

Enrico F. Macchia

Breve

Introduzione

a LINUX

Piccolo Manuale su Linux ad Uso Didattico

Versione 1.03.01 Febbraio 2002

Indice

1	Introduzione	7
1.1	La <i>nascita</i> di Linux	8
1.2	La storia di Linux in breve	11
1.3	Siti Internet e distribuzioni Linux	18
1.3.1	I siti ufficiali internet	18
1.3.2	Le principali distribuzioni	19
1.3.3	Altre <i>ispirazioni</i> al software libero diverse da Linux	20
2	L'installazione	23
2.1	Root e client	26
3	Primi comandi	27
4	Collegamento Internet	33
5	Creazione e Rimozione di un account	35
6	Permessi	37
7	DOSC	39
8	\LaTeX 2ϵ: Un creatore di testi	41
8.1	Elementi di tipografia	42
8.2	Preparazione del documento	42
9	Gnu General Public Licence	47
9.1	Preamble	47
9.2	Terms and Conditions	47
9.3	No Warranty	50
9.4	How to Apply these Terms to Your New Programs	51

Prefazione

*

Nello spirito della Free Software Foundation e della licenza GPL, conformemente al progetto GNU¹, è permesso distribuire in tutto o in parte, con qualsiasi mezzo, fisico od elettronico, questo manuale lasciando integri l'avviso di copyright e questi permessi su tutte le copie. È consentito copiare e distribuire versioni modificate sotto le stesse condizioni delle copie testuali, fornendo anche le sezioni che riportano la “Licenza Pubblica Generale GNU”.

È permesso copiare e distribuire traduzioni di questo manuale in altre lingue sotto le condizioni per le versioni modificate. La “Licenza Pubblica Generale GNU” e la “Licenza Pubblica Generale Librerie GNU” possono essere incluse in traduzioni approvate dalla Free Software Foundation al posto dell'originale inglese.

È vietata la distribuzione commerciale.

Eventuali lavori derivati dovranno comprendere questo documento-avviso. I lavori derivati da questo documento vanno distribuiti senza imporre restrizioni aggiuntive.

Sarò grato a quanti mi vorranno far notare errori introdotti nella presente versione del documento inviando una mail ad **enricofmacchia@libero.it**

Il testo di questo documento è stato scritto con emacs e compilato con L^AT_EX 2_ε.

¹GNU è acronimo di G[NU] not Unix. Vedi alle ultime pagine di questo manuale la licenza riportata in inglese)

Dice il saggio: un PC su cui installato un sistema operativo Linux è molto più difficile da usare rispetto ad un PC su cui è installato un sistema microsoft; infatti per operare sul primo occorrono 102 tasti, per operare sul secondo ne bastano solo tre: CTRL + ALT + DEL. A volte uno solo: il pulsante di Reset!

Capitolo 1

Introduzione

Sino a pochi anni fa il mondo dell'informatica, almeno quello della grande utenza, sembrava inesorabilmente condannato dal predominio monopolistico della microsoft e del suo proprietario b.g. La comparsa sul mercato del sistema operativo denominato Windows 95, unitamente alla produzione di più potenti processori, azzerò di fatto il sistema operativo (DOS) che per più di dieci anni aveva favorito, assieme alle prime versioni di windows la diffusione dell'informatica, e fece fare d'un balzo passi innanzi ai PC che si vedevano non più relegati nell'angusto spazio degli home computers, ma che andavano a rivestire il ruolo di vere e proprie workstation di tutto rispetto, anche se casalinghe.

La comparsa ancora, pressoché all'unisono, di altri prodotti eccellenti, applicazioni di Office quali Acces ed Excel, sistemi operativi avanzati quali NT4, nella versione workstation e server, fece il resto. Era sufficiente cliccare su un'icona per veder comparire il contenuto del disco o della directory: non importava quello che il PC stesse facendo, importava che lo facesse per noi. Paradossalmente, fu proprio quest'ampia diffusione informatica sotto Windows che favorì dapprima il crescere di una coscienza e di una conoscenza informatica, e quindi l'evolversi lento, ma costante e tenace, del sistema operativo che andiamo qui a descrivere e che porta il nome di Linux.¹

Linux oggi non costituisce più una moda, ma rappresenta piuttosto la seria necessità da parte di utenti cresciuti di avere a che fare per usi professionali con un sistema operativo che non solo non si blocchi in continuazione, ripetendo la frase che tanti di noi hanno visto scritta sul computer: ... questo programma ha eseguito un'operazione non valida e sarà chiuso, ma che aiuti a conoscere l'informatica.

Il discorso che cercherò di qui in avanti di condurre si basa anzitutto su alcune divergenze fra i due prodotti che saranno sommariamente esaminate, quindi ci si addenterà lievemente nella conoscenza di Linux (occorrerebbero diversi volumi di qualche migliaia di pagine ciascuno per descrivere Linux), tenendo presente che si tratta comunque di un sistema operativo che richiede un minimo (si fa per dire) di sacrificio, e che costringe inevitabilmente a ristudiare da capo l'informatica, o meglio, a studiarla per la prima volta.

Quindi ancora si esamineranno, nel campo d'interesse comune, alcune applicazioni di Linux. Oggi come oggi, Linux è l'unico concorrente serio di Xp, l'evoluzione di NT4 (NT2000 è stato un flop ancora peggiore di WindowsME), e sia nella versione workstation che server, i prodotti microsoft non sono in grado di concorrere seriamente con Linux: il paragone con sottoprodotti informatici quali Windows 95 e 98 è offensivo per Linux. Un server serio, che non sia facilmente bucabile da parte di chi abbia profonde conoscenze informatiche, in ambiente microsoft non esiste.

Ma cerchiamo di fare innanzitutto un poco di cronistoria.

¹Questa è l'unica volta che nell'indice analitico comparirà la voce "Linux". Trattandosi di voce ricorrente non verrà più ripetuta anche perché questo è il tema del presente documento.

1.1 La nascita di Linux

Linux nasce da una scommessa, fatta più a se stesso che ad altri, di un giovane studente della facoltà di Ingegneria di Helsinki, Linus Torvalds, il quale un giorno di una decina d'anni fa, in seguito all'uscita sul mercato dei processori della serie i386, constatata l'effettiva innovazione, lettine i data sheet, lanciò la sua scommessa in rete.

La scommessa consisteva nel desiderio di costruire un efficiente sistema operativo compatibile con lo standard UNIX, per quanto possibile migliore ed operante sotto comuni PC. L'idea principale ruotava soprattutto attorno a quattro perni: il sistema doveva essere:

- stabile;
- effettivamente multitasking;
- gratuito ispirandosi al principio della libera circolazione non onerosa delle idee;
- insieme all'eseguibile andavano distribuiti i sorgenti in modo da rendere possibile a chiunque l'implementazione del nuovo sistema.

Scritto in linguaggio C, Linux crebbe lentamente con Internet ed in Internet (allora gli albori), e proprio quella che sembrava la sua fragilità (un gruppo di strenui volontari informatici), si trasformò col tempo nella sua forza: il volontariato informatico generò lentamente, ma senza sosta, un codice sorgente sempre più affidabile, un nucleo, il famoso kernel, che una volta compilato mette in esecuzione l'intero sistema.

Linux è una versione gratuita di UNIX, (il nome è naturalmente un misto di Linus e Unix) di cui è sensibilmente più veloce, e pur usando gli stessi comandi di UNIX si adatta sia ai PC, sia ad altre macchine, quali Digital Alpha, PowerPC, Sun Sparc, Apple Mac, Amiga, MIPS.

Le sue caratteristiche si possono riassumere così: pienamente multitasking, multiuser, multithreading, TCP/IP, comunica ottimamente con altri sistemi (Microsoft, Apple, Novel), ha un'interfaccia grafica di prima grandezza: la ben nota Xwindows. Il suo principale vantaggio non è però quello di essere gratuito (:in informatica la stabilità del sistema non ha prezzo!), ma di essere un sistema che oltre ad essere gratuito, è in aggiunta effettivamente stabile, anche se, diciamo subito, raggiungere la stabilità del sistema non è cosa facile per i non esperti, richiede una profonda conoscenza di quanto Linux fa, di quanto Linux esige che sia fatto, in relazione all'hardware che si trova in quel momento disponibile con riferimento al software applicativo montato.

Non è questione di drivers; in quanto questi ormai sono a disposizione a iosa grazie alla non onerosità del prodotto, è questione che il sistema richiede un diverso e nuovo approccio all'informatica.

Tanto per spiegarci con un esempio, la lettura dei dati contenuti in un CD da parte del neofita, se non guidata da opportuni manuali è tuttora in alcune distribuzioni un'impresa ardua. Invano si cercherebbe di attivare il ben noto comando d: cliccando sull'apposita icona che, tra l'altro, non c'è! In ambiente Unix non esiste il concetto di periferiche tanto caro e noto agli utenti di windows; in ambiente Unix il CD si monta, ed una volta montato non è possibile neanche toglierlo (prima di smontarlo) perché tanto il CD come hardware quanto il suo contenuto come software vanno a far parte dell'hardware e del software del PC, o meglio della stazione di lavoro.

La differenza fra un utente di sistema windows, Mac o Amiga, rispetto ad un utente di ambiente Unix è paragonabile alla differenza che esiste fra chi sa fare le quattro operazioni perché conosce le tabelline (utente Linux), e chi ha invece assoluto bisogno della calcolatrice, ed è perduto senza di essa perché o non le conosce o ha completamente dimenticato la logica delle quattro operazioni.

I punti di forza di Linux, riassumendo quanto sin qui detto, possono essere quindi, principalmente, quelli di seguito elencati:

1. possibilità reale (e non virtuale come per windows) di eseguire più programmi nello stesso momento (multitasking)
2. possibilità di sopportare più utenti sulla stessa macchina senza licenze aggiuntive (multiuser);

3. possibilità di girare su diverse macchine (multiplatform);
4. possibilità di librerie statiche e dinamiche; e) gestione avanzata della memoria;
5. compatibilità con SCO, SVR3, SVR4 a livello di runtime;
6. predisponibilità da sempre per il 2000;
7. pseudoterminali pty's;
8. consolle virtuali multiple tramite l'uso combinato di tasti ALT & CTRL+F1, F2, F3, F4, F5, F6, F7; F8;
9. filesystem proprietario con partizioni a 4TB;
10. possibilità di nomi sino a 250 caratteri;
11. client o server in reti Novel,....

A meno di dieci anni dal suo concepimento, Linux è oggi ampiamente usato non solo a livello di home computer ma anche a livello di aziende piccole, medio-grandi e multinazionali, le quali optano per questa scelta non soltanto perché vedono azzerarsi i prezzi delle licenze d'uso, ma perché, anche, risparmiano sulle Vaxstation e, soprattutto, lo si ripete ancora, optano per un sistema stabile, un sistema che quando si blocca costringe l'operatore ad individuare il proprio errore.

In Italia lo si trova in molte aziende dell'Emilia Romagna dalle dimensioni medio piccole, e per quanto riguarda le grandi aziende ci si può limitare a dire che la NASA ha "mandato" LINUX nello spazio utilizzandolo sullo Space Shuttle nel 1998, Kodak lo usa per il processo del colore, Boeing, Xerox e Mercedes lo hanno adottato. Corel "clona" pressoché tutto il proprio software a prezzi irrisori per ambienti Linux, ed infine, mentre i prodotti in ambiente windows, nelle loro varie versioni, arrancano ancora verso i 32 bit, Linux è già da tempo disponibile a 64 bit, e come dicevamo poco sopra non ha mai saputo cose fosse il "millennium bug", nè ha mai compreso, assieme all'autore di questo scritto, perché fra il 1999 ed il 2000 siano stati gettati via tanti denari inutilmente.

Il rilascio della prima versione del kernel avvenne nel 1991: non so se si possa affermare che a quella data Linux esistesse già, perché il kernel sapeva fare solo poche cose: uno dei primi risultati ottenuti da Torvalds, fu riuscire a stampare, alternativamente, quattro A e quattro B! Attualmente lo sviluppo di Linux è supportato da migliaia di programmatori sparsi su tutta la terra, ed il lavoro viene coordinato via internet. I sorgenti del kernel sono disponibili in rete: le versioni stabili riportano il numero finale pari, quelle di sviluppo (non necessariamente instabili, il numero finale dispari).

Per quanto sia difficilissimo fatturare un mercato che non esiste, si calcola (e la stima è per difetto) che nel 1998 l'utenza Linux sia cresciuta del 220 per cento rispetto all'anno precedente, e che abbia ormai raggiunto ed abbondantemente superato il mercato Mac. Oggi come oggi Linux conta circa 12.000.000 di utenti effettivi e stabili. Linux è rilasciato secondo la licenza GNU General Public Licence della Free Software Foundation, e mentre il kernel di Linux è sotto copyright da parte del Dr. Linus Torvalds, i vari programmi che girano sotto il kernel sono di proprietà dei rispettivi autori. È consentito a chiunque modificare, copiare o distribuire il software nel rispetto della licenza GNU-GPL. Rimando comunque ad Internet per contenuti più approfonditi in materia perché spesso si fa confusione fra l'open source e software gratuito. Non sono la stessa cosa!

Tutto positivo dunque? Sì ma con qualche distinguo. Tra i sistemi operativi in circolazione, se esistesse un Oscar per il carattere più brutto questo andrebbe assegnato di diritto a Linux, anche se questo brutto carattere è frutto più che altro dell'approccio iniziale che presenta una certa ruvidità per chi è ormai abituato al plug and play, ribattezzato peraltro in ambiente Linux plug and prei o prega e piangi, da parte di chi è ormai assuefatto soltanto a cliccare per ottenere. Ma questo carattere ripaga offrendo un ambiente Unix più completo perché conserva ancora la validità delle vecchie chiamate Unix, un'interfaccia con sette consolle virtuali, un numero indefinito di terminali

seriali, un multitasking serio, una velocità che non ha paragoni con le lumache microsoft sia per quanto riguarda l'interfaccia che l'esecuzione dei singoli programmi, ed il tutto senza ricorrere a processori da 1 giga o più: basta un 486 a 66 Mhz per offendere windows montato su un MMX a 166 MHz. Sarà un caso se più del 50 per cento dei provider mondiali usa Linux ed il famoso web-server Apache e non NT4 server? Anche Infoseek Corporation ha reso disponibile il sistema Ultraseek Server 3.0 per Linux.

Le cose comunque va detto stanno cambiando radicalmente. Case come la RedHat, da sempre forti estimatrici del sistema operativo, hanno giudicato i tempi maturi e stanno agevolando sempre più le operazioni delicate dall'installazione, e sorgono distribuzioni (chiariremo fra breve questo concetto) in continuazione. Se Linux può accontentarsi di 50 Mb, se può addirittura girare su un floppy, se si sceglie l'opzione di installazione massima (tutti gli applicativi forniti con la distribuzione) si arriva anche ad un Giga e 100 di software d'installazione e più, e le nuove versioni possono sì girare anche su un .386, ma con quale conseguenza dei tempi vi lascio immaginare!

Il software? È notevolissimo e tutto di prima scelta, perché proviene da università o da programmatori di prima grandezza. Finora tanto il sistema operativo quanto i pacchetti si scaricavano unicamente dalla rete. Ora si comprano a prezzi stracciati o nelle edicole assieme alle riviste d'informatica.

