

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

6. Benvenuti in un mondo strano

Per chi viene dal mondo di Windows, Linux è un ambiente davvero inconsueto. Comincio ad anticiparvi qui alcune delle sorprese e delle principali differenze e analogie fra Windows e Linux che incontreremo strada facendo.

Per questo capitolo non serve impegnare il computer, basta una bella dose di elasticità mentale.

Cominciamo dalle cose importanti

Innanzitutto, facciamo il possibile per non sembrare pivelli. Le prime due cose da fare sono imparare la corretta pronuncia della parola *Linux* e saper riconoscere il simbolo di Linux.

In un *linuxiano* (così si chiama chi usa Linux), vedere che un utente Windows come voi sa cos'è e come si pronuncia Linux e addirittura ne riconosce l'icona suscita commossi sentimenti di immediata fratellanza, che vi torneranno molto utili quando avrete bisogno del suo aiuto.

La corretta pronuncia di *Linux* è, in tutte le lingue, *li-nucs*. Non pensate di fare i poliglotti pronunciandolo "all'inglese", cioè *lai-nacs*. E non fidatevi delle pronunce che sentite scaturire dalla bocca di inglesi e americani: anche molti di loro non sanno qual è la giusta pronuncia e tirano a indovinare.

Veniamo al simbolo: è un pinguino di nome *Tux*. Il suo nome si pronuncia *tùcs* in italiano. Gli americani (e gli italiani aspiranti americani modello Alberto "Uazzamèriga" Sordi) lo chiamano *tacs*, come il telefonino. Facciano pure, per carità, ma voi chiamatelo *tucs*. L'importante è non pronunciarlo *tiucs*.



Figura 6–1. Tux, il pinguino simbolo di Linux.

Si può amare un sistema operativo?

Di solito un utente Windows *detesta* Windows, o nella migliore delle ipotesi lo sopporta o gli è indifferente; gli utenti Linux, per contro, *amano* il loro sistema operativo, anche nelle sue più incoerenti stravaganze.

Gli utenti Linux sono convinti che il loro sistema operativo sia il migliore in assoluto (un po' come ogni seguace di ogni religione è convinto che solo la sua sia la Vera Fede), e non fanno alcuno sforzo per non ostentare questo atteggiamento.

Questo, a prima vista, può sembrare un comportamento borioso, ma in effetti è difficile argomentare il contrario. Se non è il migliore, Linux è obiettivamente fra i migliori sistemi operativi in assoluto.

Allo stesso tempo, questo non significa che gli utenti Linux abbiano la puzza sotto il naso e non vogliano avere a che fare con gli utenti Windows. Il loro disdegno è per Windows, non per chi lo usa; per loro, vedere un utente Windows chiedere di imparare Linux è come vedere un peccatore che si redime e si converte. Vi accoglieranno a braccia aperte, lieti di aver sottratto un'altra anima al "demonio": in altre parole, di aver sottratto un altro utente allo strapotere di Microsoft.

Non denigrate gli utenti Linux per questa loro mania di atteggiarsi a membri del popolo eletto. È tipico di chi ha faticato non poco per andare controcorrente e si volta indietro a guardare chi ancora nuota insieme alla massa, facendosi portare dovunque vada la corrente. Cosa ancora più importante, se imparate a conoscere Linux, ben presto potreste trovarvi anche voi a comportarvi in questo modo.

Libero e gratuito

È disorientante usare un prodotto che non costa nulla ed è liberamente modificabile e distribuibile, dopo che abbiamo passato anni a spendere l'irriducibile per gli aggiornamenti, angosciarci per i controlli antipirateria ("Cesira, c'è l'ispettore della Finanza, dove diavolo hai messo il certificato di licenza di Windows?") e a vederci imporre dall'alto le regole di comportamento del software. Infatti per una delle stranezze della licenza d'uso di Linux, chiunque lo può duplicare e distribuire e non è vietato farsi pagare per farlo.

Mi spiego. Io posso dare una copia di Linux a un amico, a cento amici, o anche disseminarla via Internet senza alcun problema: è perfettamente lecito. Posso addirittura farmi pagare centomila lire, un milione o quanto mi pare per ogni copia distribuita. Se mi va, posso addirittura andare in un negozio, comperare per centomila lire una copia di Linux confezionata da una società come Caldera, duplicarla e poi rivenderne dieci o mille copie al prezzo che mi pare. È tutto legale. Solo io, però, avrò diritto all'assistenza tecnica compresa nelle centomila lire; gli altri si dovranno arrangiare da soli.

Posso anche prendere quella singola copia pagata di Linux e installarla legalmente su dieci, cento o mille computer, e usufruire dell'assistenza tecnica su ciascuno di quei computer, a patto che sia io in persona a richiederla.

Attenzione: questo non vuol dire che tutto il software per Linux sia gratuito. Il sistema operativo in sé lo è, ma le applicazioni possono anche essere a pagamento. Controllate sempre le condizioni d'uso, distribuzione e licenza dei programmi che usate.

Per chi si avvicina all'informatica adesso, l'idea di produrre un programma e poi regalarlo all'umanità può sembrare strana e commercialmente suicida, ma in realtà non è né nuova né originale: risale agli albori dell'informatica personale, quando scrivere un programma e poi distribuirlo liberamente e gratuitamente, rivendicandone al massimo la proprietà intellettuale, era prassi comune.

A questo proposito c'è una celebre lettera, pubblicata nel 1976 sulla *newsletter* degli utilizzatori di Altair (uno dei primissimi personal computer), in cui uno studente diciannovenne sfidava la tradizione informatica dell'epoca e rifiutava il concetto, scontato a quei tempi, di regalare il frutto delle proprie fatiche. Lo studente era Bill Gates, e sappiamo tutti com'è andata.

In questi ultimi anni l'idea di tornare al vecchio modo di distribuire programmi si è riaffacciata, debitamente trasformata e aggiornata, con l'etichetta di *open source*. Molti dei nuovi imprenditori rampanti sono convinti che l'open source sia una novità sfornata dalla *new economy* e l'abbracciano perché fa tendenza e fa notizia (cioè fa pubblicità). Beata ignoranza.

