloce anche sulla connessione Internet più scalcinata.

Chiaramente la telemanutenzione ha i suoi rischi. In pratica, è come lasciare l'auto aperta in strada, con le chiavi nel cruscotto, e chiedere al meccanico di venirsela a prendere. È ovvio che qualcun altro potrebbe passare di lì, notare la macchina aperta e decidere di portarsela via, rubare l'autoradio o spiaccicarvi un *cheeseburger* sui sedili.

Usare una semplice sessione di telnet e lasciare che il linuxiano entri nel vostro computer come *root*, quindi con poteri assoluti (necessari per fare manutenzione), non è affatto una bella idea. Dato che la comunicazione fra il linuxiano e il vostro computer avviene via Internet, chiunque con un po' di astuzia può entrare al posto del linuxiano e fare opera di sistematica devastazione.

Ci vuole quindi un canale di comunicazione sicuro e protetto. Io e Odo abbiamo usato due programmi:

- per Red Hat Linux 7.0 abbiamo adoperato *Openssh*, incluso nei CD di installazione;
- per Red Hat Linux 6.2, che non include *Openssh*, abbiamo adottato *ssh*, un programma open source e gratuito facilmente reperibile su Internet. Esiste una versione 2, ma è a pagamento e ha varie magagne che non è il caso di approfondire qui. Prelevate la versione 1.2.27, che funziona bene.

Le istruzioni che seguono spiegano come installare *Openssh* e *ssh* per una connessione protetta via Internet che consenta la telemanutenzione. Per semplificare, darò per scontato che il computer da mantenere sia il vostro.

Installazione di Openssh (Red Hat Linux 7.0)

Il programma *Openssh* è costituito da tre parti, contenute in altrettanti pacchetti RPM:

- un pacchetto *base*, da installare su entrambi i computer da collegare, il cui nome inizia per *openssh* seguito da un numero (ad esempio *openssh-2.1.1p4-1.i386.rpm*); questo pacchetto viene installato automaticamente durante l'installazione di Red Hat Linux 7.0;
- un pacchetto *client*, che va installato sul computer che deve collegarsi al vostro e si chiama *openssh-clients* seguito da un numero; anche questo pacchetto viene installato automaticamente durante l'installazione di Red Hat Linux 7.0;
- un pacchetto *server*, che installate sul vostro computer e si chiama *openssh-server* seguito da un numero; questo pacchetto non viene installato automaticamente e dovete provvedere a mano (lo trovate sul primo CD di installazione).

Il vostro linuxiano manutentore deve quindi installarsi la parte client più la base; tuttavia, per semplificarvi la vita, vi conviene installare tutti e tre i pacchetti. Nel database di kpackage, il pacchetto base e il client di Openssh vengono registrati nella sezione *Applications > Internet*, mentre il server va in *System Environment > Daemons*.

Terminata l'installazione con kpackage, riavviate Linux. Esiste un modo per attivare Openssh senza dover riavviare, ma non ve lo voglio infliggere. Al riavvio noterete una congrua serie di messaggi che annunciano la generazione delle chiavi (password) e il messaggio nuovo *Starting sshd*. Questo conferma che il demone di Openssh è attivo automaticamente.

Installazione di ssh (Red Hat Linux 6.2)

Il programma *ssh* è in realtà costituito da tre parti, distribuite via Internet in altrettanti pacchetti RPM:

- una *base*, da installare su entrambi i computer da collegare, che si chiama *ssh-1.2.27-6cl-i386.rpm*;
- una *client*, che va installata sul computer che deve collegarsi al vostro e si chiama *ssh-clients-1.2.27-6cl.i386.rpm*;
- una *server*, che installate sul vostro computer e si chiama *ssh-server-1.2.27-6cl.i386.rpm*.

Il vostro linuxiano manutentore deve quindi installarsi la parte client più la base; tuttavia, per semplificarvi la vita, vi conviene installare tutti e tre i pacchetti. La sequenza giusta di installazione è questa: prima la base, poi il client e/o il server. Se non installate prima la base, kpackage vi segnalerà un problema con le dipendenze. Ovviamente dovete essere *root* per eseguire l'installazione.

Nel database di kpackage, il pacchetto base e il client di *ssh* vengono registrati nella sezione *Applications > Internet*, mentre il server va in *System Environment > Daemons*.

