

**Piccola guida alla
comprensione del software
non proprietario e delle
licenze ad esso correlate**

Introduzione

Nel variegato mondo dell'informatica, relativamente nuovo ed in continua evoluzione, è necessario chiarire e distinguere alcune definizioni e concetti, spesso piuttosto confusi, relativi al software, poiché molti sono gli orientamenti di pensiero e le motivazioni etiche o pratiche che, negli ultimi 50 anni, hanno guidato svariate imprese e comunità di sviluppatori.

Una suddivisione pratica per introdurre le differenze esistenti nello sviluppo e distribuzione dei software prodotti nel mondo potrebbe essere di tipo insiemistico: software **proprietario** e **non proprietario**. $S = P + \bar{P}$

Il **software proprietario** è definito come il software che è sottoposto ad un qualche tipo di restrizione, imposto da un proprietario, sui seguenti aspetti del suo utilizzo:

- esecuzione;
- modifica;
- riproduzione;
- distribuzione.

Le barriere imposte alla libertà di fruizione dell'utente possono essere tecniche (rilascio del solo codice binario già compilato), oppure legali attraverso licenze, copyright e brevetti.

L'insieme del software **non proprietario** racchiude al suo interno una variegata gamma di importanti sfumature semantiche che saranno di seguito analizzate in rapporto ad un breve excursus storico dell'informatica degli ultimi 30 anni. Sarà importante coglierne le differenze, poiché spesso vi è una grande confusione tra terminologie correntemente utilizzate come sinonime nell'ambito del software non proprietario.

Un po' di storia

Agli inizi, anni '40-'50 i computer erano rarissimi, di dimensioni mastodontiche, dedicati a compiti specifici e programmati con software direttamente funzionale all'hardware: erano oggetto di studio scientifico e le conoscenze ad essi correlate liberamente condivise, nei limiti imposti da eventuali segreti militari. In seguito, durante gli anni '60, ebbe inizio un processo di standardizzazione nello schema di risoluzione dei problemi: il computer evolse da macchina costruita espressamente per il calcolo della soluzione di un problema specifico a macchina dedicata all'esecuzione di un programma generico. Nacquero i sistemi operativi. Nei Bells Laboratories ebbe inizio lo sviluppo di un software speciale che fungesse da strato intermedio di comunicazione tra l'hardware ed il software applicativo, allo scopo di separare sempre più il tipo di macchina fisica utilizzata dal compito specifico ad essa assegnato: il sistema operativo Unix/Posix. In seguito ad alcune vicissitudini, il sorgente di Unix fu liberamente concesso alle università ed a chiunque ne facesse richiesta; in breve si formò una comunità mondiale a livello universitario incentrata sullo sviluppo di nuove componenti ed applicazioni per Unix. Tutto il materiale venne condiviso, rigorosamente a codice aperto; avvenne il passaggio definitivo dal laboratorio isolato a un ambiente di lavoro comunitario, dove tutti potevano contribuire.

Per tutti gli anni 70 l'hardware fu ancora la componente di costo principale del computer: da ciò la scelta dei produttori di vendere il loro prodotto accompagnato da più software possibile, in quanto considerato parte naturale del prodotto complessivo. Tuttavia differenze tecnologiche e standard non ancora affermati limitò inizialmente la portabilità del software, in quanto esso funzionava solo su un preciso tipo di computer e non su altri, neanche dello stesso produttore. Con l'introduzione dei sistemi operativi, i programmi divennero sempre più portabili, in quanto diversi modelli di hardware potevano eseguire lo stesso sistema operativo.

La presenza di sistemi operativi funzionanti su macchine di differenti produttori ampliava ulteriormente le possibilità di usare lo stesso codice in modo relativamente

indipendente dall'hardware usato.

Gli anni '80 videro lo sviluppo iniziale di tre fenomeni importanti: il crollo progressivo dei costi legati all'hardware, il contemporaneo aumento di complessità - velocità di elaborazione – memoria e prestazioni, l'aumento della complessità del software e delle difficoltà legate al suo sviluppo.

I costi discendenti dell'hardware fecero concentrare il business sulla realizzazione del software, il cui sviluppo iniziò a passare di mano dalle università alle aziende: in questa fase le imprese avventurate in questo nuovo mercato iniziarono ad avvertire l'esigenza di tutelare il proprio lavoro, servendosi degli strumenti classici che il diritto industriale metteva a disposizione: il segreto industriale, il brevetto, il diritto d'autore.

