Software Libero e Brevetti sul Software

Gianluca Amato

Facoltà di Economia
Università “G. D'Annunzio” di Chieti-Pescara
ultimo aggiornamento: 21/03/11

Un po' di terminologia

Hardware e Software

- Nell'ambito dell'informatica, due termini in lingua inglese molto utilizzati sono hardware e software.
  - Hardware: indica tutto ciò che ha una consistenza fisica. Monitor, stampante, tastiera, mouse, etc. sono esempi di hardware.
  - Software: indica tutto ciò che non ha consistenza fisica, quindi i programmi (Windows, Linux, Word, ...)
    - In senso più generale, ma non in queste note, con software si intendono anche file audio (ad esempio gli MP3) o video (i film in DVD).
- Quando comprate dell'hardware, pagate un prezzo che comprende
  - Il costo per produrre fisicamente l'oggetto in questione.
    - Questo costo è irrisorio se si parla di software.
  - Il costo sostenuto per la fase di ricerca che ha portato allo sviluppo del prodotto.

Computer e Programmi

- Un programma per computer non è altro che una serie di istruzioni, a un altissimo livello di dettaglio, che istruiscono il computer su come svolgere determinate operazioni:
  - Cosa fare quando l'utente fa un click su una icona sullo schermo?
  - Cosa fare quando viene premuto il tasto Stampa?
  - Come fare a interpretare il contenuto di un DVD e visualizzarlo sullo schermo?
- Una analogia interessante è con la cucina:
  - Un programma è l'equivalente di una ricetta.
  - Il computer equivale a un cuoco.
  - Così come il computer esegue un programma e ci da dei risultati sullo schermo o sulla stampante, il cuoco esegue la ricetta per produrre un gustoso pranzetto!

Programmi e linguaggi

- Per specificare le istruzioni si usano i linguaggi di programmazione.
  - C, Java, Pascal, etc..
- Problema: i computer sono stupidi, capiscono un solo linguaggio di programmazione: il linguaggio macchina.
- Ma il linguaggio macchina è troppo cervellotico per un essere umano.. come si risolve il problema?
- C'è un programma speciale, chiamato compilatore.
  - trasforma da un linguaggio di programmazione “per esseri umani” a linguaggio macchina

Il programma “Hello World”

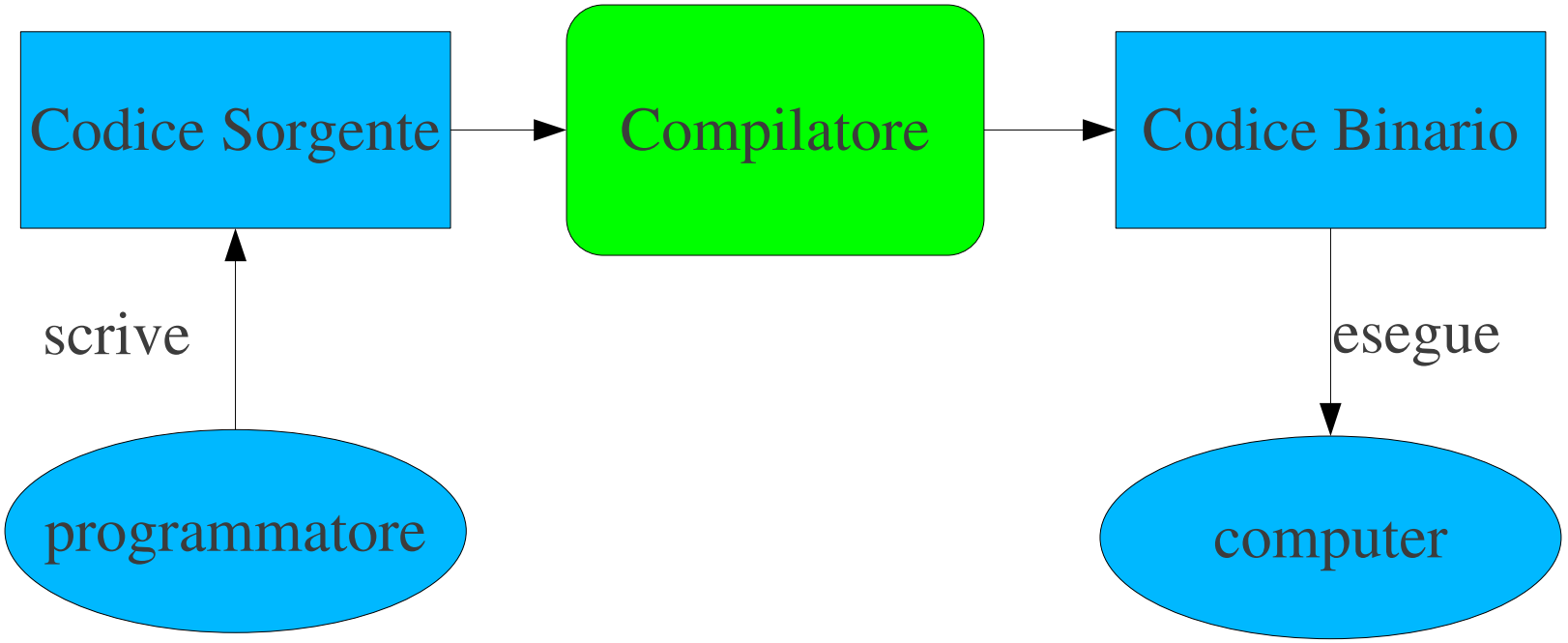
```
class Hello {
public static void main(String[]args) {
      System.out.println("Hello world!");
  }
}
```

