Introduzione (veloce) al software libero

Storia di una macchina a vapore e di una stampante sola nel corridoio

Introduzione

Questo testo vuole essere un compendio più articolato, sebbene non certo esaustivo, degli argomenti esposti attraverso la presentazione odierna. Il presente testo può essere stampato e consegnato prima della presentazione.

Per qualsiasi errore o segnalazione vi prego d'inviare una mail all'autore del presente testo: carlo.toniolo@gmail.com. Vi chiediamo di riportare questi riferimenti: manifestazione “XPocalypse, il nuovo giorno” di sabato 5 aprile 2014 presentazione dal titolo Introduzione (veloce) al software libero.

Di cosa parliamo: Software

Software e hardware sono “indivisibili”. Per lo hardware l'identificazione è immediata, fisica, per il software è più eterea, misteriosa. Un po' come si identifica il programma televisivo con il televisore che lo “contiene”. Un televisore posso prenderlo a martellate, altrettanto non posso dire del programma televisivo che attraverso di esso viene trasmesso.

Ecco, possiamo dire che il software è tutto ciò che in un computer *non* posso prendere a martellate. Con una metafora un po' metafisica, posso dire di poter prendere a martellate un libro, la rappresentazione stampata del testo, ma non il testo in sé.

Ciò non di meno, il software è comunque qualcosa di reale e necessario. Come nell'esempio del televisore senza televisione, sarebbe come guardare la nebbia statica, o per i nati nell'era della DTV, lo schermo nero con la scritto “Assenza di segnale”.[1] Come un libro senza rappresentazione testuale, con pagine completamente bianche. Senza software I nostri dispositivi dal design ricercato sarebbero degli inerti quanto inutili agglomerati di plastica e metallo da centinaia di euro.

Il software dice allo hardware cosa fare e come farlo (mica perché è intelligente, altri gli hanno detto esattamente come e cosa fare).

Il software quindi lo possiamo pensare come una ricetta. È una sequenza di operazioni da seguire per poter raggiungere uno scopo. In una parola algoritmo[2]: una sequenza di passi minimi.

Partendo da 1, per arrivare a 5 devo compiere 4 passi (addizionare 1 per 4 volte). Tradotto in pseudo-software:

```
Parti!;
Sei a 1:
Aggiungi 1; Sei a 2! ; Sei a 5?; NO!;
Aggiungi 1; Sei a 3! ; Sei a 5?; NO!;
Aggiungi 1; Sei a 4! ; Sei a 5?; NO!;
Aggiungi 1; Sei a 5! ; Sei a 5?; SÌ!;
Fermati!;
```

Questo è un algoritmo. Inserito all'interno di un dispositivo adeguato diventa software.

Di cosa parliamo: Licenze ed esempi

La licenza, lo sappiamo tutti, è una autorizzazione a compiere determinate azioni. Avete presente la licenza di 007? Lo stesso concetto viene applicato a molti software. Si chiama EULA, End User License Agreement (Accordo di licenza con l'utente finale). È un contratto

1 “Io credo che la televisione vada guardata, ma non accesa” Alessandro Bergonzoni
2 Algoritmo dal matematico astrologo persiano Al-Khwārizmī (780-850 d.C.)

vincolante nei limiti imposti dalla legge. In pratica sottoscrivo un contratto. Accetto cioè che mi vengano imposte delle limitazioni sull'utilizzo del software. Per esempio di usare il programma per i motivi (e solo quelli) dichiarati dalla licenza stessa. Di non controllare il software o di attuare azioni atte alla sua analisi.

Immaginate di comperare un'auto e che il concessionario, una volta acquistata, vi imponga per consegnarvela d'impegnarvi a non usarla se non nei modi specificati dalla licenza, e soprattutto di non analizzarla o farla analizzare da altri nel caso non funzioni come vi aspettate. Già, perché nella licenza d'uso dell'auto voi accettate di usarla “a vostro rischio e pericolo, accettandone altresì il rischio riguardante la qualità, le prestazioni, la precisione e l'impiego soddisfacenti”[3]. Cioè, avete acquistato un'auto, vi dicono come doverla usare e soprattutto che se consuma come un treno non potete sistemarla o portarla a sistemare. Dovete aspettare che vi facciano un aggiornamento (peraltro non dovuto o previsto dalla licenza).