Il codice sorgente era divenuto un investimento prezioso che poteva da un lato far acquisire una fetta di un mercato in rapida crescita, dall'altro legare gli utenti al proprio software mantenendo il segreto sui metodi utilizzati per lo sviluppo di sistemi e applicazioni. Non potendo vedere e modificare il codice sorgente del software, divenne necessario chiedere alle aziende di farlo.

Tuttavia un gruppo sempre più folto di informatici di vecchio stampo opposero resistenza a questo trend ispirato ad un'ottica unicamente di profitto, volendo dimostrare al mondo che il software poteva (anzi, doveva) rimanere uno strumento il più possibile libero da vincoli giuridici e fruibile da chiunque volesse intervenire sulla sua struttura e sulle sue funzionalità.

A questo scopo era fondamentale poter sempre disporre del codice sorgente, ovvero il codice in linguaggio di programmazione grazie al quale è possibile capire la struttura del software ed eventualmente modificarlo e correggerlo. Le imprese di software invece, sfruttando i diritti esclusivi del regime di copyright, distribuivano il software solo in linguaggio macchina (codice binario), criptando il codice sorgente e rendendo così ogni operazione di modifica, aggiornamento e adattamento impossibile o quantomeno difficile.

Essendo il software ormai un'opera sottoposta a copyright ed essendo il copyright un regime di tutela che si applica automaticamente con la creazione dell'opera, gli sviluppatori che avessero voluto distribuire le loro opere liberamente avevano solo due scelte: rilasciarle esplicitamente in un regime di public domain (ma questa scelta sarebbe stata controproducente poiché chiunque avrebbe potuto appropriarsi dell'opera e sfruttarla anche a fini commerciali, per di più criptando successivamente il codice sorgente); oppure rilasciarle sotto un particolare regime giuridico cristallizzato in una licenza nella quale l'autore, sempre fondandosi sui principi del copyright, disciplinava le modalità di utilizzo e distribuzione dell'opera. Tale particolare regime, nato nell'ambito di un progetto di ampio respiro denominato GNU, prese il nome emblematico di copyleft e i suoi principi cardine furono condensati in un'apposita licenza chiamata GNU General Public License (GPL): essa garantiva che il software fosse liberamente eseguibile, copiabile e modificabile, e soprattutto che chiunque ridistribuisse copie di quel software o creasse altro software derivato da quel codice mantenesse il medesimo regime di licenza. Uno scaltro escamotage che garantiva la persistenza all'infinito delle libertà caratteristiche del software libero.

Il mondo del software non proprietario

E' comune ascoltare, anche tra gli addetti ai lavori, l'utilizzo di espressioni tipo "open source", "free software", "freeware", "Linux", "Gnu"... come se fossero tutte sinonime ed intercambiabili indistintamente all'interno di uno stesso discorso. Per comprendere i vari movimenti alle spalle del software non proprietario è importante al contrario analizzare le differenze esistenti tra le varie terminologie e conoscerne l'origine.

Il progetto GNU

Richard M. Stallman, programmatore-guru pioniere di una nuova cultura basata sulla libera condivisione della conoscenza, è dagli anni '70 il principale artefice della ribellione ai meccanismi legali con cui viene protetto il software. Un famoso aneddoto narra che volesse implementare funzionalità aggiuntive al driver della stampante dell'ufficio: la ditta produttrice rifiutò di fornire i sorgenti poiché "segreti". Sarebbe assurdo imputare alla sola conseguente frustrazione la nascita di GNU, il più efficiente antagonista ai sistemi operativi e software applicativi "closed source", tuttavia Stallman si arrabbiò molto, e di lì a poco lasciò il MIT, rifiutò numerose offerte di lavoro presso importanti industrie e fondò la Free Software Foundation, una associazione senza scopo di lucro per la difesa del software, da lì in poi definito "libero".

GNU, che sta per "Gnu's Not Unix", è il nome del sistema software completo e Unix-compatibile che Stallman ha iniziato a scrivere una volta lasciato il MIT. Come ogni grande intellettuale segnò la storia scrivendo un [Manifesto](#) programmatico nel quale, tra le altre cose, risponde alle potenziali obiezioni circa lo sviluppo cooperativo di software ed alla libera distribuzione dei sorgenti.