Terminata l'installazione con kpackage, riavviate Linux. Esiste un modo per attivare ssh senza dover riavviare, ma non ve lo voglio infliggere. Una volta tanto potete anche riavviare, suavia. Al riavvio noterete il messaggio nuovo *Starting sshd*, che conferma che il demone di ssh è attivo automaticamente.

Ssh per Windows

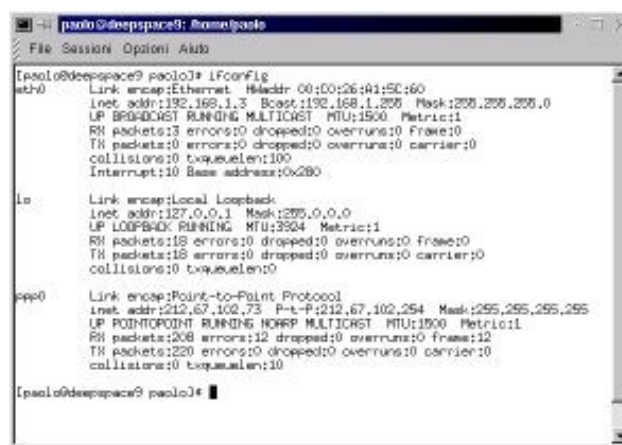
Non è detto che il computer usato dal manutentore debba per forza essere una macchina Linux. È ovviamente presumibile che un linuxiano abbastanza bravo da fare manutenzione alle macchine altrui usi Linux, ma ci possono essere parecchie circostanze in cui il manutentore non ha a disposizione il proprio computer e deve lavorare seduto davanti a una macchina Windows.

Per questo tipo di esigenze c'è *Putty*, un client ssh gratuito per Windows, scritto da Simon Tatham e reperibile facilmente in Rete (ad esempio presso <http://www.chiark.greenend.org.uk/~sgtatham/putty/>). Non ha bisogno di installazione: è un programmino da 150 K che si esegue così com'è.

Telemanutenzione all'atto pratico

Una volta che avete installato ssh o Openssh, siete pronti. Telefonate al vostro manutentore e avvisatelo che state per rendervi disponibili per la telemanutenzione e poi collegatevi a Internet. Potrebbe venirvi il dubbio di dovervi collegare come *root*, dato che state predisponendo il computer per attività di manutenzione, ma non è così. Mai, mai, *mai* collegarsi a Internet come *root*.

Quando siete collegati a Internet, dovete scoprire il vostro indirizzo IP temporaneo: quello che il fornitore d'accesso vi assegna per la durata della connessione. Aprite una console o una finestra di terminale e digitate **ifconfig**. Otterrete un'elencazione delle connessioni di rete locale e Internet attive, simile a quella mostrata in Figura 19–1.



```

paolo@deepspace9: ~$ ifconfig
eth0:
  Link encap:Ethernet  HWaddr 00:D0:26:41:5C:60
  Inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
  RX packets:13  errors:0  dropped:0  overruns:0  frame:0
  TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
  collisions:0  txqueuelen:100
  Interrupt:10  Base address:0x280

lo:
  Link encap:Local Loopback
  Inet addr:127.0.0.1  Mask:255.0.0.0
  UP LOOPBACK RUNNING  MTU:65536  Metric:1
  RX packets:18  errors:0  dropped:0  overruns:0  frame:0
  TX packets:18  errors:0  dropped:0  overruns:0  carrier:0
  collisions:0  txqueuelen:0

ppp0:
  Link encap:Point-to-Point Protocol
  Inet addr:212.67.102.73  P-t-P:212.67.102.254  Mask:255.255.255.255
  UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1900  Metric:1
  RX packets:208  errors:12  dropped:0  overruns:0  frame:12
  TX packets:1220  errors:0  dropped:0  overruns:0  carrier:0
  collisions:0  txqueuelen:10

paolo@deepspace9: ~$

```

Figura 19–1. Scopriamo il nostro indirizzo IP durante una connessione a Internet.

La sezione che vi interessa è quella etichettata *ppp0*: l'indirizzo IP assegnatovi per la connessione a Internet è quello indicato dopo *inet addr*. Ora che lo sapete, comunicatelo al manutentore, ad esempio via e-mail o con una telefonata (usando un'altra linea, ovviamente, o un telefonino).

