

Software e libertà

1.1	Software libero	13
1.2	Software non libero	15
1.3	Software commerciale	15
1.4	Dal software all'arte in generale: Creative Commons	15
1.5	Origini del software libero	18
1.5.1	BSD	18
1.5.2	GNU	19
1.5.3	Minix	19
1.5.4	Linux	19
1.5.5	Debian	20
1.5.6	Internet Archive	20
1.5.7	Open Source	20
1.5.8	Creative Commons	20
1.5.9	Open Invention Network	21
1.5.10	Standard Unix	21
1.6	Riferimenti	21

Il software è un codice che è trattato dalle leggi sul diritto di autore, in maniera simile a quanto avviene per le opere letterarie. Tradizionalmente, il contratto che regola l'uso del software è la *licenza* ed è sempre importante conoscere i termini di questo accordo per il software con cui si intende avere a che fare.

La legge sul diritto di autore stabilisce già quali sono i diritti di chi produce l'opera e di chi la utilizza, ma nell'ambito del software si è introdotto nella pratica un contratto che non viene firmato (la licenza), il cui scopo è spesso quello di limitare ulteriormente i diritti di chi ne fruisce.

Il software ha un proprietario (salvo il caso del software di dominio pubblico che viene descritto in seguito) che è tale in quanto «detiene i diritti di 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*.

L'utilizzo del software può essere concesso gratuitamente o a pagamento, per le operazioni stabilite nel contratto di licenza, o in sua mancanza per quanto stabilito dalla legge. Il pagamento per l'«acquisto» di software, non si riferisce all'acquisizione dei diritti di autore, ma solo delle facoltà stabilite dalla legge, ovvero da quanto indicato nella licenza.

La natura del software è tale per cui questo è composto da due parti fondamentali: il codice sorgente e il codice eseguibile. Il primo è intelligibile (comprensibile), il secondo è adatto all'esecuzione e non è 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.

Figura 1.1. Compilazione.



In molti paesi esiste anche la possibilità di brevettare algoritmi e altri concetti riconducibili al software. Il brevetto impedisce così l'uso

dell'algoritmo o dell'idea registrata, se non dopo un'autorizzazione esplicita da parte del detentore di questo tipo di diritto.

1.1 Software libero

Il **software libero** è software che fornisce il permesso per chiunque di utilizzarlo, copiarlo e distribuirlo, in forma originale, o anche dopo averlo modificato, sia gratuitamente, sia a pagamento. Il software libero può essere tale solo se viene messo a disposizione assieme al codice sorgente, per cui, a questo proposito, qualcuno ha detto: «se non è sorgente, non è software» (*if it's not source, it's not software*).

È importante sottolineare che la «libertà» del software libero non sta tanto nel prezzo, che eventualmente può anche essere richiesto per il servizio di chi ne distribuisce le copie, ma nella possibilità di usarlo senza vincoli, di copiarlo come e quanto si vuole, di poterne distribuire le copie, di poterlo modificare e di poterne distribuire anche le copie modificate.

In alcuni ambienti si preferisce utilizzare la definizione «Open Source» per fare riferimento al software libero nei termini che sono stati descritti, per evitare ambiguità nella lingua inglese. Sotto questo aspetto, nella lingua italiana, come in molte altre lingue, è più appropriato l'uso della definizione «software libero».

Si distinguono quattro punti fondamentali, necessari perché il software possa essere considerato «libero»:

Livello	Descrizione
libertà 0	la libertà di eseguire il programma per qualunque scopo;
libertà 1	la libertà di studiare come funziona il programma e di adattarlo alle proprie esigenze (in tal caso, deve essere disponibile il sorgente);
libertà 2	la libertà di ridistribuire copie del programma;
libertà 3	la libertà di migliorare il programma e di distribuire tali miglioramenti (anche per questo è necessario disporre dei sorgenti).