GNU non è Linux!

Linux è un kernel, ossia un programma che si interfaccia a basso livello con l'hardware e fornisce funzionalità complesse ai pacchetti software del sistema operativo sovrastanti, da esso dipendenti.

Linux vede la luce nel 1991 grazie al giovane studente Linus Torvalds, grande appassionato di programmazione. Decise di creare il suo kernel partendo da zero e di utilizzare la licenza GPL per permettere a chiunque di contribuire e di utilizzarlo in modo libero. Non è l'unico kernel esistente: altre università ed aziende hanno sviluppato kernel Unix-compatibili, tra cui NetBSD, HURD, Solaris.

Il sistema GNU associato ad un kernel Linux forma un sistema operativo completo di tutto, anche dei programmi applicativi, tra cui citiamo la suite da ufficio OpenOffice, l'equivalente del più noto, ma proprietario, Microsoft Office. La denominazione corretta, quando si parla del pinguino (il logo di linux è un pinguino), è pertanto GNU/Linux.

Free software...

Con Stallman e la Free Software Foundation si inizia a parlare di “free software”. L'aggettivo "free", che in inglese significa sia gratuito che libero, a volte è mal interpretato. Non ha niente a che vedere col prezzo del software: si tratta di libertà.

“free speech, not free beer”

Ecco, dunque, la definizione di software libero: un programma è software libero per un dato utente se:

- l'utente ha la libertà di eseguire il programma per qualsiasi scopo;
- l'utente ha la libertà di modificare il programma secondo i propri bisogni (perché questa libertà abbia qualche effetto in pratica, è necessario avere accesso al codice sorgente del programma, poiché apportare modifiche a un programma senza disporre del codice sorgente è estremamente difficile);

- l'utente ha la libertà di distribuire copie del programma, gratuitamente o dietro compenso;
- l'utente ha la libertà di distribuire versioni modificate del programma,

così che la comunità possa fruire dei miglioramenti apportati.

Poiché "free" si riferisce alla libertà e non al prezzo, vendere copie di un programma non contraddice il concetto di software libero. In effetti, la libertà di vendere copie di programmi è essenziale: raccolte di software libero vendute su CD-ROM sono importanti per la comunità, e la loro vendita è un modo di raccogliere fondi importante per lo sviluppo del software libero. Di conseguenza, un programma che non può essere liberamente incluso in tali raccolte non è software libero.

... ed Open Source

Se lo scritto programmatico di Stallman ricorda tanto il romanticismo europeo, altrettanto non fa la Open Source Definition, il documento redatto nel '98 da Bruce Perence (attivista Debian), un nuovo manifesto incentrato sui diritti dell'utente e sui requisiti tecnici delle licenze ammesse a far parte del nuovo movimento antagonista al software proprietario.

Per invogliare il restio mondo imprenditoriale ad investire nel concetto di “free software”, ovviando alle possibili ambiguità, e per sollevarsi dall'apparato di principi etici piuttosto monolitico che accompagnava l'operato della Free Software Foundation – compagni scomodi per una promozione del modello free software a livello commerciale - venne coniato il termine "Open Source", ossia “sorgente aperto”, che appunto cerca di focalizzare l'attenzione sulle caratteristiche tecniche del software (la disponibilità del codice sorgente) piuttosto che sugli aspetti etici.

La Open Source Initiative è l'ente che coordina le attività di promozione, tutela e informazione relative al software certificato open source.

LICENZE INFORMATICHE

(a cura di Lorenzo Faedi)

