

Che cos'è il Software Libero ?



L'espressione **Software libero** si riferisce alla libertà dell'utente di usare e migliorare il software. Più precisamente, può essere riassunto in quattro libertà fondamentali:

- Libertà di eseguire il programma, per qualsiasi scopo
- Libertà di studiare come funziona il programma e adattarlo alle proprie necessità. L'accesso al codice sorgente ne è un prerequisito.
- Libertà di ridistribuire copie in modo da aiutare il prossimo.
- Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente ne è un prerequisito.

Questa definizione è stata formulata agli inizi degli anni 80, quando la maggior parte del software cominciò a passare da libero a proprietario, ponendo un pesante freno alla collaborazione che caratterizzava il lavoro di gran parte dei programmatori e dei sistemisti; fino ad allora praticamente tutto il software era stato libero, ma ora si andavano moltiplicando contratti che tra le clausole imponevano agli sviluppatori di mantenere il segreto sui metodi utilizzati per lo sviluppo di sistemi e applicazioni. In questo modo le aziende si assicuravano il controllo dei propri clienti che, senza più poter vedere e modificare il [codice sorgente] del software, non potevano più adattarlo alle loro esigenze ma dovevano chiedere alle aziende di farlo per loro. Nel 1983 [Richard Stallman](#) fondò il Progetto [GNU](#) con l'intenzione di creare un [sistema operativo](#) completamente libero. Grazie alla collaborazione di molti sviluppatori volontari, all'uso di [Internet](#) per la coordinazione del progetto e al [kernel Linux](#) di [Linus Torvalds](#), nel 1991 nacque [GNU Linux](#).

La maggior parte del software libero viene distribuito con licenza [GNU GPL](#), scritta da Richard Stallman e Eben Moglen per garantire legalmente a tutti gli utenti le quattro libertà fondamentali.

In inglese la parola *free* viene usata sia in *freeware* (software gratuito) sia in *free software* (software libero). Sebbene gran parte del software libero sia anche gratuito (nel senso che non vengono chiesti soldi per ottenerne una copia) la definizione di software libero non pone alcun vincolo al suo prezzo: la vendita del software è una componente importante per il suo sviluppo e nulla vieta a chi lo distribuisce di chiedere dei soldi in cambio; inoltre gran parte del software libero è sostenuto economicamente attraverso sottoscrizioni volontarie: non vengono chiesti soldi per ottenere una copia del software, ma viene chiesto di supportarne lo sviluppo qualora lo si apprezzi.

La principale differenza tra software libero e proprietario è stata descritta da Eric Raymond

come quella che passa tra mercanti e costruttori di cattedrali: gli uni collaborano lun l'altro per creare un insieme che possa soddisfare ogni cliente e sono sempre disposti ad accettare nuovi collaboratori che diano nuova vita al mercato; gli altri creano progetti che possano soddisfare la maggior parte dei clienti tenendone segreto ogni componente, in modo che loro soltanto possano ampliarli. Questo si ripercuote pesantemente sul tipo di persone che abitualmente è possibile trovare nei due ambiti: i primi non lavorano necessariamente o soltanto per denaro, ma per migliorare le loro capacità e ottenerne il riconoscimento all'interno del loro ambiente, in termini di fama personale. Lo stesso modo di affrontare i problemi è completamente differente: se un "mercante" riconosce un problema è pronto a mettersi al lavoro per risolverlo, in modo che lavorando insieme si possano trovare e risolvere più problemi in meno tempo; ciò non può accadere nelle "cattedrali" sia per inferiorità numerica dei "costruttori" sia perchè chi trova un problema raramente ha i mezzi (l'accesso al codice sorgente, ad esempio) per risolverlo e deve aspettare che sia un costruttore a farlo.

Di conseguenza, nel software libero prevale la *full disclosure*, ovvero il rendere pubblici tutti i problemi del software in modo che tutti possano sapere dove e come si verificano e correre ai ripari riparandoli velocemente, mentre in quello proprietario vige la *non disclosure* e la *security through obscurity* ovvero il mantenere quanto più segreti i problemi del software per evitare che possano essere usati per danneggiare gli utenti durante tutto il periodo per il quale rimangono senza soluzione.