La carenza di documentazione è ormai un ricordo lontano. La scarsa documentazione era dovuta soprattutto alla non capillare presenza della rete nel nostro Paese sino a solo il 1996. Internet è ormai massicciamente presente ovunque, e se non a casa, nel proprio ufficio, non esiste quasi persona che non abbia usato questo strumento.

I Libri non sono più scritti soltanto per l'esperto come la famosa The Bible of Linux (tradotta anche in italiano) e richiedono soltanto serenità d'animo e voglia d'apprendere. Ma la documentazione va crescendo a livello di articoli. Sulle riviste d'informatica non esiste ormai un CD che non dedichi qualche centinaia di Mb ad applicazioni Linux, e non esiste più alcuna rivista che non parli ad ogni sua uscita del fenomeno Linux. Pensate a Linux & C, che vide la luce nel novembre 1999, a LinuxJournal, a LinuxMagazine, a LinuxPratico, ed a tutte le numerose riviste che escono saltuariamente con due o tre CD.

Vedere i siti www.oltrelinux.com, www.mondolinux.com.

Si sarebbe quasi tentati di dire che Linux si è conquistato il mercato, senonché il mercato non c'è, e bisogna limitarsi a dire che Linux vede, giorno dopo giorno, aumentare le proprie utenze, infittirsi le proprie schiere.

Un fatto è inequivocabile: oggi come oggi, il pinguino scelto tempo fa da uno sconosciuto (allora) studente d'ingegneria di un'università della Finlandia appare di frequente nelle pagine web sia di centri universitari che di singoli utenti. Quel pinguino dall'aspetto tranquillo, tipico di un pinguino che ha appena degustato il proprio pranzo, come lo volle lo studente Linus, quel pinguino docile e sorridente ammicca sempre più da ogni sito web del mondo, quel pinguino che sa essere però, proprio come è nella caratteristica della sua specie, anche aggressivo, come pochi immaginano sia quella particolare specie di animali, aggressivo al punto giusto solo per difesa personale, tanto che un uomo chiamato b.g. cerca ormai seriamente di correre ai ripari nei confronti di un sistema operativo che non riempie i suoi files e le sue librerie di dati inutili e di alcuna importanza per l'applicazione, come egli è invece da anni assuefatto a fare, unicamente per far scoppiare, sempre più, la memoria e l'HD dei PC e costringere la gente a comprare nuovi PC, nuovi software,... unicamente per farlo arricchire.

S'impone la risposta ad una domanda: perché usare Linux? Ne vale la pena? Se il sistema si ferma, non risparmio tempo nel riavviarlo piuttosto che apprendere un nuovo sistema tutt'altro che semplice? La risposta sta, come sempre, nell'uso che si fa del PC, dipende da quello che voglio e debbo fare con il software. Se debbo scrivere una lettera, non microsoft, ma il DOS e WordStar 4 bastano e avanzano. La validità di Linux va distinta in gradini: a livello commerciale ed a livello amatoriale, sempre però precisando che vediamo entrambi i gradini da un'angolazione strettamente professionale.

Per il settore commerciale abbiamo già esposto i vantaggi: un abbattimento assoluto di prezzi in termini di prima licenza più licenze aggiuntive ed un sistema funzionale su PC senza investire

milioni per l'acquisto di una Vaxstation, la realizzazione di server affidabili a costo zero come software.

Per il settore amatoriale, a seconda del singolo campo di ricerca (astronomia, geologia, sismologia, chimica, grafica ...) c'è la possibilità di aver sottomano, sempre a costo zero il miglior software del mondo, quello usato nei più avanzati istituti di ricerca, con la conseguente possibilità di fornire, se impegnati in un'attività di ricerca, risultati affidabili, precisi e puntuali.

Nel caso di Linux non "esce" un nuovo sistema operativo ogni due anni. Ogni sei mesi, al più ogni anno esce un nuovo kernel che si sovrappone al precedente senza reinstallare tutto (anche se pare che gli utenti di linux, me compreso) trovino la loro massima soddisfazione nel vedere un'installazione riuscita!

A livello personale, l'adozione di Linux al posto di un altro sistema operativo trova spiegazione e fondamento soltanto nella opportunità di apprendere. *er divertirsi!* avvertiva già dieci anni fa il Dr. Torvalds, allora non ancora dottore. Per quanto si detesti, e giustamente, microsoft, non si può trasformare la scelta del sistema operativo in una guerra di religione. Mi sento comunque di affermare, in tutta convinzione, che se non si tratta di una guerra di religione, siamo molto vicini ad una guerra per la libertà, e non solo del software. Le multinazionali informatiche, e non solo microsoft, stanno conducendo una guerra scellerata ai prodotti dell'open source, vedendo che alcuni utenti professionali si indirizzano verso altre mete. Se ciò non bastasse i governi le sostengono. Il nostro è completamente supino al dominio microsoft, e gli SU non sono certo da meno. Andate a vedere cosa è quella scelleratezza chiamata Digital Millennium Act e vi accorgete di essere veramente cittadini a tempo e condizioni da altri dettate. L'11 settembre 2001 ha poi dato man forte alla cancellazione soppressione e sospensioni di diritti. Ma questa è filosofia, torniamo all'informatica.

1.2 La storia di Linux in breve

2

.... la storia di Linux ha più trame e si confonde con la storia di Linus, un hacker che ha risolto da solo problemi che solitamente richiedono squadre di programmatori che lavorano per mesi, e con la storia di internet. Linux è una rappazzatura enorme di codice che definisce una nazione cibernetica in rapida espansione, la Comunità strettamente collegata di coloro che lo fanno e lo utilizzano. Ad unire questi programmatori è la spinta per dare vita al sistema operativo più grande del mondo, più potente di ogni Unix commerciale, in grado di funzionare praticamente su qualsiasi architettura ed infinitamente personalizzabile. Inoltre è un SO che è praticamente identico al cavallo di battaglia Microsoft, Windows NT, con multitasking reale, memoria virtuale, librerie condivise, sistema di rete TCP/IP, ed altre funzionalità avanzate. Molti vedono Linux come l'unica alternativa alla monocultura Microsoft, oppure come la prova dell'ideale che afferma la necessità di avere una scelta digitale.

Ora Linux si trova ad un punto critico. Anche se il SO liberamente distribuibile può non arrivare mai a trovarsi sulla strada di Bill Gates, Linux deve guardarsi la fiducia del settore affari, specialmente se intende salire sullo stesso treno di Microsoft. Tuttavia l'indipendenza è una fonte di feroce orgoglio per i programmatori di Linux, che temono che il loro SO possa trasformarsi in un ennesimo onesto freeware come Mosaic, spazzato via da un prodotto commerciale azzeccato come Netscape. In altre parole, la stessa etica degli hackers che ha creato Linux potrebbe impedirne la relativa espansione, rendendo Linux la storia mai completata del più grande software mai esistito.

Se Linux fosse un poema epico, ed alcuni (non solo in Finlandia) tale lo ritengono, il kernel costituisce l'inizio di questo poema epico nella città di Helsinki. Quasi come se una sorta di hacker cosmico fosse stato al lavoro sul codice sorgente della vita, la casa di Linus è a 10 minuti di passeggiata dal centro città, in una strada che si chiama Kalevagatan, ossia via di Kalevala, più o meno.

²Il testo riportato in questa sezione è l'adattamento di un testo di Glyn Moody, tradotto da Massimo Nuvoli. Versione originale al sito <http://linux.nuvoli.to.it/>

Kalevagatan offre un'insieme caratteristico di abitazioni del secolo scorso e di abitazioni moderne. Un tram periodicamente percorre l'ultima parte della via giù fino ad arrivare al Hotel Seaside, oltre cui si trova il mare grigio che circonda la città dai tre lati. Linus vive con sua moglie in un edificio che sembra costruito apposta per le necessità degli studenti dell'università. File di biciclette giacciono allineate sotto le scale.

Linus stesso sembra più uno scolaro che uno sciamano: altezza media, capelli marroni chiari ed occhi blu che guardano costantemente attraverso gli occhiali con lenti rotonde e montatura colorata. Soltanto le sopracciglia, che sono marcatamente folte e scure, sono in contrasto con l'effetto generale di un viso dall'aspetto giovanile.

Un numero abbondante di libri occupa i muri del suo appartamento. Il posto è riempito con pitture e altri ornamenti, le tende sono piuttosto strane, un cocodrillo essiccato divide due poltrone e ci sono anche due gatti oltre ad alcuni computer, tre PC, un Power Mac e tre micro con processore Alpha, prestati a Linus dalla Digital, che discretamente osservano ai bordi della stanza. Uno degli elementi più interessanti è notevole: c'è un cavo che collega i calcolatori ad uno zoccolo del telefono. Questo cavo è una linea dedicata a 256Kbit verso Internet, installata e pagata a Linus da un Provider Internet locale, come piccolo segno di gratitudine nei confronti dello sciamano di Linux.

Linux, in fondo, non è una mossa vincente intenzionale, ma un processo incrementale, una combinazione di esperimenti, di idee e di pezzi molto piccoli di codice che si sono uniti gradualmente in un insieme organico. Molti anni formativi di Linus sono stati spesi nella programmazione a basso livello su di un Sinclair QL, uno strano calcolatore britannico lanciato nel 1984, che ha avuto molti difetti e una virtù: si trattava di un vero sistema di elaborazione multitask che consentiva anche ardite sperimentazioni. Ma l'evento chiave che ha condotto a Linux è avvenuto durante l'autunno del 1990, quando Linus ha iniziato un corso di UNIX all'università di Helsinki. Linus finalmente ha ottenuto una laurea in informatica presso questa facoltà questo febbraio passato. Alla fine dell'estate l'università aveva installato un MicroVAX con sistema operativo Ultrix. Ultrix è uno delle molte versioni di UNIX, un sistema operativo che domina nei dipartimenti di scienza e di ingegneria delle università, e che è usato ordinariamente dalle aziende di un certo livello per lavori importanti. Anche UNIX è collegato strettamente alla storia di Internet, infatti fa ancora funzionare la maggior parte di siti. Creato nei laboratori Bell di AT&T durante il 1969, UNIX si è sviluppato negli anni 70 come lavoro di passione, prodotto da hackers devoti come Ken Thompson, di Dennis Ritchie e Bill Joy. Nel 1993, AT&T ha venduto UNIX a Novell, che lo ha venduto a Santa Cruz Operation nel 1995. Oggi varianti sono disponibili da aziende quali SCO, IBM, Digital, HP e Sun, con l'unico effetto di un mercato altamente spezzettato, aspetto che lo rende sempre più vulnerabile nei confronti del sistema rivale di Microsoft Windows NT.

Ma durante il 1990, Linus è stato occupato da problemi più concreti: l'hardware della sua università non poteva fare fronte a più di 16 utenti alla volta. Linus stesso ricorda che era necessario aspettare in coda per ottenere ad un terminale.

Uno dei libri del corso era Sistemi Operativi Disegno e Implementazione di Andrew Tanenbaum: che forniva una guida ad una sorta di piccolo UNIX chiamato Minix. Linux afferma che quello è stato il momento in cui ho deciso di prendere un PC. Fino allora aveva resistito e lui stesso spiega perché se avessi preso un PC, avrei ottenuto una architettura scarsa con un sistema operativo quale l'MS-DOS scarso e non avrei imparato nulla

Minix era molto limitato, progettato semplicemente per insegnare i sistemi operativi, spiega Tanenbaum, attualmente professore di informatica all'università Vrije di Amsterdam. Ciò nonostante, dice, Minix ha toccato il tasto giusto: due due mesi dopo l'uscita della prima versione del 1987, esisteva un newsgroup in proposito con oltre 40.000 utenti.

Tuttavia, molti desideravano avere più possibilità. Stavo ottenendo centinaia di messaggi di posta elettronica al giorno, che chiedevano di aggiungere questa e quella caratteristica dice Tanenbaum molta gente è restata frustrata dal mio negare ripetuto.

Linus cominciava allora a sperimentare da solo, usando Minix come struttura per sviluppare un nuovo programma. Ho costruito due processi, gli ho fatto scrivere sullo schermo ed ho anche utilizzato un temporizzatore per passare da uno all'altro, ricorda un processo scriveva A, l'altro scriveva B, in modo che alla fine si vedeva AAAA, BBBB e così via

Linus dice che non ha mai avuto l'intenzione di creare un kernel, la parte di un sistema opera-

tivo in cui vengono svolti realmente i controlli e le elaborazioni. Invece, una necessità puramente pratica di leggere i newsgroup di USENET lo ha guidato alla modifica di quei primi due processi insignificanti. Ad un certo punto ricorda mi sono accorto, con stupore, di avere raggiunto quelle funzionalità.

Linus era stato un hacker atipico fin dal principio: durante l'adolescenza aveva subito programmato a fondo un Commodore VIC20, usando il linguaggio Assembler sia perché non sapeva che erano disponibili altri strumenti sia perché sembrava naturale fare così.

Nel 1991 nasceva la necessità di un programma semplice di emulazione terminale per accedere ai newsgroup. Così Linus iniziava e ne scriveva uno basato sul suo sistema a due processi. Come dice Linus fare quella cosa fu semplice tanto quanto poteva essere banale cambiare le A e le B in qualcos'altro. Un processo sta leggendo dalla tastiera e trasmettendo al modem che allora collegava il calcolatore dell'università e l'altro sta leggendo dal modem ricevendo il flusso di news e trasmettendo allo schermo.

Ma c'era qualcos'altro di cui Linus aveva bisogno: i drivers. Un driver funge da interfaccia del software fra qualche cosa di centrale (come il kernel) e qualche cosa di periferico (come una tastiera, uno schermo, o un modem). Si potrebbe inserire questa interfaccia direttamente nel kernel, così allora sarebbe necessario riscrivere il kernel ogni volta che si desidera utilizzare uno schermo più grande, una tastiera differente, o un modem più veloce. È molto meglio mettere esternamente il programma di interfaccia in un driver: per utilizzare un nuovo schermo o tastiera, basta scrivere solo il driver ed si può usarlo con il vecchio kernel, che continua a funzionare perfettamente.

Durante l'estate del 1991, appena sei mesi dopo avere ottenuto il suo primo PC, Linus scopriva che era indispensabile gestire i documenti. Ma prima di potere leggere e scrivere su un disco, ricorda, io dovevo scrivere un driver per il disco. Poi dovevo scrivere un sistema di archiviazione per i documenti in modo da potere leggere anche gli archivi di Minix, e per potere scrivere e leggere i documenti da trasferire, spiega, come se allora banalmente fosse stata l'unica cosa ragionevole da fare. Quando c'è la commutazione tra processi, un file system ed i driver per i dispositivi, c'è UNIX o almeno il relativo kernel. Linux era nato.

Questo sistema da principiante sarebbe stato di breve durata se Linus non lo avesse annunciato nel newsgroup di Minix. Quasi immediatamente dopo la spedizione del suo messaggio arriva una offerta di spazio su un server FTP presso l'università di Helsinki, permettendo alla gente di caricare la prima versione pubblica di Linux. "Linux era il mio soprannome del lavoro", dice Linus, "ma se realmente lo avessi usato come quello ufficiale, la gente avrebbe potuto pensarmi come egomaniaco e non lo avrebbe considerato seriamente. Così scelsi questo nome un pò strano: Freax. Ma non era il nome giusto" infatti Ari Lemmke, che faceva funzionare il sito FTP iniziale, non gradiva il nome Freax, e usava preferibilmente proprio il soprannome del lavoro.

