

# La libertà a partire dal software

autore e relatore: Francesco Galgani ([www.galgani.it](http://www.galgani.it))

*Talk in occasione del Linuxday 2008  
Gruppo Utenti Linux Pisa (GULP)*

## 0. Presentazione e ringraziamenti

### 1. Cosa significa "software"?

Il software è semplicemente la parte non fisica del computer.

Così come noi esseri umani siamo composti da una parte fisica, cioè il corpo, e una non fisica, cioè la mente, ognuna delle quali è indispensabile all'altra, così il computer ha una parte fisica, detta hardware, e una non fisica, detta software. Il software, quindi, è paragonabile a ciò che per noi è il pensiero.

Tralasciamo ogni definizione formale di software e consideriamo invece le attività quotidiane con il computer: **ogni volta che noi facciamo qualche cosa con un computer stiamo usando un programma e quel programma è un software.**

Chiarito ciò, la questione immediatamente successiva riguarda l'aggettivo "libero", ovvero cosa significa "software libero" e che differenza c'è con il "software proprietario".

### 2. Cosa significa "libero"?

#### Che differenza c'è tra software "libero" e "proprietario"?

Molti programmi largamente diffusi tra il grande pubblico, ottenuti sia dietro pagamento sia a titolo gratuito, non sono liberi, cioè **non rispettano le libertà fondamentali di cui ogni persona dovrebbe godere, creando problemi di collaborazione e ponendo seri ostacoli allo sviluppo della società.**

Capisco che per molti di voi tale affermazione può apparire molto forte.

Lo scopo di questa breve conversazione è proprio quello di spiegare:

- che cos'è il software libero;
- perché è così importante;
- perché c'è una reale esigenza di software libero;
- quali vantaggi concreti può portare agli utenti e alla società.

#### Breve introduzione

Anche se talvolta potrebbe esservi capitato di usare programmi senza conoscerne la relativa **licenza d'uso**, potete essere certi che una licenza d'uso c'è sempre. **Anche se non l'avete letta, il fatto di usare un programma implica necessariamente che ne stiamo accettando la licenza.**

Una licenza d'uso altro non è che *un contratto tra il titolare dei diritti sul software e l'utente, contratto che stabilisce diritti e doveri di entrambe le parti.*

Il software è legalmente equiparato alle opere d'ingegno, che ricadono sotto la normativa del diritto d'autore e, in alcune legislazioni, anche sotto la normativa dei brevetti.

Il diritto d'autore o il brevetto sono solitamente di proprietà dell'autore o del suo datore di lavoro: per questo motivo, quando **paghiamo** per **acquistare** un programma, **non se ne ottiene la proprietà**, ma solo la **possibilità di utilizzarlo** secondo quanto previsto dalle leggi vigenti e soprattutto della **licenza d'uso**.

Dal punto di vista della licenza e di ciò che essa garantisce o impone, il software si divide in numerose categorie.

In questa occasione parlerò soltanto del software libero e di ciò che lo differenzia dal software proprietario. Software libero e proprietario sono infatti le categorie principali di software, da cui le altre si differenziano per alcuni aspetti piuttosto che per altri.

### PRECISAZIONE

*“free software” = “software libero”*

*“free software” ≠ “software gratuito”*

*“software libero” ≠ “open source”*

Per evitare fraintendimenti e ambiguità, non userò le terminologie “open source” oppure “free software”, perché per quanto esse siano largamente utilizzate in modo intercambiabile con l'espressione “software libero”, in realtà il loro significato può essere male interpretato o addirittura cambiare in base al contesto.

### **3. Come sono nate queste idee? Chi le ha avute? Fanno parte di un movimento politico?**

No, queste idee non sono nate in un movimento politico, anche se sarebbe cosa buona e giusta che i politici si interessassero attentamente al software libero e alle inevitabili conseguenze positive del suo utilizzo! :) Conseguenze di cui vi parlerò tra poco.