Codice in Java

```
.data
msg:
        .string "Hello, world!\n"
msgend:
        .equ len, msgend-msg

.text

        .global _start

_start:
        mov $4,%eax
        mov $1,%ebx
        mov $msg,%ecx
        mov $len,%edx
        int $0x80

        movl $1,%eax
        xorl %ebx,%ebx
        int $0x80
```

Codice in linguaggio macchina

Codice binario e sorgente



- Il programma si chiama
  - Codice binario: (o anche codice oggetto) quando è scritto in linguaggio macchina
  - Codice sorgente: quando è scritto in un linguaggio di programmazione ad altro livello.

Il software libero

Programmi e diritti d'autore

- Le regole internazionali sui diritti d'autore proteggono il software come opera dell'ingegno (come la musica, i libri, etc..)
  - Nessuno può utilizzare il programma senza permesso.
  - All'utente finale, il permesso di utilizzo di un software viene concesso tramite una licenza d'uso.
- La maggior parte delle licenze sono restrittive:
  - Vietano la copia del software.
    - Spesso è vietata anche fare una copia di riserva, nel caso il supporto fisico originale si danneggi.
  - Vietano di ridistribuire il software, sia a pagamento che gratis.
    - Non potete dare il programma da voi acquistato a un vostro amico.
  - Vietano di modificare il programma per adattarlo alle proprie esigenze, o per correggerne i difetti.

Esempio di licenza

- Ecco, ad esempio, alcuni punti della licenza per Windows XP
  - 1.1 Installazione e Utilizzo. L'utente potrà installare,utilizzare, accedere a, visualizzare ed eseguire una copia del Software su di un singolo computer, quale una workstation, un terminale o un altro dispositivo ("Computer Workstation"). Il Software non potrà essere utilizzato da più di due (2) processori in uno specifico momento su un singolo Computer Workstation.
  - DIVIETO DI HOSTING DI SERVIZICOMMERCIALI/LOCAZIONE. L'utente non potrà concedere il Software in locazione, leasing, prestito, ovvero fornire hosting di servizi commerciali utilizzando il Software.
  - RESTRIZIONI SULLA DECODIFICAZIONE, SULLADECOMPILAZIONE E SUL DISASSEMBLAGGIO. L'utente non potrà decodificare, decompilare o disassemblare il Software, fatta eccezione per i casi in cui le suddette attività siano espressamente consentite dalla legge applicabile.

Licenze e codice sorgente

- Tra l'altro i programmi, sono di solito distribuiti soltanto sotto forma di codice binario.
- Il codice sorgente è tenuto segreto dal produttore
  - Per impedire che altri possano copiare le idee utilizzate nella scrittura del programma.
  - Anche quando la modifica del programma non è espressamente vietata, senza il codice sorgente è quasi impossibile farlo. Capire il funzionamento di un programma dal codice binario è una impresa titanica!!
- Ma è questo l'unico modo di produrre e distribuire il software?

Il software libero

- Nel 1984 Richard M. Stallman, ricercatore e hacker al MIT, lascia il proprio lavoro per protesta contro una politica sempre più restrittiva seguita dal MIT per i diritti d'autore sul software
  - Nasce il progetto GNU: l'obiettivo è produrre un sistema operativo completamente libero.
  - Poco dopo nasce anche la Free Software Foundation, una società senza fini di lucro il cui scopo è promuovere la diffusione del software libero.
- Ma cosa vuol dire software libero?
  - Stallman identifica quattro libertà principali che a un utente devono essere garantite.
  - Un programma la cui licenza garantisce all'utente queste libertà è un software libero.
  - L'opposto del software libero è il software proprietario.

Una parentesi: hacker e cracker

- Si sente spesso parlare nei giornali o in televisione del termine hacker in maniera negativa, come di un pirata informatico.
  - L'uso di questo termine è sbagliato.
- Hacker: una persona che ha una conoscenza profonda di un sistema informatico e di come è possibile utilizzarlo al limite delle possibilità in opposizione al semplice utente, che si accontenta di imparare quanto necessario.
- Cracker: una persona che viola i sistemi di sicurezza. Corrisponde al pirata informatico
  - Un hacker ha spesso le conoscenze tecniche per essere un cracker, ma ciò non vuol dire che lo sia.
  - Un cracker può essere un hacker oppure no, perché usa semplicemente dei programmi già fatti per violare i sistemi di sicurezza.

Libertà fondamentali

- Queste sono le 4 libertà fondamentali:
  - Eseguire il programma, per qualsiasi scopo
    - Quindi senza nessuna restrizione che ne impedisce, ad esempio, l'utilizzo per scopi commerciali.
  - Studiare come funziona il programma e adattarlo alle proprie necessità.
  - Copiare il programma e ridistribuire copie.
    - Quindi un software libero può essere copiato, regalato a un amico
  - Modificare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio.
- Dalla seconda e quarta libertà segue anche che è obbligatorio che per il software libero sia disponibile pubblicamente il codice sorgente. Modificare un programma senza codice sorgente è impossibile.

