

Software libero e ricerca

Francesco Potortì

Sommario– “Le licenze di software libero sono una scelta naturale negli ambienti di ricerca”. Scopo di questo articolo è sostenere quest’affermazione con ragionamenti ed esempi, cercando nel contempo di analizzare le interazioni più significative fra software libero e ricerca. In appendice un elenco di termini comuni utilizzati quando si parla di software libero, corredati di spiegazioni, e suggerimenti sulla scelta di una licenza libera per il software.

Termini di ricerca: software libero; ricerca; accademia; licenze software.

Cos’è il software libero

Le radici del software libero affondano nelle idee di libertà di parola e di libero scambio di informazioni, un concetto quest’ultimo tenuto in alta considerazione negli ambienti scientifici. È diffusa l’opinione che la facilità e la libertà di diffusione della conoscenza siano state una delle spinte principali dietro l’eccezionale crescita del sapere scientifico negli ultimi tre secoli.

Tuttavia, la libertà di scambio delle idee non è solo una questione pratica: essa è alla base dei concetti di libertà di pensiero e libertà di espressione. Al pari delle idee, il software è immateriale, e può essere facilmente riprodotto e trasmesso. Al pari delle idee, la libera diffusione ne stimola la crescita e l’evoluzione. E proprio come le idee, ogni giorno di più il software interagisce con la società, producendo effetti etici, economici, politici e in senso generale culturali.

Durante gli anni Ottanta, Richard Stallman formalizzò per la prima volta il concetto di software libero. La definizione di Stallman è generalmente assunta come definizione canonica di software libero. Essa è composta di quattro regole, le *quattro libertà*:

Libertà 0, o Libertà fondamentale:

La libertà di eseguire il programma, per qualunque scopo.

Libertà 1:

La libertà di studiare il programma, e di adattarlo alle proprie necessità.

Libertà 2:

La libertà di redistribuirne copie.

Libertà 3:

La libertà di migliorare il programma, e ridistribuire pubblicamente i miglioramenti.

Le libertà 1 e 3 richiedono accesso al codice sorgente. *Un programma che sia distribuito con una licenza software che conceda le quattro libertà è detto software libero*[1].

Licenze per il software

Una licenza software è un documento legale a corredo di un programma. Senza una licenza di copyright, così come stabilito dalla convenzione di Berna sul diritto d’autore, *un programma non può essere copiato o modificato senza l’esplicito consenso degli autori*. Al contrario, un licenza software libera permette lo studio, la copia, la modifica e la redistribuzione del programma modificato.

La gran parte delle licenze software più diffuse sono *proprietarie*, cioè non garantiscono le quattro libertà. Di solito non permettono la libera copia e modifica, spesso non permettono neanche di mantenere una copia sul proprio portatile e una su un computer fisso, né di fare una copia di prova sulla macchina di un collega o di tenere un'installazione di riserva pronta ad essere usata in caso di guasti. I programmi proprietari, essendo solitamente distribuiti senza codice sorgente, non permettono di studiare il modo in cui funzionano, né di modificarli o adattarli alle proprie esigenze.

Le licenze libere, invece, garantiscono le quattro libertà. Si distinguono licenze libere con e senza permesso d'autore.

Una *licenza con permesso d'autore (copyleft)* è una licenza libera che usa le norme sul diritto d'autore per uno scopo opposto a quello tipico delle licenze proprietarie. Mentre queste impediscono la copia o la modifica di un programma, una *licenza libera con permesso d'autore utilizza le leggi sul diritto d'autore per mantenere il programma libero*. Si può dire che la licenza è *persistente*: se ottengo un programma con permesso d'autore e lo redistribuisco, tale e quale o modificato, devo concedere a chi lo riceve gli stessi diritti che ho acquisito quando ho ottenuto il programma; a tal fine devo distribuire il programma completo di codice sorgente. La licenza libera con permesso d'autore più diffusa è la GNU GPL.

