

Linux Facile¹

di Daniele Medri

5 agosto 2000

¹© 2000 Daniele Medri - "Linux Facile" versione 2.1. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with one Front-Cover Texts: "Linux Facile - di Daniele Medri", and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License" (pag. 181).

Introduzione

Questo manuale nasce con la precisa intenzione di creare un documento in lingua italiana capace di sintetizzare i passaggi storico-evolutivi che hanno portato alla nascita di GNU/Linux¹ e introdurre un discreto numero di nozioni teorico/pratiche da utilizzare quotidianamente. L'organizzazione degli argomenti è stata pensata per costruire una formazione in ambienti Unix/Linux per un utente con poche esperienze informatiche.

La distribuzione GNU/Linux utilizzata a titolo di esempio è Red Hat Linux, scelta fatta per la diffusione di quest'ultima. Altre distribuzioni sono altrettanto valide e possono essere utilizzate con le opportune eccezioni del caso.

Organizzazione degli argomenti

Il concetto di *percorso formativo* ha caratterizzato ogni versione del manuale e ha trovato la sua massima espressione applicativa in quella attuale. Gli argomenti presentati sono stati divisi per parti:

- La prima parte analizza i fattori *ambientali* che hanno spinto la crescita di GNU/Linux. L'obiettivo era quello di offrire una chiave di lettura integrativa alle solite nozioni tecniche.
- La seconda parte introduce GNU/Linux sul piano storico e su quello tecnico illustrando le componenti del sistema analizzando applicazioni, soluzioni per l'installazione e la gestione.
- La terza parte analizza gli ambienti grafici. Un'ampia introduzione sul Server XFree86 e la descrizione degli ambienti desktop largamente diffusi e le applicazioni utili per essere produttivi e collegarsi alla rete.
- La quarta parte è stata pensata per il networking, ovvero l'utilizzo di un sistema GNU/Linux per soluzioni di rete o semplicemente per navigare su Internet tramite un modem.
- L'ultima parte ingloba informazioni complementari per un utente che si avvicina alla rete per la prima volta: smiles, codifica nazioni.

Note dell'autore

Questo manuale è stato sviluppato utilizzando LyX / L^AT_EX ed è liberamente distribuibile ed utilizzabile secondo le norme della licenza GNU Free Documentation License (pag. 181) che tutela i diritti dell'opera e dell'autore. Ogni suggerimento o consiglio per migliorare l'opera in questione è ben accetto e può essere inviato per posta elettronica all'indirizzo:

```
Daniele Medri <madrid@linux.it>  
homepage: http://www.linux.it/~madrid/  
fingerprint: EA70 54DD 1D34 6523 D90A A74D 214E 0E59 2529 210F
```

¹Linux è un marchio registrato da Linus Torvalds.

Per ottenere la versione aggiornata del manuale in formato digitale:

<http://www.linuxfacile.org/>

Questo manuale è stato sviluppato in buona parte su una workstation gentilmente concessa da Sgi Italia (Silicon Graphics Inc.) per lo sviluppo delle attività Open Source che seguo e mantengo.

Versioni precedenti

- 1.0 - 1 gennaio 2000.
 - 1.1 - 7 gennaio 2000.
 - 1.1a - per l'editore Trademedia (Milano).
 - 1.1b - per Sgi Italia.
- 1.2 - 27 marzo 2000.
- 2.0 - 30 luglio 2000.

Ringraziamenti

Una doverosa menzione di tutte le persone che ogni giorno contribuiscono a diffondere GNU/Linux nel mondo e in Italia particolarmente, persone che volontariamente spendono il proprio tempo libero per un impegno che è anche loro passione. Ringrazio particolarmente: Marco D'Itri, Luca Maranzano, Andrea Arcangeli, Luca *andrew* Andreucci, Maurizio *napo* Napolitano, Michel *ziobudda* Morelli, Marcello Missiroli, Carlo *joker* Catucci, Gianluca Rubini, Dario Burnelli, Alessandro *Alf* Forghieri, Massimiliano Gallinucci, Giorgio Zarrelli, Cristian Mezzetti, Maurizio *Tannoiser* Lemmo. Questo manuale è dedicato alla mia famiglia e Monica, coloro che condividono con me vita e sentimenti.

Hanno contribuito allo sviluppo del testo Matteo *MatOfPing* Nastasi, Federico Dalpane, Gilberto *velenux* Ficara e Massimo Scabbia. Un ringraziamento particolare a tutti i lettori che con le loro domande e i loro suggerimenti mi hanno aiutato a rendere sempre più completa questa guida.

Parte I

Storia

Capitolo 1

Hackers

L'esponenziale crescita della rete ha cambiato l'identità della società evoluta o per lo meno quella "informatizzata". Un incredibile mole di dati circola sulle autostrade dell'informazione, dati privati, transazioni economiche e quant'altro. La vita delle persone si impossessa del mezzo e porta con sé ogni umana caratteristica e fobia, reale e non. Se nel mito popolare esisteva l'uomo nero oggi esiste un nuovo termine per *descrivere qualcuno che attenta ai propri diritti*: hacker.

Avevo già opinioni in merito quando Filippo Bianchi, un giornalista di Repubblica, mi intervistò chiedendomi chiarimenti sulla terminologia a lui poco chiara e molto blasonata dai media che ruotava attorno al termine e in tale ottica riporto le mie considerazioni a titolo informativo.

Jargon: la nascita di un gergo

L'inizio della cultura hacker si può datare all'anno 1961, lo stesso anno in cui il MIT acquistò il primo calcolatore PDP-1. In questo contesto universitario il termine fu subito adottato dagli studenti del *Tech Model Railroad Club*, un gruppo di persone che si divertiva a costruire automatismi per gestire il traffico ferroviario per modellini di treni. Lo stesso club divenne in seguito il principale nucleo del laboratorio di Intelligenza Artificiale del MIT, gruppo di sviluppo delle principali tecnologie moderne informatiche.

Con lo sviluppo della rete di comunicazione ARPAnet, *il gergo hacker* (jargon) si diffuse nelle principali Università collegate. La terminologia utilizzata fu raccolta in un file da Raphael Finkel presso l'università di Stanford nel 1975.

Un nuovo polo di sviluppo della cultura hacker fu il centro di ricerca di Palo Alto dove dalla fine degli anni '70 alla metà degli anni '80 vennero sviluppate un numero altissimo di nuove tecnologie come le interfacce visuali ad icone e finestre, le stampanti laser e tecnologie per le reti locali di computer. Il legame tra *hacker* e nuove scoperte tecnologiche è quindi strettissimo. Caratteristica comune per quelle persone che definiamo hacker è il raggiungimento di un obiettivo nel minor tempo possibile, una rivisitazione in chiave moderna di "il principio giustifica il mezzo" di Macchiavelli.

Come viene riportato nel Jargon File:

“:hacker: n. [originally, someone who makes furniture with an axe] 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. 2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming. 3. A person capable of appreciating {hack value}. 4. A person who is good at programming quickly. 5. An expert at a particular program, or one who frequently does work using it or on it; as in 'a Unix hacker'. (Definitions 1 through 5 are correlated, and people who fit them congregate.) 6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example. 7. One who enjoys the intellectual challenge of creatively overcoming or

circumventing limitations. 8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence 'password hacker', 'network hacker'. The correct term for this sense is {cracker}."

Una persona che si diverte esplorando i dettagli nella programmazione di sistemi e cerca la modalità per ottenere il massimo delle prestazioni. Qualcuno che programma con entusiasmo (spesso ossessivamente) o si diverte programmando ancor prima che teorizzare sulla programmazione. Una persona capace di apprezzare. Una persona capace di programmare rapidamente. Una persona esperta nell'utilizzo di un dato programma. Una persona esperta o entusiasta per ogni ambito, anche l'astronomia.

Esiste anche una connotazione negativa del termine legata a fattori di criminalità informatica e viene espressamente indicata con il termine *cracker*. I media spesso confondono le due definizioni e utilizzano *hacker* con carattere negativo. Anche la *new economy* gioca con il fattore terrore e non è un caso che siano proprio i produttori di soluzioni di sicurezza a migliorare le proprie quotazioni in borsa durante attacchi informatici, un *effetto* che spesso autoalimenta la propria *causa* scaturante.

Cracker e Phreaker

Il termine è stato coniato all'incirca nel 1985 da hackers che cercavano di difendere e distaccarsi dalle connotazioni negative che i media usavano a dismisura.

Fondamentalmente entrambi i termini delineano persone che hanno compiuto attività di cracking e le tecniche spesso sono le medesime ma spesso la differenza è nel fine. Mentre un hacker scardina delle misure di sicurezze per ottenere uno scopo benefico e aggirare dei limiti tecnici, un cracker si appropria della conoscenza per proprio esclusivo interesse.

Ulteriore neologismo che si integra con i termini citati è phreaker, colui che compie cracking sulla rete telefonica per compiere ad esempio chiamate a lunga distanza senza spendere nulla. Fino alla prima metà degli anni '80 la tecnologia che controllava le reti telefoniche era piuttosto antiquata e facilmente scardinabile per propri scopi. Il cambiamento avvenne con l'aggiornamento tecnologico successivo che tagliò radicalmente molte delle vecchie tecniche di phreaking. Anche l'arresto di gruppi di persone che compievano questa tipologia di azioni divenne un freno ed è diventato difficile sentir parlare di *blue box*¹ al giorno d'oggi.

Linux e l'hacking

Linus Torvalds, il creatore di Linux, ha sempre definito la propria soluzione software come "*un sistema operativo per hackers scritto da un hacker*". Nessuno può trovare connotazioni negative nello sviluppo di un sistema operativo *free*, connotazione profondamente diversa da quella dei software proprietari. GNU/Linux stesso non sarebbe quello che è oggi senza l'apporto e la creatività degli hackers della Free Software Foundation di Stallman e dei cosiddetti *battitori liberi*, studenti e sviluppatori che portano la propria esperienza e il proprio tempo verso il *free software* con il desiderio di farlo espressamente.

¹Le blue box erano appositi strumenti che venivano utilizzati dai phreaker per simulare le chiamate telefoniche.

Capitolo 2

Internet

Internet ha decretato la nascita di una nuova civiltà fondata sulla comunicazione telematica. Sono cadute le barriere legate alla distanza metrica ed è possibile, grazie alla rete, comunicare in tempo reale con una persona nella parte opposta del globo. Internet ha reso il mondo a misura d'uomo.

2.1 Storia

Alla fine degli anni '60 il Dipartimento della Difesa degli Stati Uniti sviluppò soluzioni per affrontare eventuali attacchi bellici. Erano gli anni della guerra fredda e questa condizione spingeva la ricerca verso sistemi di sicurezza e comunicazione. L'iniziale progetto venne affidato dal Dipartimento della Difesa ad ARPA¹ dando vita ad un'iniziale rete di soli 4 nodi. Questa prima struttura era basata sulla pariteticità dei nodi ovvero sull'assenza di un nucleo centrale di controllo o elaborazione allo scopo di garantire il funzionamento delle singole reti in caso di eventi bellici. La trasmissione dei dati all'epoca era particolarmente vincolata alle infrastrutture hardware e i sistemi operativi utilizzati; ARPAnet era così vincolata a particolari infrastrutture di calcolo e comunicative. I limiti furono presto superati grazie allo sviluppo di un protocollo ideato per la trasmissione dei dati, il tcp/ip (Transmission Control Program/Internet Protocol), lo stesso protocollo che tuttora viene utilizzato per Internet. L'Internet Protocol Suite, completata negli anni '70, era in sintesi un mezzo flessibile per la comunicazione tra "mezzi di trasmissione" eterogenei. Dopo questa particolare innovazione ARPAnet divenne predominio dei ricercatori universitari che lavoravano per il Pentagono, la comunità scientifica che si accorse per prima della grande utilità nell'utilizzo remoto di supercalcolatori per scopi di ricerca e comunicazione tra le varie università americane sparse sul territorio nazionale. La mole di informazione crebbe a tal punto che nel 1984 ARPAnet venne scissa in due reti ben distinte: Milnet per scopi militari ed Arpanet per scopi scientifici. Il Dipartimento della Difesa Americana smise di finanziare il traffico interuniversitario ed il compito di gestione venne assunto dal National Science Foundation, che diede vita a NSFnet. Nel 1985 la rete scientifica cambiò nome in Internet e le stime mostravano 326 computer collegati dei quali 16 fuori dagli Stati Uniti. Nel 1990 erano collegati in rete circa 1000 calcolatori e tre anni dopo, nel 1993, la cifra divenne 20 volte superiore. Tutto questo fu possibile grazie alla *liberalizzazione* di Internet in ambiti accademici e scientifici e alla diffusione del sistema operativo Unix nella versione Berkeley, strumento storicamente in relazione con la rete stessa sin dalle sue prime fasi di sviluppo.

2.1.1 Autostrade dell' Informazione

All'inizio degli anni '90, come abbiamo rilevato precedentemente, il numero di calcolatori collegati ad Internet ebbe una crescita esponenziale tanto da interessare molti enti commerciali. I

¹Advanced Research Project Agency.

vincoli iniziali vietavano l'utilizzo della rete per scopi differenti da quelli scientifici e molte compagnie interessate furono spinte ad entrare in stretta correlazione con le università nell'ambito della ricerca per surclassare le limitazioni. Molte compagnie commerciali crearono loro stesse delle dorsali di comunicazione da integrare ad Internet per i propri scopi e generarono un processo evolutivo senza limitazioni verso la copertura capillare del pianeta. Dopo una situazione simile le barriere vennero abbattute e si diede libero campo alle iniziative private e all'uso della rete per scopi variegati. Le autostrade dell'informazione (Information Highways), così come venne definita Internet, erano un obiettivo fondamentale nella campagna politica di Al Gore, un sogno che divenne realtà con il suo incarico a vice-presidente degli Stati Uniti durante il governo Clinton. Gli Stati Uniti erano consci del loro vantaggio tecnologico e dell'opportunità che detenevano nello sviluppo di queste strutture globali. La posta in gioco era composta da ingenti capitali economici ma l'interesse di modellare a loro immagine e somiglianza l'intera cultura moderna era l'elemento maggiormente perseguibile. Vennero fatte diverse scelte strategiche: finanziamenti statali per lo sviluppo di strutture scolastiche / sanitarie e la liberalizzazione delle comunicazioni, i primi erano volti allo sviluppo di servizi basilari e la seconda scelta diretta a fermare eventuali comportamenti di oligopolio.

2.1.2 Il Vecchio Continente

Pur con un approccio di base differente la Comunità Economica Europea avviò il suo cammino in modo differente con la speranza di seguire un comportamento equilibrato nelle sue scelte decisionali. Esisteva un progetto del 1985 denominato Cosine (Cooperation for Open Systems Interconnection in Europe) per la costituzione di una rete europea integrando realtà nazionali già esistenti. Il processo evolutivo di globalizzazione e sviluppo delle comunicazioni venne riassunto in un rapporto nel 1992, un documento importante che ha dettato le linee di comportamento per la liberalizzazione delle comunicazioni a partire dal gennaio 1998. Nell'ambito Italiano le direttive comunitarie sono state rispettate e si è visto il passaggio in fasi differenti da un unico ente di gestione delle telecomunicazioni ad una situazione "aperta" gestita da un numero di compagnie private e supervisionata dall'autorità governativa.

2.2 DNS e i Nomi Dominio

Il DNS (Domain Name System) è lo strumento utilizzato per gestire una parte molto importante di Internet: la risoluzione dei nomi di dominio.

2.2.1 Internet e le internet

E' doveroso analizzare, in questa sezione, due termini che spesso possono esser visti come identici. Le parole in questione sono "Internet" con la lettera iniziale maiuscola e "internet". Sebbene possa sembrare una finezza, la distinzione c'è ed è importante. Internet, con la lettera maiuscola, fa riferimento alla rete che iniziò con il progetto ARPAnet e continua oggi come la confederazione di tutte le reti tcp/ip direttamente o indirettamente collegate al backbone² degli Stati Uniti. La parola "internet" con tutte le lettere minuscole rappresenta tutte le reti, di piccole o grandi dimensioni, che utilizzano lo stesso protocollo di comunicazione. Ogni internet (lettere minuscole) non necessariamente dev'essere connessa a Internet (lettera maiuscola iniziale) e non necessariamente devono utilizzare tcp/ip come loro protocollo di base. Le internet aziendali sono un esempio, come quelle di tipo Xerox XNS o DECnet. Negli ultimi anni sono nati ulteriori nuovi termini più che altro spinti da politiche di marketing. Il termine "intranet" rappresenta una rete aziendale che utilizza gli strumenti di Internet in un ambito ristretto. Parallelamente il termine "extranet" rappresenta la rete che unisce diverse compagnie tra loro, oppure una compagnia con i propri rivenditori.

²Il backbone è l'infrastruttura portante delle telecomunicazioni.

2.2.2 Le Origini del Domain Name System

In questo stesso capitolo si è già accennato ai passaggi storici da ARPAnet a Internet. Nella primordiale rete esistente ogni computer collegato era caratterizzato da informazioni che lo identificavano in maniera univoca in rete. Il file che conteneva tutte queste informazioni era HOSTS.TXT³ ed era mantenuto dal Network Information Center dello SRI⁴. Gli amministratori dei sistemi connessi in rete ARPAnet inviavano i propri cambiamenti al SRI-NIC e scaricavano successivamente il file HOSTS.TXT con le variazioni avvenute. Era chiaramente un meccanismo macchinoso e lo divenne ancor di più quando ARPAnet iniziò ad utilizzare il protocollo TCP/IP e la popolazione della rete crebbe enormemente. A questo punto il metodo basato sul file HOSTS.TXT riscontrava diversi problemi:

Collisione dei nomi Era facile avere due nomi di computer identici in rete e questo si scontrava con il presupposto di univocità preposto.

Consistenza Mantenere aggiornato il file HOSTS.TXT divenne sempre più difficile visti i continui aggiornamenti che venivano richiesti.

Traffico e carico Gli strumenti presenti presso SRI-NIC divennero inefficienti a causa del traffico di rete incredibile e dell'alto carico che dovevano subire i processori dei Server presenti.

Le persone che governavano la rete ARPAnet cercarono una tecnologia valida per sostituire il metodo fino ad allora utilizzato, capace di scalare meglio le esigenze, ridurre il traffico in rete, facile da gestire e garante dell'integrità dei nomi di dominio esistenti. Nel 1984, Paul Mockapetris creò la basilare struttura del nuovo sistema e rilasciò in rete i documenti RFC⁵ 882 e 883 dove veniva descritto il Domain Name System, o semplicemente DNS. La prima implementazione di DNS era chiamata JEEVES ma venne rapidamente susseguita da BIND, un software sviluppato per la versione di Unix BSD 4.3⁶ da Kevin Dunlap.

2.2.3 Nomi di Dominio

Un nome di dominio è una "etichetta", un nome che viene associato ad un indirizzo ip in rete ed è rappresentato da un nome e un top-level domain, ad esempio:

```
istruzione.it
```

Dove "istruzione" è un nome fittizio e ".it" rappresenta la nazionalità del dominio in questione.

Top-Level Domain Lo spazio dei nomi di dominio era diviso originariamente in 7 Top-Level:

.com Organizzazioni di carattere commerciale, come ad esempio Silicon Graphics (sgi.com), Sun Microsystems (sun.com).

.edu Istituti di educazione americani, come ad esempio U.C. Berkeley (berkeley.edu), Purdue University (purdue.edu).

.gov Organizzazioni governative, come ad esempio la NASA (nasa.gov), National Science Foundation (nsf.gov).

³In ambienti Unix rappresentato dal file /etc/hosts.

⁴SRI è l'istituto della ricerca di Stanford a Menlo Park, California.

⁵RFC o Request for Comments. Sono documenti che introducono una nuova tecnologia sulla rete Internet. Gli RFC sono distribuiti gratuitamente in rete e sono destinati agli sviluppatori che vogliono supportare o migliorare determinate tecnologie.

⁶UNIX BSD 4.3 è stato sviluppato dall'Università della California a Berkeley.

.mil Organizzazioni militari, come ad esempio l'esercito degli Stati Uniti, U.S. Army (army.mil) e la Marina, U.S. Navy (navy.mil).

.net Organizzazioni orientate alle tecnologie di networking, come ad esempio NSFNET (nsf.net).

.org Organizzazioni non commerciali, come ad esempio Electronic Frontier Foundation (eff.org).

.int Organizzazioni internazionali, come ad esempio la NATO (nato.int)

Non è un caso che l'orientamento dei top-level domain sia modellato principalmente attorno alle esigenze degli Stati Uniti; si ricorda che proprio quest'ultimi finanziarono il progetto ARPAnet, successivamente diventato Internet.

Lo sviluppo della rete ha mostrato le limitazioni degli originali top-level domain e a questi sono stati aggiunti i Domini Nazionali, ovvero una codifica di 2 lettere che identifica ogni singola nazione nel mondo secondo lo standard ISO 3166⁷. L'Italia è rappresentata dal top-level domain `it`.

Registrare un Nome di Dominio

La gestione dei nomi di dominio è gestita in modalità differenti. Nel caso di nomi di dominio che utilizzano gli originari 7 top-level domain la gestione è riservata esclusivamente all'InterNIC, un organo nazionale statunitense. Per quel che riguarda i nomi di dominio ISO 3166 la gestione viene affidata a livello nazionale. In Italia la gestione del top-level domain `it` è gestita dal GARR, autorità al quale è necessario rivolgersi per la registrazione.

2.2.4 DNS in Breve

Il Domain Name System è una banca dati distribuita che permette il controllo locale dei nomi di dominio su segmenti della rete. Tramite un sistema di replica, le modifiche fatte su un determinato segmento di rete da parte degli amministratori incaricati vengono "copiate" e distribuite pubblicamente nella totalità dei computer collegati. I programmi che gestiscono tuttora questi segmenti di rete sono denominati Name Server mentre sono Resolver i programmi che interrogano quest'ultimi per la risoluzione di un nome di dominio in rete. Per chiarire maggiormente il sistema prendiamo un esempio base. Un amministratore di DNS deve gestire un particolare dominio chiamato `linux.it`. In questo dominio è presente un computer in rete che vogliamo chiamare `erlug.linux.it`. L'amministratore per fare questo deve agire sul Name Server presente sul dominio `linux.it`, aggiungere l'indirizzo ip e il nome `erlug.linux.it`. Sarà compito del Name Server stesso, tramite il sistema che lo contraddistingue a replicare questi dati sui DNS di livello superiore finché l'informazione non è accessibile ad ogni computer in rete. Come si può capire questo metodo scompone il traffico di rete e il carico dei processori a livelli.

2.3 Servizi Telematici

La nascita e lo sviluppo della rete ha creato rapporti di comunicazione e servizi relativi. Diversi sono gli strumenti sviluppati per lo svolgimento delle attività quotidiane e di seguito vengono presentati alcuni tra i più importanti.

Posta elettronica La posta elettronica (e-mail) è lo strumento più diffuso ed importante della rete che permette alle persone di comunicare tra loro. Le persone abituate a lavorare con sistemi operativi di tipo Unix sono vicine alla messaggeria digitale da sempre presente in sistemi multiutenti. Non è un caso che la forma degli indirizzi e-mail sia la stessa che rappresenta un utente collegato @ (at, presso) un sistema. Anche se sono andate perse tutte le caratteristiche tattili delle

⁷La Gran Bretagna in accordo con lo standard ISO 3166 dovrebbe utilizzare il top-level domain di tipo `gb`, ma molte organizzazioni preferiscono utilizzare l'estensione `uk`, probabilmente per motivazioni di carattere politico.

“vecchie” lettere cartacee, utilizzare la posta elettronica vuol dire utilizzare un mezzo veloce con costi ridotti e con la possibilità di trasferire file come allegati.

Newsgroup Chi si è collegato a servizi BBS conoscerà sicuramente questo servizio telematico. I newsgroup sono gruppi di discussione divisi per argomento dove è possibile spedire un proprio messaggio in una bacheca virtuale pubblica dove altre persone potranno leggerlo ed eventualmente rispondere.

IRC (Internet Relay Chat) Probabilmente è il servizio telematico che affascina maggiormente le persone. Vi permette di comunicare in tempo reale con altre persone in modalità testuale.

Web E' il servizio Internet più recente ma quello che ha avuto maggior sviluppo negli ultimi anni. Utilizzando un navigatore o browser è possibile visualizzare pagine ipertestuali di informazione con il supporto di immagini, suoni, animazioni. Per utilizzare questo servizio, una volta posseduto il software necessario, bisogna digitare nella “barra degli indirizzi” il nome del dominio che si vuole visualizzare.

2.4 Le comunità virtuali

La telematica ha cambiato la società modificando i comportamenti di quest'ultima a un livello superiore rispetto a qualsiasi altra tecnologia. Come si è visto in questo capitolo, Internet è un progetto che ha radici molto lontane e in questo arco di tempo la rete si è evoluta per rispondere alle esigenze delle persone e delle aziende che investono in questa infrastruttura.

La rete ha formato così comunità virtuali di persone che sinergicamente si evolvono o si decimano rapidamente. Relativamente a Linux si rimanda il discorso relativo ai LUG (pagina 63).

Netiquette

Come ogni comunità sociale che si forma, anche le comunità virtuali hanno una propria regolamentazione dei comportamenti. La Netiquette, come viene chiamata, è una forma di “educazione” che porta a rispettare e proteggere le interazioni tra le persone in rete. A titolo di esempio, possiamo citare come un caso estraneo alla Netiquette l'uso improprio di uno strumento come la posta elettronica per fare pubblicità commerciale. Questo fenomeno chiamato “spam” è stato largamente combattuto negli U.S.A. e l'Europa inizia a risentirne il contraccolpo in seguito alla crescita dei servizi di vendita online. Un altro caso di Netiquette può essere l'interazione che avviene tra un gruppo di persone in chat (IRC), dove comportamenti scorretti di alcuni soggetti possono limitare o interrompere i rapporti con le altre persone.

F.A.Q.

Introduciamo, sin dal primo capitolo, la sezione della F.A.Q. ovvero Frequently Asked Questions, le domande richieste maggiormente. L'autore valuta eventuali domande da parte dei lettori per ampliare la lista ed eliminare ulteriori dubbi sorti dagli argomenti in esame.

Domanda: “Devo utilizzare obbligatoriamente il DNS per collegarmi ad Internet?” Per la connessione ad Internet non è necessario questo parametro ma lo diventa se volete utilizzare le pagine web o accedere ai servizi offerti dalla rete. Senza DNS dovrete conoscere gli indirizzi IP di ogni server in rete, mentre nel caso contrario basterebbe conoscere il nome di dominio e in automatico avviene la risoluzione. E' più semplice ricordarsi che per andare sul sito di ErLug (Emilia Romagna Linux User Group) basta aprire un browser e digitare l'indirizzo `http://erlug.linux.it` invece del numero IP che identifica il server in rete.

Domanda: “Cosa mi serve per poter gestire la posta elettronica?” Se si desidera gestire una corrispondenza di posta elettronica è chiaramente necessario disporre di una propria casella postale per le risposte. Esistono numerosi servizi che offrono gratuitamente caselle di posta elettronica. Nel caso italiano esistono molti provider Internet che insieme alla connessione includono questo servizio. Una volta ottenuta una casella postale servono pochi ulteriori parametri: il server SMTP, ovvero il servizio che gestisce la “spedizione” delle vostre e-mail. Esempio:

```
mail.provider.it
```

il server POP3 (o IMAP a seconda dei casi) per gestire la posta che arriva nella vostra casella postale. E' frequente trovare come server di questo tipo lo stesso utilizzato per la spedizione. Esempio:

```
mail.provider.it
```

Il server di “spedizione” (SMTP o Simple Mail Transfer Protocol) non richiede parametri aggiuntivi per lo svolgimento delle proprie operazioni al contrario del server di “ricezione” (POP3) dove risiedono i vostri personali messaggi. Per accedere a quest'ultimo con un programma di gestione della posta elettronica è necessario aggiungere il parametro che identifica l'utente (genericamente la parte iniziale del vostro indirizzo di posta) e una parola d'ordine (password) che solo voi conoscete.

Domanda: “Utilizzando posta elettronica e newsgroup è possibile scaricare dei virus nel proprio computer?” Con la posta o le news è possibile trovare allegati dei file. Guardatevi da questi se non conoscete chi ve li spedisce! I virus sono particolari programmi che si insediano nel vostro computer alterando le operazioni basilari. Vengono trasmessi attraverso file eseguibili (ad esempio i file .exe nel mondo Dos/Windows) o attraverso documenti particolari che possono includere linguaggi macro al proprio interno (ad esempio i file .doc o .xls). Gli ultimi virus esaminati sono chiamati “Macro Virus”.

Capitolo 3

Unix

Negli anni '60, agli albori della Scienza dei Computer (*Computer Science*) i calcolatori non *parlavano* tra loro. Spesso i calcolatori di uno stesso produttore avevano bisogno di interpreti per *condividere* informazioni ed è inutile prendere in considerazione computer di produttori differenti. Sono passati molti anni e in estrema sintesi verranno illustrati nella pagine che seguono.

3.1 Da Multics a Unix

Nel 1965, nei laboratori della Bell, una divisione AT&T, veniva aperto un progetto chiamato Multics (Multiplexed Information and Computing Service) che vedeva la collaborazione di General Electric e del MIT¹ per trovare una soluzione capace di gestire centraline telefoniche, sistemi di calcolo multiutente con costi ridotti. Nonostante le buone intenzioni, problemi di budget decretarono la fine del progetto ma questo non fermò Ken Thompson e Dennis Ritchie (immagine 3.2) che in seguito all'esperienze fatte precedentemente continuarono a lavorare su un progetto loro.

La scelta si trovò difronte ad evidenti problemi di sorta, ovvero la necessità di trovare un calcolatore per portare avanti il progetto. Venne preparato un piano di sponsorizzazioni ma questo fu rifiutato e le voci di corridoio dicono che Bill Baker, il vice-presidente dei Bell Labs, esclamò:

"Bell Labs just doesn't do business this way!"

Era un progetto molto vago sotto l'aspetto economico e non fu abbastanza convincente per ricevere i fondi necessari, là dove era già stata decretata la fine di Multics in precedenza. Ritchie e Thompson non si persero d'animo, prepararono un prospetto del sistema operativo e lo distribuirono tramite l'ufficio stampa dei Bell Labs ad un ampio numero di ricercatori. Questo scatto ebbe i suoi frutti e venne trovato un *piccolo e usato* PDP-7 da utilizzare per la sperimentazione e lo sviluppo del progetto. L'estate del '69 fu interamente dedicata a realizzare una bozza significativa per iniziare i lavori del progetto.

Nel 1970 fu battezzato *Unix* da Brian Kernighan, come gioco di parole con *Multics*, e fu sottoposto un prospetto a due responsabili dei dipartimenti di ricerca per l'acquisto di un PDP-11; Doug McIlroy e Lee McMahon furono di fondamentale importanza e una volta trovati i fondi necessari fu possibile acquistare il calcolatore per il progetto. Il porting di Unix dal PDP-7 al PDP-11 venne fatto tramite due terminali Teletype 33 (immagine 3.1) e rapidamente attrasse a sé credibilità nei confronti degli altri dipartimenti di ricerca. Sulla spinta del successo fu ulteriormente proposto di acquistare un nuovo sistema PDP-11/45 e quest'ultima non ebbe problemi a concretizzarsi.

¹Massachusetts Institute of Technology



Figura 3.1: Ritchie e Thompson alla console del calcolatore PDP-11.



Figura 3.2: Ritchie e Thompson in foto recenti.

In origine tutto il sistema operativo era scritto in linguaggio assembly. Era necessario trovare una soluzione per rendere questo iniziale sistema operativo portabile su altri calcolatori in maniera semplice. Ken Thompson inventò invece un linguaggio chiamato B, che fu influenzato da un altro linguaggio chiamato BCPL². Nel 1971 cominciò il lavoro al linguaggio che sarebbe diventato il C, un passo evolutivo che svincolava il linguaggio dal precedente calcolatore utilizzato per lo sviluppo.

Nel 1973 il kernel di Unix fu riscritto in C e poteva funzionare su dieci macchine. Nell'ottobre del 1973, Thompson e Ritchie presentarono un paper al Symposium for Operating System Principles e l'interesse per il sistema esplose. All'epoca, la AT&T era indagata per comportamenti monopolistici e le fu ordinato di rimanere fuori dall'industria dei computer; i suoi legali decisero che sarebbe stato giusto permettere alle università di disporre del sistema operativo alle condizioni dettate dalla stessa compagnia che in buona parte si sintetizzavano nella totale assenza di assistenza tecnica. Ciò costrinse gli utenti a riunirsi per potersi prestare assistenza reciprocamente, rinforzando i valori che all'inizio avevano portato alla creazione del sistema. Ma una tradizione di condivisione non avrebbe impedito che nella comunità degli utenti si formassero profonde divisioni - e due delle prime, a quel tempo stavano proprio per nascere. Nel 1975 Ken Thompson ritornò all'Università della California a Berkeley, e portò Unix con sé. Là due dottorandi, Chuck Haley e Bill Joy, fecero il porting del sistema Pascal, al quale Thompson aveva lavorato, ad uno stato in cui poteva essere utilizzato, e crearono l'editor di testi vi. Nacque così la *Berkeley System Distribution* di Unix o più semplicemente BSD Unix.

3.2 Gli interessi commerciali

Nella metà degli Anni Settanta Unix stava crescendo rapidamente, ma veniva spesso sottovalutato in rapporto a soluzioni presenti. Nel 1978 la DEC mise in commercio VAX, che era il successore del PDP-11, la culla di Unix. Questi calcolatori si diffusero rapidamente e il loro sistema operativo proprietario, il VMS, fu largamente sostituito da Unix, specialmente dopo l'uscita del BSD 4.2 nel 1982. AT&T nel 1984 fondò la propria divisione computer e vennero delineati alcuni prodotti da offrire sul mercato, tra questi Unix System V che girava su calcolatori 3B della stessa compagnia.

In coincidenza con questi sviluppi stava emergendo una nuova classe di computer, i microcomputer o workstation. Prodotti da società come Sun, Convex, Multiflow, Masscomp, Silicon Graphics, Pyramid, Stardent e NeXT, questi "killer micros", come furono chiamati, erano più veloci e meno costosi dei minicomputer che li precedettero e su tutti girava Unix. I produttori di workstation distrussero il mercato dei minicomputer e poi cominciarono a farsi guerra tra loro. Unix si era spostato al centro dell'attenzione seppure ogni produttore offriva la propria versione del sistema operativo. Alcune di queste erano basate sullo stile che si era sviluppato intorno al BSD, mentre altre si richiamavano all'USG System V Release 4 (SVR4) della AT&T. Negli anni '80 comparvero i personal computer e cominciarono a raggiungere i livelli di prestazioni al pari delle workstation del tempo e giocando a sfavore di queste ultime. Queste macchine venivano derise dai professionisti di Unix come semplici giocattoli, ma le loro capacità crebbero molto più rapidamente di quelle delle workstation, ed inoltre costavano molto meno. E tutti i PC usavano un sistema operativo chiamato DOS ed un ambiente grafico chiamato Windows. I PC basati su Windows si diffusero più velocemente di quanto i produttori di Unix potessero immaginare. Prima che essi comprendessero cosa stava succedendo, Unix era stato confinato alla stanza del server, ed anche là la sua posizione appariva minacciata.

Riferimenti

- Unix Heritage Society
<http://minnie.cs.adfa.edu.au/TUHS/>

²Basic Combined Programming Language, ideato da Martin Richards presso l'università di Cambridge.

- PDP-11 Unix Preservation
<http://minnie.cs.adfa.edu.au/PUPS/>
- *The Evolution of the Unix Time-Sharing System* di Dennis Ritchie
<http://cm.bell-labs.com/cm/cs/who/dmr/hist.html>
- La storia di Multics
<http://www.multicians.org/history.html>
- Oxford University's Virtual Museum of Computing
<http://archive.comlab.ox.ac.uk/other/museums/computing.html>

3.3 Free Software Foundation

Nel 1983 Richard M. Stallman fondò il progetto GNU e la Free Software Foundation, il cui scopo era produrre un sistema Unix-compatibile completamente gratuito e liberamente distribuibile. Tuttavia, il metodo di questo gruppo era cominciare con le utilities - editor di testo, compilatori, e tutto il resto - lavorando in direzione di un kernel completamente nuovo. Malgrado non si sia mai attuato il progetto a livello del kernel il software prodotto da GNU era di una tale qualità che gli amministratori di sistema spesso sostituirono i programmi originali delle loro distribuzioni di Unix con le versioni GNU. Malgrado questo le compagnie preferirono curare i loro interessi giocando probabilmente contro questa filosofia di libero scambio del codice senza limitazioni.

F.A.Q.

Domanda: “C'è differenza tra Unix e UNIX, scritto con lettere maiuscole?” Con *Unix* indichiamo l'insieme di tutti i sistemi di questo tipo. *UNIX* è un marchio registrato che indica la realizzazione di Unix creata da USL (Unix System Laboratories) della Novell.

Domanda: “Che importanza ha avuto il sistema operativo Unix nello sviluppo della rete Internet?” Esiste un rapporto sinergico tra Unix e Internet. Se da un lato il sistema operativo ha trovato ampiamente sviluppo in rete, Internet e le tecnologie relative hanno trovato la propria culla nel sistema operativo in questione. L'esempio del Domain Name System è rappresentativo. L'organizzazione delle informazioni è gerarchicamente molto simile alla struttura ad albero del file system Unix.

Domanda: “Unix è ancora un privilegio per pochi? Serve necessariamente un calcolatore molto potente per utilizzare questo sistema operativo?” Esistono versioni commerciali e “free”, nel concetto Open Sources. Le versioni commerciali genericamente sono sviluppate per particolari piattaforme hardware e ottimizzate per sfruttare le caratteristiche di quest'ultime. Le versioni “free” grazie alla loro natura “aperta” si evolvono rapidamente offrendo il supporto per un ampio insieme di periferiche e molte caratteristiche tecniche tipiche delle versioni commerciali. Tra i sistemi “free” possiamo ricordare GNU/Hurd e GNU/Linux: entrambi hanno un ampio supporto di periferiche e offrono prestazioni molto buone sui computer comunemente in vendita sul mercato.

Parte II
Filosofia

Capitolo 4

Open Source

Dopo aver presentato la storia di UNIX e accennato alla nascita del movimento GNU è sembrato doveroso dare maggiori dettagli sul movimento che ha coinvolto una comunità di sviluppatori molto ampia e ha dato vita tra i numerosi progetti al sistema operativo GNU/Linux.

Proprio per aumentare l'informazione in merito sono stati scelti due documenti autorevoli scritti da altrettanto autorevoli guru. Il primo documento è di Richard Stallman, l'uomo che ha fondato la Free Software Foundation, considerato per molti un visionario per molti altri un genio. Il testo vuole essere una base da analizzare per creare una propria opinione in merito. Il secondo documento è di Bruce Perens, il fondatore della distribuzione Debian ed è una dettagliata analisi delle varie licenze software presenti con relative caratteristiche. Il testo è diretto con particolare cura agli sviluppatori ma può essere un ottimo riferimento per gli utenti finali che utilizzeranno soluzioni software per le proprie esigenze.

4.1 Il progetto GNU¹

di Richard Stallman

La prima comunità di condivisione del software

“Quando cominciai a lavorare nel laboratorio di Intelligenza Artificiale del MIT nel 1971, entrai a far parte di una comunità in cui ci si scambiavano i programmi, che esisteva già da molti anni. La condivisione del software non si limitava alla nostra comunità; è un cosa vecchia quanto i computer, proprio come condividere le ricette è antico come il cucinare. Ma noi lo facevamo più di quasi chiunque altro.

Il laboratorio di Intelligenza Artificiale (AI) usava un sistema operativo a partizione di tempo (timesharing) chiamato ITS (Incompatible Timesharing System) che il gruppo di hacker² del laboratorio aveva progettato e scritto in linguaggio assembler per il Digital PDP-10, uno dei grossi elaboratori di quel periodo. Come membro di questa comunità, hacker di sistema nel gruppo laboratorio, il mio compito era migliorare questo sistema.

Non chiamavamo il nostro software "software libero", poiché questa espressione ancora non esisteva, ma si trattava proprio di questo. Quando persone di altre università o di qualche società volevano convertire il nostro programma per il proprio sistema e utilizzarlo, erano le benvenute. Se si vedeva qualcuno usare un programma sconosciuto e interessante, si poteva sempre

¹La traduzione di questo saggio è stata revisionata e curata, con la supervisione dell'autore, da Lorenzo Bettini, Antonio Cisternino, Francesco Potorti e Alessandro Rubini.

²Nota di Stallman: L'uso del termine "hacker" nel senso di "pirata" è una confusione di termini creata dai mezzi di informazione. Noi hacker ci rifiutiamo di riconoscere questo significato, e continuiamo a utilizzare la parola nel senso di "uno che ami programmare, e a cui piaccia essere bravo a farlo".



Figura 4.1: Richard Stallman

chiedere di vederne il codice sorgente, in modo da poterlo leggere, modificare, o prenderne, cannibalizzarne alcune parti per creare un nuovo programma.”

La comunità si dissolve

“La situazione cambiò drasticamente all’inizio degli anni ‘80 quando la Digital smise di produrre la serie PDP-10. La sua architettura, elegante e potente negli anni ‘60, non poteva essere estesa in modo naturale ai più grandi spazi di indirizzamento che si stavano rendendo possibili negli anni ‘80. Questo significò che quasi tutti i programmi che formavano ITS divennero obsoleti.

La comunità di hacker del laboratorio di Intelligenza Artificiale si era già dissolta non molto tempo prima. Nel 1981 la Symbolics, nata da una costola del laboratorio stesso, gli aveva sottratto quasi tutti gli hacker; l’ormai esiguo gruppo rimasto fu dunque incapace di sostenersi (il libro “Hackers” di Steve Levy narra questi eventi, oltre a fornire una fedele ricostruzione di questa comunità ai suoi inizi). Quando il laboratorio di Intelligenza Artificiale nel 1982 acquistò un nuovo PDP-10, i sistemisti decisero di utilizzare il sistema timesharing non libero della Digital anziché ITS. I moderni elaboratori di quell’epoca, come il VAX o il 68020, avevano il proprio sistema operativo, ma nessuno di questi era libero: si doveva firmare un accordo di non-diffusione persino per ottenerne una copia eseguibile.

Questo significava che il primo passo per usare un computer era promettere di negare aiuto al proprio vicino. Una comunità cooperante era vietata. La regola creata dai proprietari di software proprietario era: “se condividi il software col tuo vicino sei un pirata. Se vuoi modifiche, pregaci di farle”.

L’idea che la concezione sociale di software proprietario - cioè il sistema che impone che il software non possa essere condiviso o modificato - sia antisociale, contraria all’etica, semplicemente sbagliata, può apparire sorprendente a qualche lettore. Ma che altro possiamo dire di un sistema che si basa sul dividere utenti e lasciarli senza aiuto? Quei lettori che trovano sorprendente l’idea possono aver data per scontata la concezione sociale di software proprietario, o averla giudicata utilizzando lo stesso metro suggerito dal mercato del software proprietario. I produttori di software hanno lavorato a lungo e attivamente per diffondere la convinzione che c’è un solo modo di vedere la cosa.

Quando i produttori di software parlano di “difendere” i propri “diritti” o di “fermare la pira-

teria", quello che dicono è in realtà secondario. Il vero messaggio in quelle affermazioni sta nelle assunzioni inesprese, che essi danno per scontate; vogliono che siano accettate acriticamente. Esaminiamole, dunque.

Una prima assunzione è che le aziende produttrici di software abbiano il diritto naturale indiscutibile di proprietà sul software, e di conseguenza, abbiano controllo su tutti i suoi utenti. Se questo fosse un diritto naturale, non potremmo sollevare obiezioni, indipendentemente dal danno che possa recare ad altri. È interessante notare che, negli Stati Uniti, sia la costituzione che la giurisprudenza rifiutano questa posizione: il diritto d'autore non è un diritto naturale, ma un monopolio imposto dal governo che limita il diritto naturale degli utenti a effettuare delle copie.