A gennaio del 1992, soltanto 100 utenti o poco più stavano usando Linux, ma fornivano un battesimo in linea critico. Quegli uploads e le osservazioni iniziali erano cruciali. Specialmente importante erano i pezzi di codice trasmessi dai colleghi hackers a proposito di problemi che venivano riscontrati con il codice. Per caso Linus aveva inciampato in un Karelia in linea e stava quasi per cominciare a radunare la rappezzatura del suo Kalevala (poema epico finlandese). Ognuno, dovunque sulla rete, poteva ottenere gli archivi di base di Linux. La posta elettronica permetteva di commentare ed offrire i miglioramenti, mentre gli USENET fornivano una tribuna di discussione. Cominciando come il prodotto di una mente, Linux stava trasformandosi in una tappezzeria, un movimento di hackers simili.

Il kernel degli apprendisti sviluppatori di per sé non è più in uso, anche se è stato migliorato costantemente attraverso aggiornamenti trasmessi da alcuni hackers. Il vero motivo per cui Linux ha avuto un successo così spettacolare è che tutto il resto che era necessario per creare un SO completo allora era lì ad aspettare.

Questi programmi in attesa facevano parte del progetto GNU della Free Software Foundation. La domanda ricorrente è che cosa è GNU? GNU non è UNIX. Si tratta di un progetto iniziato nel 1984 da Richard Stallman come reazione contro alcune regole draconiane imposte agli utenti del software dai fornitori.

Lo scopo di GNU era di scrivere una versione completamente libera di UNIX, kernel ed elementi collegati, cioè, di dare agli utenti la libertà di condividere e cambiare software, senza aggiunge-

re limitazioni e senza imporlo ad altri. Con il kernel di Linux, dice Stallman, il software libero disponibile portava fino ad un sistema completo.

Piuttosto di aspettare che qualcuno scrivesse e progettasse le applicazioni specificamente per il suo sistema operativo, Linus allora decise di costruire Linux esattamente secondo le regole degli applicativi esistenti GNU. Non ho mai portato un programma dice Linus, ho sempre modificato il kernel per farlo funzionare con quel programma. Linux non è mai stato il motivo principale di nulla, i programmi applicativi sono sempre stati la ragione di tutto.

Un metodo così pragmatico di concepimento ha portato Linux ad ottenere, quasi in una notte, una interfaccia grafica simile a Windows, cosa indispensabile per una più larga accettazione. (fino ad allora Linux era stato controllato con oscuri comandi digitati come testo ad un prompt, più o meno come il DOS): La GUI è stata fornita dal progetto XFREE86, un gruppo senza scopo di lucro che fornisce software libero per le versioni PC del sistema grafico XWindows.

Linus stesso ha adottato lo schema standard di licenza GNU chiamata “copyleft”. La general public license, o GPL, permette che gli utenti possano vendere, copiare e modificare i programmi protetti, che possono anche essere coperti da copyright, ma che le modifiche stesse siano anch’esse liberamente vendibili copiabili e soprattutto modificabili. Anche il codice sorgente delle modifiche deve essere liberamente disponibile.

La GPL ha dimostrato di essere una forza poderosa per il successo di Linux. In primo luogo ha stimolato il nascente settore commerciale di Linux. Anche se Linux è liberamente disponibile su Internet, coprire un insieme di CD-ROM per pochi dollari è generalmente più economico di trasferire centinaia di Mbyte di codice, ed è anche più rapido. La GPL ha dato anche ai programmatori un motivo supplementare per associarsi allo spirito essenzialmente filantropico del movimento di Linux. La licenza ha assicurato la distribuzione gratuita del lavoro di queste persone, ci sono però stati casi negativi di pezzi di codice sfruttati o inglobati in prodotti di organizzazioni commerciali senza scrupoli.

In un certo senso la GPL ha fornito una specie di costituzione scritta per la nuova comunità degli hackers di Linux. La licenza ha affermato che era giusto usare o incorporare all’ingrosso il codice della gente, proprio come Linux, e perfino che era corretto fare denaro in qualche modo (gli hackers sono esseri umani e debbono anche mangiare...). Tutto ciò è stato concesso ma senza deroghe alla legge fondamentale dell’hacker: il codice sorgente deve essere liberamente disponibile per essere ulteriormente migliorato.

Nel mese di marzo 1994, Linux 1.0 ha iniziato a funzionare, quasi come una dichiarazione formale di indipendenza. Allora la base di utenza era già notevole e il gruppo di sviluppo del kernel ben nutrito. Uno dei mille o più documenti che compongono Linux si chiama Credits. In esso sono scritti nome, indirizzo e contributo portato allo sviluppo di Linux. La lista contiene più di 100 nomi, sparsi geograficamente in tutto il mondo. Caratteristica quasi unica per un progetto fatto da hackers, Linux è accompagnato da enormi e completi archivi di documentazione, FAQ, HOWTO, ed archivi generali di help (un esempio è l’LDP Linux Documentation Project).

La crescita della squadra di sviluppo ha rispecchiato lo stesso organico, non si può dire caotico, sviluppo di Linux. Linus ha cominciato a selezionare ed incaricare alcuni hackers che Michael K. Johnson chiama “alcuni tenenti di fiducia” da cui ottenere parti di codice sicuro, e di cui potersi fidare. I tenenti erano più o meno incaricati di gestire parti relativamente grandi del kernel.

Il metodo Linux è ingannevolmente semplice. Tutti gli hackers sono liberi di lavorare in qualsiasi caratteristica aggiuntiva o miglioramento. Anche durante le fasi iniziali, il nuovo codice è scaricabile liberamente, per essere provato e criticato: la fase di beta-test non è un tentativo di trovare i difetti più importanti prima del rilascio del software, diventa una fase importante del processo. Quando tanta gente lavora allo stesso codice, possono competere o addirittura unire gli sforzi: se competono il codice migliore vince una competizione di tipo Darwiniano.

“La comunità Linux e del software libero può essere concepita come puramente meritocratica”, dice Marc Ewing, uno dei fondatori della Red Hat Software nel 1994, casa che vanta una delle distribuzioni Linux più popolari. Le persone in un gruppo tradizionale di sviluppo, vengono assegnate a lavori su cui non hanno conoscenza specifica, o su cui si pensa possano agire meglio” .

Bruno Haible, che ha contribuito al codice per la gestione della memoria di Linux, esprime

questo concetto in modo ancora più chiaro: “ Se l'autore principale non migliora più il suo codice, sicuramente qualcun'altro è disposto a farlo”.

Questa situazione di assoluta libertà ha permesso a centinaia di migliaia di utenti di usare Linux su decine di migliaia di configurazioni hardware: Linux supporta ogni cosa a partire dall'Intel 386 fino al Pentium Pro, le piattaforme basate su Alpha (il RISC di Digital), SPARC (il RISC di Sun), MIPS (il RISC di Silicon Graphics in fase di sviluppo), e MkLinux (una versione di Linux che funziona su macchine Intel e PowerPC).

Gli utenti, nello stesso modo, via Internet, hanno mezzi veloci per comunicare ogni problema alla persona che meglio conosce il programma: l'autore. Questa cosa può essere un grande vantaggio oppure un macroscopico difetto. Mentre gli hackers seri potrebbero gradire di avere tête-à-tête con altri sviluppatori di codice, la maggior parte di utenti normali desidera solamente risposte veloci alle domande. Fornire un servizio di assistenza certo per Linux ha potuto aiutare in modo determinante un fornitore commerciale nel portare Linux alle masse.

Linux, naturalmente, non è cresciuto solo utilizzando Internet per lo sviluppo distribuito e per il ritorno di informazioni dagli utenti. la differenza principale, dice Eric Youngdale, uno dei programmatori che ha fatto parte del team che ha sviluppato il driver SCSI di Linux, “ era semplicemente Linus”. Lo sviluppo libero ed autoregolato è una cosa ottima, ma senza la giusta persona che agisce da perno, queste risorse potevano solo essere disperse. Linus è sempre presente in tutte le fasi del metodo di programmazione che ha creato. Tuttavia non interviene quasi mai, e in qualche modo ha comunque risolto tutti i problemi.

La selezione automatica dei programmatori per lavorare in zone di maggiore conoscenza, e la possibilità del sistema di espandersi in modo infinito delegando mansioni in questo modo naturalmente distribuito, ha prodotto altri benefici. sono impressionato dalla velocità con cui Linux ha raggiunto funzionalità che molti prodotti commerciali hanno avuto con anni di sviluppo, afferma Jon Maddog Hall, senior marketing manager del Digital Unix Software Group, e direttore esecutivo di Linux International senza scopo di lucro.

Effettivamente la rapidità degli aggiornamenti è vertiginosa: già dai primi giorni, tipicamente ogni settimana è disponibile una serie di modifiche. Ma parallelamente c'è sempre una distribuzione stabile della versione che viene aggiornata solo quando le nuove caratteristiche sono state esaminate completamente. Linux generalmente procede con versioni in punti 1.1, 1.2 e via così. Esiste anche un più complesso sistema di sub versioni come 1.1.12. Quando c'è un salto abbastanza grande di funzionalità allora gli sviluppatori si muovono verso la versione successiva, un processo normalmente presieduto da Linus.

Questo sistema di sviluppo a due percorsi paralleli ha portato Linux ad essere probabilmente più avanzato e stabile di qualunque altra versione di UNIX. Linux sta ora entrando in una era di puro sviluppo e non più di semplice aggiornamento dice Jacques Gélinas questo perchè il grande numero di sviluppatori e testatori fornisce un vantaggio incredibile a Linux, rispetto a qualunque altro sistema operativo. NT incluso.

Phil Holden, che è product manager Microsoft per Windows, sembra non essere eccessivamente interessato: “ abbiamo una squadra di sviluppatori di talento che rendono sicuramente NT il più potente, flessibile e facile da usare”, afferma.

Ma stare davanti si sta dimostrando sempre più difficile per Microsoft. L'ultima versione di Linux, la 2.0, offre i 64bit (mentre NT e molti Unix sono ancora fermi a 32bit), sistemi multiprocessore che permettono di inserire una serie di microprocessori in un sistem, e il supporto di rete più avanzato di qualsiasi altro sistema operativo.

Un vantaggio relativo alla struttura dello sviluppo di Linux è che i problemi di sicurezza vengono affrontati e corretti tipicamente in tempi più rapidi dei fornitori commerciali. Per esempio, quando l'assalto chiamato “ Ping of Death” basato su messaggi multipli a basso livello ha fermato parecchi sistemi operativi in tutto il mondo, una rapida modifica ha permesso di ovviare al problema in Linux nel giro di poche ore. “ Qualcuno ha suggerito una modifica apposta per Ping of Death” , ricorda Alan Cox, autore della modifica, “ così io ho solo dovuto metterla giù e spedire l'aggiornamento” . Gli utenti di altri sistemi operativi hanno dovuto sudare ben più a lungo per avere finalmente risolta la vulnerabilità.

La semplicità del codice, le qualità e la tecnica pura di Linux sono indiscutibili, ma poca gente

al di fuori di una stretta minoranza ne ha sentito parlare. Linux può facilmente sembrare qualcosa di interessante ma comunque un fenomeno marginale. I fatti suggeriscono che stia avvenendo esattamente l'opposto.

Poichè Linux può essere scaricato da numerosi siti su Internet, e gli utenti sono invogliati a passarsi i CD-ROM, è impossibile determinare con precisione quanta gente lo usa. Basandosi su una miscela tra i dati provenienti da sistemi di registrazione volontaria e una ricerca di mercato, tuttavia, il distributore di Linux Red Hat stima che esistono tra 3 e 5 milioni di utenti nel mondo (ai tempi in cui fu scritto quest'articolo!!, n.d.a.). Questi numeri sono sottostime del reale mondo di Linux, secondo Harald Alvestrand, coordinatore del progetto Linux Counter. Si valuta che "i più di 46.000 utenti che hanno scelto di registrarsi sono meno del 0,5 per cento del numero totale di utenti Linux. , affermazione possibile facendo un confronto fra i dati del Linux Counter e i sistemi che sono ritenuti funzionanti come server Web. I calcoli parlano di una percentuale di registrati dell'ordine dello 0,5 per cento. Questo indica che ci sono circa 9 milioni di utenti Linux."

A prescindere dai numeri reali, non è errato pensare che Linux possa emergere come il secondo sistema operativo dopo Windows, specialmente in un momento in cui Apple non ha ancora preso una decisione certa sul suo futuro. Tuttavia l'enorme base installata di Microsoft e la vasta gamma di applicazioni disponibili è molto improbabile che possa essere sorpassata, a prescindere da quanti credono nella magia di Linux. Linus stesso non si illude ed utilizza strumenti come Microsoft PowerPoint, però funzionante sotto Linux con uno dei numerosi emulatori Windows disponibili.

Ma l'importanza di Linux si trova non nella dimensione della base installata, ma nel dove gli utenti si trovano. Secondo Alvestrand Sono rappresentati più di 120 paesi e Linux è una potenza reale nei paesi meno sviluppati, con tassi di crescita persino più alti di Internet.

Vista il numero crescente di utenti che lavorano in collaborazione, l'odierno Linux è sempre meno un programma unico e sempre più un insieme di centinaia di pezzi di codice dei vari hackers. I paralleli con il Kalevala stanno colpendo. Il racconto epico Finlandese fu creato a metà del secolo scorso quando un ufficiale sanitario del distretto, Elias Lönnrot, viaggiando attraverso la Karelia raccolse le righe della poesia da molte sorgenti diverse, scarti di una tradizione orale antica, e le mise insieme. Quando Linux è apparso la prima volta era composto da circa 10.000 righe di codice. Ora il solo Kernel ha quasi un milione di righe, con molti milioni di programmi dipendenti che compongono una distribuzione completa di Linux.

Ammesso che Linux è immensamente potente, libero e realmente aperto, ci si potrebbe aspettare di vedere le aziende più attente ai costi correre velocemente per abbracciarlo, per ottenere con pochi soldi e senza dipendenze paralizzanti dal fornitore che il metodo Microsoft implica. La realtà è così, ma di nascosto. La parola freeware è ritenuta sporca nella maggior parte delle aziende. E certamente non qualcosa su cui basare il funzionamento di sistemi critici.

Bryan Sparks guida Caldera, un'azienda che spera di ricavare il proprio pezzo di mercato con OpenLinux. Caldera è stata fondata in collaborazione con Ray Noorda, l'ex amministratore delegato di Novell ed uno dei critici del comportamento di Microsoft. Non c'è stata rottura tra le due aziende, Caldera ha archiviato una causa antitrust contro Microsoft concernente una parte di codice DOS presumibilmente ricavata dal prodotto DR-DOS posseduto da Caldera. Sparks vede Windows NT come reale competitore ed afferma: "Se state vendendo un Sistema Operativo con Microsoft avete a che fare con un gorilla da 2000 libbre". Ma, nota Sparks, Caldera punta decisamente sui sistemi piccoli e medi, tipicamente quelli con meno di 50 server ed afferma che Caldera non punta a compagnie di primo piano come la Ford Motor Co., anche se qualche sistema è stato venduto anche lì, ma a compagnie con i piedi per terra .