Una *licenza senza permesso d'autore (non-copyleft)* è una licenza libera non persistente. Se ottengo un programma senza permesso d'autore e lo voglio redistribuire, tale e quale o modificato, non sono tenuto a concedere a chi lo riceve gli stessi diritti che ho acquisito quando ho ricevuto il programma. Le comuni licenze libere senza permesso d'autore permettono di creare versioni proprietarie del programma, o anche di redistribuire il programma immutato corredato di una diversa licenza, anche proprietaria. La licenza libera senza permesso d'autore più diffusa è la licenza BSD.

Equivoci ricorrenti

Nonostante la loro ampia diffusione, i termini *software libero* e *open source* sono frequentemente usati in modo scorretto, causando così numerose false convinzioni, e questo anche sulla stampa specializzata e presso tecnici ed esperti. In questo paragrafo cercheremo di evidenziarne e sfatarne alcune.

Open source e software libero sono due cose diverse

Dal punto di vista pratico, "open source" e "software libero" sono esattamente la stessa cosa. In effetti, a parte alcuni casi di secondaria importanza, le *licenze open source* identificate dalla "definizione di Open Source" sono le stesse *licenze libere* identificate dalla definizione di Software Libero.

La differenza sta tutta nei principi che hanno dato origine alle due definizioni; mentre il software libero è definito a partire da ragioni etiche e politiche, l'open source è definito a partire da ragioni pratiche e di convenienza. In questo documento, "open source" e "software libero" sono usati indifferentemente, ma preferiamo il secondo perché la sua definizione è più chiara e concreta.

L'open source è un modello di sviluppo

A rigor di termini, l'open source è un modello di licenza, non un modello di sviluppo. Molti sostenitori dell'open source sostengono che le due cose sono connesse, tuttavia questa argomentazione è discutibile [Bez99]. Il famoso saggio di Eric Raymond "The cathedral and the bazaar" ("La cattedrale e il bazar") traccia un discrimine netto fra metodo di sviluppo centralizzato e distribuito. Il primo è quello usato per lo sviluppo di Emacs, l'editor del progetto GNU, mentre il secondo è usato per lo

sviluppo del kernel Linux. Raymond afferma che il modello di sviluppo distribuito è intrinsecamente superiore per quanto riguarda grossi progetti, e che lo sviluppo distribuito di un progetto di grandi dimensioni richiede una licenza open source.

A fronte di queste affermazioni, va messo in rilievo che la licenza e il modello di sviluppo adottato per un progetto sono questioni in massima parte indipendenti e ben distinte. Benché sia vero che metodi di sviluppo distribuiti possano essere applicati in maniera naturale a progetti open source, essi sono anche applicati da grosse aziende software per lo sviluppo di programmi proprietari. D'altra parte lo sviluppo centralizzato, essendo generalmente più semplice da gestire, è ampiamente usato per progetti open source, ed è la norma per i programmi proprietari.

Il software libero è tecnicamente superiore (o inferiore)

Entrambe le affermazioni sono false, in generale; non c'è alcuna relazione a priori fra la licenza utilizzata e la qualità del software. Il software libero è potenzialmente più affidabile perché chiunque ne può studiare il codice sorgente, e "tutti gli errori sono evidenti davanti ad un milione di occhi", ma queste potenzialità non necessariamente si realizzano. L'unico caso di superiorità tecnica accertata del software libero è quello relativo alla sicurezza, il che è significativo per i programmi usati per la crittografia o per la firma elettronica.

Il software libero costa di più (o di meno)

In generale nessuna di queste affermazioni è vera, perché nella maggior parte dei casi la licenza è solo uno degli elementi che contribuiscono al costo del software. L'installazione, l'istruzione del personale, la manutenzione, l'aggiornamento e la personalizzazione costituiscono tipicamente la fetta più grossa dei costi del software, e non c'è relazione diretta fra la licenza utilizzata ed i costi. Sarebbe facile fare esempi in cui il costo del software libero è molto superiore o molto inferiore a quello di una analoga soluzione proprietaria.