Il software che non può essere commercializzato, pur soddisfacendo i punti elencati qui, viene considerato «software semi-libero», come viene chiarito in seguito.

Vale la pena di elencare alcune definizioni riferite al software libero.

• software di dominio pubblico

Il software di dominio pubblico è software senza copyright. Di per sé, questo tipo di software è libero, solo che, mancando chi può difenderlo, qualcuno può riuscire ad accamparvi dei diritti; pertanto, alcune copie, o varianti di questo software possono non essere più libere.

• software protetto da copyleft

La parola *copyleft* («permesso d'autore»), a cui a volte si associa il simbolo ©, è un'invenzione e 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 GNU-GPL (appendice A).

• software libero non protetto da copyleft

Il software libero non è necessariamente di tipo copyleft e ciò accade quando la licenza non vieta espressamente l'aggiunta di restrizioni da parte di chi lo ridistribuisce. Quando si utilizza software di questo tipo, non è possibile generalizzare: occorre accertarsi dei termini del contratto che riguarda la copia particolare della quale si è venuti in possesso.

• software GPL

La licenza GNU-GPL è l'esempio più importante di licenza che protegge il software libero con il copyleft. Quando si parla di

«software GPL» si intende fare riferimento a software protetto con la licenza pubblica GNU-GPL.

1.2 Software non libero

Il software non è libero tutte le volte che non sono soddisfatti tutti 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 software non libero.

• software semi-libero

Il software semi-libero è software che 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 proprietario

Il **software proprietario** è quel software che non è né libero, né semi-libero. 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.

• software freeware

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

• software shareware

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

1.3 Software commerciale

In base alle classificazioni viste in questo capitolo, il **software commerciale** è tale solo in quanto viene venduto per profitto. Lo sviluppo e la diffusione del software libero dipendono anche dalla possibilità di venderne delle copie, originali o modificate, per trarne profitto. Pertanto, il software che pur offrendo le quattro libertà fondamentali, non consente la commercializzazione per trarne profitto, non viene considerato «libero» in modo completo. In questo senso, è importante evitare di confondere il software proprietario con il software commerciale, perché non sono la stessa cosa.

Si osservi che può esistere anche del software non-libero, che non è nemmeno commerciale.

1.4 Dal software all'arte in generale: Creative Commons

I concetti di «libertà» derivanti dal mondo del software, si trasferiscono anche ad altre forme di arte sottoposta al diritto di autore, soprattutto quando queste forme di arte si esprimono in forma elettronica. Tuttavia, il software si trova in un contesto particolare rispetto ad altre forme di espressione artistica, sia perché è previsto in molti paesi il segreto industriale e il brevetto, sia perché il software può sopravvivere all'evoluzione tecnologica solo se è possibile aggiornarlo nel tempo. Pertanto, la difesa del software libero è più impegnativa rispetto al mantenimento della libera disponibilità di altre forme di espressione artistica.

Benché la questione sia controversa, è evidente che espressioni artistiche come un romanzo, un disegno, una canzone, possono considerarsi «libere» anche se non viene concessa la modifica e in certi casi anche se non viene messo a disposizione ciò che rappresenta il «sorgente» elettronico di tali lavori. Tanto per fare un esempio, la realizzazione di un libro «libero» secondo le quattro libertà del software, non consentirebbe di inserire delle citazioni, perché, secondo le convenzioni comuni, non sarebbero poi modificabili; inoltre, un libro che non disponga di un sorgente, potrebbe essere ricopiato, anche se con un certo dispendio di risorse.

Esistono diversi tentativi di risolvere il problema, compatibilmente con il concetto di copyleft, soprattutto per quello che riguarda la produzione di documenti; inoltre c'è anche chi, pregevolmente, tenta di trovare una formula che possa andare bene per ogni tipo di espressione artistica, come nel caso della licenza *Licence Art Libre* (ovvero *Free Art License* e tante altre traduzioni) <http://artlibre.org>.