Il software libero è gratuito?

- Il concetto di software libero non è direttamente collegato alla questione del suo prezzo.
  - Se dispongo di un software libero, posso cederlo a chi voglio gratuitamente...
  - ... ma posso anche venderlo.
- Esistono molte raccolte di software libero su CD-ROM.
  - Ad esempio, GNU/Linux è un software libero, si può scaricare da Internet senza pagare nulla..
  - Ma se volete evitare la fatica di scaricarlo, potete comprare i CD con il programma.
- Vendere software libero è un ottimo mezzo per ottenere fondi per finanziare lo sviluppo di altro software libero.

Software libero e commercio

- Il mondo del software libero rigetta una diffusa pratica commerciale, che è quella dello sviluppo di software proprietario, ma non è contro tutte le attività commerciali.
- Abbiamo già detto che vendere del software libero è una pratica considerata ragionevole e anzi incentivata da chi sviluppa software libero.
- Altre attività commerciali possibili sono:
  - Fornitura di supporto tecnico a pagamento.
  - Modifiche su richiesta di software per adattarlo a specifiche esigenze.
  - Produzione e vendita di manuali e libri.
- Sempre più aziende hanno una politica commerciale basata sulla diffusione e sviluppo di software libero.

Permesso d'autore (1)

- Se un programma è libero quando esce dalle mani di un autore, non è detto che altri programmi da esso derivati lo siano.
  - La sua licenza potrebbe consentire a un altro soggetto di prendere il programma, modificarlo e distribuire la modifica con una licenza diversa, proprietaria e non più libera.
  - Chi acquista il software modificato non ha più un software libero.
- Una licenza libera che consente questa “trasformazione” da software libero a proprietario è, ad esempio
  - la licenza MIT X-11 (Massachusetts Institute of Technology)
  - le licenze BSD.

Permesso d'autore (2)

- E se si vuole evitare che il proprio lavoro possa essere utilizzato in un software in proprietario?
  - Si usa una licenza basata sul “permesso d'autore” (copyleft).
  - Il permesso d'autore (copyleft) usa le leggi del diritto d'autore (copyright) ma per ottenere lo scopo opposto:
    - Invece che un metodo per privatizzare il software, diventa un metodo per mantenerlo libero.
- Il succo dell'idea è questa:
  - All’utente di un programma con lincenza basata su copy-left sono garantite le 4 libertà fondamentali.
  - Tuttavia, tutte le modifiche al programma vanno distribuite con la licenza originale.
- La prima licenza copyleft e la più famosa è la GNU GPL (General Public License) con cui è distribuito Linux.

Con o senza copy-left?

- Tradizionalmente, si è voluto affermare che
  - Le licenze senza copyleft puntano a favorire la massima diffusione del software, a scapito della libertà degli utenti.
  - Le licenze con copyleft hanno l'obiettivo primario di difendere la libertà degli utenti, anche a scapito della sua diffusione, che potrebbe essere rallentata dall'impossibilità di utilizzarlo in ambienti proprietari.
- Non tutti concordano con questo punto di vista
  - Secondo loro, le licenze senza copyleft sono più libere, proprio perché non impongono restrizioni di nessun tipo agli utenti...
  - ... ovviamente dipende dal significato esatto che si da al termine "libertà"

Un mondo di licenze

- La creazione di una nuova licenza libera è una cosa complicata...
  - Problemi di incompatibilità (quasi sempre non volute) tra licenze
    - Rendono impossibile unificare due programmi distribuiti con licenze diverse..
  - Difficoltà pratiche inaspettate nell'uso di una licenza
    - l'esempio più famoso è la licenza BSD originale.
- La BSD originale è una licenza senza copyleft:
  - Prevede l'obbligo, per qualunque materiale pubblicitario menzioni il software, di aggiungere questa clausola:
    - This product includes software developed by the University of California, Berkeley and its contributors.
  - Cosa succede se 100 persone utilizzano questa licenza, modificando la clausola per inserire il proprio nome?

Software di Pubblico dominio

- Prima ancora della diffusione del concetto di software libero esisteva il software di pubblico dominio.
- Nel software di pubblico dominio l'autore rinuncia ai diritti d'autore.
  - Il software è disponibile pubblicamente, e chiunque ne viene in possesso può fare con questo ciò che si desidera.
- Il software di pubblico dominio è software libero qualora sia disponibile in codice sorgente
  - Se è disponibile solo il codice binario, per quanto detto prima, non si può considerare libero
- Non tutto il software libero è di pubblico dominio.
  - Il software distribuito con licenze stile MIT o con copyleft non è di pubblico dominio.