Il software libero non è coperto da diritto d'autore (copyleft significa no copyright)

Queste affermazioni sono entrambe false. Il software libero deve essere distribuito con una licenza, il cui scopo è rimuovere alcune delle restrizioni standard previste dalle leggi sul diritto d'autore, cioè quelle stabilite dalla convenzione internazionale di Berna sul diritto d'autore. Una licenza con permesso d'autore è un tipo di licenza libera, e si fonda sul diritto d'autore.

Chi scrive o modifica software libero lo deve pubblicare

Nessuna licenza libera pone quest'obbligo. Chi scrive software libero per un numero ristretto di clienti non è obbligato a cederlo ad altri, né i clienti sono obbligati a farlo, né tale e quale né modificato. Tuttavia essi hanno il diritto di redistribuirlo, se desiderano.

Pubblicare un programma crea una comunità attorno ad esso

Creare una comunità attorno ad un programma richiede ben più che pubblicarlo come software libero. Da una parte chi pubblica un programma non ha alcun obbligo di gestire una comunità; dall'altra, se è una comunità che si vuol creare, bisogna costruirla e curarla: un compito difficile e impegnativo, ma che può valere lo sforzo. È anche possibile costruire una comunità attorno ad un programma proprietario, ma è generalmente molto più difficile.

Free software and research

La scienza moderna è connaturata al libero scambio di informazioni. Tutta la ricerca scientifica attuale conta su un libero ed efficiente flusso di informazioni, in tutte le forme possibili: congressi, conferenze, riviste, siti web, inviti di studiosi, seminari, cooperazioni a distanza o faccia a faccia, progetti comuni di ricerca, tutte queste sono considerate attività essenziali per un moderno ambiente di ricerca. Esse rendono possibili scambi fruttuosi di idee, favoriscono l'apertura mentale dei ricercatori, contribuiscono in maniera significativa alla nascita di nuovi concetti e costituiscono le fondamenta del processo di crescita incrementale dei risultati. Queste attività creano inoltre una rete di controllo reciproco che rende l'avanzamento della conoscenza eccezionalmente affidabile e tuttavia efficiente, quando confrontato con altri processi di grande complessità creati dalla civilizzazione umana.

Il software libero è un prodotto naturale in un ambiente di ricerca [Kel01]. La nascita e il processo di sviluppo del software ha molto in comune con quello delle idee scientifiche. Come queste, il software migliora grazie a ciò che si apprende dai risultati degli altri, un processo che è molto più efficiente se si può accedere al codice sorgente, analogamente alla possibilità di accedere ai trovati scientifici.

La crescita per accumulazione di risultati è comune alla ricerca scientifica e al software. Isaac Newton disse che se aveva visto più lontano, era stando sulle spalle di giganti. Lo sviluppo del software funziona in maniera simile: la gran parte dei programmi di successo crescono nel tempo, evolvono e migliorano in modo incrementale. Sia nel campo della ricerca che in quello del software, l'apertura delle conoscenze favorisce grandemente questo processo, che è possibile solo se è permessa la modifica.

La scienza è credibile perché in linea di principio chiunque ne può verificare i risultati. Per rendere questo principio applicabile, i ricercatori sono incoraggiati a pubblicare i propri risultati in una forma che consenta un esame indipendente accurato e completo, solitamente quella di un articolo scientifico che esponga in dettaglio tutti i punti rilevanti. Questo processo è analogo a quello con cui il software libero acquista credibilità: rendendo cioè il codice sorgente disponibile, e aprendosi così all'esame da parte di chiunque.

Credibilità ed affidabilità sono sorelle. I risultati scientifici sono affidabili perché sono indipendentemente riproducibili. Un buon articolo scientifico rende possibile riprodurre i risultati di un esperimento, che sia fisico o concettuale, fornendo dettagli sufficienti a che altri ricercatori possano indipendentemente riprodurlo e verificarne i risultati. Qualcosa di simile avviene per il software quando l'affidabilità di un programma può essere verificata rendendo il codice sorgente disponibile per l'ispezione e la ricompilazione su macchine e architetture diverse.

