

Introduzione al software libero

DANIELE MASINI

- 10 dicembre 2013 -

Partendo dalla definizione di software, il documento ne illustra la sua natura immateriale e pone l'accento sulla differenza tra *codice sorgente* e *codice eseguibile* che, di fatto, sta alla base della nascita del *software libero*. Garantendo agli utenti la massima libertà nell'uso delle applicazioni e, quindi, una maggior affidabilità delle stesse, quest'ultimo ha innescato un circolo virtuoso che ha permesso sia il fiorire di programmi e sistemi operativi robusti ormai molto diffusi e che la nascita di un nuovo modello di business, riportando il controllo dell'hardware al legittimo proprietario.

1 Il software

Il **software** [1] è l'insieme ordinato delle istruzioni, ovvero la sequenza dei passi necessari, per far svolgere un compito ad una macchina (**hardware**). Il software è dunque ciò che spesso viene anche chiamato *programma* o *applicazione*.

Le persone che realizzano il software, i *programmatore* o gli *sviluppatori*, utilizzano appositi *linguaggi di programmazione*¹ [2] che permettono di scrivere le sequenze di istruzioni da impartire alle macchine per mezzo di una struttura e sintassi ben definita, che utilizza un linguaggio vicino a quello naturale con termini derivati dalla lingua inglese. Il programma scritto in tali termini viene detto **codice sorgente** [3] (listato 1): una sequenza di istruzioni, comprensibile dai programmatori ma non dai computer.

```
#include <stdio.h>
#include <math.h>
#include <stdbool.h>

/*****
 * Crivello di Eratostene
 * Trova i numeri primi da 1 a n
 *****/
void main(int argc, char** argv)
{
    int n = atoi(argv[1]);
    int primo[n+1];
    int i, j, ultimo_valore = (int)sqrt((double)
        n)+1;
    bool primo_numero = true;

    /* inizializzazione delle spunte */
    for (i = 0; i <= n; i++)
        primo[i] = 0;

    /* elaborazione numeri primi */
    for (i = 2; i <= ultimo_valore; i++)
        if (primo[i] == 0)
            for (j = 2*i; j <= n; j += i)
                primo[j] = 1;

    /* visualizzazione del risultato */
    printf("I numeri primi tra 1 e %d sono:\n",
        n);
    for (i = 1; i <= n; i++)
```

¹per citarne alcuni il Fortran, il Pascal, il C, il C++, il Java, il Visual Basic, ...

```
    if (primo[i] == 0)
        printf((i==1?"%d":", %d"), i);
    printf("\n");
}
```

Listato 1: Esempio di codice sorgente di un programma (in linguaggio C).

Per mezzo di un particolare programma, il *compilatore*² [4], il codice sorgente (cioè il file³ che lo contiene) viene trasformato nel corrispondente **codice eseguibile**⁴ [6] (fig. 1), ovvero viene generato un apposito file che contiene le istruzioni contenute nel codice sorgente in una forma comprensibile al computer, ma non agli esseri umani. Tale operazione prende appunto il nome di *compilazione* del codice sorgente (fig. 2).

Quindi, per l'esecuzione di un programma, un computer ha bisogno soltanto del relativo *codice eseguibile*. Il codice sorgente dal quale esso è stato generato non è necessario. Inoltre, dal codice eseguibile è praticamente impossibile riuscire a ricostruire il corrispondente codice sorgente.

2 Il controllo del software

Dagli anni '80, quando i computer hanno iniziato a diffondersi in ambito domestico, le aziende che realizzavano software hanno basato il loro business

²un *compilatore* si aspetta di avere in input del *codice sorgente* scritto con un determinato linguaggio di programmazione e produce in output il corrispondente *codice eseguibile* per una specifica piattaforma (hardware e sistema operativo). Per motivi legati alla portabilità del codice, la tendenza odierna è quella di utilizzare un meccanismo più complesso, che si pone ad un livello intermedio tra la *compilazione* e l'*interpretazione* [5] del software, come la compilazione JiT (*Just in Time*), ma il ragionamento qui riportato è applicabile anche in tali casi.