AN OPEN LETTER TO HOBBYISTS	LETTERA APERTA AGLI HOBBISTI
<p>By William Henry Gates III</p> <p>February 3, 1976</p>	<p>di William Henry Gates III</p> <p>3 febbraio 1976</p>
<p><i>To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?</i></p>	<p>Secondo me, la cosa più critica nel mercato hobbistico, in questo preciso momento, è la mancanza di corsi e libri di software validi e di software di qualità. Senza software di qualità e un proprietario che s'intenda di programmazione, un computer hobbistico è sprecato. Ma verrà mai scritto del software di qualità per il mercato hobbistico?</p>
<p><i>Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds \$40,000.</i></p>	<p>Quasi un anno fa, Paul Allen e io, prevedendo un'espansione del mercato hobbistico, abbiamo assunto Monte Davidoff e abbiamo sviluppato l'Altair BASIC. Anche se il lavoro iniziale ci ha richiesto soltanto due mesi, noi tre abbiamo trascorso la maggior parte dell'anno scorso a documentare, migliorare e aggiungere funzioni al BASIC. Ora abbiamo il BASIC 4K, 8K, EXTENDED, ROM e DISK. Il valore del tempo-macchina che abbiamo utilizzato supera i 40.000 dollari.</p>
<p><i>The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however, 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent on Altair BASIC worth less than \$2 an hour.</i></p>	<p>Le reazioni che abbiamo avuto dalle centinaia di persone che dicono di usare il BASIC sono state tutte positive. Tuttavia sono evidenti due fatti sorprendenti: 1) la maggior parte di questi "utenti" non ha mai acquistato il BASIC (meno del 10% di tutti gli utenti Altair ha acquistato il BASIC); 2) se si considera l'ammontare delle royalty risultanti dalle vendite agli hobbisti, il tempo che abbiamo dedicato all'Altair BASIC vale meno di due dollari l'ora.</p>
<p><i>Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?</i></p>	<p>Come mai? Come la maggior parte degli hobbisti non può non sapere, la maggioranza di voi ruba il software che usa. L'hardware va pagato, ma il software è una cosa da condividere. A chi importa se la gente che ci ha lavorato viene pagata?</p>
<p><i>Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.</i></p>	<p>È giusto tutto questo? Una cosa che di certo non riuscite a fare rubando il software è vendicarvi della MITS per qualche problema che avete forse incontrato. La MITS non guadagna dalla vendita di software. La royalty pagata a noi, il manuale, il nastro e i costi amministrativi la rendono un'operazione che chiude in pareggio. Una cosa che invece riuscite a fare è impedire che venga scritto del buon software. Chi può permettersi di svolgere gratuitamente un lavoro di qualità professionale? Quale hobbista può dedicare 3 anni-uomo alla programmazione, al debug, alla documentazione del prodotto e poi distribuirlo gratis? Il fatto è che nessuno, a parte noi, ha investito molto denaro nel software hobbistico. Abbiamo scritto il BASIC 6800 e stiamo scrivendo l'APL 8080 e l'APL 6800, ma l'incentivo a rendere disponibile agli hobbisti questo software è molto scarso. Per dirla esplicitamente, quello che voi commettete è un furto.</p>
<p><i>What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.</i></p>	<p>E che dire dei signori che rivendono l'Altair BASIC? Non stanno forse lucrando sul software per hobbisti? Certamente, ma coloro che ci sono stati segnalati alla fine ci potrebbero rimettere. Sono costoro che danno una brutta nomea agli hobbisti e dovrebbero essere cacciati a pedate da qualsiasi raduno di club al quale si presentano.</p>
<p><i>I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write to me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.</i></p>	<p>Sarei lieto di ricevere lettere da chiunque voglia pagare il dovuto o abbia suggerimenti o commenti da fare. Scrivetemi all'1180 di Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nulla mi farebbe più piacere che essere in grado di assumere dieci programmatori e inondare il mercato hobbistico di buon software.</p>
<p>Bill Gates</p> <p>General Partner, Micro-Soft</p>	<p>Bill Gates</p> <p>Socio accomandatario, Micro-Soft</p>

(1) Episodio citato in "Alla sorgente del software libero", Corriere Multimedia, 15/2/1999. Non mi è stato possibile reperire il testo direttamente dalla fonte originale, per cui è possibile che vi sia qualche errore: se ne scoprite, segnalatemeli. La versione pubblicata qui è quella circolante su Internet e verificata con controllo incrociato fra varie fonti. La traduzione è opera mia.

Prima che me lo chiediate, ebbene sì, all'epoca *Microsoft* si scriveva con il trattino. E riconosco che anch'io, a diciannove anni, ho detto e scritto cose anche più avventate di queste. Ma siccome ho avuto il buon gusto di non diventare fantastiliardario, nessuno me le rinfaccia.

Un'osservazione interessante, riguardo ai 40.000 dollari di tempo-macchina utilizzato, è che Bill Gates si ritirò da Harvard proprio in seguito a una disputa in cui era accusato di aver sfruttato i computer dell'università per scopi commerciali personali [fonte: <http://www.boston.com/globe/metro/packages/harvard/partone.htm>].

Un nuovo concetto: la distribuzione

Windows ha un solo fabbricante: Microsoft. Tutto sgorga dalla catena di produzione di un'unica azienda. Linux, invece, proviene da una miriade di fonti diverse. Un pezzo è fabbricato di qua, uno di là, e ognuno è libero di assemblare i pezzi che preferisce.

Un'azienda, un'associazione o persino un privato può creare un *bouquet* specifico di pezzi di Linux che ritiene funzionino bene insieme e poi decidere di distribuirli (gratis o meno). Nel gergo di Linux, questo *bouquet* si chiama *distribuzione*.

In virtù di questa libertà di assemblaggio, che fra l'altro è obbligatoria in base alle condizioni di licenza di Linux, esistono tante fonti diverse per rifornirsi di Linux. È un po' quel che succede per il carburante: di pompe di benzina ce ne sono tante di marche differenti, ma tutte vendono grosso modo lo stesso prodotto, confezionato in varianti leggermente diverse (con o senza l'additivo X oppure con e senza self-service, ad esempio).

Le distribuzioni di Linux prendono il nome dalle aziende o associazioni che le assemblano. Ad esempio, questo libro è dedicato a Red Hat Linux, ossia al *bouquet* di pezzi di Linux assemblato dalla società americana Red Hat Inc., ma ci sono moltissime altre fonti di distribuzione, con nomi come *Mandrake*, *Suse*, *Corel*, *Slackware*, *Debian*, eccetera.

L'esistenza di tante fonti indipendenti di Linux non comporta necessariamente confusione e incompatibilità. Infatti tutte le varianti di Linux in circolazione contengono lo stesso *bouquet* di base, cioè le parti essenziali di Linux, e si differenziano soltanto negli accessori. Questo vuol dire che un programma scritto per Linux funziona con qualsiasi variante recente di Linux.

Quale distribuzione scegliere?

La differenza fondamentale fra le varie distribuzioni è costituita dai pezzi di Linux che vengono inclusi. Ad esempio, i CD di alcune distribuzioni potrebbero non contenere i codici sorgente di Linux, che però per legge devono essere disponibili almeno via Internet presso il sito del distributore.

Tuttavia anche i programmi accessori sono molto importanti. Ad esempio, spesso il programma di installazione di Linux spesso è scritto appositamente dalla società distributrice, e lo è anche il programma che gestisce l'installazione delle applicazioni. Ci sono programmi di installazione grafici, molto accattivanti e facili da usare, ma ci sono anche programmi basati su un'interfaccia di puro testo, più universali ma mentalmente più impegnativi.

Va a gusti, ma installare Linux o un'applicazione usando un programma ostico e scontroso, magari neppure tradotto in italiano, è in genere una vera pena. Questo è uno dei motivi per cui abbiamo scelto la distribuzione Red Hat: i suoi programmi di installazione sono piuttosto ben fatti, grafici e molto intuitivi, e soprattutto sono in italiano.

Gli utenti Linux fanno della scelta della distribuzione una guerra di religione. Red Hat è considerata la distribuzione per dilettanti, Debian o Slackware quella per professionisti, e così via. Ma sotto sotto, tutte le distribuzioni usano gli stessi pezzi fondamentali, per cui non è il caso di lasciarsi trascinare in queste polemiche. Non sentitevi inferiori perché usate

Red Hat.