In alcuni casi, ad esempio all'interno di reti aziendali filtrate oppure con collegamenti speciali tipo ADSL o simili, potreste non avere un indirizzo IP valido. Potreste per esempio trovarvi indirizzi IP del tipo *192.168.xxx.yyy*, *172.16.xxx.yyy* o *10.xxx.yyy.zzz*, che sono inaccessibili dall'esterno. In questo caso dovete procurarvi un modem e collegarvi con un metodo più "tradizionale", usando per l'occasione un fornitore d'accesso gratuito come Tiscalinet e soci. Soprattutto in ambito aziendale, vi conviene anche parlarne con il responsabile della sicurezza informatica, visto che state temporaneamente scavalcando le sue difese.

Il manutentore (se usa Linux) digita **ssh -l root** seguito dal vostro indirizzo IP. Sul suo computer compare una serie di richieste, fra cui c'è quella della vostra password di *root*. Digitandola, il manutentore entra come *root* onnipotente sul vostro computer e può procedere alla manutenzione. Se il manutentore usa Windows e Putty la procedura è sostanzialmente analoga, solo che al posto della riga di comando c'è una finestra di dialogo.

A questo punto potete procedere alla telemanutenzione. Per terminare la connessione, il manutentore digita semplicemente **exit** come per una qualsiasi sessione di telnet. Da parte vostra, potete sempre terminare la connessione chiudendo a connessione a Internet (o, nel caso peggiore, staccando la spina del telefono, ma lo sconsiglio).

In condizioni normali, il manutentore ha accesso all'interfaccia testuale di Linux e non può vedere gli effetti del suo lavoro se ha a che vedere con l'interfaccia grafica. Per queste cose è necessaria la vostra presenza davanti al computer e ci vuole un modo per comunicare durante la manutenzione, per sapere se le modifiche funzionano o meno.

Se avete entrambi una seconda linea telefonica o un telefonino, usate quelli, ma se non vi è possibile o volete evitare bollette salate, potete "parlarvi" tramite tastiera grazie al programma *talk*. Occorre installare sulla vostra macchina (e preferibilmente anche su quella del manutentore) la parte server di questo programma, che è sul CD di Red Hat Linux, nella directory *RPMS*, nel file che inizia per *talk-server* (ad esempio *talk-server-0.16-4.i386.rpm*). Inoltre conviene che sia definito almeno un altro utente normale oltre a quello che usate voi di solito.

Dopo aver installato il server di *talk*, dovrete probabilmente riavviare il computer per attivarlo. Se usate Red Hat Linux 7.0, modificate i file di controllo */etc/xinetd.d/talk* e */etc/xinetd.d/ntalk*, sostituendo *disable=yes* con *disable=no*, prima di riavviare o dare il comando */etc/rc.d/init.d/xinetd restart* già visto nel Capitolo 18.

Per comunicare tramite *talk*, dovete aprire una console e fare login come utente normale. Digitate **mesg y** per attivare la visualizzazione dei messaggi di sistema sulla console (probabilmente è già attiva, ma non si sa mai); il manutentore farà altrettanto. Se potete, comunicate al manutentore il nome dell'utente con il quale avete fatto login. Se non potete, il manutentore dovrebbe essere abbastanza abile da scoprirselo da solo (con il comando **who**). Lasciate la console sullo schermo e aspettate.

Un altro indizio del fatto che il manutentore sa quello che fa è che attiva una seconda sessione ssh oltre a quella già aperta (sì, è possibile farlo): in questa seconda sessione entra come utente normale (è a questo che serve il secondo utente normale) e usa *talk*. In questo modo la prima sessione ssh, quella iniziata come *root*, rimane libera per il lavoro e la seconda sessione viene usata per comunicare.

Il manutentore digita **talk** seguito dal nome dell'utente che state usando. Sul vostro schermo compare un messaggio che vi segnala la richiesta di comunicazione del manutentore e vi invita a rispondere digitando **talk** seguito dal nome dell'utente usato dal manutentore.

In alternativa, potete iniziare voi la "conversazione", anche in una finestra di terminale (comoda da tenere in un angolo dello schermo) oltre che in una console, digitando **talk** e il nome dell'utente usato dal manutentore.

A questo punto lo schermo si divide orizzontalmente in due parti: in quella di sopra compare quello che scrivete voi, in quella di sotto viene visualizzato quello che vi scrive il manutentore. Per terminare la chiacchierata, premete **Ctrl-C**.



