

Speciale Linux

Un po' di storia

di Giuseppe Zanetti

Non è facile spiegare in poche pagine cosa significhi Linux senza rischiare di cadere nel riduttivo e nella mera descrizione di aspetti tecnici e liquidarlo semplicemente come "un sistema operativo simile a UNIX ma gratuito". In realtà, perdendo un po' di tempo per conoscerne la storia, esso rappresenta qualcosa di molto più profondo di come appare esternamente o a prima vista: è forse uno dei simboli più importanti e significativi, di come Internet e le reti di comunicazione abbiano iniziato a modificare la vita, la società, i rapporti interpersonali e persino l'economia di questo fine millennio. Linux, e le idee che ci stanno dietro, hanno reso vere cose che solo fino a ieri erano inimmaginabili, come il lavorare in migliaia di persone, ciascuno a casa propria - che può essere in qualunque angolo del mondo - allo stesso progetto.

La nascita di Linux

Linux nacque nel 1991, dall'idea di uno studente diciannovenne di informatica dell'Università di Helsinki in Finlandia, Linus Benedict Torvalds, il quale, spinto dalla curiosità di provare a programmare il suo nuovo 386, decise di iniziare a scrivere un piccolo kernel (nucleo alla base di un sistema operativo) in grado di far girare contemporaneamente due semplicissimi programmi (uno scriveva sullo schermo una "A" maiuscola, l'altro una "B").

La fortuna e la grande intuizione di Linus furono quelle di non tenere per sé la cosa ma di condividerla con gli altri programmatori attraverso Internet, che dieci anni or sono non era ancora un fenomeno di massa ma era limitata quasi solo al mondo universitario.

La prima apparizione di Linux avvenne in un messaggio impostato nel 1991 nel newsgroup (gruppo di discussione di USENET) comp.os.minix.

In questo messaggio Linus presentava il suo lavoro come la base per un sistema operativo che avrebbe dovuto superare i limiti di Minix (un sistema operativo simile a UNIX, scritto a scopo didattico come complemento ad un suo libro da Andy. S. Tanenbaum, uno dei più eminenti ricercatori nel campo dei sistemi operativi e del networking).

Ad una critica del professore su Linux, Torvalds rispose con una critica

piuttosto feroce nei confronti di Minix. Ovviamente il professore non accettò troppo serenamente il fatto che uno studentello osasse criticare in maniera anche così pesante il suo lavoro e questo diede vita ad un intenso scambio di messaggi pubblici fra i due (trovate il testo completo della disputa su ftp://sunsite.unc.edu/pub/Linux/docs/old/misc/linux_is_obsolete.txt.z).

Il motivo principale del contendere era insito, oltre che in una questione di orgoglio, anche nella diversa struttura interna dei due sistemi operativi: Minix era basato infatti sul concetto del microkernel, in cui le diverse funzioni (gestione della memoria, filesystem, device driver, ...) sono implementate in modo separato e colloquiano fra loro scambiandosi opportuni messaggi. Linux invece era basato su una architettura più tradizionale, in cui le diverse funzioni sono inserite nel kernel in maniera monolitica.

Quando ho scritto Linux conosco UNIX da sei mesi e ho pensato di poterlo riscrivere da zero.

Se lo avessi conosciuto meglio, non avrei iniziato.

Alle critiche del luminare, lo studente rispondeva semplicemente che il suo sistema operativo era forse teoricamente meno bello ma che, messo alla prova dei fatti, avrebbe funzionato meglio rispetto a Minix.



> MINIX is a microkernel-based system.... LINUX is a monolithic style system.

If this was the only criterion for the "goodness" of a kernel, you'd be right. What you don't mention is that minix doesn't do the micro-kernel thing very well, and has problems with real multitasking (in the kernel). If I had made an OS that had problems with a multithreading filesystem, I wouldn't be so fast to condemn others: in fact, I'd do my damndest to make others forget about the fiasco.

In realtà uno degli aspetti che Linus contestava maggiormente a Tanenbaum era anche che Minix non fosse disponibile gratuitamente, cosa che egli riteneva inaccettabile per un sistema ad

Cos'è Linux

tratto da "Linux Information Sheet" di Michael K. Johnson <johnsonm@redhat.com>

traduzione a cura di Giovanni Bortolozzo, <borto@pluto.linux.it>

Potete trovare il documento completo su <http://www.pluto.linux.it/ildp/HOWTO/index.html>

Linux è una reimplementazione completamente gratuita delle specifiche POSIX, con estensioni SYSV e BSD (il che significa che sembra proprio come Unix, ma non è derivato dallo stesso codice di partenza), disponibile sia come codice sorgente sia in formato binario. Il suo copyright è posseduto da Linus Torvalds <torvalds@transmeta.com> e da altri collaboratori, ed è liberamente distribuibile nei termini della GNU General Public License (GPL). Una copia della GPL è inclusa nei sorgenti di Linux; è possibile ottenerne una copia anche da <ftp://prep.ai.mit.edu/pub/gnu/COPYING>.

Linux, di per sé, è solo il kernel del sistema operativo, la parte che controlla l'hardware, gestisce i file, suddivide i processi e così via. Esistono diverse combinazioni di Linux con insiemi di utilità e applicazioni per formare un sistema operativo completo. Ognuna di queste combinazioni è detta una distribuzione di Linux. La parola Linux, sebbene in senso stretto si riferisca specificamente al kernel, è largamente e correttamente usata per riferirsi all'intero sistema operativo costruito attorno al kernel Linux. Per un elenco e una breve descrizione delle diverse distribuzioni, si veda <http://sunsite.unc.edu/LDP/HOWTO/Distribution-HOWTO.html>. Nessuna di queste distribuzioni è il "Linux ufficiale".

Linux non è di pubblico dominio, non è nemmeno "shareware". È un software "libero", comunemente chiamato freeware o Open Source Software[tm] (si veda <http://www.opensource.org>), e si possono distribuire o venderne copie, ma si devono includerne i sorgenti o renderli disponibili nello stesso modo con il quale si distribuiscono o si vendono gli eseguibili. Se si distribuisce una qualsiasi modifica, si è obbligati per legge a distribuire i sorgenti di queste modifiche. Si veda la GNU General Public License per i dettagli.

Linux è ancora libero, come nella versione 2.0, e continuerà a esserlo. La natura del copyright GNU, a cui Linux è soggetto, renderebbe illegale il fatto che non sia più libero. Notare che: la parte "libera" implica l'accesso ai sorgenti indipendentemente dall'aspetto monetario; è perfettamente legale far pagare denaro per la distribuzione di Linux, a patto che venga distribuito anche il codice sorgente. Questa è una generalizzazione; per aspetti più accurati, leggere la GPL.

Linux gira su macchine 386/486/Pentium con bus ISA, EISA, PCI e VLB. MCA (bus della IBM) non è attualmente ben supportato, sebbene nella nuova versione in fase di sviluppo, la 2.1.x, il supporto sia stato inserito. Se siete interessati, guardate <http://glycerine.itsmm.uni.edu/mca>.

È in corso il port per diverse piattaforme Motorola 680x0 (attualmente usate in alcuni Amiga, Atari e macchine VME), che ora funziona abbastanza bene. Richiede un 68020 con MMU, un 68030, un 68040 o un 68060 e richiede anche l'FPU. Per ora il supporto di rete e X non funzionano. Si veda <news:comp.os.linux.m68k>.

Linux funziona bene sulle CPU Alpha della DEC, supportando attualmente le piattaforme "Jensen", "NoName", "Cabriolet", "Universal Desktop Box" (meglio conosciuta come Multia). Per altre informazioni, si veda <http://www.azstarnet.com/~axplinux/FAQ.html>.

Linux funziona bene su Sun SPARC e viene utilizzato già su molte macchine sun4c, sun4m e sun4u, mentre il supporto per sun4 è in fase di sviluppo. Red Hat Linux è (alla scrittura di questo testo) la sola distribuzione disponibile per SPARC; si veda <http://www.redhat.com/support/docs/rhl-sparc/>.

Linux è in attiva fase di port per architetture PowerPC, incluse le macchine PowerMac (Nubus e PCI), Motorola, IBM e Be. Si veda

<http://www.cs.nmt.edu/~linuxppc/> e <http://www.linuxppc.org/>.

Port su altre macchine, tra cui MIPS (si veda <http://linus.linux.sgi.com> e <http://lena.fnet.fr/>) e ARM, sono in corso e stanno mostrando dei progressi. Non trattenete il fiato, ma se siete interessati e capaci di contribuire, potete trovare facilmente altri sviluppatori che vorrebbero lavorare con voi.