Creative Commons (<http://creativecommons.org>) cerca di trasferire alcuni principi del software libero a opere artistiche di tipo differente, attraverso la redazione di un insieme di licenze con un diverso grado di «libertà», ovvero di restrizioni legali. Per facilitare la scelta e la comprensione delle diverse licenze proposte, Creative Commons utilizza un sistema iconografico. Questo aspetto diventa molto importante, dal momento che lo scopo dell'organizzazione è quello di favorire la diffusione consapevole di opere di ogni tipo attraverso la rete, dove spesso si presta più attenzione alle immagini, rispetto al testo scritto.

Tabella 1.3. Obblighi o facoltà principali espressi in forma iconografica.

Simbolo	Sigla	Significato
	<i>attribution</i>	Gli autori originari devono essere riconosciuti.
	<i>non-commercial</i>	Il lavoro non può essere sfruttato commercialmente.
	<i>no derivatives works</i>	Il lavoro deve rimanere intatto.
	<i>share alike</i>	La licenza deve rimanere la stessa (questa condizione ha significato solo per un lavoro che può essere modificato).
	<i>source</i>	Il sorgente deve essere preservato.

È importante osservare che, dal punto di vista della filosofia delle libertà del software, non tutte le licenze elaborate da Creative Commons sono «libere». Evidentemente, ciò che non può essere modificato non può essere «libero»; inoltre, ciò che non può essere commercializzato, non può essere considerato del tutto libero. Tuttavia, in considerazione del fatto che le esigenze che si sentono nel campo del software non sono le stesse di ogni altro tipo di espressione artistica, la presenza di combinazioni «non libere» tra le licenze di Creative Commons, non sminuisce l'importanza del lavoro di tale organizzazione.

Mentre l'approvazione di OSI (*Open Source initiative* <http://opensource.org>) dà la garanzia di avere a che fare con del software libero, quando si fa riferimento a una licenza approvata da Creative Commons, senza specificarne gli attributi, si sa solo che **dovrebbe** trattarsi di un'opera ridistribuibile e gratuita almeno per l'uso personale, non commerciale.

Nelle licenze Creative Commons si fa riferimento a un concetto, definito come *share alike*, con il quale si richiede che l'opera modificata mantenga la stessa licenza di quella originale. Si può osservare che il simbolo usato da Creative Commons per rappresentare questo vincolo è molto simile a quello usato nel software libero per esprimere il copyleft: ©. Tuttavia va osservato che il copyleft è un sottoinsieme di quanto intende il concetto di *share alike*.

Tabella 1.4. Licenze definite da Creative Commons.

Definizione	Iconografia	Descrizione
<i>attribution</i>		Si richiede soltanto che gli autori originari siano riconosciuti.
<i>attribution, no derivatives work</i>		Si richiede che gli autori originari siano riconosciuti e che il lavoro rimanga intatto.
<i>attribution, non-commercial, no derivatives work</i>		Si richiede che gli autori originari siano riconosciuti, che il lavoro non venga sfruttato commercialmente e che rimanga intatto.
<i>attribution, non-commercial</i>		Si richiede che gli autori originari siano riconosciuti e che il lavoro non venga sfruttato commercialmente.
<i>attribution, non-commercial, share alike</i>		Si richiede che gli autori originari siano riconosciuti, che il lavoro non venga sfruttato commercialmente e che la licenza rimanga la stessa.
<i>attribution, share alike</i>		Si richiede che gli autori originari siano riconosciuti e che la licenza rimanga la stessa.

Molti siti che consentono la pubblicazione di opere artistiche accettano esclusivamente un insieme di licenze approvate da Creative Commons, pertanto è bene sapere risalire alle licenze di proprio interesse attraverso la catalogazione e il riferimento di Creative Commons. Nelle tabelle successive si vedono le licenze principali di Creative Commons e il riferimento alla dichiarazione di dominio pubblico, assieme alle due licenze più importanti di Free Software Foundation.

