

Il Progetto GNU

Era il 1985 quando Richard Stallman fondó la *Free Software Foundation* con lo scopo di diffondere la filosofia del Software Libero. Queste idee filosofiche si tradussero in pratica nel 1989 con la redazione della licenza d'uso *GPL* (General Public License) e la nascita del progetto *GNU*.

GNU significa “GNU’s not Unix”; un acronimo ricorsivo che vuole rendere evidente il distacco del progetto da quello che era diventato il sistema *UNIX* ed il software in generale: un sistema proprietario coperto da licenze molto restrittive che ne impedivano, come oggi, la conoscenza all'esterno delle aziende che lo producevano.

Il sistema *GNU* ereditava da *UNIX* due importanti caratteristiche: il sistema di sviluppo aperto che aveva caratterizzato i primi anni di sviluppo, e la struttura multiutente e multiprocesso che aveva reso *UNIX* così apprezzato. Conviene, prima di proseguire, fare un passo indietro e tornare alla nascita di *UNIX*

Breve Storia di Unix

Nel 1965 il Bell Laboratory con la collaborazione del MIT e della GE lavora alla realizzazione di un sistema operativo, *Multics*, il quale vuole offrire come principali caratteristiche, capacità multi-utente, multi-processo ed un file system multi livello.

Nel 1969 AT&T abbandona il progetto, ma alcuni programmatori dei Bell Lab che avevano lavorato al progetto *Multics*, progettano ed implementano su un PDP-7 la prima versione di *UNIX*.

Nel 1973 *UNIX* viene riscritto in “C”, un linguaggio appena inventato da Dennis Ritchie, uno degli sviluppatori di *UNIX*. Questo facilitó il successivo “porting” del sistema su altre piattaforme.

Verso la fine degli anni '70 AT&T forní a prezzo molto basso le licenze di *UNIX*, che includevano i codici sorgente, a svariati college ed universitá. *UNIX* prese piede lentamente all'esterno delle istituzioni accademiche, e divenne popolare anche in ambiente commerciale.

É in questo contesto che prese l'avvio nel 1978 presso l'universitá di Berkeley, grazie anche al sostegno finanziario di ARPA, lo sviluppo dello *Unix BSD* (Berkeley Software Distribution). Questa distribuzione era rilasciata sotto licenza BSD, che rimane la progenitrice della filosofia del software libero. Nel 1992 si decise di ripulire lo *Unix BSD* dal codice proprietario e distribuirlo sotto il nome di *386BSD*. A causa di contese giudiziarie questa distribuzione fu poi eliminata, ma vennero poi sviluppati altri sistemi BSD liberi: *NetBSD*, *FreeBSD* e *OpenBSD*.

In ambito commerciale Microsoft sviluppa *Xenix*, poi acquisito da AT&T che lo incorporerá, assieme allo *UNIX* originale, nel suo *System V* rilasciato nel 1988. Nel 1993 i diritti di Unix vengono acquistati da Novell e ceduti poi a SCO nel 1995.

Nel frattempo compare sulla scena un sistema tipo-unix sviluppato per hobby da uno studente finlandese di nome Linus Torvald: Linux.

GNU/Linux

Nel giro di pochi anni dalla sua nascita la FSF aveva realizzato una grossa mole di programmi e utilit , ma era stato continuamente rimandato lo sviluppo del “kernel”, (nucleo) ossia l’insieme dei programmi di base che consentono la gestione delle risorse fondamentali degli elaboratori, in attesa della promessa liberalizzazione di un microkernel molto avanzato che era in fase di sviluppo.

Nel 1990 uno studente ventenne dell’universit  di Helsinki, Linus Torvald, decide di acquistare un calcolatore per sviluppare il suo hobby preferito, la programmazione. I nuovi PC che montano un processore Intel 386 sembrano un ottimo compromesso tra costi e prestazioni, ma il loro sistema operativo, il *DOS* di Microsoft, non permette di realizzare software ad alto livello, ed il costo di una licenza di *UNIX*   proibitivo per le sue tasche.   cos  che decide di scriversi da solo il sistema operativo, partendo da Minix, un sistema distribuito con il sorgente per scopi didattici dal Prof. Andrew S. Tanenbaum.