Un'altra assunzione inespressa è che la sola cosa importante del software sia il lavoro che consente di fare - vale a dire che noi utenti non dobbiamo preoccuparci del tipo di società in cui ci è permesso vivere.

Una terza assunzione è che non avremmo software utilizzabile (o meglio, che non potremmo mai avere un programma per fare questo o quell'altro particolare lavoro) se non riconosciamo ai produttori il controllo sugli utenti di quel programmi. Questa assunzione avrebbe potuto sembrare plausibile, prima che il movimento del software libero dimostrasse che possiamo scrivere quantità di programmi utili senza bisogno di metterci dei catenacci.

Se rifiutiamo di accettare queste assunzioni, giudicando queste questioni con comuni criteri di moralità e di buon senso dopo aver messo al primo posto gli interessi degli utenti, tenendo conto che gli utenti vengono prima di tutto, arriviamo a conclusioni del tutto differenti. Chi usa un calcolatore dovrebbe essere libero di modificare i programmi per adattarli alle proprie necessità, ed essere libero di condividere il software, poiché aiutare gli altri è alla base della società.

Non c'è modo in questa sede di trattare approfonditamente i ragionamenti che portano a questa conclusione; il lettore interessato può cercare le informazioni in rete a questo indirizzo:

<http://www.gnu.org/philosophy/why-free.html>

Una difficile scelta morale

“Una volta che il mio gruppo si fu sciolto, continuare come prima fu impossibile. Mi trovai di fronte a una difficile scelta morale. La scelta facile sarebbe stata quella di unirsi al mondo del software proprietario, firmando accordi di non-diffusione e promettendo di non aiutare i miei compagni hacker. Con ogni probabilità avrei anche sviluppato software che sarebbe stato distribuito secondo accordi di non-diffusione, contribuendo così alla pressione su altri perché a loro volta tradissero i propri compagni. In questo modo avrei potuto guadagnare, e forse mi sarei divertito a programmare. Ma sapevo che al termine della mia carriera mi sarei voltato a guardare indietro, avrei visto anni spesi a costruire muri per dividere le persone, e avrei compreso di aver contribuito a rendere il mondo peggiore.

Avevo già sperimentato cosa significasse un accordo di non diffusione per chi lo firmava, quando qualcuno rifiutò a me e al laboratorio AI del MIT il codice sorgente del programma di controllo della nostra stampante; l'assenza di alcune funzionalità nel programma rendeva ostacolo frustrante l'uso della stampante. Per cui non mi potevo dire che gli accordi di non-diffusione fossero innocenti. Ero molto arrabbiato quando quella persona si rifiutò di condividere il programma con noi; non potevo far finta di niente e fare lo stesso con tutti gli altri.

Un'altra possibile scelta, semplice ma spiacevole, sarebbe stata quella di abbandonare l'informatica. In tal modo le mie capacità non sarebbero state mal utilizzate, tuttavia sarebbero state sprecate. Non sarei mai stato colpevole di dividere o imporre restrizioni agli utenti di calcolatori, ma queste cose sarebbero comunque successe.

Allora cercai un modo in cui un programmatore potesse fare qualcosa di buono. Mi chiesi dunque: c'erano un programma o dei programmi che io potessi scrivere, per rendere nuovamente possibile l'esistenza di una comunità?

La risposta era semplice: innanzitutto serviva un sistema operativo. Questo è difatti il software fondamentale per iniziare a usare un computer. Con un sistema operativo si possono fare molte cose; senza, non è proprio possibile far funzionare il computer. Con un sistema operativo libero, avremmo potuto avere nuovamente una comunità in cui hacker possono cooperare, e

invitare chiunque a unirsi al gruppo. E chiunque sarebbe stato in grado di usare un calcolatore, senza dover cospirare fin dall'inizio per sottrarre qualcosa ai propri amici.

Essendo un programmatore di sistemi, possedevo le competenze adeguate per questo lavoro. Così, anche se non davo il successo per scontato, mi resi conto di essere la persona giusta per farlo. Scelsi di rendere il sistema compatibile con Unix, in modo che fosse portabile, e che gli utenti Unix potessero passare facilmente a esso. Il nome GNU fu scelto secondo una tradizione hacker, come acronimo ricorsivo che significa "GNU's Not Unix" (GNU non è Unix).

Un sistema operativo non si limita solo al suo nucleo, che è proprio il minimo per eseguire altri programmi. Negli anni '70, qualsiasi sistema operativo degno di questo nome includeva interpreti di comandi, assembleri, compilatori, interpreti di linguaggi, debugger, editor di testo, programmi per la posta e molto altro. ITS li aveva, Multics li aveva, VMS li aveva e Unix li aveva. Anche il sistema operativo GNU li avrebbe avuti.

Tempo dopo venni a conoscenza di questa massima, attribuita a Hillel³:

“Se non sono per me stesso, chi sarà per me?
E se sono solo per me stesso, che cosa sono?
E se non ora, quando?”

La decisione di iniziare il progetto GNU si basò su uno spirito simile.”

"Free" come libero

“Il termine "free software" (N.d.T. il termine free in inglese significa sia gratuito che libero) a volte è mal interpretato: non ha niente a che vedere col prezzo del software; si tratta di libertà. Ecco, dunque, la definizione di software libero: un programma è software libero per un dato utente se:

- utente ha la libertà di eseguire il programma per qualsiasi scopo;
- l'utente ha la libertà di modificare il programma secondo i propri bisogni (perché questa libertà abbia qualche effetto in pratica, è necessario avere accesso al codice sorgente del programma, poiché apportare modifiche a un programma senza disporre del codice sorgente è estremamente difficile);
- l'utente ha la libertà di distribuire copie del programma, gratuitamente o dietro compenso;
- l'utente ha la libertà di distribuire versioni modificate del programma, così che la comunità possa fruire dei miglioramenti apportati.

Poiché "free" si riferisce alla libertà e non al prezzo, vendere copie di un programma non contraddice il concetto di software libero. In effetti, la libertà di vendere copie di programmi è essenziale: raccolte di software libero vendute su CD-ROM sono importanti per la comunità, e la loro vendita è un modo di raccogliere fondi importante per lo sviluppo del software libero. Di conseguenza, un programma che non può essere liberamente incluso in tali raccolte non è software libero.

A causa dell'ambiguità del termine "free", si è cercata a lungo un'alternativa, ma nessuno ne ha trovata una valida. La lingua inglese ha, più termini e sfumature di ogni altra, ma non ha una parola semplice e non ambigua che significhi libero; "unfettered" è la parola più vicina come significato (N.d.T. unfettered è una parola di tono aulico o arcaico che significa libero da ceppi, vincoli o inibizioni). Alternative come "liberated", "freedom" e "open" hanno altri significati o non sono adatte per altri motivi (N.d.T. rispettivamente, liberato, libertà, aperto).”

³Nota di Stallman: Essendo ateo, non seguo alcuna guida religiosa, ma a volte mi trovo ad ammirare qualcosa che qualcuno di loro ha detto.

Software GNU e il sistema GNU

“Sviluppare un intero sistema è un progetto considerevole. Per raggiungere l’obiettivo decisi di adattare e usare parti di software libero tutte le volte che fosse possibile. Per esempio, decisi fin dall’inizio di usare T_EX come il principale programma di formattazione di testo; qualche anno più tardi, decisi di usare l’X Window System piuttosto che scrivere un altro sistema a finestre per GNU.

A causa di questa decisione, il sistema GNU e la raccolta di tutto il software GNU non sono la stessa cosa. Il sistema GNU comprende programmi che non sono GNU, sviluppati da altre persone o gruppi di progetto per i propri scopi, ma che possiamo usare in quanto software libero.”

L’inizio del progetto

“Nel gennaio 1984 lasciai il mio posto al MIT e cominciai a scrivere software GNU. Dovetti lasciare il MIT, per evitare che potesse interferire con la distribuzione di GNU come software libero. Se fossi rimasto, il MIT avrebbe potuto rivendicare la proprietà del lavoro, e avrebbe potuto imporre i propri termini di distribuzione, o anche farne un pacchetto proprietario. Non avevo alcuna intenzione di fare tanto lavoro solo per vederlo reso inutilizzabile per il suo scopo originario: creare una nuova comunità di condivisione di software.

A ogni buon conto, il professor Winston - allora responsabile del laboratorio AI del MIT - mi propose gentilmente di continuare a utilizzare le attrezzature del laboratorio stesso.”

I primi passi

“Poco dopo aver iniziato il progetto GNU, venni a sapere del Free University Compiler Kit, noto anche come VUCK (la parola olandese che sta per "free" inizia con la V). Era un compilatore progettato per trattare più linguaggi, fra cui C e Pascal, e per generare codice binario per diverse architetture. Scrissi al suo autore chiedendo se GNU avesse potuto usarlo. Rispose in modo canzonatorio, dicendo che l’università era sì libera, ma non il compilatore. Decisi allora che il mio primo programma per il progetto GNU sarebbe stato un compilatore multilinguaggio e multiplatforma. Sperando di evitare di dover scrivere da me l’intero compilatore, ottenni il codice sorgente del Pastel, un compilatore multiplatforma sviluppato ai Laboratori Lawrence Livermore. Il linguaggio supportato da Pastel, in cui il Pastel stesso era scritto, era una versione estesa del Pascal, pensata come linguaggio di programmazione di sistemi. Io vi aggiunsi un frontend per il C, e cominciai il porting per il processore Motorola 68000, ma fui costretto a rinunciare quando scoprii che il compilatore richiedeva diversi megabyte di memoria sullo stack, mentre il sistema Unix disponibile per il processore 68000 ne permetteva solo 64K.

Mi resi conto allora che il compilatore Pastel interpretava tutto il file di ingresso creandone un albero sintattico, convertiva questo in una catena di "istruzioni", e quindi generava l’intero file di uscita senza mai liberare memoria. A questo punto, conclusi che avrei dovuto scrivere un nuovo compilatore da zero. Quel nuovo compilatore è ora noto come Gcc; non utilizza niente del compilatore Pastel, ma riuscii ad adattare e riutilizzare il frontend per il C che avevo scritto. Questo però avvenne qualche anno dopo; prima, lavorai su GNU Emacs.”

GNU Emacs

“Cominciai a lavorare su GNU Emacs nel settembre 1984, e all’inizio del 1985 cominciava a essere utilizzabile. Così potei iniziare a usare sistemi Unix per scrivere; fino ad allora, avevo scritto sempre su altri tipi di macchine, non avendo nessun interesse a imparare vi ne’ ed.

A questo punto alcuni cominciarono a voler usare GNU Emacs, il che pose il problema di come distribuirlo. Naturalmente lo misi sul server ftp anonimo del computer che usavo al MIT (questo computer, prep.ai.mit.edu, divenne così il sito ftp primario di distribuzione di GNU; quando alcuni anni dopo andò fuori servizio, trasferimmo il nome sul nostro nuovo ftp server).



Figura 4.2: Logotipo di GNU

Ma allora molte delle persone interessate non erano su Internet e non potevano ottenere una copia via ftp, così mi si pose il problema di cosa dir loro.

Avrei potuto dire: "trova un amico che è in rete disposto a farti una copia". Oppure avrei potuto fare quel che feci con l'originario Emacs su PDP-10, e cioè dir loro: "spediscimi una busta affrancata e un nastro, e io te lo rispedisco con sopra Emacs". Ma ero senza lavoro, e cercavo un modo di far soldi con il software libero. E così feci sapere che avrei spedito un nastro a chi lo voleva per 150 dollari. In questo modo, creai un'impresa di distribuzione di software libero, che anticipava le compagnie che oggi distribuiscono interi sistemi GNU basati su Linux."

Un programma è libero per tutti?

"Se un programma è software libero quando esce dalle mani del suo autore, non significa necessariamente che sarà software libero per chiunque ne abbia una copia. Per esempio, il software di pubblico dominio (software senza copyright) è software libero, ma chiunque può farne una versione modificata proprietaria. Analogamente, molti programmi liberi sono protetti da diritto d'autore, ma vengono distribuiti con semplici licenze permissive che permettono di farne versioni modificate proprietarie.

L'esempio emblematico della questione è l'X Window System. Sviluppato al MIT, e pubblicato come software libero con una licenza permissiva, fu rapidamente adottato da diverse società informatiche. Queste aggiunsero X ai loro sistemi Unix proprietari, solo in forma binaria, e coperto dello stesso accordo di non-diffusione. Queste copie di X non erano software più libero di quanto lo fosse Unix.

Gli autori dell'X Window System non ritenevano che questo fosse un problema, anzi se lo aspettavano ed era loro intenzione che accadesse. Il loro scopo non era la libertà, ma semplicemente il "successo", definito come "avere tanti utenti". Non erano interessati che questi utenti fossero liberi, ma solo che fossero numerosi.

Questo sfociò in una situazione paradossale, in cui due modi diversi di misurare la quantità di libertà risultavano in risposte diverse alla domanda "questo programma è libero"? Giudicando sulla base della libertà offerta dai termini distributivi usati dal MIT, si sarebbe dovuto dire che X era software libero. Ma misurando la libertà dell'utente medio di X, si sarebbe dovuto dire che X era software proprietario. La maggior parte degli utenti di X usavano le versioni proprietarie fornite con i sistemi Unix, non la versione libera."

Il permesso d'autore (copyleft) e la GNU GPL

“Lo scopo di GNU consisteva nell’offrire libertà agli utenti, non solo nell’ottenere ampia diffusione. Avevamo quindi bisogno di termini di distribuzione che evitassero che il software GNU fosse trasformato in software proprietario. Il metodo che usammo si chiama “permesso d’autore”⁴.

Il permesso d’autore (copyleft)⁵ usa le leggi sul diritto d’autore (copyright), ma le capovolge per ottenere lo scopo opposto: invece che un metodo per privatizzare il software, diventa infatti un mezzo per mantenerlo libero. Il succo dell’idea di permesso d’autore consiste nel dare a chiunque il permesso di eseguire il programma, copiare il programma, modificare il programma, e distribuirne versioni modificate, ma senza dare il permesso di aggiungere restrizioni. In tal modo, le libertà essenziali che definiscono il “free software” (software libero) sono garantite a chiunque ne abbia una copia, e diventano diritti inalienabili.

Perché un permesso d’autore sia efficace, anche le versioni modificate devono essere libere. Ciò assicura che ogni lavoro basato sul nostro sia reso disponibile per la nostra comunità, se pubblicato. Quando dei programmatori professionisti lavorano su software GNU come volontari, è il permesso d’autore che impedisce ai loro datori di lavoro di dire: “non puoi distribuire quei cambiamenti, perché abbiamo intenzione di usarli per creare la nostra versione proprietaria del programma”.

La clausola che i cambiamenti debbano essere liberi è essenziale se vogliamo garantire libertà a tutti gli utenti del programma. Le aziende che privatizzarono l’X Window System di solito avevano apportato qualche modifica per portare il programma sui loro sistemi e sulle loro macchine. Si trattava di modifiche piccole rispetto alla mole di X, ma non banali. Se apportare modifiche fosse una scusa per negare libertà agli utenti, sarebbe facile per chiunque approfittare di questa scusa.

Una problematica correlata è quella della combinazione di un programma libero con codice non libero. Una tale combinazione sarebbe inevitabilmente non libera; ogni libertà che manchi dalla parte non libera mancherebbe anche dall’intero programma. Permettere tali combinazioni aprirebbe non uno spiraglio, ma un buco grosso come una casa. Quindi un requisito essenziale per il permesso d’autore è tappare il buco: tutto ciò che venga aggiunto o combinato con un programma protetto da permesso d’autore dev’essere tale che il programma risultante sia anch’esso libero e protetto da permesso d’autore.

La specifica implementazione di permesso d’autore che utilizziamo per la maggior parte del software GNU è la GNU General Public License (licenza pubblica generica GNU), abbreviata in GNU GPL. Abbiamo altri tipi di permesso d’autore che sono utilizzati in circostanze specifiche. I manuali GNU sono anch’essi protetti da permesso d’autore, ma ne usano una versione molto più semplice, perché per i manuali non è necessaria la complessità della GPL.”

La Free Software Foundation

“Man mano che l’interesse per Emacs aumentava, altre persone parteciparono al progetto GNU, e decidemmo che era di nuovo ora di cercare finanziamenti. Così nel 1985 fondammo la Free Software Foundation (Fondazione per il software libero), una organizzazione senza fini di lucro per lo sviluppo di software libero. La FSF fra l’altro si prese carico della distribuzione dei nastri di Emacs; più tardi estese l’attività aggiungendo sul nastro altro software libero (sia GNU che non GNU) e vendendo manuali liberi.

La FSF accetta donazioni, ma gran parte delle sue entrate è sempre stata costituita dalle vendite: copie di software libero e servizi correlati. Oggi vende CD-ROM di codice sorgente, CD-ROM di programmi compilati, manuali stampati professionalmente (tutti con libertà di redistribuzione

⁴Nota di Stallman: Nel 1984 / 1985, Don Hopkins, persona molto creativa, mi mandò una lettera. Sulla busta aveva scritto diverse frasi argute, fra cui questa: “Permesso d’autore - tutti i diritti rovesciati”. Utilizzai l’espressione “permesso d’autore” per battezzare il concetto di distribuzione che allora andavo elaborando.

⁵Nota dei Traduttori: si tratta di un gioco di parole, che qui viene reso con “permesso d’autore”: copyright (diritto d’autore) è formato dalla parola “copy” (copia) e “right” (diritto, ma anche destra), opposto di “left” (sinistra, ma anche lasciato)

e modifica), e distribuzioni Deluxe (nelle quali compiliamo l'intera scelta di software per una piattaforma a richiesta).

I dipendenti della Free Software Foundation hanno scritto e curato la manutenzione di diversi pacchetti GNU. Fra questi spiccano la libreria C e la shell. La libreria C di GNU è utilizzata da ogni programma che gira su sistemi GNU/Linux per comunicare con Linux. È stata sviluppata da un membro della squadra della Free Software Foundation, Roland McGrath. La shell usata sulla maggior parte dei sistemi GNU/Linux è Bash, la Bourne Again Shell⁶⁷, che è stata sviluppata da Brian Fox, dipendente della FSF.

Finanziammo lo sviluppo di questi programmi perché il progetto GNU non riguardava solo strumenti di lavoro o un ambiente di sviluppo: il nostro obiettivo era un sistema operativo completo, e questi programmi erano necessari per raggiungere quell'obiettivo."

Il supporto per il software libero

"La filosofia del software libero rigetta una diffusa pratica commerciale in particolare, ma non è contro il commercio. Quando un'impresa rispetta la libertà dell'utente, c'è da augurarle ogni successo.

La vendita di copie di Emacs esemplifica un modo di condurre affari col software libero. Quando la FSF prese in carico quest'attività, doveti trovare un'altra fonte di sostentamento. La trovai nella vendita di servizi relativi al software libero che avevo sviluppato, come insegnare argomenti quali programmazione di Emacs e personalizzazione di GCC, oppure sviluppare software, soprattutto adattamento di GCC a nuove architetture.

Oggi tutte queste attività collegate al software libero sono esercitate da svariate aziende. Alcune distribuiscono raccolte di software libero su CD-ROM, altre offrono consulenza a diversi livelli, dall'aiutare gli utenti in difficoltà, alla correzione di errori, all'aggiunta di funzionalità non banali. Si cominciano anche a vedere aziende di software che si fondano sul lancio di nuovi programmi liberi.

Attenzione, però: diverse aziende che si fregiano del marchio "open source" (software aperto) in realtà fondano le loro attività su software non libero che funziona insieme con software libero. Queste non sono aziende di software libero, sono aziende di software proprietario i cui prodotti attirano gli utenti lontano dalla libertà. Loro li chiamano "a valore aggiunto", il che riflette i valori che a loro farebbe comodo che adottassimo: la convenienza prima della libertà. Se noi riteniamo che la libertà abbia più valore, li dovremmo chiamare prodotti "a libertà sottratta"."

Obiettivi tecnici

"L'obiettivo principale di GNU era essere software libero. Anche se GNU non avesse avuto alcun vantaggio tecnico su Unix, avrebbe avuto sia un vantaggio sociale, permettendo agli utenti di cooperare, sia un vantaggio etico, rispettando la loro libertà.

Tuttavia risultò naturale applicare al lavoro le regole classiche di buona programmazione; per esempio, allocare le strutture dati dinamicamente per evitare limitazioni arbitrarie sulla dimensione dei dati, o gestire tutti i possibili codici a 8 bit in tutti i casi ragionevoli.

Inoltre, al contrario di Unix che era pensato per piccole dimensioni di memoria, decidemmo di non supportare le macchine a 16 bit (era chiaro che le macchine a 32 bit sarebbero state la norma quando il sistema GNU sarebbe stato completo), e di non preoccuparci di ridurre l'occupazione di memoria a meno che eccedesse il megabyte. In programmi per i quali non era essenziale la gestione di file molto grandi, spingemmo i programmatori a leggere in memoria l'intero file di ingresso per poi analizzare il file senza doversi preoccupare delle operazioni di I/O.

Queste decisioni fecero sì che molti programmi GNU superassero i loro equivalenti Unix sia in affidabilità che in velocità di esecuzione."

⁶Nota di Stallman: "Bourne Again Shell" è un gioco di parole sul nome di "Bourne Shell", che era la normale shell di Unix.

⁷Nota dei Traduttori: "Bourne Again" richiama l'espressione cristiana "born again", "rinato" (in Cristo).

Donazioni di computer

“Man mano che la reputazione del progetto GNU andava crescendo, alcune persone iniziarono a donare macchine su cui girava Unix. Queste macchine erano molto utili, perché il modo più semplice di sviluppare componenti per GNU era di farlo su di un sistema Unix così da sostituire pezzo per pezzo i componenti di quel sistema. Ma queste macchine sollevavano anche una questione etica: se fosse giusto per noi anche solo possedere una copia di Unix.

Unix era (ed è) software proprietario, e la filosofia del progetto GNU diceva che non avremmo dovuto usare software proprietario. Ma, applicando lo stesso ragionamento per cui la violenza è ammessa per autodifesa, conclusi che fosse legittimo usare un pacchetto proprietario, se ciò fosse stato importante nel crearne un sostituto libero che permettesse ad altri di smettere di usare quello proprietario.

Tuttavia, benché fosse un male giustificabile, era pur sempre un male. Oggi non abbiamo più alcuna copia di Unix, perché le abbiamo sostituite con sistemi operativi liberi. Quando non fu possibile sostituire il sistema operativo di una macchina con uno libero, sostituimmo la macchina.”

L’elenco dei compiti GNU

“Mentre il progetto GNU avanzava, e un numero sempre maggiore di componenti di sistema venivano trovati o sviluppati, diventò utile stilare un elenco delle parti ancora mancanti. Usammo questo elenco per ingaggiare programmatori che scrivessero tali parti, e l’elenco prese il nome di elenco dei compiti GNU. In aggiunta ai componenti Unix mancanti inserimmo nell’elenco svariati progetti utili di programmazione o di documentazione che a nostro parere non dovrebbero mancare in un sistema operativo veramente completo.

Oggi non compare quasi nessun componente Unix nell’elenco dei compiti GNU; tutti questi lavori, a parte qualcuno non essenziale, sono già stati svolti. D’altro canto l’elenco è pieno di quei progetti che qualcuno chiamerebbe "applicazioni": ogni programma che interessi a una fetta non trascurabile di utenti sarebbe un’utile aggiunta a un sistema operativo.

L’elenco comprende anche dei giochi, e così è stato fin dall’inizio: Unix comprendeva dei giochi, perciò era naturale che così fosse anche per GNU. Ma poiché non c’erano esigenze di compatibilità per i giochi, non ci attenemmo alla scelta di giochi presenti in Unix, preferendo piuttosto fornire un elenco di diversi tipi di giochi potenzialmente graditi agli utenti.”

La licenza GNU per le librerie

“La libreria C del sistema GNU utilizza un tipo speciale di permesso d’autore, la "Licenza Pubblica GNU per le Librerie"⁸, che permette l’uso della libreria da parte di software proprietario. Perché quest’eccezione?

Non si tratta di questioni di principio: non c’è nessun principio che dica che i prodotti software proprietari abbiano il diritto di includere il nostro codice (perché contribuire a un progetto fondato sul rifiuto di condividere con noi?). L’uso della licenza LGPL per la libreria C, o per qualsiasi altra libreria, è una questione di strategia.

La libreria C svolge una funzione generica: ogni sistema operativo proprietario e ogni compilatore includono una libreria C. Di conseguenza, rendere disponibile la nostra libreria C solo per i programmi liberi non avrebbe dato nessun vantaggio a tali programmi liberi, avrebbe solo disincentivato l’uso della nostra libreria.

C’è un’eccezione a questa situazione: sul sistema GNU (termine che include GNU/Linux) l’unica libreria C disponibile è quella GNU. Quindi i termini di distribuzione della nostra libreria C determinano se sia possibile o meno compilare un programma proprietario per il sistema GNU. Non ci sono ragioni etiche per permettere l’uso di applicazioni proprietarie sul sistema GNU, ma

⁸Nota dei Traduttori: Nel 1999 la FSF ha cambiato nome alla licenza LGPL che ora si chiama "Lesser GPL", GPL attenuata, per non suggerire che si tratti della forma di licenza preferenziale per le librerie.



Figura 4.3: GNU's not Unix

strategicamente sembra che impedirne l'uso servirebbe più a scoraggiare l'uso del sistema GNU che non a incoraggiare lo sviluppo di applicazioni libere.

Ecco perché l'uso della licenza LGPL è una buona scelta strategica per la libreria C, mentre per le altre librerie la strategia va valutata caso per caso. Quando una libreria svolge una funzione particolare che può aiutare a scrivere certi tipi di programmi, distribuirla secondo la GPL, quindi limitandone l'uso ai soli programmi liberi, è un modo per aiutare gli altri autori di software libero, dando loro un vantaggio nei confronti del software proprietario.

Prendiamo come esempio GNU-Readline, una libreria scritta per fornire a Bash la modificabilità della linea di comando: Readline è distribuita secondo la normale licenza GPL, non la LGPL. Ciò probabilmente riduce l'uso di Readline, ma questo non rappresenta una perdita per noi; d'altra parte almeno una applicazione utile è stata resa software libero proprio al fine di usare Readline, e questo è un guadagno tangibile per la comunità.

Chi sviluppa software proprietario ha vantaggi economici, gli autori di programmi liberi hanno bisogno di avvantaggiarsi a vicenda. Spero che un giorno possiamo avere una grande raccolta di librerie coperte dalla licenza GPL senza che esista una raccolta equivalente per chi scrive software proprietario. Tale libreria fornirebbe utili moduli da usare come i mattoni per costruire nuovi programmi liberi, e costituendo un sostanziale vantaggio per la scrittura di ulteriori programmi liberi."

Togliersi il prurito?

"Eric Raymond afferma che "ogni buon programma nasce dall'iniziativa di un programmatore che si vuole togliere un suo personale prurito". È probabile che talvolta succeda così, ma molte parti essenziali del software GNU sono state sviluppate al fine di completare un sistema operativo libero. Derivano quindi da una idea e da un progetto, non da una necessità contingente.

Ad esempio, abbiamo sviluppato la libreria C di GNU perché un sistema di tipo Unix ha bisogno di una libreria C, la Bourne-Again Shell (bash) perché un sistema di tipo Unix ha bisogno di una shell, e GNU tar perché un sistema di tipo Unix ha bisogno un programma tar. Lo stesso vale per i miei programmi: il compilatore GNU, GNU Emacs, GDB, GNU Make.

Alcuni programmi GNU sono stati sviluppati per fronteggiare specifiche minacce alla nostra libertà: ecco perché abbiamo sviluppato gzip come sostituto per il programma Compress, che la comunità aveva perduto a causa dei brevetti sull'algoritmo LZW. Abbiamo trovato persone che sviluppassero LessTif, e più recentemente abbiamo dato vita ai progetti GNOME e Harmony per affrontare i problemi causati da alcune librerie proprietarie (come descritto più avanti).

Stiamo sviluppando la GNU Privacy Guard per sostituire i diffusi programmi di crittografia non liberi, perché gli utenti non siano costretti a scegliere tra riservatezza e libertà.

Naturalmente, i redattori di questi programmi sono coinvolti nel loro lavoro, e varie persone vi hanno aggiunto diverse funzionalità secondo le loro personali necessità e i loro interessi. Tuttavia non è questa la ragione dell'esistenza di tali programmi."

Sviluppi inattesi

"All'inizio del progetto GNU pensavo che avremmo sviluppato l'intero sistema GNU e poi lo avremmo reso disponibile tutto insieme, ma le cose non andarono così.

Poiché i componenti del sistema GNU sono stati implementati su un sistema Unix, ognuno di essi poteva girare su sistemi Unix molto prima che esistesse un sistema GNU completo. Alcuni di questi programmi divennero diffusi e gli utenti iniziarono a estenderli e a renderli utilizzabili su nuovi sistemi: sulle varie versioni di Unix, incompatibili tra loro, e talvolta anche su altri sistemi.

Questo processo rese tali programmi molto più potenti e attirò finanziamenti e collaboratori al progetto GNU; tuttavia probabilmente ritardò di alcuni anni la realizzazione di un sistema minimo funzionante, perché il tempo degli autori GNU veniva impiegato a curare la compatibilità di questi programmi con altri sistemi e ad aggiungere nuove funzionalità ai componenti esistenti, piuttosto che a proseguire nella scrittura di nuovi componenti."

GNU-Hurd

"Nel 1990 il sistema GNU era quasi completo, l'unica parte significativa ancora mancante era il kernel. Avevamo deciso di implementare il nostro kernel come un gruppo di processi server che girassero sul sistema Mach. Mach è un microkernel sviluppato alla Carnegie Mellon University e successivamente all'Università dello Utah; GNU Hurd è un gruppo di server (o "herd of gnus": mandria di gnu) che gira su Mach svolgendo le funzioni del kernel Unix. L'inizio dello sviluppo fu ritardato nell'attesa che Mach fosse reso disponibile come software libero, come era stato promesso.

Una ragione di questa scelta progettuale fu di evitare quella che sembrava la parte più complessa del lavoro: effettuare il debugging del kernel senza un debugger a livello sorgente. Questo lavoro era già stato fatto, appunto in Mach, e avevamo previsto di effettuare il debugging dei server Hurd come programmi utente, con GDB. Ma questa fase si rivelò molto lunga, e il debugging dei server multi-thread che si scambiano messaggi si è rivelato estremamente complesso. Per rendere Hurd robusto furono così necessari molti anni."

Alix

"Originariamente il kernel GNU non avrebbe dovuto chiamarsi Hurd; il suo nome originale era Alix, come la donna di cui ero innamorato in quel periodo. Alix, che era amministratrice di sistemi Unix, aveva sottolineato come il suo nome corrispondesse a un comune schema usato per battezzare le versioni del sistema Unix: scherzosamente diceva ai suoi amici: "qualcuno dovrebbe chiamare un kernel come me". Io non dissi nulla ma decisi di farle una sorpresa scrivendo un kernel chiamato Alix.

Le cose non andarono così. Michael Bushnell (ora Thomas), principale autore del kernel, preferì il nome Hurd, e chiamò Alix una parte del kernel, quella che serviva a intercettare le chiamate di sistema e a gestirle inviando messaggi ai server che compongono HURD.

Infine io e Alix ci lasciammo e lei cambiò nome; contemporaneamente la struttura di Hurd veniva cambiata in modo che la libreria C mandasse messaggi direttamente ai server, e così il componente Alix scomparve dal progetto. Prima che questo accadesse, però, un amico di Alix si accorse della presenza del suo nome nel codice sorgente di Hurd e glielo disse. Così il nome raggiunse il suo scopo."

Linux e GNU/Linux

“GNU Hurd non è pronto per un uso non sperimentale, ma per fortuna è disponibile un altro kernel: nel 1991 Linus Torvalds sviluppò un Kernel compatibile con Unix e lo chiamò Linux. Attorno al 1992, la combinazione di Linux con il sistema GNU ancora incompleto produsse un sistema operativo libero completo (naturalmente combinarli fu un notevole lavoro di per sé). È grazie a Linux che oggi possiamo utilizzare una versione del sistema GNU.

Chiamiamo GNU/Linux questa versione del sistema, per indicare la sua composizione come una combinazione del sistema GNU col kernel Linux.”

Le sfide che ci aspettano

“Abbiamo dimostrato la nostra capacità di sviluppare un’ampia gamma di software libero, ma questo non significa che siamo invincibili e inarrestabili. Diverse sfide rendono incerto il futuro del software libero, e affrontarle richiederà perseveranza e sforzi costanti, talvolta per anni. Sarà necessaria quella determinazione che le persone sanno dimostrare quando danno valore alla propria libertà e non permettono a nessuno di sottrargliela. Le quattro sezioni seguenti parlano di queste sfide.”

1. Hardware segreto

“Sempre più spesso, i costruttori di hardware tendono a mantenere segrete le specifiche delle loro apparecchiature; questo rende difficile la scrittura di driver liberi che permettano a Linux e XFree86 di supportare nuove periferiche. Anche se oggi abbiamo sistemi completamente liberi, potremmo non averli domani se non saremo in grado di supportare i calcolatori di domani.

Esistono due modi per affrontare il problema. Un programmatore può ricostruire le specifiche dell’hardware usando tecniche di reverse engineering. Oppure si può scegliere hardware supportato dai programmi liberi: man mano che il nostro numero aumenta, la segretezza delle specifiche diventerà una pratica controproducente.

Il reverse engineering è difficile: avremo programmatori sufficientemente determinati da dedicarvisi? Sì, se avremo costruito una forte consapevolezza che avere programmi liberi sia una questione di principio e che i driver non liberi non sono accettabili. E succederà che molti di noi accettino di spendere un po’ di più o perdere un po’ più di tempo per poter usare driver liberi? Sì, se il desiderio di libertà e la determinazione a ottenerla saranno diffusi.”

2. Librerie non libere

“Una libreria non libera che giri su sistemi operativi liberi funziona come una trappola per i creatori di programmi liberi. Le funzionalità attraenti della libreria fungono da esca; chi usa la libreria cade nella trappola, perché il programma che crea è inutile come parte di un sistema operativo libero (a rigore, il programma potrebbe esservi incluso, ma non funzionerebbe, visto che manca la libreria). Peggio ancora, se un programma che usa la libreria proprietaria diventa diffuso, può attirare altri ignari programmatori nella trappola.

Il problema si concretizzò per la prima volta con la libreria Motif, negli anni ‘80. Sebbene non ci fossero ancora sistemi operativi liberi, i problemi che Motif avrebbe causato loro erano già chiari. Il progetto GNU reagì in due modi: interessandosi presso diversi progetti di software libero perché supportassero gli strumenti grafici X liberi in aggiunta a Motif, e cercando qualcuno che scrivesse un sostituto libero di Motif. Il lavoro richiese molti anni: solo nel 1997 LessTif, sviluppato dagli "Hungry Programmers", divenne abbastanza potente da supportare la maggior parte delle applicazioni Motif. Tra il 1996 e il 1998 un’altra libreria non libera di strumenti grafici, chiamata Qt, veniva usata in una significativa raccolta di software libero: l’ambiente grafico KDE.

I sistemi liberi GNU/Linux non potevano usare KDE, perché non potevamo usare la libreria; tuttavia, alcuni distributori commerciali di sistemi GNU/Linux, non scrupolosi nell’attenersi solo ai programmi liberi, aggiunsero KDE ai loro sistemi, ottenendo così sistemi che offrivano più

funzionalità, ma meno libertà. Il gruppo che sviluppava KDE incoraggiava esplicitamente altri programmatori a usare Qt, e milioni di nuovi "utenti Linux" non sospettavano minimamente che questo potesse costituire un problema. La situazione si faceva pericolosa. La comunità del software libero affrontò il problema in due modi: GNOME e Harmony.

GNOME (GNU Network Object Model Environment, modello di ambiente per oggetti di rete) è il progetto GNU per l'ambiente grafico (desktop). Intrapreso nel 1997 da Miguel de Icaza e sviluppato con il supporto di Red Hat Software, GNOME si ripromise di fornire funzionalità grafiche simili a quelle di KDE, ma usando esclusivamente software libero. GNOME offre anche dei vantaggi tecnici, come il supporto per svariati linguaggi di programmazione, non solo il C++. Ma il suo scopo principale era la libertà: non richiedere l'uso di alcun programma che non fosse libero.

Harmony è una libreria compatibile con Qt, progettata per rendere possibile l'uso del software KDE senza dover usare Qt. Nel novembre 1998 gli autori di Qt annunciarono un cambiamento di licenza che, una volta operativo, avrebbe reso Qt software libero. Non c'è modo di esserne certi, ma credo che questo fu in parte dovuto alla decisa risposta della comunità al problema posto da Qt quando non era libero (la nuova licenza è scomoda e iniqua, per cui rimane comunque preferibile evitare l'uso di Qt).

Come risponderemo alla prossima allettante libreria non libera? Riuscirà la comunità in toto a comprendere l'importanza di evitare la trappola? Oppure molti di noi preferiranno la convenienza alla libertà, creando così ancora un grave problema? Il nostro futuro dipende dalla nostra filosofia."

3. Brevetti sul software

"Il maggior pericolo a cui ci troviamo di fronte è quello dei brevetti sul software, che possono rendere inaccessibili al software libero algoritmi e funzionalità per un tempo che può estendersi fino a vent'anni. I brevetti sugli algoritmi di compressione LZW furono depositati nel 1983, e ancor oggi non possiamo distribuire programmi liberi che producano immagini GIF compresse. Nel 1998 un programma libero per produrre audio compresso MP3 venne ritirato sotto minaccia di una causa per violazione di brevetto. Ci sono modi per affrontare la questione brevetti: possiamo cercare prove che un brevetto non sia valido oppure possiamo cercare modi alternativi per ottenere lo stesso risultato. Ognuna di queste tecniche, però, funziona solo in certe circostanze; quando entrambe falliscono un brevetto può obbligare tutto il software libero a rinunciare a qualche funzionalità che gli utenti desiderano. Cosa dobbiamo fare quando ciò accade?"

Chi fra noi apprezza il software libero per il valore della libertà rimarrà comunque dalla parte dei programmi liberi; saremo in grado di svolgere il nostro lavoro senza le funzionalità coperte da brevetto. Ma coloro che apprezzano il software libero perché si aspettano che sia tecnicamente superiore probabilmente grideranno al fallimento quando un brevetto ne impedisce lo sviluppo. Perciò, nonostante sia utile parlare dell'efficacia pratica del modello di sviluppo "a cattedrale", e dell'affidabilità e della potenza di un dato programma libero, non ci dobbiamo fermare qui; dobbiamo parlare di libertà e di principi."

4. Documentazione libera

"La più grande carenza nei nostri sistemi operativi liberi non è nel software, quanto nella carenza di buoni manuali liberi da includere nei nostri sistemi. La documentazione è una parte essenziale di qualunque pacchetto software; quando un importante pacchetto software libero non viene accompagnato da un buon manuale libero si tratta di una grossa lacuna. E di queste lacune attualmente ne abbiamo molte.

La documentazione libera, come il software libero, è una questione di libertà, non di prezzo. Il criterio per definire libero un manuale è fondamentalmente lo stesso che per definire libero un programma: si tratta di offrire certe libertà a tutti gli utenti. Deve essere permessa la ridistribuzione (compresa la vendita commerciale), sia in formato elettronico che cartaceo, in modo che il manuale possa accompagnare ogni copia del programma.

Autorizzare la modifica è anch'esso un aspetto cruciale; in generale, non credo sia essenziale permettere alle persone di modificare articoli e libri di qualsiasi tipo. Per esempio, non credo che voi o io dobbiamo sentirci in dovere di autorizzare la modifica di articoli come questo, articoli che descrivono le nostre azioni e il nostro punto di vista.

Ma c'è una ragione particolare per cui la libertà di modifica è cruciale per la documentazione dei programmi liberi. Quando qualcuno esercita il proprio diritto di modificare il programma, aumentandone o alterandone le funzionalità, se è coscienzioso modificherà anche il manuale, in modo da poter fornire una documentazione utile e accurata insieme al programma modificato. Un manuale che non permetta ai programmatori di essere coscienziosi e completare il loro lavoro non soddisfa i bisogni della nostra comunità.

Alcuni limiti sulla modificabilità non pongono alcun problema; per esempio, le richieste di conservare la nota di copyright dell'autore originale, i termini di distribuzione e la lista degli autori vanno bene. Non ci sono problemi nemmeno nel richiedere che le versioni modificate dichiarino esplicitamente di essere tali, così pure che intere sezioni non possano essere rimosse o modificate, finché queste sezioni vertono su questioni non tecniche. Restrizioni di questo tipo non creano problemi perché non impediscono al programmatore coscienzioso di adattare il manuale perché rispecchi il programma modificato. In altre parole, non impediscono alla comunità del software libero di beneficiare appieno del manuale.

D'altro canto, deve essere possibile modificare tutto il contenuto tecnico del manuale e poter distribuire il risultato in tutti i formati usuali, attraverso tutti i normali canali di distribuzione; diversamente, le restrizioni creerebbero un ostacolo per la comunità, il manuale non sarebbe libero e avremmo bisogno di un altro manuale.

Gli sviluppatori di software libero avranno la consapevolezza e la determinazione necessarie a produrre un'intera gamma di manuali liberi? Ancora una volta, il nostro futuro dipende dalla nostra filosofia."

Dobbiamo parlare di libertà

"Stime recenti valutano in dieci milioni il numero di utenti di sistemi quali Debian GNU/Linux e Red Hat Linux. Il software libero ha creato tali vantaggi pratici che gli utenti stanno approdando a esso per pure ragioni pratiche.

Gli effetti positivi di questa situazione sono evidenti: maggior interesse a sviluppare software libero, più clienti per le imprese di software libero e una migliore capacità di incoraggiare le aziende a sviluppare software commerciale libero invece che prodotti software proprietari.

L'interesse per il software, però, sta crescendo più in fretta della coscienza della filosofia su cui è basato, e questa disparità causa problemi. La nostra capacità di fronteggiare le sfide e le minacce descritte in precedenza dipende dalla determinazione nell'essere impegnati per la libertà. Per essere sicuri che la nostra comunità abbia tale determinazione, dobbiamo diffondere l'idea presso i nuovi utenti man mano che entrano a far parte della comunità.

Ma in questo stiamo fallendo: gli sforzi per attrarre nuovi utenti nella comunità sono di gran lunga maggiori degli sforzi per l'educazione civica della comunità stessa. Dobbiamo fare entrambe le cose, e dobbiamo mantenere un equilibrio fra i due impegni."

"Open Source"

"Parlare di libertà ai nuovi utenti è diventato più difficile dal 1998, quando una parte della comunità decise di smettere di usare il termine "free software" e usare al suo posto "open source". Alcune delle persone che suggerirono questo termine intendevano evitare che si confondesse "free" con "gratis", un valido obiettivo. D'altra parte, altre persone intendevano mettere da parte lo spirito del principio che aveva dato la spinta al movimento del software libero e al progetto GNU, puntando invece ad attrarre i dirigenti e gli utenti commerciali, molti dei quali afferiscono a una ideologia che pone il profitto al di sopra della libertà, della comunità, dei principi. Perciò la retorica di "open source" si focalizza sul possibilità di creare software di buona qualità e

potente ma evita deliberatamente le idee di libertà, comunità, principio. Le riviste che si chiamano "Linux..." sono un chiaro esempio di ciò: sono piene di pubblicità di software proprietario che gira sotto GNU/Linux; quando ci sarà il prossimo Motif o Qt, queste riviste avvertiranno i programmatori di starne lontano o accetteranno la sua pubblicità? L'appoggio delle aziende può contribuire alla comunità in molti modi; a parità di tutto il resto è una cosa utile. Ma ottenere questo appoggio parlando ancor meno di libertà e principi può essere disastroso; rende ancora peggiore lo sbilanciamento descritto tra diffusione ed educazione civica. "Software libero" (free software) e "sorgente aperto" (open source) descrivono più o meno la stessa categoria di software, ma dicono cose differenti sul software e sui valori. Il progetto GNU continua a usare il termine "software libero" per esprimere l'idea che la libertà sia importante, non solo la tecnologia."