Figura 19–2. Talk al lavoro.

La comunicazione tramite *talk* è possibile anche durante una normale sessione di telnet; è un ottimo sistema per *chattare* a due durante una connessione via Internet (ed è molto, *molto* più veloce dei servizi di *chat* normali). Tuttavia *talk* non è

protetto contro le intercettazioni, quindi se lo usate insieme al normale telnet quello che scrivete può essere ascoltato. Se invece lo usate, come nel caso appena descritto, durante una sessione di *ssh*, non ci sono problemi, perché talk utilizza il canale protetto creato da *ssh*.

Abituati come siete a lavorare con Windows, probabilmente vi stupirà il fatto che sia possibile riavviare l'interfaccia grafica senza far cadere la connessione a Internet, ma ricordatevi che chiudere l'interfaccia e rientrare con un altro utente non comporta il riavvio di Linux. Questo è molto comodo quando il manutentore deve modificare la configurazione dell'interfaccia e vi chiede di entrarvi prima come *root* e poi come utente normale o viceversa.

Tuttavia quest'opzione non è attivata automaticamente: dovete andare nella configurazione di *kppp* (descritta nel Capitolo 14) e selezionare la scheda *PPP*, dove troverete l'opzione *Disconnetti all'uscita da X*. Togliete il segno di spunta da quest'opzione e la connessione non verrà chiusa automaticamente quando riavviate l'interfaccia grafica.

Creare un nuovo utente

Se avete letto la sezione precedente dedicata alla telemanutenzione, può darsi che vi stiate chiedendo come si crea un utente nuovo. Telemanutenzione a parte, ci sono varie altre ragioni per cui può essere necessario creare un utente nuovo: ad esempio per fare delle prove senza pasticciare con l'utente che usate per il normale lavoro col computer, o perché ci sono altre persone che condividono il computer con voi: potete assegnare loro un'utenza personale dove possono fare tutti i pasticci che vogliono senza interferire con il funzionamento degli altri utenti.

Ci sono vari modi per creare un nuovo utente. Ad esempio, potete usare il comando **adduser**, con le varie opzioni descritte nella sua pagina man; oppure potete usare *kdeadmin*, il pacchetto di amministrazione dell'interfaccia grafica. Controllate con *kpackage* se questo pacchetto è installato: se non lo è, installate il file RPM il cui nome inizia per *kdeadmin* (nella distribuzione Red Hat Linux 7.0 è sul secondo CD-ROM, nella directory */RedHat/RPMS*).

Per creare un nuovo utente dovete essere *root*. Dal Pannello, scegliete Sistema > Gestione Utenti, e cliccate sul menu Utente, scegliendo la voce *Aggiungi*. Assegnate al nuovo utente il nome che preferite, meglio se concordato con la persona che dovrà adoperarlo.

Nella schermata successiva, immettete il nome completo dell'utente; come shell di login, dategli */bin/bash* e dategli una password cliccando su *Imposta password*. Infine cliccate su *OK* per attivare le impostazioni e scegliete File > Salva per salvarle.



Figura 19–3. Creare un nuovo utente.

Se vi state chiedendo che cosa vuol dire *Copia scheletro*, ve lo dico subito. Attivando quest'opzione, viene creata automaticamente la struttura di directory standard che c'è nella home directory di ciascun utente: la directory *Desktop*, i file di configurazione, le sottodirectory *Autostart*, *Templates*, *Trash* eccetera. Siccome inizialmente la struttura è vuota, viene chiamata "scheletro".

Cancellare un utente

Come avrete intuito, con lo stesso programma si possono modificare i parametri degli utenti e si possono anche eliminare quelli non più necessari: basta cliccare sull'utente e poi sul pulsante *Elimina Utente* (DEL, il secondo da sinistra). Se confermate la vostra intenzione cliccando su *Elimina*, vi viene chiesto anche se volete cancellare il gruppo privato dell'utente. Se confermate ancora una volta, l'utente viene eliminato.

L'unico residuo che dovete eliminare manualmente è la home directory dell'utente e le sue sottodirectory.

