

CAPITOLO 1

SOFTWARE LIBERO E LINUX

Pietro Buffa

1.1 SOFTWARE: Concetti elementari

In maniera simile a quanto avviene per le opere letterarie, il software è un prodotto generalmente sottoposto alle leggi sul diritto di autore. Il contratto che regola l'uso del software è la **licenza** ed è sempre importante conoscere i termini di questo accordo per i software con cui si intende lavorare.

Il software ha un proprietario, che è tale in quanto detiene i diritti d'autore. Questo proprietario può essere l'autore originale, oppure un altro detentore che ne ha acquisito i diritti in base a un contratto. Il detentore dei diritti di autore è colui che possiede il **copyright**.

Il pagamento per l'acquisto di un software, non si riferisce all'acquisizione dei diritti di autore, bensì all'acquisizione di determinate facoltà, stabilite dalla legge ed indicate nella licenza.

La natura del software è tale per cui questo è composto da due parti fondamentali:

Codice Sorgente (Intelligibile)

Codice Eseguitabile (Adatto all'esecuzione computazionale e non intelligibile)

Dal momento che per funzionare è sufficiente il codice eseguibile, le leggi dei vari paesi che tutelano il diritto di autore per il software, tendono a consentire la distribuzione del solo codice eseguibile, lasciando che chi detiene i diritti di autore possa mantenere nascosto il codice sorgente. Inoltre, le leggi di questi paesi tendono a considerare illecita la **decompilazione**, ovvero lo studio del codice eseguibile volto a scoprirne il funzionamento.

1.2 IL SOFTWARE LIBERO

Software libero non vuol dire necessariamente software gratuito, questo termine infatti, non ha niente a che vedere con il prezzo del software. Un software viene generalmente ritenuto libero se si verificano le seguenti condizioni:

- L'utente ha la libertà di installare ed utilizzare il programma per qualsiasi scopo.
- L'utente ha la libertà di distribuire copie del programma, gratuitamente o dietro compenso.
- L'utente, avendo le necessarie competenze, ha la libertà di modificare il programma secondo i propri bisogni (questo rende necessario l'accesso al codice sorgente del programma).
- L'utente ha la libertà di distribuire versioni modificate del programma, così che la comunità possa fruire dei miglioramenti apportati.

• Software di dominio pubblico

Il software di dominio pubblico è quel software senza copyright. Di per sé, questo tipo di software è libero, solo che, non avendo un copyright, non ha nemmeno una licenza e chiunque può farne ciò che vuole, anche appropriarsi dei diritti. Questo può avere come conseguenza che alcune copie, o varianti di questo software, possano diventare non più libere.

• Software libero protetto da copyleft

La parola **copyleft** vuole rappresentare il copyright di chi, mentre difende il proprio diritto di autore, vuole difendere la libertà della sua opera, imponendo che questa e le sue derivazioni restino libere. In pratica, una licenza appartenente alla categoria copyleft impedisce che chi ridistribuisce il software, originale o modificato che sia, possa aggiungere delle restrizioni ulteriori. Il classico esempio di licenza di questo tipo è la licenza pubblica **GPL** (*GNU General Public License*).

1.3 IL SOFTWARE NON LIBERO

Il software non è libero tutte le volte che non sono soddisfatti i requisiti necessari per poterlo essere. È bene ricordare che il prezzo non è un fattore che limita la libertà, mentre altri dettagli sono più importanti. Anche in questo caso, vale la pena di elencare alcune definizioni che in generale riguardano il software non libero:

• Software semi-libero

Il software semi-libero permette agli individui di usarlo, copiarlo, modificarlo e distribuirlo, anche modificato, per qualunque scopo, escluso quello di trarne profitto. In altri termini, si potrebbe dire che si tratta di software libero a cui è stata aggiunta la limitazione per la quale questo non può essere usato e distribuito per trarne profitto. Questo dettaglio è molto importante e non va trascurato.

• Software freeware

Il termine freeware non è abbinato a una definizione precisa, ma viene inteso generalmente come software gratuito, del quale però, non viene reso pubblico il codice sorgente, esso può essere usato e copiato senza poterlo quindi modificare. In questo senso, il prefisso «free» serve solo a evidenziare la gratuità della cosa, ma non la libertà che invece richiede molto di più.

• Software proprietario

Di solito, per avere a che fare con questo software è necessario ottenere un permesso speciale, che spesso si limita a concedere l'uso su un elaboratore, o su un gruppo ben determinato di elaboratori. Si parla più generalmente di **software commerciale** in quanto viene venduto per profitto e quindi sottoposto a diversi tipi di restrizione.