Prova!

"La filosofia di Yoda ("Non c'è provare") suona bene, ma per me non funziona. Ho fatto la maggior parte del mio lavoro angustiato dal timore di non essere in grado di svolgere il mio compito e nel dubbio, se fossi riuscito, che non fosse sufficiente per raggiungere l'obiettivo. Ma ci ho provato in ogni caso perché nessuno tranne me si poneva tra il nemico e la mia città. Sorprendendo me stesso, qualche volta sono riuscito.

A volte ho fallito, alcune delle mie città sono cadute; poi ho trovato un'altra città minacciata e mi sono preparato a un'altra battaglia. Con l'andar del tempo ho imparato a cercare le possibili minacce e a mettermi tra loro e la mia città, facendo appello ad altri hacker perché venissero e si unissero a me.

Oggi giorno spesso non sono da solo. È un sollievo e una gioia quando vedo un reggimento di hacker che scavano trincee per difendere il confine e quando mi rendo conto che questa città può sopravvivere; per ora. Ma i pericoli diventano più grandi ogni anno, e ora Microsoft ha esplicitamente preso di mira la nostra comunità. Non possiamo dare per scontato il futuro della libertà; non diamolo per scontato! Se volete mantenere la vostra libertà dovete essere pronti a difenderla."

4.2 Open Sources Definition

di Bruce Perens

"Il tipico utente di computer possiede una discreta quantità di software che ha comprato nel tempo e che ormai non adopera più. Magari ha aggiornato il computer, o ha cambiato marca, e i vecchi programmi hanno smesso di funzionare. Magari il software è diventato obsoleto. O semplicemente il programma non lo soddisfa. Forse ha comprato due o più computer e non vuole pagare per una seconda copia del software. Quale che sia la ragione, il software per cui ha pagato anni fa non è più adeguato.

Tutto questo è inevitabile? Non sarebbe bello avere diritto a un aggiornamento gratuito ogni volta che il software lo richiede? Se, passando da un Mac a un PC, si potesse cambiare la versione del software gratis? Se, quando il software non funziona o non è abbastanza potente, si potesse migliorarlo o perfino ripararlo da soli? Se il software continuasse a essere supportato anche quando l'azienda produttrice abbia cessato l'attività? Non sarebbe bello usare il software sulla workstation al lavoro, sul computer di casa e sul portatile, anziché su un solo computer? È probabile che, in quel caso, si starebbe ancora usando il software pagato anni prima. Questi sono alcuni dei diritti che l'Open Source riconosce. La Open Source Definition è una carta dei diritti dell'utente di computer. Definisce certi diritti che una licenza software deve garantire per poter essere certificata come Open Source. I produttori che non rendono Open Source il loro programmi trovano difficile competere con chi lo fa, dal momento che gli utenti imparano ad apprezzare quei diritti che avrebbero dovuto sempre essere loro. Programmi come il sistema operativo Linux e il browser Web Netscape sono diventati popolarissimi, scalzando altro software sottoposto a licenze più restrittive. Aziende che usano software Open Source godono il vantaggio del suo

rapidissimo sviluppo, spesso a opera cooperativa di numerose aziende, e in gran parte fornito da soggetti individuali che operano migliorie sulla base di proprie necessità.

I volontari che hanno reso possibili prodotti come Linux ci sono, e le aziende possono cooperare, solo grazie ai diritti che vengono con l'Open Source. Il programmatore medio si sentirebbe stupido nel riversare tanto lavoro in un programma per vedere poi le sue migliorie vendute dal proprietario senza che lui ne riceva alcun ritorno. Quegli stessi programmatori contribuiscono volentieri all'Open Source perché esso assicura loro questi diritti:

- il diritto di fare copie del programma e di distribuirle;
- il diritto d'accesso al codice sorgente del software, condizione necessaria per poterlo modificare;
- il diritto di apportare migliorie al programma.

Questi diritti sono importanti per coloro che collaborano a un software perché li mantengono tutti al medesimo livello. Chiunque lo voglia è libero di vendere un programma Open Source, così i prezzi rimarranno bassi e sarà rapido lo sviluppo per raggiungere nuovi mercati. Chiunque investa il suo tempo a costruire conoscenza in un programma Open Source lo può supportare, e questo permette agli utenti la possibilità di fornire a loro volta supporto, o l'economia dovuta a un gran numero di fornitori di supporto concorrenti. Qualunque programmatore può adattare un programma Open Source a misura di mercati specifici per raggiungere clienti nuovi. Chi lo fa non è costretto a pagare diritti o concessioni di licenza. La ragione per il successo di una strategia che può suonare alquanto comunista proprio nel momento in cui il fallimento del comunismo stesso è visibile ovunque, è nel fatto che l'economia dell'informazione è sostanzialmente diversa da quella degli altri prodotti. I costi della copia di un'informazione come un programma software è infinitesimo. L'elettricità non costa quasi nulla, l'uso dell'attrezzatura poco di più. È come, per fare un paragone, se si duplicasse una pagnotta usando un solo etto di farina."

La storia

"Il concetto di free software non è nuovo. Quando le università cominciarono ad adottare i computer, essi erano strumenti per la ricerca. Il software veniva scambiato liberamente e i programmatori venivano pagati per l'atto della programmazione, non per i programmi in sé. Solo più tardi, quando il mondo degli affari e del commercio adottò i computer, i programmatori cominciarono a mantenersi limitando i diritti d'uso del loro software e facendosi pagare per ogni copia. Il free software come idea politica è stato reso popolare da Richard Stallman dal 1984, allorché formò la Free Software Foundation e il progetto GNU a essa collegato. La premessa di Stallman è che la gente dovrebbe avere più libertà e dovrebbe imparare ad apprezzarla. Egli progettò un insieme di diritti che sentiva necessari a ogni utente e li codificò nella GNU Public License o GPL. Stallman battezzò scherzosamente la sua licenza copyleft in quanto lasciava intatto il diritto alla copia. Stallman stesso sviluppò lavori fondamentali di free software quali il compilatore C GNU e GNU Emacs, un editor di testo che alcuni hanno trovato così seducente da concepirne quasi un culto. Il suo lavoro ispirò molti altri a fornire free software sotto la GPL. Per quanto non promossa con il medesimo fervore libertario, la Open Source Definition include molte delle idee di Stallman, e può ben considerarsi un derivato della sua opera.

La Open Source Definition cominciò la sua vita come un documento di linea di condotta della distribuzione Debian GNU/Linux. Debian, uno dei primi sistemi Linux, tuttora popolare, fu costruito interamente con free software. Tuttavia, dal momento che c'erano altre licenze oltre al copyleft che comportavano la gratuità, Debian ebbe qualche problema nel definire che cosa fosse gratis, e i produttori non resero mai chiara la loro politica di free software al resto del mondo. All'epoca, trovandomi a capo del progetto Debian, affrontai questi problemi proponendo un Contratto Sociale Debian e la Guida Debian del Free Software, nel luglio del 1997. Molti sviluppatori Debian inviarono critiche e miglioramenti che io incorporai nei documenti. Il Contratto Sociale

documentava l'intenzione di Debian di costituire il proprio sistema interamente con free software, e la Guida rendeva facilmente possibile la classificazione del software come free software o meno, confrontando la licenza software con la guida stessa.

La Guida Debian fu oggetto di molte lodi nella comunità del free software, specialmente fra gli sviluppatori Linux, che a quel tempo stavano preparando la loro propria rivoluzione software sviluppando il primo vero sistema operativo gratuito. Quando Netscape decise di rendere libero il suo browser Web, contattò Eric Raymond. Raymond, la Margaret Mead del free software, è autore di numerosi articoli di taglio antropologico che illustrano il fenomeno del free software e la cultura che vi è cresciuta intorno: scritti che furono i primi di un genere e che hanno messo sotto la luce dei riflettori questo fenomeno fino ad allora oscuro. La dirigenza di Netscape rimase suggestionata in particolare dal saggio di Raymond "La cattedrale e il bazaar", la cronaca di uno sviluppo free software coronato da successo con volontari non pagati, e gli chiese una consulenza, sotto patto di riservatezza, mentre sviluppavano una licenza per il loro free software. Raymond insisté che la licenza di Netscape dovesse adeguarsi alla guida Debian per poter essere presa sul serio come free software. Raymond e io ci eravamo incontrati qualche volta all'Hacker Conference, una raduno su invito di programmatori creativi e non convenzionali. Avevamo corrisposto via email su vari argomenti. Mi contattò nel febbraio del 1997 con l'idea per l'Open Source. Raymond temeva che la mentalità conservatrice dell'ambiente degli affari venisse scoraggiata dal grado di libertà di Stallman, che era al contrario popolarissimo fra i programmatori di mentalità più liberale. Era impressione di Raymond che ciò stesse sclerotizzando lo sviluppo di Linux nel mondo business laddove esso fioriva invece nell'ambiente della ricerca. Raymond ebbe incontri con uomini d'affari nell'industria Linux che stava muovendo solo allora i primi passi; insieme, essi concepirono un programma di marketing del free software indirizzato ai colletti bianchi. Furono coinvolti Larry Augustin di VA Research e Sam Ockman (che abbandonò più tardi VA per formare Penguin Computing), nonché altri non di mia conoscenza.

Alcuni mesi prima dell'Open Source, avevo elaborato l'idea dell'Open Hardware, concetto simile rivolto agli strumenti hardware e alle loro interfacce anziché ai programmi software. A tutt'oggi l'Open Hardware non ha avuto il successo dell'Open Source, ma il progetto è ancora attivo; se ne può sapere di più a <http://www.openhardware.org>.

Secondo Raymond, la Guida Debian era il documento più adatto a definire l'Open Source, ma serviva una denominazione più generale e la rimozione dei riferimenti specifici a Debian. Modificai la Guida Debian fino a ricavarne la Open Source Definition. Avevo formato per Debian un ente chiamato Software in the Public Interest, e mi offrii di registrare un marchio per Open Source in modo da poter associare il suo uso alla definizione. Raymond acconsentì, e io registrai una certificazione (una forma speciale di marchio che potesse applicarsi secondo i termini ai prodotti altrui). Circa un mese dopo la registrazione del marchio, apparve chiaro che Software in the Public Interest avrebbe potuto non essere la dimora migliore per il marchio Open Source, e trasferii dunque la proprietà del marchio a Raymond. Raymond e io abbiamo da allora formato la Open Source Initiative, un'organizzazione esclusivamente destinata alla gestione della campagna Open Source e della sua certificazione di marchio. Mentre scrivo, l'iniziativa Open Source è retta da un comitato di sei componenti scelti fra fornitori di free software di chiara fama, e sta cercando di espandere il suo comitato fino a una decina di persone.

Al momento del suo concepimento, la campagna Open Source fu oggetto di molte critiche perfino da parte del contingente Linux che già aveva approvato il concetto di free software. Molti rilevarono che il termine Open Source era già in uso nel ramo della raccolta di dati per le campagne politiche. Altri pensarono che il termine Open fosse già usurato. Per altri ancora era preferibile il nome Free Software, già consolidato. Io opinai che l'abuso del termine Open sarebbe stato sempre meglio dell'ambiguità di free nella lingua inglese, in cui sta a significare tanto libero quanto gratuito, la seconda accezione essendo di gran lunga la più comune nel mondo del commercio di computer e di software. Più tardi, Richard Stallman obiettò alla mancanza di enfasi sulla libertà che secondo lui la campagna dimostrava, e al fatto che, mentre l'Open Source acquistava popolarità, il suo ruolo nella genesi del free software, e quello della sua Free Software Foundation, venivano ignorati: si lamentò di essere stato "cassato dalla storia". Peggiorò la situazione la tendenza degli operatori del settore di contrapporre Raymond a Stallman, quasi es-

si proponessero filosofie concorrenti anziché, sia pur con metodi diversi, propagandare lo stesso concetto. Io stesso contribuì probabilmente a esacerbare gli animi mettendo Stallman e Raymond l'uno contro l'altro in dibattiti pubblici alla Linux Expo e alla Open Source Expo. Caratterizzare i due come avversari diventò un'attività tanto consueta che una discussione via email, non destinata alla pubblicazione, apparve sul periodico on-line Salon. A quel punto, chiesi a Raymond di moderare i toni di un dialogo in cui, per la verità, egli non aveva mai inteso entrare.

Quando la Open Source Definition fu scritta, esisteva già un gran numero di prodotti che potevano rientrare nella categoria. Il problema erano quei programmi che non vi rientravano, e che pure gli utenti trovavano irresistibili."

KDE, Qt e Troll Tech

"Il caso di KDE, Qt e Troll Tech è pertinente a questo saggio perché il gruppo KDE e Troll Tech cercarono di porre un prodotto non-Open Source entro l'infrastruttura di Linux, incontrando una resistenza inattesa. Le grida di pubblico scandalo e la minaccia che il loro prodotto venisse rimpiazzato da un altro, completamente Open Source, convinse alla fine Troll Tech a convertirsi a una licenza pienamente Open Source. È un segno interessante dell'accoglienza entusiastica riservata dalla comunità alla Open Source Definition il fatto che Troll dovette adeguare la propria licenza, pena l'insuccesso del suo prodotto. KDE fu il primo esperimento di un desktop grafico gratuito per Linux. Le applicazioni KDE erano esse stesse sotto GPL, ma dipendevano da una libreria grafica proprietaria nota come Qt, di Troll Tech. I termini della licenza di Qt ne proibivano la modifica o l'uso con qualunque display software che non fosse il senescente X Window System. Ogni uso diverso richiedeva allo sviluppatore una licenza del costo di 1500 dollari. Troll Tech fornì versioni di Qt per Windows di Microsoft e per Macintosh, e questa fu la sua principale fonte d'entrate. La licenza pseudo-gratuita per i sistemi X intendeva indirizzare i contributi degli sviluppatori Linux verso demo, esempi e accessori per i loro costosi prodotti Windows e Mac.

Per quanto i problemi della licenza di Qt apparissero evidenti, la prospettiva di un desktop grafico per Linux era così attraente che molti utenti furono disposti a chiudere un occhio sulla sua natura non-Open Source. I promotori di Open Source trovarono che KDE fosse in difetto perché avevano l'impressione che gli sviluppatori stessero cercando di confondere la definizione di free software allo scopo di includervi elementi solo parzialmente gratuiti, come Qt. Gli sviluppatori KDE replicarono che i loro programmi erano Open Source, anche se non esistevano versioni eseguibili di quei programmi che non richiedessero una libreria non-Open Source. Io e altri sostenemmo che le applicazioni KDE non erano che frammenti Open Source di programmi non-Open Source, e che una versione Open Source di Qt sarebbe stata necessaria prima che ci si potesse riferire a KDE come a un Open Source.

Gli sviluppatori KDE tentarono di risolvere parzialmente il problema della licenza di Qt negoziando con Troll Tech un accordo (KDE Free Qt Foundation) in cui Troll e KDE avrebbero congiuntamente controllato i rilasci delle versioni gratuite di Qt, e Troll Tech avrebbe rilasciato Qt sotto una licenza conforme a Open Source nel caso che l'azienda venisse acquisita o cessasse l'attività.

Un altro gruppo diede inizio al progetto GNOME, un concorrente interamente Open Source di KDE che mirava a fornire maggiori funzioni e sofisticazioni; un gruppo separato avviò il progetto Harmony per produrre un clone di Qt completamente Open Source che avrebbe supportato KDE. Mentre le dimostrazioni di GNOME avvenivano fra il plauso e Harmony stava per diventare usabile, Troll Tech capì che Qt non avrebbe riscosso successo nel mondo Linux se non avesse cambiato licenza. Troll Tech rilasciò dunque una licenza interamente Open Source per Qt, disinnescando il conflitto ed eliminando i motivi alla base del progetto Harmony. Il progetto GNOME continua tuttora, volto adesso a un KDE migliore in termini di funzionalità e di raffinatezza piuttosto che in termini di licenza.

Prima di rilasciare la sua nuova licenza Open Source, Troll Tech me ne fece avere copia perché la verificassi, con la preghiera che rimanesse riservata finché non fossero in grado di annunciarla.

Nel mio entusiasmo di far pace con il gruppo KDE e in un imbarazzante gesto di autoinganno,

preannunciai con otto ore di anticipo la licenza su una mailing list KDE. Quell'email, per il mio rimorso, fu raccolta immediatamente da Slashdot e da altre riviste online.

La nuova licenza Troll Tech è notevole perché approfitta di una scappatoia nella Open Source Definition che permette ai file patch di essere trattati diversamente dall'altro software. Vorrei provvedere a chiudere questa scappatoia in una revisione a venire della Open Source Definition, ma il nuovo testo non dovrebbe tuttavia collocare Qt al di fuori dell'Open Source.

Al momento in cui scrivo, i promotori di Open Source stanno crescendo in misura esponenziale. I recenti contributi Open Source di IBM e di Ericsson hanno fatto i titoli dei giornali. Due distribuzioni Linux, Yggdrasil e Debian, stanno rilasciando distribuzioni di sistemi Linux completi, ivi incluse molte applicazioni che sono interamente Open Source; e molte altre, fra cui Red Hat, ci sono assai vicine. Quando il sistema GNOME sarà completo, sarà stato realizzato un sistema operativo con desktop GUI Open Source in grado di competere con Microsoft NT."

Analisi della Open Source Definition

"Questa sezione presenta nella sua interezza il testo della Open Source Definition, corredata di commenti (in corsivo). La versione canonica della Open Source Definition si trova a <http://www.opensource.org/osd>.

Alcuni pedanti hanno voluto trovare delle ambiguità di poco conto nella Open Source Definition. Mi sono astenuto da rivederla dal momento che ha poco più d'un anno di vita e vorrei che il pubblico la considerasse stabile. Il futuro imporrà qualche adeguamento lessicale, ma quasi nessuna modifica allo scopo del documento."

La Open Source Definition

"Open Source non significa solo accesso al codice sorgente. I termini di distribuzione di un programma Open Source devono essere consoni ai criteri seguenti.

Si noti che la Open Source Definition non è propriamente una licenza software. È una specifica di quanto è ammesso in una licenza software perché vi si possa riferire come a un'Open Source. La Open Source Definition non è intesa per essere un documento di valore legale. L'inclusione della Open Source Definition nelle licenze software, quale quella proposta per il Progetto di Documentazione di Linux, sembra suggerirmi la stesura di una versione più rigorosa che sia appropriata per quell'uso.

Ai fini dell'Open Source, devono applicarsi insieme tutti i termini che seguono, in tutti i casi. Per esempio, devono applicarsi alle versioni derivate di un programma così come al programma originale. Non è sufficiente applicarne alcune e non altre, e non è sufficiente se i termini non vengono applicati sistematicamente. Dopo aver dovuto affrontare delle interpretazioni particolarmente "semplici" della Open Source Definition, sono tentato di aggiungere: sto dicendo a voi!

1. Ridistribuzione libera

La licenza non può impedire ad alcuna parte in causa la vendita o la cessione del software come componente di una distribuzione di software aggregato che contenga programmi provenienti da sorgenti diverse. La licenza non può richiedere diritti o il pagamento di altre concessioni per tale vendita. Questo significa che potete fare tutte le copie che volete del software e venderle o cederle, e non dovete pagare nessuno per questo privilegio. L'espressione "distribuzione di software aggregato che contenga programmi provenienti da sorgenti diverse" era intesa a chiudere una scappatoia nella Licenza Artistica - una licenza piuttosto malfatta, a mio parere - escogitata in origine per il Perl. Oggi, quasi tutti i programmi che usano la Licenza Artistica sono disponibili anche sotto GPL. Quella clausola non è più necessaria e sarà probabilmente tolta da una futura versione della Open Source Definition.

2. Codice sorgente

Il programma deve includere il codice sorgente e deve consentire la distribuzione tanto in codice sorgente che in forma compilata. Laddove una qualunque forma del prodotto non sia distribuita corredata del codice sorgente, devono essere disponibili mezzi ben pubblicizzati per scaricare il codice sorgente, senza costi addizionali, via Internet. Il codice sorgente deve essere la forma preferenziale nella quale un programmatore modifichi un programma. Codice deliberatamente offuscato non è ammesso. Forme intermedie quali l'output di un preprocessore o di un traduttore non sono ammesse. Il codice sorgente è un preliminare necessario alla riparazione o alla modifica di un programma. L'intento qui è che il codice sorgente sia distribuito con l'opera iniziale e con tutte le opere derivate.

3. Opere derivate

La licenza deve permettere modifiche e opere derivate e deve consentire la loro distribuzione sotto i medesimi termini della licenza del software originale. Il software serve a poco se non se ne può fare la manutenzione (riparazione dei bug, porting su nuovi sistemi, migliorie) e la modifica è indispensabile alla manutenzione. L'intento è qui di permettere modifiche d'ogni sorta. Deve essere permessa la distribuzione di un'opera modificata sotto gli stessi termini di licenza dell'opera originale. Tuttavia, non è richiesto che ogni produttore di un'opera derivata debba usare gli stessi termini di licenza, ma solo che possa farlo qualora lo voglia. Diverse licenze si esprimono diversamente in materia: la licenza BSD vi permette di mantenere private le modifiche, la GPL no. Alcuni autori di software ritengono che questa clausola possa consentire a persone prive di scrupoli di modificare il loro software in maniera che possa causare imbarazzo all'autore originale. Quello che temono è che qualcuno possa deliberatamente provocare un malfunzionamento del software in modo che l'autore originale appaia un programmatore scadente. Altri paventano un possibile uso criminale del software tramite l'aggiunta di funzioni-cavallo di Troia o di tecnologie illegali in alcuni Paesi, come la crittografia. Tutti questi atti, tuttavia, sono coperti dal codice penale. Un comune fraintendimento a proposito delle licenze è che esse debbano specificare ogni cosa, per esempio "questo software non va usato per compiere delitti". Dovrebbe tuttavia essere chiaro che nessuna licenza ha esistenza valida al di fuori del corpo del diritto civile e penale. Considerare una licenza come qualcosa separato dal corpo delle leggi applicabili è tanto sciocco quanto considerare un documento in lingua inglese separato dal vocabolario di quella lingua, un caso in cui nessuna parola avrebbe un significato definito.

4. Integrità del codice sorgente dell'autore

La licenza può proibire che il codice sorgente venga distribuito in forma modificata solo se la licenza permette la distribuzione di "patch file" con il codice sorgente allo scopo di modificare il programma al momento della costruzione. Alcuni autori temevano che altri potessero distribuire codice sorgente con modifiche che sarebbero state percepite come opera dell'autore originale e quindi avrebbero potuto gettare ombra su di lui. Questa clausola dà loro un modo di imporre una separazione fra le modifiche e la loro opera, senza proibire le prime. C'è chi considera antiestetico che le modifiche debbano venir distribuite in un file "patch" separato dal codice sorgente, anche se distribuzioni Linux come Debian e Red Hat usano questa procedura per tutte le modifiche apportate ai programmi che distribuiscono. Esistono programmi per riversare automaticamente le patch nel sorgente principale, e questi programmi si possono eseguire automaticamente quando si scompatta un pacchetto di sorgente. Questa clausola, dunque, dovrebbe causare poca o nessuna difficoltà. Si noti anche che questa clausola dice che, nel caso di file patch, la modifica avviene quando si fa il build del programma. Questa scappatoia è impiegata nella Licenza Pubblica di Qt per prescrivere una diversa, anche se meno restrittiva, licenza per i file patch, in contraddizione con la sezione 3 della Open Source Definition. C'è una proposta per chiudere questa scappatoia nella definizione e mantenere nello stesso tempo Qt entro i confini dell'Open Source. La licenza deve permettere esplicitamente la distribuzione di software costruito da codice sorgente modi-

ficato. La licenza può richiedere che le opere derivate vadano sotto nome o numero di versione differenti da quelli del software originale. Questo significa che Netscape, per esempio, può insistere per poter essa la sola a chiamare una versione del programma Netscape Navigator (tm), mentre tutte le versioni gratuite del programma debbano chiamarsi Mozilla o in altro modo.

5. Nessuna discriminazione contro persone o gruppi

La licenza non deve discriminare alcuna persona o gruppo di persone. Una licenza fornita dai Rettori dell'Università della California a Berkeley proibiva l'uso di un programma di progettazione elettronica da parte delle forze di polizia del Sud Africa. Apprezzato come merita questo sentimento in tempi di apartheid, va detto che esso non ha più senso oggi. Alcune persone si trovano ancora con software acquistato sotto quella licenza, e le loro versioni derivate devono portare la stessa restrizione. Le licenze Open Source non devono contenere tale clausola, indipendentemente dalla nobiltà dell'intento.

6. Nessuna discriminazione di settori

La licenza non deve proibire ad alcuno l'uso del programma in uno specifico campo o per un determinato proposito. Per esempio, non può impedire che il programma venga usato a scopi commerciali o nella ricerca genetica. Il software dev'essere impiegabile allo stesso modo in una clinica che pratichi aborti e in un'organizzazione antiabortista. Queste discussioni politiche sono di pertinenza del Congresso degli Stati Uniti, non delle licenze del software. Alcuni trovano questa mancanza di discernimento gravemente offensiva!

7. Distribuzione della licenza

I diritti relativi al programma devono applicarsi a tutti coloro ai quali il programma sia ridistribuito, senza necessità di esecuzione di una licenza aggiuntiva da parte di questi. La licenza dev'essere automatica, senza la richiesta di alcuna firma. Purtroppo, negli Stati Uniti non ci sono dati validi precedenti giudiziari del potere della licenza senza firma quando questa venga passata da una seconda a una terza parte. Tuttavia, questo argomento considera la licenza come facente parte della legge sul contratto, mentre qualcuno obietta che dovrebbe essere considerata come legge di copyright, campo in cui si danno più precedenti per quel tipo di licenza. Un buon precedente ci sarà senz'altro nei prossimi anni, data la popolarità del questa licenza e il boom dell'Open Source.

8. La licenza non dev'essere specifica a un prodotto

I diritti relativi a un programma non devono dipendere dall'essere il programma parte di una particolare distribuzione software. Se il programma è estratto da quella distribuzione e usato o distribuito entro i termini della licenza del programma stesso, tutte le parti a cui il programma sia ridistribuito dovrebbero avere gli stessi diritti che vengono garantiti in unione alla distribuzione software originale. Questo significa che non si può impedire a un prodotto identificato come Open Source di essere gratuito solo se lo si usa con una marca particolare di distribuzione Linux, ecc. Deve rimanere gratuito se anche lo si separa dalla distribuzione software da cui proviene.

9. La licenza non deve contaminare altro software

La licenza non deve porre restrizioni ad altro software che sia distribuito insieme a quello licenziato. Per esempio, la licenza non deve pretendere che tutti gli altri programmi distribuiti sullo stesso media siano software Open Source. Una versione di GhostScript (programma di rendering PostScript) richiede che i media sui quali viene distribuito contengano solo programmi software gratuiti. Questo non è consentito dalla licenza Open Source. Per fortuna, l'autore di GhostScript

distribuisce un'altra versione del programma (un po' più vecchia) sotto una licenza Open Source genuina. Si noti che c'è differenza fra derivazione e aggregazione. Derivazione è quando un programma incorpora di fatto in sé parti di un altro programma. Aggregazione è quando due programmi vengono inclusi sullo stesso CD-ROM. Questa sezione della Open Source Definition riguarda l'aggregazione, non la derivazione. La sezione 4 riguarda la derivazione.

10. Licenze esemplari

Le licenze GNU GPL, BSD, X Consortium e Artistica sono esempi di licenze da considerarsi conformi alla Open Source Definition. Altrettanto dicasi della MPL.

Questo sarebbe una fonte di guai nel giorno in cui una di queste licenze si modificasse e non fosse più Open Source: dovremmo pubblicare immediatamente una revisione della Open Source Definition. Ciò è pertinente per la verità al testo esplicativo, non alla Open Source Definition in sé."

Analisi delle licenze e loro conformità all'Open Source

"Per comprendere la Open Source Definition dobbiamo esaminare alcune pratiche comuni nelle licenze in quanto si riferiscono all'Open Source."

Public Domain

"La diffusa convinzione che molto del free software sia di dominio pubblico è errata. Ciò avviene perché l'idea di free software o Open Source confonde molti, che quindi definiscono erroneamente questi programmi come di pubblico dominio perché è il concetto più prossimo a quanto è loro familiare. I programmi, tuttavia, sono molto chiaramente protetti da diritti e sottoposti a licenza: solo, si tratta di una licenza che dà al pubblico più diritti di quelli a cui sia abituato.

Un programma di pubblico dominio è un programma sul quale l'autore abbia rinunciato a tutti i suoi diritti di copyright. Non si può esattamente dire che sia dotato di una licenza; è proprietà personale, per usarlo come meglio si crede. Dal momento che si può trattarlo come personale proprietà, con un programma di pubblico dominio si può fare quello che si vuole. Si può perfino ri-licenziare un programma di pubblico dominio, rimuovendo quella versione dal pubblico dominio, o togliendo il nome del suo autore e trattarlo come opera propria.

Se si sta spendendo molto lavoro su un programma di pubblico dominio, si consideri la possibilità di applicarvi il proprio copyright e di rimetterlo sotto licenza. Per esempio, se non si desidera che una terza parte operi delle modifiche che possa poi mantenere riservate, si applichi la GPL o una licenza simile alla propria versione del programma. La versione da cui si è partiti rimarrà nel pubblico dominio, ma la propria versione sarà sotto una licenza che dovrà essere osservata da chi la usa o ne derivi altre.

Un programma di pubblico dominio si rende privato facilmente, dichiarando un copyright e applicandovi la propria licenza, oppure semplicemente dichiarando "Tutti i diritti riservati"."

Le licenze Free Software in generale

"Se si ha una raccolta di free software come un disco Linux, si potrebbe credere che il programma su quel disco sia proprio. Ma questo non è del tutto vero. I programmi coperti da copyright sono proprietà di chi detiene il copyright, anche quando arrivano con una licenza Open Source come la GPL. La licenza del programma garantisce alcuni diritti, e altri si hanno sotto la definizione di uso corretto nella legge sul copyright.

È importante notare che un autore non deve necessariamente limitarsi a porre una sola licenza su un programma che pubblica. Un programma può essere posto sotto GPL, e una versione può anche essere venduta con una licenza commerciale, non-Open Source. Proprio di questa strategia si valgono molti che desiderano creare un programma Open Source e allo stesso tempo

guadagnarci qualche cosa. Chi non vuole una licenza Open Source può pagare per il privilegio, fornendo all'autore una fonte d'entrate.

Tutte le licenze che esamineremo hanno una caratteristica comune: declinano qualunque garanzia. Lo scopo è quello di proteggere il proprietario del software da qualunque responsabilità connessa al programma. Appare una richiesta ragionevole, dato che il programma viene ceduto a costo zero: l'autore non riceve dal programma una fonte d'entrata sufficiente per sostenere un'assicurazione sulle responsabilità ed eventuali spese legali.

Se gli autori di free software perdessero il diritto di declinare tutte le garanzie e si trovassero a essere citati in tribunale in base alle prestazioni dei programmi che hanno scritto, smetterebbero di fornire software gratuito al mondo. È nel nostro interesse di utenti aiutare gli autori a proteggere questo diritto.”

La GNU General Public License

“Si veda l'appendice B per il testo completo della GPL. La GPL è un manifesto politico tanto quanto è una licenza software, e la maggior parte del testo è inteso a spiegare la motivazione teorica dietro la licenza. Questo dibattito politico ha allontanato alcuni e fornito alcune delle ragioni per cui sono state scritte altre licenze per il free software. Tuttavia, la GPL è stata stilata con l'assistenza di giuristi ed è per questo assai meglio scritta della maggior parte delle licenze di quella famiglia. Io consiglio caldamente di usare la GPL, o la sua variante per librerie LGPL, ogni volta che sia possibile. Se si sceglie un'altra licenza, o se ne stila una nuova, ci devono essere delle buone ragioni per farlo. Chi formula la propria licenza dovrebbe sapere bene che non è un passo da fare con leggerezza. Le complicazioni inaspettate di una licenza affrettata possono affliggere gli utenti di un software per molti anni a venire.

Il testo della GPL non è a sua volta sotto GPL. La sua licenza è semplice: Chiunque può copiare e distribuire copie esatte di questo documento di licenza, ma non ne sono ammesse modifiche. Un punto importante, qui, è che il testo delle licenze di software Open Source di solito non è Open Source esso stesso. Ovviamente, una licenza non potrebbe offrire protezione di alcun tipo se a chiunque fosse consentito apportarvi delle modifiche.

Le clausole della GPL soddisfano la Open Source Definition. La GPL non richiede alcuna delle clausole consentite dal Paragrafo 4 della Open Source Definition. Integrità del codice sorgente dell'autore.

La GPL non permette di mantenere private le modifiche apportate. Le modifiche devono essere distribuite sotto la GPL. In questo modo, l'autore di un programma sotto GPL ha maggiori probabilità di ricevere modifiche da altri, comprese società commerciali che modificano il suo software per i propri scopi.

La GPL non ammette l'incorporazione di un programma sotto GPL in un programma proprietario. La definizione di GPL di programma proprietario lo indica come ogni programma con una licenza che non dia tanti diritti quanti la GPL.

Esistono alcune scappatoie nella GPL che permettono di usarla in un programma non interamente Open Source. Le librerie software che vengono normalmente distribuite con il compilatore o con il sistema operativo che si usa possono essere collegate a software GPL: ne risulta un programma parzialmente libero. Chi detiene il copyright (di norma l'autore del programma) è la persona che mette il programma sotto GPL e ha il diritto di violare la propria licenza. Questa scappatoia è stata usata dagli autori di KDE per distribuire il loro programma Qt prima che Troll Tech ponesse su Qt una licenza Open Source. Tuttavia, questo diritto non si estende ad alcuna terza parte che ridistribuisca il programma: esse devono seguire tutti i termini della licenza, anche quelli che vengono violati dal detentore del copyright, il che rende problematico ridistribuire un programma che contenga Qt sotto GPL.

Gli sviluppatori KDE sembrano inclini a rimediare a questo problema applicando al loro software la LGPL piuttosto che la GPL. La retorica politica presente nella GPL non è gradita a tutti. Non manca chi ha scelto, per il suo software, licenze non altrettanto adatte per semplice avversione alle idee di Richard Stallmann, pur di non aver voluto vederle ripetute nei propri pacchetti software.”

La GNU Library Public License

“La LGPL è un derivato della GPL escogitato per le librerie software. A differenza della GPL, un programma sotto LGPL può venire incorporato entro un programma proprietario. La libreria di linguaggio C fornita con i sistemi Linux e un esempio di software sotto LGPL: essa può essere usata per costruire programmi proprietari, diversamente Linux risulterebbe utile solamente agli autori di free software. Una copia di un programma sotto LGPL può essere convertita in qualunque momento in una sotto GPL. Una volta che ciò succede, quella copia non è più riconvertibile in un programma sotto LGPL, e altrettanto dicasi di qualunque suo derivato. Le rimanenti clausole della LGPL sono simili a quelle della GPL: di fatto essa include la GPL facendovi riferimento.”

Le licenze X, BSD e Apache

“La licenza X e le sue affini BSD e Apache sono molto diverse dalla GPL e dalla LGPL. Queste licenze consentono di fare quasi tutto ciò che si vuole con il software ‘licenziato’ sotto di esse, e questo perché il software originariamente coperto dalle licenze X e BSD era sovvenzionato con sussidi del Governo degli Stati Uniti. Dal momento che i cittadini statunitensi avevano già pagato il software con i soldi delle tasse, fu loro garantito il diritto di fare del software tutto ciò che volessero.

La concessione più importante, assente dalla GPL, e che si può mantenere private le modifiche licenziate sotto licenza X. In altre parole, si può ottenere il codice sorgente di un programma sotto X, modificarlo e poi vendere versioni binarie del programma senza distribuire il codice sorgente delle modifiche e senza applicarvi la licenza X. Tutto ciò rimane comunque Open Source, poiché la Open Source Definition non richiede che le modifiche debbano sempre ricadere sotto la licenza originale.

Molti altri sviluppatori hanno adottato la licenza X e le sue varianti, compresi i progetti BSD (Berkeley System Distribution) e Apache Web server. Un dettaglio molesto della licenza BSD è costituito da una clausola che prescrive che ogni volta si faccia cenno a una caratteristica di un programma sotto BSD in una sua pubblicità, si menzioni (generalmente in una nota a pie di pagina) il fatto che il software è stato sviluppato all’Università della California.

Ora, tener traccia di quale software abbia quella licenza in una cosa immensa come una distribuzione Linux, e ricordare quindi di menzionare l’Università della California ogni volta che uno di questi programmi venga citato in una pubblicità, è un vero mal di testa per i gestori commerciali del progetto. Nel momento in cui scrivo, la distribuzione Debian GNU/Linux contiene oltre 2500 pacchetti software, e se anche solo una piccola parte di essi fosse sotto BSD, la pubblicità per un sistema Linux come Debian dovrebbe contenere molte pagine solo di note! Tuttavia, la licenza dell’X Consortium non ha quella clausola della pubblicità. Se si pensa di usare una licenza tipo BSD si usi invece una licenza X.”

La Licenza Artistica

“Sebbene questa licenza sia stata in origine sviluppata per il Perl, è stata dopo allora adoperata per altro software. A mio parere si tratta di una licenza formulata con grande sciattezza, in quanto impone dei requisiti e fornisce poi delle scappatoie che rendono facile aggirarli. Forse e questa la ragione per cui quasi tutto il software sotto Licenza Artistica, ha oggi una seconda licenza, offrendo la scelta fra la Licenza Artistica e la GPL.

La Sezione 5 della Licenza Artistica vieta la vendita del software, ma permette che sia venduta una distribuzione di software aggregato di più di un programma. In questo modo, se raggruppate un programma sotto Licenza Artistica con un `helloworld.c` di cinque righe di codice, potete vendere il bundle. Questa caratteristica della Licenza Artistica è stata la sola causa della scappatoia dell’“aggregato” nel primo paragrafo della Open Source Definition. Dal momento che l’uso della Licenza Artistica è in netto declino, stiamo pensando di togliere quella scappatoia. Ciò renderebbe la Licenza Artistica una licenza non-Open Source. Non è questo un passo che fare-

mo leggermente, e ci vorrà probabilmente più di un anno di riflessione e di dibattito prima che questo accada.

La Licenza Artistica richiede che le modifiche siano rese gratuite, ma fornisce poi una scappatoia (nella Sezione 7) che permette di mantenerle private e perfino di porre sotto dominio pubblico parti del programma sotto Licenza Artistica!”

La Netscape Public License e la Mozilla Public License

“La NPL è stata sviluppata da Netscape quando rese Open Source il suo prodotto Netscape Navigator. Per la precisione, la versione Open Source si chiama Mozilla; Netscape si riserva il marchio Navigator per il suo prodotto. Eric Raymond ed io agimmo come consulenti a titolo gratuito durante lo sviluppo di questa licenza. Io cercai, senza successo, di persuadere Netscape a usare la GPL, e quando essa declinò, contribuì a comporre una licenza che si conformasse alla Open Source Definition.

Una caratteristica importante della NPL è che contiene privilegi speciali che si applicano a Netscape e a nessun altro. Essa dà a Netscape il privilegio di rilicenziare le modifiche fatte al suo software. Netscape può mantenere private quelle modifiche, migliorarle, e rifiutarsi di restituire il risultato. Questa clausola si è resa necessaria perché, quando Netscape decise per l’Open Source, aveva contratti con altre aziende che la impegnavano a fornir loro Navigator sotto una licenza non Open Source. Netscape ha creato la MPL o Mozilla Public License per rimediare a questa situazione. La MPL è molto simile alla NPL, ma non contiene la clausola che permette a Netscape di rimettere le modifiche sotto licenza.

La NPL e la MPL consentono di mantenere private le modifiche apportate.

Molte aziende hanno adottato per i loro programmi una variante della MPL. Non è una buona cosa, perché la NPL è stata progettata per la particolare situazione contrattuale in cui Netscape si trovava nel momento in cui la licenza veniva scritta, e non è detto che sia altrettanto adatta a usi diversi. Dovrebbe restare la licenza di Netscape e di Mozilla, e altri dovrebbero usare le licenze GPL o X.”

Scegliere una licenza

“Non conviene formulare una licenza nuova se è possibile usarne una di quelle qui elencate. La propagazione di molte licenze diverse e incompatibili opera a detrimento del software Open Source, perché frammenti di un programma non possono essere usati in un altro programma sotto licenza incompatibile.

Ci si tenga alla larga dalla Licenza Artistica, a meno che non si intenda studiarla a fondo ed eliminarne le scappatoie. Fatto ciò, è tempo di prendere delle decisioni.

1. Si vuole che il pubblico possa mantenere private le modifiche, o no? Se si vuole che chi ha apportato modifiche al proprio software ne rimandi il codice sorgente, si applichi una licenza che lo prescriva. La GPL e la LGPL sono delle buone scelte. Se non dispiace che il pubblico mantenga private le modifiche, si usino la licenza X o la licenza Apache.
2. Si vuole consentire a qualcuno di far confluire il proprio programma nel suo software proprietario? Se sì, si usi la LGPL, che lo permette esplicitamente senza consentire al pubblico di rendere privato il codice, oppure si usi la licenza X o Apache, che permettono che le modifiche siano mantenute private.
3. Si desidera che chi lo voglia possa comprare sotto licenza commerciale versioni non Open Source del proprio programma? Se sì, si doti il software di doppia licenza. Io consiglio la GPL come licenza Open Source; si può trovare una licenza commerciale adatta all’uso in libri come "Copyright Your Software" edito da Nolo Press.
4. Si vuole che chiunque usi il proprio software debba pagare per il privilegio? Se le cose stanno così, forse l’Open Source non è adatta. Se basta che solo alcune persone paghino,

Tipologie	A	B	C	D
GPL				
LGPL	x			
BSD	x	x		
NPL	x	x		x
MPL	x	x		
Dominio Pubblico	x	x	x	

A Può essere miscelato con software commerciale

B Le modifiche possono essere mantenute private e non restituite all'autore originale

C Può essere ri-licenziato da chiunque

D Contiene privilegi speciali sulle modifiche per chi detiene il copyright originale

Tabella 4.1: Caratteristiche delle licenze software.

si può mantenere Open Source il programma. La maggior parte degli autori Open Source considerano i loro programmi come contributi al bene pubblico, e non badano al fatto di essere pagati oppure no.

Per un quadro chiaro consultare la tabella 4.1 a pagina 50.

Il Futuro

“Al momento in cui questo saggio andava in stampa, IBM entrava nel mondo Open Source e la comunità dei venture capital lo sta scoprendo. Intel e Netscape hanno investito in Red Hat, un distributore Linux. VA Research, integratore di server Linux e hardware per workstation, ha annunciato l'ingresso di un investitore esterno. Sendmail Inc., creata per commercializzare l'onnipresente programma di posta elettronica Sendmail, ha annunciato la disponibilità di fondi per sei milioni di dollari.

L'applicazione di posta protetta Postfix di IBM ha una licenza Open Source, e un altro prodotto IBM, il compilatore Java Jikes, ha una licenza che, nell'istante in cui scrivo, mira, per il momento con parziale successo, a soddisfare le specifiche dell'Open Source Definition. Parrebbe che IBM intenda modificare la licenza di Jikes perché sia per intero Open Source, e che a questo scopo stia raccogliendo pareri nella comunità.