Licenza GNU GPL

Scritta da Richard Stallman e Ebe Moglen per distribuire i programmi del progetto GNU, che ha come scopo la creazione di un sistema operativo libero. Si contrappone alle licenze per il software proprietario e permette all'utente la libertà di utilizzo, modifica, e distribuzione del programma che incorpora tale licenza. È ancora in utilizzo la seconda versione di GNU GPL, scritta nel 1991, ma è in corso la definizione della terza versione. Se l'utente distribuisce copie del software, deve rendere disponibile il codice sorgente a ogni acquirente, incluse tutte le modifiche eventualmente effettuate (questa caratteristica è detta copyleft). Nella pratica, i programmi sotto GNU GPL vengono spesso distribuiti allegando il loro codice sorgente, anche se la licenza non lo richiede. Ci sono casi in cui viene distribuito solo il codice sorgente, lasciando all'utente il compito di compilarlo. L'utente è tenuto a rendere disponibile il codice sorgente solo alle persone che hanno ricevuto da lui la copia del programma. Dato che il software è protetto da copyright, l'utente non ha altro diritto di modifica o redistribuzione al di fuori dalle condizioni di copyleft. In ogni caso, l'utente deve accettare i termini della GNU GPL solo se desidera esercitare diritti normalmente non contemplati dalla legge sul copyright, come la redistribuzione. Al contrario, se qualcuno distribuisce un software senza rendere disponibile il codice sorgente o violando in altro modo la licenza, può essere denunciato dall'autore originale secondo le stesse leggi sul copyright. La licenza specifica anche che il diritto illimitato di redistribuzione non è garantito, in quanto potrebbero essere trovate delle debolezze legali all'interno della definizione di copyleft. La Free Software Foundation (FSF) detiene i diritti di copyright per il testo della GNU GPL, ma non detiene i diritti del software rilasciato con questa licenza. A meno che venga emessa una specifica nota di copyright da parte della FSF, l'autore del software detiene i diritti di copyright per il suo lavoro, ed è responsabile di perseguire ogni violazione della licenza riguardante il suo software. In caso di

violazione della licenza il detentore del copyright è l'unico che può richiederne il rispetto nei termini di legge ed è l'unico che può chiedere aiuto alla FSF per la protezione di tale software, quest'ultima può intervenire solo se è il detentore del copyright a richiederne il suo aiuto. La FSF, normalmente, mette a disposizione i suoi legali per proteggere il software coperto da una licenza GNU, chiedendo in un primo tempo il rispetto della licenza e solo se tale richiesta non viene accettata si arriva al tribunale. A differenza dei software con essa distribuiti, la GNU GPL non è liberamente modificabile: copiarla e distribuirla è permesso, ma modificarla è vietato. La FSF permette di creare nuove licenze basate sulla GNU GPL, a patto che tali licenze non usino il suo preambolo senza permesso. Dato che, solitamente, la nuova licenza non è compatibile con la GNU GPL, la FSF sconsiglia di creare versioni modificate.

GNU Free Documentation License (GNU FDL)

Licenza di copyleft per contenuti liberi, creata dalla Free Software Foundation per il progetto GNU. La GNU FDL è stata creata per distribuire la documentazione di software e materiale didattico. Stabilisce che ogni copia del materiale, anche se modificata, deve essere distribuita con la stessa licenza. Tali copie possono essere vendute e, se riprodotte in gran quantità, devono essere rese disponibili anche in un formato che faciliti successive modifiche. La “Sezione Secondaria” è una parte separata dal resto del documento e può contenere informazioni sulla relazione tra l'autore e il soggetto del documento. A differenza del testo del documento, la cui modifica è concessa alle condizioni equivalenti a quelle della GNU General Public License, alcune delle Sezioni Secondarie hanno delle restrizioni studiate per garantire la giusta attribuzione del testo agli autori precedenti. In particolare, gli autori di una versione precedente devono essere specificati e certe "sezioni invariati", specificate dall'autore originale e riguardanti il suo rapporto con il documento, non possono essere cambiate. Se il documento viene modificato, il suo titolo va cambiato a meno che l'autore precedente conceda il permesso di mantenerlo invariato.

Licenza GNU LGPL

Creata dalla Free Software Foundation come compromesso fra GPL e altre licenze non-copyleft. Stabilisce il copyleft solo su un singolo file e non sull'intero software. Questo comporta che, per esempio, un software rilasciato sotto licenza LGPL può essere incluso liberamente in un'applicazione commerciale closed-source, a patto che le modifiche apportate al codice sorgente del software stesso vengano rese pubbliche; tutti gli altri file dell'applicazione possono essere rilasciati con licenza proprietaria e senza codice sorgente.

Licenza BSD

E' una licenza usata per la distribuzione del sistema operativo Berkeley Software Distribution, una variante di Unix sviluppata dall'università di Berkeley. Garantisce le libertà del software al pari di una licenza GPL, ma a differenza di quest'ultima chiunque modifichi un programma con licenza BSD può ridistribuirlo usando qualsiasi altra licenza senza l'obbligo di rendere pubbliche le modifiche apportate al codice sorgente.