OpenLinux costa dovunque da 60\$ a 400\$, ma Sparks ancora è impegnato a calmare i frequenti timori dell'utente su un prodotto soprattutto considerato come un progetto di un gruppo di hackers. "C'è stato un tale impiegato presso una grande compagnia che mi ha chiamato e detto Stiamo usando Linux per i gateway e altre cose per collegare tutti i nostri siti internet. Il mio direttore ha scoperto che stavo usando Linux ed ha affermato che nella sua azienda il software gratuito non doveva essere usato. Puoi aiutarmi?" La mia risposta è stata "Puoi scommetterci, Linux è libero ma noi siamo un fornitore reale, se avete dei problemi saremo pronti a risolverli."

Marc Ewing di RedHat afferma che la sua azienda ha già molti clienti importanti, compresi

NASA, Disney, Lockheed Marti, Industrial Light&Magic, General Electric, Ernst&Young, Price Waterhouse, UPS, Nasdaq, IRS e Boenid come le più importanti università Americane.

Gli scettici affermano che Linux non entrerà mai nel mercato a causa della relativa mancanza di applicazioni desktop. Qualche tempo fa questo poteva essere vero. Gli hackers, Linus compreso, generalmente non sono interessati a scrivere applicazioni. Mnemonic, un progetto Linux per un browser, è una eccezione recente. Ma ora che il mercato di Linux è in espansione, le aziende commerciali stanno colmando la lacuna. Applix e la Star Division offrono entrambe delle suite integrate di programmi potenti, per i testi, i fogli di calcolo la grafica e via così. Caldera ha messo insieme un CD comprendente prodotti di nomi famosi come Corel (CorelDraw per Linux da aggiungere al già esistente WordPerfect) e la compagnia tedesca Software AG (per il database Adabas D).

Altre grandi aziende che hanno facilitato o portato direttamente le proprie applicazioni sotto Linux sono Netscape (con Navigator e i web server Fast-Track), Adobe (Acrobat Reader), più aziende come Marimba (Castanet). Anche Microsoft sta uscendo con una versione per Linux dello standard per la computazione distribuita, DCOM (Distributed Component Object Model, conosciuto anche come rete OLE, semplicemente un sistema per collegare oggetti software distribuiti attraverso la rete). E il supporto di Java in Linux significa che tutti gli applicativi prodotti con questo linguaggio indipendente dalla piattaforma immediatamente si andranno ad aggiungere all'insieme dei programmi per Linux.

Tuttavia vale la pena di considerare alcuni ostacoli. Uno è il modello distribuito di sviluppo che può venire a mancare. Eric Youngdale confessa che Più passa il tempo più diventa difficoltoso per gli sviluppatori comunicare a causa dell'enorme quantità di messaggi sulle liste. Questo può portare all'apertura necessaria ai programmatori principali per portare Linux definitivamente nel settore commerciale.

Anche se il processo di sviluppo continua ad inserire i nuovi miglioramenti in Linux, questo in se può anche essere alla fine un male. Uno dei grandi vantaggi di Linux è stata la relativa piccola grandezza, ma la tentazione di aggiungere più e più caratteristiche può portare a fare ingrassare troppo il sistema, ad avere programmi enormi che provano a fare troppe cose marginali.

Ed allora c'è Microsoft. Con Bill Gates che acquisisce costantemente dalla maggior parte dei sistemi operativi che si trova intorno, può Linux sperare a lungo termine di competere contro Windows NT? Una ragione è che Linux 1.0 è nato più o meno completamente formato, come il mondo di Väinämöinen, e non si è evoluto lentamente ed in modo disordinato partendo da fasi primitive. NT per contro è stato destinato ad essere l'incarnazione robusta di Window 3.0, la terza ripetizione di un programma che è nato come tentativo di rendere MS-DOS utilizzabile. MS-DOS, a sua volta, è stato basato fortemente su un sistema operativo semplice che Gates ha comprato dalla Seattle Computer Products nel 1981, proprio prima che IBM lanciasse il suo PC, e l'MS-DOS.

Linux, al contrario, è stato capace di inglobare tutti i successi ed i fallimenti passati di Unix. Questa è un vantaggio forte di Linux contro NT. "Utilizzando la lunga storia di milioni di ore uomo di sforzi gratuiti che hanno formato Unix, afferma Dick Pountain, giornalista editore della rivista Byte, le risorse che stanno dietro a Linux possono essere considerate più grandi di quelle che Microsoft ha impiegato per NT".

Tuttavia, Linux deve ancora arrivare all'importante traguardo di guadagnare la fiducia e confidenza nel settore affari. L'esempio positivo di Caldera potrebbe essere un inizio. La domanda è: se è possibile mediare con successo tra la metodologia di sviluppo di Linux e le richieste del mondo affari, e può questo metodo essere scalato per raggiungere le macchine Microsoft?

Arriva Linus 2.0, improvvisamente, alla fine del 1996, proprio mentre si intravede una congiuntura favorevole al progresso per portare Linux a livelli ancora più alti, ed un annuncio straordinario. Linus allora stava lasciando l'università di Helsinki per andare a lavorare in una azienda americana pressochè sconosciuta chiamata Transmeta, con sede in California a Santa Clara. Molti all'inizio si sono domandati se Linux fatalemente sarebbe restato ferito. Ma i maggiori sviluppatori ed i fornitori commerciali ora sembrano ragionevolmente sicuri che Linux ora abbia abbastanza inerzia sia di moto che di strutture per potere andare comunque avanti.

Cosa va a fare Linus da Transmeta? La risposta è Non posso dirlo, ma è scritto nel mio contratto che sto facendo a tempo parziale Linux.

Benchè sia ufficiale che Linus ha passato i sette ultimi anni all'università di Helsinki a lavorare alla sua laurea, nella realtà ha dedicato molto di quel tempo proprio a Linux. Lui e tutti gli altri ben sapevano che la tesi (sul porting di Linux) sarebbe stata finita, e con la tesi lo stipendio da ricercatore. Allora sarebbe arrivata la domanda perenne del mondo degli hackers: Come si può vivere nel mondo reale?

Alcuni, come Matt Welsh, rimangono in ambito accademico; altri come Hannu Savolainen, ora vendono le versioni commerciali del loro lavoro su Linux., in questo caso driver per schede audio. Mettere su una propria azienda sarebbe stata una possibilità, ma Linus non era interessato: "A questo punto non desidererei il lavoro di ufficio", poi concedendosi un pensiero abbastanza lontano dagli hackers tradizionali "vorrei un pò di soldi", aggiungendo "non è il mio obiettivo primario nella vita".

Fondamentalmente, Linus aveva bisogno di una nuova sfida e candidamente ammette che "se le cose in Transmeta vanno veramente bene, io divento abbastanza ricco continuando a fare ciò che voglio e quando voglio". L'unica cosa che si sa del nuovo lavoro di Linus è che si occuperà di VLSI (Very large-scale integration), circuiti integrati complessi "per cui è necessario anche scrivere del software".

Risulta che Transmeta, è una azienda nuova con a capo Dave Ditzel , scienziato principale del progetto Sun che ha prodotto i processori SPARC , probabilmente l'esempio più riuscito dell'idea di microprocessore RISC . Uno dei maggiori finanziatori di Transmeta è Paul Allen , l'altro fondatore di Microsoft .

Anche se non ci sono informazioni pubbliche (www.transmeta.com offre soltanto informazioni contraddittorie questa pagina c'è ma non è ancora qui), il sito web di Paul Allen accenna qualcosa a proposito di Transmeta che sarebbe coinvolta nella creazione di motori alternativi basati su VLSI per i PC multimediali.

Ma nonostante i timori di qualcuno nella comunità Linux, non è l'azienda Transmeta che sta distraendo Linus, ma un'altra cosa, personale, la genesi. Infatti Linus ha avuto una figlia - Patricia Miranda - che è nata in Dicembre 1996, proprio due giorni prima di incontrarla.

Anche i più realisti vedono più la figlia di Linus che il suo spostamento di Silicon Valley come probabilmente uno dei motivi per vedere cambiare la partecipazione allo sviluppo di Linux. Ci si può lamentare se lo shamano che ci ha portato il mondo ricco di Linux 2.0, l'ultima creatura digitale, ora desidera concentrarsi sulla sua altra creazione, Linux 2.0, l'ultima creatura analogica?

Inoltre, se un giorno futuro Väinämöinen desiderasse rinunciare completamente alla sua direzione della comunità on-line, ora c'è la certezza che qualcuno là fuori è pronto a portare la nazione di Linux in battaglia.

*... Ora le cose vanno così
adesso che c'è un nuovo sentiero
per cantanti più adatti
che conduce per molte strade più agevoli
verso i giovani che crescono
verso la popolazione che cresce...*

1.3 Siti Internet e distribuzioni Linux

Vediamo ora dove reperire Linux ed esaminiamo ora le singole distribuzioni, i problemi connessi all'installazione,...

1.3.1 I siti ufficiali internet

Chi interessato ad installare Linux dovrà per prima cosa farsi una passeggiata ai siti ufficiali e per almeno qualche settimana studiare: questa è la parte teorica del processo che non ha fatto nessuno! è sufficiente digitare linux in un qualsiasi motore di ricerca.

I siti internet sono infiniti. Raccomando questi tenendo presente che l'elencazione è incompleta e frutto di simpatie:

<http://www.libralinux.com/>
<http://www.login.campus.uniroma2.it/linux.html>
<http://www.marche.linux.it/>
<http://www.napoli.linux.it/>
<http://www.ildp.psy.unipd.it/>
<http://stud.fisica.unipg.it/>

È anche consigliabile andare a dare una sbirciatina alle FAQ o iscriversi ad una mailing-list : porgete problemi solo dopo che si sia realmente dinanzi ad una seria questione. Numerosissime le pagine di singoli autori dedicate a Linux, e sempre più frequenti le pagine web con la dicitura qui si usa Linux, simbolo ormai di aristocratica distinzione nei confronti della plebe microsoft. Le vie per ottenere Linux sono due: una, la sola considerata dai puristi, consiste nello scaricare Linux dalla rete tramite FTP, l'altra consiste più banalmente nell'acquistare il cofanetto dei CD. Il CD riferito al sistema operativo è uno solo. Gli altri CD, da 3 a 5, contengono software o altre distribuzioni di Linux o applicativi vari.

1.3.2 Le principali distribuzioni

Perché altre distribuzioni Perché essendosi formato nel mondo del volontariato, universitario in specie, Linux, è stato sviluppato secondo diverse interfacce grafiche e secondo differenze di supporti installati.

Successivamente avendo destato l'attenzione delle case commerciali, si sono create nel tempo varie distribuzioni di Linux, divergenti per la presentazione grafica in ambiente Xwindows, per un kernel leggermente diverso, e per supporti in più forniti a corredo del kernel. I comandi sono però rigorosamente gli stessi per tutte le versioni, ed i files di altre versioni si montano con un software dedicato alien senza problemi da una versione all'altra. Nota bene: a sottolineare quanto le guerre di religione non siano del tutto assenti, alien è il software della Debian, la distribuzione linux per eccellenza, per montare i pacchetti delle altre distribuzioni... aliene, appunto!

Ecco dunque le principali distribuzioni esistenti di Linux.

Red HAT

Si tratta senza alcun dubbio di uno dei sistemi più semplici da installare. Reperibile presso Red Hat Software - 02 56814423. Siamo alla versione 7.2. A mio avviso la versione 7.2, come la 7.1 e la 7.0, non sono affatto stabili. Io consiglio di installare la 6.0 e se mai di procedere in aggiornamento con la 6.2 scegliendo però la modalità testo, e non di partire con l'installazione in interfaccia grafica.

Caldera Open Linux

Facilità di montaggio ed interfaccia grafica un pò arrugginita. Reperibile presso Future Technologies, 0434 209107

S.su.S.E. Linux

Si tratta di una distribuzione commerciale tedesca che si va imponendo sempre più per relativa lità affidabilità. L'installazione, per il neofita, non è complessa. Le istruzioni di installazioni sono in italiano. Reperibile presso S.u.S.E. GmbH, Italsel, 051 320409. Siamo alla versione 7.2

Slackware

Una eccellente distribuzione con notevoli difficoltà di installazione e che necessità di conoscenze tecniche tipiche dell'ambiente Unix.

Debian

Completamente gratuita, è la distribuzione Linux più pura. Se Linux fosse una fede, la Debian sarebbe la sua Chiesa. La commercializzazione è limitata alla sola masterizzazione, ed è l'unica versione Linux che non sia stata ancora intaccata dalla produzione di serie, tanto è vero che è ferma alla versione 2.2. Molto difficile da installare, sconsigliata assolutamente a chi non ha per nulla pratica dell'ambiente, è tuttavia la più stabile in assoluto. Fornisce supporti d'aiuto completamente gratuito. Contiene all'interno della sua struttura il famoso file SAO che gestisce IRAF. Sconsigliata ai neofiti.. Versione 2.2

Mandrake

Distribuzione commerciale molto adatta ai neofiti. Sfruttava all'inizio la base della Red Hat. Con il tempo si è affrancata da essa sino a divenire facile da installare e dotata di una eccellente stabilità. Siamo alla versione 8.1. è adattatissima per i neofiti, ma sconsigliatissima per imparare qualcosa su Linux: praticamente, con un minimo di attenzione è impossibile fallire l'installazione.

Stumpede

Distribuzione recente non ancora sufficientemente testata. Instabile ed Inaffidabile

TurboLinux

Distribuzione piena ancora di bachi.

Corel

Primitiva. Adatta ai neofiti ma non isegna nulla.

EasyLinux

Il nome parla da solo. Non è una distribuzione. Non è nulla e non serve a nulla. Se Pippo sapesse usare un PC questa distribuzione sarebbe per lui. Siamo alla versione 2.4, purtroppo!, perché dovrebbe essere già defunta.

PPC2000

Stesse considerazioni di sopra

1.3.3 Altre ispirazioni al software libero diverse da Linux

Le distribuzioni sopra citate non esauriscono il tema dell'open source. Vorrei ricordare di passaggio sistemi operativi non-Linux, completamente affidabili che si ispirano al principio dell'open source: FreeBSD, OpenBsd e Sco Free 7: queste infatti non sono una versione diversa di Linux. Si tratta di nuove versioni di Unix che si vanno sempre più facendo strada.

Le versioni offrono la possibilità di provare la più aggiornata versione non commerciale di Unix sul PC; sono altamente professionali, potenti ed affidabili, anche se offrono rispetto a Linux minore possibilità di personalizzazione. I software sono, ovviamente, compatibili con Linux, ed sono reperibili anch'esse unicamente presso i giornali (!) a meno di 15 euro, se non si vuole scaricarli dalla rete: in realtà si tratta quindi di un sistema Unix puro. Sconsigliate per il neofita.

In quest'ultima carrellata ho taciuto volutamente la Beos che per me non esiste. Convengo che la scelta di una distribuzione è cosa personalissima, quasi scegliersi un'amante, legata inevitabilmente alla simpatia/antipatia verso una determinata piattaforma.

Si consiglia i *digijuni* di montare, almeno da principio, la Red Hat, versione 6.0, in quanto di gran lunga la più istruttiva da installare, e la più stabile, o al massimo la S.su.S.E. 5.2, o la

Mandrake 7.2, e solo dopo mesi e mesi di esercizio proficuo di passare a versioni più raffinate ed esotiche quali Debian e Slackware.

Capitolo 2

L'installazione

Supporteremo d'ora in poi che si stia installando la versione Red Hat nella sua configurazione 6.X .

