

Software Libero – Società Libera



Francesco Cordisco

Tesina esame di maturità 2008

Liceo Tecnico VM 2007/2008

ITSOS “Marie Curie” Cernusco sul Naviglio (MI)

Indice

Introduzione.....	3
Definizione software libero.....	4
Perché “Libero”.....	5
Supporto economico e la Free Software Foundation	7
Richard Stallman	8
Note.....	8
Biografia.....	8
Primi Anni.....	8
La cultura hacker del MIT declina.....	9
Progetto GNU.....	11
Attivismo.....	12
Terminologia.....	13
Vita personale.....	14
Progetto GNU.....	15
Manifesto GNU.....	15
Perché devo scrivere GNU.....	16
Perché GNU sarà compatibile con Unix.....	16
Come sarà reso disponibile GNU.....	16
Perché molti altri programmatori desiderano essere d'aiuto.....	16
Come si può contribuire.....	17
Perché tutti gli utenti dei computer ne trarranno beneficio.....	17
Kernel Linux.....	18
Licenze.....	19
Introduzione.....	19
Copyleft.....	19
GNU General Public License	20
GNU Lesser General Public License.....	20
GNU Free Documentation License.....	20
Pericolo dei brevetti sul software.....	21
Open Source.....	22
Introduzione.....	22
Perché L’“Open Source” Non Ha Centrato L’Obiettivo Del Software Libero.....	22
Progetto nupedia e wikimedia.....	24
Conclusioni.....	25
Software libero, economia e società.....	25
Bibliografia:.....	26
Sitografia:.....	26

Introduzione

In quest'epoca di grande sviluppo della tecnologia informatica, le grandi produzioni industriali monopolizzano il mercato dell'hardware e del software dimenticando lo spirito di collaborazione che esisteva agli albori dell'informatica.

Al laboratorio di intelligenza artificiale del MIT di Boston dagli anni '60, un gruppo di ricercatori collaborava per produrre i primi software per i grandi elaboratori di quel tempo, lo scambio di idee, di codici sorgente era una azione semplice e naturale. Non esistevano limitazioni di accesso, diritti o proprietà, la comunità del laboratorio aveva lo scopo di fare ricerca, produrre e migliorare software.

Negli anni '80 alcuni programmatori pensarono che vendendo programmi con codice sorgente non disponibile sarebbero potuti diventare molti ricchi. Nacquero le prime grandi industrie che assumevano sempre più programmatori disposti a produrre software proprietari.

La comunità che si era creata all'interno del laboratorio si distrusse, molti di loro andarono a lavorare per queste aziende che probabilmente li avrebbero fatti arricchire, togliendogli però, le libertà che avevano perseguito fino ad allora.

Richard Stallman, uno dei programmatori del MIT, non accettò di lavorare per produrre software proprietario e scontratosi con programmi proprietari che gli toglievano le libertà a cui era abituato, decise che avrebbe potuto, con l'aiuto di altri, creare un sistema operativo non proprietario.

Egli non accettava l'idea di non poter condividere il suo lavoro con altri, voleva anzi vivere una vita, anche oltre l'ambito lavorativo, che si basava sulla libera condivisione del sapere.

Tuttora vuole produrre programmi che possano essere usati, modificati, copiati e redistribuiti liberamente con o senza compenso.

Quello che i grandi monopoli hanno fatto e continuano a fare è costruire barriere tra programmatori e utenti finali, togliendogli la libertà di collaborazione che in questi ultimi anni sembra dimenticata.

Nel 1984 Stallman lanciò il progetto GNU e diede vita al movimento del software libero e ne è il principale esponente, in questo modo una parte della comunità del laboratorio si è salvata per diffondere i propri principi a tutte le persone che siano utilizzatori di computer oppure no.

Infatti i principi di collaborazione e condivisione sono alla base del movimento del software libero ma dovrebbero essere adattati e perseguiti in tutti gli aspetti della vita.

Definizione software libero

L'espressione "software libero" si riferisce alla libertà dell'utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software.

Le quattro principali libertà che un utente deve avere sono:

- Libertà 0:** Libertà di eseguire il programma, per qualsiasi scopo.
- Libertà 1:** Libertà di studiare il funzionamento del programma e adattarlo alle proprie necessità. L'accesso al codice sorgente è un prerequisito. **Libertà di aiutare se stessi**
- Libertà 2:** Libertà di ridistribuire copie in modo da aiutare il prossimo. **Libertà di aiutare i tuoi vicini**
- Libertà 3:** Libertà di modificare il programma e distribuirne le versioni, così che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente è un prerequisito. **Libertà di aiutare la comunità**

Un programma è software libero se l'utente ha tutte queste libertà. In particolare, se è libero di ridistribuire copie, con o senza modifiche, gratuitamente o facendo pagare le spese di distribuzione a chiunque ed ovunque.

Per tutte queste attività non bisogna chiedere o pagare nessun permesso o sottostare a delle licenze che violano anche in parte una delle libertà.

-**La libertà di usare un programma** permette a qualsiasi persona o organizzazione di utilizzarlo per qualsiasi attività. Lo scopo dell'utilizzo di un programma deve essere quello dell'utente e non quello dello sviluppatore.

-**La libertà di ridistribuire copie** è molto importante per il miglioramento e la diffusione del software. Per permettere modifiche e pubblicare versioni migliorate qualsiasi copia anche modificata, deve includere le forme binarie o eseguibili del programma e anche il codice sorgente, tutti requisiti necessari per il software libero.

-**Software libero non vuol dire non-commerciale.** Vendere software libero e trarne profitto è una attività legale. Tutto il software libero può essere commercializzato da chiunque e per chiunque a qualsiasi prezzo. Vendendo software proprietario l'utente non potrà eseguire un programma senza pagarlo, se questo prezzo sarà alto alcuni utenti non potranno comprarlo e quindi usarlo. Il prezzo del software libero non sarà mai troppo alto poiché sarà il mercato a deciderlo, se si vende un programma a un prezzo troppo elevato nessuno lo comprerà poiché esistono molti altri modi per procurarselo.

Quando si parla di software libero, è bene concentrare l'attenzione sulla libertà del software e non sul prezzo, quindi è meglio evitare di usare espressioni come "vendere software" ma usare piuttosto "distribuire software dietro compenso".

-**Le leggi sul controllo delle esportazioni e le sanzioni sul commercio** possono limitare la libertà di distribuire copie di programmi verso paesi esteri. Per questo è importante che i programmatori non impongano restrizioni d'uso del programma. In tal modo, le restrizioni non influiranno sulle attività e sulle persone al di fuori della giurisdizione degli stati che applicano tali restrizioni.

Per determinare se un software può essere considerato libero è necessaria un'attenta analisi, per facilitarla sono state create le licenze GPL e LGPL. Però si può chiaramente capire che se una licenza include restrizioni irragionevoli sulle principali libertà essa verrà rifiutata.

Perché “Libero”

“La tecnologia dell'informazione digitale contribuisce al progresso mondiale rendendo più facile copiare e modificare le informazioni. I computer promettono di rendere questo più facile per tutti noi.”

(Richard Stallman)

I proprietari di software non libero con il sistema del diritto d'autore limitano e non permettono di sfruttare i potenziali vantaggi che il software può portare. Si crea una frattura fra utente e programmatore.

Il sistema del diritto d'autore è difficilmente applicabile al software poiché toglie libertà agli utenti, al contrario questo sistema nato e applicato alla stampa pone limiti ai produttori di massa non ai singoli lettori. Con la digitalizzazione delle informazioni è molto più facile condividere qualsiasi cosa e per questo il sistema del diritto d'autore che è poco adatto, viene fatto accettare dalla Software Publishers Association (SPA) attraverso modi poco corretti:

- La propaganda convince l'utilizzatore che è sbagliato condividere software e aiutare gli amici.
- Richieste insistenti di informatori che forniscano informazioni su compagni di lavoro e colleghi.
- Incursioni (con l'aiuto della polizia) in scuole e uffici, durante le quali viene detto alle persone che devono provare che non fanno copie illegali.
- Citazione in giudizio (da parte del governo degli Stati Uniti, su richiesta della SPA) di persone come David LaMacchia del MIT, non per aver copiato software, ma per avere lasciato senza sorveglianza strumenti per la copia e per non averne censurato l'uso.

Tutte queste quattro pratiche vengono usate per mantenere il profitto dei proprietari di software che quindi **contrastano la diffusione delle informazioni e della conoscenza**.

I principali modi per mantenere il controllo dell'informazione da parte dei proprietari di software sono:

- **L'uso dei nomi** – I proprietari trattano il software come gli oggetti materiali che non possono essere copiati. Per questo secondo loro esiste la “pirateria”, il “furto”, la “proprietà intellettuale” e il “danneggiamento” che creano nelle persone una visione sbagliata di condivisione del software.
- **Esagerazioni** – I proprietari di software esagerano nel dire che subiscono danni quando una persona crea una copia per qualcun altro, in quanto il proprietario immagina che tutte le persone che ricevono una copia avrebbero comprato il suo software. Invece molti possibili acquirenti considerati i costi, non ne avrebbero comprato una copia.
- **La legge** – I proprietari di software basano le loro idee sulle leggi vigenti ma è evidente che non è la legge che decide cosa è giusto e cosa è sbagliato. Ad esempio ogni americano dovrebbe sapere che, quaranta anni fa, era contro la legge, in molti stati, che una persona di colore si sedesse in un autobus nei posti anteriori; ma solo i razzisti avrebbero detto che era sbagliato sedersi lì.