Edizioni numerate

Ogni versione di Windows è un blocco unico: il CD di Microsoft contiene tutti i componenti scritti su misura per quella versione. Solitamente, il successivo aggiornamento di un componente di Windows viene distribuito soltanto in occasione dell'uscita di una nuova versione dell'intero sistema operativo. Tutti i componenti di Windows, in altre parole, crescono di pari passo e marcano insieme. Una determinata versione di Windows "fotografa" un momento preciso dello sviluppo del sistema operativo.

Linux usa un sistema completamente diverso. Ogni suo pezzo cresce e si evolve separatamente e con ritmi diversi rispetto agli altri, e viene distribuito quando è pronto invece di aspettare la versione successiva dell'intero sistema operativo.

Questo ha diverse conseguenze molto interessanti. Windows procede a balzi, e ogni nuova versione viene annunciata come un grande evento (fra una versione o l'altra vengono distribuite le *patch* o *service pack*, ossia le correzioni agli errori presenti nella versione in vendita, ma si tratta di rattoppi, non di ristrutturazioni serie). L'evoluzione di Linux, invece, procede a ritmo costante. Versioni nuove dei suoi componenti possono uscire a distanza di qualche settimana l'una dall'altra. Il nucleo di Linux, il cosiddetto *kernel*, viene distribuito talvolta con cadenza addirittura quotidiana.

Siccome i componenti di Linux sono liberamente prelevabili da Internet, le sue migliorie sono immediatamente disponibili a tutti e non c'è bisogno di attendere il rilascio dell'intero sistema operativo come avviene con Windows. E, ovviamente, non c'è nulla da pagare per gli aggiornamenti.

Per carità, non abbiate paura che questo fluire incessante di aggiornamenti diventi un onere angosciante. Non è obbligatorio installare tutte le nuove versioni dei componenti di Linux. Se quelle che avete funzionano, non c'è motivo di cambiarle. Inoltre, per il modo in cui è stata concepita la struttura di Linux, gli aggiornamenti si fanno senza dover reinstallare tutto da capo. Si preleva la versione nuova del componente, la si installa nel Linux esistente, e si riprende il lavoro.

Un'altra conseguenza di questa situazione è che ogni componente di Linux ha un numero di versione diverso. Ad esempio, il kernel può essere la versione 2.2.14, l'interfaccia grafica può essere la versione 1.1.2, e così via, senza che questo pregiudichi il funzionamento del sistema. In Windows, invece, tutti i componenti prendono il numero di versione del sistema operativo a cui appartengono.

Niente di nuovo sotto il sole

Perché Linux non usa l'anno di produzione per identificare la versione, come fa Windows? La domanda è legittima, ma è l'approccio che è sbagliato. La numerazione usata da Linux non è un parto della fantasia bacata di un linuxiano eccentrico: è la norma in informatica, tant'è vero che un tempo anche Windows la seguiva.

Infatti Windows ha avuto versioni numerate 1.0, 2.0, 3.0, 3.1, 3.11 e doveva avere anche una versione 4.0. Poi è intervenuto il reparto marketing e ha avuto la trovata commerciale di appioppare le annate al posto dei numeri di versione, per cui la 4.0 è diventata 95. In questo modo venite indotti a sentirvi all'antica se usate ancora Windows 95 nel 2000, quasi come se il software avesse la data di scadenza al pari dello yogurt, e c'è una maggiore pressione psicologica a comperare la nuova versione.

Ci sono ancora tracce della numerazione standard in Windows. Nella versione 95, se andate nel Pannello di Controllo e cliccate due volte su Sistema, otterrete la dicitura *Microsoft Windows 95 – 4.00.950 B* o qualcosa di simile. Windows 98 Seconda Edizione italiano, ad esempio, risulta essere la versione 4.10.2222A o giù di lì.

Fra l'altro, numerare i programmi o i sistemi operativi per annate ha un difetto: ne potete pubblicare una sola versione l'anno, altrimenti siete costretti a chiamarle, che so, *2000A* e *2000B*, e allora tanto vale tornare al sistema di numerazione tradizionale.

Infatti Microsoft si è recentemente impegnata in questo senso. Da tempo esistono due edizioni parallele di Windows: una "per consumatori", chiamata *Windows* e basta, e una "professionale", chiamata *Windows NT*. Nei primi anni del loro sviluppo entrambe le edizioni erano numerate separatamente, per cui c'era un Windows 3.1 accanto a un Windows NT 3.5.

Poi la versione per consumatori ha adottato le annate, e così sono arrivati Windows 95 e Windows 98. Ma di recente anche la versione professionale si è unita alla festa, ed è stata battezzata *Windows 2000*. Quindi Windows 2000 *non* è il successore di Windows 98, come parrebbe invece logico supporre dal nome. Il che pone un problema: come chiamare l'edizione 2000 di Windows per consumatori? I geni del reparto marketing hanno tirato fuori il jolly: *Windows Millennium Edition*, abbreviato in Windows ME. Chiaro, no?

In Linux, inoltre, vige la consuetudine di distribuire contemporaneamente sia versioni "sperimentali" dei componenti, sia versioni stabili e ben collaudate. Per aiutare gli utenti a distinguerle, le versioni sperimentali terminano con un numero dispari e quelle stabili con un numero pari.

Numeri e ancora numeri

Per confondere un po' le cose, anche le distribuzioni di Linux hanno i loro bravi numeri di versione. I distributori di Linux pubblicano periodicamente versioni aggiornate della loro confezione di Linux, e per distinguerle le numerano: abbiamo quindi, ad esempio, una distribuzione chiamata *Linux Red Hat 7.0*. Traduzione: questa è la versione 7.0 della distribuzione di Linux curata dalla società Red Hat.

Questo numero di versione della distribuzione non ha nulla a che vedere con quelli dei componenti di Linux contenuti nella distribuzione stessa; inoltre, siccome i vari distributori non sono sincronizzati, non è detto che un numero di distribuzione più alto contenga necessariamente un Linux più aggiornato. Ad esempio, una distribuzione *Caldera 3.0* potrebbe essere più aggiornata rispetto a una *Slackware 7*.

Se tutto questo vi sembra un gran caos, avete perfettamente ragione: lo è.

Clic o doppio clic, questo è il dilemma

Una delle cose che inviperisce maggiormente chi sta cercando di imparare Windows è la storia del clic e del doppio clic. Probabilmente ormai vi siete abituati al fatto che si seleziona un oggetto con un clic singolo e lo si apre con un doppio clic, però un doppio clic sull'iconcina a sinistra nella barra del titolo non apre, ma *chiude* il programma, e un clic singolo sul pulsante di chiusura non seleziona il pulsante ma chiude ancora il programma... eccetera, eccetera, eccetera.

Non sapete la regola generale, ma a furia di sbagliare avete imparato quando ci vuole il clic singolo e quando ci vuole quello doppio. Be', ho buone notizie per voi. In Linux il doppio clic praticamente non esiste. Tutto si fa con una sola cliccata del mouse.

Questo in genere lascia disorientati, perché tipicamente si cerca di lanciare un programma in Linux facendo doppio clic sulla sua icona (come si fa in Windows), con il risultato di farne partire due copie. Nell'equivalente Linux di Esplora risorse, per lanciare un programma basta un clic solo; per aprire una cartella basta un clic solo.

In Windows 98 e ME, in realtà, esiste l'opzione di impostare Windows in modo che basti fare un singolo clic per aprire un

file. Tuttavia non ho ancora incontrato nessuno che la usi: tutti si attengono al modo "tradizionale" di usare Windows, cioè fare doppio clic per aprire i file.

