

MODALITÀ DEL TEST

Dal momento che non stiamo testando del software non vi forniremo la prova su strada di questo o quel software. Piuttosto ci siamo concentrati su come le licenze proteggono la libertà degli utenti. Ci riferiamo alla definizione di libertà così come trasfusa dalla Free Software Foundation (FSF) nelle famose "quattro libertà": **(0)** La libertà di eseguire il programma per qualsiasi scopo; **(1)** La libertà di studiare come funziona il programma e cambiarlo secondo le nostre esigenze. La disponibilità del codice sorgente è prerequisito indispensabile; **(2)** La libertà di ridistribuire copie in modo da aiutare il nostro prossimo; **(3)** La libertà di distribuire anche copie modificate del software. In questo modo si concede a tutta la comunità la possibilità di beneficiare delle modifiche da noi introdotte. L'accesso al codice sorgente è, anche in questo caso, prerequisito indispensabile.

Licenze libere

Tutto il software Open Source è libero, ma certi programmi sono più liberi di altri. Linux Pro vi offre una panoramica delle principali licenze dei nostri software "open"



Sesso parliamo di software Open Source come se dietro questo termine ci fosse un'unica idea e una singola filosofia. In realtà questa definizione ricomprende un ampio ventaglio di varianti che si contraddistinguono tra loro per la licenza volta per volta adoperata. Con le licenze gli sviluppatori – o comunque i titolari dei diritti sulle singole opere – condensano e rendono concreta la loro idea di gestione dei diritti d'autore sul singolo progetto allo stesso modo in cui un gruppo o movimento politico infonde i propri ideali in un manifesto politico, con la differenza, sostanziale, che le licenze creano dei vincoli giuridici. Con le licenze, infatti, gli sviluppatori sottolineano quali siano i diritti concessi al singolo utente e a quali limiti debbano soggiacere i futuri

sviluppatori della stessa opera. Non c'è, pertanto, un'unica definizione di *Open Source* in quanto, ad esempio, per alcuni progetti software sono state create specifiche licenze che, nel rispetto degli ideali e delle libertà del Software Libero, acquistano una certa fama nel mondo open. Le più famose licenze libere sono quelle portate avanti dalla **Open Source Iniziative**, da **Debian** e dalla **Free Software Foundation**. Esse sono leggermente differenti e vedremo, di seguito, di evidenziarne le peculiarità. Abbiamo scelto di soffermarci su cinque licenze che coprono tutte le principali varianti della filosofia Open Source e che siano applicate alla maggior parte del software "open". Queste sono (e tra parentesi indichiamo il software più famoso che le adotta): **GPLv2** (Kernel Linux),

GPLv3 (pacchetti software contenuti nella maggior parte delle distribuzioni Linux, come, ad esempio, Bash e GCC), **Apache** (adottata dal Web server Apache, Android e dalle più recenti versioni di OpenOffice.org), **LGPL** (LibreOffice) e **BSD** e sue varianti (adottate da tutti i sistemi operativi BSD oltre che da X.org). Vi presentiamo, quindi, la nostra analisi al microscopio delle suddette licenze che vi consentirà di capire come funzionano.

LA NOSTRA SELEZIONE

- GPLv2
- GPLv3
- LGPL
- Apache
- BSD-style

Copyleft



**“I buoni artisti copiano, quelli grandi rubano”
– Picasso Linux Pro. E alcune licenze lo consentono**

Creare un codice destinato a restare aperto

Probabilmente il punto più evidente di discriminazione tra le licenze open è tra quelle che richiedono che le modifiche al codice debbano essere anch'esse rilasciate libere e quelle che non prevedono questa restrizione. Con riguardo alla prima situazione (in cui si parla anche di copyleft come “opposto” di copyright), ciascun sviluppatore che aggiunge un qualcosa al software deve condividere, poi, tali innovazioni con tutti gli altri utenti. Nella seconda situazione, invece, può tenere le modifiche per sé senza doverle ridistribuire. La GPL è stata la prima grande licenza ad aver introdotto il concetto di copyleft, ed è ancora la licenza più popolare. Ogni versione della GPL conserva questa caratteristica. Gli sviluppatori, in base a questa licenza, quando ridistribuiscono il software modificato devono anche condividere le porzioni di codice da loro create o modificate. Ciò significa che se non intendono condividere anche le loro porzioni modificate non dovrebbero usare questo tipo di licenza. Ovviamente per molti progetti si sceglie da subito di renderne