- **Diritti naturali**

Gli autori di software ma soprattutto le società per cui lavorano, spesso rivendicano un legame con i propri programmi che deve permettere loro di porre i propri interessi sopra a quelli di qualsiasi utilizzatore. Questo legame è tutelato da copyright, applicato a ogni programma.

Invece è opportuno rivendicare i propri diritti naturali se si parla di oggetti materiali che hanno un “ciclo di vita” come un'automobile. Se un ladro la ruba egli danneggerà il proprietario, ma se un programma creato della stessa persona viene copiato e magari migliorato questo non farà danni. Se una comunità si unisce per migliorare quel programma il primo creatore non ha il diritto di negare le modifiche e lo sviluppo.

- **Economia**

Questo è l'unico argomento che parte dal problema di soddisfare gli utenti, infatti pagando le persone probabilmente si produrrà più software. Questo argomento parte però dall'assunto che la "produzione di software" sia ciò che vogliamo, sia che il software abbia proprietari sia che non li abbia. Infatti gli oggetti materiali possono avere un proprietario, ma questo non pregiudica ciò che è o ciò che ci si può fare con un oggetto che viene acquistato. Invece se un programma ha un proprietario egli condiziona ciò che è e ciò che ci si può fare poiché vengono poste delle limitazioni nell'utilizzo che causano un intangibile inquinamento etico producendo software di cui l'uomo non ha bisogno.

La società ha bisogno invece di una informazione che sia realmente disponibile ai suoi cittadini - per esempio programmi che si possano leggere, correggere, adattare e migliorare, non soltanto usare. Ma quello che viene consegnato di solito dai proprietari del software è una scatola nera che non si può studiare o cambiare.

“La società ha anche bisogno di libertà. Quando un programma ha un proprietario, gli utenti perdono la libertà di controllare parte della loro stessa vita.

Ma soprattutto la società ha bisogno di stimolare nei propri cittadini lo spirito di cooperazione volontaria. Quando i proprietari del software ci dicono che aiutare i nostri vicini in maniera naturale è "pirateria", essi inquinano lo spirito civico della nostra società.

Questo è il motivo per cui diciamo che il software libero è una questione di libertà, non di prezzo.

Come utenti di computer oggi ci si può trovare ad usare un programma proprietario. Se un amico chiede una copia sarebbe sbagliato rifiutare. La cooperazione è più importante del diritto d'autore. Ma una cooperazione nascosta e segreta non contribuisce a rendere giusta la società. Una persona dovrebbe aspirare a vivere una vita onesta, apertamente e con fierezza, e questo comporta dire "No" al software proprietario.

*Meritate di poter cooperare apertamente e liberamente con altre persone che usano software.
Meritate di poter imparare come funziona il software e con esso di insegnare ai vostri studenti.
Meritate di poter assumere il vostro programmatore favorito per aggiustarlo quando non funziona.*

Meritate il software libero.”

(Richard Stallman)

Supporto economico e la Free Software Foundation

Pur rimanendo sbagliato, l'aspetto economico è importante per permettere un maggiore sviluppo del software, infatti la gran parte del software libero prodotto è stato creato grazie alla passione di programmatori di tutto il mondo che si sono uniti in strette collaborazioni. Ma per permettere maggiore impegno è necessario raccogliere fondi.

Da dieci anni gli sviluppatori di software libero cercano in vari modi di dedicarsi completamente alla programmazione, ad esempio Richard Stallman si è guadagnato da vivere per anni con miglioramenti a richiesta del software libero che aveva scritto. Ciascun miglioramento è stato aggiunto alla versione standard rilasciata e reso così disponibile al pubblico. I clienti lo pagavano perché lavorasse sui miglioramenti che volevano loro.



Col crescere del numero di partecipanti al progetto GNU fu deciso di fondare nel 1985 la Free Software Foundation (FSF), un'organizzazione senza scopi di lucro per permettere lo sviluppo di software libero. La FSF svolge diverse attività ma lo scopo principale è raccogliere fondi per sostenere il più possibile il progetto.

Un grave problema che la FSF cerca di arginare è la mescolanza del software libero con quello proprietario poiché la campagna principale adottata è quella per il software libero e contro quello proprietario.

Per risolvere questo problema, la FSF ha cominciato nel 1999 a lavorare alla Free Software Directory, un database di Software Libero, ed affidò il suo mantenimento a Janet Casey.

Per ogni registrazione nella lista viene verificata la licenza e il codice sorgente per essere sicuri che la licenza sia effettivamente ciò che appare; perciò gli utenti possono fidarsi delle informazioni inserite nella Free Software Directory. Nell'Aprile 2003 l'Organizzazione delle Nazioni Unite per l'Istruzione, la Scienza e la Cultura (UNESCO) si è unita a questa attività della FSF e la sostiene ufficialmente, così ora si chiama "FSF/UNESCO Free Software Directory."

Ma la FSF può raccogliere fondi grazie anche alla vendita di CD-ROM, T-shirt, manuali, e confezioni Deluxe di GNU (che gli utenti sono liberi di copiare e modificare), e anche con donazioni.

Un gruppo di imprese che comprende Intel, Motorola, Texas Instruments e Analog Devices si sono unite per finanziare il continuo sviluppo del compilatore libero GNU per il linguaggio C. Inoltre il compilatore GNU per il linguaggio Ada fu finanziato dalla US Air Force, che riteneva questa la modalità di spesa più efficace per ottenere un compilatore di alta qualità, ora la sua manutenzione è finanziata commercialmente.

In futuro per continuare a raccogliere fondi e svolgere un'attività commerciale grazie e a favore del software libero nasceranno aziende che offriranno consulenza agli utenti, correggeranno eventuali errori e aggiungeranno particolari funzionalità richieste dall'utente.

Richard Stallman

Note

La biografia che riporto è la traduzione che ho fatto da wikipedia inglese. Questo lavoro è possibile trovarlo anche su wikipedia italiana che non aveva ancora una biografia abbastanza dettagliata di Richard Stallman. Rendendo il mio lavoro disponibile a chiunque, ho messo in pratica il principio su cui si basa il software libero: la condivisione.

Biografia

Richard Matthew Stallman nacque il 16 marzo 1953, è un attivista americano del movimento del software libero, hacker e programmatore. Nel settembre del 1983 lanciò il progetto GNU per creare un sistema operativo libero simile a Unix, egli è il principale organizzatore e creatore. Con il lancio del Progetto GNU, il movimento del software libero prese vita e nell'ottobre del 1985 fondò la Free Software Foundation(FSF).

Stallman fu il pioniere del concetto di copyleft ed è il maggiore autore di molte licenze copyleft compresa la GNU General Public License(GPL), la licenza per software libero più usata.



Dalla metà degli anni '90 Stallman spende molto del suo tempo sostenendo il software libero come anche la campagna contro il software brevettato e ciò che a lui sembra un' eccessiva estensione delle leggi sui copyright. Stallman ha anche sviluppato un numero di parti di software usati ampiamente, compreso Emacs, la GNU Compiler Collection e la GNU Debugger.

Primi Anni

Stallman nacque da Daniel Stallman e Alice Lippman nel 1953 a New York City, New York. Assunto dalla IBM New York Scientific Center, Stallman passò l'estate dopo il diploma di scuola superiore scrivendo il suo primo programma, un processore per il linguaggio di programmazione PL/I sull' IBM 360. Durante questo periodo, Stallman fu anche un volontario assistente di laboratorio al dipartimento di biologia alla Rockefeller University. Sebbene si stesse già indirizzando verso una carriera da matematico o fisico, il suo professore alla Rockefeller pensò che avrebbe avuto un futuro come biologo. Nel giugno del 1971 al primo anno da studente all'università di Harvard, Stallman diventò un programmatore al laboratorio IA (Intelligenza Artificiale) del MIT. Qui prese parte alla comunità degli hacker, dove era solitamente conosciuto con le iniziali "RMS" (che era nome dell'account del suo computer). Nella prima edizione del dizionario dell' Hacker scrisse: "'Richard Stallman' è solo il mio nome mondano; tu puoi chiamarmi 'rms'".

Stallman si laureò con lode (magna cum laude) ad Harvard in Fisica nel 1974. In seguito, Stallman si iscrisse da laureato in fisica al MIT, ma abbandonò i suoi studi universitari rimanendo un programmatore al laboratorio di intelligenza artificiale del MIT.

Alla fine del suo primo anno nel programma di laurea, Stallman si procurò una ferita al ginocchio che pose fine alla sua più grande gioia: la partecipazione ai balli folk internazionali, e con essa l'opportunità di socializzare con l'altro sesso.