Nel giro di pochi mesi   pronto il nucleo del sistema, versione 0.01, e Linus decide di diffonderlo su Internet, senza chiedere altra contropartita che la collaborazione a migliorarlo ed espanderlo. Al suo invito rispondono subito centinaia di programmatori, che poi diventeranno migliaia e porteranno nel giro di pochi anni *Linux* a diventare una entit  capace di competere sul piano della solidit  ed affidabilit  con i migliori sistemi sul mercato.

Una grossa parte della fortuna di Linux   dovuta all’accettazione da parte del suo creatore della logica del copyleft e l’inserimento del suo sistema nella grande cornice di *GNU*; nasce cos  il sistema operativo *GNU/Linux*, pienamente compatibile con lo *UNIX* proprietario ma completamente libero, distribuito sotto licenza GPL.

La Cattedrale e il Bazaar

Un famoso documento scritto da Eric S. Raymond, che é possibile trovare in rete anche in italiano, fornisce una metafora molto efficace per descrivere la contrapposizione tra modello di sviluppo aperto e chiuso.

La metafora paragona il modello di sviluppo centralizzato utilizzato dalla maggior parte delle software house, e non solo, alla costruzione di una cattedrale, dove geni dell'arte e delle scienze lavorano in un perfetto isolamento per portare a termine l'opera grandiosa, che verrà disvelata solo a lavori ultimati. A questo si contrappone il modello di sviluppo aperto, utilizzato da sempre nella comunità scientifica, molto più simile ad un confuso bazaar, dove sembra non esista alcuna guida o logica, ed in cui ognuno é libero di apportare il proprio contributo nei tempi e modi a lui più confacenti.

Come l'autore ha sottolineato, prima di Linux anche nell'ambiente del software libero il modello del bazaar era considerato valido per progetti di portata limitata. Era convinzione comune che progetti di maggiore dimensione richiedessero comunque una gestione centralizzata. Il merito di Linus Torvald stato principalmente di aver reso evidente che lo stile bazaar é efficace per ogni tipo di progetto. Per farlo funzionare Linus adotta una serie di accorgimenti che escono dagli schemi usuali nell'ambiente:

- Il software é condiviso e accessibile a tutti per copie, modifiche e sviluppo
- Ciascuno é libero di dedicarsi al progetto che più gli interessa
- Gli aggiornamenti sono distribuiti spesso perch tutti possano toccare con mano i progressi
- Viene sempre ricordata la partecipazione di tutti gli sviluppatori che hanno contribuito al progetto
- Gli utenti vengono trattati al pari di co-sviluppatori, ed il loro contributo é considerato fondamentale per eliminare i bug e migliorare il prodotto

E quest'ultimo forse é l'aspetto piú innovativo del modello adottato; nella concezione a cattedrale servono mesi di scrutinio da parte di svariate persone per sentirsi sicuri di aver risolto tutti i problemi. I tempi per l'emissione di nuovi aggiornamenti si allungano, e la delusione é forte quando essi si rivelano imperfetti.

Nel modello bazaar gli aggiornamenti si succedono rapidamente e vengono sottoposti all'attenzione di un numero elevato di co-sviluppatori che identificano ed eliminano con rapiditá ogni problema o baco del software. Tutto questo viene enunciato nel famoso "Teorema di Linus": "Date un numero sufficiente di occhi e tutti i bachi vengono a galla".

Un'altra arma vincente del modello seguito da Linus é la gratificazione morale che offre ai partecipanti ad un progetto, molte volte piú importante di quella materiale. I programmatori liberi lavorano infatti principalmente per la soddisfazione personale e la stima degli amici, senza fretta avendo la qualità come obiettivo centrale. I prgrammatori di aziende che vendono software su licenza devono produrre una nuova versione ogni due anni, per non uscire dal mercato e produrre nuovo fatturato. Ma la complessitá dei nuovi prodotti cresce molto rapidamente e soltanto sacrificando la qualità é possibile superare le sfide della complessitá.