Backup sotto Linux

Questa sezione del libro è dedicata alla memoria del disco rigido di Zane, uno dei miei revisori. Ha avuto un problema fisico serio (il disco, non il revisore) per cui Zane ha perso dieci gigabyte di dati, cinque anni della sua (intensa) vita informatica, tutta la sua posta, la musica, e altro ancora. Tutto perché non aveva fatto il backup di quei dati.

Se credete che fare un backup sia superfluo perché i dischi non si rompono mai o perché credete di non avere dati preziosi sul vostro computer, accomodatevi. Evidentemente siete più inclini a imparare dalle vostre batoste personali che dalle disgrazie degli altri.

Supporti di backup

Con le dimensioni esagerate dei dischi rigidi attuali è facile trovarsi con una quantità notevole di dati da salvare. Fare una copia di backup su floppy è semplicemente impensabile: un gigabyte equivarrebbe a circa 740 dischetti. Esistono le cosiddette *unità a nastro*, che registrano i file su un nastro di aspetto simile a quello delle cassette audio, ma sono cari, lenti e, francamente, molto macchinosi. Potreste anche considerare le unità di backup Iomega Jaz o simili, ma richiedono

un investimento iniziale piuttosto salato.

A mio avviso ci sono due sole possibilità di effettuare backup in modo efficiente ed efficace: un secondo disco rigido oppure un CD masterizzato.

Il secondo disco rigido è la soluzione più costosa ma è di gran lunga la più semplice: basta copiarvi i file, eventualmente comprimendoli con comandi tipo **gzip** (l'equivalente Linux di PKZIP) per occupare meno spazio. La copia avviene alla velocità tipica dei dischi rigidi, per cui un backup completo richiede un solo comando e pochi minuti di attesa.

Il difetto del backup su un secondo disco rigido è che il disco, salvo soluzioni piuttosto particolari (dischi rimovibili), è installato dentro il computer e non è facile rimuoverlo. Tenere originale e backup nello stesso posto non è molto saggio: se succede qualcosa al computer, dalla sovratensione al furto all'inondazione di caffè, avrete perso tutto. Inoltre lo spazio su disco tende a esaurirsi in fretta: se non usate un disco di backup molto capiente, riuscirete a farci stare tre o quattro copie di backup e basta. Peggio ancora, tenere tutti i backup sullo stesso disco significa che se quel disco si danneggia, avete perso tutto.

La seconda soluzione, quella con il CD masterizzato, ha un costo molto modesto (se avete già un masterizzatore, cosa ormai abbastanza frequente) ed è molto flessibile, però è lenta e ha dei limiti di capienza. Costa poco perché si possono usare i CD riscrivibili, per cui avete la spesa dei CD una sola volta. È flessibile perché potete espandere il numero di backup a vostro piacimento quando vi serve farlo, magari uno per volta per diluire la spesa nel tempo. Cosa più importante, i CD sono rimovibili e quindi possono essere conservati in un luogo sicuro.

Per contro, la velocità di scrittura di un CD è decisamente inferiore a quella di un disco rigido, per cui è probabile che un backup richieda più di mezz'ora. Inoltre, dato che i CD hanno un limite di capienza di circa 640 megabyte, se i file di cui dovete effettuare backup superano complessivamente questo limite è necessario comprimerli prima oppure spezzare il backup su più CD, moltiplicando il tempo necessario.

Il mio trucco preferito è il backup incrociato: se avete due computer di capienza grosso modo uguale, potete collegarli in rete locale e fare il backup di uno sull'altro. Questo sistema ha il pregio di essere più veloce del backup su CD e di consentire di tenere copia e originale su due computer diversi. Se uno dei due è un portatile, è facile tenere copia e originale in due posti fisicamente diversi (a casa e in ufficio, ad esempio).

A voi la scelta di come fare il backup. L'importante è che lo facciate.

Strumenti di backup

Esiste una caterva di programmi specificamente dedicati al backup sotto Linux, ciascuno con le sue regole, le sue peculiarità... e il suo bravo manuale. Per questo motivo non ha senso che io vi spieghi il funzionamento di un programma specifico: posso darvi una panoramica dei concetti di base comuni a tutti i programmi di backup.

Spesso, come vedrete, non è necessario prendersi la briga di usare un programma apposito. Grazie al modo in cui funziona Linux, e diversamente da quanto avviene con Windows, la semplice copia con `kfm` o con il comando `cp` è normalmente più che sufficiente. Linux include comandi appositi per il backup, come `cpio`, `tar` e il già citato `gzip`, che creano archivi compressi, ma dovete studiarvi i loro parametri e le loro righe di comando, e sto cercando di risparmiarvi quest'incombenza.