La disperazione che ne derivò culminò in un ritiro dalla vita sociale, ma trovò conforto in un'intensa concentrazione nel campo in cui i suoi risultati lo rendevano fiero: la programmazione.

Il suo dottorato di ricerca in fisica diventò una limitazione in questa vocazione; comunque Stallman è stato premiato con 6 dottorati onorari e 2 onorarie cattedre universitarie.

Mentre era studente al MIT, Stallman pubblicò una relazione su "Truth Maintenance System" (revisione controllata) intitolata "dependency-directed backtracking" con Gerald Jay Sussman. Questa relazione fu un primo lavoro sul problema dell' "Intelligent backtracking" nel "constraint satisfaction problems" (soddisfazione dei vincoli). Dal 2003, la tecnica introdotta da Stallman e Sussman è ancora il metodo più generale e potente di "Intelligent backtracking". In questa relazione fu anche introdotta la tecnica "constraint recording" (memorizzazione dei vincoli) in cui risultati parziali di una ricerca sono registrati per un successivo riutilizzo.

Come hacker al laboratorio di intelligenza artificiale del MIT, Stallman lavorò su progetti di software come TECO, Emacs, e il sistema operativo per Lisp Machine.

Sarebbe diventato fortemente critico rispetto all'accesso ristretto ai computer dei laboratori. Quando il laboratorio per le scienze del computer (LCS) del MIT installò un sistema controllato da password nel 1977, Stallman trovò un modo per decriptare le password e inviò agli utenti un messaggio contenente la loro password decodificata, con il suggerimento di cambiarla con una stringa vuota (che significa non immettere password), per permettere nuovamente l'accesso anonimo al sistema. Circa il 20% degli utenti seguirono il suo consiglio, sebbene le password alla fine prevalsero. Stallman si vantò del successo nella sua campagna per molti anni a seguire.

La cultura hacker del MIT declina

Nei tardi anni '70 e primi anni '80, la cultura hacker che Stallman aveva fatto prosperare cominciò a frammentarsi. Per evitare che il software fosse usato sui computer dei concorrenti, la maggior parte dei produttori smise di distribuire il codice sorgente e cominciò a usare copyright e licenze restrittive per il software, per limitarne o proibirne la copia e la redistribuzione. Tale proprietà dei software già esisteva prima, e divenne chiaro che sarebbe diventata la norma. Questo passaggio alle caratteristiche legali del software può essere considerato come una conseguenza scatenata da un atto legale (U.S.Copyright Act) del 1976, come dichiarato da Brewster Kahle, compagno di Stallman al MIT.

Quando Brian Reid nel 1979 mise "time bombs" in Scribe per limitare l'accesso senza licenza al software, Stallman lo considerò "un crimine contro l'umanità". Egli chiarì, anni dopo, che quello che lui considera un crimine è ostacolare la libertà dell'utente e non far pagare i software.

Nel 1980, a Stallman e ad alcuni altri hacker del laboratorio di intelligenza artificiale fu rifiutato il codice sorgente del software per la stampante laser Xerox 9700 ("Dover"), la prima del settore. Stallman aveva modificato il software su una vecchia stampante (la XGP, Xerographic Printer), così veniva mandato elettronicamente un messaggio all'utente quando veniva stampata un suo lavoro, e avrebbe mandato un messaggio a tutti gli utenti collegati quando una stampante era congestionata. Non poter aggiungere questo servizio alla stampante Dover era l'inconveniente maggiore, siccome la stampante era su un piano differente da quello della maggior parte degli utenti. Questa esperienza convinse Stallman che le persone hanno bisogno di essere libere di modificare il software che usano.

Nel 1980, Richard Greenblatt, un compagno hacker del laboratorio AI, fondò Lisp Machines, Inc. (LMI) per commercializzare Lisp machines, che lui e Tom Knight avevano creato in laboratorio.

Greenblatt rifiutò investimenti esterni, credendo che i profitti dalla costruzione e dalla vendita di un po' di macchine poteva essere reinvestito con profitto per la crescita della società.

Al contrario, gli altri hackers ritenevano che fosse meglio l'avvicinamento di una venture capital finanziatrice.

Poichè non fu possibile raggiungere alcun accordo, hackers dell'ultimo campo fondarono Symbolics, con l'aiuto di Russ Noftsker, un amministratore del laboratorio AI (AI Lab).

Symbolics reclutò la maggior parte degli hackers rimasti incluso l'insigne hacker Bill Gosper, che poi lasciò il laboratorio AI. Symbolics costrinse Greenblatt anche a dimettersi citando le politiche del MIT. Mentre entrambe le compagnie rilasciavano software proprietario, Stallman credeva che LMI, a differenza di Symbolics, avesse provato ad evitare di danneggiare la comunità del laboratorio. Per due anni, dal 1982 alla fine del 1983, Stallman lavorò da solo per clonare la produzione dei programmatori di Symbolics, con lo scopo di evitare che essi ottenessero un monopolio sui computer del laboratorio.

Comunque, fu l'ultimo della sua generazione di hacker nel laboratorio. Rifiutò un futuro dove avrebbe dovuto firmare accordi di non divulgazione per non condividere codici sorgente o informazioni tecniche con altri sviluppatori di software e compiere altre azioni che considerava tradimenti dei suoi principi. Scelse invece di condividere il suo lavoro con altri, in quello che considerava un classico spirito di collaborazione. Sebbene Stallman non partecipò negli anni '60 all'era della controcultura, fu ispirato dal suo rifiuto della ricerca della ricchezza come primo obiettivo di vita.

Stallman sostiene che gli utenti di software dovrebbero avere la libertà di “condividere con i loro vicini” e di essere in grado di studiare e fare modifiche al software che loro usano.

Ha ripetutamente detto che i tentativi dei venditori di software proprietario di proibire questi atti sono “antisociali” e “non etici”.

La frase “il software vuole essere libero” gli è spesso non correttamente attribuita, e Stallman sostiene che questa è una descrizione inesatta della sua filosofia. Egli sostiene che la libertà è vitale per il bene degli utenti e della società come un valore morale, e non puramente per ragioni pragmatiche come sviluppare in qualche modo un software tecnicamente superiore.

Nel gennaio 1984, Stallman lasciò il suo lavoro al MIT per lavorare a tempo pieno sul progetto GNU, che aveva annunciato nel settembre del 1983.

Progetto GNU

Stallman annunciò il progetto per il sistema operativo GNU nel settembre 1983 su molte mailing list ARPAnet e USENET.

Nel 1985, Stallman pubblicò il manifesto GNU, che descriveva le sue motivazioni per creare un sistema operativo libero chiamato GNU, che sarebbe stato compatibile con Unix. Il nome GNU è un acronimo ricorsivo per GNU's Not Unix (GNU non è Unix). Poco dopo, diede vita a una corporazione no profit chiamata Free Software Foundation per impiegare programmatori di software libero e fornire un' infrastruttura legale per il movimento del software libero.

Stallman è il presidente non stipendiato della FSF, un'organizzazione no-profit 501(c)(3) fondata nel Massachusetts.

Nel 1985, Stallman inventò e rese popolare il concetto di copyleft, un meccanismo legale per proteggere i diritti di modifica e redistribuzione per il software libero. Fu inizialmente implementato nella GNU Emacs General Public License, e nel 1989 il primo programma indipendente sotto licenza GPL fu rilasciato.

Da allora, gran parte del sistema GNU è stato completato. Stallman fu responsabile di aver contribuito con molti strumenti necessari, inclusi un editor di testo, un compilatore, un debugger, un build automator (un metodo automatico di compilazione da codice sorgente a codice binario).

La notevole eccezione fu un kernel. Nel 1990, membri del progetto GNU cominciarono un kernel chiamato GNU Hurd, che deve ancora raggiungere il livello di maturità richiesto per l'uso diffuso.

Nel 1991, Linus Torvalds, uno studente finlandese, usò gli strumenti di sviluppo GNU per produrre il kernel Linux. I programmi esistenti del progetto GNU furono prontamente portati a funzionare sulla piattaforma risultante, molti sorgenti usano il nome "Linux" per riferirsi al sistema operativo general-purpose così formato.

Questa controversia sul nome è da tempo presente nella comunità del software libero. Stallman sostiene che non usare "GNU" nel nome del sistema operativo sminuisce ingiustamente il valore del progetto GNU e nuoce alla sostenibilità del movimento del software libero rompendo il collegamento tra il software e la filosofia del software libero del progetto GNU.

Le influenze di Stallman sulla cultura hacker includono il nome POSIX e l'editor Emacs. Sui sistemi UNIX, la popolarità di GNU Emacs fece concorrenza all'altro editor vi, provocando una guerra fra editor. Stallman prese posizione su questo canonizzando scherzosamente se stesso come "St. IGNUcius" della Chiesa di Emacs e riconoscendo che "vi vi vi è l' editor del diavolo", mentre "usare una versione libera di vi non è un peccato; è una penitenza".