Due promemoria interni della Microsoft, noti sono il nome di Halloween Document, sono trapelati al pubblico online. Questi promemoria mostrano in modo inequivocabile come Microsoft si senta minacciata da Open Source e da Linux, e che Microsoft lancerà un'offensiva contro di loro per proteggere i suoi mercati. E' chiaro che dobbiamo prepararci a vivere tempi interessanti. Credo che vedremo Microsoft usare due principali strategie: interfacce sotto copyright e brevetti. Microsoft estenderà i protocolli di rete, che contengono caratteristiche proprietarie Microsoft in quelli che non verranno resi disponibili al free software. Essa, con altre aziende, farà ricerca aggressivamente in nuove direzioni dell'informatica e brevetterà tutto quanto potrà prima che si possano sfruttare quelle tecniche nel free software; quindi ci chiuderà fuori con le concessioni sui diritti di brevetto. Sono autore di un saggio per la webzine Linux World su come si possano battere i nemici dell'Open Source sul fronte dei brevetti.

La buona notizia è che Microsoft si è spaventata! Nel secondo degli Halloween Document, un membro dello staff Microsoft racconta della sua sensazione d'euforia nel vedere come poteva modificare facilmente parti del sistema Linux perché facesse esattamente quello che voleva, e com'era più facile per un impiegato Microsoft fare questo su Linux di quanto non lo fosse modificare NT. I tentativi di nuocerici provenienti dall'interno sono i più pericolosi. Credo che vedremo

altri sforzi per diluire la definizione di Open Source fino a includervi prodotti parzialmente gratuiti, come abbiamo visto avvenire con la libreria Qt in KDE prima che Troll Tech vedesse la luce e rilasciasse una licenza Open Source. Microsoft e altri potrebbero danneggiarci rilasciando un sacco di software gratuito quel tanto da attrarre utenti, ma senza avere le piene libertà dell'Open Source. Non è impensabile che essi possano stroncare lo sviluppo di certe categorie di software Open Source rilasciando soluzioni "abbastanza valide", "abbastanza quasi-gratis". Tuttavia, la forte reazione che si è avuta contro il progetto KDE prima che la libreria Qt divenisse completamente Open Source, non è di buon augurio per imprese analoghe di Microsoft e compagnia.

Finora abbiamo scampato i cavalli di Troia. Supponiamo che qualcuno che ci vuol male fornisca del software che contiene un cavallo di Troia un espediente per sconfiggere la protezione in un sistema Linux. Supponiamo, poi, che la medesima persona resti in attesa che il software con il cavallo di Troia sia largamente distribuito e quindi ne pubblicizzi la vulnerabilità agli attacchi alla sicurezza. Il pubblico si sarà per allora accorto che il nostro sistema Open Source può lasciarci più vulnerabili a questa sorta di attacchi che non il sistema chiuso di Microsoft; questo potrebbe ridurre la fiducia generale nel software Open Source. Potremmo obiettare che Microsoft ha la sua parte di bug di sicurezza anche se non lascia possibilità di inserirli a persone esterne e che il modello a codice sorgente aperto dell'Open Source rende più facile scoprire questi bug. Qualunque bug del genere che comparisse in Linux sarebbe riparato il giorno dopo essere stato scoperto, mentre un omologo in Windows o non sarebbe mai scoperto o dovrebbe aspettare il rimedio per anni. Ma dobbiamo rinforzare ancora la nostra difesa contro i cavalli di Troia. Identificare con sicurezza chi contribuisce alla creazione di software e delle modifiche è la difesa migliore di cui disponiamo, dal momento che ci permette di valerci del diritto penale contro chi escogita cavalli di Troia. Quando ero dirigente della distribuzione GNU/Linux di Debian, istituimmo un sistema che consentiva di identificare in modo affidabile tutti i manutentori del software e permetteva loro di partecipare a loro volta a una rete a crittografia a chiave pubblica che ci avrebbe consentito di verificare da chi proveniva il nostro software. Questo tipo di sistema si deve espandere fino a comprendere tutti gli sviluppatori Open Source.

Enormi sono i miglioramenti da intraprendere prima che Linux sia davvero alla portata dell'utente medio. L'interfaccia grafica per gli utenti è chiaramente qualcosa che manca, e a questo sono rivolti i progetti KDE e GNOME. La prossima frontiera è l'amministrazione di sistema Linuxconf vi sta parzialmente provvedendo, ma si trova ben lungi dall'essere uno strumento completo d'amministrazione di sistema per l'utente sprovvisto. Se il sistema COAS di Caldera avrà successo, potrebbe diventare la base per una soluzione completa al problema dell'amministrazione di sistema. Tuttavia, Caldera ha avuto dei problemi nel mantenere un'allocazione di risorse sufficienti a COAS per terminarne lo sviluppo, e altri sviluppatori hanno abbandonato la partita perché non notavano progressi. La pleora di distribuzioni Linux appare oggi in pieno rivolgimento, con Red Hat percepita come vincitrice e Caldera come seconda. Red Hat ha mostrato finora un solido impegno verso il concetto di Open Source, ma un nuovo presidente e voci di un'offerta pubblica iniziale (Initial Public Offering, IPO) potrebbero significare un indebolimento di quest'impegno, specialmente se concorrenti come Caldera, molto più tiepidi verso l'Open Source, riusciranno a inserirsi nel mercato di Red Hat. Se l'impegno delle distribuzioni Linux commerciali verso l'Open Source diventerà problematico, questo genererà probabilmente uno sforzo per rimpiazzarle con tentativi Open Source simili al GNU/Linux di Debian ma più diretti al mercato commerciale di quanto non sia stata Debian. Malgrado queste sfide, io predico la vittoria dell'Open Source. Linux è divenuto strumento di test per gli studenti d'informatica, che, una volta laureati, porteranno con sé quegli strumenti nei loro posti di lavoro. Molti laboratori di ricerca hanno adottato il modello Open Source in quanto la condivisione delle informazioni è essenziale al metodo scientifico, e l'Open Source consente al software di essere condiviso facilmente. Il mondo business sta adottando il modello Open Source perché consente a gruppi di aziende di collaborare nella risoluzione di un problema senza la minaccia di una causa anti-trust, e per l'impulso di cui gode quando i contributi pubblici di programmazione rendono gratuite le migliori al software. Alcune grandi società hanno adottato l'Open Source come strategia per combattere e scongiurare l'avvento di un'altra Microsoft pronta a dominare il settore informatico. Ma l'indicazione più affidabile sul futuro dell'Open Source viene dal suo passato: in pochi anni,

dal niente siamo arrivati ad avere un robusto corpus di software che è in grado di risolvere tanti problemi diversi e che si avvia a raggiungere il milione di utenti. Non c'è ragione di rallentare la corsa proprio adesso.”

Parte III
GNU/Linux

Capitolo 5

Il fenomeno GNU/Linux

GNU/Linux è una libera e distribuibile versione di UNIX sviluppata da Linus Torvalds presso l'Università di Helsinki in Finlandia. GNU/Linux è un *melting pot* di conoscenze ed esperienze che hanno trovato in quest'ultimo una piazza virtuale dove crescere rapidamente. All'interno del sistema operativo non viene utilizzato in nessun modo codice licenziato da enti commerciali e buona parte del software che ruota attorno segue questa "corrente" di pensiero abbracciata dal Progetto GNU della Free Software Foundation¹. Linux infatti rappresenta unicamente il kernel mentre GNU/Linux sarebbe la giusta formulazione per indicare il kernel di Linux e tutto il software di gestione offerto dal progetto GNU che rende un sistema operativo utilizzabile. Per maggiori informazioni in merito consiglio il documento disponibile sul sito GNU - <http://www.gnu.org/gnu/linux-and-gnu.html>².

5.1 La nascita

Il kernel di Linux è stato originariamente sviluppato per hobby da Linus Torvalds. Il progetto iniziale era ispirato a Minix, un piccola versione di UNIX sviluppata da Andy Tanenbaum, e proprio nel gruppo di discussione comp.os.minix venne rilasciato il seguente messaggio da Torvalds :

"After that it was plain sailing: hairy coding still, but I had some devices, and debugging was easier. I started using C at this stage, and it certainly speeds up development. This is also when I start to get serious about my megalomaniac ideas to make 'a better Minix than Minix'. I was hoping I'd be able to recompile gcc under Linux some day... Two months for basic setup, but then only slightly longer until I had a disk-driver (seriously buggy, but it happened to work on my machine) and a small filesystem. That was about when I made 0.01 available [around late August of 1991]: it wasn't pretty, it had no floppy driver, and it couldn't do much anything. I don't think anybody ever compiled that version. But by then I was hooked, and didn't want to stop until I could chuck out Minix."

Nessun annuncio venne fatto per la versione 0.01. Le sorgenti di quest'ultimo non erano eseguibili e contenevano unicamente le rudimentali sorgenti del kernel da compilare su un server Minix. Nell'ottobre 1991 venne annunciata la versione 0.02 di Linux, la prima versione ad essere definita "ufficiale". Quest'ultima rimaneva pur sempre scarna ma includeva la shell dei comandi bash (GNU Bourne Again Shell) e gcc, il compilatore GNU. Era il primo ed importante passo ed un'intera comunità di sviluppatori abbracciò il progetto.

Linus scrisse su comp.os.minix:

¹Cambridge, Massachusetts.

²Si ringrazia Bradley M. Kuhn della Free Software Foundation per aver consigliato la presente nota nel manuale.



Figura 5.1: Linus Torvalds fotografato dopo un talk.

“Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you sure without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you. As i mentioned a month ago, I’m working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it’s even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02... but I’ve successfully run bash, gcc, gnu-make, gnu-sed, compress, etc. under it.”

Dopo la versione 0.03 ci fu un rapido susseguirsi di versioni grazie alla natura del progetto che concentrava l’opera di una crescente comunità telematica. La versione 1.0 venne rilasciata nel dicembre del 1993 dopo essere considerata stabile e bug-free, ovvero senza problemi. Il kernel di Linux rilasciato era compatibile con molti standard UNIX come IEEE POSIX.1, System V e BSD e poteva “funzionare” su processori di classe i386, disponibili in gran parte dei personal computer esistenti.

Le frasi “Linux is not Unix” o “GNU is not Unix” hanno un senso; il marchio UNIX è licenziato da X/Open che su richiesta e a pagamento valuta un sistema operativo e definisce o meno l’utilizzo del nome “UNIX”. Linux è il paradosso perché malgrado la mancanza del “titolo” ha saputo imporsi ugualmente e diventare la linea guida nello sviluppo software per sistemi UNIX, scavalcando per propria natura ogni limite relativo a licenze con particolari note restrittive.

Dopo circa 8 anni dal messaggio inizialmente riportato lo spirito di Torvalds non è affatto cambiato. Il 10 Marzo 2000 scriveva:

“I just made a 2.3.51 release, and the next kernel will be the first of the pre-2.4.x kernels. That does NOT mean that I’ll apply a lot of last-minute patches: it only means that I’ll let 2.3.51 be out there over the weekend to hear about any embarrassing problems so that we can start the pre-2.4 series without the truly stupid stuff. There’s some NFSv3 and other stuff pending, but those who have pending stuff should all

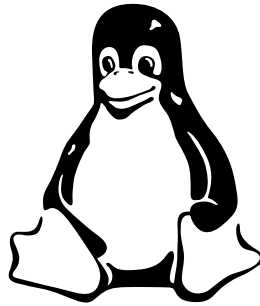


Figura 5.2: Tux, un pinguino come amico

know who they are, and for the rest it's just time to say nice try, see you in 2.5.x. The pre-2.4.x series will probably go on for a while, but these are the "bug fixes only" trees. These are also the "I hope a lot of people test them" trees, because without testing we'll never get to the eventual goal, which is a good and stable 2.4.x in the reasonably near future."

"I hope a lot of people test them", ovvero la versione "spero che molta gente lo testi" per riportare eventuali mancanze o errori da correggere prima della versione finale. Ironia e modestia, le chiavi di successo. In un messaggio ironico spedito il primo giorno di Aprile 2000, Torvalds scriveva:

"Dear Linux Users,

I'm pleased to announce jointly with Microsoft(tm)(r) Corporation release of Linux 2000 Professional Enterprise. As you probably already know I'm busy with my family and I already have full-time job with Transmeta. Thus, it has been necessary for me to look for some responsible partner who would help me develop Linux. After extensive search, I have decided upon Microsoft Corporation which has been known on market for long time from their high quality software. Thus the upcoming Linux 2.4.0 will become Linux 2000(tm)(r). Pricing will be determined at later time. However, I would like to take opportunity now to remind people who have unlicensed version of Linux to delete it from hard disk and then wait until official release of Linux 2000(tm)(r) will become available. Effective April 1st 2000, midnight, all older versions of Linux are illegal under Digital Millennium Copyright Act."

Un bel "pesce d'aprile"! :)

Un pinguino come amico

Linux viene spesso rappresentato da un pinguino che è diventata la mascotte ufficiale del sistema operativo. L'immagine ufficiale (figura 5.2) è stata creata da Larry Ewing³.

Il nome di questo amichevole pinguino, Tux, è stato suggerito da James Hughes in un messaggio apparso sulla mailing-list ufficiale del kernel di linux in data 10 giugno 1996. Tux può essere interpretato come l'abbreviazione di Torvalds UniX.

5.2 Distribuzioni

GNU/Linux si può scaricare gratuitamente dalla rete oppure lo si può ottenere tramite le "distribuzioni" esistenti, nate per diffondere il sistema operativo su supporti fisici come floppy o cd-rom e colmare la mancanza di connessioni in rete stabili e veloci.

³<http://www.isc.tamu.edu/~lewing/linux/>

La scelta giusta

Le sostanziali differenze tra le distribuzioni nascono con l'aggiunta di ulteriore software per l'installazione e per la gestione, supporto tecnico nonché materiale cartaceo come manuali e documentazione di vario genere. E' difficile valutare questo valore aggiunto e stabilire quale distribuzione possa ritenersi migliore di un'altra visto che ognuna ha una serie di caratteristiche proprie che la distinguono sotto l'aspetto della sicurezza, della gestione o della facilità d'uso.

Si potrebbe produrre una immensa serie di motivazioni e considerazioni ma il consiglio che si vuole dare è quello di leggere le note sottostanti e fare una scelta per iniziare:

- Se dovete gestire un server Internet la scelta migliore potrà essere la distribuzione Debian, molto curata sotto gli aspetti della sicurezza e facilmente aggiornabile in rete. Una ulteriore caratteristica è la totale assenza di software commerciale e questa condizione la pone in stretta correlazione con la filosofia di Richard Stallman e il progetto GNU.
- Se dovete utilizzare GNU/Linux come workstation di lavoro una buona scelta può ricadere sulla distribuzione Red Hat, facile da installare per una persona alle prime armi e semplice da gestire con il potente sistema di gestione dei pacchetti software denominato rpm. E' la distribuzione maggiormente utilizzata nel mondo.
- Distribuzione derivata da Red Hat Linux è Linux Mandrake, versione modificata e particolarmente curata per utilizzare al meglio KDE come desktop manager predefinito.
- Se desiderate installare e provare ogni sorta di software senza vincoli particolari di licenze potete optare per la distribuzione tedesca SuSE, per certi aspetti spartana ma decisamente ottima per ambiti aziendali.
- Se infine rientrate nella categoria "smanettoni" (parola che può suscitare sorrisi ma che è comune e largamente utilizzata) potrete trovare la vostra panacea nella distribuzione Slackware, soluzione ideale per chi desidera conoscere il proprio sistema nei minimi particolari.

I nomi indicati sono sostanzialmente quelli delle distribuzioni maggiormente utilizzate ed è facile trovare, oltre alle distribuzioni per processori della famiglia i386 (Intel e compatibili), versioni per processori di diverso genere come Alpha, Sparc, PPC.

Dove trovare le distribuzioni

La diffusione di GNU/Linux lo ha reso ambito e acclamato tanto da rendere molto semplice poter trovare una distribuzione completa del sistema operativo. Potete informarvi presso una qualsiasi attività commerciale che distribuisce materiale informatico o passare direttamente in edicola per acquistare una delle tante riviste che allegano cd-rom con GNU/Linux. Numerosi anche i rivenditori online, tra i quali indico i seguenti:

Rivenditori internazionali

- Infomagic - <http://www.infomagic.com>
- Cheapbytes - <http://www.cheapbytes.com>
- Linux Systems Labs - <http://www.lsl.com>
- Datom - <http://www.datom.de>

Rivenditori italiani

- Libreria Athena (Modena) - <http://www.mondolinux.com>

- Direct.it - <http://www.direct.it>

Sempre più frequentemente le distribuzioni diffuse attivano e-commerce dai propri siti Internet e quindi è consigliabile visitare direttamente quest'ultimi per le proprie necessità:

- Caldera - <http://www.caldera.com>
- Corel - <http://linux.corel.com>
- Debian - <http://www.debian.org>
- Mandrake - <http://www.linux-mandrake.com>
- LinuxPPC - <http://www.linuxppc.org>
- Red Hat - <http://www.redhat.com>
- Slackware - <http://www.cdrom.com>
- Stampede - <http://www.stampede.org>
- S.u.S.E. - <http://www.suse.com>
- Turbo Linux - <http://www.pht.com>
- VA Research - <http://www.valinux.com>

5.3 Documentazione

Una caratteristica importante nel mondo GNU/Linux è la memoria storica. Esiste un'incredibile mole di documentazione disponibile in rete e pronta a risolvere ogni singolo problema, una quantità talmente elevata di informazioni che spesso un nuovo utente si trova disorientato. Analizzeremo di seguito le varie tipologie informative presenti e i metodi più rapidi per la consultazione.

Pagine Manuali / Guide in linea

Rappresentano una forma molto comoda per consultare informazioni relative a programmi installati nel sistema. Hanno una organizzazione sostanzialmente comune che presenta un'area descrittiva, dove viene spiegato lo scopo del programma, una sezione relativa alla sintassi d'uso del comando con le varie opzioni, consigli di vario genere e comandi eventuali logicamente correlati.

Per poter consultare queste guide è necessario utilizzare il comando `man` nel seguente modo:

```
[darshan@dada darshan]$ man nomecomando
```

Per uscire dalla visualizzazione della guida in linea è necessario digitare il tasto "q". Maggiori dettagli sul comando a pagina .

How-To

Sono manuali orientati a specifici argomenti; esistono How-to per una moltitudine di argomenti diretti a risolvere particolari e ben determinate problematiche. Esiste da anni un progetto che cura la gestione razionale di questi manuali che si chiama LDP⁴, <http://metalab.unc.edu/mdw/>.

⁴Linux Documentation Project.

Il progetto è nello spirito tipico che contraddistingue GNU/Linux, ovvero opera di volontari che impiegano parte del proprio tempo per lasciare le loro esperienze a chi ne necessiterà successivamente. Siccome la gran parte della documentazione informatica nasce e viene sviluppata in lingua inglese si è formato un gruppo *parallelo* chiamato ILDP⁵, con il particolare scopo di tradurre la documentazione esistente in lingua italiana.

Mailing List

Esistono numerose mailing list, strumenti che permettono ad una molteplicità di persone di *comunicare* tramite e-mail e condividere queste informazioni. Esistono mailing list per progetti software ben determinati e mailing list create da gruppi di utenti GNU/Linux di carattere generico. Se desiderate conoscere le argomentazioni di ogni lista e trovare le modalità di iscrizione si consiglia di visitare i seguenti siti Internet:

- ErLug (Emilia Romagna Linux User Group) - <http://erlug.linux.it>
- ILS (Italian Linux Society) - <http://www.linux.it>
- Pluto - <http://www.pluto.linux.it>

I siti indicati sono fonti informative di vario genere da considerare come punti di partenza nella rete Internet per ampliare le proprie conoscenze.

Newsgroup

Esiste un serie di gruppi di discussione in lingua italiana e fanno parte della gerarchia `it.comp.os.linux` divisi in macrocategorie a seconda degli scopi finali degli utenti e visualizzabili con un comune programma client per i newsgroup.

Ulteriore Documentazione Italiana

Nello spirito che contraddistingue GNU/Linux, esistono numerose forme documentative curate da volontari e disponibili gratuitamente in rete.

Appunti Linux

Appunti Linux di Daniele Giacomini rappresenta un corposo insieme di informazioni utili per gestire le attività del vostro personale server GNU/Linux. La documentazione copre argomenti relativi all'amministrazione di sistema, la programmazione e i servizi Internet.

<http://appuntilinux.prosa.it/>

LDR

Linux Domanda e Risposta di Gaetano Paolone, rappresenta una interessante raccolta di F.A.Q. particolarmente indirizzate all'utente novizio e di medio livello. Può essere molto utile per la risoluzione di alcune problematiche e per imparare nuove soluzioni per l'ottimizzazione del sistema.

<http://web.tiscalinet.it/linuxfaq/>

⁵Italian Linux Documentation Program - <http://www.pluto.linux.it/ILDP/>

5.4 Valore aggiunto

Le persone che spesso utilizzano strumenti informatici spesso si trovano a *combattere* con soluzioni inadeguate alle proprie esigenze anche se ampiamente diffuse commercialmente. E' necessario capire che non sempre il prodotto hardware o software diffuso e famoso coincide con un prodotto di qualità. Spesso sono politiche di marketing e comunicazione a *convincere* le persone che un prodotto è valido anche se le sue caratteristiche sono inferiori a quanto promesso.

“Se dovete mangiare entrate in un supermercato e scegliete tra i prodotti presenti. Ogni prodotto, essendo un bene alimentare, presenta una etichetta dove è esposta una varietà di dati importanti: il produttore, gli ingredienti utilizzati, le analisi chimiche e perché no... comunicazione tramite la grafica e l'apparenza del prodotto. Il produttore stesso vuole tutelarvi mostrando la sua qualità attraverso la trasparenza delle informazioni.”

L'ambito informatico può avere un concetto molto vicino all'esempio visto, soprattutto in ambito sociale dove i mezzi considerati non sono più una alternativa ma un passaggio obbligato. Conoscere le caratteristiche di un prodotto può essere difficile per i non addetti ai lavori ma è una forma di trasparenza che migliora la scelta e l'evoluzione dei prodotti stessi verso la qualità totale. Focalizzandoci sul software bisogna inevitabilmente fare una distinzione. Esistono diversi sistemi operativi e software che sono vincolati a quest'ultimi. L'analisi immediata mostra che nel mondo esistono sistemi operativi largamente utilizzati per scelta diretta o inconsapevole e nella maggior parte dei casi quest'ultimi appartengono a compagnie private che *non mostrano* gli ingredienti *dei loro prodotti*. Nella società informatica questo non ha più ragione di esistere e la filosofia Open Sources è la soluzione evolutiva. Tutti devono avere la possibilità di *sapere quello che comprano*. Dal lato degli sviluppatori, poter visualizzare le sorgenti di un sistema operativo può voler dire migliorarne le funzionalità e renderlo maggiormente stabile ed affidabile. Il software “chiuso” e proprietario è sviluppato da un gruppo di persone. Il software Open Sources può vedere coinvolta una intera comunità di persone che apportano migliorie quotidianamente muovendosi verso la qualità totale e offrendo trasparenza assoluta. GNU/Linux rientra in quest'ultimo esempio. Nato e sviluppato in gran parte in rete, ha visto il contributo di una intera comunità di sviluppatori sparsa in tutto il mondo, un progetto collaborativo su larga scala che ha fatto crescere in pochi anni questo sistema operativo *free*, aperto. Molte persone possono affermare che GNU/Linux non sia la migliore soluzione. Io stesso ribadisco che non è attualmente la soluzione per tutti i casi ma ha una caratteristica di successo: è *software vivo costantemente in evoluzione verso la qualità totale*, capace di crescere e modellarsi alle vostre esigenze e alla vostra personalità. GNU/Linux è la soluzione più democratica e trasparente di una società moderna sempre più informatizzata e in rete.

Parlando con un conoscente che sviluppa software in ambito medico e chirurgico è stato messo in evidenza qualcosa di veramente angosciante che lascia riflettere. Il software sviluppato da quest'ultimo funziona su un sistema operativo proprietario e largamente diffuso che spesso si blocca causando errori, perdita di dati ed inevitabile disagio.

Immaginate un intervento chirurgico al cuore che viene interrotto perché il computer che gestisce le informazioni e i dati si è bloccato e vi mostra una finestra blu con gli errori riscontrati (la BSOD, *Blue Screen Of Death*). Giustificereste la mancanza di stabilità di questo sistema operativo proprietario? Giustificereste voi stessi sapendo che avete determinato questa scelta sbagliata?

Il dibattito è aperto.

F.A.Q.

Domanda: “GNU/Linux è free nel senso di gratuito?” Il concetto free di GNU/Linux include anche l'aspetto economico ma esiste un significato altamente superiore. GNU/Linux è liberamente distribuibile in base alle norme espresse dalla licenza GPL e “aperto” ovvero tutti possono vedere il codice sorgente ed apportare modifiche per migliorarlo ed ottimizzarlo alle crescenti

esigenze informatiche. Si può capire che il concetto “free” utilizzato per GNU/Linux è altamente differente da quello di “public domain”.

Domanda: “Se richiedo una versione di GNU/Linux, via Internet, da un distributore italiano o internazionale, dovrò pagare tale versione?” Ogni distribuzione offre soluzioni differenti con costi differenti. Le variazioni di prezzo sono dovute spesso al materiale correlato alla distribuzione software di GNU/Linux, manuali cartacei, gadget e supporto tecnico ma esistono soluzioni minime che includono la sola distribuzione software e sono molto economiche.

Capitolo 6

Linux User Group

L'esplosione dell'elettronica di massa ha modificato i comportamenti delle persone a livello sociale. Posso ricordare l'impatto che ebbero i primi personal computer come il Commodore VIC20 e lo ZX Spectrum verso la metà degli anni '80 ma indirettamente ho potuto cogliere un clima simile per il mondo HAM Radio. Le persone che si appassionarono a tali tecnologie era facile trovarle in mercatini particolari come le Fiere dell'elettronica, inizialmente ridotte a piccole aggregazioni di persone e ora momenti collettivi di ritrovo e scambio commerciale di oggetti e componenti elettronici. La telematica è stato un fattore di coesione ulteriore tra queste persone ampliando i confini territoriali esistenti. Quando le BBS (Bulletin Board System) erano gli unici strumenti per collegarsi tramite il proprio computer e il modem ad altri sistemi di questo tipo le persone potevano scambiare bit con una certa facilità e allo stesso modo messaggi e ulteriori informazioni, futili o meno. Tutto questo aveva costi e prestazioni imparagonabili con quello che rappresenta ora Internet ma era un passo evolutivo verso una nuova forma di comunicazione. Quando all'inizio degli anni '90, Internet prendeva il sopravvento sulle reti BBS, i vantaggi furono notevoli: i costi telefonici potevano essere ridotti visto che le connessioni erano per lo più verso provider presenti nel proprio distretto comunale piuttosto che interurbane costose. Anche le potenzialità della rete, sotto l'aspetto dei servizi, cambiava radicalmente ampliando i confini metrici delle comunicazioni. In questo clima di "apertura mentale" è cresciuto GNU/Linux e gli utenti che a sua volta utilizzavano quest'ultimo, nel rapido scambio d'opinioni e richieste con gli sviluppatori.

6.1 Linux User Group

Abbiamo precedentemente descritto i gruppi di persone formati con l'avvento dell'elettronica e la telematica. Un Linux User Group è una entità di persone che si avvicina e collabora per ampliare le proprie conoscenze in merito all'utilizzo del sistema operativo GNU/Linux e cerca di creare memoria storica per aiutare ulteriori persone tramite la creazione di documentazione e rapporti interpersonali. Esistono Linux User Group in ogni parte del mondo e l'Italia non è da meno.

Quando *parlare di Linux* poteva suonare quasi eretico in un determinato ambiente informatico/commerciale, si creò un micro-sistema che univa gli ambienti universitari tramite Internet. In quel preciso istante un frammento di storia ritornava attuale. Nacque il Pluto, il primo gruppo di aggregazione italiano, il quale si propose gli obiettivi di diffondere il *software free* o *software libero*. Il polo universitario che diede maggiore spazio alle attività fu l'università di Padova che offrì proprie risorse per sostenere l'iniziativa. Seguirono altri istituti educativi universitari che vedevano in GNU/Linux uno strumento eccellente sotto l'aspetto educativo e pratico.

L'azione del Pluto LUG è stata di fondamentale importanza per l'espansione della comunità italiana degli utenti GNU/Linux, comunità che è cresciuta come numero di partecipanti, comunità che ha saputo creare ulteriori poli di aggregazione come i LUG *locali* con copertura

Linux User Group	Web
ERLUG (Emilia Romagna)	http://erlug.linux.it
Filibusta (Crema, Mi)	http://filibusta.crema.unimi.it
FLUG (Firenze)	http://www.firenze.linux.it
GLUG (Genova)	http://genova.linux.it
GLUCh (Cagliari)	http://gluch.crs4.it
Italian Linux Society	http://www.linux.it
Linux Trent (Trento)	http://linuxtrent.trew.it
LUG Brescia	http://lugbs.linux.it
LUG Marche	http://marche.linux.it
LUG Perugia	http://stud.unipg.it/lug/
LUG Roma	http://linux.ing.uniroma1.it
LUG Torino	http://torino.linux.it
LUG Trieste	http://trieste.linux.it
LUG Verona	http://verona.linux.it
LUG Vicenza	http://www.vicenza.linux.it
NALUG (Napoli)	http://napoli.linux.it
PLUTO	http://www.pluto.linux.it
SPUTNIK (Sicilia)	http://sputnik.linux.it

Tabella 6.2: Mappa Linux User Group Italiani.

territoriale di carattere regionale, provinciale o limitatamente comunale. Sindacare su questa evoluzione della comunità italiana del software libero è superfluo: ogni trasformazione porta novità positive/negative ma se si realizza è una realtà e come tale merita considerazione.

Un Linux User Group può svolgere diversi compiti per raggiungere gli obiettivi che si è preposto nella diffusione del software libero ma devono essere sottolineati comportamenti non corretti e coerenti verso le persone che il gruppo rappresenta. Un LUG non dovrebbe prestare consulenze tecnico/commerciali a fine di lucro o legarsi a correnti politiche perchè non fanno parte del proprio *ambiente caratteristico*. Un Linux User Group, oltre a predisporre soluzioni tecniche per una comunità virtuale (servizi Internet) può altresì organizzare meeting per i propri utenti, giornate incontro tra appassionati nello stile dei *campus* americani. Dovrebbe ricevere eventuali sponsorizzazioni se quest'ultime possono migliorare l'efficienza e l'efficacia nel raggiungimento dei propri obiettivi.

La mappa italiana dei LUG

Grazie all'opera di Lindo Nepi, viene riportata di seguito una lista dei maggiori Linux User Group (tabella 6.2).

Capitolo 7

Installazione

La fase relativa all'installazione di Linux sul proprio computer è quella che inquieta gran parte delle persone. Nelle sezioni che seguono verrà spiegato che questa paura non è relativa a GNU/Linux ma a concetti che stanno alla base, argomenti comuni per tutti i sistemi operativi. Le informazioni sono particolari per gran parte dei personal computer disponibili sul mercato.

7.1 Cosa vogliamo fare?

Capire cosa si desidera fare è un risultato di valore in qualsiasi cosa nella vita. Nel caso in esame le aspettative sono ridotte ma il concetto di fondo è il medesimo, ovvero "installare un nuovo sistema operativo sul proprio computer". Esistono due principali direzioni per il raggiungimento di questo obiettivo;

- una installazione che ci permetta di far convivere due o più sistemi operativi.
- una installazione che veda come unico soggetto in causa un unico sistema operativo, in questo caso GNU/Linux.

In entrambi i casi è necessario *preparare il terreno*, determinare lo spazio dove queste soluzioni dovranno muoversi e agire. Il caso comune è il disco fisso presente in qualsiasi computer.

Partizioni

Immaginate il vostro disco fisso come una torta. Ogni fetta che facciamo di quest'ultima può essere intesa come una partizione, una sezione della superficie di memorizzazione dei dati. Per *partizionare* un disco esistono diversi strumenti, *liberi* come `fips` e `fdisk` oppure commerciali come Partition Magic per Windows, il quale offre una modalità sicuramente più semplice ed immediata dei programmi precedentemente citati¹.

7.1.1 Windows e GNU/Linux

Uno dei casi comuni di molti utenti è il desiderio di installare GNU/Linux per scoprire un mondo nuovo senza perdere quel terreno "già battuto" che può essere rappresentato da Windows, presenza inevitabile su qualsiasi computer che si acquista.

Per far convivere questi due sistemi operativi dobbiamo fare due cose:

1. Fare ordine:

¹Nota dell'Autore: Ho sempre utilizzato `fdisk` perchè quest'ultimo copre in gran parte le mie esigenze come quelle delle altre persone. Non ritengo necessario l'acquisto di un software commerciale per tale scopo quando ne esiste uno di qualità e free.

- (a) Avviare il sistema operativo Windows²³.
- (b) Eseguire il comando `chkdsk` oppure `scan c:` al prompt del DOS. E' possibile utilizzare anche il programma `scandisk`, utilità di sistema disponibile nel gruppo ACCESSORI (disponibile dal menù AVVIO). Queste applicazioni hanno lo scopo comune di controllare eventuali errori presenti sul disco fisso e correggerli.
- (c) Eseguire il comando `defrag` al prompt del DOS. E' possibile utilizzare l'applicazione UTILITÀ DI DEFRAMMENTAZIONE DISCHI presente nel gruppo ACCESSORI (disponibile dal menù AVVIO). L'utilizzo di questi strumenti sono diretti a organizzare i file presenti nella maniera più compatta.

2. Utilizzare lo spazio disponibile:

- (a) Inserire il cd-rom della vostra distribuzione GNU/Linux.
- (b) Cercare la cartella `DOSUTILS` e accedere a questa.
- (c) Copiare interamente la cartella `FIPS`.
- (d) Abbandonare l'ambiente Windows e accedere al Dos premendo il pulsante AVVIO, CHIUDI SESSIONE, RIAVVIA IL SISTEMA IN MODALITÀ MS-DOS.
- (e) Spostarsi all'interno della cartella `FIPS` precedentemente copiata con il comando `cd Fips` ed eseguire l'applicazione `fips.exe` presente. Quest'ultima applicazione analizza la superficie del vostro disco e vi permette di partizionare il vostro disco senza sormontare l'area occupata dai file presenti. Il comando vi segue interattivamente chiedendovi:
 - "DO YOU WANT TO MAKE A BACKUP COPY OF YOUR ROOT AND BOOTSECTOR BEFORE PROCEEDING (Y/N)", volete creare un back-up dei file di sistema? Si consiglia di farlo per stare tranquilli.
 - Dopo l'analisi del disco vi verrà proposto di creare una partizione definita 'NEW PARTITION'. Una volta scelta la dimensione ideale premere INVIO per confermare e successivamente il tasto "c" per applicare il partizionamento.
 - Al termine del processo comparirà il messaggio "READY TO WRITE NEW PARTITION SCHEME TO DISK. DO YOU WANT TO PROCEED?". Le possibilità sono "y" per confermare le partizioni scelte, "n" per annullare le scelte fatte.
 - Digitate la lettera "y".

Se l'esecuzione è stata portata a termine con successo è difficile riscontrare successivi problemi. Tuttavia, in casi di emergenza è necessario utilizzare il comando `restorrb.exe` riavviando il proprio computer con il dischetto creato precedentemente.

7.1.2 GNU/Linux

Se desiderate installare unicamente GNU/Linux su un disco fisso è sufficiente proseguire nella seguente sezione.

7.2 Fase di Installazione

In seguito alle scelte fatte precedentemente eseguite le seguenti operazioni:

1. Inserite il cd-rom della vostra distribuzione GNU/Linux nell'apposito lettore.

²Si presuppone abbiate installato questo sistema operativo.

³Per Windows NT è necessario seguire una procedura più complessa nel partizionamento. Esiste un How-to specifico per questa evenienza.

2. Riavviate il computer.
3. Entrate nella configurazione del BIOS premendo l'apposito tasto che vi viene indicato (genericamente il tasto funzione F1).
4. Cambiate l'ordine indicato nella sezione BOOT mettendo come primo elemento l'unità cd-rom.
5. Salvate le nuove configurazioni e riavviate il computer.

Cosa è cambiato?

All'avvio del computer verrà analizzata per prima l'unità cd-rom, invece del floppy o del disco fisso. Nel nostro caso, avendo inserito l'apposito cd-rom, verrà lanciato il programma di installazione.

7.2.1 Quante partizioni per GNU/Linux?

In una installazione comune sono necessarie due partizioni, una per il software e una denominata "swap", area destinata a coadiuvare la memoria Ram esistente nello svolgimento delle applicazioni.

Partizione Principale La dimensione può variare a seconda delle vostre esigenze e della vostra curiosità. Il mio consiglio è di fare una installazione completa (oltre 1 Gbyte) per poter esplorare le numerose applicazioni presenti e le loro potenzialità. Se le vostre intenzioni sono ridotte al semplice sistema operativo vi basteranno meno di 100 Mbyte.

Partizione Swap La dimensione viene fissata genericamente al pari della memoria Ram esistente. Se disponete di 64Mbyte di Ram è quindi consigliabile una partizione di Swap di uguale dimensione.

Per creare queste partizioni ogni distribuzione di GNU/Linux offre diverse soluzioni (vedi il programma DISK DRUID per Red Hat Linux) ma la scelta storica ricade su una applicazione a riga di comando chiamata fdisk, da tempo testata e garantita.

Ricollegandoci alla sezione precedente, in entrambi i casi di installazione (Windows/Linux o unicamente Linux) dovrete dividere la partizione che avete destinato a GNU/Linux nelle due parti precedentemente citate, partizione principale e partizione di Swap.

7.2.2 Utilizzare fdisk

Questa applicazione è minimale nell'interattività ma molto potente. Malgrado le numerose opzioni si può analizzare un percorso comune per gli utenti.

1. Eseguire `fdisk`. E' possibile farlo durante le fasi di installazioni di GNU/Linux o successivamente durante la normale esecuzione.
2. Digitate la lettera "p" per visualizzare le partizioni presenti. Potrete notare che ogni partizione presenta una serie di dati tra i quali un proprio DEVICE BOOT (ad esempio `/dev/hda1`) e il tipo di "System". Se avete creato le partizioni da Windows troverete nell'ultimo campo valori come VFAT o FAT16, FAT32, i tipici file system⁴ di questo sistema operativo.
3. Cambiate il partizionamento del vostro disco fisso. Utilizzate i tasti 'd' per cancellare le partizioni già esistenti e il tasto "n" per crearne di nuove. Se volete mantenere la partizione di Windows non agite su quest'ultima con le opzioni indicate.

⁴Un file system è uno strumento per la gestione e l'organizzazione dei file.

4. Digitate nuovamente la lettera "p" per visualizzare ogni cambiamento. Le partizioni create in questa fase saranno di tipo LINUX NATIVE⁵ le quali vanno bene per l'installazione della distribuzione ma non per la partizione Swap. Per quest'ultima infatti è necessario digitare la lettera "t", indicare la partizione in questione, visualizzare i tipi presenti (lettera "l") e confermare la scelta. La codifica per le partizioni Linux Swap è il numero "82".
5. Visualizzate lo stato della vostra configurazione con la lettera "p". Se siete soddisfatti confermate il tutto digitando il tasto "w".

Esiste una versione di fdisk anche per Dos ma il numero di file system supportati è minore e sono esclusi i tipi relativi a GNU/Linux. Sono inevitabili le considerazioni negative che nascono da questo ambiente che cerca di ostruire soluzioni alternative e concorrenti con ogni mezzo disponibile. Una buona mossa per una compagnia ma quanto di tutto questo è positivo ai fini evolutivi e sociali?

Nota su ReiserFS

Molte distribuzioni GNU/Linux hanno introdotto l'installazione e la configurazione di questo tipo di file system. ReiserFS è particolarmente attraente perchè introduce un file system journaled, un sistema veloce e sicuro in caso di problemi o salti di corrente elettrica che potrebbero rovinare i dischi fissi e i relativi dati contenuti. Un file system journaled può risultare un concetto nuovo per gli utenti Windows proprio perchè non ne esiste uno per questo tipo di sistemi operativi ma può risultare particolarmente utile man mano che cresce la dimensione dei dischi fissi. Utilizzare ReiserFS è già una garanzia nel momento stesso che scrivo il presente manuale quindi è possibile scegliere quest'ultimo in alternativa al classico ext2.

F.A.Q.

Domanda: "Posso avere dei problemi installando GNU/Linux sul mio computer portatile?"

Il supporto per i computer portatili è veramente ampio anche se esistono alcune limitazioni. Per avere maggiori chiarimenti si può consultare il seguente sito Internet:

<http://www.cs.utexas.edu/users/kharker/linux-laptop/>

Esistono documenti come il Laptop-Howto, PCMCIA-Howto e USB-Howto.

Domanda: "Ho una scheda sonora e vorrei utilizzarla. Cosa devo fare?" Gran parte delle schede sonore attualmente in commercio sono supportate. Un metodo generico per la configurazione è l'utilizzo del programma isapnp, comune in gran parte delle distribuzioni GNU/Linux ma sicuramente per utenti con una certa esperienza. La distribuzione Red Hat offre uno strumento chiamato `sndconfig` capace di testare la scheda sonora posseduta e configurarla automaticamente.

Domanda: "Vorrei disinstallare LILO dal mio disco fisso. Cosa devo fare?" In ambiente Windows è necessario aprire una finestra del prompt e digitare:

```
c:\FDISK /MBR
```

Con GNU/Linux bisogna digitare al prompt:

```
lilo -u
```

⁵Linux Native è rappresentato dal file system ext2.

Domanda: “Tento di lanciare il Server X Window ma ricevo degli errori particolari. Cosa devo fare?” Nella maggior parte dei casi i problemi sono relativi a configurazioni errate del file `XF86Config`, dove risiedono tutti i parametri necessari per l’esecuzione. In prima istanza consiglio di utilizzare `XF86Setup`, oppure `xconfigurator` (Red Hat) o `sax` (S.u.S.E.) per ricreare questo file essenziale, probabilmente inesistente o corrotto. Se i problemi persistono si consiglia di visitare il sito:

<http://www.xfree86.org/>

controllando che la propria scheda grafica sia supportata.

Domanda: “Esistono documenti che elenchino l’hardware supportato da GNU/Linux?” Ogni grossa distribuzione pubblica sui propri siti Internet queste liste e a tale scopo consultare questi documenti on-line può risultare molto utile per capire i limiti delle possibilità e orientarsi successivamente verso l’acquisto di hardware particolare.

Capitolo 8

Il Sistema

“GNU/Linux è un sistema multitasking a memoria protetta e multiutente”

multitasking perché riesce a gestire più processi (programmi, applicazioni) contemporaneamente.

memoria protetta ogni processo in corso utilizza una ben determinata area di memoria per il proprio funzionamento. Questa caratteristica offre sicurezza di sistema garantendo la sua integrità nonostante il blocco di un processo.

multiutente perché diverse persone possono collegarsi al sistema contemporaneamente e gestire le proprie attività.

Queste caratteristiche sono comuni in ambienti Unix/Linux, ma non lo sono in altri sistemi operativi.

8.1 Processi in fase di avvio

System V Init

Init è un programma che viene avviato dal kernel nel momento che il computer viene avviato. Si occupa di avviare tutti i processi necessari allo svolgimento delle operazioni comuni. Il modello SysV Init è diventato lo standard in ambiente GNU/Linux perché è molto più semplice e flessibile da utilizzare del tradizionale BSD Init.

Le configurazioni della distribuzione Linux Red Hat considerate vengono concentrate nella directory `/etc/rc.d/` dove potrete trovare le seguenti directory:

```
init.d
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
```

La disposizione e l'organizzazione degli script di inizializzazione dei runlevel è caratteristica di ogni distribuzione ma è una questione di organizzazione e non differisce nel metodo.