Altre categorie di software

- Software Freeware: non esiste una definizione univoca. Di solito si intende del software gratuito non modificabile. Non si ha accesso ai sorgenti e talvolta neanche possibilità di redistribuzione.
  - Esempio: Internet Explorer
- Software Shareware: software di cui è permessa la distribuzione ma che bisogna pagare per l'uso. Spesso non si ha accesso ai sorgenti.
  - Esempio: Winzip
- Software commerciale: è un software sviluppato da una azienda per trarre profitto dal suo uso. Può essere libero oppure proprietario.
  - Esempio di software commerciale ma libero: OpenOffice
  - Esempio di software commerciale proprietario: Micorosft Office

Doppia licenza

- Tra i produttori di software commerciale libero è molto diffusa la pratica della doppia licenza
  - Lo stesso software viene distribuito in due modi diversi:
    - gratuitamente con una licenza libera (spesso con clausola copyleft);
    - a pagamento con una licenza proprietaria.
  - Vantaggio:
    - La maggior parte degli utenti usa la versione libera che (essendo protetta da copyleft) non potrà essere utilizzata da dei concorrenti per produrre varianti proprietarie del software.
    - Chi ha bisogno di utilizzare il software all'interno di un prodotto proprietario (per cui non può utilizzare la versione gratuita con copyleft) può richiedere a pagamento la versione proprietaria.
- Ad esempio: MySQL è distribuito con licenza GPL o proprietaria.

Software Open Source (1)

- Un termine che si è molto diffuso ultimamente è quello di software open source (sorgente aperto).
- Dal punto di vista pratico, software libero e software open source sono quasi la stessa cosa, ma i principi che stanno alla base delle due terminologie sono molto diversi:
  - Software libero: pone l'accento sulla questione della libertà degli utenti.
  - Software open source: difende l'idea che il modello di sviluppo del software libero (sorgenti disponibili a tutti, chiunque può modificare e migliorare i prodotti...) crei dei prodotti migliori. Il principio di libertà degli utenti è assente o in secondo piano.
- Un saggio che si può considerare il manifesto del movimento open-source è "*La cattedrale e il bazaar*" di Eric S. Raymond
  - http://www.apogeonline.com/openpress/doc/cathedral.html