Un numero di sviluppatori vide Stallman come un personaggio difficile con cui lavorare da un punto di vista politico, interpersonale o tecnico. Intorno al 1992, sviluppatori alla Lucid Inc. facendo il loro lavoro su Emacs si scontrarono con Stallman e alla fine fecero un fork sul software. Il loro fork più tardi diventò XEmacs. Un archivio email pubblicato da Jamie Zawinski documenta le loro critiche e le risposte di Stallman. Ulrich Drepper, che Stallman aveva nominato per lavorare sul GNU libc per il Progetto GNU, pubblicò lamentele contro Stallman nelle note per la release di glibc 2.2.4. Drepper accusa Stallman di aver tentato una "scalata ostile " del progetto, riferendosi a lui come un "maniacco del controllo e matto da legare." Eric S. Raymond, che a volte reclama di parlare per parti del movimento open source, ha scritto molti pezzi spiegando il disaccordo del movimento con Stallman e il movimento per il software libero, spesso in termini aspramente critici verso Stallman.

Attivismo

Stallman ha scritto molti saggi sulla libertà del software e dai primi anni '90 è un attivista politico schietto a favore del movimento del software libero. I discorsi che ha regolarmente tenuto sono intitolati “Il progetto GNU e il movimento del software libero”(*The GNU project and the Free Software movement*), “I pericoli dei brevetti software”(*The Dangers of Software Patents*) e “Copyright e comunità nell'era delle reti di computer”(*Copyright and Community in the age of computer networks*).

Nel 2006 e 2007, durante il diciottesimo mese della consultazione pubblica per la prima stesura della versione 3 della GNU General Public license, aggiunse un quarto tema per spiegare i cambiamenti proposti.

La fedele difesa di Stallman per un software libero ispirò “Virtual Richard M. Stallman” (vrms), software che analizza i pacchetti correntemente installati su un sistema Debian GNU/Linux, e riporta quelli che derivano da un albero non libero. Stallman sarebbe stato in disaccordo con parti della definizione di software libero di Debian.

Nel 1999, Stallman richiese lo sviluppo di un' enciclopedia libera on-line invitando il pubblico a contribuire con articoli.

In Venezuela, Stallman ha pronunciato discorsi pubblici e ha promosso l'adozione di software libero nella compagnia petrolifera di stato (PDVSA), nel governo municipale, e nell'esercito della nazione.

Sebbene di solito sostenitore di Hugo Chávez, Stallman ha criticato alcune politiche inerenti la programmazione televisiva, i diritti per la libertà di parola, e la privacy negli incontri con Chavez e nei discorsi pubblici in Venezuela.

Stallman è nel Consiglio Consultivo di teleSUR, una stazione televisiva Latino-Americana.

Nell' Agosto del 2006 durante gli incontri con il governo dello stato indiano di Kerala, convinse i funzionari ad abbandonare il software proprietario, come quello di Microsoft, nelle scuole statali.

Questo è sfociato in una significativa decisione di portare tutti i computer scolastici di 12.500 scuole superiori da Windows a un sistema operativo libero.

Dopo incontri personali, Stallman ha ottenuto dichiarazioni positive sul movimento del software libero dall'allora presidente dell'India, Dr. A.P.J. Abdul Kalam, dal candidato alla presidenza francese nel 2007 Ségolène Royal, e dal presidente dell'Ecuador Rafael Correa.

Stallman ha partecipato a proteste sui brevetti del software, DMR, e software proprietario.

Protestando contro il software proprietario nell'aprile 2006, Stallman tenne un cartellone con scritto “Non comprate da ATI, nemico della vostra libertà” durante un discorso di un rappresentante dell'ATI nell'edificio dove Stallman lavora, con il risultato che fu chiamata la polizia. ATI da allora si è fusa con AMD e ha fatto piccoli passi per rendere la documentazione per il loro hardware disponibile all'uso della comunità del software libero.

Stallman ha anche aiutato e supportato il tentativo di riportare online il progetto di una biblioteca online di spartiti musicali internazionali (International Music Score Library Project), dopo che era stata tolta il 19 ottobre 2007 in seguito a una lettera (cease and desist letter) della Universal Edition.

Terminologia

Stallman dà molta importanza alle parole e le etichette che la gente usa per parlare del mondo, incluse le relazioni tra il software e la libertà. Instancabilmente chiede alle persone di dire “software libero”, “GNU/Linux”, e di evitare il termine “proprietà intellettuale” o “pirateria” (legato ai computer). Le sue richieste che le persone usino certi termini, e i suoi continui sforzi per convincere la gente dell'importanza della terminologia sono fonte di regolare incomprensione e contrasto con parti della comunità per un software libero e open source.

Uno dei suoi criteri per concedere un'intervista ad un giornalista è che il giornalista accetti di usare la sua terminologia dall'inizio alla fine dell'articolo. Alcune volte ha anche richiesto ai giornalisti di leggere parti della filosofia GNU prima di un' intervista, per “motivi di efficienza”. E' conosciuto per aver rifiutato interventi su alcune questioni di terminologia.

Stallman rifiuta un comune termine alternativo “open-source software” perché non fa venire in mente ciò che Stallman considera come valore del software: la libertà. Di conseguenza non informerà le persone sulle questioni della libertà, e non porterà la gente a dare valore e difendere la propria libertà. Due alternative che Stallman accetta sono “libre software” e “unfettered software” (software senza restrizioni), comunque, “free software” è il termine che chiede alle persone di usare in inglese. Per ragioni simili, sostiene il termine “software proprietario” piuttosto che “closed source software”(sorgente chiuso), quando ci si riferisce ad un software che non è libero.

Stallman chiede ripetutamente che il termine “GNU/Linux”, che pronuncia “GNU Slash Linux”, venga usato per riferirsi al sistema operativo creato dalla combinazione del sistema GNU e il kernel Linux. Stallman si riferisce a questo sistema operativo come “una variante di GNU, ed il progetto GNU è il suo principale sviluppatore”. Reclama che la connessione tra la filosofia del progetto GNU e il suo software viene rotta quando le persone si riferiscono alla combinazione semplicemente come “Linux”. A cominciare circa dal 2003, cominciò ad usare anche il termine “GNU+Linux” che pronuncia “GNU plus Linux”.

Stallman sostiene che il termine “Proprietà Intellettuale” è stato ideato per confondere le persone, e viene usato per evitare una discussione intelligente sulle specifiche di copyright, brevetti, e leggi sul marchio, rispettivamente, trattando senza distinzione aree di leggi che sono più dissimili che simili. Egli sostiene anche che riferendosi a queste leggi come leggi “di proprietà”, il termine influenza la discussione quando si pensa a come trattare queste questioni. “Queste leggi ebbero origine separatamente, si svilupparono differentemente, coprono attività differenti, hanno regole diverse, e sollevano diverse questioni politiche pubbliche. La legge sul Copyright fu creata per promuovere l'attività dello scrittore e l'arte, e tratta esaurientemente i dettagli di un lavoro di scrittore o arte. La legge sul brevetto aveva l'intenzione di incoraggiare la pubblicazione di idee, al prezzo di limitati monopoli su queste idee, un prezzo che potrebbe valer la pena di pagare in alcuni campi e non in altri. La legge sul marchio non si proponeva di promuovere alcuna attività di business, ma semplicemente di permettere agli acquirenti di conoscere cosa stavano comprando.”

Un esempio di avvertimento per evitare altra terminologia offrendo anche suggerimenti per possibili alternative, è questo ciò che emerge da una email di Stallman a una mailing list pubblica:

“Io penso che sia buona cosa per gli autori (per favore non chiamateli “creatori”, non sono dei”) chiedere denaro per le copie dei loro lavori (per favore non svalutate questi lavori chiamandoli “contenuto”) in modo da guadagnare (il termine “compenso” falsamente implica un questione di risarcire alcuni tipi di danni).”

Vita personale

Stallman ha dedicato la maggior parte delle energie della sua vita all'attivismo politico e nel campo del software. Manifestando di curarsi poco dei beni materiali, spiega che “ha sempre vissuto grossolanamente...come uno studente, sostanzialmente. E mi piace questo, perchè significa che il denaro non mi dice cosa fare.”

Per molti anni, Stallman non tenne una residenza permanente al di fuori del suo ufficio al laboratorio CSAIL del MIT, definendosi uno “squatter”(occupante abusivo) nel campus. La sua posizione di “research affiliate” al MIT non è pagata.

In una nota a piè di pagina di un articolo che scrisse nel 1999, dice “Come ateo, io non seguo nessuna religione guida, ma qualche volta mi trovo ad ammirare qualcosa di quello che dicono.” Stallman ha scelto di non festeggiare il Natale, celebra invece il 25 dicembre una festa di sua invenzione, “Grav-mass”. Il nome e la data sono riferiti a Isaac Newton, che nacque in quel giorno. Quando gli vengono fatte domande sulle sue influenze, risponde che ammira Mahatma Gandhi, Martin Luther King, Jr., Nelson Mandela, Aung San Suu Kyi, Ralph Nader, e Dennis Kucinich, e commenta anche:

“Io ammiro Franklin D. Roosevelt e Winston Churchill, anche se critico alcune delle cose che hanno fatto”. Stallman è un sostenitore del Green Party.