• Software shareware

Con il termine shareware si fa riferimento a software proprietario che può essere ridistribuito, ma per il quale viene espressamente richiesto il pagamento dopo un periodo di prova.

1.4 LA NASCITA DEL SOFTWARE LIBERO

L'esigenza di libertà nel settore del software è sempre stata sentita. Se oggi questo tipo di software rappresenta concretamente una scelta possibile, lo si deve all'azione di persone che con costante impegno hanno agito, legalmente, verso il raggiungimento di questo obiettivo.

UNIX e BSD

I primi utenti di UNIX sono state le università, a cui questo sistema operativo veniva fornito a costo contenuto, con i relativi codici sorgenti, ma senza alcun tipo di supporto tecnico, né alcuna garanzia. Proprio questa assenza di sostegno da parte della casa che lo aveva prodotto, stimolava la cooperazione tra gli utenti competenti, in pratica tra le università. Il maggior fermento intorno a UNIX si concentrò presso l'università della California a Berkeley, dove a partire dal 1978 si cominciò a distribuire una variante di questo sistema operativo: **BSD** (*Berkeley Software Distribution*). Per questo software, nacque una licenza d'uso che rimane ancora oggi il progenitore della filosofia del software libero: la licenza BSD. Tuttavia, questa edizione libera dello Unix BSD non ebbe vita facile, dato che da quel momento iniziarono delle contese giudiziarie sulla proprietà di alcune porzioni di codice ritenute libere, a torto o a ragione che fosse. Da questi problemi si svilupparono altri progetti indipendenti per ottenere, finalmente, un sistema BSD libero. Il primo di questi fu nominato **NetBSD**, al quale si aggiunse subito dopo **FreeBSD** e più tardi, anche **OpenBSD**. Allo stato attuale, le differenze più importanti tra le tre varianti, riguardano il tipo di piattaforma hardware su cui possono essere installate e l'origine della distribuzione stessa.



Nel 1985, Richard Stallman (foto a lato), fondò la **FSF** (*Free Software Foundation*) con lo scopo preciso di creare e diffondere la filosofia del software libero. Libertà intesa come la possibilità data agli utenti di distribuire e modificare il software a seconda delle proprie esigenze e di poter distribuire anche le modifiche effettuate. Queste idee filosofiche si tradussero in pratica nella redazione di un contratto di licenza d'uso, la **GPL** (*General Public License*), studiata appositamente per proteggere il

software libero in modo che non potesse essere “accaparrato” da chi poi avrebbe potuto impedirne la diffusione libera. Oggi, il copyright del software protetto in questo modo, viene definito copyleft. Il software libero richiedeva però delle basi, prima di tutto un sistema operativo. In questo senso, l’obiettivo pratico che si prefiggeva Richard Stallman era quello di realizzare, con l’aiuto di volontari, un sistema operativo completo e libero. Nacque così il progetto **GNU** (*Gnu’s Not Unix*), con il quale, dopo la realizzazione di un compilatore C, si desiderava costruire una serie di programmi di servizio necessari nel momento in cui il cuore del sistema operativo (kernel), fosse stato completo. Il progetto GNU diede così vita a una grande quantità di software utilizzabile sulla maggior parte delle piattaforme UNIX, indirizzando implicitamente il software libero nella direzione di sistemi di questo tipo.

MINIX

Alla fine degli anni ‘80, l’autorevole professore universitario Olandese Andrew S. Tanenbaum completa un sistema operativo Unix-like per elaboratori con architettura i86, realizzato specificamente per uso didattico, il suo nome era Minix. Molti studenti potevano finalmente comprendere i fondamenti della progettazione ed il funzionamento di un sistema operativo, mettendoci materialmente “le mani sopra”. Era infatti sufficiente acquistare il libro scritto dallo stesso professor Tanenbaum e si otteneva un sistema Minix completo di sorgenti. Tuttavia, Minix aveva seri problemi: poteva essere usato, distribuito e modificato, solo per fini didattici e comunque sembrava essere molto limitato sotto numerosi aspetti.