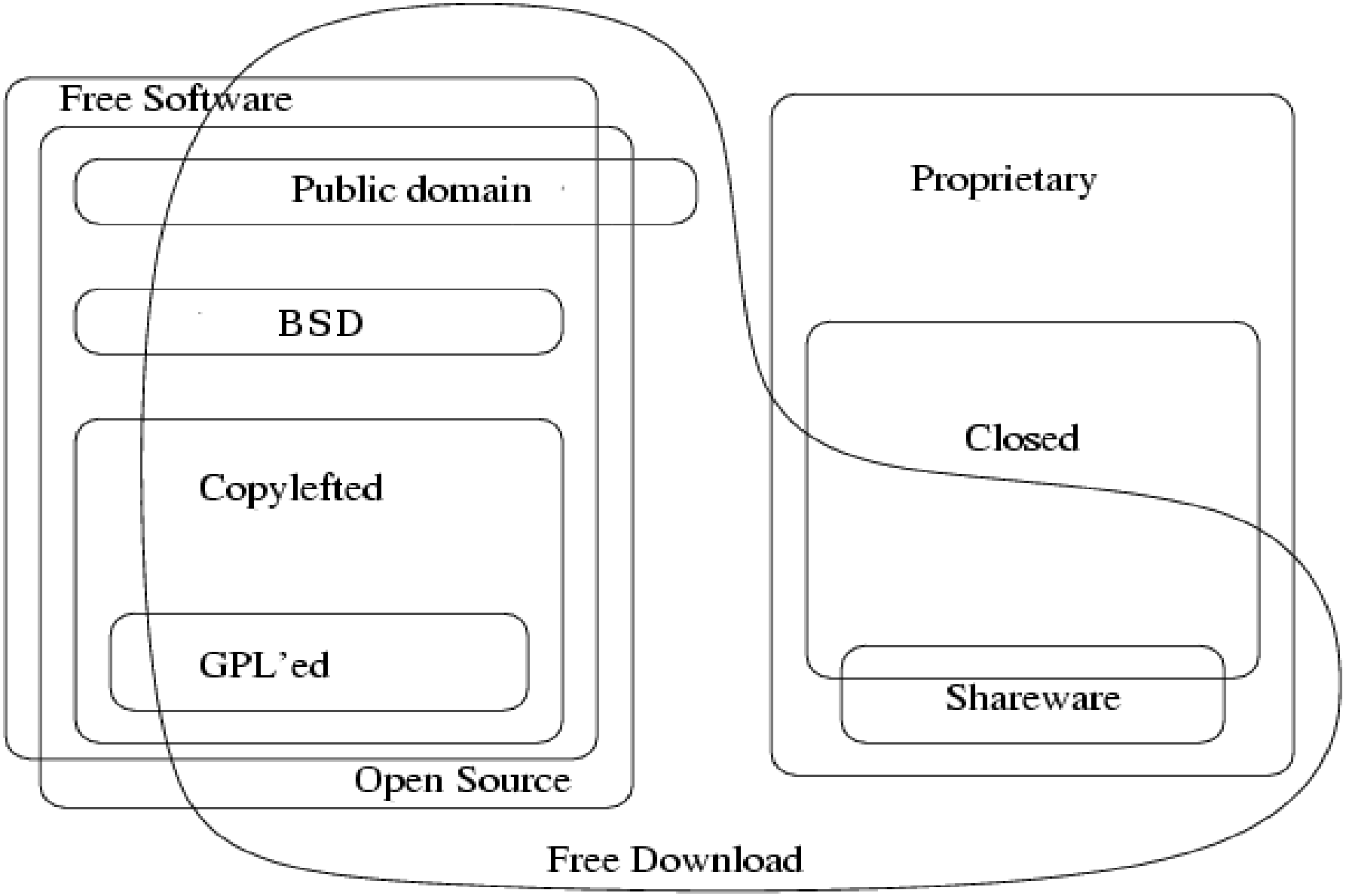
La cattedrale e il bazaar

- Sono due metodi di sviluppo di software
  - La cattedrale
    - Il programma viene realizzato da un numero limitato di esperti che scrivono il codice in quasi totale isolamento. Il progetto ha una suddivisione gerarchica molto stretta e ogni sviluppatore si preoccupa della sua piccola parte di codice.
    - Tipico del software proprietario, ma utilizzato anche per qualche software libero.
  - Il bazaar
    - Il codice sorgente della revisione in sviluppo è disponibile liberamente, gli utenti possono interagire con gli sviluppatori e se ne hanno le capacità possono modificare e integrare il codice. Lo sviluppo e decentralizzato e non esiste una rigida suddivisione dei compiti.
    - Tipico del software libero.

Software Open Source (2)

- L'idea di software open source è più appetibile alle aziende, in quanto pone l'accento sui vantaggi competitivi e quindi sui profitti.
  - Di contro, il termine è particolarmente avversato dalle organizzazioni come la FSF (Free Software Foundation) che vedono il motivo fondamentale dell'esistenza del software libero/open source non nella qualità dello stesso ma, appunto, nella libertà che garantisce.
- Il termine si è diffuso di più del termine “software libero”, grazie proprio alla spinta delle aziende private e della Open Source Initiative (una controparte pro-open source della FSF)

Mappa del software



Software Proprietario (1)

- I produttori di software, per difendere il software proprietario, parlano di “difendere i loro diritti” o di “fermare la pirateria”.
  - Le loro richieste sono in linea con la giurisdizione dei nostri giorni...
  - ...ma quello che è preoccupante nelle loro affermazioni sta nelle assunzioni inespresse.
- 1^ assunto: le aziende produttrici di software anno il diritto naturale di proprietà del software e, di conseguenza, il controllo sui suoi utenti.
  - Non è vero. La costituzione degli USA, ad esempio, considera il diritto d'autore non un diritto naturale di chi crea un prodotto, ma una limitazione al diritto naturale degli utenti di copiarlo.

Software Proprietario (2)

- 2^ assunto: non avremmo software utilizzabile se non riconoscessimo ai produttori il controllo su quello che un utente può fare.
  - È quello che giustifica in primo luogo il diritto d'autore.
  - Potrebbe sembra plausibile, ma il movimento del software libero ha dimostrato che ciò è falso
- 3^ assunto: quello che importa in un software è quello che ci consente di fare.
  - Non importa, secondo questo punto di vista, il modo in cui questo software è prodotto e il tipo di società in cui si vive.
- Se rinunciamo a questi assunti e giudichiamo il mondo del software mettendo in primo piano l'utente, arriviamo a conclusioni ben diverse.

Pirateria

- Anche il termine stesso pirateria è discutibile..
  - Cosa ha a che fare una copia illegale di un software coi rapimenti e gli omicidi tipicamente praticati dei veri pirati?
  - Ancora un volta, il termine pirata viene scelto dai produttori di software e dalle loro associazioni per inculcare una idea predeterminata del mondo del software.
- Piuttosto che usare il termina pirateria andrebbe usato qualcosa di più neutro come copia proibita o copia non autorizzata.
- Qualcuno potrebbe anche preferire termini "positivi" come "condividere informazioni con il tuo vicino".
  - La frase "la pirateria è illegale" è scontata..
  - .. ma "condividere informazioni con il tuo vicino è illegale" può essere molto fastidioso.

Hardware segreto

- Il mondo del software libero ha ottenuto grandi successi.. ma tante sono le difficoltà che ancora si incontrano.
- Sempre più spesso, i costruttori di hardware mantengono segrete le specifiche dei loro prodotti.
  - Senza le specifiche tecniche, non è possibile scrivere software in grado di sfruttarle.
  - Esempio tipico (almeno in passato): i winmodem
    - Modem pensati per funzionare solo con il sistema operativo Windows, dei quali non viene rilasciata documentazione tecnica
- Due soluzioni:
  - Ricostruire le specifiche hardware con un processo noto come reverse engineering
    - Che può anche essere illegale in alcune nazioni.
  - Educare alla scelta di hardware supportato da software libero.