disponibile il codice sorgente. Se essi tengono le modifiche per sé allora non devono condividerne queste modifiche con tutti gli utenti. Questo sistema, però, non si adatta bene agli ambienti di cloud computing in cui gli sviluppatori non distribuiscono il software ma consentono, unicamente, agli utenti di utilizzarlo. In questa situazione uno sviluppatore potrebbe modificare software rilasciato con licenza GPL e concederne, dietro pagamento, l'accesso agli utenti senza rendere disponibile all'intera comunità il codice sorgente modificato. Per evitare ciò la FSF ha rilasciato la **AGPL** (Affero General Public License), la quale richiede agli sviluppatori di condividere il loro codice sorgente quando rendono disponibile l'applicazione al pubblico indipendentemente dal fatto che invece di distribuire il software ne concedano unicamente l'utilizzo. Le licenze Apache o della famiglia BSD, invece, non hanno alcuna forma di copyleft, e le aziende non hanno preclusioni di sorta a incorporare il codice sorgente in prodotti commerciali. I fautori di questo tipo di licenze ritengono che esse siano

più adatte alle attività commerciali e pertanto vengono inserite in molti prodotti. Ad esempio, quando Microsoft dovette introdurre il sistema di rete TCP/IP in Windows, utilizzò del codice rilasciato sotto licenza BSD. In questo modo Microsoft ottenne uno stack di rete che poteva rendere interoperabili molti sistemi diversi. Questo potrebbe non essere un bene per il Software Libero, ma è stato un bene per i sistemi Microsoft in quanto ha consentito alla maggior parte dei computer (che usa Windows) di funzionare meglio. Allo stesso modo quando Apple decise di ripensare all'intero sistema operativo lo fece basandosi su codice FreeBSD. In questo modo Apple ha potuto concentrare la maggior parte del tempo dei suoi sviluppatori nel creare un'interfaccia pulita e non ha dovuto perdere tempo nella creazione ex novo di un kernel. Le licenze copyleft intendono migliorare il mondo del Software Libero, mentre le licenze non-copyleft (chiamate spesso “permissive”) si sono occupate di migliorare il mondo dell'informatica in genere. Molti dei software più utilizzati sono stati rilasciati sotto licenze permissive, come Android (tranne il kernel), Chromium, il server Web Apache e OpenOffice.org. Infatti il kernel Linux è uno dei pochi grossi progetti rilasciati sotto licenza copyleft.

Verdetto

- GPLv3** ★★★★★
- GPLv2** ★★★★★
- LGPL** ★★★★★
- Apache** ★★★★★
- BSD-style** ★★★★★

» Questa è una categoria tutto-o-niente, con la FSF in cima alla classifica.

Tivoizzazione

Mettiamo le mani dentro la “scatola nera”



Il mondo dell'informatica è noto per il suo linguaggio particolarmente ostico, ma anche in questo ambiente è presente da alcuni anni il termine *tivoizzazione*. TiVo è un piccolo scatolotto che consente agli utenti di vedere e registrare la TV digitale grazie a un software rilasciato sotto licenza GPL. Il produttore, nel rispetto della licenza, ha reso pubblica la versione del software installato in TiVo. Tuttavia, benché agli utenti fosse concessa la possibilità di modificare il software che gira su TiVo, dopo averlo modificato si accorgevano che, a causa di una limitazione hardware, non potevano reinstallare il software da loro modificato sul dispositivo. Ciò ha determinato uno spaccamento nella comunità Open Source. Linus Torvalds