Nulla vi vieta, ovviamente, di approfondire in seguito le tante opzioni di backup disponibili. Per ora quello che conta è che cominciate immediatamente a salvare il vostro lavoro di impostazione di Linux.

Backup dei dati

Il lavoro sotto Linux tende a essere molto più disciplinato di quello sotto Windows, semplicemente perché l'utente normale può scrivere soltanto in directory specifiche e quindi non può spargere file per ogni dove. Tipicamente, l'utente ha accesso alla propria home directory e alle directory condivise (ad esempio */tmp*).

Questo significa che salvare i dati generati dagli utenti richiede semplicemente la copia della directory */home* e delle sue sottodirectory.

Backup incrementale (quello svelto)

Sentirete spesso parlare di backup *incrementale*. Funziona così: si fa un grande backup generale, diciamo una volta la settimana, e poi si fa giornalmente il backup esclusivamente dei file modificati quel giorno. Finita la settimana, si fa un nuovo backup generale e si ricomincia.

Questo rende molto veloce il backup quotidiano, ma richiede ordine e disciplina di ferro, e so che non tutti sono tagliati per queste cose (io per primo). Il backup incrementale è l'approccio adottato molto spesso nelle aziende, dove c'è una persona addetta a queste cose, ma non credo sia uno strumento pratico per l'uso individuale. Visto che presumibilmente la mole di dati che generate ogni giorno non è immensa, una normale copia giornaliera dell'area dati è probabilmente altrettanto veloce e molto più facile da gestire. Insomma, sappiate che esiste, ma non usatelo.

Backup completo

Se avete esperienza di backup sotto Windows, saprete che il modo migliore per avere una copia di sicurezza dal funzionamento garantito è un backup completo sotto forma di *immagine della partizione*. Questo tipo di backup non copia semplicemente tutti i file, ma prende l'intero contenuto della partizione e lo duplica bit per bit (o quasi). Se si può paragonare la normale copia dei file a una fotocopia, l'immagine della partizione è come una fotocopia che riproduce persino lo stesso tipo di carta e di inchiostro dell'originale.

Questo tipo di backup piuttosto particolare è reso necessario da una particolarità di Windows che Linux non ha: in Windows ci sono file che devono assolutamente occupare posizioni fisiche ben precise sul disco, altrimenti il sistema operativo non funziona. Infatti se provate a copiare (con la normale copia di Esplora Risorse) un Windows già installato da un disco da 4 GB a un altro disco da 10 GB e poi installate il 10 GB al posto del 4 GB, Windows non funzionerà. Ci vuole l'immagine della partizione, realizzata con programmi commerciali come il *Ghost* di cui ho già parlato.

Venendo dal mondo Windows, quindi, avrete due abitudini: la prima è che per fare il backup completo del computer ci vuole un programma apposito che non è gratuito. La seconda è che l'unico modo di avere un backup funzionante è creare l'immagine della partizione.

Sono due abitudini che per fortuna potete perdere in Linux. Non avendo file obbligati ad occupare punti precisi del disco, è sufficiente fare la semplice copia di tutti i file e di tutte le directory che stanno sotto la root directory, e non occorre comperare un programma apposito.

È ovviamente inutile copiare il contenuto della directory */mnt*, dato che non contiene solo file ma soltanto punti di mount (directory vuote), quello della directory */tmp* (dedicata ai file temporanei "usa e getta") e quello della directory virtuale */proc*. Se volete snellire i backup e non avete gravi esigenze di sicurezza, potete anche sacrificare il backup di */var/log*, la directory che contiene i file di log.

Il ripristino è altrettanto semplice: si ricopiano i file desiderati oppure, in caso di azzeramento completo di Linux, si avvia dal floppy di boot e si ritrasferisce tutto dal backup al disco originale. Se persino il Master Boot Record è stato azzerato,

dovrete eseguire **lilo** per consentire l'avvio dal disco rigido.