Tabella 1.5. Licenze definite attraverso un URI, secondo Creative Commons.

Definizione	URI
<i>attribution</i>	http://creativecommons.org/licenses/by/3.0/
<i>attribution, no derivatives work</i>	http://creativecommons.org/licenses/by-nd/3.0/
<i>attribution, non-commercial, no derivatives work</i>	http://creativecommons.org/licenses/by-nc-nd/3.0/
<i>attribution, non-commercial</i>	http://creativecommons.org/licenses/by-nc/3.0/
<i>attribution, non-commercial, share alike</i>	http://creativecommons.org/licenses/by-nc-sa/3.0/
<i>attribution, share alike</i>	http://creativecommons.org/licenses/by-sa/3.0/

Creative Commons suggerisce l'uso di un'iconografia semplificata per dichiarare le caratteristiche di massima della licenza utilizzata. In passato, tutte le licenze scritte da Creative Commons, si associavano a un solo tipo di icona, dove si poteva solo intendere che non tutti i diritti erano riservati: . Attualmente l'iconografia è più chiara, come descritto nella tabella successiva.

Tabella 1.6. Icone suggerite da Creative Commons per dichiarare in linea di massima il genere di licenza utilizzato. Non tutte le licenze approvate dal Creative Commons appaiono in questa tabella.

Licenza	Icona associata	Licenza	Icona associata
<i>attribution, no derivatives work</i>		<i>attribution, no derivatives work</i>	
<i>attribution, non-commercial, no derivatives work</i>		<i>attribution, non-commercial</i>	

Licenza	Icona associata	Licenza	Icona associata
attribution, non- commercial, share alike		attribution, share alike	

All'interno di un file HTML (o XHTML) è possibile inserire del codice che consente a un motore di ricerca o a un navigatore di classificare il contenuto, in linea di massima, dal punto di vista legale, riguardo alle facoltà concesse o meno a chi ne deve fruirne.

1.5 Origini del software libero

« Il «software libero» è un concetto semplice e intuitivo, ma il percorso attraverso cui si conquista una libertà è sempre difficile. Se oggi questo tipo di software rappresenta concretamente una scelta possibile, lo si deve all'azione di persone che con impegno hanno agito, **legalmente**, verso il raggiungimento di questo obiettivo. Confondere invece la facoltà tecnica di copiare ciò che si vuole, soprattutto ciò che la legge vieta, non è libertà, ma dipendenza dallo stesso sistema che si crede di combattere.

1.5.1 BSD

« Negli anni 1970, i primi utenti di UNIX sono state le università, a cui in particolare questo sistema operativo è stato fornito a costo contenuto, con i sorgenti, ma senza alcun tipo di assistenza, né alcuna garanzia. Proprio questa assenza di sostegno da parte della casa che lo aveva sviluppato originariamente, ha stimolato la cooperazione tra gli utenti competenti, in pratica tra le università. Il maggior fermento intorno a UNIX si è concentrato presso l'università della California a Berkeley, dove a partire dal 1978 si è cominciato a distribuire una variante di questo sistema operativo, con il nome BSD, ovvero *Berkeley software distribution*. Per questo software è nata una licenza d'uso che rimane il progenitore della filosofia del software libero: la licenza BSD. Per molto tempo, la variante BSD di UNIX è rimasta relegata all'ambito universitario o a quello delle aziende che avevano acquistato i diritti per utilizzare il codice sorgente dello UNIX originale. Ciò fino a quando si è sentita l'esigenza di ripulire lo Unix BSD dal codice proprietario.