³un qualunque elemento salvato in un computer è un file: ad esempio un documento, una foto, un video, un'applicazione, ... In questo caso si fa riferimento ad un file di testo particolare che, opportunamente elaborato, può generare un file eseguibile.

⁴detto anche *codice macchina* o *formato binario*.

invece, le cose cambiano radicalmente. Se Alice ha un'idea (bene immateriale) può passarla a Bob, dopodiché sia Bob che Alice la condividono e possono usufruirne entrambi: l'idea di Alice non si è cancellata dalla sua mente per il fatto di essere stata fornita a Bob.

La generazione di un bene immateriale, inoltre, non richiede una materia prima, che invece è indispensabile per la produzione di beni materiali. Alice, per generare la propria idea magari si è dovuta sforzare mentalmente ricorrendo alla sua esperienza passata ed al suo intuito, ma il falegname Charlie, oltre alla sua esperienza ed al suo estro, ha dovuto approvvigionarsi del legno per poter realizzare la sedia di Alice. Quindi, una volta avuta l'idea, se Alice vuole fornirla a 10 persone può farlo tranquillamente senza doversi impegnare in modo significativo, cosa che invece non può fare Charlie, poiché se egli vuol fornire una sedia a 10 persone deve realizzare 10 sedie, ovvero la fatica, il tempo ed il materiale necessari per realizzarne una dovranno essere moltiplicati per 10.

Pertanto, se la realizzazione di un bene materiale comporta un lavoro per ogni pezzo, l'atto creativo di un bene immateriale, come il software, è unico ed inoltre per quest'ultimo la realizzazione di copie identiche all'originale è praticamente immediata e a costo nullo.

4 Il software libero

Sempre negli anni '80, mettendo in discussione la politica di fornire soltanto il codice eseguibile del software, Richard Stallman [10], fondatore del progetto *GNU* [11] e della *Free Software Foundation* [12], ha ideato il **software libero** (*free software*)⁶ [13] e la licenza *GNU General Public License*⁷ [14] (in breve *GNU GPL*, che introduce il concetto di © **copyleft**⁸ [15]) per la distribuzione dello stesso. Secondo questa linea di pensiero, chi rilascia un software con la licenza *GNU GPL*, non solo deve fornire all'utente il codice eseguibile, ma deve mettere a sua disposizione anche il relativo codice sorgente. Inoltre la licenza concede all'utente la massima libertà d'uso del software stesso: l'utente può lo far

⁶per contrapposizione, il software del quale viene distribuito soltanto il codice eseguibile viene detto "software proprietario". Successivamente è stato coniato il termine *open source* [17] (*Open Source Initiative*) per indicare il software con caratteristiche analoghe al software libero, con la sostanziale differenza che l'idea del *software libero* è incentrata sulla libertà degli utenti, mentre quella dell'*open source* riguarda unicamente la metodologia di distribuzione del software.

⁷<http://www.gnu.org/licenses/gpl.html>. Oggi esistono anche altre licenze, dette "licenze libere" [16] (forti e deboli), che garantiscono all'utente libertà analoghe a quelle della *GNU GPL*, come le licenze *Creative Commons* [18].

⁸un gioco di parole sul termine *copyright*. Si tratta di un particolare utilizzo del copyright stesso: l'autore di un'opera può concedere ai suoi fruitori il massimo dei diritti, anziché riservarsi, come usualmente avviene.

eeguire da un computer per qualsiasi scopo, ne può studiare il codice sorgente, lo può modificare e distribuire ad altri. L'unico vincolo da rispettare è che il software (anche se modificato) deve essere ridistribuito con la stessa licenza (per garantire che qualcuno non se ne appropri indebitamente). Cioè un software libero, cioè rilasciato con una licenza "libera", rimarrà tale.