Se avete fretta di arrivare al sodo, vi dico come fare il backup di una directory di Linux racchiudendolo in un unico file, chiamato tecnicamente *archivio*, con questo comando:

- **tar -cf** (**c** sta per "crea un nuovo archivio", **f** consente di specificare il dispositivo di backup);
- seguito dal nome del dispositivo su cui volete scrivere il backup e dal nome del file che deve contenere il backup (ad esempio, se il dispositivo è il secondo disco rigido e il file è *pippo.tar*, scrivete **/dev/hdb1/pippo.tar**);
- seguito dal file o dalla directory di cui volete creare il backup.

Il risultato finale è una riga di comando del tipo **tar -cf /dev/hdb1/pippo.tar /home**, che crea un file di nome *pippo.tar* sul secondo disco rigido e in questo file archivia il contenuto della directory */home*. Usate questa riga di comando per ciascuna delle directory contenute nella root directory (tranne le eccezioni già citate) e avrete un perfetto backup di Linux.

Se volete un backup compresso, potete aggiungere il parametro **z**. Per ripristinare tutto dal backup creato con **tar**, invece, il comando è **tar -xf** seguito dal nome del file di archivio. Se volete estrarre un singolo file dal backup di **tar**, invece, specificatene il nome alla fine del comando precedente. Il contenuto di un archivio si elenca digitando **tar -tvf** seguito dal nome dell'archivio e da **| less** per permettervi di sfogliare l'elencazione sullo schermo.

Backup di Linux, versione light

Gran parte della struttura di directory di Linux non viene mai modificata ed è quindi inutile eseguirne il backup. Vi conviene concentrare i vostri sforzi sulle directory di configurazione (*/etc* e */usr/X11R6/lib/X11*) e su quelle degli utenti (*/home*) e reinstallare il resto dal CD di installazione.

Adirittura, in molti casi la soluzione più rapida a un grave sconquasso del sistema operativo è reinstallare da capo Linux e poi ricopiare dal backup i file di configurazione e quelli degli utenti. In entrambi i casi, le dimensioni di un backup essenziale come questo sono davvero modeste e vi consentono di eseguire backup giornalieri (sempre usando **tar**) senza occupare quantità di spazio esagerate.

Backup... su carta

Vi conviene annotare sempre su carta le modifiche che apportate a Linux: le applicazioni che installate o rimuovete, i file di configurazione che modificate, e così via. Dico "su carta" perché se scrivete le annotazioni sul computer, non potrete leggerle in caso di crisi, cioè proprio quando ne avete il massimo bisogno.

Inoltre quando installate o rimuovete un'applicazione, usate i dettagli offerti da *kpackage* per scoprire quali directory verranno alterate dall'installazione o rimozione, e fate il backup soltanto di quelle.

Duplicare un floppy di boot e simili

Durante l'installazione di Linux avete creato un dischetto di boot per l'installazione (con il programma per DOS *rawrite.exe*) e un dischetto di boot normale. Nel corso dei capitoli precedenti avete probabilmente creato anche altri floppy, ad esempio il dischetto che fa da "chiave di avviamento", quello di parted, e così via.

Molti di questi floppy sono vitali per il funzionamento di Linux o per il suo ripristino in caso di problemi. Dipendere da un'unica copia di questi dischetti, però, non è saggio. Se li smarrite o se si danneggiano, siete davvero in braghe di tela.

Per copiare esattamente un dischetto di quelli citati, usate questa sequenza di comandi, accessibile anche all'utente comune:

- inserite il dischetto da duplicare nel drive;
- digitate **dd if=/dev/fd0 of=** seguito da un nome di file. Questo comando legge il dischetto contenuto nel dispositivo `/dev/fd0` e ne crea un'immagine in un file a cui impone il nome specificato nel comando. Ad esempio, **dd if=/dev/fd0 of=parted.img** crea un'immagine del dischetto in un file di nome `parted.img`;
- rimuovete il dischetto originale e inserite quello su cui volete scrivere la copia;
- digitate **dd if=** seguito dal nome di file usato per il comando precedente e da **of=/dev/fd0**. Ad esempio, se avete usato il nome `parted.img`, il comando diventa **dd if=parted.img of=/dev/fd0**;
- al termine, cancellate dal disco rigido il file che contiene l'immagine (nell'esempio `parted.img`).

Quando l'interfaccia grafica fa le bizze

Capita ogni tanto che Linux vero e proprio funzioni egregiamente mentre l'interfaccia grafica va periodicamente in *crash* oppure rallenta paurosamente la macchina. I motivi di questo comportamento possono essere tanti, ma una cosa è certa: in attesa che arrivino soccorsi, ci vuole un modo per evitare l'avvio dell'interfaccia grafica.