Linux non è più considerato in beta test da quando è stata rilasciata la versione 1.0 il 14 marzo 1994. C'erano ancora bug nel sistema, e nuovi bug salteranno fuori e saranno risolti col passare del tempo. Siccome Linux segue il "modello a sviluppo aperto", tutte le nuove versioni sono rilasciate al pubblico anche se non sono considerate "prodotti di qualità". Comunque, per aiutare la gente a capire se sta prelevando una versione stabile o no, è stato adottato il seguente schema: le versioni n.x.y, dove x è un numero pari, sono le versioni stabili, e come vengono risolti bug viene incrementato y. Così dalla versione 1.2.2 alla 1.2.3 ci fu la correzione di alcuni bug e non l'aggiunta di nuove caratteristiche. Le versioni n.x.y, dove x è un numero dispari, sono di qualità beta per soli sviluppatori, e possono quindi essere instabili e cadere (crash), ma hanno nuove caratteristiche aggiuntive. Non appena il kernel attualmente in sviluppo diviene stabile, verrà "congelato" in un nuovo kernel "stabile", e lo sviluppo continuerà con una nuova versione del kernel. Si noti che la maggior parte dei rilasci del kernel Linux, beta o meno, sono relativamente robuste; in questo contesto "stabile" significa che "cambia lentamente" oltre a "robusto".

La versione stabile attuale è la 2.2.15 (questa continua a cambiare a mano a mano che vengono aggiunti nuovi supporti hardware e corretti bug), ed è partito lo sviluppo del kernel sperimentali 2.3.x. I sorgenti del kernel Linux contengono un file, Documentation/Changes, che spiega i cambiamenti di cui si deve essere consci quando si aggiorna la versione del kernel. Comunque, la stragrande maggioranza degli utenti di Linux semplicemente aggiornano periodicamente la loro distribuzione per avere una nuova versione del kernel.

Molte versioni di Linux, beta o no, sono abbastanza stabili e si possono usare se fanno quel che vi serve. Un sito ha utilizzato un computer con la versione 0.97p1 (rilasciata nell'estate del 1992) per più di 136 giorni senza nessun errore o crash (e avrebbe funzionato ancora più a lungo se un operatore sbadato non avesse erroneamente staccato la spina dell'alimentatore principale...) Altri hanno segnalato uptime di oltre un anno. Un sito ha ancora un computer che usa Linux 0.99p15 da oltre 600 giorni.

Una cosa da considerare è che Linux è sviluppato usando un modello aperto e distribuito, invece di un modello chiuso e centralizzato come molto altro software. Questo significa che la versione di sviluppo corrente è sempre pubblica (con un ritardo di una o due settimane) in modo che tutti possano usarla. Il risultato è che ogni volta una versione con nuove funzionalità viene rilasciata, quasi sempre contiene bug, ma con il rapido sviluppo esistente i bug sono trovati e corretti velocemente, spesso in ore, in quanto molta gente lavora per risolverli.

Diversamente, il modello chiuso e centralizzato significa che una sola persona o un gruppo lavora al progetto, e solo loro rilasciano il software quando ritengono che funzioni bene. Spesso questo porta a un lungo intervallo tra le release, una lunga attesa per risolvere i bug e uno sviluppo lento. Certamente l'ultima release pubblica di tale software è, a volte, di qualità superiore, ma la velocità di sviluppo è generalmente molto inferiore.

La licenza GNU GPL

Questa è una traduzione italiana non ufficiale della Licenza Pubblica Generale GNU. Non è pubblicata dalla Free Software Foundation e non ha valore legale nell'esprimere i termini di distribuzione del software che usa la licenza GPL. Solo la versione originale in inglese della licenza ha valore legale. Ad ogni modo, speriamo che questa traduzione aiuti le persone di lingua italiana a capire meglio il significato della licenza GPL.

LICENZA PUBBLICA GENERICA (GPL) DEL PROGETTO GNU
Versione 2, Giugno 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

Traduzione curata dal gruppo Pluto e da ILS, ultimo aggiornamento, 30 luglio 1998.

Tutti possono copiare e distribuire copie letterali di questo documento di licenza, ma non è lecito modificarlo.

Preambolo

Le licenze per la maggioranza dei programmi hanno lo scopo di togliere all'utente la libertà di dividerlo e di modificarlo. Al contrario, la Licenza Pubblica Generica GNU è intesa a garantire la libertà di condividere e modificare il free software, al fine di assicurare che i programmi siano "liberi" per tutti i loro utenti. Questa Licenza si applica alla maggioranza dei programmi della Free Software Foundation e ad ogni altro programma i cui autori hanno scelto questa Licenza. Alcuni altri programmi della Free Software Foundation sono invece coperti dalla Licenza Pubblica Generica per Librerie. Chiunque può usare questa Licenza per i propri programmi. Quando si parla di "free software", ci si riferisce alla libertà, non al prezzo. Le nostre Licenze (la GPL e la LGPL) sono progettate per assicurarsi che ciascuno abbia la libertà di distribuire copie del free software (e farsi pagare per questo, se vuole), che ciascuno riceva il codice sorgente o che lo possa ottenere se lo desidera, che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi "liberi" e che ciascuno sappia di potere fare queste cose.

Per proteggere i diritti dell'utente, abbiamo bisogno di creare delle restrizioni che vietino a chiunque di negare questi diritti o di chiedere di rinunciarvi. Queste restrizioni si traducono in certe responsabilità per chi distribuisce copie del software e per chi lo modifica.

Per esempio, chi distribuisce copie di un Programma coperto da GPL, sia gratis sia in cambio di un compenso, deve dare ai destinatari tutti i diritti che ha ricevuto. Deve anche assicurarsi che i destinatari ricevano o possano ricevere il codice sorgente. E deve mostrar loro queste condizioni di Licenza, in modo che conoscano i loro diritti.

Proteggiamo i diritti dell'utente in due modi: (1) proteggendo il software con un copyright, e (2) offrendo una Licenza che offre il permesso legale di copiare, distribuire e/o modificare il Programma.

Infine, per proteggere ogni autore e noi stessi, vogliamo assicurarci che ognuno capisca che non ci sono garanzie per i programmi coperti da GPL. Se il Programma viene modificato da qualcun altro e ridistribuito, vogliamo che gli acquirenti sappiano che ciò che hanno non è l'originale, in modo che ogni problema introdotto da altri non si rifletta sulla reputazione degli autori originali.

Infine, ogni programma libero è costantemente minacciato dai brevetti sui programmi. Vogliamo evitare il pericolo che chi ridistribuisce un Programma libero ottenga brevetti personali, rendendo perciò il Programma una cosa di sua proprietà. Per prevenire questo, abbiamo chiarito che ogni prodotto brevettato debba essere distribuito per il libero uso da parte di chiunque, o non distribuito affatto.

Seguono i termini e le condizioni precisi per la copia, la distribuzione e la modifica.

LICENZA PUBBLICA GENERICA GNU TERMINI E CONDIZIONI PER LA COPIA, LA DISTRIBUZIONE E LA MODIFICA

0. Questa Licenza si applica a ogni Programma o altra opera che contenga una nota da parte del detentore del copyright che dica che tale opera può distribuita sotto i termini di questa Licenza Pubblica Generica. Il termine "Programma" nel seguito indica ognuno di questi programmi o lavori, e l'espressione "lavoro basato sul Programma" indica sia il Programma sia ogni opera considerata "derivata" in base alla legge sul Copyright: cioè un

lavoro contenente il programma o una porzione di esso, sia letteralmente sia modificato e/o tradotto in un'altra lingua; da qui in avanti, la traduzione è in ogni caso considerata una "modifica". Vengono ora elencati i diritti dei detentori di licenza.

Attività diverse dalla copiatura, distribuzione e modifica non sono coperte da questa Licenza e sono al di fuori della sua influenza. L'atto di eseguire il programma non viene limitato, e l'output del programma è coperto da questa Licenza solo se il suo contenuto costituisce un lavoro basato sul Programma (indipendentemente dal fatto che sia stato creato eseguendo il Programma). In base alla natura del Programma il suo output può essere o meno coperto da questa Licenza.

1. È lecito copiare e distribuire copie letterali del codice sorgente del Programma così come viene ricevuto, con qualsiasi mezzo, a condizione che venga riprodotta chiaramente su ogni copia una appropriata nota di copyright e di assenza di garanzia; che si mantengano intatti tutti i riferimenti a questa Licenza e all'assenza di ogni garanzia; che si dia a ogni altro destinatario del Programma una copia di questa Licenza insieme al Programma.