Creative Commons

La Creative Commons (CC) è un'organizzazione non-profit dedicata all'espansione della portata delle opere di creatività disponibili per la condivisione e l'utilizzo da parte degli altri e per fornire anche la possibilità di poter costruire, com'è sempre avvenuto prima che si abusasse della legge sul copyright, sul lavoro degli altri nel pieno rispetto delle leggi esistenti. Il sito web di Creative Commons permette a quanti detengono dei diritti di copyright di trasmettere alcuni di questi diritti al pubblico e di conservarne gli altri, per mezzo di una varietà di schemi di licenze e di contratti che includono la destinazione di un bene privato al pubblico dominio o ai termini di licenza di contenuti aperti (*open content*). L'intenzione è quella di evitare i problemi che le attuali leggi sul copyright creano per la diffusione e la condivisione delle informazioni. Il progetto fornisce diverse licenze libere che i detentori dei diritti di

copyright possono utilizzare quando rilasciano le proprie opere sulla Rete. Le licenze Creative Commons sono state anticipate dalle licenze Open Publication License (OPL) e GNU Free Documentation License (GFDL). La GFDL è intesa principalmente come una licenza per la documentazione software, ma è anche in uso per progetti che non riguardano strettamente il software. Entrambe la OPL e la GFDL contenevano delle sezioni opzionali che, nell'opinione dei critici, le rendevano meno libere. La GFDL si differenzia dalle licenze creative commons nella sua richiesta che i lavori licenziati con essa vengano distribuiti in una forma "trasparente", ad esempio non usando formati proprietari e/o segreti. Creative Commons è nato ufficialmente nel 2001 per volere del professore Lawrence Lessig, ordinario della facoltà di Giurisprudenza di Stanford (e in precedenza anche di Harvard) e riconosciuto come uno dei massimi esperti di diritto d'autore negli Stati Uniti. Lessig fondò l'organizzazione come metodo addizionale per raggiungere il suo scopo nel suo caso di fronte alla Corte Suprema degli Stati Uniti, *Eldred v. Ashcroft*. Il set iniziale delle licenze creative commons fu pubblicato il 16 dicembre 2002 .

Licenze per software proprietario:

EULA o *End User License Agreement* (Contratto con l'utente finale): licenza contenuta in tutti i software closed-source che l'utente finale deve necessariamente accettare per poter installare il programma. Spesso in questo testo il produttore declina ogni responsabilità per i danni all'hardware o al software installato sul computer che il programma potrebbe provocare, e molto spesso l'EULA protegge più il produttore che l'utente finale che accetta il contratto. La maggior parte degli EULA accompagna software venduti in scatola e viene presentata a volte su carta o, più frequentemente, direttamente sullo schermo del pc durante la procedura di installazione. L'utente può scegliere se accettare o rifiutare il contratto. L'installazione del software è subordinata all'accettazione del contratto.

DISTRIBUZIONI

La distribuzione software è un insieme di programmi rilasciati in un unico pacchetto e configurati in modo da formare un ambiente di lavoro più o meno omogeneo.

Il caso al quale ci si riferisce più frequentemente con il termine distribuzione è quello di ambienti software differenti che si basano però sul medesimo kernel Unix-like. La filosofia di base è quella secondo cui il sistema operativo è l'insieme di tutti i programmi necessari per un uso completo della macchina.

La distribuzione Linux comprende tutti i pacchetti software basati su kernel Linux. Come già detto, con il termine *Linux* si indica solo il kernel del sistema operativo, mentre le distribuzioni comprendono, oltre al kernel, varie componenti software quali l'interfaccia grafica, applicativi gratuiti free, open source e a volte anche software proprietario.

Le distribuzioni Linux sono sviluppate da progetti, fondazioni e anche società. Fra le distribuzioni più importanti e utilizzate è bene ricordare Red Hat, Ubuntu, Debian e Fedora Core.

RED HAT.

Red Hat Inc. (società americana produttrice di Red Hat Linux) e' stata la prima società fornitrice di soluzioni Open Source ad essere quotata alla borsa dei titoli tecnologici di Wall Street. La distribuzione Red Hat introdusse la commercializzazione del prodotto ampliando il bacino d'utenza di Linux e dei suoi applicativi. Insieme al pacchetto viene offerta la possibilità di avere assistenza tecnica sui prodotti Red Hat. Questa distribuzione ha anche inserito la possibilità di aggiornare le proprie applicazioni con la nuova distribuzione.