Stallman suggerisce di non possedere un cellulare, poiché crede che le tracce dei telefoni cellulari creino situazioni dannose per la riservatezza dei dati personali (privacy issues).

Stallman adora un grande varietà di stili musicali dal Conlon Nancarrow al folk; la “Canzone del Software Libero” prende la forma dalle parole alternative per la danza folk bulgara Sadi Moma. Più recentemente scrisse un’imitazione della canzone folk cubana Guantanamera, riguardante un prigioniero alla base navale di Guantanamo Bay, e la registrò a Cuba con musicisti cubani.

Stallman è un fan della fantascienza, inclusi i lavori dell'autore Greg Egan. Occasionalmente si reca a convegni sulla fantascienza (science fiction conventions) e scrisse la Canzone del Software Libero mentre aspettava il suo turno per cantare ad un convegno. Ha scritto due storie di fantascienza, “Il diritto di leggere” (“The Right to Read”) e “Ingegneria Genetica” (“Jinnetic Engineering”).

Madrelingua Inglese, Stallman è anche sufficientemente fluente in francese e spagnolo per tenere i suoi discorsi di due ore in queste lingue, e rivendica padronanza piuttosto frammentaria di indonesiano.

«Il "controllo sull'uso delle proprie idee" in realtà costituisce un controllo sulle vite degli altri; e di solito viene usato per rendere più difficili le loro vite.»
(Richard Stallman)

«La società ha bisogno di libertà: quando un programma ha un proprietario, l'utilizzatore perde la libertà di controllare parte della sua vita.»
(Richard Stallman)

«Non farò a voi quello che altri hanno fatto a me.»
(Richard Stallman)

Progetto GNU

Manifesto GNU

Cos'è GNU? Gnu Non è Unix!



Traduzioni del Manifesto GNU sono disponibili in Coreano, Francese, Giapponese, Russo, Svedese e Tedesco

di Richard Stallman

GNU, che sta per "Gnu's Not Unix" (Gnu Non è Unix), è il nome del sistema software completo e Unix-compatibile che sto scrivendo per distribuirlo liberamente a chiunque lo possa utilizzare. Molti altri volontari mi stanno aiutando. Abbiamo gran necessità di contributi in tempo, denaro, programmi e macchine.

Fino ad ora abbiamo un editor Emacs fornito di Lisp per espanderne i comandi, un debugger simbolico, un generatore di parser compatibile con yacc, un linker e circa 35 utility. È quasi pronta una shell (interprete di comandi). Un nuovo compilatore C portatile e ottimizzante ha compilato se stesso e potrebbe essere pubblicato quest'anno. Esiste un inizio di kernel, ma mancano molte delle caratteristiche necessarie per emulare Unix. Una volta terminati il kernel e il compilatore sarà possibile distribuire un sistema GNU utilizzabile per lo sviluppo di programmi. Useremo TeX come formattatore di testi, ma lavoriamo anche su un nroff. Useremo inoltre il sistema a finestre portatile libero X. Dopo di che aggiungeremo un Common Lisp portatile, il gioco Empire, un foglio elettronico e centinaia di altre cose, oltre alla documentazione in linea. Speriamo di fornire, col tempo, tutte le cose utili che normalmente si trovano in un sistema Unix, ed anche di più.

GNU sarà in grado di far girare programmi Unix, ma non sarà identico a Unix. Apporteremo tutti i miglioramenti che sarà ragionevole fare basandoci sull'esperienza maturata con altri sistemi operativi. In particolare abbiamo in programma nomi più lunghi per i file, numeri di versione per i file, un filesystem a prova di crash, forse completamento automatico dei nomi dei file, supporto indipendente dal terminale per la visualizzazione e forse col tempo un sistema a finestre basato sul Lisp, attraverso il quale più programmi Lisp e normali programmi Unix siano in grado di condividere lo schermo. Sia C che Lisp saranno linguaggi per la programmazione di sistema. Per le comunicazioni vedremo di supportare UUCP, Chaosnet del MIT ed i protocolli di Internet.

GNU è inizialmente orientato alle macchine della classe 68000/16000 con memoria virtuale, perché sono quelle su cui è più facile farlo girare. Lasciamo agli interessati il lavoro necessario a farlo girare su macchine più piccole.

Vi preghiamo, per evitare confusioni, di pronunciare la 'G' nella parola 'GNU' quando indica il nome di questo progetto. [N.d.T.: questa avvertenza serve ad evitare che in inglese "GNU" sia pronunciato come la parola "new"].

Perché devo scrivere GNU

Io credo che il punto fondamentale sia che, se a me piace un programma, io debba condividerlo con altre persone a cui piace. I venditori di software usano il criterio "divide et impera" con gli utenti, facendo sì che non condividano il software con altri. Io mi rifiuto di spezzare così la solidarietà con gli altri utenti. **La mia coscienza non mi consente di firmare un accordo per non rivelare informazioni o per una licenza d'uso del software.** Ho lavorato per anni presso il laboratorio di intelligenza artificiale per resistere a queste tendenze e ad altri atteggiamenti sgradevoli, ma col tempo queste sono andate troppo oltre: **non potevo rimanere in una istituzione dove ciò viene fatto a mio nome contro la mia volontà.**

Per poter continuare ad usare i computer senza disonore, ho deciso di raccogliere un corpus di software libero in modo da andare avanti senza l'uso di alcun software che non sia libero. Mi sono dimesso dal laboratorio di Intelligenza Artificiale per togliere al MIT ogni scusa legale che mi impedisca di distribuire GNU.

Perché GNU sarà compatibile con Unix

Unix non è il mio sistema ideale, ma non è poi così male. Le caratteristiche essenziali di Unix paiono essere buone e penso di poter colmare le lacune di Unix senza rovinarne le caratteristiche. E adottare un sistema compatibile con Unix può risultare pratico anche per molti altri.

Come sarà reso disponibile GNU

GNU non è di pubblico dominio. A tutti sarà permesso di modificare e ridistribuire GNU, ma a nessun distributore sarà concesso di porre restrizioni sulla sua ridistribuzione. Questo vuol dire che **non saranno permesse modifiche proprietarie.** Voglio essere sicuro che tutte le versioni di GNU rimangano libere.

Perché molti altri programmatori desiderano essere d'aiuto

Ho trovato molti altri programmatori molto interessati a GNU che vogliono dare una mano.

Molti programmatori **sono scontenti della commercializzazione** del software di sistema. Li può aiutare a far soldi, **ma li costringe in generale a sentirsi in conflitto con gli altri programmatori, invece che solidali.** L'atto di amicizia fondamentale tra programmatori è condividere programmi; le politiche di commercializzazione attualmente in uso essenzialmente proibiscono ai programmatori di trattare gli altri come amici. **Gli acquirenti del software devono decidere tra l'amicizia e l'obbedienza alle leggi.** Naturalmente molti decidono che l'amicizia è più importante. Ma quelli che credono nella legge non si sentono a proprio agio con queste scelte. Diventano cinici e pensano che programmare sia solo un modo per fare soldi.

Lavorando e utilizzando GNU invece che programmi proprietari, possiamo comportarci amichevolmente con tutti e insieme rispettare la legge. Inoltre GNU è un esempio che ispira gli altri e una bandiera che li chiama a raccolta perché si uniscano a noi nel condividere il software. Questo ci può dare una sensazione di armonia che sarebbe irraggiungibile se usassimo software che non sia libero.

Per circa la metà dei programmatori che conosco è una soddisfazione importante, che il denaro non può sostituire.

Come si può contribuire

Chiedo ai produttori di computer donazioni in denaro e macchine, ed ai privati donazioni in programmi e lavoro.

Donare delle macchine può far sì che su di esse giri ben presto GNU. Le macchine devono essere sistemi completi e pronti all'uso approvati per l'utilizzo in aree residenziali e non devono richiedere raffreddamento o alimentazione di tipo sofisticato.

Ho conosciuto moltissimi programmatori desiderosi di contribuire a GNU part-time. Per la gran parte dei progetti, un lavoro part-time distribuito risulterebbe troppo difficile da coordinare, perché le varie parti scritte indipendentemente non funzionerebbero insieme. Ma per scrivere un sostituto di Unix questo problema non si pone, perché un sistema Unix completo contiene centinaia di programmi di servizio, ognuno con la propria documentazione separata, e con gran parte delle specifiche di interfaccia date dalla compatibilità con Unix. Se ogni partecipante scrive un solo programma da usare al posto di una utility di Unix, il quale funzioni correttamente al posto dell'originale su un sistema Unix, allora questi programmi funzioneranno bene una volta messi assieme. Anche considerando qualche imprevisto dovuto a Murphy, assemblare tali componenti è un lavoro fattibile. **Il kernel invece richiederà una più stretta cooperazione, e verrà sviluppato da un gruppo piccolo e affiatato.**

Donazioni in denaro possono mettermi in grado di assumere alcune persone a tempo pieno o part-time. Lo stipendio non sarà alto rispetto agli standard dei programmatori, ma io cerco persone per le quali lo spirito della comunità GNU sia importante quanto il denaro. Io lo vedo come un modo di permettere a degli appassionati di dedicare tutte le loro energie al lavoro su GNU senza essere costretti a guadagnarsi da vivere in un altro modo.