È possibile richiedere un pagamento per il trasferimento fisico di una copia del Programma, è anche possibile a propria discrezione richiedere un pagamento in cambio di una copertura assicurativa.

2. È lecito modificare la propria copia o copie del Programma, o parte di esso, creando perciò un lavoro basato sul Programma, e copiare o distribuire queste modifiche e questi lavori sotto i termini del precedente punto 1, a patto che anche tutte queste condizioni vengano soddisfatte:

a) Bisogna indicare chiaramente nei file che si tratta di copie modificate e la data di ogni modifica.

b) Bisogna fare in modo che ogni lavoro distribuito o pubblicato, che in parte o nella sua totalità derivi dal Programma o da parti di esso, sia globalmente utilizzabile da terze parti secondo le condizioni di questa licenza.

c) Se di solito il programma modificato legge comandi interattivamente quando eseguito, bisogna fare in modo che all'inizio dell'esecuzione interattiva usuale, stampi un messaggio contenente una appropriata nota di copyright e di assenza di garanzia (oppure che specifichi il tipo di garanzia che si offre). Il messaggio deve inoltre specificare agli utenti che possono ridistribuire il programma nelle condizioni qui descritte e deve indicare come reperire questa licenza. Se però il programma di partenza è interattivo ma normalmente non stampa tale messaggio, non occorre che un lavoro derivato lo stampi.

Questi requisiti si applicano al lavoro modificato nel suo complesso. Se sussistono parti identificabili del lavoro modificato che non siano derivate dal Programma e che possono essere ragionevolmente considerate lavori indipendenti, allora questa Licenza e i suoi termini non si applicano a queste parti quando vengono distribuite separatamente. Se però queste parti vengono distribuite all'interno di un prodotto che è un lavoro basato sul Programma, la distribuzione di questo prodotto nel suo complesso deve avvenire nei termini di questa Licenza, le cui norme nei confronti di altri utenti si estendono a tutto il prodotto, e quindi ad ogni sua parte, chiunque ne sia l'autore.

Sia chiaro che non è nelle intenzioni di questa sezione accampare diritti su lavori scritti interamente da altri, l'intento è piuttosto quello di esercitare il diritto di controllare la distribuzione di lavori derivati o dal Programma o contenenti esso.

Inoltre, se il Programma o un lavoro derivato da esso viene aggregato ad un altro lavoro non derivato dal Programma su di un mezzo di immagazzinamento o di distribuzione, il lavoro non derivato non deve essere coperto da questa licenza.

3. È lecito copiare e distribuire il Programma (o un lavoro basato su di esso, come espresso al punto 2) sotto forma di codice oggetto o eseguibile sotto i termini dei precedenti punti 1 e 2, a patto che si applichi una delle seguenti condizioni:

a) Il Programma sia corredato dal codice sorgente completo, in una forma leggibile dal calcolatore e tale sorgente deve essere fornito secondo le regole dei precedenti punti 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.

b) Il Programma sia accompagnato da un'offerta scritta, valida per almeno tre anni, di fornire a chiunque ne faccia richiesta una copia



completa del codice sorgente, in una forma leggibile dal calcolatore, in cambio di un compenso non superiore al costo del trasferimento fisico di tale copia, che deve essere fornita secondo le regole dei precedenti punti 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.

c) Il Programma sia accompagnato dalle informazioni che sono state ricevute riguardo alla possibilità di avere il codice sorgente. Questa alternativa è permessa solo in caso di distribuzioni non commerciali e solo se il programma è stato ricevuto sotto forma di codice oggetto o eseguibile in accordo al precedente punto B.

Per "codice sorgente completo" di un lavoro si intende la forma preferenziale usata per modificare un lavoro. Per un programma eseguibile, "codice sorgente completo" significa tutto il codice sorgente di tutti i moduli in esso contenuti, più ogni file associato che definisca le interfacce esterne del programma, più gli script usati per controllare la compilazione e l'installazione dell'eseguibile. In ogni caso non è necessario che il codice sorgente fornito includa nulla che sia normalmente distribuito (in forma sorgente o in formato binario) con i principali componenti del sistema operativo sotto cui viene eseguito il Programma (compilatore, kernel, e così via), a meno che tali componenti accompagnino l'eseguibile.

Se la distribuzione dell'eseguibile o del codice oggetto è effettuata indicando un luogo dal quale sia possibile copiarlo, permettere la copia del codice sorgente dallo stesso luogo è considerata una valida forma di distribuzione del codice sorgente, anche se copiare il sorgente è facoltativo per l'acquirente.

4. Non è lecito copiare, modificare, sublicenziare, o distribuire il Programma in modi diversi da quelli espressamente previsti da questa Licenza. Ogni tentativo di copiare, modificare, sublicenziare o distribuire il Programma non è autorizzato, e farà terminare automaticamente i diritti garantiti da questa Licenza. D'altra parte ogni acquirente che abbia ricevuto copie, o diritti, coperti da questa Licenza da parte di persone che violano la Licenza come qui indicato non vedranno invalidare la loro Licenza, purché si comportino conformemente ad essa.

5. L'acquirente non è obbligato ad accettare questa Licenza, poiché non l'ha firmata. D'altra parte nessun altro documento garantisce il permesso di modificare o distribuire il Programma o i lavori derivati da esso. Queste azioni sono proibite dalla legge per chi non accetta questa Licenza; perciò, modificando o distribuendo il Programma o un lavoro basato sul programma, si indica nel fare ciò l'accettazione di questa Licenza e quindi di tutti i suoi termini e le condizioni poste sulla copia, la distribuzione e la modifica del Programma o di lavori basati su di esso.

6. Ogni volta che il Programma o un lavoro basato su di esso vengono distribuiti, l'acquirente riceve automaticamente una licenza d'uso da parte del licenziatario originale. Tale licenza regola la copia, la distribuzione e la modifica del Programma secondo questi termini e queste condizioni. Non è lecito imporre restrizioni ulteriori all'acquirente nel suo esercizio dei diritti qui garantiti. Chi distribuisce programmi coperti da questa Licenza non è comunque responsabile per la conformità alla Licenza da parte di terze parti.

7. Se, come conseguenza del giudizio di una corte, o di una imputazione per la violazione di un brevetto o per ogni altra ragione (anche non relativa a questioni di brevetti), vengono imposte condizioni che contraddicono le condizioni di questa licenza, che queste condizioni siano dettate dalla corte, da accordi tra le parti o altro, queste condizioni non esimono nessuno dall'osservazione di questa Licenza. Se non è possibile distribuire un prodotto in un modo che soddisfi simultaneamente gli obblighi dettati da questa Licenza e altri obblighi pertinenti, il prodotto non può essere affatto distribuito. Per esempio, se un brevetto non permettesse a tutti quelli che lo ricevono di ridistribuire il Programma senza obbligare al pagamento di diritti, allora l'unico modo per soddisfare contemporaneamente il brevetto e questa Licenza è di non distribuire affatto il Programma.

Se parti di questo punto sono ritenute non valide o inapplicabili per qualsiasi circostanza, deve comunque essere applicata l'idea espressa da questo punto; in ogni altra circostanza invece deve essere applicato il punto 7 nel suo complesso.

Non è nello scopo di questo punto indurre gli utenti ad infrangere alcun brevetto né ogni altra rivendicazione di diritti di proprietà, né di

contestare la validità di alcuna di queste rivendicazioni; lo scopo di questo punto è solo quello di proteggere l'integrità del sistema di distribuzione dei programmi liberi, che viene realizzato tramite l'uso della licenza pubblica. Molte persone hanno contribuito generosamente alla vasta gamma di programmi distribuiti attraverso questo sistema, basandosi sull'applicazione fedele di tale sistema. L'autore/donatore può decidere di sua volontà se preferisce distribuire il software avvalendosi di altri sistemi, e l'acquirente non può imporre la scelta del sistema di distribuzione.

Questo punto serve a rendere il più chiaro possibile ciò che crediamo sia una conseguenza del resto di questa Licenza.

8. Se in alcuni paesi la distribuzione e/o l'uso del Programma sono limitati da brevetto o dall'uso di interfacce coperte da copyright, il detentore del copyright originale che pone il Programma sotto questa Licenza può aggiungere limiti geografici espliciti alla distribuzione, per escludere questi paesi dalla distribuzione stessa, in modo che il programma possa essere distribuito solo nei paesi non esclusi da questa regola. In questo caso i limiti geografici sono inclusi in questa Licenza e ne fanno parte a tutti gli effetti.