In Linux, se volete semplicemente selezionare un file o una cartella, lo fate con il riquadro di selezione: cliccate al di fuori del nome del file e tenete premuto il pulsante del mouse intanto che trascinate. Si forma un riquadro che potete estendere per contenere il file da selezionare. Al rilascio del pulsante, il file viene selezionato.

Questa differenza ha un risvolto abbastanza divertente. Come si riconosce un utente Windows che sta imparando Linux? Dal fatto che in Windows clicca una sola volta su un nome di file e aspetta invano che il file si apra. Non temete, capiterà anche a voi.

Alziamo il cofano

Abbiamo cominciato a fare conoscenza con alcune delle differenze generali fra Windows e Linux, ma adesso è il momento di dedicarci a un confronto più tecnico. Ho volutamente tralasciato la descrizione delle differenze strettamente tecnologiche dei due sistemi operativi, concentrandomi su quelle che hanno un effetto pratico e tangibile sul modo di lavorare dell'utente.

L'ho già detto, ma lo ripeto per chiarezza: qui come nel resto del libro, se non specifico diversamente, quando scrivo *Windows* mi riferisco alla versione per consumatori del prodotto Microsoft. Quella professionale (Windows NT/2000) è tutt'altra creatura, con prestazioni, difficoltà d'uso e costi completamente differenti.

Multiutenza

Solitamente in Windows si accende il computer e si comincia a lavorare, punto e basta, senza che il computer vi chieda chi siete, dove andate e cosa volete fare. In effetti si può impostare Windows in modo che chieda il vostro nome e un codice di accesso, ma non è obbligatorio e non lo fa quasi nessuno (anche perché la protezione offerta da questa limitazione d'accesso è irrisoria).

In Linux, invece, identificarsi è un obbligo. Prima di iniziare a lavorare, Linux vi chiede nome e codice d'accesso (*password*), in una procedura chiamata *login* (si pronuncia "loghìn"). Se non date il nome giusto e il codice corrispondente, non potete accedere a Linux e ai dati presenti nel computer. In realtà anche questa protezione è scavalcabile, ma farlo è parecchio più impegnativo che in Windows.

Ogni coppia composta da nome e password è chiamata *utenza* o *utente*. Lo scopo fondamentale di questo rito di login è consentire a Linux di sapere in ogni momento chi è ai comandi del computer, in modo da poter offrire una serie di servizi e di accessi su misura per la persona che sta lavorando in quel momento.

Ad esempio, un'utenza può essere impostata in modo da avere accesso limitato al computer, per ridurre al minimo i danni che può combinare chi la usa, o in modo da avere accesso completo per consentire la manutenzione approfondita del computer.

In un ambiente in cui più persone condividono lo stesso computer, si può assegnare a ciascuna persona un'utenza diversa con livelli di accesso differenti: in questo modo si può evitare che una persona inesperta cancelli file vitali per il funzionamento del sistema operativo o che una persona danneggi il lavoro di un'altra. Questo tipo di protezione non è disponibile in Windows.

Questa faccenda delle utenze è utile anche quando la persona che usa il computer è una sola. Per evitare che una cliccata sbadata causi un disastro, è buona consuetudine accedere a Linux usando un'utenza con un livello di accesso basso per il normale lavoro e accedere usando un'altra utenza, con pieni poteri di accesso, soltanto quando si fa manutenzione.

In altre parole, anche se sarete solo voi a usare la macchina Linux, per il vostro bene creerete *due* utenze: una onnipotente e una normale. Quella onnipotente prende abitualmente il nome di *root* (si pronuncia "rut"), ed ha la mansione di amministratore del sistema; quella normale prende il nome che più preferite, ed ha il ruolo di lavoratore addetto alla produzione.

Schizofrenia in agguato

Il fatto di assegnare a una stessa persona non una, ma *due* utenze è ovviamente una grossa complicazione in più rispetto a Windows. Ricordarsi due nomi di accesso e due password è senz'altro una scocciatura, ma vi assicuro che è necessario e utile. Eseguire le normali attività con un'utenza che ha un basso livello di accesso è una delle migliori forme di protezione dagli attacchi dei virus e degli intrusi che bussano alla vostra porta quando vi collegate a Internet o scambiate CD e dischetti.

A furia di usare Linux e frequentare linuxiani, comunque, vi abituerete presto a questo andazzo un po' schizofrenico. Per tenere a bada la confusione, solitamente il linuxiano parla delle proprie utenze in terza persona, chiamandole *utenti*, come se si trattasse di altre persone fisiche ("l'utente *root* può fare quello che vuole, l'utente normale no"), distinte da lui. Questa è la convenzione che troverete nel resto del libro.

Permessi

Un'altra conseguenza della multiutenza di Linux è che ogni singolo file presente sul computer ha un *proprietario*, cioè appartiene a un utente specifico, ed ha una serie di *permessi*, ossia codici che determinano quali utenti possono leggere, modificare o eseguire il file. Questi codici possono dire ad esempio "questo file è di Marco, solo lui lo può leggere", "anche quest'altro file è di Marco, lo possono leggere tutti ma solo lui lo può modificare", e così via. Questa possibilità non è disponibile in Windows.

I file vitali del sistema operativo sono accessibili soltanto alla persona designata come "superutente" *root* (di solito il proprietario del computer, cioè voi).

Gruppi

Oltre agli utenti, in Linux ci sono anche i *gruppi* di utenti. Un file può appartenere simultaneamente a un singolo utente e a un gruppo designato di utenti. Se siete i soli a lavorare sul vostro computer, è difficile che abbiate bisogno di creare gruppi di utenti, perché vi basta avere l'utente *root* e quello normale, ma Linux prepara lo stesso vari gruppi per la propria amministrazione interna, per cui è importante che sappiate che esistono. Troverete dettagli in proposito nel Capitolo 12.

Linux a strati

Una delle differenze più vistose fra Linux e Windows è la parte grafica. In Windows è un componente essenziale e obbligatorio; in Linux è un accessorio facoltativo. In altre parole, quando avviate Windows *dovete* usare l'interfaccia grafica. Certo, potreste avviare il computer in modalità MS-DOS e quindi fare a meno della grafica, ma sareste così limitati in quello che potete fare che non riuscireste a combinare nulla.

Ad esempio, il Registro di Windows, uno dei file di configurazione fondamentali, non è modificabile con un semplice editor di testi: ci vuole un editor apposito, che Windows offre soltanto in versione grafica. Per modificare il Registro, quindi dovete per forza avviare l'interfaccia grafica. Più in generale, praticamente tutte le funzioni di manutenzione e amministrazione del sistema operativo dipendono da programmi che funzionano soltanto nell'interfaccia grafica.

In effetti, è improprio parlare di "modalità" MS-DOS, perché quando usate quest'opzione chiudete Windows e avviate un

sistema operativo separato (MS-DOS, appunto). Si può quindi dire che nonostante la presenza della modalità testuale dell'MS-DOS, Windows dispone esclusivamente di un'interfaccia grafica e non può essere usato diversamente.

Linux è tutt'altra pasta. L'interfaccia grafica è addirittura un componente separato dal sistema operativo vero e proprio, tant'è vero che Linux parte prima in una modalità puramente testuale e poi, soltanto se lo richiedete, attiva l'interfaccia grafica. Molte macchine Linux non avviano mai l'interfaccia grafica perché non ne hanno bisogno.