Per queste ed altre ragioni oggi il modello di sviluppo indicato dalla FSF e da Linus Torvald é apprezzato oggi in ambienti sia accademici che industriali. E c'è chi pensa che possa essere applicato anche in altri ambienti: un esempio conosciuto é la produzione di musica e libri liberamente consultabili e scaricabili da Internet, ma pochi forse sanno che esiste anche una Open Cola, bevanda la cui formula é accessibile ed a cui possono essere proposte modifiche.

Free Software e Open Source

Come le idee il software é immateriale e puó essere riprodotto e trasmesso facilmente. E come le idee, ogni giorno di piú, il software produce effetti etici, economici, politici e, in generale, culturali.

Come abbiamo visto prima, negli anni Ottanta Richard M. Stallman formalizzó l'idea di software libero, e lo fece nella forma di quattro principi di libertá:

- Libertá di eseguire il programma per qualunque scopo, senza vincoli sul suo utilizzo
- Libertá di studiare il funzionamento del programma, e di adattarlo alle proprie esigenze
- Libertá di redistribuire copie del programma
- Libertá di migliorare il programma e di distribuirne i miglioramenti

Il software distribuito con una licenza che rispetti questi principi é detto "software libero" (free software). A causa della forte carica ideologica questi principi non furono subito bene accolti nel mondo degli affari.

Nel 1998 Bruce Perens, Eric Raymond e altre personalitá nel campo del software libero, scrissero la *Open Source Definition*, il documento fondamentale del movimento Open Source. Tale documento fu scritto evitando accuratamente ogni riferimento a considerazioni politiche o di principio.

Il movimento Open Source fu un successo e contribuí a sdoganare il concetto del software libero in campo aziendale, dove prima era guardato con sospetto. Ed il termine Open Source é diventato il piú utilizzato, perché veicolato da grosse aziende che ne hanno tratto benefici sia in termini di immagine che economici.

La voluta neutralitá del movimento Open Source verso gli aspetti etici e politici del software libero é la caratteristica principale che lo distingue dalla filosofia del software libero, che al contrario pone l'accento sulle motivazioni ideali. Free Software e Open Source sono oggi due facce della stessa medaglia.

Come si é espresso qualcuno “L’Open Source é una metodologia di sviluppo, il Free Software é un movimento di carattere sociale”. Gli obiettivi sono gli stessi cosí come i mezzi per realizzarli; di fatto non esiste una contrapposizione tra le persone che ne fanno parte, e molti progetti sono condivisi dalle due comunitá.

Software Libero e Mercato

La rilevanza economica del software libero é oggi ancora relativamente ridotta, ma é in fortissima crescita, e tutto lascia supporre che questa tendenza continuerá nel prossimo futuro, in quanto il software libero si é dimostrato superiore anche come fonte di sviluppo economico.

Commercialmente il software libero é caratterizzato dal fatto di essere distribuito con licenza copyleft, tipo GPL. I grossi produttori di software distribuito con licenza proprietaria cercano in tutti i modi di demonizzare questo tipo di software con motivazioni che vertono tutte verso una presunta incompatibilitá tra mercato e software libero. In realtá sembra che sia incompatibile solo con il loro modello monopolistico di mercato, dato che esistono molti motivi per pensare che il software libero dia al mercato nuove importanti opportunitá.

La migliore caratteristica di un sistema informatico é la sua flessibilitá, ma oggi é considerato del tutto normale che sia l'utente a doversi dimostrare flessibile verso interfacce rigide che non permettono alcuna divergenza dal comportamento atteso. Educare l'utenza di base all'estensibilitá del software aprirebbe nuovi mercati ai consulenti autonomi, creando nuove opportunitá di lavoro.

Mentre il modello di software proprietario accentra tutte le conoscenze nelle mani di pochi, e l'assistenza puó essere fornita solo da un numero limitato di consulenti autorizzati, il numero di consulenti che supportano soluzioni di software libero é virtualmente illimitato e l'offerta puó velocemente adattarsi alla domanda.