ha supportato le ragioni dell'azienda produttrice di TiVo in quanto il solo fatto di aver reso pubblico il codice sorgente avrebbe consentito ad altri produttori hardware di usufruire dei benefici e delle modifiche apportate. Per Richard Stallman, invece, gli utenti non dovrebbero solo essere in grado di accedere al codice, ma anche di modificarlo e applicarlo sul dispositivo originale. Le uniche licenze che contengono una clausola contro il fenomeno della tivoizzazione sono la LGPLv3 e la GPLv3. Entrambe queste licenze prevedono che gli utenti debbano avere la possibilità di utilizzare il loro software modificato sul dispositivo originale. Però queste licenze prevedono anche delle eccezioni all'applicazione

della clausola anti-tivoizzazione, ad esempio quando il software originale è impresso su una memoria non riscrivibile. È proprio a causa di questa clausola che Ubuntu (pensando a una futura versione di Ubuntu che avrà un sistema di boot sicuro) ha, invece, deciso di applicare al bootloader la licenza GPLv2. Secondo la Free Software Foundation, però, la scelta di Canonical di un bootloader con licenza GPLv2 è dovuta a un errore interpretativo delle clausole della licenza GPLv3. Ciò non toglie che la clausola anti-tivoizzazione sia di difficile interpretazione. Questa clausola, però, non deve essere intesa necessariamente quale un punto critico, una debolezza, della licenza ma, invece, rappresenta una scelta per garantire agli utenti una maggiore libertà.

Verdetto

- GPLv3** ★★★★★
- GPLv2** ★★★★★
- LGPL** ★★★★★
- Apache** ★★★★★
- BSD-style** ★★★★★

» Non tutti se ne preoccupano, ma se lo fate, scegliete con cura la licenza.

Ulteriori problemi

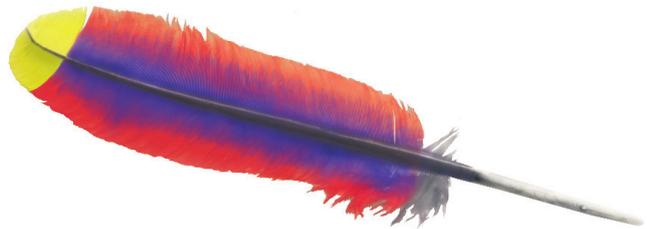
Le clausole ulteriori di cui si deve essere informati

Ciascuna licenza ha le sue particolarità, le sue clausole caratteristiche e i requisiti ulteriori che le rendono uniche. Spesso, però, licenze diverse utilizzano termini o clausole simili e ciò ingenera una certa confusione nei programmatori che di frequente sottovalutano un'attenta lettura dell'intera licenza. In questa sezione analizzeremo

queste clausole ulteriori alle quali si dovrebbe prestare comunque attenzione quando si sviluppa del codice. Analizzeremo anche le diverse versioni della stessa licenza e come i suoi termini e condizioni sono cambiati nelle diverse versioni. Come noterete, questi cambiamenti possono essere molto rilevanti.

Apache ★★★★★

La licenza Apache è piacevolmente semplice in quanto ci sono solo due versioni di essa, opportunamente chiamate "uno" e "due". Abbiamo deciso, però, di analizzare solo l'ultima versione, considerato che da quando nel 2004 è stata lanciata la versione 2 della licenza Apache quasi tutto il software già licenziato con la precedente versione è transitato ad essa. Se un file di testo NOTICE è incluso nel software, allora questo deve essere incluso nelle versioni modificate (o reso agevolmente reperibile all'interno del programma). Ulteriori informazioni possono essere aggiunte a questo file, a condizione che sia possibile distinguere le nuove parti da quelle preesistenti. Le informazioni sulla licenza devono essere presenti in ogni file, e se si cambia qualcosa, tali modifiche dovranno essere documentate anche nel file. Al di là di queste due restrizioni, e delle informazioni brevettuali comprese altrove, siete più o meno liberi di fare quello che vi piace con il codice con licenza Apache.