Al di là del poter essere eseguito per qualunque scopo, le libertà appena descritte potrebbero apparire inutili per l'utente, in quanto quest'ultimo, nella maggior parte dei casi, non avrà le competenze di uno sviluppatore e quindi potrebbe trovarle superflue. In realtà ne trae comunque un beneficio poiché, al di là del fatto che, volendo, potrebbe rivolgersi ad un programmatore di sua fiducia per far controllare il codice sorgente del software che viene eseguito dal suo computer, facendogli verificare che questo non abbia comportamenti indesiderati, grazie al circolo virtuoso innescato dalla licenza *GNU GPL* (ad esempio) il codice sorgente, passando sotto agli occhi di migliaia di programmatori, è sempre andato migliorando nel tempo sia in termini di stabilità che di funzionalità, poiché coloro che ne hanno apportato delle modifiche hanno in genere mantenuto le eventuali parti buone correggendo eventuali *bug*⁹ ed eliminando i rami infruttuosi (parti inutili o malevole) in esso presenti.

Oggi esiste un'insieme sempre crescente di software appartenente a questa categoria che garantisce agli utenti di fare ciò che dichiara. Ovviamente non ci possono essere garanzie sul fatto che un'applicazione libera non abbia falle e che la sua affidabilità sia tale da poter essere utilizzata anche per sistemi di massima sicurezza, ma, poiché è libera, migliaia di occhi hanno potuto spulciare il suo codice sorgente e questo, statisticamente, fornisce un'ottima garanzia sul fatto che al suo interno non ci siano comportamenti indesiderati rispetto ad un'altra applicazione della quale non è disponibile il relativo codice sorgente.

Di fatto, finora, grazie al software libero, hanno visto la luce molte applicazioni come *Apache* [19] il server web più utilizzato al mondo, *Firefox* [20] uno dei browser web più utilizzati dagli utenti, *LibreOffice* [21] un'ottima suite di strumenti per l'ufficio, *TEX* e *L^AT_EX* [22] gli eccellenti sistemi di composizione di testo¹⁰, *VLC* [23] uno dei più diffusi riproduttori audio/video, *GIMP* [24] un editor di immagini con molte funzionalità, *Notepad++* [25] un editor di testo molto versatile, . . . , ma anche *sistemi operativi*¹¹ [26] come *GNU/Linux*¹² [30], *FreeBSD* [35],

⁹falla presente all'interno del programma che lo fa comportare in maniera non controllata quando si verificano determinate condizioni non previste dallo sviluppatore.

¹⁰utilizzati per scrivere questo documento.

¹¹software di base per il funzionamento di un computer. A titolo esemplificativo, alcuni noti sistemi operativi proprietari sono *Windows* [27], *Mac OS* [28], *OS/400* [29].

¹²un sistema operativo, da molti erroneamente chiamato

*Android*¹³ [36] e molti altri.

E non solo, dopo la GNU GPL la *Free Software Foundation* ha creato la licenza per la documentazione libera *GNU Free Documentation License* [38] (GNU FDL) che ha permesso il diffondersi di progetti interessanti, tra i quali quello più noto è l'utilizzatissima enciclopedia libera *Wikipedia* [39], che conta ad oggi più di un milione di voci in lingua italiana.

5 Il modello di business

Il meccanismo per la generazione del profitto utilizzato dalle aziende che realizzano software proprietario è analogo a quello usato dai produttori di beni materiali: il software viene realizzato e generalmente fornito (solo il codice eseguibile) con una licenza che non concede alcun diritto all'utente, se non quello del suo mero utilizzo (a determinate condizioni). Si ricordi che una volta realizzato un applicativo è possibile crearne, senza alcun costo aggiuntivo, un numero illimitato di copie da destinare alla vendita. Poiché la licenza non permette la redistribuzione del software da parte degli utenti, l'unico modo per ottenere quel determinato software è di rivolgersi a chi lo realizza/distribuisce, che quindi può stabilirne il prezzo.