La prima cosa da fare è preparare il PC. Se il nostro HD è completamente occupato da una partizione dedicata a Dos o a Windows non si potrà procedere: due scelte sono possibili o acquistare un nuovo HD anche solo da 1 Gb, a patto che se ne trovino ancora di tali dimensioni, oppure ripartizionare l'HD tramite un programma quale Partition Magic o l'utility fips.exe presente nella directory dosutils del CD. Quest'ultima opzione è possibile ovviamente solo se una certa parte del disco è ancora libera. La via migliore sarebbe comunque quella di prendere un vecchio 486 a 66 MHz, meglio un Pentium da 100 MHz, con almeno 800 Mb di HD disponibili e procedere con l'installazione.

Se non repute la spesa necessaria e se volete optare per partizionare il vostro HD o per aggiungerne uno nuovo, tenete presente comunque che il vostro nuovo sistema operativo può convivere tranquillamente anche sul vostro HD partizionato con il sistema operativo che fino ad ora usate.

Grazie ad un particolare file di configurazione lilo, Linux vi chiede (immediatamente dopo che il PC ha esaminato la configurazione di sistema) se optate per Linux o per altri sistemi operativi. Quando create la nuova partizione (suggerisco almeno 800 Mb), non formattatela da DOS, l'operazione è completamente inutile. Chi scrive, per un certo periodo ha avuto ben 5 sistemi operativi diversi su un HD da 2 Gb: Dos, windows '95, NT4, QNX e Linux, e li gestiva tranquillamente tutti.

Tenete conto che 800 Mb, per un utente iniziale possono essere anche troppi: se installate il sistema operativo nella configurazione minima, l'installazione difficilmente supera i 400-500 Mb, e gli altri 600 saranno così disponibili per software aggiuntivo e per esercitazioni.

Fatta la partizione sul vecchio disco o installatone uno nuovo (se installate un nuovo disco vi potete anche risparmiare la fatica della partizione sul nuovo, sempre che intendiate dedicarlo interamente a Linux), qualora il vostro PC non supporti il boot dal CD, bisogna crearsi il dischetto di avvio. Le operazioni sono qui descritte tenendo conto di dover procedere ad installare la RedHat nella versione 6.

Acceso il computer andate in DOS. Inserite il CD con l'installazione di Linux, Red Hat 6.X, ed entrate nella directory " dosutils".

Al prompt di scrivere **rawrite**. Il sistema risponde: **Enter disk image source file name.**

A questo punto lanciate dalla directory **images** il file **boot.img**.

Il sistema risponde: **Enter target diskette drive:**

Rispondete a o b, a seconda del caso. Il sistema risponde: **Please insert a formatted disk into drive A: and pres -ENTER-:**

Il dischetto si comincerà a caricare di dati. Ad operazione avvenuta, lasciate il cd e il floppy nel PC, spegnete la macchina, attendete qualche secondo (non fa mai bene all'elettronica ricevere tensione finché non sia completato il ciclo di spegnimento con una buona scarica dei condensatori), e riaccendete il PC. Adesso state per cominciare. Affinché la vostra prima installazione non

fallisca subito, ricordate che Linux non è il Dos, e pertanto Linux distingue fra lettere maiuscole e minuscole: **Carota** è diverso da **CAROTA**, da **CaRoTa**, da **carota**, da **CarotA**,.....

Quest'informazione è vero che vi servirà più in là, ma alla fine dell'installazione voi dovrete digitare la password che non è per nulla opzionale come in Windows, quindi state attenti e cercate di ricordare quello che digitate., se maiuscolo o minuscolo. Non digitate expert se non lo siete e rispondete alle domande che il sistema vi farà.

Scegliete quindi la tastiera, il dispositivo da cui avviene il montaggio del sistema (nel nostro caso il CD),... Il sistema vi conduce ora al primo passo dell'installazione chiedendovi se volete servirvi dell'utility **diskfruid** o del classico fdisk. Le prime volte che fate l'installazione, finché non vi sarete familiarizzati con il sistema scegliete diskdruid. Vi si presenterà una schermata tipo.

Come vedrete, le singole partizioni non vengono nominate C, D, E,.. ma /hda1, hda2, hda3, e nel caso di un ulteriore disco, /hdb1, /hdb2,...

Le dimensioni delle partizioni vengono determinate dall'utente in questa fase a seconda delle sue necessità e della capacità dell'HD. In questa fase il Disk Druid, l'ottima utility per neofiti della Red Hat per effettuare la partizione, vi chiederà di definire due cose: il file swap ed il mount point.

L'indicazione del mount point viene richiesta dopo la preparazione delle partizioni: finché non l'avete eseguita è inutile digitare F12 (OK), il sistema non va avanti. L'indicazione del mount, che serve per prima cosa per specificare quale sia la partizione di root, avviene ponendo la sbarra / nell'apposita casella indicata appunto mount point quando si costruisce la partizione native. Le dimensioni del file swap sono lasciate all'utente. Sarà egli a scegliere, secondo la memoria RAM a disposizione e secondo le capacità residue dell'HD, le dimensioni del file swap. Per un PC con una RAM di 32 Mb un file swap di 64 MB è più che sufficiente. Linux non fa un ricorso estremo alla memoria come windows, ma talvolta ha bisogno di maggior memoria per certi programmi.

Le dimensioni del file swap evitano l'acquisto di memorie aggiuntive: io ho fatto girare con soli 32 Mb di RAM programmi molto complessi senza apprezzare notevoli rallentamenti nell'esecuzione. Deve essere chiaro comunque che l'accesso all'HD per l'uso di memoria virtuale aggiuntiva è inevitabilmente più lento dell'uso di effettiva memoria RAM.

Le altre domande che il sistema vi porrà dovrebbero essere senza difficoltà di risposta, trattandosi di domande quali la partizione da formattare, il check bad della partizione, ecc. Quindi si sceglieranno i pacchetti da installare scegliendo i suggerimenti del sistema.

Non esagerate con le selezioni perché ovviamente se non siete esperti non sapreste neanche che uso farne di tutti i pacchetti che avrete installato. Similmente non segnate con un asterisco l'opzione Everything: sarebbe inutile. Alla stessa maniera, se marcate con asterisco l'opzione Select individual packages, non esagerate. Tramite l'opzione upgrade potrete procedere a questa installazione in una seconda fase.

Entro 10 minuti l'installazione sarà completata. Terminata l'installazione saranno chieste all'utente alcune informazioni aggiuntive, quali il tipo di monitor, il tipo di stampante, l'ora,... Il sistema grafico XFree86 chiede di definire la configurazione, attraverso il programma Xconfigurator . Il programma esegue una scansione diagnostica alla ricerca della scheda video. Se si tratta di una scheda PCI è molto probabile che venga identificata.

Se la ricerca fallisce, viene richiesto di scegliere un tipo di scheda, o direttamente il server grafico. Successivamente si passa all'indicazione del tipo di monitor. È difficile che si riesca a trovare il proprio modello tra quelli proposti dall'elenco, per cui conviene indicare Custom.

Seguono le normali istruzioni di frequenza orizzontale e verticale. Le frequenze selezionate non debbono superare i limiti stabiliti dalla casa costruttrice del monitor. Potrebbe essere richiesta la selezione del cosiddetto RAMDAC. Se richiesto, nel dubbio si può rinunciare a specificarne il valore. Un punto delicato è dato invece dal cosiddetto Clockchip. Se non si sa di cosa si tratti, è bene non indicare alcunché. Successivamente deve essere selezionata la quantità di memoria a disposizione della scheda video. È importante non indicarne più di quanta realmente presente. Infine, si devono indicare le modalità video, cioè la dimensione dello schermo espressa in punti. Per evitare fastidi inutili, sarebbe conveniente indicare una sola risoluzione per tutti i tipi di profondità di colori.

Avvertenza: se avete una scheda dell'ultimo grido uscita da poco tempo e per la quale Linux non conosce i drivers, niente paura, potete egualmente avviare il sistema digitando al momento

del lancio del file lilo il seguente comando: `linux vga=791`. Con questo comando Linux entra istantaneamente in modalità grafica vga 1024x768.

Le ultimi fasi dell'installazione riguarderanno la password e la configurazione del file lilo.

La **password** è importante per accedere al sistema come root, cioè come utente privilegiato: ricordatevi nel digitarlo che Linux distingue tra minuscole e maiuscole, mentre il file lilo serve solo se sul PC è installato un altro sistema operativo, si può omettere se si usa un PC interamente dedicato a Linux.

Se l'installazione è stata correttamente eseguita, al prompt del computer presentatevi come utente principale digitando root. Quindi digitate la vostra password. Digitando quindi **startx** sarà attivata la potente interfaccia Xwindows. A questo punto cominciate a familiarizzarvi con qualche comando. Se il comando startx non dovesse attivare Xwindows o se vi fosse qualche altro ostacolo all'avvio, riprovate da capo l'installazione. Consolatevi, pochissimi sono coloro che hanno proceduto alla corretta esecuzione di Linux la prima volta!

Al termine della sessione di lavoro, vanno smontati i files installati dal sistema: il Computer non va spento brutalmente pena conseguenze che possono anche essere gravi.

La chiusura della sessione di lavoro avviene con il comando **shutdown -h now**; **shutdown -r now** se si intende rebootare il sistema. Anche il comando halt può essere usato, ma lo shutdown è più sicuro. Lo spegnimento può essere *ritardato* tramite il comando **shutdown -h 20**, dove il numero 20 indica quanto tempo trascorrerà ancora prima che il sistema venga spento. Questo comando è indispensabile in una logica server-client, in quanto notifica agli utenti che stanno per essere staccati e li invita a chiudere le loro sessioni mandando eventualmente agli stessi un messaggio ricompreso fra parentesi quadre [...message].

La pressione contemporanea di CTRL ALT DEL attiva lo shutdown e non spegne brutalmente il sistema.

2.1 Root e client

I sistemi UNIX sono sorti e si sono sviluppati nell'ambito universitario, cioè nell'ambito della ricerca vera e propria, di quella chiamata pura. Per di più sono nati e si sono sviluppati trent'anni fa, quando in giro non c'era l'abbondanza di PC che c'è ora: un server allora rappresentava un investimento tutt'altro che indifferente ed semplice consolle operativa costava \$ sonanti. È per questo che i sistemi Unix risentono, e badate che non è un male, della primitiva impostazione.

Essendo il server uno, ne discende che *uno ne è l'amministratore* e tanti i potenziali client. Ma appunto perché i client sono tanti, appunto perché lo spazio dell'HD è limitato (fate mente a quegli anni quando gli HD avevano dimensioni di decine di centimetri restituendo qualche centinaio di Mb), appunto perché un non esperto può mettere in crisi il sistema con operazioni sbagliate, se ha accesso ai files di sistema, ecco che sorge la necessità che ci sia un amministratore di sistema, una persona che distribuisca fra i vari fruitori lo spazio disco, i tempi e le modalità di accesso, graduandole per di più diversamente a seconda del tipo di utente (es.: professore o studente), una persona che renda accessibili gli applicativi, la posta, una persona che in sostanza dia quelli che si chiamano i permessi.

Non si possono passare intere giornate a rimettere in piedi un sistema costato migliaia di \$ solo perché un neofita inesperto ha compiuto un'operazione sbagliata! La gestione dei permessi (e degli account è fondamentale in questa filosofia. Abilitare l'utente ad operazioni non-dannose è una necessità. Questo comporta alcune restrizioni. L'utente può accedere al sistema, può usare gli applicativi messi a disposizione, ma non possiede alcun diritto di conoscere, ad esempio, quanto spazio libero c'è sull'HD e come sono state formate le partizioni, ad esempio.

Si tratta di informazioni privilegiate cui solo l'utente principale "root" può accedere. Se comprendete questa filosofia vi riederete conto dei tanti *permission-denied* che Linux all'inizio della vostra esperienza vi regalerà di continuo, specie se siete, come è naturale, abituati a windows dove la differenza, quando c'è, è veramente virtuale.

Non agite mai come root, sempre come account, utente cioè, limiterete i danni. Se proprio dovete lavorare come root non fatelo alle due di notte: io ho perso un'intera banca dati per un **rm** *sta per: remove* sbagliato dato a quell'ora. Meditate gente! Meditate!

Capitolo 3

Primi comandi

N.B.: Gli spazi fra i comandi, le lettere maiuscole e minuscole sono significative.

I primi comandi che descriveremo sono finalizzati a leggere e scaricare i primi files ed a configurare il collegamento internet. Per quanto il configuramento di internet non sia logicamente una delle prime operazioni da compiere da parte di un nuovo utente Linux, che ha invece bisogno di impraticarsi sui comandi e sulla diversa logica di sistema, tuttavia ritengo che il dare al nuovo utente la possibilità di operare un collegamento internet dia allo stesso fiducia nelle proprie capacità e la possibilità di iniziare a valutare il nuovo sistema.

Inoltre l'accesso alla rete farà muovere il neo-utente nell'ambiente ideale di Linux, quello in cui è nato ed in cui si è sviluppato. L'apprendistato di alcuni comandi basilari si renderà dunque necessario, e questi saranno esaminati a poco a poco. Qui appresso ne presenteremo alcuni la cui conoscenza è indispensabile per muoversi nel sistema operativo.

- **Comando cd ..** Per prima cosa sarà da andare a vedere le partizioni del nostro HD. Dopo esserci scalati di una directory tramite il comando `cd ..` (notare lo spazio fra `cd` ed i puntini). Scalati di una directory, il passo successivo sarà andare a leggere eventuali altre sottodirectory e i files.
- **Comando ls** Opereremo allora una lettura delle radici e dei files tramite il comando `ls`. Questo comando ci leggerà, indistintamente i files e le radici. Nelle ultime distribuzioni dei desktop i files e le radici vengono evidenziati con colori diversi. Le radici sono inoltre generalmente seguite da una barra: esempio `mieifiles/` è una directory.
- **Comando ls -f** Per leggere soltanto le radici (directory) dovremo usare il comando `ls -f`. In questo modo visualizzeremo le directory come seguite da una sbarra. Ad esempio la scrittura `home/`, indica che `home` è una directory. Per spostarci all'interno della directory `home` dovremo eseguire il comando `cd home`.
- **Comando ls — more** Per leggere i files presenti all'interno di questa directory useremo il comando già detto `ls`. Se i files da visualizzare sono numerosi e non è possibile contenerli in una sola schermata, si usa il comando `ls — more`.
- **Comando ls — less** Anche il comando `ls — less` è utilizzabile, ma esso presenta per il neo utente una maggiore difficoltà d'uso, in specie nell'uscire dai comandi. Abbiamo appreso come spostarci nelle directory e come leggere il loro contenuto.
- **Comando ls -al** Questo comando legge tutto sui files: dimensioni, data creazione, proprietario, permessi,...
- **Comando fdisk** Vediamo ora le partizioni del nostro HD. Per questa operazione ci serviamo

del comando `fdisk`, che è il fratello maggiore dell'utility `diskdruid`¹ che abbiamo usato durante l'installazione per effettuare le partizioni.

Dopo l'esecuzione del comando abbiamo dinanzi a noi due scelte: o digitare **m** per avere un rudimentale help in linea, oppure digitare subito **p** ed il sistema editerà a schermo le nostre partizioni. Ricordate, se per caso siete così previdenti da crearvi un account, che l'operazione `fdisk` non vi sarà accessibile da account, ma solo da root, perché solo root, nella filosofia Unix -come detto sopra- può modificare le partizioni e dare il consenso alla creazione di nuovi utenti che vengono tutti immagazzinati sotto la directory `/home`.