BSD-style ★★★★★

Le licenze in stile BSD sono le più semplici e non impongono pressoché alcun limite alla modifica del codice licenziato. Tuttavia, a causa della loro semplicità, molti hanno preferito integrarle o modificarle creando così nuove e differenti versioni della stessa licenza. La licenza originaria ha quattro clausole, di seguito riassunte.

- 1 La distribuzione del codice sorgente deve includere un avviso di copyright.
- 2 La distribuzione del codice binario deve includere l'avviso di copyright nella documentazione o nei file di "help".
- 3 Il materiale pubblicitario deve indicare chi sono gli sviluppatori.
- 4 I nomi degli sviluppatori non possono essere utilizzati per promuovere opere derivate.

La terza clausola rende le licenze BSD-style incompatibili con la GPL ed è stata rimossa nel 1999 per dare così vita alla nuova "terza clausola" della licenza BSD. Il team di FreeBSD ha semplificato ulteriormente la licenza rimuovendo anche la quarta clausola.

Leggibilità

Capire ciò che viene scritto in caratteri piccolissimi

La maggior parte delle persone che usa software Open Source – così come quelle che utilizzano altri tipi di software – non sono avvocati e non capiscono gli aspetti più "sottili" del software licensing. Tutto ciò, inoltre, è complicato dal fatto che oggi un software può essere distribuito in tutto il mondo nel giro di qualche ora e, quindi, la licenza necessita di essere compresa e applicabile in differenti ordinamenti giuridici. Le licenze BSD-style sono certamente avvantaggiate, in questo aspetto, proprio perché sono più

semplici rispetto alle altre. Le licenze Apache sono un po' più verbose e scritte in un linguaggio più complesso dal punto di vista legale, ma sono perfettamente comprensibili anche da chi non si occupa di diritto. Ciò è dovuto ai suoi termini e condizioni redatti in una forma abbastanza semplice. Benché sia più complessa, la GPLv2 conserva una certa semplicità e comprensibilità. La licenza LGPL è praticamente uguale alla GPL versione 2.1. Lo stesso non può dirsi per la versione 3 della GPL. In una certa misura ciò dipende dal

fatto che la FSF ha cercato di arginare diversi effetti non contemplati dalla GPLv2 ma ritenuti disastrosi dalla stessa FSF. Sul punto lasciamo l'ultima parola allo sviluppatore di SelectTor: "Vi chiederete perché ho adottato la GPLv2 invece dell'ultima versione? È semplice, io sono stupido e non posso incasinarmi la testa con la GPLv3. Secondo me la GPLv2 è scritta molto meglio ed è comprensibile dall'uomo medio, si adatta alle mie esigenze e non mi importa di quanti avvocati abbiano lavorato alla GPLv3".

Verdetto

- GPLv3 ★★★★★
- GPLv2 ★★★★★
- LGPL ★★★★★
- Apache ★★★★★
- BSD-style ★★★★★

» Anche se non sono un giurista mi piace capire le licenze che adotto.

GPLv2 ★★★★★

La versione 2 della GPL è una licenza di agevole lettura e ciò spiega, forse, la sua popolarità e diffusione. Diversamente dal copyleft, la GPLv2 stabilisce poche condizioni sull'utilizzo del codice. In effetti la sua unica condizione è che nessuno aggiunga ulteriori restrizioni! Ad esempio, non potrete limitare i campi nei quali il software viene utilizzato. Questo aspetto ha causato delle controversie in passato in quanto alcuni progetti intendevano introdurre delle restrizioni all'uso del software in ambiti militari. I creatori di GPU, un client Gnutella, hanno realizzato una nuova licenza, basata sulla GPL, ma che consente di applicare queste ulteriori restrizioni, rendendo, cioè, il software fruibile solo per finalità "pacifiche". A causa di queste modifiche, però, non è considerata compatibile con la GPL e per tale ragione non ha preso piede.