Brevetti software

- I brevetti software possono rendere impossibili da implementare algoritmi e funzionalità.
  - Ad esempio, metodi di compressione (come l'MP3) possono essere brevettati e i programmatori non possono scrivere software che usa questi metodi, se non pagando per ottenere un permesso.
- La situazione è resa piuttosto grave dai bassi standard qualitativi dell'ufficio brevetti degli USA, che consente di brevettare anche idee banali.
  - Esempio: il singolo click per l'acquisto online di Amazon.
- Soluzioni:
  - Provare la non validità di un brevetto
  - Usare metodi alternativi per svolgere la stessa funzione.
    - Usare il formato Ogg Vorbis invece che MP3

Vantaggi pratici

- Il prezzo (ma non è detto)
  - Posso anche pagarlo caro se richiedo un contratto di assistenza.
- Indipendenza dal fornitore
  - Se un altro fornitore offre una soluzione migliore, sono libero di cambiare, senza perdere i dati.
- Verificabilità del codice
  - Posso essere sicuro che il programma non contenga nulla di insolito..
  - ...altrimenti, come fare a essere sicuri che il programma che sto utilizzando non diffonda dei documenti privati a persone estranee?
- Riutilizzabilità del codice
  - Se il programma non fa esattamente quello che dovrebbe, posso adattarlo alle mie esigenze senza riscriverlo da capo!

Vantaggi sociali

- Patrimonio pubblico
  - Il software libero si configura come un bene pubblico a disposizione di tutti. Viene perciò a costituire una infrastruttura al servizio della società.
- Accesso alla tecnologia
  - Il software libero consente di superare il divario digitale che divide paesi ricchi e poveri.
- Valore formativo
  - La libertà di studiare e modificare il codice sorgente mette tutti in grado di imparare ed operare con software allo stato dell'arte. Lo sviluppo collaborativo permette la partecipazione diretta.
- Supporto alle economie locali
  - La competizione si sposta dal software (in mano alle multinazionali) a quelli dei servizi (e le realtà locali competere).

Filosofia GNU

- Concludo questa presentazione del software libero con un estratto di un famoso scritto di R. Stallman:
  - R. Stallman. Perché il software non deve avere padroni

    http://www.gnu.org/philosophy/why-free.it.html

- 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

Riferimenti (1)

- Associazione Software Libero
  - http://www.softwarelibero.it
- Classificazione del software libero e non
  - http://www.gnu.org/philosophy/categories.it.html
- Licenze del software libero (in inglese)
  - http://www.gnu.org/licenses/license-list.html
- Cos'è il copyleft (in inglese)
  - http://www.gnu.org/copyleft/copyleft.html
- Cos'è il software libero ?
  - http://www.gnu.org/philosophy/free-sw.it.html
- Sito della Free Software Foundation
  - http://www.fsf.org

Riferimenti (2)

- Sito della Free Software Foundation Europe
  - http://www.fsfe.org
- Sito della Open Source Initiative
  - http://www.opensource.org/
- Sito della Business Software Alliance, associazione di produttori di software
  - http://www.bsa.org/italia
- Sito di Ubuntu Linux, sistema operativo commerciale e libero
  - http://www.ubuntu.com

I Brevetti sul Software

Strumenti di tutela del software

- Ci sono tre strumenti giuridici per la tutela di un prodotto:
  - Marchio di fabbrica
  - Diritto d'autore
  - Brevetto
- Nel mondo del software gli strumenti relativi al marchio di fabbrica e al diritto d'autore sono stati usati da sempre.
  - Sul marchio di fabbrica non c'è mai stata nessuna opposizione
  - Sul diritto d'autore c'è qualche obiezione non tanto sull'idea che il software debba godere di questa tutela quanto sui metodi utilizzati per porla in atto
    - Vedi ad esempio il decreto Urbani che prevede sanzioni penali per chi scambia materiale protetto da copyright...

Brevetti ed Unione Europea

- Recentemente l'Unione Europea ha considerato la possibilità di applicare il diritto brevettuale al software, in analogia a quanto avviene negli USA.
  - Forti opposizioni da parte di aziende produttrici software e mondo delle ricerca.
  - Appoggio (anche se non in maniera del tutta esplicita) dalla BSA, Business Software Alliance, una associazione di grandi produttori di software (vedi http://swpat.ffii.org/gasnu/bsa/index.en.html)
- Il Parlamento Europeo, almeno per ora, ha deciso di lasciare la decisione ai singoli stati:
  - In Italia, attualmente, il software non è brevettabile
- Perché introdurre i brevetti sul software?
  - È veramente uno strumento utile per lo sviluppo dell'Europa?
  - Chi si vuole tutelare con i brevetti software?

La proprietà intellettuale

- A difesa del brevetto sul software si sentono spesso frasi del tipo:
  - "La ricchezza del futuro è la proprietà intellettuale, è necessario tutelare questa proprietà e, pertanto, si deve ricorrere ai brevetti sul software"
- Si tratta di affermazioni che nel campo del software sono sbagliate!
  - Ci sono altri modi di difendere la proprietà intellettuale (ad esempio il diritto d'autore).
  - La Microsoft è stata in grado di creare un regime di monopolio nel mondo dei sistemi operativi usando solo le leggi del diritto d'autore... si vede che queste sono già abbastanza efficaci.

Protezione del diritto d'autore (1)

- Il diritto d'autore (copyright) tutela uno specifico codice di programma, già concretamente scritto.
  - Allo stesso modo si protegge un romanzo specifico come “Il mastino dei Baskerville” di Arthur Conan Doyle (uno dei romanzi con protagonista Sherlock Holmes)
    - Non si può prendere il romanzo, cambiare alcune righe, e mettere in vendita il libro come proprio.
  - La tutela però non si estende all'idea generale presente nel romanzo.
    - Doyle non può impedire che si scrivano storie di investigazione poliziesca su casi misteriosi ed intricati.
  - E meno male! Altrimenti avremmo dovuto rinunciare al commissario Montalbano.
  - Il diritto d’autore è automatico, non ne va richiesta la concessione.