Ci possono essere inoltre varie situazioni in cui l'interfaccia grafica non vi interessa affatto, ad esempio perché usate soltanto programmi Linux non grafici. In questo caso vi conviene disabilitare l'avvio automatico dell'interfaccia grafica, liberando memoria e sveltendo il computer.

Se volete una disattivazione manuale *una tantum*, potete digitare **linux 3** all'avvio, quando compare la richiesta di LILO. In Red Hat Linux 7.0, per far comparire la richiesta di LILO dovete digitare **Ctrl-X** nella schermata grafica in cui scegliete fra Windows e Linux.

Se preferite rendere permanente questa disattivazione, modificate il file `/etc/inittab` alla riga che inizia con `id`: sostituite la cifra 5 con la cifra 3. Questi numeri indicano il *runlevel* di Linux, che è una cosa che ho fatto di tutto per risparmiarvi (e ci sono riuscito, con quest'unica eccezione).

In entrambi i modi, l'avvio di Linux si ferma all'interfaccia di testo. Per avviare l'interfaccia grafica è sufficiente digitare **startx**. Per riattivare l'avvio automatico dell'interfaccia grafica, modificate di nuovo il file `/etc/inittab`, rimettendo un 5 al posto del 3.

```

# inittab      This file describes how the INIT process should set up
#             the system in a certain run-level.
#
# Author:     Miguel van Smeerenburg, <miguels@drinkel.nl.suget.org>
#             Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Default runlevel. The runlevels used by RPM are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:

```

Figura 19-4. Modificare l'avvio di Linux.

A caccia di elefanti

Una delle attività di manutenzione più importanti è la ricerca ed eliminazione dei file e delle directory che riempiono inutilmente il disco rigido. Tipicamente si tratta di file di log, che per loro natura crescono continuamente, o di file *core*

generati automaticamente quando un'applicazione non termina correttamente. Ci sono anche i file creati sbadatamente dagli utenti e le copie di backup obsolete.

Visto che i dischi rigidi odierni contengono tipicamente migliaia di file e centinaia di directory, conviene concentrare la propria attenzione sugli esemplari più corpulenti. Per avere un elenco delle directory che occupano più spazio, digitate questo comando:

```
du -m -S -x / | sort -n -r | less
```

Questo produce una risposta nel formato mostrato in Figura 19–5, dove le directory più ingombranti sono elencate per prime e le loro dimensioni sono espresse in megabyte.



```

root@sleepspace3: root
File Sessions Options Audo
125 /usr/bin/office52/program
108 /usr/lib
78 /usr/bin/office52/help/39
28 /usr/lib/netescape
11 /usr/doc/qt-clevel-2.1.0/val
11 /usr/i18n/lib
11 /usr/i18n/bin
10 /lib
9 /usr/share/texF/web2c
8 /usr/lib/rpm
8 /usr/bin/office52/share/gallery
7 /usr/bin/office52/share/dict
6 /usr/share/emacs/20.5/etc
6 /usr/man/man1
6 /usr/lib/netescape/java/classes
6 /usr/lib/ispell
6 /usr/lib/gcov
6 /usr/info
6 /usr/doc/qt-clevel-1.48
6 /usr/bin/office52/share/xp3/yoda
6 /usr/bin/office52/share/fonts/type1
6 /usr/share/emacs/20.5/lisp
6 /usr/man/man3
6 /usr/i18n/qt-2.1.0/lib
6 /usr/i486-linux-libc5/lib
6 /usr/bin/office52/program/resource
6 /usr/i18n/lib/xscreensaver
6 /bin
4 /usr/share/i18n/locales
4 /usr/sbin
4 /usr/lib/gshard_5
4 /usr/lib/gcc-11b/i386-redhat-linux/egcs-2.91.66
4 /usr/bin/office52/share/fonts/75dpi/bdf
4 /usr/i18n/man/man3
4 /sbin
4 /lib/modules/2.2.14-12/k1sc
4 /usr/src/linux-2.2.14/include/linux
4 /usr/share/texF/tex/context/base
4 /usr/share/i18n/charmaps
4 /usr/share/fonts/default/type1

```

Figura 19–5. Scoprire le directory più ingombranti.

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.