Free as in Freedom

GPLv3 ★★★★★

Ci sono più clausole nella GPLv3 che nelle altre licenze che abbiamo analizzato. Molte di tali clausole sono già trattate in altre parti di questo stesso articolo. Occupiamoci, qui, della clausola sui DRM (Digital Rights Management). Il DRM consiste in quello che potremmo definire un lucchetto digitale che viene imposto dal venditore sul prodotto digitale (come un video o un file musicale) per fare in modo che tale prodotto digitale sia fruibile attraverso un solo dispositivo. La GPLv3 sul punto prevede che "quando distribuisce un'opera tutelata da questa Licenza, rinunci a ogni pretesa di proibire l'elusione di efficaci misure tecnologiche nel limite in cui l'elusione si realizzi a causa del mero esercizio dei diritti, previsti da questa Licenza, sull'opera; e rinunci a ogni futura pretesa di porre dei limiti a operazioni o modifiche dell'opera come attacchi all'efficace misura tecnologica, contro i diritti degli utenti, tuoi o di terzi che impediscano l'elusione delle efficaci misure tecnologiche". Ciò significa che se provi a inserire un DRM all'interno di un codice sorgente rilasciato sotto GPLv3 non potrai impedire a nessuno di trovare dei sistemi per violare questo DRM. E se consideriamo che le persone solitamente violano i DRM per accedere al codice sorgente è altamente probabile che questo accada.



Free as in Freedom

LGPL ★★★★★

La licenza LGPL la ritroviamo in varie versioni. Le più diffuse sono la versione 2.1, che è sostanzialmente in linea con la GPLv2, e la versione 3 che si adatta meglio alla GPLv3. Benché sia simile a queste per molti aspetti, la LGPL differisce dalla GPL per quanto riguarda il discorso della compatibilità. Forse la cosa più strana che contraddistingue la LGPL è che ne viene sconsigliato l'uso dai suoi stessi creatori (FSF). La FSF riconosce che la LGPL abbia un senso (altrimenti non l'avrebbero certo creata) ma che debba essere utilizzata unicamente in certe circostanze. La LGPL, infatti, è stata ideata per fornire una licenza adatta per il linguaggio C, in modo che il software rilasciato con licenza LGPL possa essere linkato anche da software proprietario. E per questo motivo la LGPL dovrebbe essere utilizzata esclusivamente per incoraggiare la diffusione del Software Libero in quelle situazioni in cui, altrimenti, ciò non sarebbe potuto avvenire in considerazione del software proprietario che esegua il link al software rilasciato sotto licenza LGPL. A complicare ulteriormente le cose si mette anche il significato da attribuire alla "L" iniziale di LGPL: originariamente faceva riferimento a "library" (GPL adatta per le librerie), ma di recente si intende riferita al termine "lesser" (meno libera della GPL).

Compatibilità

Le licenze devono poter interagire senza scontrarsi tra loro?

Ci sono due aree in cui la compatibilità può rappresentare un problema: quando si decide di cambiare la licenza al software e quando si devono linkare due software. Il linking si ha quando un software fa riferimento a una libreria esterna. La GPLv2 e versioni superiori impedisce che un software che abbia licenza non-libera possa far riferimento a un software sotto GPL. Per questo motivo la GPL viene spesso definita "virale", ossia una licenza che tende ad assicurare che tutto il software sul sistema sia di tipo "libero". La FSF ha stilato un elenco di licenze

compatibili con la GPL (www.gnu.org/licenses/license-list.html). Tuttavia la FSF riconosce che questa natura virale della GPL tenda a rallentare o limitare la diffusione del Software Libero. A causa di questo "limite" della GPL alcuni decidevano di optare per librerie con licenze meno "restrittive" e, solitamente, non copyleft. Per far fronte a questo fenomeno è stata rilasciata la LGPL ("lesser" o "library" GPL). Il codice rilasciato sotto questa licenza conferisce le stesse libertà della GPL ma può essere linkato a codice proprietario. Le licenze di tipo BSD sono in genere

più compatibili e il codice rilasciato sotto i loro termini può linkare o, comunque, consente di ri-licenziare lo stesso codice sotto licenze più restrittive. L'eccezione a questo è dato dalla quarta clausola (vecchio stile) della licenza BSD. La clausola "advertising" la rende incompatibile con la GPL. A causa della mole di software rilasciata sotto licenza Apache, la FSF ha pensato bene di fare in modo che la GPLv3 fosse compatibile con questa (con la versione 2 della licenza Apache), sciogliendo così i dubbi interpretativi che derivavano dalla compatibilità tra GPLv2 e licenza Apache.