Tutto quello che potete fare nell'interfaccia grafica è fattibile anche nell'interfaccia testuale (con l'ovvia eccezione dell'attività grafica). I file di configurazione di Linux sono praticamente tutti file di testo semplice, modificabili con un comune editor, e Linux dispone di editor di testi per l'interfaccia testuale.

La separazione fra interfaccia grafica e interfaccia testuale è talmente netta che l'interfaccia grafica è considerata da Linux come un qualsiasi altro programma. All'interno di questo programma "madre" (chiamato X) gira un altro programma, il *window manager*, all'interno del quale girano i programmi che necessitano di usare la grafica. Nel resto di questo libro, per semplicità parlerò di "interfaccia grafica" intendendo sia X, sia il window manager.

Va detto, per onestà, che questo approccio stratificato in effetti riduce le prestazioni grafiche di Linux rispetto a quelle di Windows, in cui l'interfaccia grafica è integrata molto più intimamente nel sistema operativo. Tuttavia è la separazione in strati distinti che consente ad esempio di scegliere fra interfacce grafiche completamente diverse, dai nomi come *KDE* e *Gnome*, e di modificarle e aggiornarle separatamente dal resto del sistema operativo. Questa separazione permette inoltre il pieno controllo a distanza di un computer anche tramite una connessione molto lenta.

Aperto o chiuso?

Fate attenzione a una differenza molto importante di Linux rispetto a Windows: in Linux, quando si chiude l'interfaccia grafica non si chiude il sistema operativo: Linux *continua a funzionare* in modalità non grafica, e i programmi che erano in esecuzione nella parte non grafica di Linux (ad esempio i trasferimenti di file o le connessioni a Internet) continuano a girare indisturbati.

In Windows, invece, chiudere l'interfaccia grafica significa chiudere Windows e spegnere il computer, cessando ogni attività, oppure uscire a DOS, ma comunque interrompere tutti i programmi in esecuzione. Ricordatevi che l'interfaccia grafica è per Linux soltanto un programma come tanti altri.

Come contano gli ingegneri

Si sa che gli ingegneri hanno la fama di vivere un po' fuori dal mondo normale. Evidentemente il gene dell'ingegnere era ben radicato nei creatori di Linux, perché sotto Linux si conta quasi sempre *partendo da zero*.

Faccio un esempio. Se vi chiedo di contare le dita di una mano, senza pensarci due volte toccherete un dito per volta e reciterete "uno, due, tre, quattro, cinque". Se chiedete a un utente Linux di vecchia data di fare la stessa cosa, vi risponderà probabilmente "zero, uno, due, tre, quattro"; eppure se gli chiedete quante dita ha in quella mano, la sua risposta sarà un normalissimo "Cinque!".

Questo principio si applica a (quasi) tutto quello che si può numerare in Linux. Ad esempio, se avete due drive per floppy, il primo sarà il drive *zero* (non il drive 1) e il *secondo* sarà il drive *uno*. E così via.

Il file è tutto, e tutto è un file

In Windows siete abituati a distinguere chiaramente fra dischi rigidi, floppy, modem, stampanti, monitor eccetera, e sapete benissimo che non c'entrano niente con file e cartelle.

In Linux le cose stanno molto diversamente. Dal suo punto di vista, tutti gli elementi che ho elencato sono file. Sì, avete capito bene: il disco rigido, per Linux, è un file; il modem è un file; una cartella è un file; una stampante è sempre un file.

Questo approccio incredibilmente unificato è disorientante all'inizio, ma ha i suoi vantaggi: la gestione di tutti questi elementi avviene nello stesso modo, rendendo molto più snello il funzionamento di Linux. È una questione paurosamente tecnica che non è il caso di affrontare adesso; per il momento vorrei soltanto mettervi in preallarme a proposito di questa stranezza.

Una delle conseguenze di questo modo di operare è che i nomi dei dispositivi, in Linux, sono tutti nomi di file. Ad esempio, il floppy si chiama `/dev/fd0`, il disco rigido non si chiama `C:`, ma `/dev/hda`, e così via. Li vedremo tra un attimo. Prima, però, ci vuole una precisazione che sarà fondamentale per chi viene dal mondo Windows e DOS.

Slash e backslash

Se avete guardato attentamente i nomi di "file" Linux che ho appena citato, avrete notato che contengono il carattere `/`, che in informatica si chiama *slash* (fa rima con *splash*) e nel mondo normale si chiama *barra*. In Linux, questo carattere ha lo stesso ruolo che in Windows è ricoperto invece dal carattere *backslash*, cioè dalla barra rovescia (il carattere `\`).

Ad esempio, una cartella che in Windows viene descritta come `\formaggi\provolone\`, in Linux è descritta come `/formaggi/provolone/`. All'inizio è facile confondersi fra barra rovescia e barra normale, e questa differenza d'uso fra Windows e Linux fa dannare un po' tutti. Se un comando non funziona, una delle prime cose che dovete controllare è la digitazione corretta delle barre.

Perché Linux non segue la convenzione di Windows e usa la barra rovescia? Il mondo sarebbe un pochino più vivibile se non ci fosse da ricordare anche questa differenza. In realtà il problema è da capovolgere. È *Windows* che non rispetta lo standard. Infatti UNIX, il sistema operativo da cui deriva Linux, era in giro molto, molto tempo prima che arrivasse Windows, e usava le barre normali (e le usa tuttora). Se ci fate caso, anche Internet usa le barre normali dappertutto, proprio perché Internet è nata su macchine UNIX. Il DOS e Windows si sono uniti tardi alla festa e adesso pretendono di dirigerla.

Nomi nuovi per cose già viste

Linux usa convenzioni molto diverse da quelle di Windows per identificare molti componenti del computer. Ecco un po' di tabelle che riepilogano i nomi dei componenti principali.

Porte seriali

Nome Windows	Nome Linux
COM1	/dev/ttyS0
COM2	/dev/ttyS1
COM3	/dev/ttyS2
COM4	/dev/ttyS3

Porte parallele

Nome Windows	Nome Linux
LPT1:	/dev/lp0
LPT2:	/dev/lp1

Floppy

Nome Windows	Nome Linux
A:	/dev/fd0
B:	/dev/fd1

CD-ROM

Nome Windows	Nome Linux
Primo CD-ROM: una lettera da D: in poi	/dev/hdc
Secondo CD-ROM: una lettera da E: in poi	/dev/hdd

Dischi rigidi

I dischi rigidi seguono delle regole troppo diverse in Linux e in Windows, per cui non è possibile metterle in forma di semplice tabella di equivalenze. Ci vogliono degli esempi.

Immaginate di avere un unico disco rigido contenente una singola partizione. In questo caso, per Windows quella partizione è C:. Per Linux, la partizione è invece */dev/hda1*, dove:

- la *h* indica che si tratta di un disco rigido IDE (nel caso di un disco rigido SCSI sarebbe *s*);
- la *a* indica che si tratta del primo disco rigido del computer (per il secondo disco la lettera diventa *b*, per il terzo diventa *c*, eccetera);
- la cifra 1 è un numero progressivo che indica che si tratta della prima partizione di quel disco rigido (la seconda sarebbe *hda2*, la terza *hda3*, e così via).