Il risultato iniziale è stato 386BSD, pubblicato nel 1992 con la versione 0.1. Tuttavia, questa edizione libera dello Unix BSD non ha avuto vita facile, dal momento che da quel punto sono iniziate delle contese giudiziarie sulla proprietà di alcune porzioni di codice ritenute libere (a torto o a ragione che fosse). Dai problemi di 386BSD che hanno causato la sua eliminazione dalla distribuzione pubblica, si sono sviluppati altri progetti indipendenti per ottenere, finalmente, un sistema BSD libero. Il primo di questi è stato nominato NetBSD, al quale si è aggiunto subito dopo FreeBSD e più tardi anche OpenBSD. Tuttavia, i problemi legali non sono terminati con 386BSD: la variante denominata FreeBSD è stata «libera» solo all'inizio del 1995 con la versione 2.0. Il pezzo seguente è tratto da *A Brief History of FreeBSD* di Jordan K. Hubbard, marzo 1998.

The first CDROM (and general net-wide) distribution was FreeBSD 1.0, released in December of 1993. This was based on the 4.3BSD-Lite ("Net/2") tape from U.C. Berkeley, with many components also provided by 386BSD and the Free Software Foundation. It was a fairly reasonable success for a first offering, and we followed it with the highly successful FreeBSD 1.1 release in May of 1994.

Around this time, some rather unexpected storm clouds formed on the horizon as Novell and U.C. Berkeley settled their long-running lawsuit over the legal status of the Berkeley Net/2 tape. A condition of that settlement was U.C. Berkeley's concession that large parts of Net/2 were "encumbered" code and the property of Novell, who had in turn acquired it from AT&T some time previously. What Berkeley got in return was Novell's "blessing" that the 4.4BSD-Lite release, when it was finally released, would be declared unencumbered and all existing Net/2 users would be strongly encouraged to switch. This included FreeBSD, and the project was given until the

end of July 1994 to stop shipping its own Net/2 based product. Under the terms of that agreement, the project was allowed one last release before the deadline, that release being FreeBSD 1.1.5.1.

FreeBSD then set about the arduous task of literally re-inventing itself from a completely new and rather incomplete set of 4.4BSD-Lite bits. The "Lite" releases were light in part because Berkeley's CSRG had removed large chunks of code required for actually constructing a bootable running system (due to various legal requirements) and the fact that the Intel port of 4.4 was highly incomplete. It took the project until December of 1994 to make this transition, and in January of 1995 it released FreeBSD 2.0 to the net and on CDROM. Despite being still more than a little rough around the edges, the release was a significant success and was followed by the more robust and easier to install FreeBSD 2.0.5 release in June of 1995.

« Allo stato attuale, le tre varianti *BSD sono tutte riconducibili a BSD 4.4-Lite, dove le differenze più importanti riguardano le piattaforme hardware in cui possono essere installate e l'origine della distribuzione.

1.5.2 GNU

« Nel 1985, Richard Stallman fonda 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 fatte. Queste idee filosofiche si sono tradotte nella redazione di un contratto di licenza d'uso, la General Public License (appendice A), studiato appositamente per proteggere il software libero in modo da non poter essere accaparrato da chi poi potrebbe impedirne la diffusione libera. Per questo motivo, oggi, il copyright di software protetto in questo modo, viene definito *copyleft*.

Il software libero richiede delle basi, prima di tutto il sistema operativo. In questo senso, l'intento pratico di Richard Stallman è stato quello di realizzare, con l'aiuto di volontari, un sistema operativo completo. Con questo obiettivo è nato il progetto GNU (*Gnu's not Unix*), puntando inizialmente alla realizzazione di un compilatore C, quindi alla costruzione di programmi di servizio, soprattutto quelli standard di un sistema Unix tipico, lasciando per un momento successivo lo sviluppo di un kernel.¹ Così facendo, il progetto GNU ha dato vita subito a una grande quantità di software utilizzabile sulla maggior parte delle piattaforme Unix, indirizzando implicitamente il software libero nella direzione dei sistemi di questo tipo.