Gestione dei demoni di sistema

La directory `init.d` contiene un numero cospicuo di file che vengono eseguiti per eseguire i principali comandi e processi di sistema. Se volessimo lanciare un processo successivamente alla fase di Avvio è possibile farlo digitando:

```
[root@dada /root]$ /etc/rc.d/init.d/lpd start
```

Nel caso preso in esame decidiamo di lanciare il demone di stampa.

```
[root@dada /root]$ /etc/rc.d/init.d/lpd stop
```

In questo caso fermiamo il processo.

```
[root@dada /root]$ /etc/rc.d/init.d/lpd restart
```

In quest'ultimo caso viene riavviato il servizio, soluzione molto utile quando vengono apportate nuove configurazioni e si desidera che siano attivate immediatamente senza riavviare il computer. La sequenza di esecuzione seguita da `init.d` è la seguente:

- `init` esegue il file `/etc/rc.sysinit`
- `init` avvia tutti i file per il runlevel specifico
- `init` avvia il file `/etc/rc.local`

I Runlevel di Init

Esistono 7 runlevel e i relativi file di configurazione sono divisi nelle directory precedentemente viste in `/etc/rc.d/` nella forma `rc#.d`. I runlevel determinano particolari stati del vostro sistema:

- 0 Sistema spento.
- 1 Modalità utente singolo.
- 2 Modalità multi-utente senza servizio NFS.
- 3 Modalità multi-utente, la modalità di base.
- 4 Non utilizzata.
- 5 Modalità grafica in ambiente X.
- 6 Riavvio.

Il tipo di runlevel viene indicato dal file `/etc/inittab` che presenta al suo interno una linea del tipo:

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

Se è nelle vostre intenzioni avviare il vostro sistema direttamente con l'interfaccia X Window basterà sostituire il valore 3 con 5, quello del runlevel specifico. Il vostro file `/etc/inittab` potrà risultare come segue:

```
# Default runlevel. The runlevels used by RHS 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:

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

10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6

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

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

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

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

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

# Run xdm in runlevel 5
# xdm is now a separate service
x:5:respawn:/etc/X11/prefdm -nodaemon
```

8.2 File System

Un file system è il metodo e la struttura dei dati che un sistema operativo utilizza per gestire i file disposti sulle unità fisiche presenti (disco fisso, floppy, cd-rom). La parola viene spesso utilizzata per indicare una partizione o un disco dedicato ad una particolare funzione. Dicendo "ho due file system" intendo indicare che ho due partizioni sul mio disco fisso con due differenti metodi di gestione dei file; è il caso comune per la maggior parte delle persone che tengono su una partizione GNU/Linux e un altro sistema operativo sull'altra (es. Windows).

Con l'affermazione "creare il file system" si vuol intendere l'inizializzazione di una partizione o un disco utilizzando il metodo di gestione dei file scelto. Ogni sistema operativo utilizza un

file system in maniera nativa, di base, e ne supporta ulteriori per determinate esigenze. Questo è il caso di GNU/Linux che adotta il file system ext2 in forma nativa ma supporta una ampia e variegata serie di file system esistenti: minix, xia, ext, msdos, umsdos, iso9660 (il file system standard per i cd-rom), nfs (network file system), hpfs (os/2), sysv (systemV/386, Coherent e Xenix), xfs (Silicon Graphics), vfat (Windows 95/98), ntfs (Windows NT) ed altri ancora.

Organizzazione dei file Esiste uno standard denominato FHS (Filesystem Hierarchy Standard) che descrive la disposizione dei file e delle directory su sistemi di tipo Unix-compatibili. Le distribuzioni GNU/Linux più diffuse si attengono anche loro a queste specifiche ed organizzano il proprio file system nella struttura ad albero successivamente descritta.

/ Viene denominata la root di sistema. E' la base della nostra alberatura, il livello più alto del nostro sistema di riferimento. Genericamente non contiene file ma solamente le altre importanti directory di sistema.

/bin Directory dove risiedono i comandi basilari utilizzati durante l'avvio del sistema e per le procedure basilari di autenticazione.

/sbin Directory dove risiedono i comandi basilari per la gestione del sistema, chiaramente non destinati agli utenti.

/dev Directory dove sono posizionati i dispositivi necessari per accedere a periferiche o sistemi di memorizzazione.

/etc Directory dove sono disposti i file di configurazione del sistema.

/home Directory dove vengono dislocate le home degli utenti, ovvero lo spazio riservato ad ogni singolo utente (es. /home/darshan è la directory home dell'utente darshan).

/lib Directory delle librerie condivise dai programmi e utili per il loro funzionamento.

/proc E' un file system virtuale inesistente sul disco. Viene utilizzato per disporre informazioni di sistema.

/tmp Directory dei file temporanei.

/var Dati variabili, code di stampa

/usr E' la directory che contiene gran parte dei programmi esistenti nel sistema e per questo motivo ha dimensioni superiori alle directory precedentemente esaminate. Al suo interno esiste una ulteriore gerarchia:

/usr/X11R6 Directory destinata al sistema X Window

/usr/bin Gran parte dei comandi utente.

/usr/games Giochi e programmi educativi.

/usr/include Directory destinata a contenere i file header necessari per la compilazione e creazione di programmi.

/usr/lib Librerie.

/usr/local Gerarchia locale.

/usr/sbin Programmi di sistema non essenziali.

/usr/share Dati indipendenti dall'architettura hardware

/usr/src Directory delle sorgenti.

La rappresentazione data può sembrare difficile da immaginare in una finestra di terminale ma può risultare amichevole e chiara in ambienti desktop come GNOME o KDE (figura 8.1).

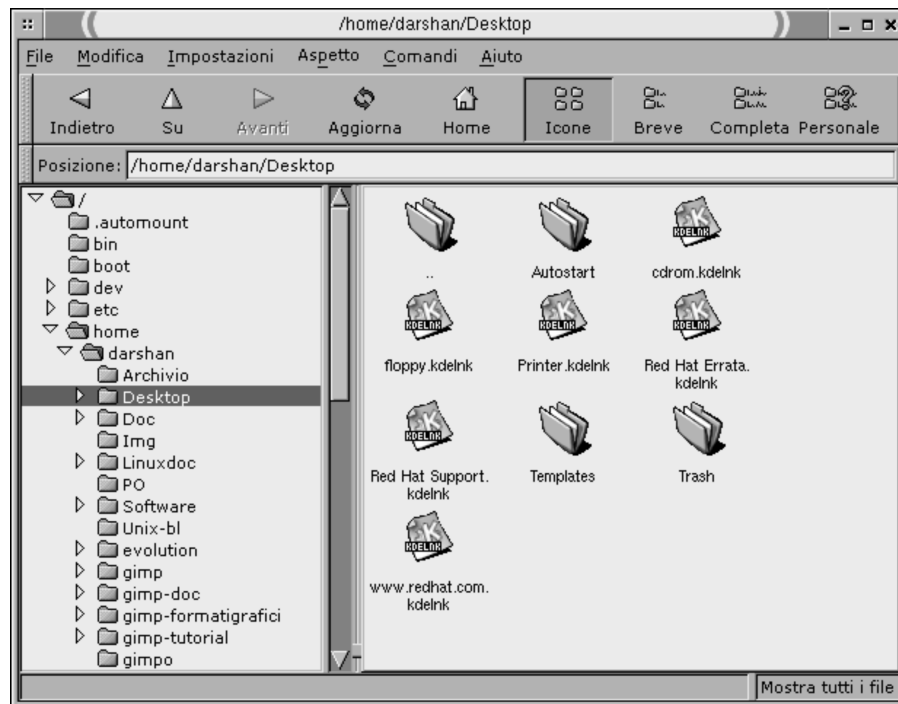


Figura 8.1: GMC, il file manager di GNOME 1.2

8.3 Permessi sui file

Un sistema multiutente necessita di una soluzione per la protezione dei file e per garantire la sicurezza delle attività degli utenti come quella del sistema stesso. In sistemi operativi Unix *everything is a file*, ogni cosa è un file con annesse le informazioni necessarie per la gestione.

Analizziamo le informazioni di alcuni file:

```
[darshan@dada darshan]$ ls -l
-rw-r--r--  2 utente gruppo    0 Nov 23 01:54 fileuno
lrwxrwxrwx  1 utente gruppo    4 Nov 23 01:55 link -> file
drwxr-xr-x  2 utente gruppo 1024 Nov 23 01:57 Directory
```

Utilizzando il comando `ls` come indicato visualizziamo i file nella directory mostrando i permessi relativi espressi dalle prime 10 combinazioni di lettere visualizzate. La prima lettera può variare come segue:

- un file comune
- d** una directory
- c** un character special file
- p** un named pipe
- s** un file socket
- l** un link simbolico

Nell'esempio visto abbiamo riscontrato un file link, una directory e un file comune. Le rimanenti 9 lettere rappresentano a gruppi di 3 i permessi rispettivi per l'utente *proprietario*, il *gruppo* e agli

Numero Ottale	Permesso	Descrizione
0	---	Nessuna autorizzazione
1	--x	Esecuzione
2	-w-	Scrittura
3	-wx	Scrittura, Esecuzione
4	r--	Lettura
5	r-x	Lettura, Esecuzione
6	rw-	Lettura, Scrittura
7	rwX	Lettura, Scrittura, Esecuzione

Tabella 8.1: Permessi sui file.

altri, cioè tutti gli utenti nel sistema non considerati. Per modificare i permessi sui file è necessario utilizzare il comando `chmod` utilizzando determinate combinazioni numeriche presentate nella tabella 8.1.

Se desideriamo cambiare i permessi sul file `fileuno` dovremo digitare una combinazione di numeri nella forma seguente:

```
[darshan@dada darshan]$ chmod 700 fileuno
```

Un modo alternativo per la medesima azione è il seguente:

```
[darshan@dada darshan]$ chmod u+rxw fileuno
```

Nel modo illustrato sono stati decretati sul file in esame i diritti di lettura, scrittura ed esecuzione per il solo proprietario. La seconda forma può risultare più semplice da apprendere perchè indica quali dei tre soggetti vengono considerati e quali diritti questi possono avere. In forma estesa:

```
[darshan@dada darshan]$ chmod ugo+rxw fileuno
```

Dove `u` è user (utente proprietario), `g` è group (il gruppo di appartenenza), o è other (tutti quelli che non sono il proprietario o non sono presenti nel gruppo di riferimento). Visualizziamo il cambiamento avvenuto:

```
[darshan@dada darshan]$ ls -l fileuno
-rwxrwxrwx 1 darshan darshan 0 lug 15 18:06 fileuno
```

Per togliere il permesso di eseguibilità la forma si discosta di poco:

```
[darshan@dada darshan]$ chmod ugo-x fileuno
```

8.4 Le Shell

La necessità dell'uomo di dialogare con la macchina ha portato lo sviluppo di uno strumento capace di mediare tra essi e rendere flessibile e minima la distanza tra "quello che vogliamo fare" e l'effettiva esecuzione: la shell.

Le Shell a caratteri

La shell è un interprete che legge ed esegue dei comandi imposti dagli utenti. A differenza di particolari sistemi operativi dove si è vincolati ad una sola ed unica shell, nei sistemi operativi

Unix / Linux, è possibile scegliere quale shell si desidera utilizzare a seconda delle propria esperienza e delle proprie necessità. Per importanza storica e per diffusione le shell che meritano di essere citate sono la shell Korn, Bourne, Bash (pagina 79) e la shell C. In ambito del sistema operativo GNU/Linux viene adottata la shell Bash, strumento flessibile che risponde alle esigenze più particolareggiate di gestione del sistema.

Le Shell grafiche

Nel corso dell'evoluzione dei sistemi operativi, sono stati creati strati superiori all'interfaccia a caratteri per permettere un approccio maggiormente istintivo e comodo con il computer. In ambienti di tipo Unix l'interfaccia di questo tipo è rappresentata da una combinazione software tra il sistema X Window¹ e un'ampia serie di windows manager e ambienti desktop. A pagina 129 troverete adeguati riferimenti.

¹X Window System è un marchio registrato di The Open Group.

Capitolo 9

La Bash

La Bash (Bourne Again Shell) include quasi tutte le funzioni offerte dalle shell C e Korn e ne aggiunge ulteriori che ampliano largamente le possibilità di un utente. Un'ulteriore nota di rilievo di questa shell è la sua forma free secondo la filosofia Open Sources, che da un lato permette di ricevere i miglioramenti di una comunità di sviluppatori e dall'altro è liberamente utilizzabile per la sua natura totalmente gratuita.

Lo storico dei comandi (history)

Ogni comando lanciato viene memorizzato in una sequenza storica. Questa caratteristica serve a riutilizzare i comandi senza dover ridigitarli ogni volta scorrendo con i tasti direzione (le frecce per muoversi in alto, basso).

Alias

Per abbreviare la digitazione di comandi con molte opzioni è possibile utilizzare gli alias. Questi ultimi vengono utilizzati nei file di configurazione della shell in seguito trattati nella forma seguente:

```
alias ls="ls -color"
```

L'esempio crea un alias del comando ls per utilizzare i colori e dev'essere inserito nel file `.bashrc` presente nella propria home directory.

Standard Input, Standard Output

Lo standard output, o stdout, è rappresentato da quello che vedete sul vostro monitor, risultato dei comandi digitati ed eseguiti. Nel momento stesso che voi digitate sulla vostra tastiera, utilizzate lo standard input, o stdin.

Standard Error

La shell permette di filtrare i messaggi di errore dallo standard output. Fanno parte dello standard error tutti i messaggi che possono essere riportati da applicazioni mal funzionanti o mal configurate. Per poter gestire e immagazzinare in un file questi errori è necessario utilizzare una propria sintassi. Nell'esempio che segue memorizziamo gli eventuali errori che possono sorgere dal lancio del Server X:

```
[darshan@dada darshan]$ X 2> errori.txt
```

Come potete notare il tutto si concentra nei caratteri "2>".

Ridirezionamento dei comandi

Se desideriamo redirigere il risultato di un comando dallo standard output su un file è necessario utilizzare il carattere “>” come segue:

```
[darshan@dada darshan]$ ls > esempio.txt
```

Se il file `esempio.txt` esisteva precedentemente verrà sostituito il contenuto del file esistente con il risultato del comando utilizzato. Se volessimo aggiungere il risultato del nostro comando accodandolo al contenuto del file esistente raddoppiamo il carattere di ridirezione:

```
[darshan@dada darshan]$ ls >> esempio.txt
```

E’ possibile invertire il percorso da standard input a standard output utilizzando il carattere “<” come segue:

```
[darshan@dada darshan]$ cat < /etc/passwd
```

Il comando digitato visualizza il contenuto del file `/etc/passwd`. Se decidessimo di ridirigere il risultato di un comando verso un ulteriore comando è necessario utilizzare il pipe “|” come segue:

```
[darshan@dada darshan]$ ls -la | more
```

Il comando digitato mostra il contenuto di una directory e lo passa al comando `more` per consultare il risultato pagina per pagina sullo standard output.

Wildcards

Nell’uso quotidiano dei comandi capita spesso di dover “automatizzare” delle procedure agendo su un numero variabile di file. A tale scopo vengono utilizzati dei “caratteri jolly”, che permettono di agire su una lista numerosa di file. I metacaratteri supportati dalla gran parte delle shell sono:

* espande zero o più caratteri

? espande un unico carattere

[a-z] espande un carattere nell’intervallo a-z

[abc] espande un carattere nell’insieme (a, b, c)

^ vieta la corrispondenza successiva (ad esempio [^abc])

~ rappresenta la home directory dell’utente

Per disabilitare l’uso dei metacaratteri è necessario anteporre il carattere “\”.

Nell’esempio che segue viene dimostrato come utilizzare un comando della shell associandolo con il metacarattere “*” per rimuovere tutti i file che hanno “stat” come parte iniziale del nome:

```
[darshan@dada darshan]$ rm -fr stat*
```

Nell’esempio che segue viene eseguito il comando per consultare il file “esempio” presente nella nostra directory “home” da una diversa posizione nel file system di sistema:

```
[darshan@dada darshan]$ cat ~/esempio
```


I comandi utilizzati verranno spiegati in seguito.

Per ampliare le proprie conoscenze sulla bash si consiglia di consultare il sito Internet <http://www.linuxdoc.org> e scaricare le seguenti guide:

- Bash Prompt HOWTO
- Bash Programming HOWTO

Capitolo 10

Pratica

L'elenco dei comandi di seguito riportati è in ordine alfabetico per dare la possibilità di ritrovare il comando conoscendone il nome. Può capitare che vengano citati comandi non discussi e si possa sottintendere informazioni importanti. A tale scopo è stato creato il capitolo seguente, esposizione pratica di lavoro con la shell dei comandi. Le opzioni riportate con i comandi sono una minima parte di quelle esistenti. Molto spesso ogni comando offre una ampia possibilità di scelta ma la nostra scelta è caduta sulla quotidianità delle operazioni.

10.1 Eseguitività

Ogni comando può differire da un altro per il linguaggio in cui è stato programmato ma esistono caratteri comuni per ogni sistema Unix/Linux. Tutti i comandi sono tendenzialmente organizzati in particolari posizioni nel file system. Le posizioni dove il sistema cerca i comandi sono indicati nella variabile di ambiente `PATH`. Per conoscere queste locazioni digitate:

```
[root@dada /root]# echo $PATH
/usr/sbin:/bin:/usr/bin:/usr/bin/X11
```

Ogni posizione è separata dal carattere ":". Per ampliare i percorsi di ricerca comandi è necessario indicare al sistema quest'ultime digitando quanto segue:

```
[root@dada /root]# export PATH=$PATH:/usr/local/bin
```

rieseguendo il comando precedente la situazione cambierà come segue:

```
[root@dada /root]# echo $PATH
/usr/sbin:/bin:/usr/bin:/usr/bin/X11:/usr/local/bin
```

In questo caso è stata aggiunta la directory `/usr/local/bin` al `PATH` e il sistema potrà lanciare i comandi presenti in quest'ultima posizione. E' possibile lanciare comandi anche fuori dai percorsi presenti nella variabile `PATH` e per fare questo è necessario precedere il nome di comando con le lettere `./` come segue:

```
[root@dada /root]# ./comando
```

Resta inteso che sia in quest'ultima situazione come nelle precedenti il comando o file deve avere i diritti di esecuzione precedentemente analizzati.

10.2 Comandi

Come è già stato anticipato, questo manuale è stato impostato su una distribuzione compatibile con Red Hat Linux non tanto per ragioni di qualità ma di diffusione tra le persone e livello di semplicità. Molto spesso le uniche differenze tra una distribuzione GNU/Linux e l'altra sono minime e si possono risolvere in semplici disposizioni fisiche dei file differenti che non alterano certo il funzionamento del sistema.

10.2.1 adduser

Questo comando è necessario per aggiungere un utente al sistema ed è una operazione concessa unicamente all'utente `root`, l'amministratore. Digitate:

```
[root@dada /root]# adduser nome_scelto
```

dove `nome_scelto` è il nostro utente da aggiungere. Per configurare una password a questo nuovo utente fate riferimento al comando `passwd` a pagina 106.

10.2.2 apropos

Questo comando cerca all'interno degli elenchi `whatis` (pagina) tutte le corrispondenze relative alla parola indicata. Esempio:

```
[darshan@dada darshan]$ apropos keyboard
```

Verranno presentate tutte le stringhe di testo contenenti la parola "keyboard".

10.2.3 apt-get

Questo comando è relativo alla distribuzione Debian GNU/Linux. Ha funzionalità correlabili ai comandi `rpm` (pagina 108) e `dpkg` (pagina 91) aggiungendo caratteristiche ulteriori e rendendo la procedura di gestione dei pacchetti software molto più semplice per un utente della distribuzione pocanzi citata. Per eseguire l'installazione di un pacchetto `.deb` è necessario digitare:

```
debian:~$ apt-get install nomefile.deb
```

Se il pacchetto software è già installato nel sistema viene compiuto l'aggiornamento automatico di quest'ultimo con la nuova versione. Per procedere alla disinstallazione di un pacchetto software digitate:

```
debian:~$ apt-get remove nomefile
```

Il comando è utile anche per aggiornare tutti i pacchetti software del proprio sistema sfruttando la connettività ad Internet. Questo è possibile inserendo un'apposita dicitura nel file `/etc/apt/sources.list` come segue:

```
deb ftp://ftp.it.debian.org/debian unstable main contrib non-free
```

e digitare al prompt:

```
debian:~$ apt-get update
```

Quest'ultimo comando sfrutterà la connessione (se attiva) per aggiornare automaticamente il sistema e i relativi pacchetti software presenti.

10.2.4 bzip2

Questo comando serve a comprimere un file tramite l'algoritmo bz2. Per comprimere un file è necessario digitare:

```
[darshan@dada darshan]$ bzip2 nuovofile
```

Il file nuovofile verrà compresso e verrà rinominato come "nuovofile.bz2". Per scompattare il file presente è necessario utilizzare il comando *bunzip2*:

```
[darshan@dada darshan]$ bunzip2 nuovofile.bz2
```

10.2.5 cal

Questo comando stampa un calendario per uno specifico mese e/o anno. Digitando il comando senza opzioni si otterrà il calendario del mese corrente:

```
[darshan@dada darshan]$ cal
      novembre 1999
do lu ma me gi ve sa
   1  2  3  4  5  6
  7  8  9 10 11 12 13
 14 15 16 17 18 19 20
 21 22 23 24 25 26 27
 28 29 30
```

Un ulteriore esempio che mostra il mese di giugno (6) del 1969:

```
[darshan@dada darshan]$ cal 6 1969
      giugno 1969
do lu ma me gi ve sa
   1  2  3  4  5  6  7
   8  9 10 11 12 13 14
 15 16 17 18 19 20 21
 22 23 24 25 26 27 28
 29 30
```

Per mostrare l'intero calendario di un anno:

```
[darshan@dada darshan]$ cal 1999
```

10.2.6 cardctl

Questo comando è una utilità di controllo delle schede PCMCIA disponibili per computer portatili. Vediamo alcune opzioni:

```
[root@dada /root]# cardctl status
```

Mostra lo stato attuale della scheda presente.

```
[root@dada /root]# cardctl config
```

Mostra le configurazioni della scheda presente.

```
[root@dada /root]# cardctl ident
```

Visualizza le informazioni identificative della scheda.

```
[root@dada /root]# cardctl suspend
```

Interrompe la corrente che alimenta la scheda.

```
[root@dada /root]# cardctl eject
```

Disabilita la gestione e interrompe l'alimentazione della scheda.

```
[root@dada /root]# cardctl insert
```

Avvisa i driver di gestione che la carta è stata nuovamente inserita e deve essere attivata.

10.2.7 cat

Questo comando mostra il contenuto di un file di testo e può concatenare diversi file insieme. Per aggiungere il contenuto del file uno in coda al file due è necessario digitare la seguente sequenza di comandi:

```
[darshan@dada darshan]$ cat fileuno >> filedue
```

Per consultare il contenuto di un file:

```
[darshan@dada darshan]$ cat fileuno
```

Se il contenuto di un file è ampio, durante l'esecuzione del comando potreste non visualizzare la totalità delle informazioni presenti. A tale scopo è necessario utilizzare il comando `more` successivamente descritto.

10.2.8 cd

Questo comando permette di spostarvi nella struttura ad albero delle directory. Quando un utente si collega al sistema viene automaticamente posizionato nella sua directory "home", usualmente nella posizione `/home/nome_utente/`. Per spostarsi in una sottodirectory chiamata "Tesi" disposta nella corrente directory:

```
[darshan@dada darshan]$ cd Tesi/
```

Per spostarsi nella directory `/home/darshan/Tesi/Cap06/` digitate:

```
[darshan@dada darshan]$ cd /home/darshan/Tesi/Cap06/
```

Per spostarsi dalla directory attuale a quella superiore, digitate:

```
[darshan@dada darshan]$ cd ..
```

Per spostarvi nella directory "root", digitate:

```
[darshan@dada darshan]$ cd /
```

Per ritornare nella vostra “home” directory, digitate:

```
[darshan@dada darshan]$ cd
```

10.2.9 chattr

Questo comando cambia gli attributi di un file su un file system di tipo `ext2` e può essere particolarmente sotto aspetti di backup di sistema e sicurezza. Gli obiettivi del comando possono essere alquanto oscuri per un utente alle prime armi ma verranno utilizzati alcuni esempi reali per semplificare l’apprendimento.

```
[darshan@dada darshan]$ chattr +i documento.txt
```

Il comando appena digitato rende “immutabile” il file `documento.txt` da modifiche, aggiunte e non si potranno creare link a quest’ultimo. Se volessimo rendere “immutabile” una directory con tutti i file presenti al suo interno avremmo dovuto digitare:

```
[darshan@dada darshan]$ chattr -R +i documenti/
```

Ulteriori attributi da dare ad un file (o directory) sono i seguenti:

```
[darshan@dada darshan]$ chattr +a documento.txt
```

L’attributo “a” rende il file gestibile solo in modalità `append` ovvero è permesso solamente apportare aggiunte al file.

```
[darshan@dada darshan]$ chattr +c documento.txt
```

Un file con l’attributo “c” viene automaticamente compresso sul disco dal kernel. La lettura del file restituisce i dati decompressi mentre in scrittura ogni informazione viene compressa e poi scritta sul disco.

```
[darshan@dada darshan]$ chattr +d documento.txt
```

L’attributo “d” candida un file per il backup tramite il comando `dump`.

```
[darshan@dada darshan]$ chattr +s documento.txt
```

Esistono ulteriori attributi per il comando e possono essere consultati tramite l’apposita guida in linea:

```
[darshan@dada darshan]$ man chattr
```

Abbiamo visto che per aggiungere attributi è stato utilizzato il carattere “+”. Per toglierli dobbiamo alternativamente utilizzare il carattere “-” come nell’esempio seguente:

```
[darshan@dada darshan]$ chattr -i documento.txt
```

10.2.10 chmod

Questo comando cambia i permessi associati per ogni file presente nel vostro sistema. Di seguito viene mostrata la serie di informazioni relative ad un singolo file. Il comando in questione agisce

sulla prima decade di lettere, quelle che specificano i “diritti”.

```
-rw-r-r- 1 root root 133444 Sep 1 21:14 gsg.tex
```

La forma di utilizzo del comando per rendere il file eseguibile, modificabile, leggibile dal proprietario e togliere i diritti agli altri può essere il seguente:

```
[darshan@dada darshan]$ chmod 700 gsg.tex
```

e il file considerato cambierà i permessi nel seguente modo:

```
-rwx--- 1 root root 133444 Sep 1 21:14 gsg.tex
```

Per capire come funzionano i permessi su un file consultare la tabella 8.1.

10.2.11 compress

E’ un comando storico per la compressione dei dati per Unix. Esempio di compressione di un file:

```
[darshan@dada darshan]$ compress nomefile
```

Ogni file viene compresso e rinominato con l’aggiunta del suffisso “.Z”. Per decomprimere un file di questo tipo è necessario digitare:

```
[darshan@dada darshan]$ compress -d nomefile.Z
```

E il file verrà rinominato senza l’estensione “.Z”.

10.2.12 cp

Questo comando serve per copiare dei file, preservare gli originali e crearne una copia identica. Se già disponete di un file digitate:

```
[darshan@dada darshan]$ cp gsg.tex Documenti/gsg.tex
```

Nel caso sopracitato ho copiato un file presente nella directory corrente nella sottodirectory “Documenti”. Se nella directory dove ho copiato il file ci fosse stato già un documento con quel nome lo avrei sovrascritto. Per evitare questo aggiungo la seguente opzione:

```
[darshan@dada darshan]$ cp -i theGimp.tex Documenti/gsg.tex
```

L’opzione attiva una modalità interattiva che mi comunica particolari situazioni come quella descritta.

10.2.13 cpio

E’ un archiviatore di file simile al comando `tar` (pagina 112). Per lungo tempo `tar` e `cpio` erano gli archiviatori maggiormente utilizzati, poi con l’avvento di GNU/Linux il primo ebbe il sopravvento. Numerose sono le opzioni di questo comando ma ci limiteremo alle due sole principali, la creazione e l’estrazione di un archivio `cpio`. Per la creazione la forma è la seguente:

```
[root@dada /root]# cpio -o < file/ > file.cpio
```


Per l'operazione inversa, l'estrazione di un archivio `cpio`, la forma è questa:

```
[root@dada /root]# cpio -i --make-directories < file.cpio
```

La vera utilità di questo comando risiede nell'utilizzo combinato con `rpm2cpio` (pagina 110).

10.2.14 cron

Questo applicativo è utile quando si vuole automatizzare l'esecuzione di determinati processi di sistema a scadenze temporali ben definite. Per il suo funzionamento è necessario avere attivato il demone `crond` che controlla i file di configurazione e si occupa di eseguire le operazioni indicate nei limiti delle impostazioni indicate. Per verificare rapidamente se è attivo digitate:

```
[darshan@dada darshan]$ ps -ax|grep cron
```

In caso positivo apparirà quanto segue:

```
[darshan@dada darshan]$ ps -ax|grep cron
559 ?      S      0:00  crond
563 pts/1  R      0:00  grep cron
```

In caso negativo (non esiste il processo `crond` attivo) potete attivarlo sul momento. Per fare questo è necessario essere utente `root`, l'amministratore di sistema, colui che ha i diritti necessari per lanciare il demone¹ nel sistema. Digitate:

```
[darshan@dada darshan]$ su -
```

Verrà chiesta la password relativa all'utente e in caso positivo il vostro prompt cambierà come segue²:

```
[root@dada /root]#
```

A questo punto lanciate il demone digitando³:

```
[root@dada /root]# /etc/rc.d/init.d/crond start
```

e ritornate all'identità precedente con:

```
[root@dada /root]# exit
```

Il prompt dei comandi ritornerà come era. A questo punto non resta altro che modificare il file di configurazione per eseguire i processi ad una data scadenza temporale:

```
[darshan@dada darshan]$ crontab -e
```

Il comando edita un file (utilizzando l'editor `vi`, pag. 114) dove andranno impostate le vostre preferenze nella forma seguente:

minuti	ora	gg del mese	mese	gg settimanale	applicativo
data					comando

¹demone, processo che viene attivato nel sistema e procede in background, ovvero senza disturbare lo svolgimento delle ulteriori attività.

²I prompt di comandi possono cambiare tra le varie distribuzioni.

³Il percorso del comando può cambiare a seconda della distribuzione e a seconda dell'organizzazione data.

Per eseguire il comando `pon` alle 18,30 di ogni giorno la configurazione da aggiungere al file di cron sarà la seguente:

```
30 18 * * * /usr/bin/pon
```

Gli asterischi “*” indicano che il processo dovrà essere eseguito ogni giorno del mese, ogni mese, ogni giorno della settimana. A questo punto non vi resta altro che salvare il file e ritornare al prompt dei comandi iniziale. Per visualizzare il file appena modificato digitate:

```
[darshan@dada darshan]$ crontab -l
```

Se desiderate eliminare il vostro file di configurazione di cron:

```
[darshan@dada darshan]$ crontab -r
```

Per maggiori informazioni digitate al prompt `man cron`.

10.2.15 date

Questo comando mostra giorno, data, ora ed anno corrente:

```
[darshan@dada darshan]$ date
gio lug 6 11:14:54 CET 2000
```

10.2.16 df

Questo comando serve a mostrare lo spazio libero sui dischi.

```
[darshan@dada darshan]$ df
Filesystem 1k-blocks  Used Available  Use%  Mounted on
/dev/hda1   1486339 1246706   162825    88%  /
/dev/hda2    694727  618131    40712    94%  /home
/dev/hda3    121007  73550     41208    64%  /home/redhat
```

Dal risultato del comando è possibile vedere lo spazio utilizzato, lo spazio disponibile, la percentuale di utilizzo e dove sono montati i dischi analizzati.

10.2.17 diff

Questo comando serve per analizzare due file e riportare le differenze.

```
[darshan@dada darshan]$ diff -urN it.po it.po.new
```

Dopo aver digitato il comando vedrete a video le differenze tra i due file. Il comando `diff` viene utilizzato per la creazione di patch con le seguenti opzioni:

```
[darshan@dada darshan]$ diff -urN it.po it.po.new > patch
```

Per creare una patch su file binari è necessario digitare quanto segue:

```
[darshan@dada darshan]$ diff -a --binary bin bin.2 > patch
```

Per applicare i file con le differenze è necessario il comando `patch` (pagina 106).

10.2.18 dpkg

Questo comando è relativo alla distribuzione Debian GNU/Linux. E' principalmente utile per installare e aggiornare i pacchetti software sul proprio sistema e può essere comparato al comando `rpm` (pagina 108).

```
debian:~$ dpkg -i nomefile.deb
```

Questo comando compie sia l'installazione che la procedura di aggiornamento se viene riscontrato un pacchetto maggiormente recente. Per compiere l'operazione opposta, ovvero la disinstallazione del pacchetto software `.deb` digitate:

```
debian:~$ dpkg -r nomefile
```

Per evitare compromissioni del sistema è bene antecedere la seguente opzione di controllo:

```
debian:~$ dpkg --no-act -r nomefile
```

Per ottenere informazioni su un pacchetto è possibile digitare quanto segue:

```
debian:~$ dpkg -s nomefile
```

Per ottenere informazioni sulla provenienza di un particolare file presente sul sistema possiamo conoscere il pacchetto software relativo digitando:

```
debian:~$ dpkg -S blackbox
```

Per avere la lista dei pacchetti software installati nel sistema digitate:

```
debian:~$ dpkg -l
```

Una recente alternativa a `dpkg` è `apt-get` (pagina) che offre numerose caratteristiche aggiuntive offrendo maggiore semplicità di utilizzo.

10.2.19 dvipdfm

Questo comando converte file in formato `.dvi` in formato `.pdf`. Digitare:

```
[darshan@dada darshan]$ dvipdfm -o documento.pdf documento.dvi
```

Dove l'opzione `"-o"` indica il nome finale del file nel nuovo formato. Per maggiori informazioni relative consultare pagina 99.

10.2.20 dvips

Questo comando converte file in formato `.dvi` in formato `.ps` (Postscript⁴).

```
[darshan@dada darshan]$ dvips -o documento.ps documento.dvi
```

L'opzione `"-o"` indica il nome finale del file postscript che verrà generato. Esistono numerose opzioni consultabili tramite la propria pagina manuale ma quelle interessanti nell'uso comune possono essere le seguenti:

⁴Postscript è un marchio registrato da Adobe Inc.

```
[darshan@dada darshan]$ dvips -A -o pgpari.ps documento.dvi
```

```
[darshan@dada darshan]$ dvips -B -o pgdispari.ps documento.dvi
```

Dove le opzioni aggiuntive “-A” e “-B” indicano rispettivamente di creare il file `pgpari.ps` con le pagine pari dell’originale documento e `pgdispari.ps` con le pagine dispari. Quest’ultime opzioni possono risultare utili se si dispone di una comune stampante per poter stampare su entrambe i lati dei fogli. Per maggiori informazioni relative consultare pagina 99.

10.2.21 echo

Il comando `echo` serve esclusivamente a stampare sullo standard output stringhe di testo:

```
[darshan@dada darshan]$ echo "Ciao Daniele"
Ciao Daniele
```

Un altro esempio può essere la stampa a video di alcune variabili di sistema:

```
[darshan@dada darshan]$ echo $HOME
/home/darshan
```

Dove `$HOME` rappresenta la nostra directory home.

10.2.22 env

Questo comando mostra le variabili di ambiente della vostra sessione. Digitate:

```
[darshan@dada darshan]$ env
```

10.2.23 export

Questo comando viene utilizzato per esportare delle variabili di ambiente, valori che permettono al sistema di funzionare come desideriamo.

```
[darshan@dada darshan]$ export DISPLAY=monika:0.0
```

Nell’esempio appena visto si è utilizzato il comando per esportare il display di X Window dal server `dada` al computer `monika` che corrisponde ad un IP nel file `/etc/hosts`. I comandi che lanciano applicazioni per X verranno aperti sull’ultimo computer ma il carico di processo sarà della macchina dove vengono eseguiti (in questo caso il computer `dada`).

10.2.24 file

Questo comando indica il formato di un file indicato. Digitate:

```
[darshan@dada darshan]$ file esempio
esempio: ASCII text
```

Verrà indicato il formato del file che si chiama “esempio”.

10.2.25 find

Questo comando serve per fare ricerche di file nel sistema. Per eseguire una semplice ricerca del file "miofile.txt" nella directory corrente è necessario digitare quanto segue:

```
[darshan@dada darshan]$ find . -name miofile.txt -print
```

Se desiderate trovare tutti i file con estensione ".txt" è necessario digitare quanto segue:

```
[darshan@dada darshan]$ find . -name "*.txt" -print
```

In questi esempi appena visti si è utilizzato il punto "." per indicare la directory corrente. Se la nostra ricerca vuole essere estesa ad un'altra directory comportatevi come segue:

```
[darshan@dada darshan]$ find /home/ -name miofile.txt -print
```

La ricerca andrà ad analizzare se esistono file denominati "miofile.txt" nella directory /home e mostrerà il risultato finale.

```
[darshan@dada darshan]$ find / -cmin 60 -mmin 60
```

Con le opzioni appena viste cerchiamo i file cambiati e modificati entro l'arco di tempo di 60 minuti.

10.2.26 finger

Il comando finger riporta informazioni sugli utenti di un sistema, visualizzando nome, cognome, ufficio e numero di telefono, ultimo login ed ulteriori informazioni. Di seguito un esempio di utilizzo comune:

```
[darshan@dada darshan]$ finger darshan
Login: darshan
Name: Daniele Medri
Directory: /home/darshan
Shell: /bin/bash
Never logged in.
No mail.
No Plan.
```

Se l'utente darshan (preso di esempio) ha un accesso presso un server differente nella rete interna, dovrete specificare il nome del server nel seguente modo:

```
[darshan@dada darshan]$ finger darshan@erlug
```

La forma è nome utente@nome server.

Se l'utente ha un accesso presso un server differente e quest'ultimo esce dalla rete interna ed è dislocato su Internet, la forma diventa:

```
[darshan@dada darshan]$ finger darshan@erlug.linux.it
```

Il nome del server è conforme alle regole canoniche di Internet.

10.2.27 `fmt`

Questo comando serve per formattare un documento di testo con un numero definito di colonne:

```
[darshan@dada darshan]$ fmt -w 60 doc.txt > doc2.txt
```

Il comando esaminerà il file `doc.txt`, lo formatterà con un numero di 60 colonne e passerà il risultato ridirizionandolo al file `doc2.txt`.

10.2.28 `fold`

Questo comando ha funzionalità simili al comando `fmt` e aggiunge alcune caratteristiche. Se si decide di formattare un file con un numero definito di colonne e non ricevere parole troncate al termine di ogni riga sarà necessario digitare quanto segue:

```
[darshan@dada darshan]$ fold -w 50 -s doc.txt > doc2.txt
```

Verrà esaminato il file `doc.txt` formattandolo con un numero massimo di 50 colonne e disabilitando la parole troncate. Il risultato verrà passato al file `doc2.txt`.

10.2.29 `free`

Questo comando stampa sullo standard output l'utilizzo della memoria Ram e dello Swap:

```
[darshan@dada darshan]$ free
              total    used    free   shared  buffers   cached
Mem:           95504  88308   7196   40768     7568   57640
-/+ buffers/cache: 23084 72420
Swap:           60476
```

Dai dati si può notare l'esistenza di 96 Mbyte di memoria Ram, e una partizione di Swap di circa 60 Mbyte.

10.2.30 `ftp`

Per poter trasferire dei file da un server locale ad un server remoto è necessario utilizzare questo comando. E' indispensabile avere un accesso al server remoto o utilizzare una connessione anonima. Per aprire una sessione digitate:

```
[darshan@dada darshan]$ ftp
ftp>
```

Per conoscere l'insieme delle opzioni presenti digitare "help":

```
ftp> help
```

Per aprire una connessione ad un server:

```
ftp> open nomeserver
```

Per chiudere una connessione:

```
ftp> close
```

Si è accennato alla connessione anonima. L'unica sostanziale differenza è in fase di autenticazione:

```
[darshan@dada darshan]$ ftp dada
Connected to dada.
220 dada FTP server Tue Sep 21 16:48:12 EDT 1999) ready.
Name (dada:darshan): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: *****
```

Quando viene richiesta la password utilizzare il proprio indirizzo di posta elettronica. Non è una richiesta obbligatoria ma è buona regola farlo se si utilizza il servizio in questione.

10.2.31 gpg

GnuPG è la versione GNU di PGP, un programma di crittografia per la protezione della privacy personale. Questa breve descrizione non può essere esaustiva vista la vastità delle opzioni disponibili ma sintetizzerà i passi fondamentali necessari e utilizzati comunemente. Il primo passo è la generazione della directory personale di gpg lanciando il comando:

```
[darshan@dada darshan]$ gpg
gpg: /home/darshan/.gnupg: directory creata
gpg: /home/darshan/.gnupg/options: creato un nuovo file delle opzioni
```

Come secondo passo è necessario generare le proprie chiavi (pubblica/privata) lanciando il comando seguente:

```
[darshan@dada darshan]$ gpg --gen-key
```

Verrà avviata una procedura interattiva che vi aiuterà a creare la chiave chiedendovi alcune configurazioni e alcuni vostri dati personali (nome, cognome, email, commenti). Quando tutti i dati saranno confermati dovrete digitare la vostra passphrase, o frase segreta e proseguire nella creazione delle chiavi. Per visualizzare la vostra chiave pubblica:

```
[darshan@dada darshan]$ gpg --export -a
```

Per copiare la chiave su un file di testo è necessario aggiungere la seguente opzione:

```
[darshan@dada darshan]$ gpg --export -a -o chiave_pubblica.asc
```

Il file `chiave_pubblica.asc` è la vostra chiave pubblica che potete rendere disponibile in rete. Se volete importare la chiave pubblica di altre persone nella vostra e ampliare il livello di sicurezza dovrete digitare quanto segue:

```
[darshan@dada darshan]$ gpg --import altra_chiave_pubblica.asc
```

Per conoscere le chiavi importate:

```
[darshan@dada darshan]$ gpg --list-keys
```

Fingerprint Il fingerprint, o "impronte digitali" sono informazioni univoche e invariabili tratte da ogni chiave, informazioni che non cambiano importando ulteriori chiavi. Servono ad identificare la chiave di una persona e la sua validità. Per conoscere il proprio fingerprint:

```
[darshan@dada darshan]$ gpg --fingerprint
```

E' utile copiare questi dati e utilizzarli nella propria firma delle vostre email in modo che sia pubblica e rapidamente confrontabile.