Verdetto

- GPLv3 ★★★★★
- GPLv2 ★★★★★
- LGPL ★★★★★
- Apache ★★★★★
- BSD-style ★★★★★

» Bisogna fare molta attenzione quando si combina del codice rilasciato sotto differenti licenze.

Brevetti

Non è solo il copyright a tenere il codice al sicuro

Negli ultimi anni la normativa sui brevetti è stato un terreno di scontro tra le più ricche società produttrici di software. I brevetti proteggono le idee piuttosto che le loro implementazioni pratiche. Prendiamo, ad esempio, il recente caso dello "Slide to unlock" di Apple che serve a sbloccare determinati dispositivi con sistema operativo iOS. Le norme sul copyright possono impedire a chiunque di copiare il codice sorgente di un software altrui per crearne un altro. Tuttavia Apple non intendeva proteggere unicamente il codice sorgente di "Slide to unlock" ma la stessa idea sottostante. Pertanto Apple ha richiesto e ottenuto un brevetto su "Slide to unlock" e, in tal modo, impedisce a chiunque altro di implementare una simile feature a prescindere dal fatto che sia stato copiato o meno il codice sorgente di "Slide to unlock" (nb: questo brevetto potrebbe essere ritenuto inefficace in quanto nel momento in cui scriviamo si celebra un procedimento per valutarne la validità). Tutto quanto premesso significa che una licenza software, pensata per proteggere un'opera a livello di copyright, non può offrire adeguata protezione contro la minaccia

dei brevetti. Le licenze della famiglia BSD non offrono alcuna protezione contro la minaccia dei brevetti. Un produttore di software potrebbe rilasciare del codice sotto una di queste licenze e tuttavia continuare a richiedere il pagamento delle royalty alle persone che usano il loro software. Con riferimento a questo ambito, la LGPL offre la stessa protezione della corrispondente versione della GPL. La clausola della GPLv2 relativa ai brevetti viene chiamata "libertà o morte". Ciò significa, semplicemente, che se vengono limitati i diritti degli utenti in base a un brevetto, allora non si avrà il permesso di usare il software. Una delle ragioni per aggiornare la GPL alla versione 3 è determinata dalla volontà di rendere più esplicita la sezione relativa ai brevetti software. Nella versione più recente della GPL, infatti, si rafforza la protezione del software avendo a mente il caso che portò all'accordo tra Novell e Microsoft. Grazie a questo accordo Microsoft, in cambio di denaro, offrì la garanzia di immunità per tutti i clienti di Novell (per tutte quelle ipotesi in cui Microsoft avrebbe potuto lamentare un suo brevetto software da parte di Novell). Un accordo simile è perfettamente lecito - o meglio non impedito - dalla licenza GPLv2, e per questo la FSF ha

introdotta una clausola nella GPLv3 che prevede che se taluno si accorda in modo analogo a quanto successo tra Novell e Microsoft allora l'accordo dovrà essere tale da estendere sua efficacia a tutti gli utenti del software licenziato sotto la GPLv3 e non limitarla solamente ai clienti dell'azienda da cui ricevono il software. La licenza GPLv3 offre un'eccellente tutela contro i pericoli che i brevetti impongono al software libero, ma raramente ha trovato applicazione in giudizio. Probabilmente la licenza che si evidenzia tra tutte in questo ambito è la licenza Apache. Ma non si evidenzia per motivi legali ma unicamente perché è la licenza più diffusa nell'ambiente con il più alto tasso di "litigiosità", ossia quello della telefonia mobile. Mentre le battaglie tra Apple e i vari produttori di cellulari Android sono ben note, non è dato rilevare di problemi legali tra i rivali produttori di telefoni Android, nonostante ciascuno di essi detenga grandi portafogli di brevetti. E ciò, in parte, è merito della sezione 3 della licenza Apache in cui si dice, molto semplicemente, che "se distribuisce del software nei termini di questa licenza devi anche concedere tutte le facoltà derivanti dai brevetti sullo stesso software". Occorre notare che la licenza Apache non contiene una clausola analoga a quella della GPLv3 e che ci sono stati già molti accordi tra produttori Android e Microsoft sui brevetti: accordi che la GPLv3 avrebbe impedito.