Attenzione! Per uscire digitare **q**. Se avete effettuato -anche involontariamente- delle modifiche, non digitate **w** -per write- perché ogni modifica sarà scritta ed i dati persi! Se vi trovate in difficoltà, potete uscire in qualsiasi momento con CTRL+C.

- **Comando mount** Un'altra ovvia necessità è quella di andare a leggere il contenuto del CD. S'era già detto che in ambienti Unix il Cd va montato. Anche se ormai molte distribuzioni, dalla RedHat alla Suse, alla Mandrake,.. consentono il mount automatico all'inserimento del CD, è bene impratichirsi manualmente in questa operazione per non far sì che, per via di qualche errore, il CD resti incastrato e sia impossibile smontarlo (aprire il coperchio) del CDrom.

I comandi saranno pertanto del tutto particolari e distinti da quelli classici usati nei prodotti microsoft. Apriamo lo sportello ed inseriamo un CD. A questo punto diamo in sequenza i seguenti comandi: a) `mount /mnt/cdrom`, e lo sportello del CD si chiuderà; b) `cd /mnt`; c) `cd cdrom`.

Per leggere il contenuto del CD o le sue directory useremo allora i comandi che già conosciamo: `ls` ed `ls -f`. Potremo anche avere bisogno di leggere un determinato file presente all'interno del CD, come un file `Readme`. Ci occorre quindi un editor di testi.

Per quanto Linux sia generosissimo di editor di testi, essi sono quasi tutti però destinati ad un pubblico adulto e non presentano la caratteristica classica degli editor in ambienti microsoft caratterizzati di solito da quello che tu vedi (a schermo) è quello che appare, cosiddetti editor WYSIWYG.

Data appunto questa complessità iniziale, consiglio, almeno all'inizio, un editor semplice come `joe` o `pico`, `nedit`, solo poi passerete ad `emacs`, il re degli Editor, che tranne il caffè alla mattina, fa tutto. Almeno così si dice in giro!

Per chi ricorda ancora il vecchio `wordstar` e lo trovava semplice, `joe` fa al suo caso. `joe` può infatti essere editato anche digitando `jstar`: in questo caso `joe`, che è un vero camaleonte, assume la caratteristica ed i comandi del `wordstar`, e fornisce un valido aiuto.²

Uscire da un editor in ambiente Unix è cosa abbastanza complicata, specie se si tratta di `vi`, un editor concepito da una mente masochista, per non dir peggio. È per questo che si consiglia l'uso di `joe` o della sua versione `jstar` o di `pico` perché sono reperibili, tramite la pressione di tasti ottimi help in linea che guidano l'utente in maniera infallibile. L'installazione di `joe` va selezionata in fase di installazione del sistema (si scusi la ripetizione di parole). Questo sempre se agite in modalità testuale. Se siete in ambiente grafico Xwindows, potete scegliere editor come `vedit` o `nedit`, o tanti altri che il sistema ha installato. Comunque tenete presente che nelle ultime distribuzioni quasi tutti gli editor agiscono da interfaccia grafica.

- **Comando cp** Potrà essere necessario anche copiare qualche file sul nostro HD. Il comando copia che presenta la seguente sintassi: `cp`.

Per copiare il file carota dal CD alla directory `/files`, sottodirectory di `/home` del nostro HD useremo allora la seguente sintassi: `cp carota /home/files`: in questo caso si suppone

¹Ogni distribuzione, esclusa la Debian, ha un'interfaccia grafica per partizionare i dischi. `diskdruid` è propria della Red Hat.

²`joe` può essere editato dando anche i seguenti comandi: `"jmacs"`, `"jpico"`, `"rjoe"`

di copiare il file carota contenuto in una directory del CD nella sottodirectory /files della directory /home dell'HD.

- **Comando umount:** smonta CD. Quando le nostre operazioni sul CD saranno terminate potremo smontare il CD attraverso una sequenza di operazioni, dopo essere ritornati nella directory principale del CD se, per caso, ci eravamo spostati in qualche subdirectory. I comandi per smontare il CD sono i seguenti: a) `cd ..`; b) `cd ..`; c) `umount/mnt/cdrom`.

Soltanto ora sarà possibile aprire lo sportello del CD. Prima non era consentito nel nome della filosofia Unix che non considera le periferiche, ma vede tutta la macchina nel suo insieme: dal momento in cui il CD è stato montato, esso è divenuto, allo stesso tempo, software di macchina ed hardware di macchina. Tenete presente che non è obbligatorio, né indispensabile assegnare un suffisso ad un file, per quanto questa operazione sia sempre bene farla perché distingueremo i vari tipi di files.

- **Comandi d'installazione pacchetti** Supponiamo che nel Cd Rom che abbiamo montato ci siano due o tre pacchetti che ci interessa installare e che questi non siano compressi nell'installazione effettuata. Se i pacchetti sono compressi, prima di essere montati vanno tutti prima, ovviamente, scompattati ed in seguito montati. La comprensione del fatto se un certo pacchetto può essere istantaneamente montato è immediata. È sufficiente dare, dentro la directory in cui ci troviamo, il comando `ls -al`, e se il nostro nomefile sarà seguito da un asterisco (*), ciò indicherà che il pacchetto è montabile immediatamente mediante l'apposito comando.

Un file da installare si presenta in genere in questa forma:

Messier.01.01.05.file.rpm*

Come si vede, e come si ricorderà da quanto abbiamo detto, Linux fa ampio uso della possibilità di usare nomi di files lunghi. Solo una parte del file è dedicata al nome, le restanti parti sono, più che altro, informazioni per l'utente, il quale è edotto all'istante della versione che sta installando.

Per l'installazione di un file occorre seguire la seguente procedura. Individuato il file che si trova in una determinata directory, ci sono tre possibilità diverse di comando, ognuna delle quali fa qualcosa di diverso dall'altra. Esaminiamole.

1. **rpm -i nomefile.** Questa opzione si limita ad installare il file (il pacchetto più propriamente) senza alcun controllo
2. **rpm -u nomefile.** Questa opzione disinstalla un pacchetto precedentemente installato
3. **rpm -Uvh nomefile.** Questa opzione installa un pacchetto e controlla il software. Se nel corso dell'installazione ci assalisse qualche dubbio amletico o qualche timore di commettere errori, potremo trovare conforto digitando **man rpm**. In questo modo richiameremo le istruzioni del manuale relative all'installazione dei pacchetti. Vedi comunque sotto in sede di riepilogo comandi.³

- **Scompattamento di un file.** È una delle operazioni che più gettano, all'inizio, nello scoramento.

I files compressi di Linux, specie per chi è abituato all'ambiente winzip che fa tutto da sé, sembrano fatti apposta per far desistere dall'apprendimento. Chi scrive più di una volta si è imbattuto in questi problemi e continuamente si è chiesto: ma chi me lo fa fare? Poi la voglia di conoscere ha avuto il sopravvento. Per fortuna. Supponiamo di avere un file da

³L'opzione **man** seguita dall'istruzione di cui si vuole conoscere il valore è un ottimo metodo per imparare i comandi e le opzioni sotto Linux. Fino a poco tempo fa **man** rappresentava, tramite la lettura del manuale in linea, l'unico modo per impraticarsi di questo sistema operativo. Ora la situazione è cambiata, in quanto la letteratura intorno a Linux sta diventando assai vasta, ma la lettura del manuale con cui, occorre dirlo?, bisogna impraticarsi un poco, rappresenta un metodo veramente insostituibile per l'apprendimento di questo sistema operativo

scompattare. Se non siamo in ambiente Xwindows lanciamo startx, entriamo in interfaccia grafica ed apriamo una finestra di terminale (xterm).

Una volta scaricato il file ci occuperemo di copiarlo in una directory come, ad esempio, la directory /home. Ci sposteremo quindi nella directory /home con il comando appreso cd /home, e dando il comando ls verificheremo che il file sia presente nella nostra directory.

A questo punto siamo pronti per scompattare il file col comando **tar xvzf miofile.tar**. Al termine dell'operazione di scompattamento digitando ls -f nella directory /home potremo verificare l'esistenza di una subdirectory /nome_subdirectory con l'esistenza del file scompattato.

Al termine dell'operazione lanciamo il comando ./nome_fileed il programma andrà in esecuzione.⁴

- **Altri comandi e riepilogo comandi basilari. Ulteriore descrizione, Nuovi Comandi**

Nella convinzione di far cosa buona e giusta, si riportano qui di seguito i comandi basilari minimi per prendere conoscenza dell'ambiente.

- **CD. Monta/Smonta**

1. monta (comandi da dare in sequenza) a) **mount /mnt/cdrom**, b) **cd /mnt**; c) **cd cdrom**.
2. smonta (comandi da dare in sequenza) a) **cd ..**; b) **cd ..**; c) **umount/mnt/cdrom**.
3. **Directory (crea, cancella, cambia, visualizza, muove fra le directory)**

- **Crea, cancella, leggi, scala directory**

1. Per creare una directory: **mkdir mia_directory**
2. Per cancellare una directory: **rmdir mia_directory**
3. Per cancellare una directory e le subdirectory in essa contenute, nonché i relativi files: **rmdir -f -r mia_directory/**
4. Per scalare di una directory: **cd ..**
5. Per leggere le singole directory: **ls -f**
6. Se non ricordate piu' in quale directory siete, il comando: **pwd** stampa schermo la directory corrente
7. Se volete muovervi fra le directory il comando: **pushdvi** ricorda dove eravate ed il comando **popdvi** vi ci riporta.

- **Comando cerca file**

1. Cerca in tutti i dischi il file dal manuale.tex: **find / astro**
2. Cerca i files modificati negli ultimi due giorni (48 ore): **find / -ctime 2**
3. Cerca files che hanno dimensioni maggiore/uguale a 1024: **find / -size 1048**
4. Il comando **whereis** effettua una ricerca su alcune directory e restituisce il percorso del sorgente, dei files binari e delle pagine del manuale corrispondenti. Ad esempio il comando **whereis emacs** restituisce i seguenti risultati per la ricerca effettuata: **emacs: /usr/bin/emacs /usr/lib/emacs /usr/share/emacs /usr/man/man1/emacs.1ereis.**
5. Il comando **which** effettua una ricerca sulla directory indicata. Ad esempio il comando **which zmore** restituisce i seguenti risultati: **/usr/bin/zmore**

⁴Non sempre, e neanche per i files che terminano con lo stesso suffisso, l'operazione è così semplice!

6. L'opzione cerca file, nella sua specialità cerca il contenuto di un file può essere attivata anche dal menu di KDE⁵. Se state usando la RedHat, andate sull'icona di gnome e seguite questo percorso: pKDE Menu, Ricerca file

- **Comandi leggi file**

1. Per leggere all'interno di una directory i singoli files: **ls**.
2. Per leggere più files a schermo: **ls — more**.
3. Per leggere gli attributi di un file: **ls -a**.
4. Per leggere i permessi e gli attributi di un file: **ls -al**.

- **Per leggere il contenuto di un file**

Per leggere un file usare un qualsiasi editore, come joe anche nella versione jstar o pico. L'uso di editor come vi è al momento sconsigliato.

- Ad esempio per leggere il file lilo.conf, digitare jstar /etc/lilo.conf.

- **Comando Muove, Copia, Cancella file(s)**

1. Per *muovere* un file: **mv mio_file.astro /home/temp** muove il file dal nome mio_file.astro dalla directory in cui si trova alla subdirectory /temp di /home;
2. Per *copiare* un file: **cp astro /home/temp** copia il file dal nome astro nella subdirectory /temp di /home;
3. Per *cancellare* un file: **rm mio_file.astro** cancella il file dal nome mio_file.astro. Il comando **rm *.astro** cancella tutti i files il cui suffisso termina con astro.
4. Il comando **rm -fr *.*** forza la cancellazione dei files e delle directory. **Attenzione!**: il comando **rm *.*** dato da root (/) cancella tutto l'HD!!!!!!

- **Comando legge partizioni**

1. Per leggere le partizioni dell'HD: **fdisk**
2. Per leggere lo spazio libero e occupato delle singole partizioni dell'HD: **df**

- **Installazione pacchetti e funzioni concatenate**

1. Per installare un pacchetto: **rpm -ivh nome_file**
2. Per installare un pacchetto: **rpm -Uvh nome_file**: a differenza della precedente operazione questa esegue anche l'upgrade di un pacchetto installato (installa anche un pacchetto ex novo)
3. Per rimuovere un pacchetto: **rpm -e nome_file**
4. Per verificare la versione di un pacchetto: **rpm -q nome_file**
5. Per verificare la corretta installazione di un pacchetto: **rpm -V nome_file**
6. Per capire di quale pacchetto è parte un certo file: **rpm -qf nome_file**
7. Per visionare i files contenuti in un pacchetto: **rpm -qlp nome_file**

- **Configurazione hardware**

1. **XF86Setup** permette di configurare schermo, scheda video, tastiera, frequenza schermo, mouse,...
2. **xvidtune** permette di configurare con maggiore risoluzione le proprietà dello schermo.

⁵KDE è una parola ceca che significa *dove*. In realtà KDE è acronimo di K.... Desktop Environment. La lettera "K" non ha alcun significato particolare tranne che essere la lettera... precedente alla "L" di Linux. Siccome questo desktop grafico si adatta bene anche a UNIX, OpenBsd e soprattutto FreeBSD, i programmatori, pur scrivendolo quasi interamente pensando a Linux non l'hanno voluto personalizzare con riferimento a questo sistema operativo

3. **Xconfigurator** crea o modifica il file **XF86Config**.
4. **mouseconfig** configura il mouse
5. **sndconfig** configura la scheda audio... se riconosciuta.

- **Scompattamento file: opzioni “tar”**

1. **tar tvzf nome_file** mostra l'elenco dei file contenuti nel file compresso
2. **tar xvzf nome_file /miadirectory/temp** scompatta il nome_file nella subdirectory /temp di /miadirectory

- **Cose utili da ricordare**

1. Se il sistema si blocca, è sufficiente “buttar giù” lo schermo (senza bloccare brutalmente la macchina) tramite il comando CTRL+ALT+BackSpace: Sono più di sette anni che uso Linux sotto varie distribuzioni, e non ho mai, dico mai, dovuto fare un reset!
2. Tramite i tasti CTRL+ALT+F1, F2, F3, F4, F5, F6, F7 ed F8, è possibile attivare le console virtuali numero 1, 2, 3, 4, 5, 6, 7 ed 8
3. ALT+TAB Scambia fra le ultime due finestre attive
4. ALT + F4 chiude le finestre e pulisce lo schermo. Questo è l'effetto. In realtà si passa al 4° desktop virtuale. Lo schermo si riattiva cliccando in basso, sulla barra degli strumenti, nella finestra in cui ci si trovava. Praticamente questo comando corrisponde ad riduci ad icona, soltanto che riduce ad icona tutto ciò che si trova attivo sullo schermo. Un click e tutto torna a posto.

È ovvio che questi comandi, volutamente pochi, non sono che uno spillo nel pagliaio della marea dei comandi. Essi sono però, a mio modo di vedere, quelli fondamentali da cui a cascata deriveranno tutti gli altri.