Impedire l'analisi dell'auto serve a ché il meccanico non possa capire come è fatto il motore. Quindi impedirgli di migliorarne le prestazioni o sbellicarsi dalle risate per come è stato costruito. Risultato: macchine scadenti e meccanici tristi.

Di fatto, alcune di queste coercizioni sono invalidate dalla legislazione italiana a protezione del consumatore. Il fatto che questa rinuncia di responsabilità da parte di, per esempio,Apple™ non sia valida in Italia, fa diminuire la sua quota di mercato? Decisamente no. Eppure i dispositivi sono gli stessi venduti anche dove queste limitazioni sono valide. A chi giova questo comportamento? Non al consumatore.

Della licenza non possiamo farne a meno, possiamo però scegliere delle licenze aperte, vedi il capitolo “Licenza (CC BY-SA 4.0)” a pagina 6.

Brevetti – cenni storici: Macchina a vapore

Non è in questa sede che andremo a definire il bene o il male dei brevetti. È mia intenzione fare solo un esempio di come, già nel 1700, la faccenda fosse di primaria importanza.

Nel **1698 Savery** inventa la “macchina a vapore”. Serve principalmente le miniere di carbone sollevando l'acqua dal fondo delle gallerie. La macchina è molto costosa, sia in termini di “carburante” che di gestione. Il problema è l'unico pistone che viene riscaldato e raffreddato per avere il movimento alternato che muove le pompe. Un po' come continuare a riscaldare e raffreddare la stessa pentola a pressione. La macchina di Savery era protetta da brevetti che ne tutelavano l'uso e lo sfruttamento.

Nel **1712 Newcomen**, migliora la macchina di Savery, ma lo deve fare in società con Savery stesso (in quanto proprietario dei brevetti).

Nel **1729 (30 anni dopo)** ci sono circa un centinaio di macchine installate. Qualcuna anche in Europa.

Nel **1764 Watt (a 60 anni dalla prima m. di Savery)** sta riparando nella sua officina una macchina di Savery-Newcomen. Watt la migliora sensibilmente e nel **1775** la brevetta. Per gli 11 anni successivi non inizia **nessuna produzione**. Con manovre molto moderne riesce, attraverso delle amicizie nel governo, a far estendere il brevetto fino al **1800**. Sono praticamente 30 anni in cui Watt (con il socio Boulton che è anche l'aggancio politico) non produce nulla, ma **fa produrre su licenza** e poi assembla le macchine. Ogni sforzo di Watt e del socio è orientato alla protezione legale dei brevetti e a mantenerne le royalties. Lo stesso Watt è vittima di un brevetto (altrui): quello del volano, che trasforma il moto alternato del pistone in continuo. Inventa l'ingranaggio “planetario” per poter avere lo stesso risultato, ma in maniera molto meno efficiente. A fine secolo ci sono circa 750 hp[4] installati in tutta la Gran Bretagna.

I brevetti di Watt **scadono nel 1800**. Da quel momento vi è una vera e propria esplosione nell'utilizzo delle m. a vapore. Dai 750 hp installati a fine secolo (nel corso di 100 anni), si passa ad tasso di crescita di **4000 hp/anno**. Ad aumentare non è solo la potenza installata. Le macchine vengono migliorate e rese ancor più efficienti, complice la scadenza dei brevetti. **Il rendimento aumenta infatti di 5 volte**. Soprattutto si apre la strada alla m. a vapore ad alta

3 Clausola 7.2 dello EULA per l'utilizzo del sistema iOS 7.0 di iPhone e iPad.
4 Significa horse power (cavalli vapore) è l'unità di misura della potenza introdotta da Watt stesso.

pressione, cioè al treno. Nei 30 anni dopo il 1800 si passa da meno di 1000 hp a 120.000 hp installati.[1]

Cosa avrebbe potuto produrre Watt come innovazioni o invenzioni se non fosse stato occupato in questioni legali? Cosa avrebbe significato anticipare di quasi 50 anni l'inizio della rivoluzione industriale?