1.5.3 Minix

« Alla fine degli anni 1980, il professor Andrew S. Tanenbaum sviluppa Minix, un sistema operativo Unix per elaboratori x86-16,² realizzato specificamente per uso didattico. Era sufficiente acquistare il libro a cui era abbinato e si otteneva un sistema completo di sorgenti. Tuttavia, Minix aveva un problema: poteva essere usato, distribuito e modificato, solo per fini didattici.

I diritti di questo sistema operativo sono stati ceduti inizialmente alla casa editrice del libro con il quale questo veniva diffuso; tuttavia, nell'anno 2000, Andrew S. Tanenbaum ha ottenuto dalla casa editrice di poter distribuire il sistema Minix con una licenza simile a quella di BSD.

1.5.4 Linux

« Linux nasce all'inizio degli anni 1990 come un progetto personale di studio delle funzionalità di multiprogrammazione dei microprocessori x86-32 da parte di Linus Torvalds, all'epoca uno studente all'università di Helsinki, in Finlandia. Inizialmente il lavoro di Linus Torvalds si basava su un sistema Minix, riscrivendo il kernel e adattando successivamente il compilatore e i programmi sviluppati dal progetto GNU. L'idea di Linus Torvalds era quella di realizzare «a better Minix than Minix», ovvero, voleva costruire qualcosa di meglio del sistema Minix.

Dopo molto lavoro, Linus Torvalds arriva a un sistema minimo e soprattutto autonomo da Minix. Il 5 ottobre 1991 invia il messaggio seguente al gruppo di discussione *comp.os.minix*.

Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.

As I mentioned a month ago, I'm working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02...but I've successfully run bash, gcc, gnu-make, gnu-sed, compress, etc. under it.

Linux ha coinvolto in breve tempo un numero molto grande di persone, unite dal fatto che si trattava di un progetto libero da qualunque restrizione legale al suo utilizzo, alla sua diffusione, alla possibilità di modificarlo ecc. In pratica, la fortuna di Linux rispetto a Minix, è stata quella di avere scelto subito la licenza GNU-GPL (appendice A), quella che ancora oggi rappresenta la difesa ideale per il software che viene scritto perché sia a disposizione di tutti. Tuttavia, per il successo del kernel Linux è stato determinante il lavoro parallelo del progetto GNU, il quale ha dato al primo tutto quello che serve per un sistema operativo completo: GNU/Linux appunto.

1.5.5 Debian

Nel 1993 Ian Murdock fonda il progetto Debian, con lo scopo di realizzare una distribuzione GNU/Linux libera e accurata. Successivamente, Bruce Perens stila la bozza di ciò che attualmente costituisce le «linee guida», per stabilire cosa sia da intendere «software libero» e cosa invece non rientri in questa categoria, ai fini della distribuzione stessa. Tali linee guida sono note con la sigla DFSG; successivamente, queste hanno dato origine alla definizione di Open Source.

Il progetto Debian ha curato distribuzioni GNU/Linux per le architetture più comuni e anche distribuzioni con combinazioni differenti di kernel e software applicativo.

1.5.6 Internet Archive

Nel 1996 viene fondato Internet Archive (<http://archive.org>), con lo scopo di creare una libreria digitale di tutte le opere espresse in tale forma, purché possano essere ripubblicate. Tra le altre cose, presso Internet Archive è possibile pubblicare ogni tipo di lavoro in forma elettronica, purché se ne conceda l'uso o purché siano venuti meno i diritti di autore relativi; inoltre è disponibile un archivio storico dei siti individuati e scanditi nell'ambito dell'intera rete Internet, con esclusione di quelli che, attraverso un file `'robot.txt'` o con richiesta formale, si siano esclusi espressamente.

1.5.7 Open Source

Sulla scia delle linee guida del progetto Debian, nel 1998 nasce la definizione Open Source, a identificare i principi secondo cui il software può essere ritenuto «libero», stilando e aggiornando un elenco delle licenze certificate (<http://www.opensource.org>).