Altri comandi l'utente li imparerà a poco a poco, sovente, a sue spese, perché anche errare è una tappa importantissima dell'apprendimento. Chiudiamo la lista dei comandi raccomandando di sperimentare, prima di effettuare lo shutdown, i comandi logout ed exit.⁶

⁶Con questi due comandi si abbandona l'account dal quale si sta operando per passare ad un altro account, a root, o per chiudere il sistema ed effettuare lo shutdown

Capitolo 4

Collegamento Internet

E veniamo ora al tanto sospirato collegamento internet.

Il collegamento ad internet nelle varie distribuzioni è agevolato dalle utility previste sia sotto il KDE che sotto GNOME. Entrambe consentono una configurazione automatica veramente difficile da fallire. Ma supponiamo che vogliamo soffrire, ecco questa è una delle vie per il dolore. Non spaventatevi per i primi fallimenti, e non considerateli tempo perso! È tutto apprendistato necessario ed ineludibile! Per prima cosa collegiamo il modem al PC ed accendiamolo. Quindi facciamo in modo che Linux lo riconosca.

Sempre ricordando che stiamo operando come root, diamo da xterm il seguente comando:

```
ln -sf /dev/cua1 /dev/modem.
```

A questo punto digitiamo **ls -la /dev — more**. Se tutto è andato bene dovrebbe comparire una schermata di cinque colonne.

Scorrendo con la barra spaziatrice deve comparire una riga che mostra alla prima colonna la scritta:

```
lrwxrwxrwx 1 root root data ora modem -i /dev/cua1 .
```

La comparsa regolare di questi output indica che sul PC tutto funziona regolarmente. Ed allora possiamo veramente iniziare. Sempre dalla solita finestra di terminale (xterm) lanciamo il comando **netcfg**.

Vedremo comparire una schermata abbastanza semplice ed intuitiva in cui ci dovremo occupare di riempire alcuni campi. la schermata è composta di quattro sotto schermate: NAMES, HOSTS, INTERFACES, ROUTING.

In Names mettiamo il nome del dominio mioserver.it, ed i numeri per la connessione DNS.

Cliccando su Hosts, Interfaces e Routing sarete guidati in operazioni che non hanno bisogno di ulteriore commento data la loro semplicità. Se il sistema non funziona subito vuol dire che avete sbagliato qualcosa. Accertatevi ad esempio che la porta del modem sia correttamente settata. Per qualche alchimia, che io stesso ho avuto più volte occasione di verificare, spesso occorre spostarsi su Interfaces, cancellare quanto è stato scritto in via automatica, riscriverlo di nuovo.

In realtà non c'è nessuno segreto. si tratta delle informazioni che Linux va a scrivere sul file della directory /sbin che spesso conffigono. Ed i conflitti, si sa, vanno rimossi.

Se il modem lampeggia ed emette suoni abituali indicanti un buon funzionamento, da una finestra xterm lanciate il comando

```
tail -f /var/log/messages.
```

Vedrete tutti i particolari del vostro collegamento. In sostanza vi conviene, come credo avrete capire, aprire piu' di una finestra.

Da una finestra, per esempio, possiamo lanciare il comando **ifconfig**, comando che risiede nella directory /sbin. Questo comando mostrerà che l'interfaccia **ppp0** è collegata.

Il collegamento così descritto è quello facilitato tramite l'interfaccia netcfg. Per chi volesse provare l'emozione di un collegamento totalmente manuale, ecco le istruzioni. Nota: suggeriamo l'utente di non tralasciare questa parte della descrizione, perché essa abitua a sapere cosa la macchina sta facendo dietro il nostro comando o, meglio, cosa le si dice di fare.

Servendoci del solito editor di testi joe e senza entrare in ambiente grafico (se già ci siamo possiamo spostarci sulla consolle n.° 2 tramite la pressione dei tasti CTRL+ALT+F2), digitiamo **joe /etc/resolv.conf**. Se abbiamo già configurato, e corettamente, la connessione internet, deve apparire un file composto di due righe:

```
search mio_provider.it 222.222.222.222
nameserver 222.222.222.222
```

(i numeri sono ovviamente in questo caso casuali) Digitando un nuovo file tramite il solito editor e scrivendo **joe /etc/ppp/options**, deve comparire la parola clock. È buona usanza commentare questa parola mettendo il simbolo# dinanzi alla parola.

Digitando **joe /etc/sysconfig/network-scripts/ifcfg-ppp0** vedremo ora comparire le istruzioni qui di seguito riportate. Se abbiamo fatto in precedenza diverse operazioni e cancellazioni sull'interfaccia di netcfg questo file può anche terminare ed apparire ppp1 o anche ppp2, anziché come ppp0.

```
PERSIST=yes;
DEFROUTE=yes;
ONBOOT=no;
INITSTRING=ATZ;
MODEMPORT=/dev/modem;
LINESPEED=115200;
ESCAPECHARS=no;
DEFABORT=yes;
HARDFLOWCTL=yes;
DEVICE=ppp1;
PPPOPTIONS= ;
DEBUG=no;
PAPNAME=mio:name;
REMIP= ;
IPADDR= ;
BOOTPROTO=none ;
MTU= ; MRU= ;
DISCONNECTTIMEOUT= ;
RETRYTIMEOUT= ;
USERCTL=no .
```

Digitando ancora **joe /etc/sysconfig/network-scripts/ifup-ppp** vi apparirà un lungo file che conterrà tutti gli eseguibili quando si lancia il comando **ifup ppp0**.

Consiglio adesso di andare a dare un'occhiata tramite il solito joe ai seguenti files che si trovano nella directory /etc : hosts.allow; hosts.deny; inetd.conf . Questi files contengono le autorizzazioni e le protezioni per internet. È bene editare il file /etc/hosts.deny e scrivere in una riga il comando **ALL:ALL**. in questo modo internet è un po' più protetto.

Attivato il collegamento internet ci resta che configurare la posta elettronica. Suggestisco di usare direttamente Netscape. Andati in preferences, sotto il menu di edit, occorre riempire con i nostri dati di riconoscimento la categoria identity. Quindi nel campo denominato mailservers metteremo i dati di riconoscimento del server della posta: ad esempio, mail.namemioserver.it. La posta elettronica è così configurata.

Adesso che con l'aiuto di un poco di fortuna abbiamo configurato Internet, la posta elettronica ed abbiamo a disposizione un programma per l'FTP (suggestisco, almeno all'inizio GFTP che si trova nel menu sotto rete, o scaricare l'emulatore Linux del famoso WS.FTP), possiamo muoverci nel nostro ambiente ideale, cominciare a giocare con questo nuovo sistema operativo, scoprirne le potenzialità.

Navigare sotto windows o sotto linux non fa, in ultima analisi, una grande differenza, ma è la coscienza di cominciare a possedere le chiavi di un sistema operativo serio, questo sì che fa differenza!

La configurazione di Internet può essere effettuata anche dal menu di KDE scegliendo l'opzione Internet, KPPP. Ma fa troppo windows!

Capitolo 5

Creazione e Rimozione di un account

La creazione di un account è la prima operazione che viene fatta. Essa non consente di danneggiare il sistema ed ha limitati privilegi. L'utente non può cancellare una directory di root quale, ad esempio /etc, neanche inavvertitamente, perché ne riceve il diagnostico *permission denied*.

Per creare un account bisogna essere innanzi tutto root. Solo il superutente (root) può dare nella filosofia Unix il consenso alla creazione di singoli account. Nella nostra ipotesi noi stiamo operando sempre come superutente (root). Il comando da scrivere è il seguente:

adduser francesco, dove **ad** sta per aggiungi e **user** sta per utente, quindi: aggiungi un utente di nome francesco.

Il passo successivo consiste nell'assegnare una password al nuovo utente francesco. Il comando è:

passwd francesco. Il sistema risponde

Changing passwd for user francesco. Il sistema risponde

New Unix passwd Digitate la passwd e ricordate che non vedete quello che scrivete!!!!!!

Rispondiamo scrivendo, ad esempio, **linuxlinux**. Il sistema chiede la conferma scrivendo:

Retype new Unix passwd, e noi riscriveremo la nostra passwd **linuxlinux**. Da questo momento l'utente francesco potrà usare il sistema senza danneggiarlo!

Se la passwd digitata non è *in regola* con la filosofia Unix delle passwd (intorno alla questione esistono trattati che sovente sconfinano nella più folle filosofia), basterà digitare un'altra passwd che sia gradita al sistema.

Potremmo cioè incappare in un messaggio del genere: **BAD PASSWD : it does not contain enough DIFFERENT characters**. Occorrerà semplicemente ridigitare la passwd in modo che essa sia gradita al sistema. Il successo dell'operazione sarà garantito dal seguente diagnostico:

passwd: all authentication tokens updated succesfully.

Se ora ci sposteremo nella directory /home potremo vedere che c'è una subdirectory chiamata /francesco. È lì che il sistema caricherà tutti i nostri file quando agiremo come utente francesco.

Potremo anche avere in seguito bisogno di togliere all'utente francesco la directory che gli abbiamo creato.

Questo non deve stupire. Voi siete abituati a pensare al vostro PC come una cosa vostra in cui voi potete fare tutto. I sistemi Unix concepiscono tali privilegi solo per il superutente, cioè per root, e danno dei permessi di lavoro che possono anche essere temporanei, perché l'idea base è che la vostra macchina sia in realtà... un terminale collegato in rete.

Per cancellare un utente creato, l'amministratore di sistema (root), cioè voi, darà queste istruzioni:

userdel francesco A questo punto l'utente francesco non può più operare e se prova a collegarsi gli viene risposto **login incorrect**.

Se voi andate nella directory /home noterete però che esiste ancora una subdirectory /fran-

cesco. Perché avviene questo? Perché l'amministratore di sistema (root) per qualsiasi motivo può sospendere all'utente francesco i permessi temporanei, ridarglieli in seguito, senza distruggere i suoi files, al meno per ora.

Se invece si trattasse di un utente che va definitivamente cancellato, assieme ai suoi files, occorre dare il comando: **rm -f -r francesco/**. In questo modo tutti i files vengono persi e dell'utente francesco non esiste più traccia.

Capitolo 6

Permessi

Ammettiamo che il vostro configurazione di internet dopo qualche tentativo sia andato a buon fine. Ma avete finora configurato internet come root, come superutente, non avete ancora dato il permesso a francesco di usare internet. Per fare questo dobbiamo configurare un file rendendolo accessibile anche all'utente francesco.

Il file in questione è **pppd**. Per prima cosa debbo sapere dove risiede fisicamente il file. Ma prima ancora, se mi trovo già ad operare come utente, devo operare come root. Tramite la pressione combinata dei tasti CTRL+ALT+F2 colleghiamoci allora al sistema come root: potremo sempre tornare ad essere francesco tramite la pressione combinata dei tasti CTRL+ALT+F7.

Scriviamo questo comando: **which pppd**, che tradotto in parole povere vuol dire: *dove diavolo si trova il file denominato pppd?*

Vedrete che il sistema risponderà: /usr/sbin/pppd. Il che vuol dire che il file denominato pppd si trova nella sottodirectory /sbin della directory /usr.

Scalandoci in questa directory con il comando cd /usr/sbin, diamo ora il comando che già conosciamo ls -al pppd. Otterremo il seguente listato:

```
-rwxr-xr-x    1 root   root   138596   Sep 20   1999   pppd
```

Ricordando che **r** sta per read, **x** per execute e **w** per write capite ora che questo file è proprietà di root e da lui solo eseguibile.

Per renderlo accessibile all'utente francesco diamo il seguente comando:

```
chown francesco pppd
```

Dando di nuovo il comando ls -al, il file apparirà ora così modificato:

```
-rwxr-xr-x    1 francesco  root   138596   Sep 20   1999   pppd
```

Il che significa che anche l'utente francesco potrà usare internet. Questo permesso vale ovviamente anche per altri files.

Capitolo 7

DOSC

Il riconoscimento del disco C: su cui è montato un sistema DOS o windows avviene ora in maniera automatica durante l'installazione.

Ma è bene apprendere le operazioni manuali da compiere in caso qualcosa non andasse a buon fine. L'operazione è possibile attraverso i seguenti passi.

Spostiamoci in root e creiamo la directory **/dosc** dando il comando **mkdir dosc**.

Adesso, con un editor testuale come joe si edita, sempre da root, il file `/etc/fstab`, scrivendo: `joe /etc/fstab`. Appare il seguente listato:

<code>/dev/hdb5</code>	<code>/</code>	<code>ext2</code>	<code>defaults</code>	<code>1 1</code>
<code>/dev/hda5</code>	<code>/documenti</code>	<code>ext2</code>	<code>ddefaults</code>	<code>1 2</code>
<code>/dev/hdb1</code>	<code>/home</code>	<code>12</code>	<code>ext2</code>	<code>defaults</code>
<code>/dev/hdb6</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0 0</code>
<code>/dev/fd0</code>	<code>/mnt/floppy</code>	<code>ext2</code>	<code>owner,noauto,</code>	<code>0 0</code>
<code>/dev/cdrom</code>	<code>/mnt/cdrom</code>	<code>iso9660</code>	<code>owner,noauto,ro</code>	<code>0 0</code>
<code>none</code>	<code>/proc</code>	<code>proc</code>	<code>defaults</code>	<code>0 0</code>
<code>none</code>	<code>/dev/pts</code>	<code>devpts</code>	<code>gid=5,mode=620</code>	<code>0 0</code>
<code>/dev/hda1</code>	<code>/dosc</code>	<code>vfat</code>	<code>default</code>	<code>0 2</code>

Tabella 7.1: file in `/etc/fstab`. L'ultima riga è quella che va aggiunta.

Nell'ultima riga aggiunta `/dev/hda1` sta ad indicare la partizione dell'HD in cui risiede il Dos o win.* e `/dosc` il nome che forniremo a Linux per individuare la partizione dos di windows. Ricordatevi di andare a capo dopo l'ultima istruzione (`0 2`).

Salvate il file. Al termine forzate il sistema a riconoscere la partizione dos montandola, scrivendo **mount -a**. Se ora vi spostate nella directory digitando `cd /dosc`, e se non avete sbagliato nulla, potrete vedere tutti i vostri vecchi files scritti in dos.

Capitolo 8

L^AT_EX 2_ε: Un creatore di testi

L^AT_EX 2_ε non può essere considerato nel senso puro del termine un editor per testi. Esso è infatti qualcosa di più, qualcosa di unico nell'ambito degli editor testuali.

Generalmente si è abituati a lavorare con editor testuali quali word, wordperfect, o al più con il vecchio e glorioso ventura (sia nella velocissima versione sotto DOS che in quella sotto windows 3.x), ¹staroffice,... editor che appartengono alla cosiddetta categoria WYSIWYG, acronimo che sta a significare, tradotto dall'inglese, quello che tu vedi è quello che poi appare (in fase di stampa: *What You See It What You Get*).

L^AT_EX 2_ε non appartiene a questa categoria. Con esso non si scrive il testo quale si vede, bensì quale... *si pensa*. Se non si entra in questa ottica non si riesce a comprendere il modo in cui L^AT_EX 2_ε lavora, e quindi non si riuscirà mai ad usare a pieno le capacità "potenti" di questo *generatore* di testi professionali. È un consiglio *caldo*: cercate di mettervi sempre, ogni volta che usate un qualsiasi nuovo programma dalla parte dei programmatori, cercate di entrare nella loro testa, di intuire quello che essi volevano ottenere. Solo così potrete sperare di venire in breve alla risoluzione di tutte le difficoltà che incontrate.