I service provider, sapendo di dover contollare i costi, si orientano sempre più verso soluzioni basate su hardware e software open source. In questo contesto Red Hat si è posto alla ribalta come il software provider leader di Linux nel settore delle Telecomunicazioni. La famiglia dei prodotti di Red Hat Enterprise Linux è in grado

di soddisfare i requisiti sulle prestazioni, affidabilità e sicurezza richiesti dai service provider. Red Hat può vantare un numero maggiore di partner hardware e software certificato rispetto ad altri distributori Linux.

Red Hat lavora in stretto legame con la community open source in modo da poter stabilire la direzione e le priorità dei futuri programmi di Linux. Un gran numero di persone che mantengono i progetti open source, risultano anche dipendenti di Red Hat. Attraverso una programmazione e coordinazione dettagliata con la community, Red Hat è in grado di dettare i tempi in modo da avere sempre un prodotto aggiornato.

Red Hat Enterprise Linux è la distribuzione Linux più stabile presente nel mercato.

Il modello di sviluppo di Red Hat rappresenta la guida per una innovazione del settore, pur mantenendo i contenuti di affidabilità.

UBUNTU

Ubuntu è una distribuzione Linux nata nel 2004 e basata su Debian, che si concentra sulla facilità di installazione e d'uso e sul rilascio regolare delle nuove versioni. Il nome deriva da una antica parola Zulu diffusa in varie parti dell'Africa e che corrisponde indicativamente al concetto di "*umanità verso gli altri*", a volte tradotto anche "*io sono ciò che sono per merito di ciò che siamo tutti*". La distribuzione Ubuntu Linux si prefigge l'obiettivo di portare nel mondo del software questo nobile concetto.

Viene finanziata dalla società Canonical Ltd pur rimanendo in tutto e per tutto un software libero. L'ideatore dell'iniziativa e titolare di Canonical è Mark Shuttleworth, un giovane imprenditore sudafricano diventato fiero sostenitore dell'open source, al cui servizio ha posto le sue risorse. Il 1 luglio 2005, è nata la Fondazione Ubuntu con un fondo iniziale di 10 milioni di dollari.

Ubuntu è basata sull'ambiente desktop GNOME, che è specificatamente progettato per fornire un'interfaccia libera, semplice, intuitiva ed allo stesso tempo un'ampia

gamma di moderne applicazioni desktop. Dispone di più di di 16.000 applicazioni, tra le quali software per la produttività personale, incluso OpenOffice, il browser web Mozilla Firefox e l'editor grafico GIMP ma anche client di posta elettronica, editor per la programmazione, giochi e molto altro. La distribuzione è tradotta in tutte le lingue occidentali, orientali ed ovviamente anche africane.

Ubuntu utilizza APT (Advanced Packaging Tool), noto gestore di pacchetti per Debian. Probabilmente il grande successo di Ubuntu è da affidare proprio a questo tipo di gestione di pacchetti che consente a Ubuntu di essere installabile in una completa versione tramite un solo CD, per poi aggiornarsi via APT con i più nuovi pacchetti disponibili. Quasi la completa totalità dei pacchetti disponibili per Linux sono disponibili tramite i repository APT, e questo consente una rapida installazione dei programmi, una sicura ed efficiente rimozione ed una ottima gestione degli aggiornamenti.

DEBIAN

Debian, creata dal Debian Project, è una distribuzione di software libero largamente usata e sviluppata attraverso la collaborazione di volontari da ogni parte del mondo.

Debian è conosciuta per la sua aderenza alle filosofie di GNU e del software libero, le rigide politiche riguardo alla qualità dei pacchetti, il modo aperto di sviluppare e testare il software e la libertà di scelta concessa all'utente. Il Debian Project ha formalizzato con i suoi utenti questi impegni attraverso il Contratto Sociale Debian e le Linee Guida Debian per il Software Libero.