Adesso immaginate di avere ancora un unico disco rigido, ma diviso in due partizioni. Per Windows, la prima partizione diventa C: e la seconda diventa D:; per Linux, la prima partizione è */dev/hda1* e la seconda è */dev/hda2*.

Altro esempio: due dischi rigidi, entrambi con una partizione a testa. Per Windows, la partizione del primo disco è C: e la partizione del secondo è D:. Per Linux, la partizione del primo disco rigido è */dev/hda1* e la partizione del secondo disco è */dev/hdb1*.

Il vantaggio del sistema di Linux è che le partizioni non cambiano mai nome anche se aggiungete o togliete un disco rigido o ripartizionate quello esistente. Ad esempio, considerate un normale computer che usa Windows: il disco rigido è C: e il CD-ROM è D:.

Se ripartizionate il disco e lo dividete in due partizioni, la lettera D: che era del CD-ROM viene riassegnata alla seconda partizione del disco rigido. Di conseguenza, tutti i programmi che si aspettano che il CD-ROM sia D: non sanno più che pesci pigliare e vanno reinstallati o riconfigurati. Su macchine Windows con due o più dischi rigidi, ciascuno con più partizioni, le lettere vengono riassegnate secondo uno schema non lineare e assurdamente complesso.

Se invece ripartizionate un disco in una macchina Linux, i nomi delle partizioni non cambiano: il disco rigido acquista semplicemente una partizione in più, chiamata */dev/hda2*, ma il CD-ROM continua ad essere */dev/hdc*.

Altri nomi per cose appena viste

Non basta doversi imparare tutta una serie di nomi nuovi per i dispositivi collegati al computer; bisogna impararne *due*. Infatti a ciascuno dei dispositivi di memoria di massa (floppy, dischi rigidi, CD-ROM) è associato anche un altro nome, che è quello che si usa nella normale interazione con Linux. Questi secondi nomi si chiamano *mount point*, ossia "punti di montaggio".

Ad esempio, il drive per i floppy è */dev/fd0*, ma quando dovete sfogliarne il contenuto vi rivolgete al drive usando il suo secondo nome, che è */mnt/floppy*. Il lettore di CD-ROM, vale a dire */dev/hdc*, prende il secondo nome di */mnt/cdrom*. Tecnicamente, si dice che il mount point di */dev/hdc* è */mnt/cdrom*. Per complicare ulteriormente le cose, in Red Hat Linux 6.2 il lettore di CD-ROM ha anche un *terzo* nome, che è */dev/cdrom*.

La partizione del disco rigido che contiene Linux (in genere */dev/hda2*, se Linux convive con Windows) viene invece gestita chiamandola semplicemente */*, che per convenzione si chiama *root* (niente a che fare con l'utente *root* già visto).

Sempre a proposito di nomi, c'è un'altra particolarità da imparare. Le cartelle in Linux non si chiamano *cartelle*: si chiamano *directory* ("dai-réctori"). In Windows, una cartella contenuta in un'altra cartella è una *sottocartella*, mentre in Linux si chiama, per simmetria, *sottodirectory*.

Addio alle lettere

Come avrete notato dalla descrizione dei nomi usati da Linux per i dischi rigidi, in Linux non esistono le lettere dei drive, come C:\ o D:\, a cui siete abituati in Windows. In realtà Linux considera tutto l'insieme dello spazio disponibile sui vari dischi, dischetti e CD-ROM, locali e connessi in rete, come un'unica grande struttura di directory, senza spezzarla in drive distinti.

Al livello più alto di questa struttura c'è la directory principale o *root directory*, quella contrassegnata dal simbolo */*, che in genere risiede sul primo disco rigido del computer. Tutte le altre unità di memoria di massa, compresi gli eventuali altri dischi rigidi, i floppy e i lettori di CD-ROM, vengono visti da Linux come sottodirectory della directory principale.

È a questo livello che entrano in gioco i "secondi nomi" dei dispositivi di memoria, cioè i mount point. I mount point sono fisicamente delle sottodirectory della root directory, e Linux vede i dispositivi di memoria attraverso queste sottodirectory, come se il contenuto dei dispositivi fosse effettivamente residente in esse. È un concetto piuttosto astratto, e non vi preoccupate se non lo capite al volo: la pratica renderà tutto molto più chiaro.

Questo criterio dell'unica, grande struttura di directory vale persino per dischi rigidi situati su altri computer in una rete locale o addirittura per file situati all'altro capo di una connessione via Internet (non nel normale accesso via Web, ma nel caso di una condivisione di file e directory effettuata attraverso Internet).

In effetti anche Windows cerca di approssimare questo approccio. L'icona *Desktop* in Esplora risorse raggruppa sotto di sé tutti i dischi rigidi del computer e le risorse di rete, ma è una convenzione grafica più che un modo di operare effettivo del sistema operativo.

L'importanza delle presentazioni: mount

In Linux, qualsiasi dispositivo di memoria di massa (disco rigido, floppy, CD-ROM, Zip, eccetera), per essere visibile al sistema operativo, deve essergli presentato formalmente, altrimenti Linux lo ignora. Anche se il drive è fisicamente

collegato al computer, per Linux è come se non esistesse.

La presentazione formale avviene tramite un apposito comando (chiamato *mount*), che specifica il tipo di dispositivo e il modo in cui va utilizzato: ad esempio, si può dire a Linux "guarda, qui c'è un floppy, lo puoi leggere ma non scrivere", oppure "eccoti una partizione Windows, leggila e scrivila pure".

Una volta fatte le presentazioni, Linux comunica con il dispositivo senza ulteriori problemi e lo protegge come se fosse roba sua finché date il comando di rilascio, *umount*, che in sostanza dice a Linux "fai ciao a questo dispositivo, scrivici eventuali dati ancora in sospeso e liberalo". Queste due operazioni si chiamano abitualmente *montare* e *smontare*.

Se volete una spiegazione (leggermente) più tecnica, il montaggio è l'operazione che associa il nome "vero" di un dispositivo (ad esempio */dev/fd0*) con un secondo nome a scelta (ad esempio */mnt/floppy*, ma potreste anche scegliere */mnt/dischetto* o */mnt/pizzamargherita*).

La conseguenza di questo modo di operare è che non basta infilare un CD-ROM o un floppy nel computer per rendere visibile al sistema operativo il suo contenuto: bisogna dare il comando *mount*. Persino le partizioni in cui risiede Linux gli vengono presentate in questo modo durante l'avvio: la presentazione avviene in automatico grazie a parametri memorizzati nel file *fstab*, che elenca le partizioni e i dispositivi da attivare e indica le corrispondenze fra nomi dei dispositivi e loro mount point.

Allo stesso modo, non si può semplicemente rimuovere un floppy o un CD dal drive. Si deve dare il comando di rilascio *umount*, altrimenti Linux si arrabbia perché gli togliete i file da sotto il naso. Nel caso dei CD, addirittura, non è fisicamente possibile estrarre il disco dal lettore: Linux disabilita il pulsante di espulsione. Si ha proprio l'impressione che la macchina infernale abbia conquistato il potere e sia dotata di vita propria. Le espressioni di panico dei novellini di Linux le prima volte che incappano in questa particolarità sono sempre molto divertenti.