Perché tutti gli utenti dei computer ne trarranno beneficio

Una volta scritto GNU, ognuno potrà avere liberamente del buon software di sistema, così come può avere l'aria.

Questo significa molto di più che far risparmiare a ciascuno il costo di una licenza Unix: vuol dire evitare l'inutile spreco di ripetere ogni volta lo sforzo della programmazione di sistema. Queste energie possono essere invece impiegate ad avanzare lo stato dell'arte.

I codici sorgente completi del sistema saranno a disposizione di tutti. Di conseguenza, un utente che abbia necessità di apportare dei cambiamenti al sistema sarà sempre in grado di farlo da solo o di commissionare i cambiamenti ad un programmatore o ad un'impresa. Gli utenti non saranno più in balia di un solo programmatore o di una impresa che, avendo la proprietà esclusiva dei sorgenti, sia la sola a poter fare le modifiche.

Le scuole avranno la possibilità di fornire un ambiente molto più educativo, incoraggiando gli studenti a studiare e migliorare il software di sistema. I laboratori di informatica di Harvard avevano una politica per cui nessun programma poteva essere installato nel sistema senza che i sorgenti fossero pubblicamente consultabili, e la praticarono rifiutandosi effettivamente di installare alcuni programmi. Questo comportamento mi è stato di grande ispirazione.

Infine, scompariranno le necessità burocratiche di tener conto di chi sia il proprietario del software di sistema e di chi abbia il diritto di farci cosa.

Ogni sistema per imporre tariffe d'uso di un programma, comprese le licenze d'uso per le copie, è sempre estremamente costoso in termini sociali a causa del complesso meccanismo necessario per decidere quanto (cioè per quali programmi) ognuno debba pagare, e solo uno stato di polizia può costringere tutti all'obbedienza. Immaginate una stazione spaziale dove l'aria deve essere prodotta artificialmente ad un costo elevato: far pagare ogni litro d'aria consumato può essere giusto, ma indossare la maschera col contatore tutto il giorno e tutta la notte è intollerabile, anche se tutti possono permettersi di pagare la bolletta. E le videocamere poste in ogni dove per controllare che nessuno si tolga mai la maschera sono offensive. Meglio finanziare l'impianto di ossigenazione con una tassa pro capite e buttar via le maschere.

Copiare un programma in tutto o in parte è tanto naturale per un programmatore quanto respirare ed è altrettanto produttivo. Dovrebbe essere altrettanto libero.

Kernel Linux

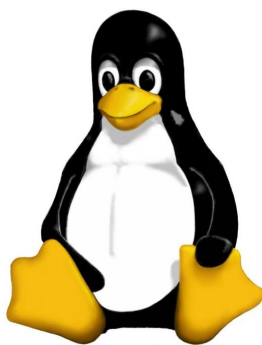
Nel 1990 il sistema GNU era quasi completo ma il kernel HURD non era ancora del tutto pronto per la diffusione, ma nel 1991 uno studente finlandese, Linus Torvalds, creò un nuovo kernel Unix compatibile che chiamò Linux.

Nel 1992 questo kernel fu integrato assieme al software GNU rendendolo un sistema completo, per questo tuttora si utilizza un sistema operativo GNU/Linux. Il kernel Linux è distribuito sotto licenza GNU GPL versione 2.

Come ogni progetto di software libero, anche il kernel Linux è in continua evoluzione e le nuove release sono reperibili tramite internet. La dimensione del kernel Linux cresce in maniera esponenziale, aggiungendo nuovi moduli, nuovo hardware supportato e così via.

Linux è un kernel che supporta il multitasking ed è multi utente. Ciò permette che diversi utenti (con privilegi differenziati) possano eseguire sullo stesso sistema diversi processi software in simultanea. Attualmente linux supporta gran parte dell'hardware disponibile per PC e supporta un numero enorme di architetture (tra cui SPARC, PowerPC e le più moderne CPU a 64 bit).

Il codice sorgente di Linux è disponibile a tutti, è ampiamente personalizzabile anche da utenti meno esperti, al punto da rendere possibile, in fase di compilazione, l'esclusione di codice non strettamente indispensabile. La flessibilità di questo kernel lo rende adatto a tutte quelle tecnologie embedded emergenti fino ad essere incorporato in alcuni videoregistratori digitali e nei telefoni cellulari.



Licenze

Introduzione

La licenza che accompagna un software quando viene distribuito in pubblico è un contratto che l'utente accetta prima di poter utilizzare il prodotto. Questo contratto stabilisce le condizioni d'utilizzo del software, imponendo obblighi e proteggendo diritti.

Il software proprietario è protetto da licenze che ne limitano l'uso, la condivisione e la modifica, al contrario le licenze per il software libero ne salvaguardano le libertà che ogni utente deve avere.

La licenza a cui un software libero si deve riferire è la Licenza Pubblica Generica GNU (GNU GPL) che ne difende tutte le libertà e principi ma anche altre licenze possono essere usate se sono compatibili con la GPL.

Nel caso del software proprietario le licenze si basano sul copyright, al contrario la GNU GPL si basa sul copyleft o permesso d'autore.



Copyleft

Il copyleft o permesso d'autore è un termine che indica il carattere opposto rispetto al copyright, in quanto pone l'attenzione sulla libertà degli utilizzatori del software e non sui produttori.

Il metodo più semplice per rendere libero il software è quello di renderlo di pubblico dominio, ma alcune persone poco rispettose potrebbero fare delle piccole modifiche e rendere il software proprietario.

Il copyleft invece, ribalta il metodo utilizzato dal copyright per difendere il software libero, infatti pur rimanendo modificabile e ridistribuibile, lo protegge da una possibile conversione a software proprietario, garantendo che ogni utente conservi le libertà.

Il principio del copyright viene quindi usato per proteggere dalla “privatizzazione” del software libero.

Questo principio è anche usato per incentivare i programmatori a produrre software libero anche in campo lavorativo.

Infatti un programmatore professionista che produce dei miglioramenti a del software libero, può solo redistribuirli mantenendoli liberi, il datore di lavoro preferisce distribuire questo lavoro piuttosto che non utilizzarlo.

Un esempio di software prodotto grazie a questi incentivi è il compilatore GNU C++.

Con il copyleft i programmi e la loro libertà diventano inseparabili anche a livello legale.

Per definire nei dettagli il copyleft sono nate diverse licenze che difendono in diversi modi la libertà.

GNU General Public License

La GNU GPL è la più importante licenza per il software libero, fu scritta inizialmente da Richard Stallman e Eben Moglen nel 1989, ed è la prima a sfruttare il principio del copyleft. Con questa licenza l'utente è tenuto a rendere disponibile il codice sorgente solo alle persone che hanno ricevuto da lui la copia del programma o, in alternativa, accompagnare il software con una offerta scritta di rendere disponibile il sorgente su richiesta e per il solo costo della copia. Questo significa, ad esempio, che è possibile creare versioni private di un software sotto GNU GPL, a patto che tale versione non venga distribuita a qualcun altro.

Essa è progettata per e poter essere applicata facilmente ai programmi. Per far questo non si deve modificare la GNU GPL ma basta aggiungere nel programma una nota che vi faccia riferimento. Chi vuole applicare la GNU GPL ai programmi deve usarne il testo nella sua completezza, non è ammesso l'uso versioni parziali della licenza (lo stesso vale per la LGPL e la FDL).

Se qualcuno distribuisce un software (in particolare, versioni modificate) senza rendere disponibile il codice sorgente o violando in altro modo la licenza, può essere denunciato dall'autore originale secondo le stesse leggi sul copyright. È un intelligente cavillo legale, ed è per questo che la GNU GPL è stata descritta come un "*copyright hack*".

Free Software Foundation (FSF) detiene i diritti di copyright per il testo della GNU GPL, ma non detiene i diritti del software rilasciato con questa licenza.

GNU Lesser General Public License

La GNU LGPL si basa sulla GNU GPL ma è meno restrittiva poiché fu studiata come compromesso tra la GNU General Public License e altre licenze non-copyleft come ad esempio la Licenza BSD, la Licenza X11 e la Licenza Apache. Nacque principalmente per essere associata alle librerie ma ben presto fu usata anche per molti altri programmi, il primo nome era infatti "GNU Library General Public License". Questa licenza essendo meno restrittiva si è diffusa molto e le aziende commerciali la usano spesso per alcuni progetti commerciali, in ogni caso essa difende la comunità del Software Libero da possibili abusi delle stesse.

La licenza stabilisce il copyleft sul singolo file di codice sorgente, ma non sull'intero software. Per questo un software o libreria rilasciata sotto licenza LGPL può essere incluso liberamente in un'applicazione commerciale closed-source, a patto che le modifiche apportate al codice sorgente del software o alla libreria stessa vengano rese pubbliche; tutti gli altri file dell'applicazione possono essere rilasciati con licenza proprietaria e senza codice sorgente.