L'ultimo rilascio stabile di Debian contiene più di 15.000 pacchetti software per una decina di architetture. Il progetto Debian ha tra i suoi obiettivi quello di creare "il sistema operativo universale": sono stati sviluppati dei port per altri kernel, fra i quali GNU Hurd, Solaris, NetBSD, e FreeBSD. Questi port non sono ancora stati rilasciati ufficialmente e quindi, al momento, Debian è esclusivamente una distribuzione GNU/Linux.

Debian è sostenuta da donazioni attraverso "Software in the Public Interest" (SPI Inc.), una organizzazione non-profit per i progetti di software libero.

FEDORA

Fedora Core è una distribuzione Linux curata dal Progetto Fedora, un progetto Open Source sponsorizzato (ma non direttamente supportato) da Red Hat e supportato dalla community. L'obiettivo è quello di lavorare con la community di Linux per creare un sistema operativo completo, utilizzando esclusivamente software liberi e forum pubblici con processi aperti. Questo porta a un continuo aggiornamento delle distribuzioni, aggiornamento che è diventato molto semplice, anche da una distribuzione all'altra. È un settore adibito a nuove tecnologie che in futuro potrebbero essere integrate all'interno dei prodotti di Red Hat.

Fedora è indicata per qualsiasi utilizzo poiché contiene tutti gli strumenti per lavorare in ambiente server senza tralasciare una serie di tool grafici per la configurazione, utili per gli utenti alla ricerca di un desktop user-friendly. Basata sulla ricerca del massimo grado di aggiornamento è rilasciata su base fissa semestrale con le versioni più recenti di ogni pacchetto, compreso il kernel.

Fedora si distingue per la facilità di installazione, ottimi strumenti di configurazione, il sostegno da parte della comunità Linux. Fedora è anche in grado di riconoscere automaticamente l'hardware su cui è installato. È supportata solo da macchine recenti e ancora non può essere considerata una distribuzione stabile.

Fedora è oggi una delle distribuzioni Linux più diffuse, seconda solo a Ubuntu, anche perché diretta filiazione della distribuzione Red Hat, una delle più diffuse negli anni passati.

Copyleft

Articolo riportato dal libro "Compendio di libertà informatica e cultura open" a cura di Simone Aliprandi

L'espressione "copyleft" nasce dalla prassi goliardica di alcuni sviluppatori di software che distribuivano copie dei loro lavori riportanti la dicitura "copyleft - all rights reversed" (con una © rovesciata). In effetti il termine è molto significativo poichè racchiude un duplice gioco di parole: "left" è appunto il participio passato di "leave" (lasciare, permettere) e comunica l'idea di un regime più libero; ma è anche l'opposto di "right" (destra) e comunica un'idea di ribaltamento dei principi.

Dopo il 1989 (anno di nascita della GPL) comparvero altre licenze ispirate alla stessa filosofia, ma gli ideologi/porta-voce del progetto GNU si preoccuparono di dare al termine copyleft una configurazione piuttosto netta: ovvero nell'accezione originaria è considerata vera licenza copyleft quella che impone il mantenimento all'infinito del medesimo regime (nel linguaggio Creative Commons, la cosiddetta clausola "share-alike", cioè "condividi allo stesso modo"). Questo nuovo modello di gestione dei diritti d'autore ha avuto fin da subito grande rilevanza socio-culturale e col tempo l'espressione "copyleft", forse per la sua particolare efficacia semantica, è stata usata per indicare più ampiamente tutto questo fenomeno giuridico di rivisitazione del modello tradizionale di gestione dei diritti d'autore. E nonostante le critiche dei puristi del movimento, questo allargamento semantico è ormai un dato di fatto in gran parte della documentazione e saggistica in materia.

Con la diffusione del software libero e del software open source anche in un circuito commerciale e di massa, ci si è spesso trovati di fronte ad un paradosso: tutta la documentazione (istruzioni tecniche, manuali, presentazioni) relativa al software libero e prodotta dagli stessi sviluppatori, veniva editata in un regime di copyright tradizionale. Molti autori, soprattutto i "guru" del movimento (primo fra tutti Richard Stallman) pubblicavano i loro articoli d'informazione e sensibilizzazione

accompagnati da una breve nota di copyleft che suonava più o meno così: "è permessa la copia letterale dell'opera con ogni mezzo a condizione che venga riportata questa nota".