Torniamo ai giorni nostri. **Nel 1990, al CERN di Ginevra** uno scienziato proponeva un progetto per poter distribuire, tra i ricercatori, alcune pagine informative che fossero collegate tra loro. Il giudizio del direttore è “Fumoso ma interessante” e gli permette di continuare.[2] Quel ricercatore si chiama Timothy John Berners-Lee, ed ha posto le basi del World Wide Web che conosciamo oggi.

25 anni dopo, se Sir Tim Berners-Lee avesse fatto come Savery o Watt, e se fossimo tanto incoscienti da fare un parallelo, staremmo ancora guardando delle pagine come quella di Fig 1.

Fortunatamente la strada intrapresa è stata diversa. Da quella scintilla d'innovazione è scaturito un incendio di proporzioni epocali. Ora possiamo estendere le nostre cattive abitudini a livello globale.

A questo punto mi pongo una domanda. Il brevetto è veramente di stimolo all'innovazione? Secondo me no. Non dico che l'alternativa sia più semplice, dico solo che le ricadute sociali sarebbero nettamente migliori.

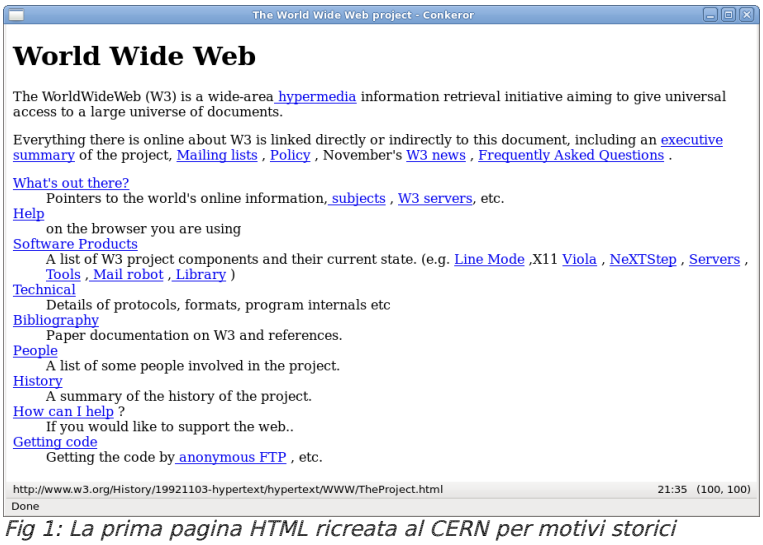

Fig 1: La prima pagina HTML ricreata al CERN per motivi storici

Brevetti – cenni storici: Nascita di GNU/Linux

Saltiamo nel 1980. Sono gli anni in cui Bill Gates (ex CEO di Microsoft™) porta[5] MSDOS (sistema derivato da DOS[6] di IBM) su architetture 8086 grazie al lavoro di un suo collaboratore.

Dobbiamo anche indicare qual'era la consuetudine dell'epoca. Era quella di un'ampia condivisione d'idee e collaborazioni tra varie università e istituti di ricerca. I programmi (nella forma di codice sorgente[7]) venivano scambiati, modificati, adattati alle proprie esigenze.


Fig 2: Richard Stallman

Al MIT (Massachusetts Institute of Technology), nel laboratorio d'intelligenza artificiale dove lavora Richard Stallman (sì, assomiglia ad Agrid), un giorno arriva in dono una nuova (meravigliosa e velocissima) stampante della Xerox. Questa stampante ha però un piccolo problema. Si inceppa. Ora, il punto è che nel corridoio la stampante è sola: deve servire infatti un intero dipartimento, non un solo ufficio. Dopo aver lanciato una stampa, il ricercatore si sarebbe accorto dell'inceppamento solo quando, davanti alla stampante, avrebbe visto altri colleghi imbestialiti strappare carta inceppata dalle viscere della (meravigliosa e velocissima) stampante. Il punto (pragmatico) è “basta saperlo”. Avendo qualcosa che mi avvisi dell'inceppamento, non manderei in stampa nulla ma andrei a sbloccarla. “Semplice” pensano, siamo il dipartimento d'Intelligenza artificiale, facciamo in modo che la stampante avvisi il server[8] del problema.