LINUX



Tra gli utenti insoddisfatti di Minix c’era anche Linus Benedict Torvalds (foto in alto). Era il 1991 quando Torvalds, allora studente al secondo anno di Informatica all’Università di Helsinki in Finlandia, decise di avviare un progetto sullo studio delle funzionalità di multiprogrammazione dei microprocessori i386 con l’idea di realizzare qualcosa di innovativo che portasse la tecnologia dei sistemi operativi UNIX sui Personal Computer. Torvalds non riusciva affatto ad accettare i limiti di Minix ed inoltre il progetto GNU, avviato anni prima da Richard Stallman, aveva reso disponibile un ottimo compilatore C. Fu quello l’anno di nascita del primo sistema basato sul kernel Linux, anche se, non si poteva ancora parlare di sistema operativo vero e proprio, ma soltanto della dimostrazione che la strada era giusta.

Linux non rimase il progetto personale di una persona, Torvalds infatti domandò aiuto a tutti coloro che fossero interessati al suo progetto servendosi di internet ed in breve tempo coinvolse un enorme numero di persone, unite dal fatto che si trattava di un progetto libero da qualunque restrizione legale. La fortuna di Torvalds, fu infatti quella di avere scelto subito la licenza GNU-GPL per la sua “creatura”, quella che ancora oggi rappresenta la difesa ideale per il software che viene scritto perché sia a disposizione di tutti. In questo modo egli superò il limite originale di Minix, che lo rendeva interessante solo per professori e studenti. La licenza GPL rendeva Linux interessante per chiunque. Dopo solo una quindicina di giorni dall’annuncio di Torvalds, quando ormai circa un migliaio di persone aveva messo mano a Linux e su tutti i newsgroup non si parla d’altro, finalmente anche il professor Tanenbaum in persona, fece sentire la sua voce. La condanna era netta: “*LINUX is obsolete*”.

Ci sono voluti più di quindici anni per arrivare dove siamo ora. E’ stata fatta tanta strada, costruendo qualcosa di unico e prima d’oggi neppure pensabile. Sviluppatori, Tester, Sistemisti, Utenti, hanno dato ciascuno il loro contributo e si sono tutti guadagnati un pezzetto di merito nella realizzazione del più grande progetto informatico libero mai realizzato. **Oggi il sistema Linux viene utilizzato in tutti i continenti, nei laboratori di ricerca scientifica, nelle basi spaziali della NASA, gestisce la maggior parte dei server che diffondono internet nelle nostre case e costituisce la più seria minaccia allo strapotere della Microsoft®. Stime non ufficiali parlano di circa 12 milioni di utenti in tutto il mondo, non male per un sistema che non ha una casa madre ne tanto meno un reparto marketing.**

1.5 OPEN SOURCE

Una volta compresa l'importanza del software libero, nel momento in cui hanno cominciato a delinearsi interessi economici, o di altro genere, si è posto il problema di definire in modo preciso e inequivocabile cosa sia effettivamente il «software libero». Nel 1998, nasce la definizione di **Open Source** (sorgente aperto), allo scopo di identificare i principi secondo cui il software può essere ritenuto libero. Tuttavia, nonostante le buone intenzioni, il termine resta ancora piuttosto ambiguo, dal momento che non sintetizza il significato che vorrebbe avere. Tecnicismi a parte, il vero problema, come sempre, rimane l'ignoranza, il software libero non è infatti ancora un concetto radicato e compreso a sufficienza in molti paesi, eppure la qualità del software Open Source migliora continuamente e, come è spesso accade, mette in seria difficoltà i prodotti di aziende proprietarie.

1.6 SETTE MOTIVI PER UTILIZZARE GNU/LINUX

- 1- **GNU/Linux**, nelle sue diverse distribuzioni, è oggi un sistema affidabile, completo, stabile ed estremamente potente.
- 2- **GNU/Linux** è immune al 98% dei Virus, ai Dialers ed agli Spyware, con conseguente aumento della privacy ed una maggiore sicurezza del proprio sistema e dei propri dati personali.
- 3- **GNU/Linux** è gratuito e può essere liberamente installato anche su diverse macchine poiché sottoposto alla licenza GPL (GNU Public License).
- 4- **GNU/Linux** supporta il Multi-Tasking e gestisce una reale Multi-Utenza.
- 5- **GNU/Linux** possiede interfacce grafiche estremamente intuitive e personalizzabili.
- 6- **GNU/Linux** è facile da aggiornare, con costanti miglioramenti sia al software in dotazione che al kernel in termini di: sicurezza, stabilità e riconoscimento hardware.
- 7- **GNU/Linux** gode di un vastissimo assortimento di Free-Software per svolgere ogni tipo di attività, a questi vanno anche aggiunti tutti i pacchetti sviluppati da grosse aziende come: IBM, Novell, Oracle, Compaq etc.