9. All'occorrenza la Free Software Foundation può pubblicare revisioni o nuove versioni di questa Licenza Pubblica Generica. Tali nuove versioni saranno simili a questa nello spirito, ma potranno differire nei dettagli al fine di coprire nuovi problemi e nuove situazioni.

Ad ogni versione viene dato un numero identificativo. Se il Programma asserisce di essere coperto da una particolare versione di questa Licenza e "da ogni versione successiva", l'acquirente può scegliere se seguire le condizioni della versione specificata o di una successiva. Se il Programma non specifica quale versione di questa Licenza deve applicarsi, l'acquirente può scegliere una qualsiasi versione tra quelle pubblicate dalla Free Software Foundation.

10. Se si desidera incorporare parti del Programma in altri programmi liberi le cui condizioni di distribuzione differiscano da queste, è possibile scrivere all'autore del Programma per chiederne l'autorizzazione. Per il software il cui copyright è detenuto dalla Free Software Foundation, si scriva alla Free Software Foundation; talvolta facciamo eccezioni alle regole di questa Licenza. La nostra decisione sarà guidata da due scopi: preservare la libertà di tutti i prodotti derivati dal nostro free software e promuovere la condivisione e il riutilizzo del software in generale.

NON C'È GARANZIA

11. POICHÉ IL PROGRAMMA È CONCESSO IN USO GRATUITAMENTE, NON C'È GARANZIA PER IL PROGRAMMA, NEI LIMITI PERMESSI DALLE VIGENTI LEGGI. SE NON INDICATO DIVERSAMENTE PER ISCRITTO, IL DETENTORE DEL COPYRIGHT E LE ALTRE PARTI FORNISCONO IL PROGRAMMA "COSÌ COM'È", SENZA ALCUN TIPO DI GARANZIA, NÉ ESPlicitA NÉ IMPLICITA; CIO' COMPRENDE, SENZA LIMITARSI A QUESTO, LA GARANZIA IMPLICITA DI COMMERCIALIZZABILITÀ E UTILIZZABILITÀ PER UN PARTICOLARE SCOPO. L'INTERO RISCHIO CONCERNENTE LA QUALITÀ E LE PRESTAZIONI DEL PROGRAMMA È DELL'ACQUIRENTE. SE IL PROGRAMMA DOVESSE RIVELARSI DIFETTOSO, L'ACQUIRENTE SI ASSUME IL COSTO DI OGNI MANUTENZIONE, RIPARAZIONE O CORREZIONE NECESSARIA.

12. NÉ IL DETENTORE DEL COPYRIGHT NÉ ALTRE PARTI CHE POSSONO MODIFICARE O RIDISTRIBUIRE IL PROGRAMMA COME PERMESSO IN QUESTA LICENZA SONO RESPONSABILI PER DANNI NEI CONFRONTI DELL'ACQUIRENTE, A MENO CHE QUESTO NON SIA RICHIESTO DALLE LEGGI VIGENTI O APPAIA IN UN ACCORDO SCRITTO. SONO INCLUSI DANNI GENERICI, SPECIALI O INCIDENTALI, COME PURE I DANNI CHE CONSEGUONO DALL'USO O DALL'IMPOSSIBILITÀ DI USARE IL PROGRAMMA; CIO' COMPRENDE, SENZA LIMITARSI A QUESTO, LA PERDITA DI DATI, LA CORRUZIONE DEI DATI, LE PERDITE SOSTENUTE DALL'ACQUIRENTE O DA TERZE PARTI E L'INABILITÀ DEL PROGRAMMA A LAVORARE INSIEME AD ALTRI PROGRAMMI. ANCHE SE IL DETENTORE O ALTRE PARTI SONO STATE AVVISATE DELLA POSSIBILITÀ DI QUESTI DANNI.

FINE DEI TERMINI E DELLE CONDIZIONI



Lavorare con Linux

Ogni singola parte di Linux deriva dal contributo di qualcuno ed è stata scritta con lo scopo di andare incontro ad una esigenza, fosse anche quella specifica della singola persona che ha contribuito il pezzettino. Se ci pensate bene non si può perdere tempo o investire ore di lavoro dietro alle esigenze di una sola persona se lo scopo ultimo è solamente quello di fare soldi.

Non per questo voglio dire che chi sviluppa Linux sia povero o votato alla povertà o alla missione. Il professionista medio che lavora con vera passione su e con Linux non naviga certamente nell'oro - e in questo momento potrebbe benissimo farlo, dato il successo commerciale che sta avendo il nostro sistema operativo - anche perché di solito è abbastanza intelligente da aver capito che i soldi non possono essere l'unico aspetto ricercato nella vita, ma solamente un pezzettino che non deve essere confuso o barattato in cambio di tutte le altre cose belle che la vita può dare, come lavorare, magari comodamente a casa propria, ad un progetto che piace.

Le opportunità offerte da Linux e dal software libero sono moltissime: sviluppo e personalizzazione del software, consulenza, rivendita di programmi, hardware, editoria, formazione, ... e persino... fabbricazione di pinguini di peluche. Per ognuna di queste tipologie c'è una casistica di aziende o persone che si sono riuscite a ritagliare il proprio spazio, piccolo o grande. Alcuni nomi: Red Hat, Caldera, S.u.S.E., Linuxcare, O'Reilly, ... e, nel loro piccolo, anche Mario Rossi, Andrea Verdi, Antonio Bianchi, ...

Fare il consulente per Linux è a prima vista difficile per la complessità del sistema operativo, ma in realtà c'è talmente tanta documentazione che con un po' di buona volontà imparare diventa facile. Inoltre il fatto di poter vedere come sono fatti dentro i programmi e la possibilità di adattarli alle esigenze proprie o del cliente, semplifica enormemente la vita, in quanto permette di trovare delle soluzioni a problemi nuovi riutilizzando molto lavoro fatto da altri.

Linux mette a disposizione - intrinsecamente per come è costruito - moltissimi strumenti per il telelavoro, cosicché usando Internet non vi è moltissima differenza nel lavorare sulla macchina sopra la propria scrivania oppure su una a 10000 chilometri di distanza. Quando vedrete un tipo triste e accaldato dopo una giornata di corse avanti e indietro per la città, fategli un sorriso amichevole: potrebbe essere un consulente di un altro sistema operativo al termine di una giornata di lavoro.

E' possibile guadagnarsi da vivere facendo gli sviluppatori di software libero e si è facilitati in questo dal fatto di poter incorporare nei propri programmi il lavoro scritto da altri, senza doversi scrivere tutto da zero o spendere soldi in licenze per librerie o strumenti di sviluppo. Il giusto che si potrà chiedere al committente in cambio del proprio lavoro sarà il valore effettivo di quanto lavorato, senza costi fittizi. Non si venderà infatti un software o una licenza, bensì la propria professionalità.

Un'aspetto interessante che ha iniziato a farsi strada negli ultimi tempi è quello legato allo sviluppo cofinanziato di progetti Open Source, che consiste nella possibilità per più aziende che abbiano la necessità di disporre di un determinato programma o driver di finanziarne lo sviluppo secondo un modello Open Source. Il contatto con i programmatori interessati avviene via Internet, utilizzando uno dei siti che coordinano questo tipo di attività. Una specie di borsa del software, con domanda e offerte. Nel caso un programmatore accetti la commessa, gli verrà offerto un contratto di sviluppo, esattamente come avviene per la commessa di un qualunque software personalizzato. Poiché non necessariamente offerente e programmatore si trovano nella stessa nazione, il sito che funge da contatto di solito si occupa di fare in modo che alle parti vengano fornite le opportune garanzie (pagamento del lavoro, garanzia di riuscita del progetto, ...). Il prodotto così realizzato verrà poi distribuito secondo la licenza GPL. I principali progetti di cofinanziamento del software sono Free Software Bazaar (<http://visar.csustan.edu/bazaar/>), CoSource.com (<http://www.cosource.com/>) e SourceXchange (<http://www.sourceexchange.com/>).

uso didattico.

**Make Minix freey available
and one of my biggest gripes
with it will disappear**

Alla fine, anche se i libri di Tanenbaum rimangono delle letture fondamentali nelle aule universitarie, la storia ha dato la sua parte di ragione anche a Linus.

Il "progetto GNU" ed il software libero

La grande intuizione di Linus fu quella di rendere pubblici i frutti del proprio lavoro. Se non l'avesse fatto, probabilmente Linux non sarebbe diventato quello che è. Infatti, spinti dall'entusiasmo per il lavoro dello studente finlandese, molti programmatori, anche parti-

colamente esperti, offrirono il loro contributo al progetto, aggiungendo nuove funzioni e aiutando a correggere i bug presenti.