GNU Free Documentation License

La GNU FDL è una licenza che sfrutta i principi del copyleft per distribuire i contenuti liberi della documentazione di software e materiale didattico. Questa licenza garantisce a chiunque di copiare e redistribuire questo materiale con o senza modifiche e che ogni copia deve essere distribuita con la stessa licenza. Tali copie possono essere vendute e, se riprodotte in gran quantità, devono essere rese disponibili anche in un formato che faciliti successive modifiche. Wikipedia è la più grande raccolta di documentazione che utilizza questo tipo di licenza.

Pericolo dei brevetti sul software

Come già scritto, il software è difficile da trattare come un oggetto materiale, e tutto ciò che viene fatto rientrare nell'espressione "proprietà intellettuale" come il copyright, i marchi registrati e i brevetti sono difficilmente applicabili senza comportare danni.

Il sistema dei brevetti ha lo scopo di favorire lo sviluppo di nuove idee per incrementare il progresso, ma applicato al mondo del software esso ne limita lo sviluppo.

Infatti i brevetti coprono delle idee solitamente semplici che devono essere usate per la creazioni di altro software.

L'andamento che si sta riscontrando è che vengono rilasciati troppi brevetti su idee troppo semplici e non su programmi più complessi.

Col passare del tempo i brevetti coprono aree sempre maggiori di "spazio mentale", lo "spazio libero" diventa sempre più piccolo e inutile ai programmatori che vorrebbero creare nuovi programmi sempre migliori che probabilmente necessitano di idee brevettate. Ecco quindi che diventa impossibile per un programmatore qualsiasi, anche a uno studente, poter pubblicare un programma, anche semplice, che però ingloba una o più idee già brevettate magari da grandi industrie monopoliste.

I brevetti creano dei monopoli sulle idee nel campo della programmazione poiché essendo astratta è difficile da gestire come l'infinito spazio mentale.

Quando si progetta una casa gli elementi da gestire per la progettazione sono materiali, condizionati quindi dalla fisica e devono rispettare molti parametri, quelli che si riscontrano programmando un software complesso, sono molti, astratti e semplici.

Quando si programma si ha a disposizione dei comandi base come if, else, while o and, or ecc. che permettono di creare qualsiasi cosa, come disse Torvalds "Dentro i confini del computer, sei tu il creatore. Controlli – almeno potenzialmente – tutto ciò che vi succede. Se sei abbastanza bravo, puoi essere un dio. Su piccola scala."

Se il sistema dei brevetti continuasse sulla strada intrapresa, moltissimi programmatori lavorerebbero per grandi aziende monopolistiche, che possiedono già moltissimi brevetti, mentre altri programmatori sarebbero costretti a non pubblicare i loro programmi.

Per rendere chiara l'idea del problema dei brevetti applicati al software ci si può immaginare se i brevetti fossero stati applicati alla musica. Se Verdi avesse brevettato la scala enigmatica presente in "Ave Maria" chiunque avesse ripreso quel *modo* avrebbe dovuto pagare qualcosa o ancora peggio se si brevettassero alcuni intervalli fra note sarebbe praticamente impossibile per i compositori scrivere musica simile a quella conosciuta fino ad ora.

La situazione in America peggiora sempre più, mentre per la maggior parte degli stati europei i brevetti non sono ancora applicabili al software grazie all'importante rifiuto (648 voti contrari su 680) del parlamento europeo alla direttiva sui brevetti software.

E' importante continuare a sostenere questa battaglia che richiede la consapevolezza e l'informazione dei cittadini, troppo spesso tenuti all'oscuro di queste questioni.

Open Source



Introduzione

E' evidente che il software ha una grande potenzialità e le industrie informatiche hanno compreso ormai da tempo che l'accesso al codice sorgente può diventare un'ottima risorsa per produrre buon software. Per questo il termine "Open Source" fu coniato nel 1997 da Eric S. Raymond per avvicinare il software libero al mondo degli affari e aumentarne la diffusione. Pur partendo da uno scopo valido, i sostenitori dell'open source mettono in secondo piano i veri principi che hanno dato vita al movimento del software libero. Come ricorda Stallman la terminologia è importante e usare il termine "Open source" per definire il software libero è sbagliato e controproducente poiché oscura le reali ragioni per le quali è nato il software libero: la condivisione e la libertà.

Tuttavia la commercializzazione del software libero è ben accetta ma nelle forme adeguate e che rispettano i principi del movimento e di chi lo sostiene.

Perché L'“Open Source” Non Ha Centrato L'Obiettivo Del Software Libero

di Richard Stallman

Quando definiamo “libero” un software, intendiamo che rispetta le libertà essenziali degli utenti: la libertà di eseguire il programma, di studiare il suo codice e modificarlo, e di ridistribuirne copie con o senza modifiche. E' una questione di libertà, non di prezzo, quindi pensate alla “libertà di parola”, non ad una “birra gratis” (*ndt*: free in inglese significa sia gratuito che libero, sfumatura che in italiano si perde)

L'importanza di queste libertà è vitale. Si tratta di libertà essenziali, non soltanto per l'utente in sé, ma perché promuovono la solidarietà sociale, ossia la condivisione e la cooperazione. La loro importanza aumenta al crescere della digitalizzazione delle nostre culture ed attività quotidiane. In un mondo di suoni, immagini e parole digitali, ***il software libero è un concetto che inizia a sovrapporsi con un concetto più ampio, quale quello della libertà in generale.***

Oggi esistono decine di milioni di persone in tutto il mondo che usano software libero; abbiamo scuole, in alcune regioni dell'India e della Spagna che insegnano ai loro studenti come utilizzare il sistema operativo GNU/Linux, un sistema libero. Tuttavia la maggior parte di questi utenti non sanno nulla delle ragioni etiche per cui abbiamo sviluppato questo sistema e abbiamo creato la comunità del software libero, in quanto spesso, oggi, ci si riferisce a questo sistema e a questa comunità con il termine “open source”; nulla di più sbagliato, considerato come queste cose vengono riferite a una diversa filosofia nella quale raramente si fa menzione di tali libertà.

Il movimento per il software libero iniziò una campagna per le libertà degli utenti dei computer nel 1983. Nel 1984 cominciammo a sviluppare il sistema operativo libero GNU, in modo da non essere costretti ad utilizzare i sistemi operativi non liberi che negano la libertà ai propri utenti.

Nel corso degli anni Ottanta, oltre a sviluppare la maggior parte dei componenti essenziali di questo sistema, abbiamo redatto la GNU General Public License, una licenza studiata appositamente per proteggere la libertà di tutti gli utenti di un programma.

Tuttavia non tutti gli utenti e sviluppatori di software libero furono d'accordo sugli obiettivi del movimento del software libero.

Nel 1998, una parte della comunità del software libero si staccò, dando origine ad una campagna nel nome dell'“**open source**”. Questo era stato originariamente proposto per evitare la possibile confusione derivante dal termine inglese “free software”, ma ben presto iniziò anche ad essere associato a **visioni filosofiche tendenzialmente dissimili rispetto a quelle del movimento del software libero.**

Alcune delle persone che promuovono l'“open source”, **lo considerano come una “campagna di marketing per il software libero”, che vorrebbe attrarre i dirigenti delle aziende mostrando i benefici pratici ed evitando di citare gli ideali** che tentano di distinguere tra ciò che è giusto e sbagliato, argomenti che queste persone sicuramente non gradirebbero. Altri promotori dell'“open source” rigettano in toto i valori etici e sociali del movimento del software libero. Qualunque sia la loro visione, quando fanno una campagna per l'“open source” non fanno riferimento né sostengono questi valori.

Il termine “open source” iniziò ben presto ad essere associato alla deprecabile pratica di citarne soltanto valori pratici, come fornire software più potente e affidabile. Molti dei sostenitori dell'“open source” hanno adottato questa prassi, preferendo un significato per così dire “pratico”.

Quasi tutto il software open source è software libero; i due termini descrivono all'incirca la stessa categoria di software. **Ma si basano su valori fondamentalmente diversi.**

L'open source è una metodologia di sviluppo; il software libero è un movimento sociale.

Per il movimento per il software libero, il software libero è un imperativo etico, perché soltanto il software libero rispetta la libertà degli utenti.

Di contro, la filosofia dell'open source pensa a come “migliorare” il software soltanto da un punto di vista pratico. Considera il software non libero come una soluzione subottimale. Per il movimento per il software libero, comunque, il software non libero è un problema sociale e la soluzione è migrare verso software libero.

Software Libero. Open source. Se si tratta dello stesso software, ha importanza quale nome utilizziamo?

Certo, perché parole differenti sottendono idee diverse.

Benché un programma libero, in qualunque modo venga chiamato, vi dia oggi la stessa libertà, **stabilire la libertà in modo che perduri nel tempo dipende soprattutto dall'insegnare alla gente il valore della libertà.** Se volete aiutarci in questo è essenziale che parliate di “software libero”.