Verdetto

GPLv3
★★★★★
GPLv2
★★★★★
LGPL
★★★★★
Apache
★★★★★
BSD-style
★★★★★

» Molte licenze libere sono rimaste al passo con la minaccia dei brevetti.

Software, licenze e numeri

Una licenza è buona solo se lo è il suo software

Una licenza che protegge perfettamente gli utenti è perfettamente inutile se nessun utente vuole quel software. Diamo quindi uno sguardo a ciò che viene protetto dalle rispettive licenze. BlackDuck Software stila un elenco di software Open Source su <http://osrc.blackducksoftware.com/data/licenses>, e da questo elenco possiamo vedere quanto è diffusa ogni singola licenza in base alla diffusione del relativo software. Vediamo questa top-ten:

- 1 - GPLv2 (36%)
- 2 - Mit License (12%)
- 3 - Apache Licence (10%)
- 4 - GPL v3 (10%)
- 5 - BSD Licence (7%)
- 6 - Artistic Licence (6%)

7 - LGPL v2.1 (6%)
8 - LGPL v3 (2%)
9 - Code Project Open Licence (2%)
10 - Microsoft Public License (1%)
È bene evidenziare che in questo articolo abbiamo sempre considerato in modo unitario le licenze MIT e BSD in quanto sono assai simili. Naturalmente, come dice un famoso proverbio, ci sono bugie, dannate bugie e le statistiche. Questa lista non vuole in alcun modo evidenziare la qualità del software coperto da ogni licenza, o la sua diffusione, e per questo non può rappresentare il modo in cui i vari software vengono utilizzati. Tuttavia questo elenco rispecchia correttamente la nostra esperienza. Forse la cosa più sorprendente

è scoprire che la GPLv2 è ancora molto più popolare della successiva GPLv3. Ci sono almeno un paio di giustificazioni per questo: in parte la causa va ricercata in vecchi progetti che non hanno ancora effettuato la migrazione del codice alla nuova licenza per ragioni legali. Tuttavia, molti sviluppatori preferiscono semplicemente la versione precedente e ritengono che le novità introdotte nella GPLv3 abbiano limitato le libertà invece di estenderle.



Verdetto

GPLv3
★★★★★
GPLv2
★★★★★
LGPL
★★★★★
Apache
★★★★★
BSD-style
★★★★★

» Alcune volte si finisce con una gara di popolarità.

Licenze libere

Il verdetto

Abbiamo testato le licenze tenendo in considerazione la definizione di Software Libero data dalla FSF, e quindi non è sorprendente che vinca la licenza preferita dalla FSF. Tuttavia, questo non significa necessariamente che sia la migliore licenza. Ogni licenza Open Source, infatti, deve trovare un equilibrio tra ciò che viene "concesso" e ciò che viene "impedito" in nome della libertà degli utenti. Il fatto che la GPLv3 fatichi a prender piede al di fuori del progetto GNU dimostra che questa licenza impone restrizioni al fine di garantire un numero di libertà future ed eventuali maggiori rispetto a quelle cui molti sviluppatori sono abituati. Una definizione più pura di libertà dovrebbe prevedere le restrizioni tipiche del copyleft come imposizioni delle libertà essenziali per fare qualsiasi cosa con