Crittografia/Decrittografia Per utilizzare gpg per spedire un messaggio segreto ad una persona conosciuta è necessario digitare il seguente comando:

```
[darshan@dada darshan]$ gpg -e -a messaggio_segreto.asc
```

L'opzione "-e" significa "encrypt" e "-a" è necessaria per esportare il messaggio crittato in un file di testo. Se ora visualizziamo il file apparirà quanto segue:

```
[darshan@dada darshan]$ more messaggio_segreto.asc
```

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.0.1 (GNU/Linux)
Comment: For info see http://www.gnupg.org

hQE0AxMnLse+b6VzEAP/cgfcbq2duvNZ3Msd/fUIjKRfOSj9yvWti
pOwrbvtTm3my2pQbr+w8iy7ltxPyP+okWUyPYMYc0wZSYCxsert+
pZKkIftq9PMZO/Ef5EoMysq/kcA0j0vmrKF/Ol45VzAzYifMW043E
/iQ9LdZogUpH/w8/rs0laf5YJmSNwn1TB7q/MI0ywQh6UNNEsSerb
dekdlssd7nV1TpxIxcClj1Z4V68uyxXWUJZoc8Mn1IAhwLCnza/Q
9GZN7PaJ94qFNQE69wqGz4TUA05NWwJaqFSmwwDEmTIyyaferrerV
eue3SJGlfBqzJHJ2HsdLL4XpR3kZMcbZwzVhfdIbpI4gYDd/tLXR
-----END PGP MESSAGE-----
```

La sequenza di caratteri è il vostro messaggio protetto. Per visualizzarlo in forma "leggibile" è necessario digitare quanto segue:

```
[darshan@dada darshan]$ gpg -d messaggio_segreto.asc
```

Vi verrà chiesto di digitare la vostra frase segreta (passphrase) e in seguito alla verifica di quest'ultima apparirà il messaggio reale. Per ulteriori informazioni visualizzate la guida in linea:

```
[darshan@dada darshan]$ man gpg
```

oppure visitate il sito Internet ufficiale: <http://www.gnupg.org>.

10.2.32 grep

Questo comando è necessario per ritrarre da una fonte di ingresso (un file, standard input) una particolare serie di dati che corrisponde alle richieste poste. Una semplice forma di utilizzo può essere la seguente:

```
[darshan@dada darshan]$ grep "pippo" doc.txt
```

Il comando visualizzerà a video tutte le righe del file `doc.txt` dove sarà presente la parola "pippo". La ricerca può assumere richieste particolari e a tale scopo entrano in gioco le espressioni regolari. Un esempio pratico:

```
[darshan@dada darshan]$ grep '\<[a-z0-9-]\>' doc.txt
```


Verranno visualizzate solamente le righe che contengono i caratteri dalla lettera “a” alla lettera “z” e i numeri da “0” a “9”. Per maggiori informazioni consultare la pagina manuale del comando.

```
[darshan@dada darshan]$ man grep
```

10.2.33 gzip

Questo comando è utile per comprimere file. La forma tipica di utilizzo è:

```
[darshan@dada darshan]$ gzip documento.tex
```

Per decomprimere tale file si può ricorrere al comando `gunzip` (pagina 97).

10.2.34 gunzip

Questo comando è utile per decomprimere file. La forma tipica di utilizzo è:

```
[darshan@dada darshan]$ gunzip documento.tex.gz
```

10.2.35 kill

Questo comando è utilizzato come ultima risorsa per distruggere un programma sospeso o che si desidera riavviare. Esistono diversi segnali/opzioni che si possono utilizzare con il comando `kill` per interrompere un programma. Nell’uso quotidiano può essere comune quanto segue:

```
[darshan@dada darshan]$ kill -9 id_processo
```

Per visualizzare il codice identificativo di un processo è necessario utilizzare il comando `ps` (pagina 107).

10.2.36 halt

Questo comando è necessario per fermare il vostro computer. Una volta lanciato compierà diverse operazioni volte a interrompere i servizi attivi e disabilitare i dispositivi che compongono il file system.

```
[root@dada /root]# halt
```

Comando con funzionalità analoghe può essere `shutdown` (pagina 110).

10.2.37 history

Come è stato citato in precedenza (pagina 97), la Bash offre la possibilità di memorizzare la sequenza di comandi utilizzati da un utente. Per visualizzare lo storico dei comandi basta digitare quanto segue:

```
[root@dada /root]# history
```

Con maggiore utilità se l’output video viene rediretto al comando `more` o direttamente in un file di testo. Rispettivamente:

```
[root@dada /root]# history | more
[root@dada /root]# history > storico.txt
```

10.2.38 hwclock

Questo comando vi permette di gestire l'orologio hardware di sistema apportando modifiche a piacimento. Il primo utilizzo illustrato è quello ovvio:

```
[root@dada /root]# hwclock --show
Sat Jul 15 13:19:14 2000 -0,888683 seconds
```

Questa è l'ora presente nell'orologio di sistema. Per cambiarla è necessario aggiungere i seguenti parametri:

```
[root@dada /root]# hwclock --set --date="7/15/2000 16:45:05"
```

Esistono due modalità di gestione dei dati: *localtime*, macchina locale, o *utc* (Universal Coordinated Time). Se non viene specificato la modalità è quella locale mentre per configurare l'impostazione nel secondo tipo è necessario indicarlo:

```
[root@dada /root]# hwclock --set --date="7/15/2000 16:45:05" --utc
```

Gli orologi in un sistema GNU/Linux Esistono due principali tipi di orologi: *l'orologio hardware* e *l'orologio di sistema*. Il primo è indipendente da ogni programma di controllo e rimane attivo anche a computer spento. Spesso viene definito con diverse terminologie: *rtc* (Real Time Clock), *BIOS clock*, *CMOS clock*. L'orologio di sistema è il sistema di gestione presente in ogni kernel di ogni sistema operativo; al momento dell'avvio della macchina vengono consultati i dati dell'orologio hardware e l'orologio di sistema si configura rispettando i *timezone* indicati, le fasce del fuso orario. Nei sistemi Unix l'orologio di sistema ha come data iniziale l'ora 00:00:00 del 1 Gennaio 1970 UTC.

Una nota di carattere generale. Quando si parlava di Millenium Bug, il baco di fine millennio che doveva colpire i computer di tutto il mondo, si intendeva una inadeguatezza degli orologi hardware come degli orologi di sistema dei vari sistemi operativi che poteva creare problemi nello svolgimento delle normali attività esecutive.

10.2.39 irc

Questo comando lancia un minimale programma per la chat di tipo IRC.

```
[darshan@dada darshan]$ irc irc.tin.it
```

Digitando il solo comando parte il collegamento in automatico con il server italiano `irc.tin.it` che fa parte della rete Ircnet, una delle reti molto diffuse. Durante il collegamento compariranno messaggi informativi e vi verrà chiesto il nickname, ovvero il vostro "nomignolo" per la chat. Se desiderate passare quest'ultimo come parametro al lancio del programma è necessario digitare quanto segue:

```
[darshan@dada darshan]$ irc nickname irc.tin.it
```

Una volta collegati al server specificato con il nickname scelto è necessario entrare in un "canale" di persone, piazze virtuali dove le persone si trovano per comunicare. Se sapete già in quale canale andare digitate al prompt:

```
/join #italia
```

Dove `/join` indica di entrare nel canale `#italia`. Una volta entrati potete comunicare con le altre persone collegate in quel momento. Se desiderate uscire dal canale digitate:

```
/leave #italia
```

Se desiderate ottenere la lista dei canali esistenti e aperti in quel preciso momento digitate:

```
/list -MIN 5 -MAX 20 -PUBLIC -NAME
```

Il comando `/list` richiede tutti i canali aperti. Le opzioni aggiunte servono a limitare il flusso di dati richiesti e cercare i canali con un numero di partecipanti che varia da 5 (MIN) a 20 (MAX) utenti, che sia aperto a chiunque (PUBLIC) e ordinati per nome (NAME). Per ottenere la lista dei comandi possibili digitate:

```
/help
```

Per la guida specifica di un comando (esempio `list`):

```
/help list
```

Per scollegarsi dal server IRC e chiudere il programma:

```
/quit
```

E' possibile ottenere la lista dei parametri del comando `irc` digitando al prompt:

```
[darshan@dada darshan]$ irc -help
```

In recenti distribuzioni GNU/Linux, questo comando non è presente perchè sostituito con client IRC grafici come `xchat` o `kvinc`, o l'insuperabile `bitchx` per la console.

10.2.40 last

Ogni accesso al vostro sistema GNU/Linux viene monitorato e memorizzato nel file `/var/log/wtmp` visualizzabile con qualsiasi editor di testi. Malgrado questo è possibile utilizzare il comando `last` appositamente studiato per l'azione. La forma per l'utilizzo è la seguente:

```
[darshan@dada darshan]$ last
```

Per visualizzare al meglio il risultato del comando è consigliabile indicare il numero di righe nel seguente modo:

```
[darshan@dada darshan]$ last -10
```

Alternativamente è possibile ridirigere l'output del comando a `more` (pagina 104) in modo di visualizzare il tutto una pagina alla volta. La forma è:

```
[darshan@dada darshan]$ last|more
```

10.2.41 latex

Chi già conosce questo "comando" sarà conscio che la sua presenza in questa lista lo riduce notevolmente di valore. Non è assolutamente quello che l'autore desidera ottenere ma è vero che spesso un utente utilizza quest'ultimo in rare occasioni con limitate opzioni e in tale senso si

desidera analizzare il comando. In fase di installazione potrete notare le dimensioni dei pacchetti relativi a \LaTeX e capire solamente da questo particolare la propria estensione.

Questo strumento è un insieme di macro per la formattazione e produzione di documenti scritti in \TeX . Se si dispone di documenti con estensione `.tex` o documenti di questo tipo digitate:

```
[darshan@dada darshan]$ latex documento.tex
```

Se l'esecuzione del comando termina con successo potrete notare nella stessa directory dove è presente `documento.tex` il nuovo file `documento.dvi` formato pronto per la stampa. Per visualizzarlo è possibile utilizzare l'applicazione `xdvi` (vedi pagina 116), per la conversione in altri formati fate riferimento ai comandi `dvips` (pagina 91) e `dvipdfm` (pagina 91).

10.2.42 `ln`

GNU/Linux permette di creare dei link ai file per una migliore gestione. Un link vi permette di dare ad un singolo file nomi multipli. I file nel sistema sono identificati da un proprio numero inode. Una directory è una lista di numeri inode con i corrispondenti nomi dei file. Esistono due tipi di link (collegamenti): `hard link` e `link simbolici`. Entrambi sono rappresentazioni di un file ma con particolari differenze. Analizziamo ora l'utilizzo e in seguito le differenze tecniche. Per creare un semplice link (`hard link`) digitiamo:

```
[darshan@dada darshan]$ ln doc filelink
```

Con il comando appena digitato troveremo nella nostra directory due file che sono la medesima cosa: `doc filelink`. Modificare `filelink` vuol dire apportare delle variazioni a `doc`, il file di riferimento. Per creare un link simbolico è necessario aggiungere l'opzione `"s"` al nostro comando:

```
[darshan@dada darshan]$ ln -s doc filelinksimbolico
```

Analizzando il contenuto della nostra directory troviamo ora, oltre a `filelink`, `filelinksimbolico`, il nuovo riferimento a `doc`. L'approccio è il medesimo. Modificando `filelinksimbolico` apportiamo delle variazioni a `doc`.

Differenze tra `hard link` e `link simbolici` Un `hard link` è un collegamento al numero inode di un file. I permessi su questo tipo di file sono gli stessi del file originale. Se decidete di cancellare un `hard link` il file originale non viene eliminato. Un `link simbolico` a differenza del caso precedentemente esaminato è un "collegamento ad un nome di file" e non ad un numero inode. Questa forma offre maggiore flessibilità operativa nell'uso comune ed è la scelta largamente utilizzata. Il `link simbolico` ha permessi differenti da quello originali ma nella manipolazione del file, gli ultimi riferimento sono fatti sui permessi del file originale.

10.2.43 `locate`

Questo comando è utile per trovare file nel sistema ma ha un comportamento diverso dal comando `find` analizzato a pagina 93. Quest'ultimo compie una ricerca nel sistema, mentre il secondo, `locate` (o `slocate` per distribuzioni recenti compatibili con Red Hat), analizza un file di testo indicizzato migliorando le prestazioni della ricerca e riducendo il carico delle risorse utilizzate. Il file indicizzato viene creato con un processo di `cron` (pagina 89) generalmente in orari notturni. Se l'utilizzo del vostro computer è quello comune e saltuario potrete saltare spesso l'esecuzione del processo di aggiornamento e in tal senso è possibile agire in modi differenti, modificando il file di esecuzione del processo o lanciando quest'ultimo manualmente. Con i dovuti diritti dell'utente `root` digitate:

```
[root@dada /root]# /etc/cron.daily/slocate.cron
```

Al termine dell'esecuzione del comando, la modalità di utilizzo del comando `locate` per qualsiasi utente del sistema è la seguente:

```
[darshan@dada darshan]# locate stdio.h
/usr/include/stdio.h
/usr/include/bits/stdio.h
/usr/include/lam/stdio.h
```

Se da un lato la ricerca con `locate` risulta molto più efficiente, l'efficacia nel trovare determinate cose è relativa alla fase di aggiornamento del file indicizzato. In tal senso il comando `find` può risultare la scelta migliore.

10.2.44 `lpc`

Tramite questo comando è possibile gestire le code di stampa. Salvo particolari configurazioni è necessario essere l'utente `root` per poter gestire il programma e le operazioni annesse sulla coda di stampa. La forma è la seguente:

```
[root@dada /root]# lpc
lpc>
```

Come si nota viene attivato il prompt del programma. Per avere l'elenco dei comandi disponibili digitate:

```
lpc> ?
Commands may be abbreviated.  Commands are:

abort  enable  disable  help      restart  status  topq    ?
clean  exit    down     quit      start    stop    up
```

Potete notare le diverse soluzioni disponibili per spostare file di stampa in cima alla coda, fermare i processi e altro ancora. Il comando che probabilmente può risultare utile nei casi reali per avere un monitor sui processi in corso è il seguente:

```
lpc> status
lp:
    queuing is enabled
    printing is enabled
    1 entry in spool area
    lp is ready and printing

lp0:
    queuing is enabled
    printing is enabled
    no entries
    no daemon present
```

Per ottenere informazioni in merito ai comandi disponibili bisogna anteporre la parola "help" al comando come segue:

```
lpc> help clean
clean      remove cruft files from a queue
```

10.2.45 lpr

Questo comando serve per la stampa su un sistema BSD compatibile come GNU/Linux. Per lanciare il processo di stampa di un file, digitate:

```
[darshan@dada darshan]$ lpr doc
```

Nell'esempio appena visto si lancia in stampa il file `doc`. Se si desidera stampare un file con una stampante specifica è necessario digitare quanto segue:

```
[darshan@dada darshan]$ lpr -Pstampante doc
```

Nel caso comune le stampanti sono collegate alle porte parallele dei computer comuni ed è possibile sostituire il nome della stampante con la porta stessa alla quale è collegata:

```
[darshan@dada darshan]$ lpr -Plpt1 doc
```

Dove `lpt1` corrisponde ad una porta parallela specifica.

10.2.46 lprm

Questo comando rientra nell'attività di stampa per rimuovere i processi di stampa accodati. Esempio di utilizzo comune per rimuovere ogni cosa:

```
[darshan@dada darshan]$ lprm
```

Se si desidera terminare un determinato processo è necessario digitare:

```
[darshan@dada darshan]$ lprm 9
```

Verrà terminato il processo di stampa numero 9 tra tutti quelli attivi. Per conoscere le code di stampa relative ad ogni stampante fate riferimento al comando `lpq`, di seguito esposto.

10.2.47 lpq

Questo comando mostra le code di stampa, ovvero i documenti messi in file per essere stampati con ordine sequenziale. Digitate:

```
[darshan@dada darshan]$ lpq
```

10.2.48 ls

Questo comando mostra il contenuto di una directory. Digitate:

```
[darshan@dada darshan]$ ls
```

Nei sistemi unix esistono file visibili e file nascosti. Per "nascondere" un file è necessario rinominarlo con un punto davanti al nome (`.file`) e per poter visualizzarlo con il comando in questione bisogna utilizzare l'opzione seguente:

```
[darshan@dada darshan]$ ls -a
.
..
.nomefile_nascosto
```

```
nomefile_visibile
```

Se si desidera visualizzare permessi, proprietari e dimensioni dei file digitate:

```
[darshan@dada darshan]$ ls -l
.
..
drwxrwxrwx 2 darshan darshan 1024 Set 18 22:32 Documenti
-rw-rw-rw- 4 darshan darshan 2056 Set 20 21:50 nomefile_visibile
```

10.2.49 mail

Semplice programma da console per spedire e ricevere messaggi di posta. Digitate:

```
[darshan@dada darshan]$ mail
```

Visualizza se c'è posta per l'utente.

```
[darshan@dada darshan]$ mail -s "compito" -u madrid
```

Spedisce un messaggio di posta all'utente madrid con il soggetto "compito".

```
[darshan@dada darshan]$ mail -f
```

Legge il contenuto della vostra casella postale.

10.2.50 man

Questo comando visualizza la pagina manuale di un comando ed è molto utile quando si vuole conoscere le opzioni relative. A titolo di esempio, visualizziamo la pagina manuale di un comando già citato precedentemente:

```
[darshan@dada darshan]$ man ls
```

Esiste una pagina manuale per lo stesso comando man:

```
[darshan@dada darshan]$ man man
```

Se non ricordate il nome del comando che vi serve a compiere una determinata operazione potete avvalervi dell'opzione "k" nel seguente modo:

```
[darshan@dada darshan]$ man -k parola
```

Verranno mostrate le pagine manuale che contengono la parola indicata.

10.2.51 mc

Midnight Commander, una interfaccia per navigare tra le directory e i file del sistema. È un programma molto simile al famoso Norton Commander. Per eseguirlo:

```
[darshan@dada darshan]$ mc
```

Nelle recenti distribuzioni di GNU/Linux, questo programma è stato sviluppato con le librerie Gtk⁵ ed è la base sulla quale si muove l'interfaccia grafica Gmc, il file manager di Gnome.

10.2.52 mesg

Gli utenti di un sistema possono interagire tra di loro attraverso diversi programmi via terminale. Potete disabilitare le richieste esterne digitando:

```
[darshan@dada darshan]$ mesg n
```

Per riattivare in modalità "recettiva":

```
[darshan@dada darshan]$ mesg y
```

Il comando indicato può tornare particolarmente utile durante richieste di chat con il programma talk di seguito descritto.

10.2.53 mkdir

Questo comando vi serve per creare una directory. La sintassi tipica è:

```
[darshan@dada darshan]$ mkdir esempio
```

Il comando è particolarmente sensibile ai permessi di sistema. E' chiaro che non vi sarà concesso di creare una directory dove non avete i diritti per farlo.

10.2.54 more

Questo comando visualizza il contenuto di un file una schermata alla volta.

```
[darshan@dada darshan]$ more file
```

Nonostante questo semplice esempio, il comando viene utilizzato in associazione con altri. L'esempio che segue legge l'output del comando ls e lo redirige al comando more per visualizzare il risultato una pagina per volta:

```
[darshan@dada darshan]$ ls -la|more
```

Il simbolo "|" (pipe) viene utilizzato per passare il risultato di un comando al successivo.

10.2.55 mount

Questo comando serve a montare un file system. Nell'esempio che segue viene mostrato come montare un cd-rom con file system iso9660 alla directory /mnt/cdrom:

```
[root@dada /root]# mount /dev/cdrom -t iso9660 /mnt/cdrom
```

Le distribuzioni maggiormente diffuse di GNU/Linux inseriscono i dati utilizzati nel file /etc/fstab per facilitare ed automatizzare le operazioni comuni. Se nel file appena citato è presente una riga di configurazione simile:

⁵Gimp Tool Kit - <http://www.gtk.org>


```
/dev/cdrom /mnt/cdrom iso9660 owner,noauto,ro 0 0
```

basterà utilizzare il comando `mount` con una serie ridotta di opzioni:

```
[root@dada /root]# mount /dev/cdrom
```

Dopo aver digitato il comando potrete accedere al contenuto del vostro cd-rom. Si utilizza il comando `mount` per condividere aree file condivise in rete tramite il sistema NFS. La formulazione tipica può essere la seguente:

```
[root@dada /root]# mount monika:/mnt/nfs -t nfs /mnt/nfs
```

Dove si indica di montare la partizione `/mnt/nfs` presente sul server `monika` e disporla nel file system locale come `/mnt/nfs`. Chiaramente è necessaria la specifica configurazione lato server (pagina 18).

10.2.56 mv

Questo comando può avere una duplice funzionalità: muovere un file e rinominarlo. Nell'esempio che segue viene rinominato un file con un nuovo nome:

```
[darshan@dada darshan]$ mv vecchionome nuovonome
```

Per muovere un file verso un'altra directory, digitate:

```
[darshan@dada darshan]$ mv nuovonome ~/documenti/
```

In questo ultimo caso viene utilizzato il simbolo `"~"` che indica la directory home dell'utente. Il file viene spostato nella directory `/home/darshan/documenti/`.

10.2.57 nice

Serve a cambiare la modalità di esecuzione di un programma. Se viene lanciato senza opzioni viene presentata la priorità corrente:

```
[darshan@dada darshan]$ nice  
0
```

Nel caso in esame abbiamo ricevuto un valore pari a `"0"`. I valori di priorità vanno da `-20` (la più alta) a `19` (la più bassa). Se volessimo abbassare la priorità di esecuzione di un programma in modo che non disturbi le attività del sistema digitiamo:

```
[darshan@dada darshan]$ nice -n 5 nomeprogramma
```

L'opzione `"-n"` mi permette di indicare un valore scelto nella scala esposta precedentemente.

10.2.58 nl

Se si desidera numerare ogni singola riga di un documento di testo questo è il comando che serve. Esistono diverse opzioni interessanti:

- `-v n` definisce il numero iniziale per la numerazione
- `-i n` stabilisce il valore di incremento della numerazione

- `-p` non ricomincia la numerazione per ogni nuova pagina

Per numerare le righe di un documento partendo dal valore 1 e incrementando di una singola unità per volta dovremo digitare quanto segue:

```
[darshan@dada darshan]$ nl -v 1 -l 1 doc.txt > doc2.txt
```

Il documento con la numerazione è stato ridiretto al file `doc2.txt`.

10.2.59 nohup

Serve a lanciare un programma in particolari modalità. L'utilizzo di questo comando è particolarmente utile quando si deve eseguire un programma di calcolo e non lo si interrompere nel momento in cui vi disconnettete dal sistema (es. in una connessione telnet).

```
[darshan@dada darshan]$ nohup nomeprogramma
nohup: appending output to 'nohup.out'
```

Il programma continuerà a girare in background ed eventuali messaggi saranno riportati nel file `nohup.out`, creato nella directory corrente.

10.2.60 passwd

Questo comando serve per cambiare la propria password (parola d'ordine) necessaria per l'autenticazione nel sistema. Esistono due differenti livelli di utilizzo del comando, un livello ampliato e completo per l'utente `root` (l'amministratore di sistema) e un livello ristretto ma essenziale per il resto degli utenti nel sistema. Mentre un utente può utilizzare `passwd` per gestire la propria password, l'utente `root` può utilizzare il comando per gestire oltre alla propria password quelle degli utenti nel server amministrato.

```
[darshan@dada darshan]$ passwd
Changing password for darshan
(current) UNIX password: *****
New UNIX password: *****
Retype new UNIX password: *****
```

Analizziamo ogni singolo passo. In un primo momento viene chiesta la password corrente, la parola d'ordine che attualmente ci permette di collegarci al sistema. Se quest'ultima corrisponde viene chiesto di digitare una nuova password con successiva ri-digitazione per verifica. Il comando è intelligente e cerca di colmare l'inesperienza di un nuovo utente o scelte incaute che possono compromettere la sicurezza del sistema. Le nuove password digitate vengono controllate e se non rispettano determinati parametri si riceve un messaggio di avviso che invita a seguire determinate norme. Una combinazione di numeri e lettere maiuscole/minuscole può essere una valida soluzione. Esempio:

```
A3r0pLan0 <-> aeroplano
MaNuCha0 <-> manuchao
F3deRic0 <-> federico
```

10.2.61 patch

Il comando serve per applicare i file `patch` (creato con il comando `patch`) creato con le differenze tra un file e un file più recente. La modalità di utilizzo è la seguente:

```
[darshan@dada darshan]$ patch -p0 < patch
```

Per applicare patch binarie è necessaria la seguente opzione:

```
[darshan@dada darshan]$ patch -binary -p0 < patch
```

10.2.62 pico

Il seguente comando lancia un editor di testo semplice ed intuitivo, presente nel sistema se viene installato il pacchetto software pine. Per utilizzarlo:

```
[darshan@dada darshan]$ pico
```

Per poter salvare il file editato premere contemporaneamente i tasti CTRL e X, oppure CTRL e O. Un'alternativa GPL di questo editor è nano che aggiunge nuove funzionalità.

10.2.63 pr

Questo comando serve per preparare un file per la stampa. La forma di utilizzo è la seguente:

```
[darshan@dada darshan]$ pr doc.txt
```

Il file `doc.txt` verrà formattato con una lunghezza di 66 righe (numero per la lunghezza configurabile a piacimento) e i dati relativi al file (data, ora, nome, numero di pagina).

10.2.64 ps

Riporta lo stato dei processi attivi nel sistema e si utilizza nel seguente modo:

```
[darshan@dada darshan]$ ps
PID TTY          TIME CMD
648 pts/2      00:00:00 bash
649 pts/2      00:00:00 ps
```

Nell'uso comune è frequente utilizzare il comando con le seguenti opzioni:

```
[darshan@dada darshan]$ ps afx
```

Dove la lettera "a" mostra i processi degli altri utenti collegati al sistema, la lettera "f" la sequenza dei processi attivi organizzata in una struttura ad albero e la lettera "x" che mostra i processi senza terminali di controllo.

10.2.65 pstree

Questo comando ha un funzionamento molto simile a `ps` ma rappresenta i processi attivi in un'alberatura particolare. Digitate:

```
[darshan@dada darshan]$ pstree -n
init--kflushd
  |-kupdate
  |-kpiod
  |-kswapd
  |-mdrecoveryd
  |-syslogd
  |-klogd
```

```

|-xfs
|-login---bash
|-5*[mingetty]
|-gdm-+-X
|   '-gdm---Default---gnome-session
|-esd
|-gnome-smproxy
|-sawfish
|-xscreensaver
|-gmc
|-panel
|-gnome-terminal-+-gnome-pty-helper
|   '-bash---pstree
|-gnome-name-serv
|-tasklist_applet
|-rpciod
`-gv---gs

```

10.2.66 pwd

Questo comando vi indica il percorso assoluto in cui vi trovate:

```

[darshan@dada darshan]$ pwd
/home/darshan/

```

10.2.67 reboot

Questo comando è necessario per riavviare il sistema. Una volta lanciato compierà diverse operazioni volte a interrompere i servizi attivi, disabilitare i dispositivi che compongono il file system e riavviare il computer.

```

[root@dada /root]# reboot

```

10.2.68 rm

Questo comando rimuove un file in maniera permanente. Nell'esempio che segue viene mostrata la modalità interattiva ottenuta con l'opzione "i":

```

[darshan@dada darshan]$ rm -i file

```

La modalità "forzata" agisce in maniera inversa. Elimina il file senza chiedere preventivamente:

```

[darshan@dada darshan]$ rm -f file

```

Se si desidera cancellare una directory e i file in essa contenuti è necessario utilizzare l'opzione "r" (recursive) nel seguente modo:

```

[darshan@dada darshan]$ rm -fr documenti/

```

10.2.69 rpm

Questo comando è lo strumento di gestione dei pacchetti software caratteristico della distribuzione Linux Red Hat. Questo comando gestisce i pacchetti software organizzati secondo le specifiche

del formato che vengono indicati comunemente con l'estensione `.rpm`. Analizziamo le varie opzioni:

```
[root@dada /root]# rpm -i nomefile.rpm
```

Il comando installa un pacchetto software nel sistema.

```
[root@dada /root]# rpm -U nomefile.rpm
```

Il comando aggiorna un pacchetto software nel sistema.

```
[root@dada /root]# rpm -e nomefile
```

Il comando rimuove il software dal sistema.

```
[root@dada /root]# rpm -qf esempio.txt
```

Il comando esamina quale pacchetto software conteneva il file `esempio.txt`.

```
[root@dada /root]# rpm -qa
```

Il comando elenca i pacchetti software installati sul proprio sistema. Nelle fasi di installazione e aggiornamento possono essere utilizzate ulteriori opzioni che informano sullo stato di esecuzione del processo e mostrano una barra di progresso.

```
[root@dada /root]# rpm -ivh nomefile.rpm
nomefile          #####
```

```
[root@dada /root]# rpm -Uvh nomefile.rpm
nomefile          #####
```

In casi particolari può capitare che il pacchetto che si desidera installare subisca particolari vincoli di dipendenza verso altri software. Se vogliamo sopprimere questi ultimi digitiamo:

```
[root@dada /root]# rpm -i nomefile.rpm --nodeps
```

Una scelta particolarmente orientata ad un utente esperto può anche essere la seguente:

```
[root@dada /root]# rpm -i nomefile.rpm --force
```

Il comando digitato forza l'installazione senza controlli preventivi dei pacchetti esistenti o sui vincoli di dipendenza. Un'ulteriore necessità può essere quella di conoscere ogni possibile informazione sul pacchetto software che ci si approssima ad installare nel proprio sistema. Per questo è necessario digitare quando segue:

```
[root@dada /root]# rpm -qpil geekcode-1.7-2.i386.rpm
Name       : geekcode           Relocations: /usr
Version    : 1.7                 Vendor: Chris Gushue
Release    : 2                   Build Date: mar 22 feb 2000 15:59:14
Install date: (not installed)   Build Host: DeepThought.prefect.net
Group      : Amusements/Games   Source RPM: geekcode-1.7-2.src.rpm
Size       : 159494             License: GPL
Packager   : Chris Ausbrooks <weed@bucket.pp.ualr.edu>
URL        : http://seymour.lazygenes.net/geekcode.html
Summary    : Generates your geek code.
```

```
Description :
This program will generate a geek code block for you by simply
choosing which codes suit you from the screen.
/usr/bin/geekcode
/usr/doc/geekcode-1.7
/usr/doc/geekcode-1.7/CHANGES
/usr/doc/geekcode-1.7/COPYING
/usr/doc/geekcode-1.7/INSTALL
/usr/doc/geekcode-1.7/README
/usr/doc/geekcode-1.7/geekcode.txt
```

10.2.70 rpm2cpio

Se desiderate convertire un pacchetto software in formato .rpm in un archivio comune cpio questo è quello che vi serve. La forma tipica è la seguente:

```
[root@dada /root]# rpm2cpio nomefile.rpm > archivio.cpio
```

Per estrarre successivamente l'archivio creato fate riferimento al comando `cpio` (pagina 88).

10.2.71 set

Questo comando mostra le variabili di ambiente della vostra sessione:

```
[darshan@dada darshan]$ set
```

Il comando `set` ha la medesima funzionalità del comando `env`.

10.2.72 shutdown

Questo comando prepara il sistema per essere spento fisicamente e permette opzioni anche per il riavvio. Presenta analogie con il comando `halt` (pagina 97) ma porta ulteriori opzioni relative al riavvio del computer. La forma tipica è la seguente:

```
[root@dada /root]# shutdown -r now
```

L'opzione “-r” sottointende reboot, riavvio; era possibile utilizzare anche l'opzione “-h” che sottointende a halt, ferma. E' cosa gentile preavvisare gli eventuali utenti collegati al sistema e dare loro il tempo per concludere le operazioni in corso. Digitando:

```
[root@dada /root]# shutdown -r +1 "Riavvio della macchina"
```

Lanciamo l'interruzione del sistema con 1 minuto di attesa e un messaggio di avviso per gli utenti collegati.

10.2.73 slocate

Le modalità di funzionamento di questo comando sono simili a quelle di `locate` (pagina 100). Differenza che contraddistingue i due comandi è una maggiore sicurezza per `slocate` nella rappresentazione della ricerca. Quest'ultima infatti mostrerà unicamente i file accessibili (in termini di diritti) dagli utenti del sistema. La modalità di utilizzo è la seguente:

```
[darshan@dada darshan]$ slocate nomefile
```

10.2.74 split

Questo comando è utile quando si desidera dividere un file con dimensioni elevate in varie parti. Digitate:

```
[darshan@dada darshan]$ split -b 1m moni.tif
```

In questo caso il file `moni.tif` verrà diviso in vari file della dimensione di 1 Mbyte i quali saranno nominati con una sequenza di lettere per determinare l'ordine. Per riottenere il nostro file è necessario digitare:

```
[darshan@dada darshan]$ cat xaa xab xac > moni.tif
```

Il file originale sarà ripristinato.

10.2.75 ssh

Utilizzare `telnet` (pagina 113) per connessioni remote può compromettere la propria sicurezza trasferendo dati *in chiaro*. A questo viene incontro il client `ssh` che instaura una *connessione sicura* tra due macchine in rete crittando la trasmissione dei dati. La forma tipica di utilizzo è la seguente:

```
[darshan@dada darshan]$ ssh -l nomeutente nomeserver
```

L'utilizzo del client prevede l'esistenza del demone `sshd` (lato server) attivo e configurato per accettare le connessioni di questo tipo.

10.2.76 su

Questo comando serve per cambiare la propria modalità utente in una diversa o quella relativa all'utente `root`. Digitate:

```
[darshan@dada darshan]$ su -  
Password: *****  
[root@dada /root]#
```

L'opzione `--` attiva la configurazione della shell dell'utente indicato. In ogni caso è richiesta la password tranne per il passaggio da utente `root` a utente normale per ovvie ragioni.

10.2.77 talk

Questo comando vi permette di comunicare con altri utenti presenti sul vostro server o collegati a server remoti nella vostra rete locale o su Internet. Se desiderate parlare con l'utente `madrid` è necessario digitare quanto segue:

```
[darshan@dada darshan]$ talk madrid@localhost
```

Nell'esempio appena visto si sottintende che l'utente `madrid` sia presente e collegato al nostro server. Per conoscere queste informazioni fate riferimento al comando `ps` che vi permette inoltre di conoscere l'interfaccia utilizzata dall'utente nel sistema ed utilizzarla come alternativa al suo nome:

```
[darshan@dada darshan]$ talk pts/2
```

Nell'esempio che segue viene illustrato l'utilizzo del comando utilizzando la forma `utente@dominio` che in questo caso è la macchina locale:

```
[darshan@dada darshan]$ talk madrid@localhost
```

Il nome dominio è `localhost`, genericamente inteso come il server locale sul quale vi siete collegati. Se disponete di un server collegato ad Internet e desiderate "chiamare" un vostro amico collegato su un altro server in rete è necessario specificare un dominio con la denominazione tipica della rete:

```
[darshan@dada darshan]$ talk madrid@erlug.linux.it
```

Se l'utente chiamato accetta la chiamata potrete dialogare direttamente con lui e scambiarvi dati in modalità testo.

10.2.78 tar

Questo comando serve per archiviare un insieme di file in un solo file e viene utilizzato nel seguente modo:

```
[darshan@dada darshan]$ tar -c archivio.tar Directory/
```

Nell'uso quotidiano il file archivio viene spesso compresso utilizzando sempre lo stesso carattere ma con opzioni aggiuntive:

```
[darshan@dada darshan]$ tar -zcvf archivio.tar.gz Directory/
```

Nell'esempio appena visto si utilizza l'opzione "z" per la compressione dei dati tramite gzip, l'opzione "c" che indica la creazione di un archivio, l'opzione "v" (verbose) per visualizzare i messaggi di esecuzione e l'opzione "f" per definire il nome del file archivio. L'estensione ".tar" è puramente indicativa. Se si volesse creare un archivio multivolume utilizzando dei floppy-disk HD, digitate:

```
[darshan@dada darshan]$ tar -cvfM /dev/fd0H1440 archivio/
```

L'opzione "M" creerà vari volumi a seconda della situazione. Per riunire i vari multivolumi è necessario utilizzare il comando `cat` nel seguente modo:

```
[darshan@dada darshan]$ cat 1.tar 2.tar > archivio.tar.gz
```

Per scompattare il file creato è necessario digitare:

```
[darshan@dada darshan]$ tar -zxvf archivio.tar.gz
```

Oppure con

```
[darshan@dada darshan]$ tar -xvf archivio.tar
```

Se l'archivio non è compresso gzip.

10.2.79 tcpdchk

Questo comando controlla la configurazione di `TCP_WRAPPER` e vi segnala eventuali configurazioni errate. Digitate:

```
[root@dada /root]# tcpdchk
```

Se vengono riscontrati errori è necessario modificare i file `/etc/hosts.deny` e `/etc/hosts.allow`. Per maggiori informazioni cercate l'esempio a pagina 145.

10.2.80 telnet

Se c'è un programma "storico" questo è senza dubbio `telnet`, un programma che permette di gestire connessioni remote ad un altro server per amministrarlo ed eseguire programmi. Per poterlo utilizzare è chiaramente necessario avere un accesso sul server interessato:

```
[darshan@dada darshan]$ telnet
telnet>
```

Per conoscere l'insieme delle opzioni presenti digitare "help":

```
telnet> help
```

Per aprire una connessione ad un server:

```
telnet> open nomeserver
```

Per chiudere una connessione:

```
telnet> close
```

Utilizzando una forma compatta:

```
[darshan@dada darshan]$ telnet -l utente nomeserver
```

Per effettuare una *connessione sicura*, crittata con particolari algoritmi, si consiglia l'utilizzo di `ssh` (pagina 111).

10.2.81 top

L'utilizzo di questa applicazione è utile se si intende monitorare il carico del processore e visualizzare i processi attivi. Digitate il comando nel seguente modo:

```
[darshan@dada darshan]$ top
```

e successivamente apparirà quanto segue:

```
6:32pm up 1:16, 2 users, load average: 0.15, 0.11, 0.09
46 processes: 44 sleeping, 2 running, 0 zombie, 0 stopped
CPU states: 6.4% user, 3.6% system, 0.0% nice, 89.8% idle
Mem: 95504K av, 63940K used, 31564K free, 47496K shrd, 4964K buff
Swap: 60476K av, 0K used, 60476K free 34156K cached
```

```
PID USER PRI NI SIZE RSS SHARE STAT LIB %CPU %MEM TIME COMMAND
573 root 15 0 8784 8784 1656 R 0 5.2 9.1 3:54 X
```

Sintassi	Azione
:q!	uscire senza salvare
:w	salvare il contenuto
:wq!	salvare il contenuto e uscire
:help	consultare la guida programma
:+n	spostarsi alla riga n (numero)
:s/old/new/g	sostituzione di una parola (da old a new)
:r file	inserire il file nel testo aperto
:r! comando	eseguire un comando di sistema
:syntax on	syntax highlighting attivo
:set syntax=html	configura il syntax highlighting per html

Tabella 10.1: Comandi per l'editor vi

```

641 root      9   0  1040 1040   836 R      0  2.7  1.0  0:01 top
631 root      4   0  3392 3392  2656 S      0  1.7  3.5  0:01 gnome-terminal
577 root      1   0  1324 1324  1008 S      0  0.3  1.3  0:02 fvwm2
  4 root      0   0     0   0     0 SW      0  0.0  0.0  0:00 kpiod
...

```

Per uscire premete contemporaneamente i tasti CTRL e C.

10.2.82 uname

Questo comando è utile se si desidera avere un quadro generale sulle caratteristiche del computer e conoscere alcune importanti informazioni di sistema. La forma rapida e comune di utilizzo è la seguente:

```
[darshan@dada darshan]$ uname -a
Linux dada 2.2.14-5.0 #1 Tue Mar 7 21:07:54 EDT 2000 i686 unknown
```

Dove l'opzione "-a" indica di rappresentare tutte le informazioni disponibili.

10.2.83 umount

Questo comando compie una operazione inversa al comando mount (pagina 104), ovvero "smon-ta" il file system aggiunto. Riprendendo l'esempio del cd-rom:

```
[root@dada /root]# umount /mnt/cdrom
```

Per svolgere l'operazione è necessario essere in una directory differente da quelle presenti nel file system che si desidera smontare.

10.2.84 vi

Questo comando lancia un editor di testo molto diffuso e largamente diffuso nelle varie famiglie di Unix. Per creare un nuovo file, digitate:

```
[darshan@dada darshan]$ vi pippo.txt
```

E' sicuramente un editor minimale nella propria interfaccia ma allo stesso tempo molto potente. I comandi presentati nella tabella 10.1 riassumono i comandi essenziali.

10.2.85 vlock

Questo comando blocca la console testo. Serve per proteggere ogni “intrusione locale” al vostro sistema. Per attivarlo è necessario essere in modalità console testo e digitare:

```
[darshan@dada darshan]$ vlock
This TTY is now locked.
Use Atl-function keys to switch to other virtual consoles.
Please enter the password to unlock.
darshan's Password:
```

Per togliere la protezione è necessario digitare la propria password. Se l'ambiente è quello grafico del Server X un comando analogo è `xlock` (pagina 116).

10.2.86 zip

Questo comando serve a comprimere un file. Per comprimere un file è necessario digitare:

```
[darshan@dada darshan]$ zip file.zip racconto.txt
```

Il file `racconto.txt` verrà compresso e verrà rinominato come `file.zip`. Per scompattare il file presente è necessario utilizzare il comando `unzip`:

```
[darshan@dada darshan]$ unzip file.zip
```

10.2.87 wc

Questo comando serve per conoscere dati relativi ad un file di testo:

```
[darshan@dada darshan]$ wc documento.txt
```

Verrà presentata una serie di informazioni, selezionabili con le seguenti opzioni:

- c per ottenere solo il numero dei caratteri
- w per ottenere solo il numero delle parole
- l per ottenere solamente il numero delle righe presenti

Digitando:

```
[darshan@dada darshan]$ wc -w documento.txt
560 documento.txt
```

otteniamo il numero di parole presenti nel file.

10.2.88 whatis

Questo comando serve a ricercare la descrizione per ogni comando indicizzando la pagine manuale. Se desiderate avere la descrizione del comando `ls` digitate:

```
[darshan@dada darshan]$ whatis ls
```

Verrà presentata una breve descrizione. Se dovesse apparire un messaggio del tipo:

```
ls: nothing appropriate
```

E' necessario eseguire da amministratore il comando `makewhatis` per creare l'indice delle descrizioni. Un comando analogo a `whatis` è `apropos` (pagina 84).

10.2.89 write

Questo comando permette di mandare messaggi ad un determinato utente collegato sul server locale. A differenza del comando `talk`, la comunicazione in questo caso è unidirezionale. Per utilizzarlo è necessario digitare il comando e il nome dell'utente di riferimento (nell'esempio seguente l'utente `madrid`), premere il tasto `INVIO` e scrivere il proprio messaggio da riportare:

```
[darshan@dada darshan]$ write madrid
Questo è un messaggio di esempio
```

Se l'utente è collegato riceverà quanto segue:

```
[madrid@dada madrid]$
Message from darshan@dada on pts/3 at 03:46 ...
Questo è un messaggio di esempio
```

10.2.90 who

Attraverso questo comando è possibile avere informazioni sugli utenti connessi al sistema. La sintassi di utilizzo è la seguente:

```
[darshan@dada darshan]$ who
root  tty1  Nov 8 15:02
root  tty2  Nov 8 16:00
```

10.2.91 xdvi

Applicazione per l'ambiente X Window. Permette di visualizzare un file con estensione `.dvi`. Digitate:

```
[darshan@dada darshan]$ xdvi documento.dvi
```

Per maggiori informazioni relative al formato file andate a pagina 99.

10.2.92 xlock

Questo comando vi permette di bloccare lo schermo e le vostre attività da intrusioni altrui. Il concetto è simile a quello visto per `vlock` (pagina 115) in modalità console testo. Per attivare questa protezione è necessario digitare:

```
[darshan@dada darshan]$ xlock
```

10.2.93 xvidtune

Applicazione per l'ambiente X Window. Permette di configurare le frequenze video del monitor e operare sulla centratura:

```
[root@dada /root]# xvidtune
```

10.2.94 xterm

Applicazione per l'ambiente X Window. Rappresenta uno dei terminali maggiormente utilizzati e supporta un'ampia serie di emulazioni terminale.