Dopo la definizione Open Source è apparso l'uso dell'acronimo OSS, per *open source software*, a cui si sono affiancati anche FOSS, per *free open source software*, e FLOSS, per *free-liberal open source software*, ma sostanzialmente equivalenti.

1.5.8 Creative Commons

Come già descritto nella sezione 1.4, nel 2001 viene fondato Creative Commons (<http://creativecommons.org>) con lo scopo di trasferire alcuni principi del software libero a opere artistiche di tipo differente. Il lavoro importante di Creative Commons sta nello studio

delle problematiche legali che riguardano le opere artistiche in generale, classificando diversi tipi di esigenze da parte degli autori, le quali si traducono in pratica in diversi tipi di licenza, ognuna con un diverso grado di «libertà».

1.5.9 Open Invention Network

Alla fine del 2005, un gruppo di aziende importanti che offrono prodotti e servizi basati sul kernel Linux, fondano Open Invention Network (OIN), con lo scopo di raccogliere un insieme di brevetti, utili a difendere i sistemi basati su kernel Linux. In pratica, viene concesso l'uso delle idee alla base di tali brevetti, purché non si attacchino in giudizio i sistemi operativi basati su kernel Linux, per l'uso di idee contenute in altri brevetti.

1.5.10 Standard Unix

Attorno ai sistemi operativi che si rifanno al modello di Unix, quello da cui ha origine il concetto di software libero, si sono definiti due standard fondamentali: il linguaggio C (ISO/IEC 9899) e la definizione del sistema «Unix» standard, noto come POSIX negli anni 1990 (*Portable operating system interface*) e poi come SUS negli anni 2000 (*Single unix specification*). Per quanto riguarda Unix si distinguono dei livelli, definiti come POSIX.1, POSIX.2,... SUSv1, SUSv2,... che sono raccolti all'interno di standard IEEE e ISO; in particolare IEEE 1003.1 e ISO/IEC 9945. Le definizioni di POSIX.1-2001 e SUSv3 sono allineate.

La standardizzazione del linguaggio C e dei sistemi Unix è un veicolo importante per la consapevolezza del valore dei sistemi Unix in qualità di sistemi liberi.

Riquadro 1.7. Definizioni alternative di «software libero».

Definizione	Significato letterale
OSS	<i>Open Source software</i>
OSS/FS	<i>Open Source software/Free software</i>
F/OSS	<i>Free/Open Source software</i>
FOSS	<i>Free Open Source software</i>
FLOSS	<i>Free Liberal Open Source software</i>
<i>software libre</i>	
<i>unfettered software</i>	

1.6 Riferimenti

- Stop!, *Guida per scuole e famiglie al diritto d'autore*, <http://stop.zona-m.net/it/2010/03/guida-per-scuole-e-famiglie-al-diritto-dautore/>
- Wikipedia, *Free software*, http://en.wikipedia.org/wiki/Free_software
- Wikipedia, *Alternative terms for free software*, <http://en.wikipedia.org/wiki/FOSS>
- *Categories of Free and Non-Free Software*, <http://www.gnu.org/philosophy/categories.html>
- *Various Licenses and Comments about Them*, <http://www.gnu.org/philosophy/license-list.html>
- *The Free Software Definition*, <http://www.gnu.org/philosophy/free-sw.html>
- *Some Confusing or Loaded Words and Phrases that are Worth Avoiding*, <http://www.gnu.org/philosophy/words-to-avoid.html>
- Eben Moglen, *Enforcing the GNU GPL*, <http://www.gnu.org/philosophy/enforcing-gpl.html>
- *Frequently Asked Questions about the GNU GPL*, <http://www.gnu.org/licenses/gpl-faq.html>
- *Debian Social Contract, The Debian Free Software Guidelines (DFSG)*, http://www.debian.org/social_contract
- *Debian Free Software Guidelines (DFSG) FAQ*, <http://people.debian.org/~bap/dfsg-faq.html>
- *Artlibre.org*, <http://artlibre.org>