5 Traduce da un'architettura hardware a un'altra
6 Disk Operating System
7 Indica una forma intermedia tra linguaggio umano e semplice sequenza di 0 e 1
8 Un PDP-11 che loro stessi programmavano


Fig 3: Linus Torvalds

Peccato che ci sia un però. La Xerox non forniva con la stampante il software di funzionamento della stessa (cosa tra l'altro assolutamente inusuale per l'epoca). Anzi, si scopre che una università vicina aveva il software, ma che aveva giurato di non consegnarlo a nessuno. Un ostacolo eticamente inaccettabile. Stallman avrà a quel punto quella che si può definire una Epifania. Capisce la portata globale del problema e comincia a elaborare quello che porterà alla creazione del movimento per il software libero. Creerà anche tutta una serie di software che per diritto di nascita saranno liberi. Questo insieme di software verrà chiamato **GNU** (**G**nu is **N**ot **U**nix) con una definizione "ricorsiva" che richiama il sistema operativo proprietario Unix.

Dopo il WWW di Berners-Lee e la "rivelazione" di Stalmann, nel 1991 uno studente d'informatica pubblicherà su una bacheca elettronica[9] un sistema operativo da lui creato per motivi didattici.[10] Subito stimola l'interesse di tanti sviluppatori che provano, correggono e collaborano al miglioramento del sistema operativo. Lo studente in questione si chiama Linus Benedict Torvalds.

Il connubio tra GNU e Linux è praticamente automatico, infatti sono un po' l'uno il completamento dell'altro. Nasce il sistema operativo GNU/Linux, che diventerà (ed è tutt'ora) parte fondamentale dello sviluppo e l'avanzamento della Internet mondiale.

Free = Libero

Free as in Freedom. Libero perché libera chi lo usa dai lacci e lacciuoli che spesso legano un utente alla licenza d'uso del software proprietario. Di base, lo statuto stilato dalla Free Software Foundation "impone" il rispetto di alcuni principi base. Vengono chiamate le 4 libertà fondamentali del software libero. Le riportiamo come sono nella versione italiana.[3]

Libertà 0: Libertà di eseguire il programma, per qualsiasi scopo.

Libertà 1: Libertà di studiare come funziona il programma e di modificarlo in modo da adattarlo alle proprie necessità. L'accesso al codice sorgente(nota 7 a pagina 3) ne è un prerequisito.

Libertà 2: Libertà di ridistribuire copie in modo da aiutare il prossimo.

Libertà 3: Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti da voi apportati (e le vostre versioni modificate in genere), in modo tale che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente ne è un prerequisito.

Questo approccio permette che anche i lavori derivati da software libero rimangano liberi. Come vedete non ci sono riferimenti all'aspetto monetario. Il software libero infatti non impedisce di chiedere denaro in cambio del software o di servizi associati a questo. Vedremo infatti nel prossimo capitolo che non è detto sia gratis.

Free = Gratis ... Anche No!

Lo so, la prima forza della natura è il risparmio. Sfatiamo però un mito: Linux non vien via "a gratis". Lo sviluppo costa tempo, quindi denaro. Il tempo-uomo usato per lo sviluppo del kernel Linux viene però contabilizzato. In uno studio del 2011[4] si è calcolato che svilupparlo da zero costerebbe circa 3 miliardi di dollari.

Non ha nemmeno senso dire che è sviluppato bene (o male) perché viene sviluppato solo da volontari nei ritagli di tempo. La natura delle contribuzioni è articolata e viene annualmente analizzata dalla *The Linux Foundation* tramite la pubblicazione "Linux Kernel Development".[5] Da questo rapporto si evince che l'80% dei contributi sono dovuti a lavoro remunerato. È lavoro di programmatori e analisti pagati da società terze per sviluppare, correggere o implementare

9 25 agosto 1991 "Hello everybody out there [...] i'm doing a (free) operating system, just a hobby, won't be big and professional [...]"
10 Scatenando un'accesa discussione con il (tuttora luminare) dell'informatica Andrew S. Tanenbaum

nuove feature nel kernel. Ci sono due società che spiccano tra le tante. Red hat contribuisce per un 11% dello sviluppo e ha un fatturato di 1,3 miliardi di $.[6] Intel contribuisce con l'8% dello sviluppo e ha un fatturato di 53 miliardi di $[7]. Pochi anni fa era IBM il maggior contribuente. Come vedete non è scevro da investimenti e ancor meno è da considerarsi un sistema operativo da “smanettoni”.