In realtà l'idea che il software, o per meglio dire l'accesso al suo codice sorgente, inteso come frutto della conoscenza umana dovesse essere disponibile liberamente non era nuova. Già alcuni anni prima Richard Marshall Stallman, del MIT, aveva pubblicato un progetto, in cui si proponeva di

"realizzare un sistema operativo compatibile con UNIX, chiamato GNU (acronimo ricorsivo di "Gnu is Not Unix") e di renderlo disponibile liberamente in modo che ognuno che lo possa usare"

Il termine "free" utilizzato da Stallman ha però il doppio significato di "libero" e di "gratuito" e questa ambiguità è stata fonte di moltissimi problemi e fraintendimenti.

Il lettore infatti era portato a pensare che il "free software" fosse semplicemente un programma disponibile gratuitamente. Il concetto che Stallman voleva esprimere invece è completamente diverso: il free software è quello in cui la licenza con cui viene reso disponibile non limita la visibilità al codice sorgente e la possibilità delle persone di studiarlo ed eventualmente modificarlo per migliorarlo o adattarlo alle proprie specifiche esigenze. Esso deve poi essere ridistribuibile liberamente - sempre rendendo disponibile anche il codice sorgente ! - eventualmente anche facendosi pagare, se lo si ritiene opportuno. Nonostante tutti possano prelevare liberamente Linux spendendo soltanto il proprio tempo e i costi della connessione ad Internet, nessuno trova immorale che ci sia gente che produce dei CD con dentro Linux facendosi pagare per questo servizio. Lo stesso Stallman, nel "manifesto GNU", mise bene in evidenza tale possibilità. Spesso la "gratuità" è una conseguenza dell'"essere libero", ma essa non deve essere considerata la più importante.

In italiano per riferirsi al free software conviene usare il termine "software libero".

E' un peccato che nella lingua inglese non esista un vocabolo in grado di spiegare chiaramente e senza ambiguità questo concetto. Per questo motivo recentemente il software libero è stato ribattezzato "Open Source" (sorgente aperto), su suggerimento di Eric Raymond - la persona che convinse Netscape a pubblicare i sorgenti del Na-

vigator/Mozilla - allo scopo di dare maggiore visibilità a questo aspetto e di superare la pesante ambiguità di significato del termine "free". Inoltre la nuova definizione è più accettata a livello commerciale.

Potete trovare maggiori informazioni su GNU, compresa la traduzione in italiano del progetto, al sito <http://www.gnu.org/>.

La licenza GPL

Stallman elaborò un documento, la "GNU General Public License", in cui spiegava in modo formale, sotto forma di contratto, le condizioni secondo cui un programmatore, che rimane il detentore del copyright sul proprio lavoro, ne può concedere l'uso come software libero.

Anche se il più usato, GPL non è l'unico contratto applicabile al software libero. Ad esempio il sistema grafico X Window, utilizzato da Linux, viene distribuito secondo una licenza scritta appositamente dal MIT. Molto software di rete viene diffuso invece secondo la licenza BSD.

Il singolo programmatore, avendo il diritto d'autore sul proprio lavoro, può decidere secondo quale contratto renderne possibile l'utilizzo agli altri. Volendo fare del software libero, il contratto GNU è un'ottima scelta, già pronta all'uso e collaudata. Non è detto che sia la migliore in tutti i casi. Per una libreria di funzioni è, ad esempio, più adatta la sua variazione detta LGPL.

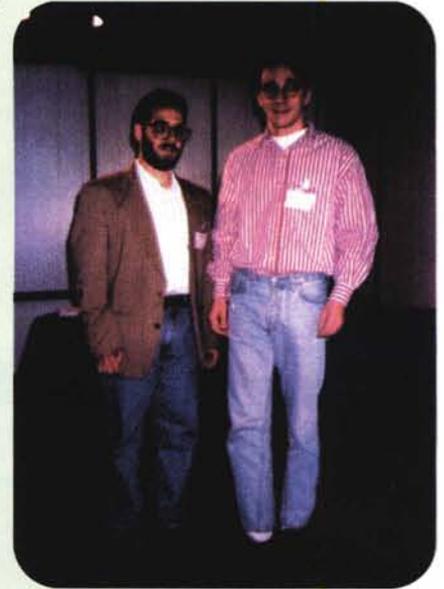
Bisogna prestare particolare attenzione a non confondere il software libero con il software di "pubblico dominio". Infatti in quest'ultimo caso il programmatore rinuncia espressamente a qualunque diritto su quanto ha scritto, al punto che chiunque potrebbe svegliarsi una mattina e mettersi sopra il proprio nome, cosa non possibile col software distribuito secondo la GPL. La licenza elaborata da GNU infatti stabilisce contemporaneamente sia i diritti dell'utente finale (possibilità di accesso al codice sorgente e di modifica, ...) che quelli del programmatore (mantenimento del proprio nome nel programma, sicurezza che non verrà cambiata la licenza d'uso, ...).

Linus Trovalds

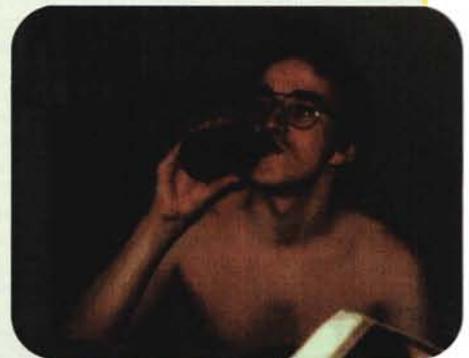
Una curiosità: il nome Linux non venne scelto da Torvalds, bensì dall'amministratore del sito ftp.nic.funet.fi, su cui Linus rendeva disponibili al pubblico le primissime versioni della sua creatura. Questo voler essere lontano dai riflettori rispecchia molto bene la figura di Linus come persona. Ho avuto la fortuna di conoscere lui e Tove (allora sua fidanzata e più tardi sua moglie e madre delle sue bambine) nel 1994, in occasione della Convention degli utenti Linux di I2U.

In occasione di quell'incontro mi sono reso conto di come il carattere schivo della persona non fosse un segno di orgoglio, bensì più probabilmente di timidezza come di paura nei confronti della notorietà che stava per travolgerlo. Anche adesso Linus, che nel frattempo si è trasferito negli Stati Uniti per lavorare alla Transmeta, riesce a fare in modo che il successo e la fama non intacchino la sua vita personale. Rispetto alla sua visita in Italia non sembra essere cambiato di molto: è disponibile a partecipare alle varie conferenze, capendo l'importanza del suo ruolo nei confronti degli altri e contemporaneamente non rinuncia ad essere un buon marito e padre di famiglia, portandosi dietro le due figlie e Tove e non dimenticandosi di dedicare una parte importante del suo tempo anche quando tutti gli altri lo vorrebbero solo per loro. Come sei anni fa.

Ricordo la fatica che abbiamo fatto io e Leo Sorge - che a quel tempo curava la rubrica di UNIX su MC - per convincere Linus a posare assieme al sottoscritto per la foto che vedete pubblicata in questo articolo. A quel tempo infatti c'era un alone di mistero (dovuto alla sua riservatezza) attorno alla persona di Linus Torvalds e le uniche tre foto che era possibile reperire in rete lo ritraevano solo davanti ad un bicchiere di birra (che da quel momento è diventata la bevanda ufficiale di ogni linuxiano). Mistero che si è ancora più complicato quando Linus ha lasciato l'Europa (e sottolineo con orgoglio che Linux è nato in Europa) per andare a lavorare al progetto segretissimo di Transmeta, il chip Crusoe di cui abbiamo parlato alcuni numeri or sono sulla rubrica.



Il vostro autore preferito assieme a Linus Torvalds



Le foto "storiche" di Linus



Per maggiori dettagli consiglio di vedere i siti di GNU (<http://www.gnu.org/>) e di Free Software Foundation (<http://www.fsf.org/>). Altri spunti interessanti li potrete trovare su <http://www.opensource.org/> e su <http://www.linux.it/GNU>. Una buona lettura è il libro di Tim O'Reilly "Open Sources", disponibile stampato o prelevabile da <http://www.oreilly.com/>. Le slide di una ottima presentazione a cura di Alessandro Rubini si possono trovare su <ftp://ftp.systemy.it/pub/develop/finenze.ps.gz>.