L^AT_EX 2_ε è nato soprattutto per gli ambienti scientifici, il suo lato forte è la matematica e le formule matematiche, le tabelle, ma può fare una miriade di altre cose, come scrivere musica²

Questo particolare programma per scrivere testi che deriva da T_EX_{fu} ideato, intorno alla fine degli anni ottanta, dal prof. E. Knuth il quale era stanco del fatto che gli articoli scientifici che egli inviava alle riviste per la pubblicazione gli tornassero indietro con un numero maggiore di errori di quanti egli ne avesse già introdotti. Questo eccentrico professore sospese per questo per due-tre anni la ricerca scientifica e si dette anima e corpo ad ideare un programma che fornisse un testo su cui il tipografo fosse impossibilitato ad intervenire.

T_EX è un programma abbastanza ostico da usare. Per questo un altro scienziato, afflitto dallo stesso problema di Knuth, Leslie Lamport ideò, alcuni anni dopo, una derivazione semplificata (si fa per dire) di T_EX, e nacque il L^AT_EX prima e il L^AT_EX 2_ε poi: quest'ultimo divenne da allora in poi, in breve, lo strumento comunemente utilizzato da tutta la comunità scientifica. ³.

¹Ventura rappresenta un creatore di testi professionali che tuttora conserva una notevole validità per chi miri all'efficienza creativa e non esclusivamente alle mode stilistiche. Predisposto sin dall'origine per qualsiasi composizione di formule matematiche, è strutturato in maniera abbastanza simile a L^AT_EX 2_ε dal quale deve aver appreso qualcosa, dal momento che per ogni "file" capitolo salvato (estensione *.chp) genera files di stile, di indici, di immagini... con un ottimo risultato finale. Appartiene però anch'esso alla categoria degli editor WYSIWYG.

²Tramite L^AT_EX 2_ε si possono anche scrivere partiture musicali, sia per singolo strumento, sia polifoniche, sia per orchestra: in questo caso volersi far del male è dire veramente poco. Neanche immaginate quanto testo bisogna digitare per scrivere una nota! Però, ancora una volta, quello che fa T_EX, o meglio "musixtex", anche in questo caso, non ve lo fa nessuno.

³Fatemi fare alcune considerazioni che non hanno nulla a che vedere con l'informatica! L'uomo deve avere una vocazione particolare per la sofferenza perché non si capisce altrimenti come mai migliaia e migliaia di utenti in tutto il mondo si affannino a studiare L^AT_EX 2_ε quando otterrebbero gli stessi risultati agevolmente utilizzando programmi quali ventura, staroffice, wordperfect, word,... C'è però qualcosa di più della voglia di soffrire e farsi del male! C'è la voglia di conoscere e, tutto sommato, di distinguersi, di autoconsiderarsi qualcosa di più rispetto agli altri. Tutto questo risiede nella gioia ineffabile, che ogni appassionato di informatica conosce, di vedere scritto a

Guardate soltanto il modo in cui i loghi \TeX , \LaTeX e $\text{\LaTeX} 2_{\epsilon}$ vengono scritti a stampa dopo la compilazione e vi renederete conto, anche a livello intuitivo che vi trovate di fronte qualcosa di diverso, di insolito.

8.1 Elementi di tipografia

Nello scrivere, meglio, nel *pensare* un testo da compilare con $\text{\LaTeX} 2_{\epsilon}$ vanno tenuti presenti alcuni elementi basilari di tipografia per quel che riguarda i caratteri. Il **carattere** è un segno grafico utilizzato in tipografia per rappresentare lettere e numeri. Esso si evidenzia per

- **Specie alfabetica.** La collezione di segni per una certa scrittura: la specie alfabetica comune è quella dell'alfabeto latino.
- **Stile.** È una variante del carattere ed ha a che fare con il gusto estetico. *Times* ed *Helvetica*, ad esempio, sono stili.
- **Serie.** Distingue un gruppo di caratteri all'interno di uno stile in base alla forma. Uno stile comunemente si distingue per *pendenza*, *tono*, *larghezza*. In base a queste si può avere il *tondo*, il *corsivo*, il *chiaro*, il *chiarissimo*, il *neretto*. La larghezza distingue uno stile di scrittura a seconda che sia *stretto*, *strettissimo*, *largo* o *larghissimo*.

Nello scrivere un testo finalizzato alla compilazione con il $\text{\LaTeX} 2_{\epsilon}$ bisognerebbe conoscere inoltre alcune impostazioni tipografiche essenziali come il **corpo**, le **dimensioni** e la **scala** dei singoli caratteri, misura che, come dovrebbe essere noto, è espressa in **pica** e che è equivalente ad un 1/6 di pollice. Un punto equivale ad un 1/12 di pica.

I punti di *forza* di un documento fatto compilare da $\text{\LaTeX} 2_{\epsilon}$ possono essere individuati nei seguenti:

1. Giustezza, cioè la larghezza della colonna all'interno della quale si distribuisce il testo. La giustificazione del testo può essere ovviamente sia orizzontale che verticale;
2. Il *font* a disposizione o che si vuole usare e che fa riferimento ad una *famiglia*
3. il *legato* che unisce determinati caratteri. Per esempio i caratteri “f f” ed “f f i”, divengono una volta che sia processati il file **ff** e **ffi**.

8.2 Preparazione del documento

I passi per scrivere un documento sono assai diversi da quelli usati con gli editor tradizionali. Innanzi tutto si scrive il documento senza alcuna formattazione, con un qualsiasi editor, tipo emacs, xemacs, vi, joe, pico, nedit,... inserendo però tutti i comandi⁴ che durante il processo di compilazione vengono riconosciuti da $\text{\LaTeX} 2_{\epsilon}$. Il file va salvato quale **miofile.tex**. Il suffisso ***.tex** è quello che permette al $\text{\LaTeX} 2_{\epsilon}$ in fase di processo del documento di fargli intendere automaticamente che sta trattando un file adatto alla compilazione grafica.

Quindi si procede a generare la bibliografia, l'indice analitico e si processa infine il documento intero che presenta un'eleganza stilistica fuori del comune.

Questo testo, ad esempio, è stato prima scritto con emacs, l'editor professionale cui più volte abbiamo accennato in questo manualetto inserendo tutti i comandi necessari. Quindi ha subito un

terminale “l'operazione è andata a buon fine”, perché dietro questo diagnostico ci sono ore di ricerca degli errori, e la ricerca degli errori è da sempre uno dei modi cardini di conoscenza dell'informatica

⁴Alla fine di questo capitolo è riportato il sorgente *principale* di questo documento. Questo documento si compone di un file principale denominato “linux.tex” e di files “secondari” denominati “parte1.tex, parte2.tex, parte3.tex, gnu.tex”. Si processa in questo caso il file primario “linux.tex” che si trascina dietro nella compilazione gli altri file. Per la creazione dell'indice analitico vale un discorso diverso. Vedi in seguito.

primo debug⁵ e al termine, tolti gli errori che la battitura manuale aveva inevitabilmente inserito, il testo è stato compilato, ottenendone un file con l'estensione .dvi. Successivamente, con un altro comando, è stato ottenuto il file postscript che è il formato classico dei files testuali in ambiente Unix.

Per chi non avesse tanta familiarità con i files in formato postscript e decidesse di optare per il più conosciuto formato acrobat in formato “pdf” non esistono problemi. Dopo aver generato gli indici analitici e la bibliografia è sufficiente lanciare il comando **pdflatex miofile.tex** e si genererà un file con il suffisso *.pdf.

Perché scrivere con T_EX o L^AT_EX 2_ε ?

A livello personale ci può essere poco più che la soddisfazione di riuscire a *masticare* un editor veramente tossico per le “arrabbiature” che fa prendere. Ma se qualcuno dovesse un giorno intraprendere la carriera professionale, e se dovesse quindi un giorno avere l'esigenza di inviare un articolo per la pubblicazione ad una rivista scientifica, allora non può certo sognarsi di inviarglielo, oggi come oggi, in word. Le riviste scientifiche accettano *unicamente* articoli i cui sorgenti siano scritti pronti per essere processati con T_EX o L^AT_EX 2_ε .

Queste sono un po' le disquisizioni filosofiche. Questo trattatello è troppo generico e superficiale perché possa presumere di avere la pretesa di esaurire qui tutto quello che fanno T_EX e L^AT_EX 2_ε . Se vi interessa collegatevi ad internet e scaricatevi di lì tutti i manuali che esistono. Nei siti FTP CTAN e DANTE, tanto per citarne due: esistono centinaia di esempi di files da scaricare.

Volete scrivere un titolo curvo, ad onda, ad ala di gabbiano,... senza ricorrere a Corel? Grazie alla potenzialità grafica di considerare lo schermo come una lavagna formata di tanti pixel T_EX e L^AT_EX 2_ε vi fanno tutto questo. Siamo alle solite: se volete soffrire... Se volete imparare!!!

T_EX e L^AT_EX 2_ε sono forniti in ogni distribuzione Linux o Unix.

Se volete impratichirvi in questo modo particolare di scrivere testi qui appresso presento un esempio di come è stato composto questo manuale.

Alcune cose vanno tenute presenti.

L^AT_EX 2_ε (e con lui T_EX: d'ora innanzi parleremo solo del primo) conoscono ambienti ben definiti che vanno subito precisati, a seconda delle necessità in quel momento richieste. Gli ambienti “principali” definiti sono i seguenti: **article, book, slides, report, letter**

Inoltre si possono definire nuovi comandi che consentono all'utente esperto una notevole flessibilità nell'ambiente di programmazione editoriale di cui qui si tratta.

L^AT_EX 2_ε genera automaticamente i capitoli, gli indici, gli indici analitici, la formattazione del documento, i riferimenti bibliografici,...

Qui appresso viene stampato il file principale scritto per il presente manuale come scritto con un qualsiasi editor testuale in ambiente Unix e salvato come *.tex. Ad un attento lettore non potrà sfuggire che nel finale il file comprende degli **include** ⁶. Questo vuol dire che il file principale, una volta processato, include nel processo di compilazione tutti i files collegati e procede alla compilazione dell'indice generale e dell'indice analitico. ⁷

Se Linus Torvalds aveva definito il suo linux *Just for fun*, solo per divertirsi, a buon diritto l'uso di questo editor può definirsi *Just for suffering*.... Ogni traduzione dall'inglese è inutile!

Tutta questa sofferenza si può comunque in gran parte evitare se si usa, in ambiente linux l'editor **lyx**. Questo editor permette con gran facilità di inserire tabelle, figure,... e di esportare alla fine il lavoro in L^AT_EX 2_ε.

Il file così ottenuto va processato (come se fosse un normale file *.tex) secondo il comando:

latex miofile.tex

Dopo si lancia il comando

⁵Il “debug” si ha *...automaticamente...*, con l'*inceppamento* in fase di compilazione e la segnalazione che alla tale riga del vostro editor è presente un errore. Va segnalato comunque che L^AT_EX 2_ε per riga intenda il paragrafo, quindi se l'errore, poniamo caso, è indicato alla linea 131, vuol dire che sta in una delle linee che compongono il paragrafo 131

⁶qui è compreso solo l'ultimo, prima dell'indice, relativo a questo capitolo

⁷Qui gli “include” non sono presentati sia perché troppo lungo sia perché irrilevanti comparando già compilati. Notate soltanto due cose nel file principale “linux.tex”: a. gli include compaiono **senza** il suffisso *.tex, b. gli include **non** contengono più il prambolo o la dichiaratoria di stile, di package,...

dvips -o -t miofile.ps miofile.dvi

Si vede e si stampa il file tramite il comando

gv miofile.ps

Se si vuole ottenere un file acrobat il comando da dare è

pdflatex miofile.tex. Prima però bisogna aver ricompilato anche in questo caso gli indici tramite il comando **makeindex miofile.idx**. Parimenti che in $\text{\LaTeX} 2_{\epsilon}$, il file **miofile.ind** così ottenuto va copiato quale **indice.tex**, solo così l'**include indice.tex** sarà processato e generato quale indice analitico. Esiste anche un altro sistema più sbrigativo per quest'ultima operazione. Scopritelo da soli...

Segue il testo `linux.tex` e parzialmente il file `parte3.tex` scritto per il presente articolo. Chi vuole legga i commenti *parziali*.

```
%INIZIA INSERIMENTO TESTO CHE NON VIENE PROCESSATO

%      PREAMBOLO

\documentclass[a4paper,10pt]{book} % formato carta, punti di stampa, tipo pubblicazione
\usepackage[italian]{babel}
\usepackage[T1]{fontenc} %caratteri ISO Se si desidera compilare il file in formato pdf va tolto
\usepackage[latin1]{inputenc} % Utilizza accentazione italiana
\usepackage{float} % consente alle figure ed alle tabelle di ‘galleggiare’ nel testo
\usepackage{a4wide} % larghezza
\usepackage{array} % vettoriale
\usepackage{graphicx} % grafica
\pagestyle{empty}
\usepackage{epsfig} % figure
\usepackage{makeidx} % indice
\frenchspacing %inserisce ritorno a capo

%-----
%          ‘PAGINA DI COPERTINA’
%-----

\footskip      1cm
\pagestyle{headings}
\makeindex
\begin{document}
\begin{titlepage}

\begin{center}
    \null

\small Enrico F. Macchia \hfill

\vspace{1cm}

\vspace{1.2cm}

\vfill \thicklines

{\Huge\bf
\begin{tabular}{|@{\hspace{1cm}}c@{\hspace{1cm}}|}
\hline
```

```

\\
\\
Breve\\
\\
\\
Introduzione \\
\\
\\
a LINUX\\
\\
\\
{\bf \small Manuale di Informatica ad uso didattico } \\
\\
\hline
\end{tabular}
}

\vspace{3cm}

\end{titlepage}

\bibstyle

\tableofcontents

\include{parte1} %Il testo è composto di 3 parti più l'indice
\include{parte2}
\include{parte3}
\include{indice}

%-----
%QUI INIZIA LA PARTE TERZA DEL DOCUMENTO INCLUSO
%-----

```

Esempio:

Una frase tipo quella riportata al secondo paragrafo del presente capitolo: “al più con il vecchio e glorioso ventura⁸”

⁸Ventura rappresenta sia nelle versioni sotto DOS che in quelle sotto windows...

viene “digitata così:

```
(.....vecchio e glorioso ventura\index{ventura} \footnote{Ventura.... }
```

```
rappresenta sia nelle versioni sotto DOS\index{DOS} )
```

QUESTI COMANDI GENERANO l’indice analitico per la voce “ventura”

```
\index{ventura}
```

e la nota a pie’ di pagina.

```
\footnote{Ventura rappresenta..... }
```

I COMANDI SONO RACCHIUSI, QUASI SEMPRE FRA PARENTESI GRAFFE PRECEDUTE,

ALL’INTERNO O ALL’ESTERNO DA UNA BARRA {\VOCE_COMANDO}

```
%TERMINA INSERIMENTO TESTO
```

Capitolo 9

Gnu General Public Licence

Version 2, June 1991

9.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

9.2 Terms and Conditions

- This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License.

The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

- You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

- You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

1. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
2. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
3. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

5. a- Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
6. b- Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
7. c- Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

8. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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.
9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
10. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
13. The Free Software Foundation may publish revised and/or new versions of the General Public 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.
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
14. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

9.3 No Warranty

- BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
- IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE

THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES

9.4 How to Apply these Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

one line to give the program’s name and an idea of what it does.

Copyright © 19yy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139,

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright © 19yy name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’. This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.
signature of Ty Coon, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.