Con il software libero il discorso cambia. Visto che gli utenti sono potenzialmente in grado di modificare il software e di redistribuirlo, senza renderne conto a chi gliel'ha fornito, il business del tipo precedentemente descritto non può funzionare: al massimo chi idea e realizza un nuovo software può venderne qualche copia, poi, grazie al permesso che la licenza "libera" concede all'utente, lo stesso programma, o addirittura una sua versione con funzionalità aggiuntive, potrà essere reperito da altri, ovvero, estendendo il discorso su larga scala, su internet e dal miglior offerente. Quindi il business si sposta sulla manutenzione del software stesso più che sulla sua vendita. Chi realizza un'applicazione potrà mettere a disposizione dell'utente finale la propria professionalità (e quindi il proprio tempo e la propria esperienza – sia come programmatore in generale ma soprattutto come autore dell'applicativo in questione) per fare formazione sul suo utilizzo, o per apportarvi delle specifiche modifiche, o altro ancora.

A scanso di equivoci è opportuno sottolineare che il software libero non è necessariamente gratuito¹⁴, sebbene la stragrande maggioranza degli applicativi

semplicemente *Linux*, disponibile in varie versioni, dette *distribuzioni* (per citarne alcune: *Debian* [31], *Fedora* [32], *Ubuntu* [33], *OpenSUSE* [34], ...)

¹³nonostante sia basato su *Linux*, *Android* non è un sistema operativo libero [37].

¹⁴il termine anglosassone *free* (usato appunto in *free software*) significa sia *libero* che *gratuito* e quindi non aiuta nella disambiguazione. Questo è uno dei motivi che ha portato alla nascita della dizione *open source* [17].

appartenenti a tale categoria lo sia. L'idea del software libero è quella di andare incontro alla libertà degli utenti nell'uso del software ed è completamente svincolata dai termini economici ad esso relativi. Comunque sia, esistono anche aziende che hanno realizzato un business con il software libero. Un esempio tra tutte è quello di *Red Hat, Inc* [40], un'azienda nata nel 1995 e oggi quotata in borsa con un fatturato che sfiora il miliardo di dollari.

6 La TiVoizzazione

Sempre più apparecchi elettronici di uso domestico, come televisori, console e lettori multimediali, utilizzano software libero per il proprio funzionamento, poiché questa scelta da un punto di vista offre al produttore una migliore garanzia di funzionalità e dell'altro gli consente di avere un sistema a costo zero, poiché quasi tutto il software libero è reperibile gratuitamente.

La maggior parte dei dispositivi domestici che fanno uso di software libero, utilizzano *Linux*¹⁵. Alcuni dei loro produttori, obbligati, dalla licenza, a mettere a disposizione dell'utente il codice sorgente del software che utilizzano, hanno sfruttato una piega della licenza, non rendendo possibile all'utente di verificare se il software in esso presente corrisponde al codice sorgente che loro stessi mettono a disposizione. Essi, infatti, non consentono che il software presente all'interno del loro dispositivo possa essere rimpiazzato dall'utente. Questa strategia è detta **TiVoizzazione** [41] dal nome del primo apparecchio presente sul mercato (*TiVo* [42]) che ha sfruttato questa "lacuna" presente nelle versioni della licenza GNU GPL antecedenti alla 3.

Nel giugno del 2007 è stata redatta la versione 3 della licenza GNU GPL che non permette più il comportamento sopra descritto, ma per scelta di Linus Torvalds [43], l'autore di *Linux*, quest'ultimo viene ancora oggi rilasciato con la versione 2.

7 Conclusioni

L'importanza del software libero è spesso sottovalutata dagli utenti, che, purtroppo, nella maggior parte dei casi ne ignorano l'esistenza, sebbene il termine abbia ormai compiuto quasi 30 anni (la sua prima pubblicazione risale al 1986).

Oltre ad essere la scintilla che consente l'instaurazione di un ciclo virtuoso per il miglioramento dei programmi, il software libero ha la fondamentale caratteristica di consentire all'utente di riappropriarsi del controllo del proprio computer verificando (o sfruttando il fatto che statisticamente qualcuno l'abbia fatto) che le operazioni eseguite quando il software viene lanciato in esecuzione siano quelle

¹⁵il *kernel* (il cuore) del sistema operativo *GNU/Linux*.

opportune. Questo importantissimo vantaggio viene spesso trascurato dagli utenti che affidano magari la protezione del sistema ad altri software, tipo antivirus [44] o altro: questi applicativi, per i quali si rende necessario un aggiornamento continuo, rallentano il funzionamento computer perché devono essere sempre in esecuzione per poter controllare costantemente le applicazioni eseguite. Quasi sempre, inoltre, tali programmi non appartengono alla categoria del software libero e quindi richiedono una fiducia incondizionata. Siamo davvero disposti ad affidare ciecamente ad altri il controllo dei nostri dispositivi elettronici?