Nella ricerca scientifica, la cooperazione è essenziale. La ricerca langue senza cooperazione: è un abito mentale per i ricercatori, che dovrebbero considerare, e generalmente considerano, una cosa naturale scambiarsi idee e risultati. Un ambiente cooperativo è fertile per il software così come lo è per la ricerca: gli sviluppatori trovano facile e naturale scambiarsi parti di codice e idee, e possono avvantaggiarsi del lavoro altrui.

Vantaggi per gli ambienti accademici

La produzione di software libero può risultare vantaggiosa per gli ambienti accademici o in generale di ricerca, sia dal punto di vista della pura produzione scientifica in termini di articoli pubblicati, sia da un punto di vista di immagine, sia per quanto riguarda lo sviluppo di applicazioni commerciali.

I lavori scientifici che riguardano o includono implementazioni di software libero sono intrinsecamente adatte alla pubblicazione, perché:

- I metodi utilizzati sono divulgati nella loro interezza, essendo il codice sorgente disponibile, e ciò non potrà che essere valutato positivamente dai revisori.
- È incoraggiata la diffusione dei risultati, perché il codice sorgente può essere liberamente copiato e ripubblicato, e ciò può incoraggiare le citazioni, che aumentano il valore dell'articolo.
- I risultati possono essere facilmente riutilizzati e migliorati in maniera incrementale, grazie alla licenza di software libero, aumentando così le possibilità che l'articolo possa assumere un ruolo trainante. Se la licenza prevede il permesso d'autore (copyleft) si ha la garanzia che anche i miglioramenti pubblicati successivamente siano liberi.

Poter esibire un corpus di software libero sviluppato e pubblicato in loco è conveniente per un'istituzione di ricerca:

- Rappresenta motivo di orgoglio, perché costituisce una vetrina dove l'istituzione può esibire i propri contributi all'avanzamento della conoscenza scientifica e metterli a disposizione della comunità.
- Aiuta a ricevere contributi da più parti, perché i ricercatori sono spesso ben lieti di inviare miglioramenti e correzioni al software libero che viene pubblicato, in special modo se lo usano o lo studiano.
- Rendere chiaro fin dall'inizio di un progetto che i risultati saranno distribuiti pubblicamente con una licenza libera spesso riduce i problemi di attribuzione dei diritti. Per esempio, chiunque può modificare un programma abbandonato da un autore precedente. Inoltre gli autori sono spesso più inclini a contribuire a progetti di software libero piuttosto che a progetti che rimarranno di proprietà di un'istituzione.

È un luogo comune che, per ottenere ritorni economici, la strada più sicura è il software proprietario. Tuttavia, il software libero offre molti vantaggi per quanto riguarda le relazioni fra accademia e affari:

- Se, successivamente ad una fase di ricerca interna all'istituzione, lo sviluppo viene affidato ad uno spinoff, non ci saranno problemi di licenza, perché non nasceranno problemi di assegnazione dei diritti fra l'istituzione madre e lo spinoff.
- Pubblicare un programma come software libero è un buon modo per creare e promuovere nuovi standard. Si può pubblicare un programma libero dimostrativo, oppure come implementazione di riferimento per lo standard proposto, o altrimenti come implementazione di qualità industriale, completamente utilizzabile.
- Il trasferimento tecnologico è più semplice, perché scompaiono i problemi di assegnazione dei diritti all'atto di trasferire il programma dall'istituzione madre all'azienda.
- Sezioni commerciali interne o spinoff possono coesistere con gruppi di ricerca interni, e portare benefici reciproci, scambiando codice in entrambe le direzioni senza la necessità di accordi ad hoc, purché il codice sia ceduto con una licenza libera.

Come incoraggiare la produzione di software libero nelle istituzioni di ricerca

Noi sosteniamo che le istituzioni di ricerca dovrebbero incoraggiare i ricercatori ad usare licenze libere quando cedono o pubblicano software prodotto nella propria attività di ricerca.

In particolare, la ricerca finanziata pubblicamente dovrebbe, come regola generale, disseminare i risultati prodotti in forma di software pubblicandoli con una licenza libera, in maniera che possano essere liberamente studiati, copiati e modificati. Una tale politica avrebbe un effetto positivo su alcuni degli obiettivi primari della ricerca pubblicamente finanziata perché:

- contribuirebbe alla conoscenza pubblica;
- contribuirebbe all'avanzamento della ricerca;
- creerebbe un corpus di software sfruttabile industrialmente.