Naturalmente i consulenti privati non riescono a coprire tutti i bisogni degli utenti. Molte attivitá non possono essere gestite da singoli individui, ma RedHat e SuSE stanno dimostrando che creare e tenere aggiornata una distribuzione puó costituire una fonte di reddito. Alle grandi imprese che utilizzano mezzi informatici come supporto alla produzione non basteranno né il consulente esterno né il tecnico interno. Esse vorranno poter contare su una struttura in grado di garantire il funzionamento continuo dei loro strumenti tecnologici. Le grandi societá di assistenza saranno in grado di

gestire in modo efficace tali esigenze.

Nelle pubbliche amministrazioni di molti paesi, inclusa l'Italia, prevale l'acquisto di programmi da società commerciali già troppo ricche piuttosto che l'impiego e lo sviluppo di risorse umane. Questo, oltre a limitare lo sviluppo di una conoscenza diffusa dello sviluppo del software, crea anche pericolose dipendenze di stati interi verso queste società che, senza voler ricordare recenti eventi tipo Echelon, possono sfociare nel disastro totale qualora la società scompaia o semplicemente decida di impiegare le sue risorse altrove. Di questo ben se ne sono accorti paesi come Francia, Germania, Messico, Perù, Cina, India ed altri che hanno già annunciato progetti per l'utilizzo di software libero a vari livelli.

Non ultimo vi è un discorso che possiamo definire di etica liberale. L'economia di mercato ha come missione dichiarata l'utilizzo della tecnologia per eliminare la scarsità. Oggi noi abbiamo i mezzi per duplicare ogni tipo di informazione che può essere rappresentata su supporto digitale e replicarla in tutto il mondo a miliardi di persone a costi veramente bassi. Invece il mantenimento e lo sviluppo di sempre nuove leggi che proteggono il copyright a favore di grosse società, contribuiscono a perpetuare la scarsità. In questo modo molte organizzazioni o paesi con buone risorse intellettuali ma scarsi mezzi rimangono tagliate fuori dallo sviluppo per favorire una piccola porzione di società.

Conclusione

Per i motivi storici, filosofici e pratici fin qui descritti io ritengo il software libero superiore rispetto al software sviluppato in ambiente proprietario, e penso che l'adozione di questo standard porti ad un significativo miglioramento della qualità della vita di tutti. Per questo motivo ho scelto di dedicare parte del mio tempo e risorse alla diffusione del software libero, e mi piace pensare che questo sia anche il motivo che ha spinto qui oggi tutti noi per questa giornata dedicata a Linux. Spero che voi possiate approfittare dell'occasione che vi viene offerta.

Indice delle fonti

Ringrazio gli autori degli articoli sotto elencati, da cui ho tratto ampi spunti per la redazione di questa relazione.

- *Appunti di informatica libera* di Daniele Giacomini
<http://ildp.pluto.linux.it/AppuntiLinux/>
- *OpenSource e Software Libero* di Angelo Raffaele Meo
<http://www.osservatoriotecnologico.net/software/opensource.htm>
- *Introduzione a Unix* di Frank G. Fiamingo
<http://www.freebsd.org/doc/it.IT.ISO8859-15/books/unix-introduction/>
- *La storia di Unix* di Bruno Badiali
http://digilander.libero.it/bunix/it_hist.html
- *Introduzione a UNIX ed alla sua evoluzione* di Fabio Criscuoli
<http://www.univ.trieste.it/niredc/doc/oldunix/>
- *La Cattedrale e il Bazaar* di Eric S. Raymond
<http://www.apogeeonline.com/openpress/doc/cathedral.html>
- *Perché "Software Libero" è meglio di "Open Source"* della FSF
<http://www.gnu.org/philosophy/free-software-for-freedom.it.html>
- *Economia del software libero: vantaggi e pericoli* di Robert J. Chassell
<http://www.softwarelibero.it/altri/economia-sl.shtml>
- *Software libero e sostenibilità commerciale* di Alessandro Rubini
<http://www.gnu.org/philosophy/software-libre-commercial-viability.it.html>
- *Cosa c'è che non va nella protezione dalla copia* di John Gilmore
<http://www.toad.com/gnu/whatswrong.it.html>