Noi del movimento del software libero non pensiamo che il campo dell'open source sia il nemico; il nemico è il software proprietario. Vogliamo però che la gente sappia che noi ci battiamo per la libertà e che pertanto non vogliamo essere scorrettamente identificati come sostenitori dell'open source.

Progetto nupedia e wikimedia



Nel marzo del 2000 da Jumbo Wales e Larry Sanger fu fondato il progetto **nupedia**, una enciclopedia in inglese accessibile via web e scritta da esperti.

Sebbene le voci di Nupedia fossero pubblicate sotto GNU Free Documentation License il contenuto era libero ma non wiki, non poteva essere modificato da chiunque, in questo modo veniva mantenuta l'alta qualità. Esisteva però una revisione paritaria attraverso la quale accademici lavoravano, come volontari, per migliorare i contenuti.

Nel 2003 questo progetto fu terminato e le voci furono aggiunte all'interno dell'edizione inglese di Wikipedia.

Wikipedia nacque il 15 gennaio 2001, è un'enciclopedia disponibile in numerose lingue e può essere modificata da chiunque, ed è rilasciata sotto GNU Free Documentation License.

Wikipedia è scritta in 250 lingue di cui 180 attive.

Wikimedia Foundation, Inc. è una organizzazione senza scopi di lucro fondata il 20 giugno 2003 da Jimbo Wales e dall'amministratore della Bomis. Questa fondazione promuove la condivisione di contenuti editoriali come testi, immagini, audio e video sul modello open content e wiki. Il concetto di open content si basa sulla filosofia del software libero ovvero la condivisione del sapere che in questo modo è disponibile in diverse forme a chiunque.

Wikimedia raggruppa numerosi progetti:

- Wiktionary**: un dizionario multilingue e un thesaurus.
- Wikiquote**: un'enciclopedia di citazioni.
- Wikibooks**: una collezione di e-book dedicati specificatamente agli studenti.
- Wikinews**: uno spazio dedicato esclusivamente all'attualità, ovvero l'agenzia di stampa.
- Wikisource**: una raccolta di opere di autori celebri.
- Wikispecies**: un archivio tassonomico di tutte le forme di vita conosciute.
- Wikiversity**: lo strumento del wiki diventa uno spazio per costruire collettivamente corsi e lezioni.
- Commons**: l'archivio di immagini e files multimediali rilasciati con licenza libera comune a tutti i progetti della Wikimedia Foundation.

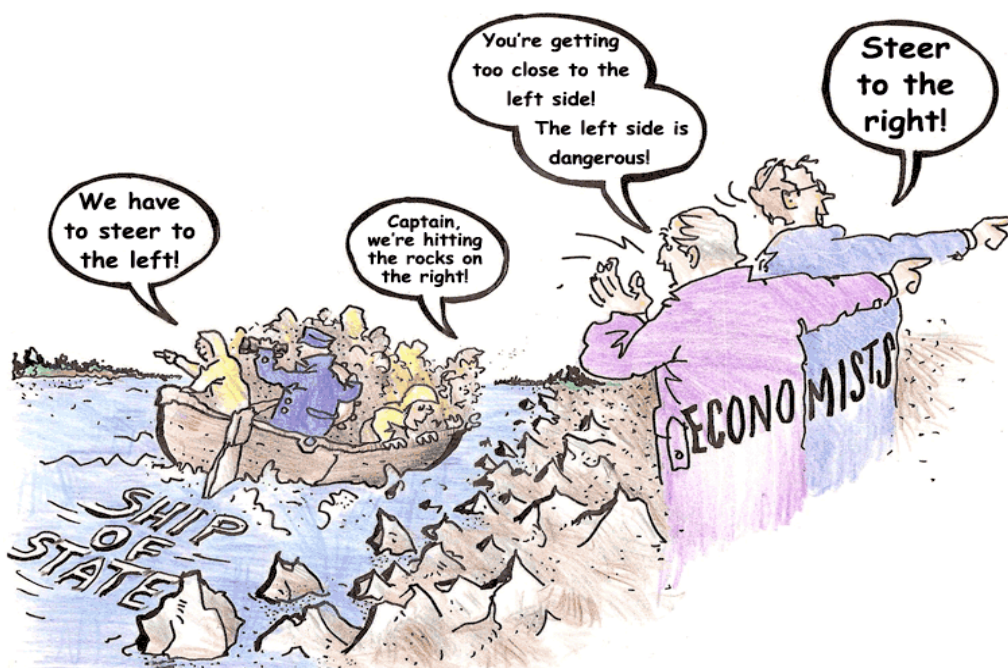
Questi progetti vengono tutti sostenuti da volontari, che mettono a disposizione il proprio tempo e lavoro; il livello non è come quello di una enciclopedia professionale ma il materiale a disposizione è molto e valido.

Si possono però trovare degli errori all'interno delle voci che possono far innescare delle discussioni fra i collaboratori, ma credo che fondamentalmente sia una cosa positiva, poiché emergono tutte le diverse opinioni che si possono avere su un argomento, accordandosi poi per una visione il più oggettiva possibile.

Credo che per poter migliorare sempre di più questi progetti, sia necessario creare ad esempio un marchio di qualità che garantisca alcune voci più meritevoli.

Conclusioni

Software libero, economia e società



Copyright 2007 Richard Stallman and Inne ten Have. Verbatim copying, distribution, and public display of this cartoon are permitted worldwide without royalty in any medium if this notice and the copyright notice are preserved.

Questa vignetta di Richard Stallman riassume bene il ruolo che ha l'economia nella società moderna.

Sulla barca dello stato c'è una massa di persone che naviga lungo il fiume della vita che può presentare delle insidie, le rocce. Al timone della barca c'è il governo che scruta con il binocolo le possibili direzioni da prendere, l'economia che non si trova sulla barca dà consigli secondo un suo punto di vista. Come si è visto, durante la storia gli stati possono prendere direzioni sbagliate che possono portare alla distruzione dello stato stesso; il governo ha il difficile compito di prendere delle decisioni giuste che consentano alla barca di continuare per la corretta direzione.

Ma da quando l'economia ha avuto sempre più influenza come ad esempio durante l'età del consumismo, la barca ha cominciato a sbandare sempre di più seguendo le sue indicazioni che spesso non coincidono con le reali esigenze del popolo; gli ideali perseguiti durante periodi più floridi per la cultura vengono man mano lasciati per far posto agli interessi delle persone che governano l'economia. Alcune persone che si trovano sulla barca si accorgono di questo, pensano, hanno idee e danno consigli sulla direzione da prendere, sono coinvolte direttamente dalle decisioni che prenderà il governo. Gli economisti si trovano invece fermi sulla riva e pur avendo una visione più ampia della situazione danno consigli sbagliati.

Essi infatti consigliano di andare a destra verso le rocce, riferendosi in questo caso, al copyright applicato al software al contrario sarebbe opportuno andare a sinistra verso il copyleft. Spetta al governo prendere la decisione, ma intanto la barca continua a spostarsi sempre più a destra verso le rocce. Il governo ascolterà il popolo che si trova sulla barca o gli economisti?

Fintanto che l'economia avrà sempre più influenza lo stato non potrà prendere delle decisioni corrette, la vita di ogni singola persona sarà condizionata, specialmente quelli che si trovano all'interno della barca che non possono avere una migliore visione di quello che sta accadendo.

Se la barca fosse a più piani le persone potrebbero vedere meglio ciò che accade ma troppo spesso l'economia, si oppone in tutti i modi. Un cittadino poco o mal informato è un cittadino che può essere convinto facilmente da chi vuole fare ricchezza sfruttandolo.

Per costruire questo tipo di barca è necessaria la coscienza di tutti i cittadini a migliorare attraverso il progresso sostenibile e non quello proposto dalle grandi aziende.

L'istruzione è il primo passo per poter informare le persone, specialmente i giovani dovrebbero avere dei principi sani da seguire, come quelli su cui si basa il software libero.

Bibliografia:

-Richard M. Stallman, *Free Software, Free Society: The Selected Essays of Richard M. Stallman*, Boston 2002

-*Software Libero Pensiero Libero*. Saggi scelti di Richard Stallman vol. 1,

-*Software Libero Pensiero Libero*. Saggi scelti di Richard Stallman vol. 2, tr. it. Bernardo Parrella e Gruppo traduttori italiani del progetto GNU, Viterbo 2003.

Sitografia:

-GNU Operating System www.gnu.org

-Free Software Foundation www.fsf.org/

-Free Software Foundation Europe www.france.fsf.eu.org/

-Free Software Directory <http://directory.fsf.org/>

-GNUvox www.gnuvox.info/

-Wikimedia www.wikimedia.org

-Robin Goods MasterNewMedia www.masternewmedia.org/it/

-Attivazione www.attivazione.org

Documento prodotto con:

OpenOffice.org

openSUSE

Ringrazio tutti coloro che mi hanno introdotto all'argomento e chi mi ha aiutato nella stesura del documento.