Il software libero ha le sue origini nel mondo universitario americano degli anni 60/70, nel quale erano disponibili computer per i quali le università stesse provvedevano a scrivere il software per le loro esigenze, rendendoli utilizzabili e modificabili da chiunque, in forma abbastanza vicina a quella del pubblico dominio.

Facciamo un paragone culinario: queste università erano come delle grandi cucine e gli studenti erano al tempo stesso cuochi e commensali. Qual era la cosa più naturale se non quella che gli studenti stessi si scambiassero le ricette, in pieno spirito di condivisione? Non è forse questo ciò che è accaduto per millenni? C'è forse qualcosa di strano nello scambiarsi le ricette degli spaghetti al ragù o del sugo di pesce?

La stessa cosa accade con i computer: le ricette, invece di chiamarsi con questo nome, si chiamano algoritmi e codice, ma la finalità è la stessa, cioè produrre qualcosa che sia il più vicino possibile alle proprie esigenze, in questo caso il software.

Pare scontato che la libera condivisione del sapere, in questo caso espressa sotto forma di software, sia quindi una cosa naturale.

*Per maggior chiarezza vorrei far notare, ritornando al paragone culinario, che condividere la ricetta per fare una pizza è cosa ben diversa dal condividere la pizza stessa, anche condividere il software è cosa ben diversa dal condividere un bene materiale.*

*Se do la mia automobile a qualcun altro, significa che me ne sto privando.*

*Se offro la mia piazza a qualcun altro, significa che io non la mangerò.*

*Ma se io voglio condividere un'idea, non me ne sto privando, anzi, è probabile che dallo scambio di idee nasceranno ulteriori idee, magari migliori delle precedenti.*

Con il software è esattamente la stessa cosa: se io do la possibilità a qualcun altro di usare i programmi da me scritti, io non me ne sto privando.

Del resto, sapete bene che dato un programma, ciascuno di noi ha la possibilità di farne quante copie vuole e senza perdere nulla dell'originale.

Torniamo al mondo universitario americano.

All'inizio degli anni '80, molte università cominciarono ad usare software proprietari, rilasciati sotto licenze comprendenti clausole di non divulgazione: ciò limitava le possibilità di collaborazione tra le università.

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

Questo è ciò che accade anche oggi ogni volta che usiamo software proprietario.

Purtroppo, come se ciò non bastasse, un altro pericolo, ancor più subdolo e devastante, stava cominciando a limitare fortemente non soltanto la collaborazione, ma persino la formulazione di nuove idee. Torniamo nuovamente al paragone culinario: cosa succederebbe se gli spaghetti al pomodoro fossero brevettati? Riuscite a immaginarne le conseguenze?

... Nessuno potrebbe più cucinarsi un piatto di spaghetti al pomodoro senza pagare un compenso al proprietario del brevetto. Questo compenso si chiama royalty.

Non sarebbe neanche più possibile cucinare liberamente un qualunque piatto di spaghetti con qualunque tipo di sugo o ragù, perché qualunque nuova ricetta, che ha come base spaghetti e pomodoro, richiederebbe comunque il pagamento della royalty.

Brevettare un'idea, che sia essa una ricetta o un software, equivale a imporre una TASSA SULLE IDEE.

La brevettabilità del software, che dopo varie cause giudiziarie è stata per la prima volta applicata negli Stati Uniti nel 1981, rappresenta quindi un immenso danno per tutta la società.

In questo contesto, Richard Matthew Stallman, allora ricercatore del MIT (Massachusetts Institute of Technology), programmatore e hacker, fece una scelta difficile ma fondamentale per la nascita del software libero.

Allontanatosi dal MIT proprio per non rimanere obbligato a usare il software a quelle condizioni, nel settembre del 1983 lanciò il progetto GNU per creare un sistema operativo libero simile a Unix, di cui egli è il principale organizzatore e creatore.