```
[darshan@dada darshan]$ xterm
```

Se desideriamo configurare maggiormente questa applicazione possiamo aggiungere delle opzioni sulla riga di comando:

```
[darshan@dada darshan]$ xterm -rightbar -bg white -fg black
```

In questo modo lanciamo `xterm` con la barra di scorrimento sulla destra e con caratteri neri su sfondo bianco. Premendo contemporaneamente il tasto sinistro (o il tasto destro) del mouse e il pulsante CTRL compariranno dei menu descrittivi per rapide configurazioni del programma.

10.2.95 xsetpointer

Applicazione per l'ambiente X Window. Questo comando permette di mostrare e configurare strumenti di puntamento per l'ambiente grafico. I puntatori devono essere precedentemente configurati nel file di configurazione `XF86Config`:

```
[darshan@dada darshan]$ xsetpointer -l
"keyboard"  [XKeyboard]
"pointer"   [Xpointer]
"wacom"     [wacom]
```

Nell'esempio appena visto l'opzione `"-l"` ci elenca i puntatori configurati. Se decidiamo di cambiare il puntatore dal mouse comune alla tavoletta grafica presente digitiamo:

```
[darshan@dada darshan]$ xsetpointer wacom
```

10.2.96 xsetroot

Applicazione per l'ambiente X Window. Permette di configurare lo sfondo dell'ambiente utilizzando immagini (in formato `.bmp`) e colori solidi:

```
[darshan@dada darshan]$ xsetroot -solid blue
```

Il comando appena digitato imposta il colore blu come colore di sfondo.

F.A.Q.

Domanda: "Molti dei comandi presentati funzionano in modalità console? Esistono anche applicazioni con interfaccia grafica?" La direzione dei sistemi operativi, GNU/Linux compreso, punta ad ambienti desktop grafici. Le motivazioni di questa direzione sono da trovare in una maggiore semplicità ed intuitività nell'utilizzo di una soluzione informatica per i propri personali obiettivi. Nonostante gli aspetti di interattività citati, conoscere la shell e comandi in modalità testo rende facile lo svolgimento di comandi sequenziali nella logica del "pipe". La shell, seppur ostica per un utente alle prime armi, può offrire una flessibilità molto superiore a quella che offre un'interfaccia desktop.

Domanda: “Ho letto le informazioni relative a GnuPG a pagina 95. Quanto è importante proteggere i propri dati e i propri messaggi in rete?” La propria privacy è un diritto innegabile e la possibilità che altrui persone possano captare illecitamente le vostre comunicazioni via email è un caso remoto ma non impossibile. Misurate il vostro livello personale di paranoia e giudicate voi stessi quanto possa esservi utile proteggervi grazie alla crittografia.

Capitolo 11

Post installazione

Se utilizzate GNU/Linux sul vostro computer personale all'avvio della macchina potrete notare il seguente prompt

lilo:

E' il **boot loader**, il *meccanismo* che compie diverse operazioni ma essenzialmente carica i sistemi operativi scelti e presenti nel vostro computer. Per avere visione delle possibilità offerte si consiglia di premere il tasto tabulatore. Nei casi comuni GNU/Linux viene lanciato in automatico dopo 5 secondi. Durante il caricamento del sistema operativo è possibile visualizzare le estensioni, il controllo delle unità-disco e il caricamento dei demoni di sistema fino al prompt di autenticazione:

```
Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.14-5.0 on a i686
login: darshan
password: *****
```

Ad autenticazione avvenuta sarete dentro al sistema e vi comparirà il prompt dei comandi:

```
[darshan@dada darshan]$
```

Il caso appena esposto è quello comune anche se ultimamente le distribuzioni più recenti cercano di portare sin dalla fase di installazione verso la modalità visuale del Server X Window per una ricerca di appariscenza ed intuitività. Malgrado questa modalità il concetto non cambia. Comparirà una interfaccia grafica per l'autenticazione richiedendo nome utente e password relativa.

L'utente *darshan* è una indicazione personale. Recenti distribuzioni GNU/Linux vi permettono di creare utenti di sistema in fase di installazione ma il caso comune e classico è quello di entrare nel sistema per la prima volta come utente *root*, l'amministratore, e aggiungere ulteriori utenti tramite i comandi *adduser* (pagina 84) e *passwd* (pagina 106). Dopo aver aggiunto i propri utenti è bene entrare nel sistema utilizzando quest'ultimi; limitate l'accesso come utente *root* solo ai casi necessari.

11.1 Gestione dei file

In questa sottosezione applicheremo i comandi precedentemente visti. Cerchiamo di capire innanzitutto "dove ci troviamo" all'interno del sistema grazie al comando *pwd*:

```
[darshan@dada darshan]$ pwd
/home/darshan/
```

Ora creiamo un file di base sul quale compieremo varie operazioni successivamente:

```
[darshan@dada darshan]$ touch nuovofile
```

Visualizziamo il contenuto della directory:

```
[darshan@dada darshan]$ ls -la
-rw-rw-r- 1 darshan darshan 0 Nov 8 23:43 nuovofile
```

Il file può essere letto e modificato dall'utente `darshan` e dalle persone appartenenti al gruppo `darshan`. Le altre persone nel sistema possono solamente leggerlo. Allo stato attuale il file ha dimensione di 0 Kbyte. Decidiamo di editare il presente file utilizzando un editor.

```
[darshan@dada darshan]$ pico nuovofile
```

Verrà lanciato il programma `pico` che vi permetterà di modificare ed eventualmente salvare il file indicato premendo congiuntamente i tasti `CTRL` e `X`. Se l'editor `pico` non fosse installato si può utilizzare l'editor `vi`. Esistono numerosi altri editor ma non verranno citati in questo manuale per non creare confusione. Per uno scopo puramente dimostrativo decidiamo di salvare nel nostro file di testo chiamato "nuovofile" le variabili di ambiente del sistema:

```
[darshan@dada darshan]$ env >> nuovofile
```

In questo caso utilizziamo il comando `env` e accodiamo il risultato al file `nuovofile` tramite i simboli `">>"`. Se avessimo utilizzato un singolo `">"` avremmo cancellato il contenuto del file memorizzando unicamente le variabili di ambiente. Vogliamo ora consultare il seguente file in modo comodo, una pagina per volta:

```
[darshan@dada darshan]$ cat nuovofile|more
```

Per uscire dalla rappresentazione del file digitate il tasto `"Q"`. A questo punto vogliamo compiere una serie di operazioni per mettere "al sicuro" il nostro file. Creiamo una directory:

```
[darshan@dada darshan]$ mkdir Documenti
```

Copiamo il file con un nome differente e lo comprimiamo in `bz2`:

```
[darshan@dada darshan]$ cp nuovofile filenuovo ; bzip2 filenuovo
```

Il simbolo `","` (punto e virgola) permette di gestire una serie di comandi in sequenza. Possiamo utilizzare anche il pipe, indicato da una barretta verticale `"|"` per passare il risultato di un comando ad un altro. Esempio:

```
[darshan@dada darshan]$ ps -afx|grep parola
```

Il comando in prima istanza mostra tutti i processi attivi sul computer e passa il risultato al comando `grep` che lo elabora e stampa a video la riga che contiene la "parola" richiesta. Muoviamo il nostro file di testo compresso `filenuovo.bz2` nella directory creata `Documenti`:

```
[darshan@dada darshan]$ mv filenuovo.bz2 Documenti/
```


A questo punto decidiamo di proteggere la directory in questione cambiandole i permessi:

```
[darshan@dada darshan]$ chmod 700 Documenti
```

Visualizziamo i cambiamenti fatti:

```
[darshan@dada darshan]$ ls -la
drwx--- 4 darshan darshan 2048 Nov 9 23:20 Documenti
```

Nessun altro al di fuori dell'utente darshan potrà accedere alla directory Documenti. Decidiamo ora di stampare il file di testo nuovofile sulla stampante locale:

```
[darshan@dada darshan]$ lpr nuovofile
```

Il processo di stampa verrà mandato in background in coda di stampa. Per visualizzare quest'ultima ed avere un'idea dei tempi di attesa, digitiamo:

```
[darshan@dada darshan]$ lpq
```

Il risultato del comando è una sequenza di processi di stampa ordinati da un codice sequenziale. Siccome siamo molto occupati e non possiamo aspettare, cancellando il nostro codice utilizzandolo con il comando che segue:

```
[darshan@dada darshan]$ lprm 34
```

Verrà cancellato il processo di stampa numero 34. Resta chiaro che è possibile cancellare i processi di stampa relativi alla nostra utenza.

Script di Shell

Decidiamo di automatizzare una serie di comandi. A tale scopo è utile creare uno script di shell, un semplice file di testo che richiama in sequenza vari comandi di sistema. In prima istanza lanciate un semplice editor per creare il file `esempio.sh`:

```
[darshan@dada darshan]$ pico esempio.sh
```

Verrà aperto l'editor pico e a questo punto digitate la seguente sequenza di comandi:

```
#!/bin/sh
echo "Come ti chiami?"
#stampa a video della domanda
read nome
#legge il nome digitato
echo "Benvenuto $nome!"
#stampa a video la frase utilizzando il nome inserito
```

La riga iniziale `#!/bin/sh` è necessaria per richiamare la shell dei comandi. Le righe precedenti da `"#"` sono dei commenti. Se l'editor è pico, premendo i tasti CTRL e O salverete il file, CTRL e X per uscire. A questo punto cambiate i permessi al file:

```
[darshan@dada darshan]$ chmod 700 esempio.sh
```

Il nostro script di shell è pronto. Per poterlo utilizzare potete spostarlo in una delle directory del `PATH` o lanciarlo digitando quanto segue:

```
[darshan@dada darshan]$ ./esempio.sh
```

Abbiamo antecedito il nome del comando con i caratteri `./` (punto, slash). Lo script verrà eseguito pur non essendo nel `PATH` dei comandi.

Utilizzare cd-rom e floppy

Come si è già potuto vedere nei precedenti capitoli, esistono diversi file system che spesso variano da un sistema operativo all'altro. Nonostante queste differenze esistono degli standard ISO come il file system dei cd-rom (iso 9660), comune per tutte le piattaforme. Se disponete di una moderna distribuzione GNU/Linux con il desktop manager Gnome potrete accedere all'unità floppy con un singolo clic sull'icona relativa. Il "montaggio" dell'unità cd ha due differenti comportamenti a seconda che sia cd-audio o cd-rom ma in entrambe i casi tutto quanto avviene in maniera istintiva ed automatica. Se non potete utilizzare un ambiente grafico è necessario utilizzare il comando `mount` visto nel capitolo precedente e ripreso nell'esempio successivo:

```
[root@dada /root]$ mount /dev/cdrom
mount: can't find /dev/hdc in /etc/fstab or /etc/mtab
```

In questo caso è stato utilizzato il comando `mount` senza aver alcun riferimento nel file `/etc/fstab` o `/etc/mtab`.

```
[root@dada /root]# mount /dev/cdrom -t iso9660 /mnt/cdrom
```

La stessa cosa deve essere fatta per gestire i floppy ma a differenza del cd-rom è necessario sapere che tipo di file system è presente nell'unità in questione. Il file system di base in ambiente GNU/Linux è `ext2` ma è facile trovare dischetti formattati con file system di tipo `msdos` o `vfat` per ambiente Microsoft Windows. A seconda dei casi comportatevi come segue:

File system `ext2`:

```
[root@dada /root]# mount /dev/fd0 -t ext2 /mnt/floppy
```

File system `ms-dos`

```
[root@dada /root]# mount /dev/fd0 -t msdos /mnt/floppy
```

File system `vfat`

```
[root@dada /root]# mount /dev/fd0 -t vfat /mnt/floppy
```

File archivio o compressi

Un archivio o un file copresso può essere una soluzione per ridurre lo spazio, spostare un cospicuo numero di file in modo semplice o semplicemente creare una copia di sicurezza. Chi viene dal mondo Ms Windows è abituato al formato `.zip` mentre chi proviene dal mondo MacOS conosce `.sit` e `.hqx`. Nel mondo Unix esistono ulteriori soluzioni per archiviare/comprimere. La tabella mostra le estensioni e i relativi programmi necessari.

Estens.	Compressione	Decompressione	Pag.
.Z	compress <file>	uncompress file.Z	88
.gz	gzip <file>	gunzip file.gz	97, 97
.bz2	bzip2 <file>	bunzip2 file.bz2	85
.tar	tar cvf file.tar <file>	tar xvf file.tar	112
.tar.gz	tar zcvf file.tar.gz <file>	tar zxvf file.tar.gz	112
.tgz	tar zcvf file.tgz <file>	tar zxvf file.tgz	112
.zip	zip file.zip <file>	unzip file.zip	115
.lzh	lzh a file.lzh <file>	lha x file.lzh	
.rar	rar a file.rar <file>	unrar x file.rar	
.zoo	zoo -add file.zoo <file>	zoo -extract file.zoo	

Tabella 11.1: Comandi per archiviare/comprimere.

Il comando `tar` è di fatto quello più utilizzato in ambienti Unix anche perchè rappresenta uno standard consolidato nel tempo. Il comando per comprimere con maggiori risultati allo stato attuale è `bzip2`.

Unità Zip Iomega¹

Se desiderate utilizzare queste unità esterne di memorizzazione esterne e disponete di una unità collegata tramite la porta parallela è necessario digitare:

```
[root@dada /root]# modprobe ppa
```

Per attivare il modulo del kernel necessario a gestire l'unità e successivamente montare il disco come qualsiasi altro dispositivo.

```
[root@dada /root]# mount /dev/sda4 -t vfat /mnt/zip
```

Viene utilizzato un dispositivo di tipo scsi (`/dev/sda4` o qualsiasi altro) perché l'unità collegata sulla porta parallela viene emulata come se fosse scsi. Il tipo "vfat" è il file system comune in ambienti Windows ma potrebbe essere qualsiasi altro. La directory `/mnt/zip`, dove viene collegata l'unità, dovrà necessariamente esistere in precedenza.

Creare un disco di Avvio

Per garantire una certa sicurezza è bene creare un boot disk o disco di Avvio. Per tale scopo è necessario utilizzare il comando `mkbootdisk`.

Come primo passo è necessario visualizzare il file `/etc/lilo.conf` per capire quale kernel è stato caricato all'avvio della computer.

```
[root@dada /root]# cat /etc/lilo.conf
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
image=/boot/vmlinuz-2.2.12-20
    label=linux
    initrd=/boot/initrd-2.2.12-20.img
    read-only
```

¹<http://www.iomega.com>

```
root=/dev/hda1
```

Il campo di interesse è `image`, che ci indica il campo necessario per creare il disco di Avvio. Una volta conosciuto questo dato, inseriamo un dischetto formattato e procediamo digitando il seguente comando:

```
[root@dada /root]# mkbootdisk --device /dev/fd0 2.2.13-20
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort.
```

Premete `INVIO` per la creazione del dischetto.

Una volta terminata questa fase è bene riavviare il computer con il dischetto inserito e verificare il reale funzionamento.

Parte IV

Interfacce Grafiche

Capitolo 12

X Window

Un sistema X Window permette di gestire ambienti grafici a finestre e sfruttare al meglio le possibilità offerte dal networking, reti locali o extranet. I componenti fondamentali per un sistema di questo tipo si possono sintetizzare in:

Server X Si tratta del programma che gestisce l'interfaccia grafica e accetta le connessioni di rete per la gestione remota.

Client X Sono tutti i programmi che utilizzano l'ambiente grafico gestito dal Server X, sia sulla macchina locale che su computer in rete.

Protocollo X Rappresenta il protocollo di comunicazione utilizzato dal Server X e dai relativi client.

Window Manager Sono particolari software diretti a gestire le finestre ed ulteriori componenti sopra il Server X agendo anch'essi come client. Gli ambienti desktop sono l'evoluzione di questi ultimi perchè offrono la possibilità di gestire eventi in maniera intuitiva tramite il mouse, la periferica di puntamento largamente diffusa.

12.1 Window Manager

Un window manager è un particolare software che si aggiunge al server X, e offre una interfaccia visuale come sicuramente conoscerete. Finestre gestibili, riducibili e quant'altro per automatizzare le vostre operazioni e gestirle in buona parte con il vostro mouse. Nelle attuali distribuzioni viene data la possibilità di installare diversi window manager e tra quelli più diffusi e stabili si consiglia:

- Blackbox
<http://bb.themes.org>
- Window Maker
<http://www.windowmaker.org>
- Enlightenment
<http://www.enlightenment.org>

12.2 Ambienti desktop

Un ambiente desktop è qualcosa in più di un semplice window manager. Esso rappresenta un insieme di applicazioni volte a gestire in simbiosi diversi processi per rendere maggiormente

user friendly l'interfaccia visuale. Utenti GNU/Linux con una leggera esperienza alle spalle hanno visto rapidamente la venuta di queste soluzioni integrate sulla propria Linux box. Esistono due progetti sostanzialmente che si combattono il trono del desktop ufficiale per le distribuzioni GNU/Linux: il Kde e Gnome. Entrambi i progetti hanno caratteristiche di pregio e la sintesi nel presente testo non potrebbe essere aggiornata alla continua evoluzione di quest'ultime soluzioni software. Per ampliare la propria conoscenza in merito si consiglia di visitare i siti ufficiali seguenti:

- Progetto GNOME
<http://www.gnome.org>
- Helixcode GNOME
<http://www.helixcode.com>
- Progetto KDE
<http://www.kde.org>
- Informazioni su KDE
<http://www.mosfet.org>

Capitolo 13

Il Server XFree86

Il server XFree86 è una implementazione di X Window System, un potente e complesso ambiente grafico per sistemi operativi Unix. Il codice originale fu sviluppato al MIT; aziende commerciali hanno fatto di X lo standard per le piattaforma Unix.

Una libera e distribuibile versione di X Window System del MIT è stata successivamente sviluppata per processori Intel da parte di un team di sviluppatori e attualmente XFree86, il nome dell'implementazione citata, è supportato su diversi sistemi Unix compatibili e principalmente su GNU/Linux che ne distribuisce liberamente tutti gli eseguibili necessari, librerie di sviluppo e applicazioni per la gestione e configurazione.

13.1 Caratteristiche Tecniche

Il Server XFree86 essendo sviluppato per una tipologia di computer di largo consumo è compatibile con un ampia lista di schede video disponibili sul mercato. Salvo eccezioni dovute a schede video sconosciute o troppo aggiornate potreste riscontrare problemi di configurazione e supporto. Per evitare frustrazione e chiarire immediatamente i propri dubbi è consigliato visitare il sito di XFree86, <http://www.xfree86.org>, e cercare la lista di hardware compatibile. I tempi di attesa per il rilascio di driver aggiornati per la gestione di schede video molto recente è in continua riduzione sia per l'interesse in crescita verso GNU/Linux che mobilita le case produttrici di hardware sia per l'implementazione di standard come Vesa che accomunano caratteristiche tecniche che verranno successivamente illustrate.

Le distribuzioni GNU/Linux vengono correlate da diversi pacchetti software relativi al Server XFree86. Questi pacchetti hanno un ruolo o rispondono a vostre determinate esigenze personali. La modularità è dunque una delle caratteristiche fondamentali. Modularità che ruota essenzialmente ad un pacchetto base, il Server XFree86, al quale possono essere aggiunti driver specifici per la propria scheda video (dichiaratamente essenziali) ed insiemi di caratteri da utilizzare a seconda delle esigenze grafiche e linguistiche (esistono ad esempio caratteri cirillici tipici della lingua russa o ideogrammi della lingua cinese).

Le recenti versioni rilasciate di XFree86 associate alla generazione del kernel di Linux 2.2 e successivi hanno visto la comparsa di ulteriori pacchetti che hanno apportato nuove condizioni di utilizzo. Faccio riferimento ai pacchetti aggiuntivi XFS (X Font Server), un font server capace di gestire file caratteri di svariati formati e condividerli in rete con altri computer, e VFB (Virtual Frame Buffer) utile per chi possiede schede grafiche recenti senza supporto ma compatibili con lo standard Vesa. Non è questo un manuale per gestire il vostro Server XFree86 e non si vuole avere la pretesa di poter rispondere ad ogni vostra domanda o problema. L'ambiente X Window è dunque impraticabile a livello tecnico e di dettaglio in questo manuale ma è utile sapere che informazioni aggiornate e nuove caratteristiche sono disponibili sul sito internet specifico precedentemente citato.

13.1.1 Il file `XF86Config`

La configurazione del Server X per l'esecuzione comune in ambito locale viene rappresentata dal file `XF86Config`, usualmente disposto sotto la directory `/etc/X11/`. Il file in questione è composto da diverse sezioni, ognuna delle quali è diretta a configurare particolari dispositivi.

Sezione "Files" Vengono indicate in questa sezione la disposizione dei file necessari all'esecuzione del Server X, dalla codifica dei colori alla disposizione dei caratteri da utilizzare.

Sezione "Flags" Vengono gestite alcune opzioni importanti tra le quali:

- **DontZap**
Direttiva per attivare o meno l'interruzione del Server X tramite la combinazione di tasti `CTRL + ALT + BACKSPACE`.
- **DontZoom**
Direttiva per attivare o meno la possibilità di cambiare modalità video tramite la combinazione di tasti `CTRL + ALT + "+"` del tastierino numerico o `CTRL + ALT + <NUMERO>`.

Sezione "Keyboard" Sezione relativa alle configurazioni della tastiera.

Sezione "Pointer" Sezione relativa alla configurazione della periferica di puntamento, usualmente il comune mouse.

Sezione "Monitor" Sezione destinata a presentare le caratteristiche del vostro monitor sotto l'aspetto delle frequenze video supportate.

Sezione "Device" Sezione relativa al dispositivo da utilizzare per la gestione della vostra particolare scheda video.

Sezione "Screen" In questa sezione vengono aggiunte informazioni a quelle presentate dalla sezione "Monitor" e "Device" per indicare la definizione grafica da utilizzare.

Sezione "Xinput" Questa sezione è opzionale. E' diretta alla configurazione di particolari dispositivi di puntamento come le tavolette grafiche.

Sebbene sia possibile la configurazione manuale del file in questione, si sconsiglia altamente di farlo per ragioni pratiche e funzionali. Esistono diversi applicativi che vi permettono di configurare il vostro Server X in maniera semplice ed intuitiva come:

- `xf86config`
Programma storicamente correlato al sistema XFree86. Funziona a modalità carattere.
- `XF86Setup`
Questo strumento è disponibile su tutte le distribuzioni GNU/Linux. Tramite una modalità grafica è possibile configurare ogni aspetto di configurazione.
- `Xconfigurator`
E' lo strumento presente nella distribuzione Red Hat Linux. Permette di scegliere le configurazioni tramite il probing di quest'ultime. Funziona a modalità a carattere.
- `sax`
E' lo strumento presente nella distribuzione S.u.S.E. Linux. Permette di configurare il Server X in modalità grafica.

13.1.2 Esecuzione

Una volta configurato il Server X, possiamo lanciarlo digitando:

```
[darshan@dada darshan]$ startx
```

Il comando analizza le configurazioni presenti nel file XF86Config precedentemente citato e lancerà il Server X. Se decidessimo di utilizzare un particolare Window Manager o automatizzare l'avvio di determinati programmi, creando un file nella nostra home directory denominato `.xinitrc` ed aggiungere le nostre preferenze come segue:

```
# Questo è un commento
xsetroot -solid Gray
exec xterm &
exec fvwm
```

Nell'esempio presentato si sceglie di utilizzare il Window Manager `fvwm` con lo sfondo del colore grigio. Viene inoltre lanciato il programma `xterm`.

13.2 La Gestione dei Font

Il server X11R6 del X Consortium (X Server di default in gran parte delle distribuzioni GNU/Linux) è stato particolarmente curato sotto l'aspetto della gestione dei font. Dalle versioni X11R3 e X11R4, il formato di base gestito dei font era del tipo SNF (server normal format) un formato che era dipendente dal tipo di piattaforma sulla quale veniva utilizzato. Con la versione X11R5 questo limite fu abbattuto con la definizione di un protocollo di gestione progettato per gestire font indipendenti dal tipo di piattaforma e con il cambiamento del formato di base dei font da SNF a PCF (Portable Compiled Font) originalmente sviluppato da DEC. Il protocollo definito per la gestione dei font abbatteva ogni logica e limite precedentemente stilato. Mentre in precedenza la gestione dei font era vincolata dal tipo e dalla locazione dei font destinati all'utilizzo, venne data la possibilità di gestire font indipendenti dal tipo di piattaforma e dislocati in rete su altri server rompendo il precedente vincolo di avere ogni set di font su ogni macchina. Con la versione X11R6 il motore di renderizzazione del formato Type1 era stato implementato di base e non serviva più software aggiuntivo per gestire questo formato. Una cosa simile sta accadendo per il supporto dei font True Type, allo stato attuale gestito grazie alle librerie Freetype (www.freetype.org).

13.2.1 Formati riconosciuti

Ci sono diversi formati riconosciuti dal server X e qui di seguito viene presentata una breve descrizione di ognuno:

- **TTF** (True Type Font) Sono il formato carattere largamente diffuso in ambienti Windows e MacOS. Per utilizzare quest'ultimi sono necessarie le librerie FreeType, installate nelle configurazioni basilari di qualsiasi recente distribuzione GNU/Linux.
- **BDF** (Bitmap Distribution Format) E' il formato standard nelle distribuzioni di font e ci sono molto utility per convertire altri formati a questo tipo e viceversa. BDF è un formato definito da Adobe.
- **SNF** (Server Normal Format) E' il vecchio formato di font citato in precedenza ed è vincolato dal tipo di piattaforma e dalla locazione.
- **PCF** (Portable Compiled Font) E' un formato binario per i font bitmap capace di contenere sufficienti informazioni per essere letto da differenti piattaforme.

- **PHIGS** Questo tipo di font è applicabile in ambienti PEX e non ha nessuna relazione con il meccanismo per la gestione dei font X.
- **DWF** (DEC Windows Font) Sono font di tipo bitmap.
- **Intellifont** Sono font HP scalabili.
- **PFA/PFB** E' il formato Adobe Type1 PostScript utilizzabile dalla versione X11R5 grazie al contributo IBM, la quale ha donato un motore di rendering per questo formato al X Consortium.
- **Speedo** E' il formato sviluppato da Bitstream, la quale ha donato il motore di rendering e una serie di font al X Consortium.
- **FB** E' il formato Sun X11/NeWS, un formato di tipo bitmap utilizzato dai sistemi OpenWindows.
- **F3/F3B** E' il formato scalabile Sun Folio, anch'esso utilizzato dai sistemi OpenWindows.

13.3 Estensione Xinput

L'estensione XInput si aggiunge modularmente a X e permette di utilizzare dispositivi diversi dal mouse o la tastiera. L'estensione supporta una ampia gamma di periferiche, incluse tavolette grafiche, touch-screen e joystick. L'uso probabilmente più comune resta quello legato alle tavolette grafiche. In quest'ultimo caso, lo sforzo è dovuto al fatto che il puntatore di un mouse necessita di minori informazioni di una penna ottica, la quale a differenza deve valutare fattori come la pressione e l'inclinazione.

Allo stato attuale il supporto è decisamente buono anche se migliora rapidamente sia sotto l'aspetto di periferiche supportate che di gestione dei segnali di sensibilità ed inclinazione. Le più recenti distribuzioni di XFree86 (dalla versione 3.3.2) includono il modulo `xf86Wacom.so`, una libreria dedicata all'utilizzo di tavolette grafiche Wacom, sostanzialmente lo standard sul mercato per qualità e prestazioni. Si rimanda la lettura nel seguito di questo manuale per avere basilari informazioni di installazione.

F.A.Q.

Domanda: "Sono abituato all'ambiente desktop come quello di MacOS o Windows. Esiste qualcosa di simile per GNU/Linux?" Esistono diverse valide soluzioni. Nel corso degli ultimi anni si sono diffusi prevalentemente due ambienti desktop: GNOME¹ e KDE². Entrambi cercano di essere compatibili con la tecnologia CORBA e sono molto flessibili nell'uso quotidiano.

Domanda: "Tento di avviare il Server X ma dopo pochi secondi ritorno al punto di partenza in modalità caratteri. Che succede?" Mille le possibili cause. Un modo per analizzare la situazione è digitare:

```
[darshan@dada darshan]$ startx 2> errori.txt
```

Nel file `errori.txt` verranno riportati gli eventuali errori. Genericamente la maggiorparte degli errori avviene per una configurazione errata del file `XF86Config`, file di configurazione del Server X. Risulta molto utile tentare di rigenerare questo file con le utility indicate precedentemente.

¹<http://www.gnome.org>

²<http://www.kde.org>

Domanda: “Durante l’installazione ho scelto di riavviare la macchina direttamente in modalità grafica sotto il Server X. Una scelta razionale?” Il Server X è largamente diffuso e largamente testato sul piano della sicurezza. Utilizzare l’ambiente grafico avviandolo sin dalla fase di avvio può creare problemi nel caso che il file di configurazione `XF86Config` venga danneggiato. Se questo avviene premete contemporaneamente il tasto CTRL e i tasti funzione F1, F2...F6. Potrete accedere ad un altro terminale e lanciare le utility per rigenerare un corretto file di configurazione.

Domanda: “Ho scelto di avere la modalità grafica all’avvio del computer ma non riesco ad arrivare alla maschera di login per autenticarmi. Sempre che qualcosa disturbi quest’ultimo che ripetitivamente appare e scompare immediatamente. Cosa è necessario fare?” Le cause possono essere molteplici ma i casi tipici sono riscontrabili in una errata configurazione del file `XF86Config` e problemi di permessi sui file caratteri. Per risolvere la situazione e avere nuovamente la possibilità di accedere al sistema le soluzioni pratiche possono essere:

- riavviare la macchina, digitando al prompt di lilo “linux single”. In questo modo la macchina avvierà il vostro sistema GNU/Linux senza accessi di rete sul runlevel 2 e senza lanciare il server X. Al termine delle fasi di avvio di sistema apparirà il prompt di login testuale e da questo potrete accedere al sistema.
- un’ulteriore possibilità utile per una macchina con servizi di rete attivi è quella di accedere alla macchina da un’altra postazione in rete locale e cambiare il file `/etc/inittab`.

Domanda: “Dispongo di un computer con limitatissime risorse hardware. Come posso ottimizzare X in queste condizioni?” L’ambiente visuale `XFree86` richiede almeno 6Mb di ram e ulteriori esigenze sono richieste dagli ambienti desktop che si decide di utilizzare. GNOME e KDE sono particolarmente esosi verso soluzioni alternative limitate ma ottime in questo caso e senza grosse pretese. Utilizzare un Window Manager è una scelta conveniente.

Capitolo 14

Tipografia

La gestione dei font con GNU/Linux può risultare utile se si utilizza un ambiente grafico X Window e in tale ottica è stato ripreso un articolo scritto anni addietro per una rivista del settore apportando le necessarie modifiche e i necessari aggiornamenti.

14.1 La tecnologia Postscript

Il formato font PostScript precede di circa 6 anni il tipo True Type. Prima di questo c'erano diversi formati di font digitalizzati ma nessuno di questi diventò *standard*. L'Apple adottò il PDL (PostScript *Page Description Language*) da Adobe e lo utilizzò per la sua stampante Apple Laserwriter nel 1985. La combinazione di questa tecnologia applicata e l'introduzione di software per il publishing generò una rivoluzione delle tecnologie di impaginazione. Presto il linguaggio PostScript fu adottato su dispositivi di alto livello e divenne in maniera nativa linguaggio base per soluzioni grafiche. Adobe, detentrici dei diritti su questa tecnologia, rese disponibile le informazioni tecniche con alcuni limiti. Non rese pubbliche le specifiche dei font Type1 per la gestione del *hinting*. Le uniche specifiche rilasciate da Adobe sono relative ai font Type3 qualitativamente inferiori rispetto alla classe descritta precedentemente. Per Apple e Microsoft, in pieno sviluppo nella seconda metà degli anni '80, il fatto di vincolare i propri sistemi operativi con una gestione dei font scalabili posseduta da terze parti pesava e così dopo prolungati studi tecnici si arrivò alla creazione di una nuova specifica. La risposta di Adobe a questa presa di posizione fu il rilascio delle specifiche per il formato PostScript Type1 nel marzo del 1990 e la creazione del software ATM (Adobe Type Manager) capace di gestire questo tipo di font sia a video che in fase di stampa. L'ascesa della tecnologia True Type proseguì comunque inesorabilmente fino a raggiungere il 95% dei personal computer, crescita sicuramente legata alla diffusione degli stessi sistemi operativi che utilizzavano nativamente questa tecnologia.

14.2 Type1 e True Type

I font True Type e Postscript Type1 sono entrambi caratteri con specifiche tecniche aperte che li rendono multiplatforma e Outline. Multiplatforma significa che entrambe i tipi di font possono essere usati su diversi sistemi operativi mentre l'*outline* di un carattere ha un contorno (*glyphs*) descritto da un insieme di punti definiti da linee e curve, caratteristica che permette la rappresentazione di ogni singolo font e il suo eventuale ridimensionamento senza la perdita delle caratteristiche che lo contraddistinguono. Il processo di conversione da outline alla modellazione dei punti sull'output device è chiamata *rasterizzazione*. Essendo basati sull'*outline*, questi font possono subire, nel caso di piccole dimensioni o basse risoluzioni, inconsistenza nella rappresentazione di certe lettere non garantendo le iniziali caratteristiche grafiche proposte con dimensioni e risoluzioni

medie. Entrambi i due tipi i font hanno delle informazioni codificate al proprio interno (*hinting*) che soccorrono a questa problematica di rappresentazione.

14.3 MM e Quickdraw GX

Un'estensione dei font Type1 è la possibilità di sfruttare la tecnologia MM (*Multiple Master*), capacità di variare diverse caratteristiche del font stesso in maniera elastica interpolando l'originale forma su assi dimensionali. Il basilare esempio di utilizzo di questa estensione tocca il peso (*sottile* o *grassetto*), la larghezza (*condensata* o *estesa*) e la visualizzazione ottica (da *testo* a *video*).

L'alternativa studiata per aggiungere funzionalità simili alla tecnologia MM ha portato a Quickdraw GX, tecnologia che aggiungeva ulteriori caratteristiche ai font MM. Nonostante le estese qualità, GX non ha avuto ancora uno sviluppo elevato.

14.4 Opentype

Nel 1996 una iniziativa intrapresa da Adobe e Microsoft ha portato al *progetto Opentype*, soluzione per utilizzare l'*outline* della tecnologia Postscript e True Type in un *wrapper* in stile True Type. Questo significa che il formato supporterà ogni caratteristica delle due tecnologie, le estensioni GX/MM e il supporto per la codifica Unicode.

14.5 Configurazioni del Server X

Per aggiungere un font di tipo standard (*.pcf*, *.snf*, *.bdf*) bisogna spostarlo nell'apposita directory del path settati di default nel file `XF86Config`.

```
FontPath      "/usr/X11R6/lib/X11/fonts/misc"
FontPath      "/usr/X11R6/lib/X11/fonts/URW"
FontPath      "/usr/X11R6/lib/X11/fonts/Type1"
FontPath      "/usr/X11R6/lib/X11/fonts/Speedo"
```

Una volta sistemato quest'ultimo in questa sua nuova locazione è necessario lanciare un comando per rigenerare il file di configurazione che il server X andrà a leggere al momento dell'avvio denominato `fonts.dir` dove vengono riportate le caratteristiche principali di ogni singolo font presente (tipo, famiglia, grandezze, stili). Il file `fonts.dir` è presente in ogni directory del path ed è relativo al tipo di font presente.

Il comando in questione per aggiornare il `fonts.dir` è `mkfontdir`. Una volta lanciato quest'ultimo avviene l'aggiornamento del `fonts.dir` con in aggiunta la stringa precedentemente citata che prima dell'installazione mancava nel file. Un altro comando utile da conoscere può essere `xlsfonts` che elenca tutti i font attualmente disponibili all'uso e riconosciuti dal Server X elencandoli in una lunga lista organizzata di base alfabeticamente ma facilmente configurabile secondo le proprio esigenze con le opzioni presenti.

Parte V
Network

Capitolo 15

Introduzione

Il concetto di reti di computer si è evoluto nel corso di decenni ma solamente nell'arco degli ultimi anni ha subito una crescita esponenziale. Diverse possono essere le cause anche se la principale è da ritenersi il calo dei prezzi dei componenti hardware necessari: schede di rete, HUB, router, modem seriali o ISDN, connessioni ADSL.

La prima semplice carattere di distinzione di una rete è la sua dimensione ed espansione sul territorio:

- LAN (*Local Area Network*), anche denominata Rete Locale, è la tipologia più diffusa.
- MAN (*Metropolitan Area Network*), rete di computer con dimensione cittadine/metropolitane.
- WAN (*Wide Area Network*), rete di computer con dimensioni superiori alla circoscrizione cittadina o metropolitana.

La distinzione tra LAN e MAN è decretata dalle leggi vigenti sulle comunicazioni perchè quest'ultime impongono particolari vincoli nell'utilizzo di connessioni di rete che attraversino il suolo pubblico. La distinzione tra MAN e WAN è decretata semplicemente dal confine territoriale della città o metropoli di riferimento. L'avvento di Internet ha reso confusa la distinzione tra le tipologie illustrate, offuscandone i caratteri principali e trasformando ogni cosa in una *grande rete globale*. Sono così nate le seguenti parole chiave:

- Intranet
- Extranet

Altro non sono che porzioni di Internet circoscritte in determinate realtà, come ad esempio quelle aziendali, svincolate dai limiti fisici delle connessione.

Che cosa serve per creare una rete locale?

Possono esistere diverse soluzioni:

- Rete tra due computer
 - 2 scheda di rete Ethernet
 - 1 cavo di rete incrociato con jack RJ45
- Rete tra N computer
 - N schede di rete Ethernet
 - N cavi di rete con jack RJ45

– un HUB con N porte

Gli elementi che interessano GNU/Linux come supporto fisico sono le schede di rete e sull'installazione di quest'ultime poniamo il nostro interesse. La scelta di una scheda di rete può essere fatta in base alle personali esigenze ma i rapidi abbassamenti di costo che questi componenti hardware hanno subito nel corso degli anni permettono l'acquisto di soluzioni di qualità con cifre modeste. Dopo aver acquistato quest'ultime e averle installate all'interno del proprio computer negli slot PCI è necessario configurare il sistema operativo, il lato software, per utilizzare questi dispositivi.

La distribuzione presa come riferimento (Red Hat), offre `kudzu` come strumento di riconoscimento e configurazione di nuovo hardware installato sul proprio computer. Può essere lanciato da console:

```
[root@dada /root]# kudzu
```

oppure utilizzato ad ogni avvio del sistema. Per fare questo, attivarlo nei processi di boot con il seguente comando:

```
[root@dada /root]# ntsysv
```

Quando `kudzu` viene lanciato, vengono confrontati i componenti hardware presenti nel sistema e il file `/etc/sysconfing/hwconf`. Se vengono riscontrate configurazioni differenti viene automaticamente lanciato lo strumento idoneo alla nuova componente hardware, nel nostro caso per la scheda di rete. Insieme alla configurazione del componente hardware vi verrà chiesto di inserire degli indirizzi IP che in una situazione domestica potranno essere i seguenti:

Computer dada	
indirizzo IP della macchina:	192.168.0.1
maschera di sottorete (netmask):	255.255.255.0
names server primario:	212.216.112.222
names server secondario:	212.216.172.162

Computer monika	
indirizzo IP della macchina:	192.168.0.2
maschera di sottorete (netmask):	255.255.255.0
names server primario:	212.216.112.222
names server secondario:	212.216.172.162

Per avviare i servizi di rete e rendere attive le impostazioni digitare:

```
[root@dada /root]# /etc/rc.d/init.d/network start
```

per il computer `dada`, e:

```
[root@monika /root]# /etc/rc.d/init.d/network start
```

per il computer `monika`.

Capitolo 16

Sicurezza

Entrare *in rete* può portare vantaggi ma anche esporsi a problematiche di sicurezza e stabilità delle proprie attività. I passi seguenti sono semplici passi per garantire un buon livello di sicurezza per la vostra macchina.

Configurare `inetd.conf`

Il demone `inetd` viene chiamato il “super server” perchè si occupa di gestire un numero elevato di programmi su chiamate di rete. Il file `/etc/inetd.conf` illustra al proprio demone quali porte devono rimanere in ascolto e quali programmi si devono occupare delle singole porte.

Nel caso comune è frequente avere una configurazione non adeguata che lascia servizi attivi senza alcuno scopo proponendo varchi per attacchi da parte di maleintenzionati. Per tutelarvi seguite i seguenti passi:

1. Cambiare i permessi sul file di configurazione:

```
[root@dada /root]# chmod 600 /etc/inetd.conf
```

in modo che il vostro file abbia i permessi di lettura per il solo proprietario.

2. Assicuratevi che il proprietario del file sia l'utente `root`:

```
[root@dada /root]# stat /etc/inetd.conf
```

Se così non fosse digitare quanto segue:

```
[root@dada /root]# chown root /etc/inetd.conf
```

3. Modificate il file `/etc/inetd.conf` commentando ogni riga relativa a servizi non necessari.