Closed Source

Bene, spero di essere riuscito a spiegarvi cosa sia il software libero. Vediamo per completezza anche la controparte: il software chiuso.

Rimaniamo alla metafora dell'auto. Il software “closed source” è un'auto della quale non conosco (e non posso conoscerne) il funzionamento. Significa che ho degli impedimenti legali (la licenza) e tecnici (sigilli al cofano) che mi precludono l'accesso alle parti fondamentali. Alcuni esempi sono Skype, il Blackberry o il “Clipper Chip”. Andiamo a vederli brevemente.

Skype attua delle contromisure per impedire e ostacolare la propria analisi. L'auto modello Skype ha intanto il cofano sigillato. Riuscendo comunque ad aprire il cofano vedrei molti più tubi e fili del necessario, questo serve a confondere il meccanico. Se anche il meccanico provasse a metterci le mani, nel momento in cui il motore si accorgesse di essere osservato (variazione dei giri motore perché il meccanico prova a limitare l'afflusso di benzina dai tubi per capire quale sia vero e quale sia lì per confondere) questi si ferma. Parte del funzionamento di Skype è per impedire di capire il funzionamento di Skype.

Qualche tempo fa è stato sospeso l'uso del servizio Blackberry in Arabia Saudita. Il motivo è presto detto: permetteva l'uso non controllato (i server della RIM si trovano in Canada) di strumenti di comunicazione come posta elettronica e messaggi in maniera irrintracciabile dalle autorità. L'azienda RIM ha installato dei server fisicamente posizionati in Arabia Saudita permettendo alle autorità di accedervi per motivi di sicurezza. Il servizio Blackberry è stato riattivato.[8]

Il “Clipper Chip” doveva essere un chip inserito nei telefoni cellulari che doveva permettere alle autorità americane di “decodificare” le telefonate on demand tra due interlocutori. Le proteste di vari gruppi di attivisti hanno impedito l'adozione di questo sistema.[9]

Non è scopo di questo documento giudicare l'opportunità o meno di tali soluzioni, o la loro applicabilità etica. Quello che si vuole sottolineare è che in tutti questi sistemi viene usato un principio di sicurezza chiamato “security through obscurity” (trad: sicuro perché oscuro). Sono scatole nere “sicure” perché non si sa come funzionino. Il problema è quando qualcuno di particolarmente bravo o addentro al sistema espone alcune parti fondamentali. A quel punto il re è nudo. Non è raro che il risultato sia l'esposizione incontrollata degli utenti, o di informazioni ad essi associate.

GNU/Linux è [comodo && spaziale && buono]

Gnu/Linux è comodo (anche quando ha problemi). Sempre ordinato, aggiornato e in costante miglioramento. Un solo punto da dove installare più di quanto software potreste mai voler usare nella vostra vita. Questo perché spesso, se qualcuno si costruisce un software per controllare il pesce rosso tramite la propria webcam, anche chi vuole controllare i bambini nella cameretta potrebbe trovarlo utile, condividendolo lo si rende disponibile a chiunque. Con il “moderno” sistema disoperativo che ha per logo una bandierina colorata dell'azienda di Tusaichi dovreste girare per ore in internet a scaricare programmi di questo o quell'altro sito. Con Gnu/Linux avete un posto solo, un comodo elenco ordinato per categoria. Dai giochi al calcolo vettoriale, dalla progettazione d'interni alla pianificazione di un attacco termonucleare totale, a portata di 3 click (forse 4 click). C'è da dire che esistono programmi Free Software anche per il “moderno” sistema disoperativo che ha per logo una bandierina colorata dell'azienda di Tusaichi. Spesso sono infatti quelli più facilmente adattabili ad altre architetture.