Non mi dilungherò a spiegarvi che cosa è un sistema operativo e che cosa è Unix: per semplicità, diciamo che un sistema operativo è un software essenziale di cui non si può fare a meno per usare un computer, mentre Unix è un sistema operativo proprietario. Ciò che ha fatto Stallman, quindi, è stato l'avvio di un progetto che ha reso possibile lo sviluppo di un sistema operativo libero, che quest'anno compie il suo 25° anno.

Questo è un compleanno davvero speciale: 25 anni di GNU equivalgono a 25 anni di libertà.

Con il lancio del Progetto GNU, il movimento del software libero prese vita e oggi coinvolge milioni di persone, enti, associazioni.

Nell'ottobre del 1985 Stallman fondò la Free Software Foundation (FSF), che è una

organizzazione senza scopo di lucro, il cui obiettivo principale è la promozione della libertà universale di distribuire e modificare il software senza restrizioni.

La FSF pone anche condizioni per la corretta definizione del “software libero”.

### **LE QUATTRO LIBERTA'**

Il “software libero” è quel software la cui licenza soddisfa le seguenti quattro condizioni di libertà:

- la libertà di usare il programma, per qualunque scopo;
- la libertà di studiarne il funzionamento e di adattarlo ai propri bisogni;
- la libertà di ridistribuirne copie, in modo tale da poter aiutare il prossimo;
- la libertà di migliorare il programma e di distribuire tali migliorie, in modo tale che l'intera comunità ne tragga beneficio.

Il software “proprietario” è quel software che viene rilasciato sotto licenze che concedono all'utente solo ed esclusivamente l'utilizzo del prodotto, sotto condizioni restrittive.

Prendersi arbitrariamente anche una sola delle suddette quattro libertà fondamentali senza il consenso di chi ha venduto il software, cioè violando la licenza d'uso, anche a fin di bene, anche per aiutare un amico, o semplicemente perché non si voglia o non si possa spendere il denaro richiesto per la licenza d'uso, corrisponde di fatto a un reato civile o penale, a seconda della legislazione.

Posto in questi termini, il software libero è anche una questione morale e non soltanto una questione pratica.

### **IL COPYLEFT**

A conclusione di questa breve parentesi storica, vi faccio notare che Stallman fu il pioniere del concetto di copyleft (“permesso d'autore”), un concetto fondamentale per il software libero e in contrasto con il concetto di copyright (“diritto d'autore”). Il concetto è semplice: il permesso d'autore (copyleft) usa le leggi sul diritto d'autore (copyright), ma le capovolge per ottenere lo scopo opposto, ovvero invece che un mezzo per privatizzare il software, diventa un mezzo per mantenerlo libero.

La “GNU GPL” (GNU General Public License) è la licenza copyleft più diffusa.

Vediamo adesso la rilevanza sociale ed economica che avrebbe l'applicazione generalizzata di quanto esposto ed auspicato.

#### **4. Quali sarebbero le conseguenze della corretta applicazione di queste idee nella vita quotidiana?**

**Quali sarebbero i vantaggi effettivi?**

**Ci sono delle buone ragioni per fare lo sforzo di provare a usare software libero?**

**Sono soltanto proposte inapplicabili o ci sono vantaggi concreti?**

Ovviamente, se noi convinti sostenitori del software libero, oggi, qui e nel resto mondo, sosteniamo una battaglia in difesa della libertà e vi invitiamo a unirvi a noi, le nostre ragioni vanno ben oltre le semplici proposte.

#### **A) Per cominciare parliamo di SOLDI e di LAVORO**

1. Risparmio economico sia sul software sia sull'hardware, potenzialmente nell'ordine di miliardi di euro! Facciamo un esempio pratico: da quando le scuole italiane in Provincia di Bolzano sono passate al Software Libero, esse risparmiano un milione di euro all'anno. Pensate se questo passaggio avvenisse in tutta Italia, o meglio in

tutto il mondo, quale sarebbe il risparmio!