```
# To re-read this file after changes, just do a 'killall -HUP inetd'
#
#echo    stream  tcp      nowait  root    internal
#echo    dgram   udp      wait    root    internal
#discard stream  tcp      nowait  root    internal
#discard dgram   udp      wait    root    internal
#daytime stream  tcp      nowait  root    internal
#daytime dgram   udp      wait    root    internal
#chargen stream  tcp      nowait  root    internal
#chargen dgram   udp      wait    root    internal
#time    stream  tcp      nowait  root    internal
#time    dgram   udp      wait    root    internal
#
```

```

# These are standard services.
#
#ftp    stream  tcp    nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
#telnet stream  tcp    nowait  root    /usr/sbin/tcpd  in.telnetd
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
#shell  stream  tcp     nowait  root    /usr/sbin/tcpd  in.rshd
#login  stream  tcp     nowait  root    /usr/sbin/tcpd  in.rlogind
#exec   stream  tcp     nowait  root    /usr/sbin/tcpd  in.rexecd
#comsat dgram   udp     wait    root    /usr/sbin/tcpd  in.comsat
#talk   dgram   udp     wait    root    /usr/sbin/tcpd  in.talkd
#ntalk  dgram   udp     wait    root    /usr/sbin/tcpd  in.ntalkd
#dtalk  stream  tcp     wait    nobody  /usr/sbin/tcpd  in.dtalkd
#
# Pop and imap mail services et al
#
#pop-2  stream  tcp     nowait  root    /usr/sbin/tcpd  ipop2d
#pop-3  stream  tcp     nowait  root    /usr/sbin/tcpd  ipop3d
#imap   stream  tcp     nowait  root    /usr/sbin/tcpd  imapd
#
# The Internet UUCP service.
#
#uucp   stream  tcp    nowait  uucp   /usr/sbin/tcpd  /usr/lib/uucp/uucico -l
#
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers." Do not uncomment
# this unless you *need* it.
#
#tftp   dgram   udp     wait    root    /usr/sbin/tcpd  in.tftpd
#bootps dgram   udp     wait    root    /usr/sbin/tcpd  bootpd
#
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers." Many sites choose to disable
# some or all of these services to improve security.
#
#finger stream  tcp    nowait  root    /usr/sbin/tcpd  in.fingerd
#cfinger stream  tcp    nowait  root    /usr/sbin/tcpd  in.cfingerd
#systat  stream  tcp    nowait  guest   /usr/sbin/tcpd  /bin/ps -auwx
#netstat stream  tcp    nowait  guest   /usr/sbin/tcpd  /bin/netstat -f inet
#
# Authentication
#
#auth   stream  tcp    nowait  nobody  /usr/sbin/in.identd in.identd -l -e -o
#
# End of inetd.conf

```

Una volta commentate le righe indicate è necessario riavviare il demone `inetd` digitando:

```
[root@dada /root]$ killall -HUP inetd
```

Una ulteriore precauzione può essere quella di rendere sicuro il nostro file di configurazione da modifiche e alterazioni. Per renderlo "immutabile" digitiamo:

```
[root@dada /root]$ chattr +i /etc/inetd.conf
```

Il comando digitato "blocca" il nostro file da ogni alterazione o modifica e non si possono crea-

re link a quest'ultimo. L'unico modo per poter nuovamente apportare modifiche è utilizzare lo stesso comando precedentemente digitato con un opzione differente:

```
[root@dada /root]$ chattr -i /etc/inetd.conf
```

Per maggiori informazioni sul comando `chattr` andate a pagina 144.

TCP_WRAPPERS

Distribuzioni GNU/Linux come quella di Red Hat permettono le richieste per ogni servizio. Utilizzando `TCP_WRAPPERS` si rende maggiormente sicuro il vostro computer da intrusioni malintenzionate dalla rete. Per applicare questi parametri di sicurezza è necessario agire su due file:

```
/etc/hosts.allow  
/etc/hosts.deny
```

Una configurazione tipo può essere la seguente:

1. Editare il file `/etc/hosts.deny` (utilizzando qualsiasi editor di testo, esempio `vi`) e aggiungere le seguenti stringhe di comandi:

```
#Accesso vietato a chiunque  
ALL:ALL@ALL, PARANOID
```

Ogni servizio richiesto da qualsiasi locazione è bloccato.

2. Editare il file `/etc/hosts.allow` e aggiungere:

```
sshd: 193.207.49.111 host.server.org
```

La configurazione è un esempio che concede unicamente all'ip indicato con risoluzione `erlug.linux.it` di connettersi al vostro computer utilizzando il demone `sshd` per le connessioni criptate.

3. Verificate le vostre configurazioni lanciando il seguente comando:

```
[root@dada /root]$ tcpdchk
```

16.0.1 Il file `/etc/aliases`

Il file relativo agli `aliases` può essere utilizzato per attacchi o per ottenere particolari privilegi all'interno del sistema. Per tale motivo è bene apportare alcune modifiche per garantirsi. Ecco di seguito un esempio di file `/etc/aliases`:

```
#  
# @(#)aliases 8.2 (Berkeley) 3/5/94  
#  
# Aliases in this file will NOT be expanded in the header from  
# Mail, but WILL be visible over networks or from /bin/mail.  
#  
# >>>>>>>> The program "newaliases" must be run after  
# >> NOTE >> this file is updated for any changes to  
# >>>>>>>> show through to sendmail.  
#  
# Basic system aliases -- these MUST be present.  
MAILER-DAEMON: postmaster
```

```

postmaster:    root

# General redirections for pseudo accounts.
bin:           root
daemon:        root
#games:        root    ← commentato o rimosso
#ingres:        root    ← commentato o rimosso
nobody:        root
#system:        root    ← commentato o rimosso
#toor:          root    ← commentato o rimosso
#uucp:          root    ← commentato o rimosso

# Well-known aliases.
#manager:       root    ← commentato o rimosso
#dumper:        root    ← commentato o rimosso
#operator:      root    ← commentato o rimosso

# trap decode to catch security attacks
#decode:        root    ← commentato o rimosso

# Person who should get root's mail
#root:          marc

```

le righe indicate possono essere commentate o rimosse a piacimento e limiteranno le possibilità di eventuali attacchi. Ad esempio, l'alias "decode" viene utilizzato per convertire eventuali file binari convertiti in ASCII con uuencode. Potete capire da voi come sia importante commentare tale voce per limitarne totalmente l'utilizzo.

Una volta terminate le modifiche è necessario lanciare il comando seguente:

```
[root@dada /root]$ /usr/bin/newaliases
```

16.0.2 Mascherare informazioni di sistema

Per camuffare informazioni relative al proprio sistema a persone che si collegano remotamente è bene prendere alcune precauzioni. Se le uniche connessioni remote sono concesse tramite telnet è necessario modificare il file `/etc/inetd.conf` aggiungendo alla riga seguente:

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

l'opzione "-h" come segue:

```
#telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd -h
```

Se si desidera ampliare il mascheramento a tutte le eventuali connessioni remote con servizi diversificati è bene modificare il file `/etc/rc.d/rc.local`:

```

...
# This will overwrite /etc/issue at every boot. So, make any changes you
# want to make to /etc/issue here or you will lose them when you reboot.
#echo "" > /etc/issue
#echo "$R" >> /etc/issue
#echo "Kernel $(uname -r) on $a $(uname -m)" >> /etc/issue
#cp -f /etc/issue /etc/issue.net
#echo >> /etc/issue
...

```

Come si può notare le righe riportate sono state commentate con un “#” per evitare che venga generato il file `issue` e `issue.net`. A questo punto bisogna eliminare i seguenti file:

```
[root@dada /root]$ rm -fr /etc/issue
[root@dada /root]$ rm -fr /etc/issue.net
```

I file cancellati sono i “banner” informativi presentati sia durante una connessione di rete che locale.

16.03 Il file `/etc/securetty`

Questo file permette di configurare quale device `TTY` è concesso all’utente `root`. Di seguito una configurazione tipo del file:

```
tty1
#tty2
#tty3
#tty4
#tty5
#tty6
#tty7
#tty8
```

Come potete notare sono stati commentati tutti i device `TTY` tranne il primo. L’utente `root` potrà connettersi al sistema unicamente da quest’ultimo. Se avessimo commentato ogni device sarebbe stato possibile entrare nel sistema come utenti comuni e diventare `root` con il comando `su` (pagina 111).

Capitolo 17

Web server

Per attivare un web server e *pubblicare* pagine ipertestuali è necessario un determinato software. Secondo le statistiche presentate da Netcraft¹ oltre il 60% dei web server nel mondo utilizza Apache, un software maturo e stabile disponibile con ogni distribuzione di GNU/Linux.

Verifichiamo l'esistenza del software nel sistema:

```
[root@dada /root]$ rpm -qa|grep apache
apache-devel-1.3.12-2
apache-1.3.12-2
```

In questo caso il software è installato, altrimenti:

```
[root@dada /root]$ rpm -ivh apache-devel-1.3.12-2.rpm
[root@dada /root]$ rpm -ivh apache-1.3.12-2.rpm
```

Le configurazioni di Apache sono presenti nei file:

```
/etc/httpd/conf/access.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/conf/srm.conf
```

anche se, per una scelta di Red Hat, tutto è stato concentrato in un unico file di configurazione: `httpd.conf`.

httpd.conf Per un utilizzo comune di Apache bisognerà cambiare unicamente la direttiva specifica per il nome del server (*ServerName*). Ogni ulteriore configurazione può essere trascurata. Viene riportato di seguito il file in esame considerando le direttive maggiormente importanti e comuni.

```
ServerType standalone
```

Il web server può girare come software standalone o essere gestito con il superserver Inetd.

```
ServerRoot "/etc/httpd"
```

La directory base dove disporre i file di configurazione, errori e log.

```
PidFile /var/run/httpd.pid
```

¹<http://www.netcraft.com>

Il file che memorizza il numero pid (process id) quando il server web viene lanciato.

```
ScoreBoardFile /var/run/httpd.scoreboard
```

Il file indicato con la direttiva ScoreBoardFile viene utilizzato per memorizzare informazioni relative ai processi interni.

```
#ResourceConfig conf/srm.conf  
#AccessConfig conf/access.conf
```

I file di configurazione precedentemente citati e attualmente commentati.

```
Timeout 300
```

Il tempo di scadenza per ogni richiesta.

```
LoadModule vhost_alias_module modules/mod_vhost_alias.so  
LoadModule env_module          modules/mod_env.so  
...  
AddModule mod_vhost_alias.c  
AddModule mod_env.c  
...
```

Le direttive per caricare i moduli con Apache.

```
User nobody  
Group nobody
```

L'utente e il gruppo con il quale utilizzare il demone.

```
ServerAdmin root@localhost
```

Il riferimento e-mail dell'amministrazione del servizio.

```
#ServerName localhost  
ServerName monika
```

Il nome di riferimento per *chiamare* il servizio web.

```
DocumentRoot "/home/httpd/html"
```

La directory dove riporre i propri documenti ipertestuali.

```
<Directory />  
    Options FollowSymLinks  
    AllowOverride None  
</Directory>  
  
<Directory "/home/httpd/html">  
    Options Indexes Includes FollowSymLinks  
    AllowOverride None  
    Order allow,deny  
    Allow from all  
</Directory>
```

Direttive per assegnare permessi su determinate directory.

```
DirectoryIndex index.php index.html index.htm index.shtml
```

Il nome delle pagine che vengono automaticamente richiamate all'apertura di un URL. Se per esempio visitassimo il sito `http://monika/` verrebbe aperta automaticamente la pagina `index.html` che abbiamo disposto nella directory `/home/httpd/html/`.

```
AccessFileName .htaccess
<Files ~ "\.ht">
    Order allow,deny
    Deny from all
</Files>
```

Configurazione per determinare i permessi di accesso su determinate directory protette da password.

```
ErrorLog /var/log/httpd/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

CustomLog /var/log/httpd/access_log common
#CustomLog /var/log/httpd/referer_log referer
#CustomLog /var/log/httpd/agent_log agent
#CustomLog /var/log/httpd/access_log combined
```

Direttive per determinare la forma dei log del servizio e indicare nome e dislocazione nel file system di quest'ultimi.

```
Alias /icons/ "/home/httpd/icons/"

<Directory "/home/httpd/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Directory dove è possibile disporre immagini/icone per uso generico e comune a tutte le pagine senza specificazione di percorso.

```
ScriptAlias /cgi-bin/ "/home/httpd/cgi-bin/"

<Directory "/home/httpd/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

Direttive che indicano la disposizione per i cgi, (*common gateway interfaces*) e i permessi per la loro esecuzione. I cgi possono essere applicazioni compilate come script di bash.

Per maggiore informazione sulle direttive non considerate si può fare riferimento al manuale del software Apache, disponibile nella directory `/home/httpd/html/manual/` oppure direttamente sul sito <http://www.apache.org>. Per lanciare il server web:

```
[root@dada /root]$ /etc/rc.d/init.d/httpd start
```

Per attivarlo ad ogni avvio:

```
[root@dada /root]$ ntsysv
```

e attivare la voce *httpd*.

Capitolo 18

NFS

NFS è l'acronimo di Network File System e rappresenta la soluzione maggiormente diffusa in ambienti Unix per esportare aree del disco locale in rete, in modo da creare aree comuni per la condivisione file.

Per attivare un servizio di questo tipo sulla propria rete locale è necessario dare per scontato alcune cose:

1. Le due (o più) macchine in rete abbiano già configurato scheda di rete e un proprio indirizzo IP.
2. Sia installato sulla macchina *server* che esporta la partizione alcuni pacchetti software.

Per il primo punto trovate le risposte a pagina 141, mentre per la voce successiva partiamo con la ricerca dei pacchetti nel sistema definito *server*:

```
[root@dada /root]$ rpm -qa|grep nfs
[root@dada /root]$ rpm -qa|grep portmap
```

Se l'esito non visualizza la presenza dei pacchetti è necessario installarli:

```
[root@dada /root]$ rpm -Uvh nfs-utils-0.1.6-2.i386.rpm
[root@dada /root]$ rpm -Uvh portmap-4.0-19.i386.rpm
```

A questo punto definiamo il nostro obiettivo:

“...creare un'area comune su una macchina dove tutti possano disporre i propri file e dividerli per eventuali modifiche”

Creiamo una directory per il nostro scopo:

```
[root@dada /root]$ mkdir /mnt/areacondivisa
```

Secondo le nostre intenzioni dobbiamo dare i permessi di lettura, esecuzione, scrittura a chiunque quindi:

```
[root@dada /root]$ chmod o+rwX /mnt/areacondivisa
```

Ora è necessario aggiungere quanto segue al file `/etc/exports`:

```
/mnt/areacondivisa    monika(rw,root_squash)
```

Come si nota, viene indicata la directory creata precedentemente e il nome host autorizzato a montare quest'ultima. Nell'esempio la macchina di riferimento si chiama `monika` ma è possibile indicare anche l'IP di quest'ultima. Tra le parentesi possono essere presenti diverse opzioni tra cui segnaliamo:

- `rw` (Read Write), lettura e scrittura.
- `root_squash`, per negare la scrittura all'utente `root`.
- `ro` (Read Only), sola lettura.

A questo punto l'unica cosa necessaria da fare sulla macchina *server* è avviare i seguenti servizi:

```
[root@dada /root]$ /etc/rc.d/init.d/portmap start
[root@dada /root]$ /etc/rc.d/init.d/nfs start
```

Se desiderate aggiungere ulteriori aree da condividere nel file `/etc/exports` non sarà necessario riavviare i servizi ma basterà semplicemente il comando seguente:

```
[root@dada /root]$ exportfs -a
```

Nel caso in cui ad ogni riavvio della macchina vorreste automaticamente attivare il servizio fate riferimento al comando `ntsysv` e attivate i servizi `nfs` e `portmap`. Per verificare che l'area condivisa sia veramente esportata:

```
[root@dada /root]$ showmount -e
/mnt/areacondivisa monika
```

Come mostra il risultato del comando tutto è andato a buon fine. A questo punto andiamo a modificare la macchina *client*, colei che dovrà utilizzare l'area condivisa in rete. Come prima cosa creiamo una directory come segue:

```
[root@monika /root]# mkdir /mnt/dadanfs
```

Montare l'area condivisa in rete associandola alla directory creata:

```
[root@monika /root]# mount dada:/mnt/areacondivisa /mnt/dadanfs
```

Operazione conclusa!

Se le vostre intenzioni sono quelle di utilizzare l'area condivisa direttamente all'avvio della macchina sarà necessario mettere le seguenti argomentazioni nel file `/etc/fstab`:

```
dada:/mnt/areacondivisa /mnt/dadanfs nfs exec,dev,suid,rw 1 1
```

oppure utilizzate `linuxconf` per essere agevolati da un'interfaccia amichevole.

Capitolo 19

Collegarsi ad Internet

GNU/Linux è uno strumento perfetto per la rete e offre una incommensurabile serie di strumenti per *dominare* quest'ultima. La condizione minima per accedere ad Internet si può riassumere in pochi strumenti:

- un accesso tramite un Internet Service Provider
 - numero di telefono
 - username e password
- un modem analogico (o ISDN)
- un collegamento telefonico

Se il modem è esterno basterà collegarlo ad una qualsiasi porta seriale (9 aghi) mentre se fosse una scheda interna sarà sufficiente installarla nell'apposito slot sulla scheda madre. Nella tabella 155 viene illustrata la tabella di dispositivi seriali e la correlazione nel mondo DOS/Windows.

Per comodità d'uso viene generato un link simbolico all'interfaccia seriale da utilizzare in base alla porta seriale in cui è collegato il modem. Nell'esempio che segue il modem è collegato alla COM2, ovvero `/dev/ttyS1`:

```
[root@dada /root]$ ln -s /dev/ttyS1 /dev/modem
```

controlliamo:

```
[root@dada /root]$ ls -l /dev/modem
lrwxrwxrwx 1 root root 11 mag 24 06:33 /dev/modem -> /dev/ttyS1
```

Dispositivi delle porte seriali		
DOS/Windows	Linux kernel 2.0	Linux kernel 2.2
COM1	<code>/dev/cua0</code>	<code>/dev/ttyS0</code>
COM2	<code>/dev/cua1</code>	<code>/dev/ttyS1</code>
COM3	<code>/dev/cua2</code>	<code>/dev/ttyS2</code>
COM4	<code>/dev/cua3</code>	<code>/dev/ttyS3</code>

Tabella 19.1: Rappresentazione delle porte seriali per GNU/Linux

wvdial / wvdialconf

Questa applicazione è un sistema intelligente per collegarsi ad Internet. Possiede la capacità di testare i parametri adeguati di configurazione per il modem grazie a preventivi esami euristici e offre la possibilità di connettersi ad Internet anche senza un preventivo script di configurazione. I parametri di connessione generici sono presenti nel file di configurazione `/etc/wvdial.conf`.

Per configurare quest'ultimo digitate:

```
[root@dada /root]$ wvdialconf /etc/wvdial.conf
```

Saranno scansionate le porte seriali per verificare l'esistenza di un modem e gli eventuali parametri saranno memorizzati nel file indicato. Per dare una rappresentazione visiva della cosa viene riportato il file `/etc/wvdial.conf`:

```
[Modem0]
Modem = /dev/ttyS14
Baud = 115200
Init1 = ATZ
SetVolume = 1
Dial Command = ATDT
Init4 = ATM1L1

[Dialer Defaults]
Modem = /dev/ttyS14
Baud = 115200
Init1 = ATZ
SetVolume = 1
Dial Command = ATDT
Init4 = ATM1L1
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
; Phone = <Target Phone Number>
; Username = <Your Login Name>
; Password = <Your Password>

[Dialer libero]
Username = mariorossi@libero.it
Password = lamiaparolasegreta
Phone = 0547421010
Inherits = Modem0
Stupid mode = 0
```

Lo script riportato presenta anche la configurazione per la connessione ad un noto provider Internet italiano.

rp3 / rp3-config

Questo applicativo è un'interfaccia grafica a `wvdial` e vi permette di configurare la connessione Internet in modo semplice ed intuitivo seguendovi passo per passo.

Digitate:

```
[root@dada /root]$ rp3-config
```

L'interfaccia che si presenterà è piuttosto semplice e vi permetterà di configurare il modem e gli script di connessione. Una volta terminato sarà possibile lanciare la connessione nel seguente modo:

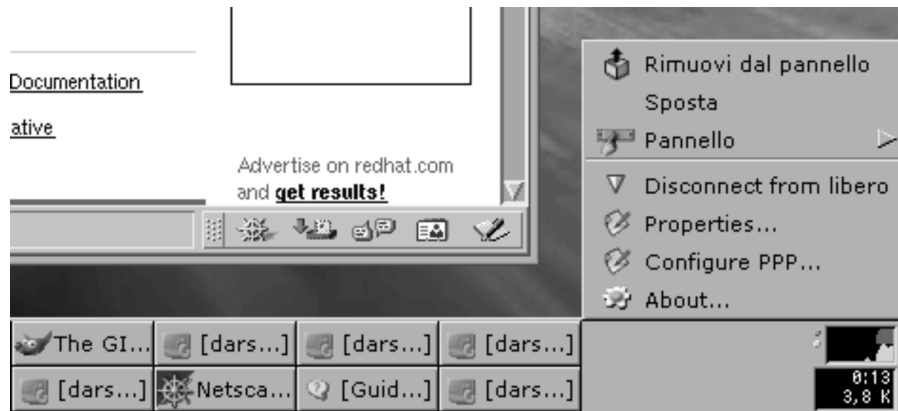


Figura 19.1: Il pannello di GNOME e l'applet rp3.

```
[root@dada /root]$ rp3
```

L'applicazione rp3 si integrerà perfettamente al pannello di GNOME come è mostrato in figura 19.1.

linuxconf

Linuxconf è un sistema integrato per accentrare tutte le configurazioni del vostro sistema in un'interfaccia comune come struttura logica e disponibile per console, X e Web. Senza entrare troppo nel merito dell'applicativo, tocchiamo la sezione che ci interessa in questo momento.

Per configurare la connessione Internet è necessario scendere al seguente livello:

- Configurazione
 - Rete
 - Client tasks
 - PPP/SLIP/PLIP

I passi da seguire non si discostano da quelli precedentemente visti. Sarà necessario configurare il tipo di interfaccia (PPP) e indicare il numero di connessione, username, password e il dispositivo da utilizzare per il modem (`/dev/modem`). Una volta seguiti questi passi, il collegamento ad Internet sarà possibile digitando:

```
[root@dada /root]$ ifup ppp0
```

L'identificativo dell'interfaccia da attivare è ppp0. Per disconnettersi:

```
[root@dada /root]$ ifdown ppp0
```

pppconfig

Questo comando è presente nella distribuzione Debian (2.2). Vi permette di configurare lo script di connessione ad Internet seguendovi nelle varie fasi.

```
debian:~$ pppconfig
```

Una volta terminata la procedura e salvato lo script, il collegamento ad Internet sarà attivato digitando:

```
debian:~$ pon nomescript
```

Se non è specificato il comando tenta di lanciare lo script con nome `Provider`. La procedura per disconnettersi è la seguente:

```
debian:~$ poff
```

Capitolo 20

Soluzioni per la stampa

Con Red Hat Linux, la distribuzione presa come riferimento, è piuttosto semplice configurare una stampante per diversi tipi di soluzioni. Il sistema di stampa è identico a quello di tutte le distribuzioni esistenti ma è possibile utilizzare il comando `printtool` (figura 20.1). per configurare la stampa in maniera semplice e veloce.

Prima di procedere a qualsiasi eventuale configurazione accertatevi che sia stato installato il software necessario come segue:

```
[root@dada /root]# ps afx|grep lpd
2618 ? S 0:00 lpd
3692 pts/1 S 0:00 \_ grep lpd
```

Questo è il caso in cui il demone di stampa è attivo. Se non rilevate questi dati è possibile attivarlo con il comando seguente:

```
[root@dada /root]# /etc/rc.d/init.d/lpd start
```

Se la sfortuna ci persegue e non esiste il comando indicato si potrebbe sospettare che il software non è presente nel sistema. Verifichiamo il nostro dubbio.

```
[root@dada /root]# rpm -qa|grep lpr
```

Se anche quest'ultima analisi non presenta risultati sarà necessario installare il software di stampa dal relativo cdrom della distribuzione GNU/Linux.

```
[root@dada /root]# mount /dev/cdrom
[root@dada /root]# rpm -Uvh /mnt/cdrom/RedHat/RPMS/lpr-0.50-4.rpm
```

A questo punto attiviamo il demone di stampa come indicato sopra.

Stampante locale

Questo è probabilmente il caso più frequente; una stampante collegata al computer domestico su porta parallela. La figura indica i dati necessari per la configurazione.

Stampante remota su GNU/Linux

Il caso può essere il seguente. Una rete locale in cui esistono diverse piattaforme GNU/Linux e si vuole condividere la periferica di stampa. In questo caso possiamo determinare una macchina Server, colei che offre il servizio, una macchina Client, colei che si avvale di ciò che le viene autorizzato a fare.

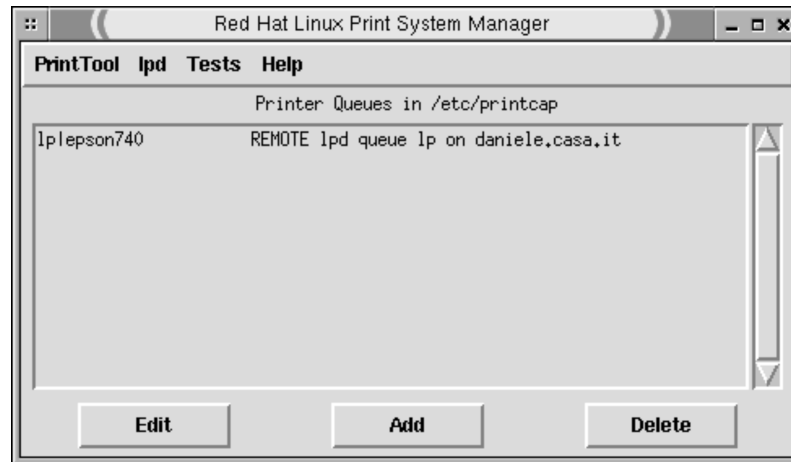


Figura 20.1: Il comando printtool.

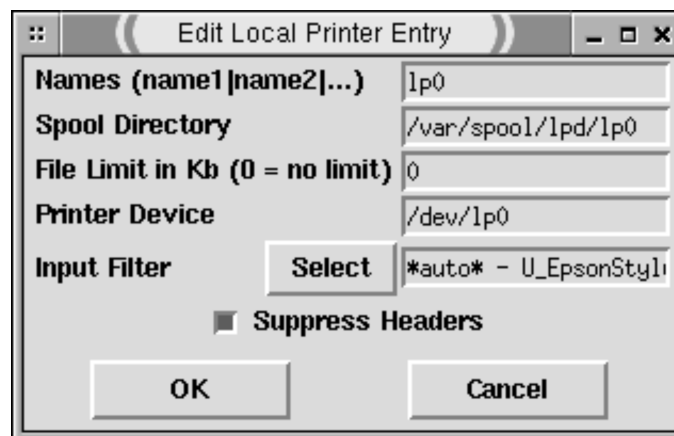


Figura 20.2: Configurazione di una stampante locale.

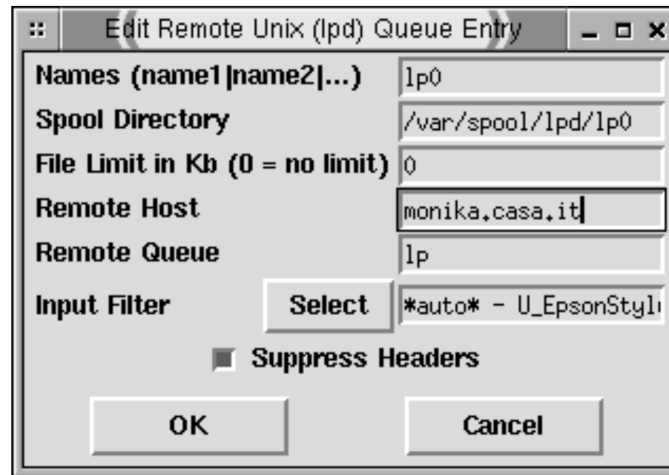


Figura 20.3: Configurazione di una stampante in rete GNU/Linux.

La configurazione della macchina Server per il servizio di stampa è identico a quello di una stampante locale. Dal lato della macchina Client le cose cambiano ed è richiesto di indicare la denominazione della macchina Server, la coda di stampa e ulteriori parametri.

Per rendere effettiva la stampa remota di questo tipo è necessario compiere un passo aggiuntivo. All'inizio del manuale si è parlato del file `/etc/hosts` e ora è necessario controllare se esistono le indicazioni per trovare in rete la macchina Server.

```
127.0.0.1 localhost localhost
192.168.0.1 dada dada.casa.it
192.168.0.2 monika monika.casa.it
```

Dove monika è la macchina Client dove stiamo configurando la stampa remota sulla macchina Server dada. A questo punto è necessario generare un file che indichi al demone di stampa locale quale macchina Server può effettuare il servizio di stampa. Il file sarà `/etc/hosts.lpd` e conterrà semplicemente l'indicativo seguente:

```
dada.casa.it
```

La macchina Server.

Stampante su rete Windows

Questa situazione è molto simile alla precedente ma è necessario far dialogare due sistemi operativi differenti e l'approccio tecnico cambia. E' inanzitutto necessario il pacchetto software Samba (samba-client) che permette appunto a GNU/Linux il dialogo con le piattaforme Microsoft.

```
[root@dada /root]# rpm -qa|grep samba
```

Se non compaiono voci sarà necessario installare il software.

```
[root@dada /root]# rpm -Uvh /mnt/cdrom/RedHat/RPMS/samba-client-2.0.6-9.rpm
```

Il comando `printtool` si occuperà di chiedervi le informazioni necessarie per la configurazione come appare nella figura 20.4.

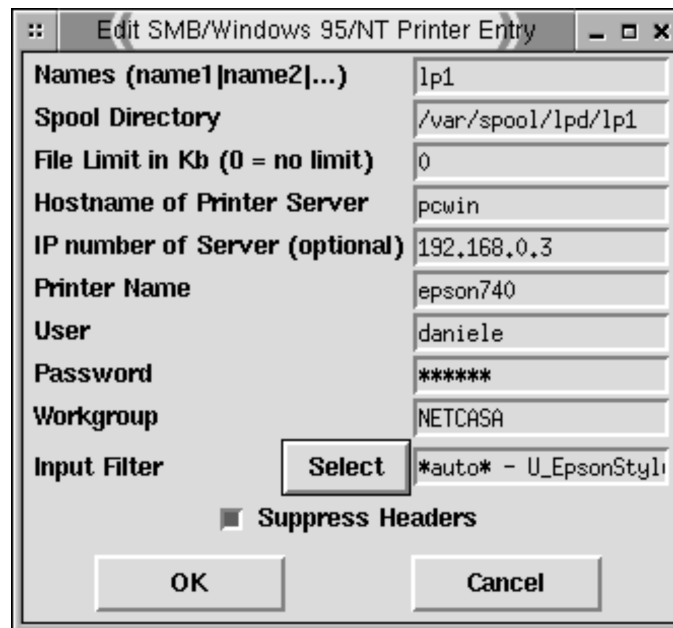


Figura 20.4: Configurazione di una stampante in rete MS windows

Parte VI

Complementi

Acronimi

Acronimo	Significato	Traduzione
AFAICT	As Far As I Can Tell	Per quanto possa dirne
AFAIK	As Far As I Know	Per quanto ne so
AFK	Away From Keyboard	Lontano dalla tastiera
AIUI	As I Understand It	Per come la capisco
B4	Before	Prima
BBL	Be Back Later	Torno più tardi
BCNU	Be Seeing You	Ci vediamo
BRB	Be Right Back	Torno subito
BTW	By The Way	A proposito
BWQ	Buzz Word Quotient	Quoziente di parolona
CUL	See You Later	Ci vediamo più tardi
EOF	End Of File	Fine del file
F2F	Face to Face	Faccia a faccia
FAQ	Frequently Asked Question	Domande poste di frequente
FOC	Free Of Charge	Gratis
FUBAR	Fucked Up Beyond All Recognition	Fottuto oltre ogni limite
FWIW	For What It's Worth	Per quel che vale
FYA	For Your Amusement	Per il tuo divertimento
FYE	For Your Entertainment	Per il tuo intrattenimento
FYI	For Your Information	Per tua informazione
<G>	Grin	Sogghigno
GA	Go Ahead	Va avanti
GAL	Get A Life	Svagati pensando ad altro
HHOJ	Ha Ha, Only Joking	Ha, Ha, stavo solo scherzando
HHOS	Ha Ha, Only Serious	Ha, Ha, dico sul serio
IMBO	In My Bloody Opinion	La mia fottuta opinione
IME	In My Experience	Secondo la mia esperienza
IMHO	In My Humble Opinion	Dal mio umile punto di vista
IMO	In My Opinion	Dal mio punto di vista
IOW	In Other Words	In altre parole
IRL	In Real Life	Nella realtà
ISTM	It Seems To Me	Mi sembra
JAM	Just a Minute	Un minuto...
L8R	Later	Più tardi
LOL	Laughts Out Loud	Risata di gusto
MUD	Multi User Dungeon	Gioco di ruolo multiutente
MUG	Multi User Game	Gioco di ruolo
OAQ	Over and Out	Passo e chiudo
OBTW	Oh, By The Way	Ah, a proposito

OIC	Oh, I See	Ah, ho capito
OMG	Oh My God	Oh dio mio
OTOH	On The Other Hand	D'altra parte
RFD	Request For Discussion	Richiesta di discussione
ROFL	Rolls On Floor Laughing	Mi rotolo sul pavimento dalle risate
RSN	Real Soon Now	Ormai manca poco
RTFAQ	Read The FAQ	Leggi le FAQ
RTFM	Read The Fuckin' Manual	Leggi il fottuto manuale
RUOK	Are You Ok?	E' tutto a posto?
TIA	Thanks In Advance	Grazie in anticipo
TNX	Thanks	Grazie
TTYL	Talk To You Later	Ne parliamo più tardi
TVM	Thanks Very Much	Grazie Mille
WTH	What The Hell	Che diavolo
YHM	You Have Mail	C'è posta per te

Smiley

Smiley	Significato
: -)	Sorriso di base per trasmettere ironia e gioia
;-)	Ammicciamento
: -(Delusione
: -	Indifferente
: ->	Commento sarcastico e pungente
>:->	Espressione diabolica
: -	Arrabbiato
>:-	Molto arrabbiato
: -)>	Barbuto
%+(Picchiato di recente
: ^)	Naso rotto
:-)	Sopacciglia cespugliose
3:-)	Mucca o cornuto
: -t	Irritato
x-)	Strabico
: ' -)	Piangente
: -e	Molto deluso
<:-)	Domanda stupida
s:-)	Elvis
: "-(Fiumi di lacrime
: -!	Mi sto trattenendo
8-)	Occhialuto
: *)	Raffreddato
: -*	Bacio
: -D	Grassa risata di gusto
: -9	Mi lecco i baffi
: 8)	Maiale
=:-)	Punk
o:-)	Angelo
: -v	Urlante
: -i	Fumatore
: -6	Gusto acido nella bocca
: -w	Parla con lingua biforcuta
: -/	Indeciso
: -[Vampiro
: -))	Felicissimo
: -((Tristissimo
: -c	Molto felice

d:-)	Uomo con cappello
-o	Sbadiglio

Codici delle Nazioni¹

Codice	Nazione	Codice	Nazione
ap	Antartide	ar	Argentina
at	Austria	au	Australia
be	Belgio	br	Brasile
ca	Canada	ch	Svizzera
cl	Cile	cr	Costa Rica
cs	Cecoslovacchia	cy	Cipro
de	Germania	dk	Danimarca
ec	Ecuador	ee	Estonia
es	Spagna	fi	Finlandia
fr	Francia	gl	Groenlandia
gr	Grecia	hk	Honk Kong
hr	Croazia	hu	Ungheria
ie	Irlanda	il	Istraele
in	India	is	Islanda
it	Italia	jp	Giappone
kr	Corea	kw	Kuwait
lu	Lussemburgo	mx	Messico
my	Malaisia	nt	Olanda
no	Norvegia	nz	Nuova Zelanda
pl	Polonia	pr	Portorico
pt	Portogallo	ru	Federazione Russa
se	Svezia	sg	Singapore
sk	Slovacchia	sl	Slovenia
th	Tailandia	tn	Tunisia
tr	Turchia	tw	Taiwan
uk	Regno Unito	us	Stati Uniti
ve	Venezuela	za	Sud Africa

¹Riferimento a pagina 14.

Indice

I Storia	7
1 Hackers	9
2 Internet	11
2.1 Storia	11
2.1.1 Autostrade dell' Informazione	11
2.1.2 Il Vecchio Continente	12
2.2 DNS e i Nomi Dominio	12
2.2.1 Internet e le internet	12
2.2.2 Le Origini del Domain Name System	13
2.2.3 Nomi di Dominio	13
2.2.4 DNS in Breve	14
2.3 Servizi Telematici	14
2.4 Le comunità virtuali	15
3 Unix	17
3.1 Da Multics a Unix	17
3.2 Gli interessi commerciali	19
3.3 Free Software Foundation	20
II Filosofia	23
4 Open Source	25
4.1 Il progetto GNU	25
4.2 Open Sources Definition	39
III GNU/Linux	53
5 Il fenomeno GNU/Linux	55
5.1 La nascita	55
5.2 Distribuzioni	57
5.3 Documentazione	59
5.4 Valore aggiunto	61
6 Linux User Group	63
6.1 Linux User Group	63
7 Installazione	65
7.1 Cosa vogliamo fare?	65
7.1.1 Windows e GNU/Linux	65
7.1.2 GNU/Linux	66

7.2	Fase di Installazione	66
7.2.1	Quante partizioni per GNU/Linux?	67
7.2.2	Utilizzare fdisk	67
8	Il Sistema	71
8.1	Processi in fase di avvio	71
8.2	File System	73
8.3	Permessi sui file	75
8.4	Le Shell	76
9	La Bash	79
10	Pratica	83
10.1	Eseguibilità	83
10.2	Comandi	84
10.2.1	adduser	84
10.2.2	apropos	84
10.2.3	apt-get	84
10.2.4	bzip2	85
10.2.5	cal	85
10.2.6	cardctl	85
10.2.7	cat	86
10.2.8	cd	86
10.2.9	chatr	87
10.2.10	chmod	87
10.2.11	compress	88
10.2.12	cp	88
10.2.13	cpio	88
10.2.14	cron	89
10.2.15	date	90
10.2.16	df	90
10.2.17	diff	90
10.2.18	dpkg	91
10.2.19	dvipdfm	91
10.2.20	dvips	91
10.2.21	echo	92
10.2.22	env	92
10.2.23	export	92
10.2.24	file	92
10.2.25	find	93
10.2.26	finger	93
10.2.27	fmt	94
10.2.28	fold	94
10.2.29	free	94
10.2.30	ftp	94
10.2.31	gpg	95
10.2.32	grep	96
10.2.33	gzip	97
10.2.34	gunzip	97
10.2.35	kill	97
10.2.36	halt	97
10.2.37	history	97
10.2.38	hwclock	98
10.2.39	irc	98

10.2.40 last	99
10.2.41 latex	99
10.2.42 ln	100
10.2.43 locate	100
10.2.44 lpc	101
10.2.45 lpr	102
10.2.46 lprm	102
10.2.47 lpq	102
10.2.48 ls	102
10.2.49 mail	103
10.2.50 man	103
10.2.51 mc	103
10.2.52 mesg	104
10.2.53 mkdir	104
10.2.54 more	104
10.2.55 mount	104
10.2.56 mv	105
10.2.57 nice	105
10.2.58 nl	105
10.2.59 nohup	106
10.2.60 passwd	106
10.2.61 patch	106
10.2.62 pico	107
10.2.63 pr	107
10.2.64 ps	107
10.2.65 pstree	107
10.2.66 pwd	108
10.2.67 reboot	108
10.2.68 rm	108
10.2.69 rpm	108
10.2.70 rpm2cpio	110
10.2.71 set	110
10.2.72 shutdown	110
10.2.73 slocate	110
10.2.74 split	111
10.2.75 ssh	111
10.2.76 su	111
10.2.77 talk	111
10.2.78 tar	112
10.2.79 tcpdchk	113
10.2.80 telnet	113
10.2.81 top	113
10.2.82 uname	114
10.2.83 umount	114
10.2.84 vi	114
10.2.85 vlock	115
10.2.86 zip	115
10.2.87 wc	115
10.2.88 whatis	115
10.2.89 write	116
10.2.90 who	116
10.2.91 xdvi	116
10.2.92 xlock	116
10.2.93 xvidtune	116

10.2.94 xterm	117
10.2.95 xsetpointer	117
10.2.96 xsetroot	117
11 Post installazione	119
11.1 Gestione dei file	119
IV Interfacce Grafiche	127
12 X Window	129
12.1 Window Manager	129
12.2 Ambienti desktop	129
13 Il Server XFree86	131
13.1 Caratteristiche Tecniche	131
13.1.1 Il file XF86Config	132
13.1.2 Esecuzione	133
13.2 La Gestione dei Font	133
13.2.1 Formati riconosciuti	133
13.3 Estensione Xinput	134
14 Tipografia	137
14.1 La tecnologia Postscript	137
14.2 Type1 e True Type	137
14.3 MM e Quickdraw GX	138
14.4 Opentype	138
14.5 Configurazioni del Server X	138
V Network	139
15 Introduzione	141
16 Sicurezza	143
16.0.1 Il file /etc/aliases	145
16.0.2 Mascherare informazioni di sistema	146
16.0.3 Il file /etc/securetty	147
17 Web server	149
18 NFS	153
19 Collegarsi ad Internet	155
20 Soluzioni per la stampa	159
VI Complementi	163

Elenco delle tabelle

4.1	Caratteristiche delle licenze software.	50
6.2	Mappa Linux User Group Italiani.	64
8.1	Permessi sui file.	76
10.1	Comandi per l'editor vi	114
11.1	Comandi per archiviare/comprimere.	123
19.1	Rappresentazione delle porte seriali per GNU/Linux	155

Elenco delle figure

3.1	Ritchie e Thompson alla console del calcolatore PDP-11.	18
3.2	Ritchie e Thompson in foto recenti.	18
4.1	Richard Stallman	26
4.2	Logotipo di GNU	30
4.3	GNU's not Unix	34
5.1	Linus Torvalds fotografato dopo un talk.	56
5.2	Tux, un pinguino come amico	57
8.1	GMC, il file manager di GNOME 1.2	75
19.1	Il pannello di GNOME e l'applet rp3.	157
20.1	Il comando printtool.	160
20.2	Configurazione di una stampante locale.	160
20.3	Configurazione di una stampante in rete GNU/Linux.	161
20.4	Configurazione di una stampante in rete MS windows	162

Indice analitico

- adduser, 84
- Alix, 35
- Alpha, 58
- Andy Tanenbaum, 55
- Appunti Linux, 60
- apropos, 84
- apt-get, 84
- ARPAnet, 11

- Bash, 32, 55
- BCPL, 19
- Bill Joy, 19
- BIND, 13
- boot disk, 123
- Brian Kernighan, 17
- Bruce Perens, 39
- BSD, 48
- BSD Unix, 19
- bzip, 85

- C, 19
- cal, 85
- Caldera, 51
- cardctl, 85
- cat, 86
- cd, 86
- chattr, 87, 144
- chiave pubblica, 95
- chmod, 87
- Chuck Haley, 19
- COAS, 51
- compress, 88
- copyleft, 31
- cp, 88
- cpio, 88, 110
- crittografia, 95
- cron, 89

- date, 90
- Debian, 40, 51
- DEC, 19
- Dennis Ritchie, 17
- df, 90
- diff, 90
- Digital, 26

- Disco di Avvio, 123
- DNS, 12
- DOS, 19
- dpkg, 91
- dvipdfm, 91
- dvips, 91

- echo, 92
- Emacs, 40
- emacs, 29
- env, 92
- Eric Raymond, 34, 49
- ErLug, 60
- export, 92
- exportfs, 154

- file, 92
- find, 93
- finger, 93
- fingerprint, 95
- fmt, 94
- fold, 94
- free, 94
- free software, 28
- Free Software Foundation, 20, 31
- ftp, 94

- GARR, 14
- gcc, 29, 32
- GhostScript, 45
- GNOME, 34, 42
- GNU Hurd, 35
- GnuPG, 95
- gpg, 95
- GPL, 46, 47
- grep, 96
- gunzip, 97
- gzip, 97

- hacker, 25
- Halloween Document, 50
- halt, 97
- Harmony, 42
- Hillel, 28
- history, 97

- hosts.allow, 145
- hosts.deny, 145
- hwclock, 98

- i386, 58
- IBM, 50
- ILDP, 60
- ILS, 60
- inetd.conf, 143
- Intel, 58
- irc, 98

- James Hughes, 57
- JEEVES, 13

- KDE, 36, 51
- Ken Thompson, 19
- Kevin Dunlap, 13
- kill, 97
- killall, 144

- Larry Augustin, 41
- last, 99
- LaTeX, 99
- LDP, 59
- Ldr, 60
- LGPL, 33
- Linus Torvalds, 55
- Linux, 36, 50, 55
- Linuxconf, 51
- ln, 100
- locate, 100
- Logotipo di GNU, 30
- lpc, 101
- lpq, 102
- lpr, 102
- lprm, 102
- ls, 102
- LZW, 34

- Macintosh, 42
- mail, 103
- man, 103
- mc, 103
- mesg, 104
- MIT, 27, 30
- mkbootdisk, 124
- mkdir, 104
- modprobe, 123
- more, 104
- Motif, 36
- mount, 104, 123
- Mozilla, 45, 49
- MPL, 49
- Multics, 17, 28

- mv, 105

- Netiquette, 15
- nice, 105
- nl, 105
- nohup, 106
- NPL, 49

- Pascal, 19, 29
- passwd, 106
- patch, 106
- Penguin Computing, 41
- Perl, 48
- pico, 107
- Pluto, 60
- Postfix, 50
- PostScript, 45
- PPC, 58
- pr, 107
- ps, 107
- pstree, 107
- pwd, 108

- Qt, 42, 51

- reboot, 108
- Red Hat, 51, 58
- Richard Stallman, 20, 41, 58
- rm, 108
- rp3, 156
- rpm, 108

- Sam Ockman, 41
- set, 110
- shutdown, 110
- Silicon Graphics, 4, 19
- Slackware, 58
- Sparc, 58
- split, 111
- ssh, 111
- su, 111
- Sun, 19
- SuSE, 58
- Symbolics, 26
- System V, 19

- talk, 111
- tar, 34, 112
- TCP_WRAPPERS, 145
- tcpdchk, 113
- telnet, 113
- TeX, 29
- top, 113
- Troll Tech, 42
- Tux, 57

umount, 114
uname, 114
useradd, 84
USG, 19

VA Research, 41
VAX, 19
vi, 19, 114
vlock, 115
VMS, 19
VUCK, 29

wc, 115
whatis, 115
who, 116
Windows, 19
write, 116
wvdial, 156
wvdialconf, 156

X, 30
xdvi, 116
XFree86, 36
xlock, 116
xsetpointer, 117
xsetroot, 117
xterm, 117
xvidtune, 116

zip, 115

Licenza

GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, \LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3. You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Note