GNU/Linux e sicuro. Dice il detto: “un computer sicuro è un computer spento”[10], ma Gnu/Linux è un sistema operativo molto sicuro anche da acceso. Soprattutto per come viene sviluppato (e corretto). Free Software significa che il codice sorgente è apertamente disponibile. Il motore è aperto, molti meccanici possono guardarci dentro a piacimento. Molti occhi

controllano e correggono i problemi a volte addirittura in anticipo rispetto alla segnalazione. I problemi ci sono, come in qualsiasi sistema, ma sono chiaramente identificabili e sopratutto la segnalazione e la correzione non sono dettate da logiche economiche.

Fig 4: REVEAL

GNU/Linux è spaziale perché è proprio spaziale. Viene usato in un satellite orbitante di simulazione dell' ESA (Agenzia Spaziale Europea): il progetto OPS-SAT[11]. Viene anche usato in aerei di monitoraggio scientifico terrestre della NASA. Con il nome di progetto REVEAL[12]. Oggi potrete usarlo nel vostro computer, con la stessa affidabilità.

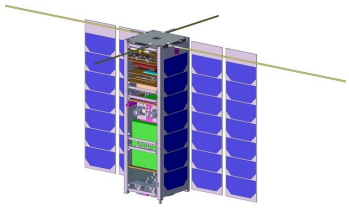

Fig 5: OPS-SAT

Fig 6: Logo OLPC

GNU/Linux è buono. Grazie all'utilizzo del software libero è stato possibile creare un computer con un costo inferiore ai 100 dollari. Si tratta del progetto “One Laptop per Child”. Un computer completo (tastiera, mouse, schermo e audio) con la possibilità di essere anche ricaricato attraverso una dinamo. L'obiettivo è di ridurre il divario digitale nei paesi in via di sviluppo.

Per un esempio di “Pensiero globale e azione locale” invece è possibile visitare l'aula computer del Centro Giovanile. Aula creata grazie al lavoro e all'impegno dei volontari del GrappaLUG. È stato usato unicamente Free Software e sono stati recuperati PC oramai non utilizzabili con il “moderno” sistema disoperativo che ha per logo una bandierina colorata dell'azienda di Tusaichi. Questo ha tra l'altro contribuito a migliorare l'ambiente riutilizzando risorse ancora utili.

Conclusioni

Spero che questo testo vi sia stato utile almeno la metà di quanto sia stato utile a me scriverlo. Spero abbiate potuto apprezzare un modo diverso di vedere la società e di esserne partecipi.

“Se tu hai una mela, e io ho una mela, e ce le scambiamo, allora tu ed io abbiamo sempre una mela per uno. Ma se tu hai un'idea, ed io ho un'idea, e ce le scambiamo, allora abbiamo entrambi due idee.”[11]

Bibliografia

1: Michele Boldrin, David K. Levine, "Abolire la proprietà intellettuale" , 2012 Laterza
2: CERN , WWW The project http://info.cern.ch/hypertext/WWW/TheProject.html
3: Free Software Foundation , Cos'è il Software Libero?
4: Blog , Linux Cost http://linuxcost.blogspot.it/2011/03/cost-of-linux.html
5: The Linux Foundation, "Linux Kernel Development" , 2013 The Linux Foundation
6: Red Hat , Red Hat Annual Report http://investors.redhat.com/annuals.cfm
7: Intel , Intel Annual Report http://www.intc.com/annuals.cfm
8: Il Post , Perchè gli Emirati Arabi hanno paura del blackberry http://www.ilpost.it/2010/08/01/perche-gli-emirati-arabi-hanno-paura-del-blackberry/
9: New York Times , Battle of the clipper chip http://www.nytimes.com/1994/06/12/magazine/battle-of-the-clipper-chip.html
10: Kevin Mitnick, "L'arte dell'inganno" , Lafeltrinelli
11: ESA , OPS-SAT Operations http://www.esa.int/Our_Activities/Operations/OPS-SAT
12: NASA , REVEAL Project http://www.nasa.gov/centers/dryden/research/ESCD/OTH/Tools_Technologies/reveal.html

Licenza (CC BY-SA 4.0)

11 George Bernard Shaw (1856 – 1950), drammaturgo, narratore e saggista irlandese.