Una copia della licenza GPL può essere reperita all'interno di qualunque

programma distribuito secondo questa licenza (ad esempio nel file `/usr/src/linux/COPYING` all'interno dei sorgenti di Linux).

GNU+Linux = GNU/Linux

Linux è solamente il kernel, ovvero il nucleo del sistema operativo. Per farlo funzionare è necessario aggiungerci tutta una serie di programmi e utility accessorie.

Anche il progetto GNU prevedeva la

realizzazione di un kernel, Hurd, basato sulla filosofia del microkernel, nonché di tutti i programmi necessari a ricreare un ambiente simile e compatibile con UNIX: un editor, una shell, un compilatore C, un linker, un assembler e tutto quanto sarebbe servito per sviluppare programmi e farli girare nel nuovo ambiente.

Tutto il software GNU avrebbe dovuto essere disponibile liberamente assieme al suo codice sorgente e senza la necessità di firmare un NDA. In realtà la scrittura del codice iniziò dal basso (compilatore C, librerie e tool di sviluppo) e dalle cose più semplici (i program-

I vantaggi del software libero

Il dare accesso al codice sorgente e libertà d'uso era una novità sostanziale in un mondo in cui anche per utilizzare un semplice programma fornito come codice eseguibile occorreva firmare un NDA (Not Disclosure Agreement, contratto di non divulgazione), in quanto offriva la possibilità non solo di imparare studiando il lavoro degli altri, ma anche di adattare il programma alle proprie esigenze. In questo modo si era slegati dalle decisioni economiche e dai tempi di sviluppo di un particolare produttore di software. Inoltre eventuali problemi nel software potevano essere risolti in casa oppure facendo fare il lavoro da un programmatore di fiducia.

Il fatto di essere libero garantisce una veloce correzione di eventuali errori. Nel software è normale che l'utente che trova un problema o una limitazione usando un programma lo segnali all'autore dello stesso. Nel caso del software libero, avendo i sorgenti, è possibile correggere da soli eventuali bachi o aggiungere funzioni e poi mandare all'autore la modifica in modo che la incorpori nella prossima versione.

Per i programmi più diffusi la correzione di eventuali errori è una cosa "automatica" e veloce, dato il vasto numero di utenti in grado di dare il proprio contributo. Un programma libero ha infatti tanti potenziali sviluppatori e correttori di bug quanti sono i suoi utenti capaci di programmare, al contrario di un programma commerciale che ha al massimo qualche decina o qualche centinaio di persone che possono accedere al suo codice sorgente e modificarlo.

Nel caso del software commerciale poi bisogna rifare una nuova versione del software, riportarla nei negozi, verificare che gli utenti abbiano il diritto di usarla, ... nel caso del software libero è invece sufficiente rendere disponibile la nuova versione su un sito.

La velocità di sviluppo è probabilmente il vantaggio più interessante del software libero nei confronti di quello commerciale. E' questo il segreto del successo di Linux e dei grandi progetti di software libero.

La possibilità di far provare il programma a migliaia di persone, farsi aiutare nel trovare e correggere gli eventuali errori e rendere disponibili in tempi brevissimi le correzioni, porta ad un modello di sviluppo innovativo del software, in cui molte teste lavorano - in "parallelo" - meglio e in modo più efficiente rispetto al modello tradizionale, con una qualità del prodotto finale migliore.

Vi possono essere, è vero, dei problemi di coordinamento nel caso ad uno stesso programma lavorino contemporaneamente più persone, ma anche nel caso del software libero esistono degli strumenti (sia come metodi ingegneristici che come software, ad esempio il programma CVS) in grado di risolvere il problema in modo egregio. Ovviamente di tutti lo strumento più importante è Internet!

Per una discussione su questi due modelli, si legga lo storico articolo di Eric Raymond "The Cathedral and the Bazaar" (<http://sagan.earthspace.net/~esr/writings/cathedral-bazaar/>).

Il vantaggio del nuovo modello sono notevoli anche e specialmente per gli utenti finali, in particolare per le aziende, che, avendo libero accesso ai sorgenti, possono adattare i programmi alle loro esigenze. Purtroppo a distanza di quasi due decenni molti non l'hanno ancora capito e continuano a tentare di adattare la propria azienda e sé stessi al software invece che viceversa.

Abbiamo visto che in alcuni casi la GPL potrebbe non essere a licenza più adatta. Ad esempio essa in un punto specifica che ogni prodotto derivato da software protetto da GPL deve a sua volta essere distribuito secondo questa licenza, con tutti i vincoli che essa comporta, come la visibilità dei sorgenti. Tali vincoli possono non essere accettabili, ad esempio nel caso di software sviluppato da aziende in regime di concorrenza. In questo caso la soluzione deve essere quella di rinunciare ad utilizzare parti di sorgente GPL nel proprio programma, riscrivendosi a proprie spese il software necessario oppure acquistando la licenza d'uso di un prodotto o di una libreria commerciale. Utilizzare pezzi di software GPL in un programma commerciale è infatti una violazione dei termini della licenza, che ha valore legale.

Il fatto di poter studiare e riutilizzare il lavoro fatto da altri è sicuramente il vantaggio più grande, sia per il programmatore, che si trova parte del lavoro già fatta e può usare pezzi già pronti e collaudati, che per il committente, il quale paga solamente la parcella del consulente e non il costo di eventuali tool di sviluppo o royalty su librerie scritte da altri. Indubbiamente da tutto questo la professionalità del consulente ne guadagna moltissimo.

È sbagliato applicare al software libero lo stesso modello di remunerazione del software tradizionale, ma è ugualmente sbagliato presupporre che il lavoro utilizzato per scrivere software libero non debba essere pagato. Dovrebbe essere pagato ancora meglio, in quanto la scelta di usare e fare software libero comporta moltissimi vantaggi per l'azienda.

Se si decide di rendere pubblico il codice sorgente vi saranno inoltre altre persone che contribuiranno alla correzione degli errori ed alla manutenzione del programma, con ulteriori risparmio di tempo e di soldi per il committente.

Il poter vedere e studiare il codice scritto da altri è senz'altro il modo migliore per imparare a programmare. Il poterlo riutilizzare e modificare significa che non bisogna sprecare energie per reinventare cose che altri hanno già fatto e che le risorse risparmiate possono essere utilizzate per migliorare l'esistente o per inventare qualcosa di nuovo.

Moltissimi programmi oggi disponibili sono stati realizzati come software libero, oltre a Linux e a tutti i programmi originari del progetto GNU. Da programmi di pochissime righe a progetti impegnativi con migliaia di sviluppatori e centinaia di migliaia di utenti, come GNOME, sendmail, GIMP, TCL/TK, Perl, Apache, Mozilla, ...

mi accessori). Ci si trovò perciò con tutto pronto... fuorché il kernel, che non si sapeva quando sarebbe stato disponibile. Perché allora nel frattempo non utilizzare Linux ?

If the GNU kernel had been ready last spring, I'd not have bothered to even start my project: the fact is that it wasn't and still isn't. Linux wins heavily on points of being available now.

Nacque così un sistema operativo completo. Come sistema grafico venne scelto l'X Window System del MIT, in quanto anch'esso era distribuito con una licenza simile a quella di Stallman ed inoltre era lo standard de facto per la grafica nei sistemi UNIX. Il software di rete in gran parte deriva dallo UNIX di Berkeley (BSD).

Anche se tutti lo fanno, è perciò improprio definire un sistema completo solamente come "Linux". Sarebbe più giusto nel nome dare credito anche all'altro maggiore contributore e chiamarlo GNU/Linux (oppure anche GNU/X11/BSD/.../Linux).

È difficile dire quante siano le persone che, in un modo o nell'altro hanno collaborato in qualche modo a Linux durante tutti questi anni. Solo al kernel hanno dato apporto, mandando a Linus i propri contributi, centinaia di persone. Migliaia se si considerano anche le modifiche meno significative e le correzioni di bug. Inizialmente il kernel era gestito direttamente da Linus, che raccoglieva i contributi sviluppati dalle altre persone in modo indipendente e sceglieva quali inserire nel kernel. Ora il lavoro viene portato avanti in modo più preciso e coordinato. Linus ora lavora alla Transmeta e per questo ha passato il compito di manutentore della versione stabile ad Alan Cox.