Riferimenti

- [1] <http://it.wikipedia.org/wiki/Software>
- [2] http://it.wikipedia.org/wiki/Linguaggio_di_programmazione
- [3] http://it.wikipedia.org/wiki/Codice_sorgente
- [4] <http://it.wikipedia.org/wiki/Compilatore>
- [5] [http://it.wikipedia.org/wiki/Interprete_\(informatica\)](http://it.wikipedia.org/wiki/Interprete_(informatica))
- [6] http://it.wikipedia.org/wiki/File_eseguibile
- [7] http://it.wikipedia.org/wiki/Sistema_numerico_esadecimale
- [8] http://it.wikipedia.org/wiki/Microsoft_Word
- [9] http://it.wikipedia.org/wiki/LibreOffice_Writer
- [10] http://it.wikipedia.org/wiki/Richard_Stallman
- [11] <http://www.gnu.org>
<http://www.gnu.org/gnu/manifesto.it.html>
<http://www.gnu.org/gnu/gnu-history.it.html>
<http://www.gnu.org/gnu/thegnuproject.it.html>
- [12] <http://www.fsf.org>
- [13] http://it.wikipedia.org/wiki/Software_libero
<http://www.gnu.org/philosophy/free-sw.it.html>
<http://www.gnu.org/philosophy/why-free.it.html>
<http://www.gnu.org/philosophy/free-software-for-freedom.it.html>
- [14] <http://www.gnu.org/licenses/gpl.html>
- [15] <http://it.wikipedia.org/wiki/Copyleft>
- [16] it.wikipedia.org/wiki/Licenza_libera
http://it.wikipedia.org/wiki/Comparazione_di_licenze_libere
- [17] <http://opensource.org>
http://it.wikipedia.org/wiki/Open_source
http://it.wikipedia.org/wiki/Differenza_tra_software_libero_e_open_source
- [18] <http://creativecommons.it>
- [19] <http://httpd.apache.org>
- [20] <http://www.mozilla.org/it/firefox>
- [21] <https://it.libreoffice.org>
- [22] <http://www.guit.sssup.it/guit>
- [23] <http://www.videolan.org/vlc>
- [24] <http://www.gimp.org>
- [25] <http://notepad-plus-plus.org>
- [26] http://it.wikipedia.org/wiki/Sistema_operativo
- [27] <http://it.wikipedia.org/wiki/Windows>
- [28] http://it.wikipedia.org/wiki/Mac_OS
- [29] <http://it.wikipedia.org/wiki/OS/400>
- [30] <http://it.wikipedia.org/wiki/Linux>
- [31] <http://it.wikipedia.org/wiki/Debian>
- [32] [http://it.wikipedia.org/wiki/Fedora_\(informatica\)](http://it.wikipedia.org/wiki/Fedora_(informatica))
- [33] <http://it.wikipedia.org/wiki/Ubuntu>
- [34] <http://it.wikipedia.org/wiki/OpenSUSE>
- [35] <http://www.freebsd.org>
- [36] <http://www.android.com>
- [37] <http://www.gnu.org/philosophy/android-and-users-freedom.html>
- [38] <http://www.gnu.org/copyleft/fdl.html>
- [39] <http://www.wikipedia.org>
- [40] <http://it.redhat.com>
- [41] <http://en.wikipedia.org/wiki/Tivoization>
- [42] <http://www.tivo.com>
- [43] http://it.wikipedia.org/wiki/Linus_Torvalds
- [44] <http://it.wikipedia.org/wiki/Antivirus>