Per chi fosse interessato ad approfondire questa notizia, segnalo la puntata di Report (RAI3) che se ne è occupata, che può essere vista all'indirizzo:

<http://www.youtube.com/watch?v=ITzX7zRgOkA>

2. Qualità e valore economico di alcuni progetti assai alti.  
Kernel 2.6 = software libero, scaricabile gratuitamente da [www.kernel.org](http://www.kernel.org)  
Kernel 2.6 = \$611,757,037 (stima del costo necessario per svilupparlo ex-novo)

*per dettagli su questa stima:*

<http://www.dwheeler.com/essays/linux-kernel-cost.html>

3. Maggiori possibilità nel mondo del lavoro
  - maggiore flessibilità mentale;
  - maggiore capacità di adeguarsi ai cambiamenti;
  - larga diffusione del software libero
4. Ideale per la scuola e la pubblica amministrazione.
5. Forte interesse sia da parte di privati, sia di aziende piccole, medie e grandi, che finanziano molti progetti di software libero e che si servono del software libero per le loro attività.  
Possiamo considerare il caso di Google, che fa del software libero un pilastro portante del suo business e che sta finanziando non soltanto la Mozilla Foundation con 56,8 milioni di dollari per lo sviluppo di Firefox, ma anche tanti altri progetti con cifre considerevoli.

## **B) GARANZIE CONCRETE E SICUREZZA**

1. Uso di standard condivisi, certificati e documentati, per garantire:
  - accesso ai documenti anche dopo molto tempo;
  - accesso ai documenti anche senza il programma che li ha memorizzati; rispetto della privacy, poiché le procedure di trattamento dei dati sono sempre "note".
2. Maggiore sicurezza. Garanzie concrete e non "a parole":
  - verificabilità dei sistemi di sicurezza;
  - aggiornamenti tempestivi;
3. Protezione degli investimenti:
  - investimento non in beni, ma in persone e know-how (conoscenze e competenze);
  - non è detto che vi sia un risparmio immediato;
  - maggior indipendenza da evoluzioni tecnologiche;
  - minor obsolescenza dell'hardware;
  - maggior indipendenza dai fornitori di software;
  - investimento a lungo termine.
4. Buon funzionamento dei sistemi GNU/Linux:
  - assenza di virus;
  - ottima stabilità;
  - controllo completo del sistema.

## C) CRESCITA CULTURALE

1. Disponibilità di una comunità assai ampia di persone pronte a offrire la loro assistenza, formata da una miriade di gruppi, associazioni, forum, con luoghi d'incontro sia virtuali (cioè su Internet) sia reali (ad esempio, le sedi dei vari LUG).
  - Scambio di informazioni, esperienze, programmi.
4. Documentazione "libera", solitamente disponibile in tante lingue: niente viene tenuto nascosto!
  - Possibilità di accrescere le proprie conoscenze e competenze.
5. Crescita culturale, specialmente nelle nuove generazioni.

## D) FORNIRE GLI STRUMENTI GIUSTI

Offerta assai ampia (e quasi sempre gratuita) di applicativi di ogni genere e per ogni esigenza, disponibili in una grande varietà di lingue, di sistemi operativi e piattaforme, da semplici giochi a programmi professionali, fino a strumenti per grandi aziende e centri di ricerca.

Tabelle con equivalenze tra software libero e proprietario:

[http://linuxpedia.netsons.org/index.php/Tabella\\_di\\_equivalenza\\_programmi\\_Windows/\\_Linux](http://linuxpedia.netsons.org/index.php/Tabella_di_equivalenza_programmi_Windows/_Linux)

[http://rsoftware.altervista.org/php5/index.php?mod=none\\_linux&file=winlinux](http://rsoftware.altervista.org/php5/index.php?mod=none_linux&file=winlinux)

Software libero per Windows:

[http://linuxpedia.netsons.org/index.php/Software\\_Libero\\_per\\_Windows](http://linuxpedia.netsons.org/index.php/Software_Libero_per_Windows)

TheOpenCD:

<http://linux.studenti.polito.it/theopencd.php>

## E) PERFETTA INTEGRAZIONE E MASSIMA FLESSIBILITA'

1. L'uso di software libero non preclude la possibilità di usare software proprietario, ma anzi favorisce la coesistenza e l'integrazione.  
Esempi tipici sono:
  - Dual boot
  - Uso di programmi liberi all'interno di sistemi operativi proprietari
  - Uso di programmi proprietari all'interno di sistemi operativi liberi
2. Massima flessibilità e adattabilità a contesti ed esigenze particolari. Ad esempio "GNU/Linux" viene usato non soltanto su computer fissi e portatili, vecchi e nuovi, ma anche su piccoli dispositivi... e perfino dalla NASA nello Shuttle!

Il concetto di "software libero" è infatti applicabile ovunque ci sia del software. quindi non soltanto su computer, ma anche su telefoni cellulari, console per giocare (tipo Playstation), navigatori GPS, televisori, lavatrici, automobili, apparecchiature mediche, ecc.. Il fatto che su tali dispositivi ci sia o non software libero non è la conseguenza di problemi tecnici, ma di precise scelte da parte dei produttori.

3. La massima flessibilità dei sistemi operativi liberi porta in modo naturale all'esistenza di tante "distribuzioni", ovvero di tanti modi diversi per offrire "agglomerati" di software libero in modo da soddisfare le esigenze di certe categorie di utenti piuttosto che di altre.

Alcune distribuzioni, come Ubuntu, sono molto popolari, gratuite e semplici da usare.

Altre distribuzioni, specialmente quelle rivolte ad ambienti aziendali, sono a pagamento e accompagnate da supporto tecnico commerciale.

Esistono centinaia di distribuzioni.

Sul sito DistroWatch ci sono informazioni su un gran numero di distribuzioni:

<http://distrowatch.com/>

Questa grande varietà potrebbe disorientare. Per aiutarvi a comprendere come, in realtà, avere molte opportunità non sia un problema, vorrei farvi un altro paragone gastronomico.

Consideriamo il pane. Quando ci rechiamo in un supermercato o in un panificio per acquistare il pane, possiamo scegliere il tipo che più ci piace tra tanti, che sono diversi per forma, colore, sapore, ingredienti, tipo di lievitazione, ecc..

Qualunque tipo sceglieremo, sempre di pane si tratterà. La stessa cosa vale con le distribuzioni di GNU/Linux, che differiscono per molteplici aspetti, adattandosi così a tanti “gusti” diversi.

## **5. Alcuni modi per contribuire al software libero**

METODI DI BASE (per tutti):

Diffondere cultura e informazione sul software libero.

Usare software libero.

Incentivare l'uso del software libero.

COLLABORAZIONI DIRETTE (non è necessario essere sviluppatori):

Tradurre software libero.

Tradurre la documentazione a corredo di software libero.

Provare software libero per segnalare problemi o possibili migliorie agli sviluppatori.

ATTIVITA' DI SVILUPPO:

Migliorare software libero preesistente.

Sviluppare nuovo software libero.

CONTRIBUTI ECONOMICI:

Donazioni volontarie ad uno o più progetti.

Adesione alla Fellowship della FSFE (Free Software Foundation Europe) per supportarne le attività: [www.fsfe.org](http://www.fsfe.org)

## **6. Discorso conclusivo**

Come utenti di computer, oggi ci si può trovare ad usare un programma proprietario. In questo caso, dovremmo rifiutare la copia ad un amico che ce la chiede. Se riteniamo però più importante la collaborazione del diritto d'autore, ci sentiremo in obbligo di dargli quella copia, seppur di nascosto. 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 e "No" ai brevetti sul software.

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 figli e ai vostri studenti. Meritate di poter assumere il vostro programmatore favorito per aggiustare il software quando non funziona o per adattarlo a nuovi bisogni.

In altre parole, meritate il software libero.

E con questo ho concluso. Grazie.