- *Patent Commons*, <http://patentcommons.org>
- *Creative Commons*, <http://creativecommons.org/>
- *Creative Commons Licenses*, <http://creativecommons.org/licenses/>, <http://creativecommons.org/choose/>
- *Internet Archive*, <http://archive.org>
- *Wikipedia, History of free software*, http://en.wikipedia.org/wiki/History_of_free_software
- *John Holden, YAPP - Yet another PDP-11 Page*, <http://www.psych.usyd.edu.au/pdp-11/>
- *Unix Programmer's Manual*, prima edizione, 1971, <http://cm.bell-labs.com/cm/cs/who/dmr/1stEdman.html>
- *Unix Seventh Edition Manual*, 1979, <http://plan9.bell-labs.com/7thEdMan/index.html>
- *FreeBSD, NetBSD, OpenBSD*, <http://www.freebsd.org/>, <http://www.netbsd.org/>, <http://www.openbsd.org/>
- *The Open Source Page*, <http://www.opensource.org/>
- *The GNU Project and the Free Software Foundation (FSF)*, <http://www.gnu.org/>, <http://www.fsf.org/>
- *Stephen White, A Brief History of Computing*, <http://trillian.randomstuff.org.uk/~stephen/history/timeline.html>
- *Éric Lévénez, Unix History*, <http://www.levenez.com/unix/>
- *Andy Tanenbaum, 20 May 2004, Some Notes on the "Who wrote Linux" Kerfuffle*, <http://www.cs.vu.nl/~ast/brown/>
- *Andy Tanenbaum, 21 May 2004, Ken Brown's Motivation*, <http://www.cs.vu.nl/~ast/brown/followup/>
- *Andy Tanenbaum, 20 May 2004, Comparison of Linux code with MINIX code*, <http://www.cs.vu.nl/~ast/brown/codecomparison/>
- *Andy Tanenbaum, 6 June 2004, Rebuttal to Ken Brown*, <http://www.cs.vu.nl/~ast/brown/rebuttal/>
- *Wikipedia, SCO v. IBM*, http://en.wikipedia.org/wiki/SCO_v._IBM
- *Open Invention Network*, <http://openinventionnetwork.com>
- *Pagina di manuale standards(7)*, <http://linux.die.net/man/7/standards>
- *Brian W. Kernigham, Dennis M. Ritchie, The C programming language*, prima edizione, Prentice-Hall 1978,³ <http://cm.bell-labs.com/cm/cs/cbook/>
- *ISO/IEC 9899:2011, Information technology -- Programming languages -- C*, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57853, <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf>
- *The Open Group, Single UNIX Specification*, http://www.unix.org/what_is_unix/single_unix_specification.html
- *The Open Group, The Open Group Base Specifications Issue 6*, IEEE Std 1003.1, 2004 Edition, <http://pubs.opengroup.org/onlinepubs/009695399/download/>
- *Wikipedia, Unix*, <http://en.wikipedia.org/wiki/Unix>
- *Dennis M. Ritchie, Dennis M. Ritchie home page*, <http://cm.bell-labs.com/cm/cs/who/dmr/index.html>

¹ Nel 1990 inizia lo sviluppo del kernel Hurd e intorno al 2000 inizia la distribuzione del sistema GNU/Hurd, ovvero del sistema GNU basato su kernel Hurd.

² Successivamente Minix 2 viene distribuito in due versioni, una per architettura x86-16 e l'altra per x86-32, mentre Minix 3 abbandona l'architettura x86-16.

³ Questo è il primo documento che definisce il linguaggio C, per ciò che oggi è noto come «K&R C», ovvero il linguaggio C di Kernigham e Ritchie. Di questo libro esiste una seconda edizione, del 1988, rivista secondo le convenzioni in corso di definizione dallo standard ANSI C.