Che cosa stupida. Perché mai bisogna fare questa trafila di *mount* e *umount* ogni volta che si infila un CD? In Windows non ce n'è bisogno: quando inserite un floppy o un CD-ROM, ne potete visualizzare subito il contenuto. Anzi, c'è persino una comoda funzione *autoplay*, per cui basta inserire un CD audio perché Windows cominci a suonarlo. Meglio ancora, in Windows c'è l'*autorun*: infila un CD-ROM e il programma che contiene viene eseguito automaticamente. Un classico esempio è dato dai CD-ROM autoinstallanti per l'accesso a Internet. Comodissimo.

Già: comodissimo anche per gli intrusi. In Windows, qualsiasi programma presente sul CD-ROM e specificato nel file *autorun.inf* sul CD-ROM verrà eseguito automaticamente, senza alcun controllo preliminare. E se quel programma fosse un virus? Uno dei trucchetti preferiti dei buontemponi è scrivere un CD-ROM autoinstallante che installa Back Orifice (un potentissimo strumento di sorveglianza e manutenzione a distanza, utilizzabile anche per scopi meno ortodossi). Basta infilare il CD-ROM nel computer bersaglio per prenderne completamente il controllo, *anche se il computer è protetto* (si fa per dire) *dallo screen saver con la password*.

Ci sono vari altri motivi per cui Linux vi assoggetta all'onere di montare e smontare i dispositivi. Uno dei più validi è evitare quello che succede spesso in Windows: inserite un CD-ROM o un floppy per usarlo con un'applicazione (ad esempio un'enciclopedia o un dischetto per salvarvi sopra un documento), poi sbadatamente rimuovete il dispositivo prima di aver chiuso l'applicazione. Risultato: schermo blu di errore se vi va bene, schermo blu della morte (crash di Windows) se vi va male.

Va detto che è comunque possibile attivare la funzione *automount*, che scavalca l'impostazione normale di Linux e lo rende più simile a Windows, per cui un floppy o un CD-ROM infilati vengono immediatamente visti (ma *non* eseguiti) da Linux. Tuttavia, dato che si tratta di un'opzione che indebolisce la sicurezza di Linux, non è completamente attivata nelle installazioni standard (funziona soltanto se cliccate sull'icona del desktop grafico che rappresenta il dispositivo).

Nomi dei file

Alcuni di voi, utenti Windows sin dalle prime versioni (la mitica 3.1 dei primi anni novanta, che fu la prima ad avere grande successo commerciale), o utenti MS-DOS dall'alba dei tempi, ricorderanno che per anni chi usava questi sistemi operativi è stato obbligato a rispettare la regola "otto punto tre". In altre parole, i nomi dei file potevano essere composti da un massimo di otto caratteri, seguiti facoltativamente da un punto e da altri tre caratteri (chiamati *estensione*).

Problemi di lunghezza

Ad esempio, *aldogiovannigiaco.doc* non era un nome accettabile per un file. Gli utenti erano obbligati ad inventarsi abbreviazioni e contrazioni facilmente ricordabili (che però immancabilmente si dimenticavano due giorni più tardi), come *algiogia.doc*. Nel frattempo, gli utenti Mac e Unix si rotolavano per terra dal ridere di fronte alle acrobazie ortografiche dei loro malcapitati colleghi utenti DOS e Windows. Già, perché chi usava il Mac o computer Unix non aveva queste limitazioni.

Gli utenti MS-DOS e Windows, in realtà, non le hanno mai perse: tuttora questi sistemi operativi devono sottostare alla regola "otto punto tre", ma usano un *escamotage* per far sembrare all'utente che i nomi possano essere più lunghi. Di conseguenza, in Windows potete dare a un file il nome *AldoGiovanniGiacomo.doc*, ma il sistema operativo lo salva come *ALDOGI~1.DOC* e usa un truccetto software per associare al file il nome più lungo. Se elencate il file *AldoGiovanniGiacomo.doc* con il comando **dir** di MS-DOS, vedete i due nomi: quello vero, tutto in maiuscolo e spesso terminante con una tilde, e quello "falso", lungo e con maiuscole e minuscole.

Ma lasciamo stare il punto di vista tecnico (che però tra poco ci ripiomberà addosso) e guardiamo quello pratico. *Escamotage* o meno, resta il fatto che Windows vi consente di assegnare ai file nomi molto lunghi e di usare maiuscole, minuscole e spazi. Le estensioni possono avere più di tre caratteri. Per la maggior parte delle situazioni sono limiti più che ragionevoli e in effetti nessuno se ne lamenta.

Linux, invece, usa un approccio molto più concreto. I nomi dei file possono contenere fino a 256 caratteri, e non sono fittizi: il sistema operativo scrive i file usando direttamente i nomi lunghi, senza adoperare nomi brevi intermedi.

Cosa importante, Linux distingue fra maiuscole e minuscole; Windows no.

Un problema minuscolo ma non troppo

Un momento. Poche righe fa ho detto che Windows consente di usare le maiuscole e le minuscole. Adesso dico che Windows non le distingue. Sono rimbambito?

Probabilmente sì, ma questa è un'altra storia. Attenzione, infatti alla differenza: Windows può *usare* maiuscole e minuscole, ma non le *distingue*. Vi propongo un piccolo esercizio che chiarisce la differenza e fa quasi sempre restare a bocca aperta gli utenti Windows.

- In Windows, create nella stessa cartella due file (anche vuoti) di nome *Prova.txt* e *test.txt*. Fate attenzione a scrivere la P in maiuscolo e il resto in minuscolo.
- Rinominate *test.txt* in *prova.txt*. A questo punto vi aspettereste di ottenere due file distinti: *prova.txt* (P minuscola) e *Prova.txt* (P maiuscola). Mica vero! Windows non vi lascia rinominare il file: vi dice *Impossibile rinominare test: File già esistente. Specificare un altro nome file*.

Perché succede una cosa del genere? Succede per via di quell'*escamotage* accennato prima: nonostante quello che vi fa vedere sullo schermo, sotto sotto Windows non fa distinzione fra maiuscole e minuscole. Per Windows, *prova.txt*, *Prova.txt*, *PROVA.TXT*, *pRoVa.TxT* sono tutti lo stesso nome di file, e non è ammesso avere file omonimi in una

medesima cartella, per cui *prova.txt* e *Prova.txt* non possono coesistere.

In Linux, invece, dato che il sistema operativo gestisce *realmente* i nomi lunghi, non ci sono problemi. *Prova.txt* e *prova.txt* sono, per Linux, due nomi di file assolutamente distinti e possono coesistere in una stessa directory. Più in generale, l'intero sistema operativo distingue fra maiuscole e minuscole.

Questa è una delle cose che gli utenti Windows trovano più difficile da padroneggiare a causa dell'abitudine acquisita con il sistema operativo Microsoft. Ad esempio, in Windows potete lanciare il Blocco Note digitando indifferentemente `c:\windows\notepad.exe`, `c:\Windows\Notepad.exe`, `C:\WinDOWS\NOTepad.EXE`, eccetera. In Linux, se non digitate il nome del file (o della directory) rispettando esattamente le maiuscole e le minuscole, non otterrete un bel nulla.

Il silenzio è d'oro

Linux è pieno di stranezze, per chi arriva dal mondo Windows. Ad esempio, ha un atteggiamento estremamente laconico, che regolarmente fa imbestialire gli utenti Windows; infatti spesso risponde ai comandi con il silenzio più totale (c'è una buona ragione per questo comportamento, ma ve la racconto altrove). La regola generale è che se non ottenete alcuna risposta a un comando, il comando è andato a buon fine, ma non sempre è così.