In questo laconico disclaimer si condensa in effetti molto efficacemente il senso pratico del modello copyleft persistente; dal punto di vista giuridico però tale laconicità poteva essere foriera di abusi e interpretazioni fuorvianti. Tra l'altro l'uso di questa nota nel caso di documentazione poteva non essere particolarmente appropriato poiché non si contemplava la possibilità di modifica dei contenuti dell'opera: possibilità determinante trattandosi di manuali di software liberamente modificabile, oltre che liberamente copiabile. Alcuni autori scelsero di applicare la GPL anche alle opere di documentazione, ma come è già emerso si tratta di una licenza pensata e palesemente riferita ad un'opera tecnico-funzionale come il software. Ecco che nel 2000 nacque (sempre in seno al progetto GNU) la Free Documentation License: una licenza appositamente pensata per le opere letterarie, dunque una delle prime licenze copyleft in ambito content e non solo strettamente software.

Sulla scia di questo nuovo spiraglio apertosi in ambito informatico e più in generale della diffusione massiccia di Internet, in quegli anni (cioè dalla fine degli anni Novanta) si attivarono alcuni progetti di promozione della libera circolazione delle informazioni e delle opere creative. Ogni progetto propose la propria "ricetta" per sdoganare i principi del copyleft anche in quell'ambito non più strettamente informatico: nacquero così alcune licenze come - per citarne solo alcune - la Open Publication License (del progetto OpenContent), la OpenAudio License (della Electronic Frontier Foundation), la OpenMusic License (del progetto tedesco OpenMusic), Licence Art Libre (del progetto francese Art Libre). Fu però un gruppo di giuristi di Stanford (capitanati dal professor Lawrence Lessig) a fare il passo più determinante in questo senso, con l'attivazione del progetto Creative Commons e la diffusione nel 2002 delle relative licenze: queste licenze erano pensate in modo da

poter funzionare per tutti i tipi di opere creative e in modo da poter essere tradotte e possibilmente adattate ai vari ordinamenti giuridici. Tra l'altro la loro struttura si articolava in clausole modulari che permettevano all'autore di decidere quali usi consentire per la sua opera, a quali condizioni e in quali contesti: in poche parole, consentivano all'autore di graduare la libertà di utilizzo dell'opera, chiarendone le condizioni.

Il senso del copyleft in sintesi.

Cercando di dare una definizione semplice e chiara al concetto di copyleft, possiamo dire che si tratta di un modello alternativo di gestione dei diritti d'autore, che opera - a differenza del modello tradizionale - in un'ottica non esclusiva e non standardizzata e che deriva originariamente dalla libera scelta dell'autore. Esso si realizza in concreto grazie all'applicazione di alcuni contratti-licenza che disciplinano la diffusione dell'opera e chiariscono a quali condizioni essa può essere condivisa, modificata, commercializzata.

I principali effetti di tale modello sono:

- disintermediazione, nel senso che è l'autore stesso a decidere a priori alcune regole relative alla diffusione della sua opera e tali regole sono rivolte a tutta la comunità degli utenti, senza più necessità di un soggetto imprenditoriale che si occupi della distribuzione e commercializzazione dell'opera (editore, produttore, etc.);
- riequilibrio, nel senso che, qualora ci fosse comunque l'interazione di un soggetto imprenditoriale (cosa spesso auspicabile), gli equilibri contrattuali relativi ai diritti sull'opera verrebbero ampiamente ridefiniti, spostandosi maggiori prerogative nella sfera dell'autore e creandosi maggiori libertà nella sfera dell'utente finale;
- elasticità e differenziazione, nel senso che in questo modello ogni opera ha un suo specifico regime giuridico e tale regime può strutturarsi in modo

decisamente più elastico e dinamico, così da adattarsi meglio alla nuova (e in continua evoluzione) compagine del mondo della comunicazione multimediale;

- sostenibilità economica: grazie a queste sue caratteristiche, tale paradigma riesce a realizzare un ideale di modello economico sostenibile, tanto nel mondo informatico (ne è la prova da più di un decennio il successo anche economico del software open source) quanto nel mondo della produzione intellettuale in generale.

Bibliografia

- Wikipedia, l'enciclopedia libera (<http://it.wikipedia.org>)
- [Scritti originali di Richard Stallman \(http://www.gnu.org/philosophy\)](http://www.gnu.org/philosophy)
- “Compendio di libertà informatica e cultura open” a cura di Simone Aliprandi
– Edizioni PRIMAORA