Come dimenticare, oltre agli sviluppatori del kernel ed alle persone coinvolte nei progetti più importanti, tutti coloro che hanno contribuito alla crescita di Linux e del software libero anche solo con un piccolo programmino, con la correzione - o la semplice segnalazione - di un bug, scrivendo un pezzo di documentazione o un articolo divulgativo, oppure semplicemente parlando bene di Linux e magari aiutando un amico ad installarlo ? Forse i più importanti artefici del successo di Linux sono proprio gli utenti, che ne hanno capito l'importanza e che hanno avuto il coraggio di provare ad installarlo nel proprio computer.

Infine, come dimenticare che molto

Intervista ad Alessandro Rubini



Alessandro Rubini è probabilmente una delle figure di maggior rilievo nel mondo di Linux, non solo italiano. Nonostante la sua avversione per le tecnologie moderne, ha conseguito laurea e dottorato in ingegneria elettronica presso l'Università di Pavia. Il suo programma più usato è senz'altro Gpm, il server per il mouse per la console di Linux. È autore di diversi articoli per Linux Journal (<http://www.linuxjournal.com>) e Linux Magazine e del libro "Linux Device Drivers" per l'editore O'Reilly (<http://www.oreilly.com/catalog/linuxdrive/>). Alessandro è uno dei più accaniti sostenitori di GNU e della Free Software Foundation. Da lui ho imparato a chiamare Linux "GNU/Linux" e il free software "software libero". Nella vita divide il proprio tempo fra la famiglia e la sua attività di programmatore libero professionista. Potete contattarlo all'indirizzo e-mail rubini@linux.it.

D: da "free software" ad "Open Source". Potresti spiegare brevemente cosa ne pensi di questa "evoluzione" nell'idea del software libero ?

Non sono entusiasta del movimento "open source", perché pone al centro la qualità dei programmi e non la libertà degli utenti. Indubbiamente il termine "free" in inglese è ambiguo (significa anche "gratuito") e proprio per questo viene sempre più usato anche in italiano. Preferisco parlare di programmi liberi o di informatica libera, anche se purtroppo in certi contesti occorre spiegare dicendo "open source" all'americana.

D: ricordo che alcuni brani del "Manifesto GNU" di Richard Stallman erano delle vere e proprie prese di posizione quasi politiche, mentre ora i toni sono stati piuttosto smorzati e si è andati verso una visione meno utopistica.

La politica è l'arte di condurre una comunità (la "polis"). Il discorso di Stallman è un discorso politico. E il movimento del software libero non è mai stato utopistico, anche se ha raggiunto il grande pubblico solo negli ultimi tempi. Non noto, tra l'altro, una smorzatura dei toni. Quello che sta succedendo invece è il proliferare di licenze semi-libere, prodotte dalle aziende che vogliono apparire "open source" senza abbandonare la gestione monopolistica della produzione.

D: è forse la sconfitta della filosofia nei confronti dell'economia e di una visione più pratica della stessa medaglia ?

No. Non c'è niente di economicamente sbagliato nel produrre programmi liberi.

Si noti a questo proposito come l'informatica è l'unico campo in cui i produttori mantengono il controllo completo sul prodotto, permettendo di usarlo (con molti limiti) ma non di aggiustarlo, adattarlo ai propri bisogni, rivenderlo, farlo usare a terzi. Questo sopruso della libertà è tra l'altro tutelato dalla legge: se copio un libro pago una multa mentre se copio un programma vado in carcere. Oggigiorno l'informatica è sempre più importante per la vita di tutti ed è per questo che le persone si ribellano sempre più a questi soprusi e si sente parlare sempre più spesso di "Linux", etichetta usata per indicare tutto un mondo di programmi liberi.

D: che si possa vivere, anche dignitosamente, "facendo" (sono tue parole) Linux ed il software libero è ormai dimostrato. Le opportunità sono moltissime, ma come riuscire a conciliare progetti di un certo rilievo (kernel, gnome, ...), che necessitano di continuità con le necessità della vita. Non c'è il rischio che le distrazioni dovute ai bisogni economici facciano perdere di vista il progetto, magari per dedicarsi a fare delle consulenze che permettano di arrivare tranquillamente alla fine del mese ?

Naturalmente non tutti i programmatori devono dedicarsi a progetti importanti e non tutti sono abbastanza bravi per farlo. E in ogni caso adesso secondo me c'è molto più bisogno di consulenti che di programmatori che seguano i grossi progetti. Quello che serve perché le persone (i non informatici) sfruttino la libertà offerta dai programmi che usano non è Linux-3.0 o Emacs-25.67, è piuttosto la disponibilità di consulenti che sappiano scegliere le macchine e i pacchetti da installare, che li sappiano amministrare e personalizzare e che li sappiano aggiustare in caso di problemi.

Una delle obiezioni che vengono fatte ai programmi liberi è che "il codice sorgente è inu-

tile perchè chi non è del mestiere non lo sa usare". Sarebbe come dire "non mi interessa poter aprire il cofano dell'auto". Il punto è che chi non sa usare il sorgente può comunque rivolgersi ad un consulente, o un meccanico, che possa mettere le mani nel motore proprio perchè può aprire il cofano.

Per quanto riguarda i grossi progetti, sono convinto che chi è abbastanza bravo da essere produttivo in tali contesti trovi facilmente chi lo finanzia perchè possa dedicarci il suo tempo.

D: cosa ne pensi del cofinanziamento di progetti basati su software libero? Può essere questa la soluzione del problema?

Sicuramente è una via interessante. È importante però fare attenzione che chi finanzia non voglia asservire un progetto a scopi particolari, perchè farebbe più male che bene. È questo il motivo per cui Linus ha scelto di lavorare in una ditta che non si occupa specificamente di Linux: per evitare che gli interessi del suo datore di lavoro interferiscano con le sue scelte relative allo sviluppo del kernel.

D: a tuo vedere, quali sono le possibilità e le prospettive serie di lavoro per chi volesse lavorare nel campo del software libero in Italia?

Al momento non vedo grossi sbocchi. La diffusione dei programmi liberi in Italia è molto limitata: le aziende preferiscono comprare tutto dal monopolista di turno e la scuola non fa eccezione. Naturalmente perchè nasca un mercato occorre sia la domanda che l'offerta e se non ci sono le competenze nessun cliente andrà a cercarle. Credo che i primi sbocchi occupazionali seri siano nell'amministrazione di sistema e di rete, in particolare nel campo banche dati. Il vero lavoro di programmazione arriverà dopo, quando chi ha fatto il cambio "perchè Linux è gratuito" si renderà conto delle potenzialità di personalizzazione ed estensione.

D: quanti posti di lavoro ha creato il software libero? Di che qualità?

Confesso che non ne ho idea. Attualmente in Italia solo i fornitori di accesso a Internet fanno uso massiccio di sistemi liberi. Credo però che si apriranno grosse possibilità, perchè la flessibilità offerta dal sistema permette alle aziende alla pubblica amministrazione di dirottare le cifre adesso spese in macchinari e licenze verso la manutenzione e l'automazione delle procedure. E chiunque possa aumentare la propria produttività diminuendo i tempi di fermo-macchina è disposto ad investire in capitale umano.

D: in generale come vedi la situazione di Linux in Italia? Puoi descriverci degli esempi positivi, sia campo delle aziende che in quello degli appassionati?

In generale mi sembra abbastanza negativa: come al solito siamo gli ultimi in Europa. Vedo molto interesse nel campo delle aziende di automazione, interessate a sfruttare in prima persona la libertà offerta dai sistemi GNU/Linux: in molti stanno basando i propri prodotti su sistemi liberi o stanno pensando di farlo presto. Spesso però si fermano perchè non trovano chi gli dia una mano in questo salto, in quanto i consulenti a disposizione sono pochi in particolare nei settori di interesse per l'automazione. Mi sembra che la comunità degli appassionati sia abbastanza attiva e devo dire che in Italia abbiamo anche dei programmatori di spicco in molti progetti importanti; spesso queste persone sono finanziate da aziende straniere o non finanziate affatto, alcuni hanno anche lasciato il paese per poter lavorare.

D: si sente parlare delle aziende che "fanno" Linux come dei fenomeni che riescono a crescere molto velocemente e ad ottenere finanziamenti colossali da parte di grandi aziende (Red Hat) e singoli investitori (Linuxcare).

Probabilmente è un fenomeno temporaneo, dettato dalla moda e dall'entusiasmo per la "novità" (quanto in effetti la novità è fittizia: all'inizio tutti i programmi erano liberi).