Protezione del diritto d'autore (2)

- Nel campo del software: non ci possiamo appropriare del software di un altro, ma si può scrivere un programma che ha le stesse funzionalità di un altro.
  - OpenOffice è un programma equivalente a Microsoft Office
- Riassumendo:
  - Si tutela l'autore affinché non venga scippato di un proprio prodotto originale.
  - Si permette la libera competizione in base al merito.
- Crescita globale della ricchezza!

Protezione del brevetto

- È introdotto su una base completamente diversa dal diritto d'autore:
  - Sollecitare gli inventori a rendere pubblico nei passaggi tecnici la loro invenzione, invece di tenerla segreta per proteggere la propria attività.
- Motivazione:
  - La scoperta viene resa subito disponibile.
  - Gli inventori ricevono un privilegio: il diritto esclusivo di sfruttamento commerciale dell'invenzione, per un tempo limitato.
- Limitazioni:
  - L'invenzione deve essere valida industrialmente, non ovvia, deve contenere elementi di novità e funzionare veramente.
  - Il brevetto è concesso da un ufficio apposito che valuta la richiesta.

Tutela dell'inventore

- La tutela dell'inventore è il punto principale della normativa brevettuale.
- Ha ragion d'essere nel momento in cui lo sviluppo di una invenzione richiede investimenti costosi.
  - Anche perché non tutte le invenzioni si riescono a convertire in un prodotto commercialmente interessante.
  - Se non si concedesse il diritto esclusivo di utilizzo, nessuno realizzerebbe l'invenzione.
- Nel campo del software, questa motivazione non ha alcun riscontro:
  - Non esistono costi di progettazione e sviluppo consistenti
  - Non è necessario concedere l'esclusiva sull'utilizzo della invenzione, perché questa viene realizzata in ogni caso.

Tutela del patrimonio culturale

- Il monopolio sancito nel brevetto è limitato nella sua estensione temporale, al fine di non bloccare lo sviluppo tecnologico del sistema.
  - Questo perché scopo primario del sistema dei brevetti è favorire lo sviluppo e far progredire lo stato dell'arte.
- La copertura brevettuale standard è di 20 anni.. questa ha senso nel caso di invenzioni meccaniche..
  - Il software ha un ciclo di vita di pochi anni
- Una copertura brevettuale molto superiore al ciclo di vita dei prodotti che si vuole proteggere blocca lo sviluppo del sistema!

Problemi pratici (1)

- Quelli appena esposti sono problemi di principio alla brevettabilità del software.. ma la realtà è anche peggio
  - I brevetti sul software concessi negli USA tutelano un metodo astratto per risolvere un problema, non uno specifico programma.
    - Spesso anzi, un programma specifico non esiste neanche.
- Più che di brevetti sul software, si dovrebbe parlare allora di brevetti sulle idee.
- Perché questa deviazione?
  - Molto è dovuto a problemi pratici non indifferenti nell'applicare la normativa brevettuale al software.

Problemi pratici (2)

- Come valutare lo stato dell'arte per i sistemi informatici?
  - L'evoluzione dei sistemi informatici è troppo rapida.
  - La maggior parte dei brevetti concessi coprono realizzazioni obsolete.
- Come valutare il passo inventivo?
  - Ogni programma è ottenuto combinando una serie di idee di base, alcune banali, altre molto sofisticate (ad esempio dei metodi di calcolo particolarmente complessi).
  - Si può essere d'accordo che procedure di calcolo molto sofisticate siano protette da brevetto..
  - ... ma in realtà spesso questi metodi sono liberi e pubblicati su riviste scientifiche.

Problemi pratici (3)

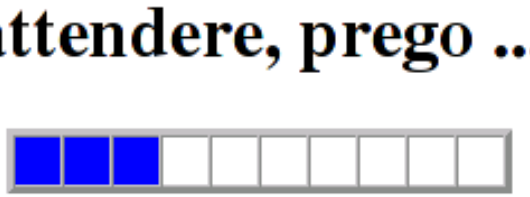
- La situazione paradossale è che quanto più un'idea è banale, tanto più difficile è dimostrare che era nota al momento della richiesta del brevetto.
  - Le idee complesse sono probabilmente pubblicate in qualche rivista scientifica...ma quelle banali nessuno le pubblica.
  - magari sono implementate da decenni, ma purtroppo il software non pubblicato non costituisce una prova che una idea non è nuova (almeno nella giurisdizione statunitense)
- Inoltre, i funzionari degli uffici brevetti raramente sono persone esperte di informatica
  - molto più spesso è costituita da persone che credono di essere esperte perché sanno usare Word.
  - solo da poco tempo in USA i legali che si occupano di brevetti possono essere laureati in informatica.