Ci sono diversi modi per incoraggiare la produzione di software libero in un'istituzione di ricerca scientifica.

Innanzitutto, ogni finanziamento ad un'attività di ricerca dovrebbe normalmente esigere che i risultati della ricerca siano pubblicati con una licenza di software libero. Se non è possibile imporre condizioni, l'istituzione che eroga il finanziamento dovrebbe chiaramente indicare che lo scopo del progetto è l'avanzamento della conoscenza pubblica, e che di conseguenza l'uso di licenze libere è incoraggiato.

Oltre alle istituzioni che erogano finanziamenti, *tutte* le istituzioni scientifiche dovrebbero avere una politica che imponga l'uso di licenze libere per il software prodotto con denaro pubblico. Se la licenza debba essere persistente o meno (copyleft o no) è questione da valutare. In generale, il vantaggio di una licenza senza permesso d'autore è di non impedire l'incorporazione del software in un programma proprietario, il che rende le cose più semplici per l'industria. Il vantaggio di una licenza con permesso d'autore è di obbligare all'uso di una licenza libera per la cessione degli aggiornamenti e miglioramenti, allargando così la base di software libero disponibile.

Le istituzioni di ricerca che finanziano ed incoraggiano gli spinoff dovrebbero favorire quelli che si basano su software libero rispetto agli altri, perché le attività economiche basate su software libero creano un ambiente economico sano e promuovono la crescita di competenze locali.

Da un punto di vista puramente accademico, scrivere e pubblicare software libero dovrebbe essere visto come un risultato scientifico di per sé, analogo a pubblicare un lavoro scientifico per mezzo di un articolo su rivista. Perché ciò diventi possibile, sarebbe necessaria una rete di revisori di software, analogamente a quella esistente per la pubblicazione degli articoli scientifici. Noi sosteniamo la necessità di creare una simile rete, e sosteniamo la necessità di revisionare il software libero pubblicato alla stessa maniera in cui si revisionano gli articoli scientifici, perché sia similmente considerato un risultato di ricerca.

Così come diffondere i propri risultati è ritenuto meritevole nella comunità dei ricercatori, così dev'essere ritenuta meritevole la diffusione di software. Le istituzioni di ricerca dovrebbero incoraggiare questo processo culturale dichiarando pubblicamente che pubblicare software libero è altrettanto importante che pubblicare altri risultati di ricerca, e organizzando adeguati processi interni di valutazione.

Così come gli autori e i lettori riconoscono il grande valore della pubblicazione dei precisi dettagli delle procedure descritte in un articolo scientifico, così devono riconoscere la necessità di poter leggere e modificare il sorgente dei programmi descritti negli articoli. Questa necessità dovrebbe essere presa in considerazione da editori, redattori e revisori di riviste scientifiche ogni volta che valutano un contributo per la pubblicazione.

Ringraziamenti

Ringrazio il Prof. Piero Maestrini per il suo supporto e le sue idee circa la revisione del software. Ringrazio anche l'Associazione software libero, per le discussioni e l'incoraggiamento.

Riferimenti

[Bez99] Open source development as a special type of academic research (critique of vulgar Raymondism), Nikolai Bezroukov, ottobre 1999, First Monday, vol. 4, n. 10.

[Kel01] Free software/free science, Christopher M. Kelty, novembre 2001, First Monday, vol. 6, n. 12.

Note

[1] In inglese la parola free in "free software" si riferisce alla libertà, non al prezzo; questa ambiguità non esiste nelle lingue latine.

Copyright © 2004-2005 Francesco Potortì

La più recente versione ipertestuale, completa di puntatori web, è disponibile ad <<http://fly.isti.cnr.it/sl/sl-e-ricerca>> assieme ad una versione in PDF ed una in Postscript.

La copia letterale e integrale di questo articolo è permessa con ogni mezzo, a condizione che questa nota sia riprodotta.

Ultimo aggiornamento: 2005-02-24