D: secondo qualcuno Linux sembra essere una delle più grandi opportunità della cosiddetta "new economy"

Sicuramente è una grande opportunità per risanare una gestione storpiata dell'informatica, dando a ciascuno il controllo sulle proprie risorse informatiche. L'opportunità non è tanto per le grandi aziende quanto per le piccole società ed i singoli individui. Quando l'informazione è accessibile a tutti il suo sfruttamento è per forza distribuito sul territorio. Anche l'elettricità, come oggi l'elaborazione automatica dell'informazione, è stata una grande opportunità

del lavoro che è entrato a far parte di Linux era stato scritto già molti anni prima, magari da persone che non hanno avuto la fortuna di conoscere Linux? Se si fosse trattato di software non libero, probabilmente sarebbe semplicemente andato perduto... semplicemente... andato... perduto.

Le distribuzioni di Linux

All'inizio era compito del singolo utente procurarsi i vari pezzi necessari ed installarsi il proprio sistema. Subito però a qualcuno venne l'idea di fare una "distribuzione" che contenesse tutto il necessario assieme ad un programma che ne semplificasse l'installazione. Nacque così SLS, la prima distribuzione di Linux. A questa ne seguirono molte altre, fra cui Slackware, per un certo tempo la più utilizzata, Red Hat e Debian. Oggi tocca a queste due spartirsi la fama di più diffuse.

La prima è un prodotto commerciale, in cui l'azienda guadagna non tanto rivendendo Linux (la distribuzione è eventualmente disponibile gratuitamente su Internet o su CD a basso prezzo), ma per il fatto di fornire un prodotto testato, aggiornato e particolarmente adatto ad un'utenza professionale, con manuali stampati e supporto tecnico (che ovviamente non sono inclusi nella versione gratuita). Le software house commerciali, per motivi tecnici o per accordi di marketing, supportano i loro prodotti solo su certe distribuzioni.

La seconda è invece gestita e distribuita secondo lo spirito originario del progetto GNU e perciò è particolarmente rivolta ai programmatori (anche se ormai tutte le distribuzioni sono abbastanza equivalenti).

Altre distribuzioni importanti, ognuna con delle proprie particolari prerogative, sono Mandrake, S.u.S.E., Corel Linux, ...

La documentazione

La documentazione di Linux è gestita sotto forma di un progetto a parte, denominato LDP (Linux Documentation Project, a cui tutti possono dare il proprio contributo. Originariamente organizzato da Lars Wirzenius, il progetto è ora coordinato da Matt Welsh con l'aiuto di Michael K. Johnson. I documenti prodotti sono raccolti e resi disponibili nel sito <http://www.linuxdoc.com/>. Ov-

viamente quasi tutti sono originariamente scritti in inglese. I diversi Linux Users Group hanno tuttavia realizzato le traduzioni nelle diverse lingue nazionali. La traduzione in italiano è stata curata dal PLUTO (<http://www.pluto.linux.it/>), il più importante LUG del nostro paese ed è disponibile nel sito Internet del progetto ILDP (Italian Linux Documentation Project) <http://ildp.pluto.linux.it/>. Non tutto il materiale disponibile per Linux è stato tradotto e perciò eventuali volontari sono sempre ben accolti. Ovviamente il lavoro prodotto è disponibile a tutti secondo la licenza GPL.

LDP è composto da diversi libri, chiamati anche "Guide". Le principali sono: "Installation and Getting Started", che spiega come prelevare ed installare Linux, "The Linux System Administrator's Guide", che insegna ad amministrare un sistema, "The Linux Kernel Hackers' Guide", un libro su come programmare e modificare il kernel di Linux e "The Linux Network Administration Guide", che insegna tutto ciò che serve sapere per installare, configurare ed usare il software di rete.

I Linux HOWTO sono invece documenti che descrivono molto in dettaglio determinati aspetti della configurazione e dell'uso di Linux. Essi sono come delle vere e proprie guide o tutorial che spiegano passo-passo ad esempio come configurare il PPP, piuttosto che un server WWW, una funzionalità avanzata di networking o una scheda audio. Sono sicuramente i documenti di riferimento per ogni utente di Linux, da quello alle prime armi fino al più esperto. E' consigliabile consultare gli HOWTO prima di ogni altro documento, in quanto, oltre ad essere ben curati ed aggiornati, contengono spesso riferimenti ad altri siti o documenti.

Gli HOWTO e le principali guide del progetto LDP sono anche disponibili come libri stampati e rilegati, come "The Linux Bible" o "Dr. Linux". La traduzione in italiano a cura del PLUTO è stata stampata da Apogeo.

Rispetto alla versione originale, ILDP aggiunge anche i seguenti documenti:

✓ Italian-HOWTO, di Marco "gaio" Gaiarin, che spiega tutto ciò che c'è da sapere per usare Linux in Italia, dalla personalizzazione della tastiera e del font di caratteri fino agli indirizzi dei negozi che vendono CD di Linux.

✓ AppuntiLinux, un librone di 2000 pagine scritto in italiano, da Daniele Giacomini e disponibile nel sito di ILDP, che spiega praticamente ogni aspetto

di sviluppo. Per l'umanità, non per poche aziende.

D: Non si rischia in questo modo alla fine di snaturare l'idea filosofica originaria di Linux e del software libero ?

Non vedo perchè. A me va benissimo le aziende lucrino sulla propria attività. Anche i miei clienti mi pagano perchè io scriva loro i programmi. Non vedo niente di male nel fatto che qualcuno finanzia qualcun altro perchè crede nelle sue abilità.

D: Cosa ne pensi delle grandi aziende che si basano sul software libero (Red Hat, Corel, ... e altre a tua scelta) ?

Non ho un giudizio "generale" perchè ognuno si pone sul mercato in modo diverso. Nella maggior parte dei casi sono persone che fanno il loro lavoro. Non è un problema per me che RedHat (l'hai nominata tu) venda le scatole colorate. Il software che distribuiscono non è per questo meno libero: si può copiare, modificare, rivendere. Quello che viene venduto è il disco, la scatola, il manuale, e l'assistenza per l'installazione. Non i programmi. Tra l'altro, in quelle scatole c'è anche qualcosa che ho scritto io: a me conviene che qualcun altro si occupi della commercializzazione, perchè aiuta a diffondere il mio lavoro in modo che i potenziali clienti possano trovarmi. Ovviamente non sono molto contento quando vedo programmi liberi e programmi proprietari mescolati nella stessa scatola e nello stesso CD. È nel loro diritto farlo, ma è anche nel mio diritto non comprare da loro; non arrivo al punto di condannarli, però, perchè nella situazione attuale anche le distribuzioni parzialmente proprietarie sono positive rispetto a quelle totalmente proprietarie cui molte persone sono abituate.

D: cosa invidi ad uno sviluppatore Windows e cosa credi che lui debba invidiare nei confronti di uno sviluppatore Linux ?

Cos'è Windows?

di Linux. Un testo da avere assolutamente !

✓ Pluto Journal, la rivista mensile del gruppo, interamente in italiano. Oltre al resoconto sulle diverse attività del gruppo, come il meeting annuale,

offre spazio ad articoli tecnici e ad esperienze raccontate da persone che

le. I più importanti e aggiornati sono i seguenti:

<http://www.slashdot.com>
<http://www.freshmeat.com>
<http://rpmfind.net>

News su Linux
Programmi catalogati
Archivio (vastissimo) di programmi in formato RPM
Uno dei migliori siti su Linux
ZioBudda Home Page (in italiano)

<http://www.linux.com>
<http://www.ziobudda.net/>

Altri siti interessanti sono:

<http://www.linux.org>
<http://www.linuxdoc.com>
<http://www.fsf.org>
<http://www.gnu.org>
<http://www.lwn.net>
<http://www.linuxjournal.com>
<http://www.linuxgazette.com>
<http://www.pluto.linux.it>

Il sito ufficiale di Linux International
Documentazione di Linux (LDP)
Free Software Foundation
The GNU Project
LinuxWeeklyNews
Linux Journal
Linux Gazette
Sito ufficiale del PLUTO, rivista Pluto Journal, in Italiano
Documentazione di Linux in italiano (ILDLP)
User Friendly (striscia di fumetti su Linux)

<http://ildp.linux.it>
<http://www.userfriendly.org/>

si sono avvicinate per la prima volta a Linux, oltre che suggerimenti e link interessanti.

Vi è poi una miriade di siti Internet che trattano l'argomento Linux. Conviene iniziare affidandosi ad un buon porta-

Ringrazio, in ordine sparso, per l'aiuto che mi hanno fornito per questo speciale: Christopher R. Gabriel, Raffaello de Masi, Lorenzo Zanetti, Edoardo Zanetti e Alessandro Rubini. L'autore della foto del sottoscritto assieme a Linus Torvalds è Leo Sorge.