Problemi pratici (4)

- Infine, i giudici hanno spesso interpretato i brevetti sul software in maniera particolarmente estensiva
  - se una determinata procedura è coperta da brevetto, anche soluzioni alternative allo stesso problema sono state spesso considerate coperte.
  - ma in informatica è facile produrre decine di soluzioni diverse per lo stesso problema!
- Il risultato di tutto ciò, come già detto, è che i brevetti sul software sono diventati più che altro brevetti sulle idee.
  - Due articoli molto interessanti (in inglese):
    - Ordinary Skill in the Art dello scienziato Jeffrey Ulmann
      - http://www-db.stanford.edu/~ullman/pub/focs00.html
    - The Anatomy of a Trivial Patent di Richard M. Stallmann
      - http://www.gnu.org/philosophy/trivial-patent.html

Idee banali

- Alcune idee banali che sono state brevettate
  - Il metodo “compra con un click” brevettato dalla Amazon,
    - copre l'idea di accumulare acquisti online in un carrello della spesa e comprare tutto assieme con un singolo click.
      - http://swpat.ffii.org/patents/effects/1click/index.en.html
    - il contenuto tecnico del brevetto: nullo!
  - La “barra di progresso” che si trova in molte applicazioni
    - copre l'idea di mostrare l'avanzamento di qualche processo in maniera grafica.



    - il contenuto tecnico del brevetto: nullo!

Idee impossibili

- Non mancano i casi di brevetti "impossibili"
  - il brevetto statunitense n° 5,533,051 parla di possibilità di comprimere dei dati completamente casuali, facendo occupare meno spazio in memoria
    - afferma che qualunque file di dati può essere compresso e occupare meno spazio
  - quanto affermato è matematicamente impossibile, come sa chiunque abbia studiato i rudimenti di una branca dell'informatica chiamata "*teoria dell'informazione*"
    - il motivo per cui programmi noti come ZIP, RAR e simili funzionano, è che i dati non sono praticamente mai del tutto casuali!
  - equivale in pratica a un brevetto sul "moto perpetuo"
    - Vedere http://www.teaser.fr/~jlgailly/05533051.html per una analisi del brevetto.

Ancora sull'inventore (1)

- Ma il nostro inventore, alla fine, è veramente tutelato?
- Per realizzare la sua idea sarà costretto ad utilizzare altre decine di idee, banali, ma già brevettate da grandi aziende.
  - Ottenere la licenza di questi brevetti ha spesso costi proibitivi.
  - Se si tratta di un pesce piccolo:
    - Unico modo per arrivare a un prodotto finale è che una grossa azienda rilevi l'invenzione.
    - Ma non volevamo assicurare all'inventore lo sfruttamento delle sue idee?
  - Se si tratta di un pesce grosso:
    - Scambio di licenze.

Ancora sull'inventore (2)

- Ad esempio IBM ha un portafoglio enorme di brevetti software
  - Scrivere software senza violare un brevetto dell'IBM è quasi impossibile.
- Non a caso a difendere il sistema dei brevetti è proprio la BSA.
  - I membri della BSA detengono il 60% dei brevetti software rilasciati dall'ufficio brevetti europeo e che per ora non hanno valore.
- Il problema è che per avere migliaia di brevetti non servono migliaia di idee geniali
  - Bastano milioni di dollari spesi in pratiche legali
- Cosa può mai fare il nostro inventore squattrinato con la sua singola idea brevettata?

Litigation Companies

- Una stortura prodotta dal sistemi dei brevetti.
- Sono compagnie che non producono nulla!
- Ottengono dei brevetti su qualche idea banale e pretendono il pagamento delle licenze da chiunque voglia sviluppare un prodotto che abbia un punto di contatto con i propri brevetti.
- Sono una spina nel fianco anche delle grosse compagnie
  - Non hanno nessun interesse a scambiarsi le licenze con altri, in quanto non producono nulla

Una obiezione più filosofica

- Il software è un mezzo che aumenta la nostra libertà di espressione
  - Scrivere, giocare, suonare, informarsi sono tutte cose che si fanno ormai comunemente con computer e internet.
- Il software è anche una forma di espressione!
  - Milioni di musicisti suonano solo per il piacere di farlo, senza remunerazione, perché la considerano una attività creativa.
  - Allo stesso modo, per molti programmatori scrivere software è una attività creativa che ha molto in comune con l'esperienza artistica.
- Questo aspetto creativo rende ancora più pericolosi i brevetti software! È come voler recintare aree della mente.

Riassumendo

- Il brevetto sul software, che negli USA esiste da 20 anni:
  - Non incentiva l'innovazione
  - Anzi, la rallenta, spostando i fondi destinati a ricerca e sviluppo verso i dipartimenti legali.
- Un tale sistema è improponibile per l'Europa:
  - Onere eccessivi per le nostre imprese, che hanno in gran parte taglia medio-piccola.
  - Dipendenza eccessiva dalle multinazionali d'oltreoceano, che dispongono di enormi portafogli di brevetti.
- Rendono la vita difficile se non impossibile per il software libero.
  - I brevetti proteggono spesso il problema piuttosto che la soluzione.
  - È difficile aggirare delle idee banali.

Riferimenti

- FFII: Software Patents in Europe
  - Associazione contro l'introduzione dei brevetti software in Europa
  - http://swpat.ffii.org/
- Associazione Software Libero
  - http://www.softwarelibero.org/
- Electronic Frontier Foundation
  - associazione non-profit per la difesa della libertà nel nuovo mondo digitale.
  - http://www.eff.com/