il codice. Questo spirito è ben rappresentato nell'acronimo DWTFYWWI (Do What The F**k You Want With It - ossia fai quel * che ti pare con esso) tipico della licenza BSD-style, che contiene solo un singolo termine: "O. L'autore consente a tutti di fare con il software qualsiasi dannata cosa desiderino, di qualsiasi dannata cosa si tratti". Questo è molto simile a quello che viene chiamato "pubblico dominio". Le altre licenze in stile BSD, in genere, aggiungono poche restrizioni: di solito richiedono solamente l'attribuzione della paternità agli autori. Le suddette due licenze (GPL e BSD) rappresentano gli estremi della filosofia Open Source. Tra le due licenze si trova la vasta area delle ragioni morali in cui stanno la maggior parte degli sviluppatori. Linus Torvalds, per esempio, aveva

da subito mostrato la sua preferenza per la GPLv2, e non ha intenzione di migrare il kernel alla versione più recente (GPLv3). Molte persone concordano con Linus, e la GPLv2 resta al primo posto nell'elenco dell'Open Source Resource Center. Questa situazione difficilmente cambierà nel prossimo futuro, in quanto la GPLv2 raccoglie i favori della maggior parte dei sostenitori del Software Libero: copyleft con qualche piccola restrizione supplementare. La licenza Apache sembra destinata a mantenere il suo predominio sul versante "commerciale" dell'Open Source. Molti contabili aziendali hanno paura del copyleft, e questo è potenzialmente in grado di limitare l'adesione al movimento del Software Libero da parte delle aziende. Con la licenza Apache, tali preoccupazioni non sono un problema, ma le minacce di brevetto sono minimizzate.



L'Open Source Initiative pubblica un elenco di 69 licenze Open Source

1° GPLv3 ★★★★★

Web: www.gnu.org/licenses/gpl.html

» Difende vigorosamente le libertà per gli utenti ma è difficile da comprendere.

4° Apache ★★★★★

Web: www.apache.org/licenses/LICENSE-2.0

» Ha clausole adatte al settore commerciale e tese a prevenire la guerra di brevetti.

2° GPLv2 ★★★★★

Web: www.gnu.org/licenses/gpl-2.0.html

» Incarna lo spirito del copyleft e raccoglie i favori di un gran numero di software.

5° LGPL ★★★★★

Web: www.gnu.org/licenses/lgpl.html

» Utile in alcune circostanze, ma si può trovare di meglio.

3° BSD-style ★★★★★

Web: en.wikipedia.org/wiki/BSD_licenses

» In poche parole, vera libertà!

A voi la parola

» Avete mai usato una di queste licenze? Cosa ne pensate? Volete saperne di più? Scriveteci a redazione@linuxpro.it.

Considerate anche...

Vi è una vasta gamma di licenze che è possibile utilizzare. Molti ritengono che gli utenti restino confusi dal fatto che in circolazione vi siano troppe licenze e che queste si somiglino tutte. Pensando a questo, di solito, è meglio usare una licenza più "rodada" rispetto a una nuova e ancora sconosciuta o, peggio, dalle clausole incomprensibili. Vogliamo segnalare, tuttavia, altre due licenze che vengono utilizzate in grandi progetti: la **Mozilla Public License (MPL)** e la **Artistic License (Perl)**. In generale, queste ultime dovrebbero essere utilizzate solo quando si abbia a che fare con i progetti collegati. Se invece siete alla ricerca di qualcosa di diverso con cui concedere in licenza il vostro codice, potreste prendere in considerazione le licenze **Creative Commons (CC)**. Queste hanno una serie di clausole che

si possono combinare per soddisfare il vostro modo di vedere. La licenza di base - Attribution - richiede unicamente che le persone che ricevono la vostra opera debbano riconoscerne la paternità. Inoltre a questa clausola, è possibile aggiungere quella *ShareAlike* (più o meno equivalente al copyleft), quella *NonCommercial*, e quella *NonDerivs* (il che significa che si può solo distribuire la tua opera nella versione non modificata). Tra queste clausole dovrete trovare la giusta combinazione per soddisfare i vostri gusti. Le licenze Creative Commons sono pensate per essere facilmente comprensibili e hanno delle icone che possono essere applicate alle opere (visive) in modo da non dover aggiungere necessariamente tutto il "legal code" ossia il testo completo della licenza CC desiderata. **LXP**