Facciamo il punto

Un'altra differenza fra Windows e Linux che vi conviene tenere a mente è l'uso del punto nei nomi dei file. In virtù delle sue origini nel DOS, Windows usa molto l'estensione del nome del file e la separa dal nome con un punto. Il punto è quindi molto importante in Windows a causa di questa sua funzione di separatore.

Per Linux, invece, il punto è un carattere come un altro. Questo significa che alcuni nomi di file perfettamente accettabili in Linux non lo sono in Windows. Ad esempio, in Linux si può avere un nome di file che inizia con il punto, come questo:

.pippo

Anzi, un nome di file del genere va così a genio a Linux che viene usato per identificare dei file speciali che vedremo più in là. Nomi di questo tipo, invece, sono inaccettabili in Windows.

La cosa buffa è che è perfettamente possibile aprire una finestra DOS di Windows e assegnare a un file un nome che inizia con un punto: Windows lo gestirà e leggerà come qualsiasi altro file. Tuttavia non potete usare Windows (ad esempio Esplora risorse) per fare la stessa cosa, il che dimostra che il prompt di MS-DOS contenuto in Windows appartiene in realtà a un sistema operativo completamente separato da Windows che ha delle regole tutte sue.

Questioni di carattere

Anche se tecnicamente in Linux un nome di file può contenere qualsiasi simbolo tranne la barra (/), ci sono alcuni caratteri che è meglio evitare nei nomi, perché possono confondere il sistema operativo. Linux infatti li potrebbe interpretare come comandi o parametri:

- & (ampersand)
- ' (apostrofo)
- * (asterisco)
- ^ (accento circonflesso)
- / (barra)
- ? (punto interrogativo)
- " (virgolette)

- | (barra verticale)

Spazio, ultima frontiera

Un altro carattere che va evitato in Linux e che invece Windows gestisce con disinvoltura è lo spazio. Per Windows, *Aldo Giovanni Giacomo.doc* è un nome perfettamente legittimo. Invece se Linux incontra in un comando un nome di file che contiene spazi, è molto probabile che lo interpreti come se si trattasse di due parametri o nomi di file distinti, anche se in senso stretto lo spazio è un carattere legittimo per i nomi dei file anche in Linux. Questo può causare non pochi problemi se scambiate file fra Windows e Linux.

La soluzione semplice a questo rischio consiste nell'evitare di usare gli spazi nei nomi dei file, sia in Windows, sia in Linux. Se proprio volete separare le parole nei nomi dei file, usate il carattere *underscore* (`_`): *Aldo_Giovanni_Giacomo.doc* va bene sia a Windows, sia a Linux.

Un'altra differenza importante (e pericolosa) a proposito di spazi è che Linux consente di usare lo spazio come primo carattere di un nome; Windows no. Il pericolo sta nel fatto che alcuni individui particolarmente dispettosi salvano i file creati con Linux usando uno spazio all'inizio del nome, e l'utente che li riceve si danno per capire perché non riesce ad aprirli. Peggio ancora, questo truccetto può essere usato per commettere vari atti distruttivi e vandalici a danno di un sistema Linux.

Estensioni

La parte del nome del file che segue l'ultimo punto, dicevo, delimita l'estensione: ad esempio, nel nome di file *comandante.straker.jpg*, l'estensione è *jpg*, non *straker.jpg*. Windows usa le estensioni per sapere di che tipo di file si tratta: tutti i documenti Word finiscono in *.doc*, i fogli di calcolo di Excel finiscono in *.xls*, e così via. Se cambiate l'estensione a un file, Windows non è più in grado di associare quel file all'applicazione che lo gestisce.

In Linux, invece, le estensioni sono meno importanti. Potete usare l'estensione che volete: magari usate *.prova* per indicare documenti non definitivi o *.ricetta* per identificare le vostre ricette di cucina. Linux preferisce distinguere i tipi di file usando una cosa chiamata *magic number*, che in realtà non è necessariamente un numero come potreste sospettare, ma è una sequenza di byte presente verso l'inizio del file.

In pratica, quando chiedete a Linux di aprire un file, Linux guarda l'inizio del file alla ricerca di una delle sequenze di byte che tipicamente contraddistinguono file di un certo tipo: ad esempio, se il file contiene verso l'inizio la sequenza *0xffd8* è un'immagine in formato JPEG. Anche se un file non ha l'estensione *.jpg*, Linux sa che è un'immagine JPEG.

Tuttavia molti programmi per Linux usano le estensioni per identificare i file, e anche in molte situazioni Linux stesso usa le estensioni o qualcosa di simile: ad esempio, un file il cui nome termina con la tilde viene identificato come file da cestinare.

Fra l'altro, Linux usa anche i *prefissi* per identificare i file: un file il cui nome inizia con il punto è considerato file di configurazione e viene nascosto alla normale elencazione dei file.

Copiare e spostare file

Il procedimento per copiare e spostare file e directory in Linux è grosso modo lo stesso adottato da Windows, a parte la questione del doppio clic (che in Linux non si usa quasi mai). La cosa più importante da sapere e a cui fare molta attenzione è la datazione dei file copiati. In Windows, infatti, se copiate un file la data del file rimane invariata. In Linux, invece, la data cambia.

Ad esempio, se il 9 dicembre 2001 copiate con Linux un file che avete aggiornato l'ultima volta il 28 settembre dello stesso anno, la copia verrà datata 9 dicembre e sembrerà quindi più recente rispetto all'originale, anche se in realtà è assolutamente identica (ma è stata creata in un momento diverso). Ci sono vari modi per scavalcare questa stranezza, e li vedremo nei capitoli successivi. Per ora, l'importante è che vi ricordiate che Linux si comporta così.

Collegamenti

Come utenti Windows, avete probabilmente dimestichezza con i *collegamenti* (chiamati anche *link*), che sono quelle icone con la freccina il cui nome inizia con *Collegamento a...*; cliccando sull'icona, viene lanciato il programma (o aperto il file) che è collegato a quell'icona.

È un sistema molto comodo per radunare in un unico posto (tipicamente sul desktop) le applicazioni e i documenti che usate più frequentemente, che materialmente restano dov'erano. Il collegamento è semplicemente un rimando.

Anche Linux ha i collegamenti, ma li chiama *link*. Questo, fra l'altro, ha il vantaggio di evitare la confusione che impera nella versione italiana di Windows fra *collegamento* e *connessione*: a Internet non ci si "collega", ma ci si "connette", ma chi si ricorda mai la differenza?

Essendo Linux quel che è, non si accontenta di avere le stesse cose di Windows: deve avere qualche variante in più. Infatti in Linux esistono *due* tipi di link: quello *hard* o *fisico* e quello *soft* o *simbolico*. Quello simbolico è l'equivalente di un collegamento di Windows: un rimando a un file (documento o programma) situato altrove. Quello *hard*, invece, è una peculiarità di Linux: è una copia integrale del file a cui è collegato.

Ad esempio, il terzo nome del lettore di CD-ROM, cioè `/dev/cdrom`, è un link simbolico collegato al nome vero del dispositivo, che è `/dev/hdc`.

Non è necessario che sappiate vita, morte e miracoli dei link di Linux; per ora è sufficiente che sappiate che esistono e che si adoperano spessissimo, così non vi spaventerete quando ne incontrerete uno.

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.