

Linux & il Software Libero

di

Rodolfo Giometti

giometti@linux.it

(f|ht)tp://ftp.enneenne.com/docs/talks/softwarelibero.p(s|df)

Il legame *Linux* ↔ *UNIX*

GNU/Linux* è un sistema stile *UNIX

UNIX è un progetto del 1969 e nasce per gioco dall'esperienza di *MULTICS*. Il sistema *UNICS* (*UNI*programmed Computer System) è stato sviluppato nei laboratori *AT&T* ed è stato distribuito gratuitamente alle Università (dal 1973).

UNIX è scritto in *C*, un linguaggio scritto appositamente per la programmazione di sistema, *C* è il successore di *B*, che è una semplificazione di *BCPL*.

La prima versione di *UNIX* era estremamente leggera, era stato sviluppato inizialmente su di un sistema con 16Kb di memoria.



16KByte



In ambiente *UNIX* sono stati sviluppati diversi progetti:

- L'editor di testi *vi* è stato sviluppato nel 1977 sotto *UNIX*.
- All'inizio degli anni '80 *TCP/IP* è stato sviluppato sotto *UNIX*.

Le caratteristiche di *UNIX*

Un sistema di tipo *UNIX* possiede diverse caratteristiche che lo rendono molto versatile, efficiente ed affidabile:

- *L'astrazione file*. Tutto all'interno del sistema è visto come se fosse un file: il CD-ROM, la stampante, una scheda di I/O digitale, ecc..



CD-ROM



Stampante



I/O digitale

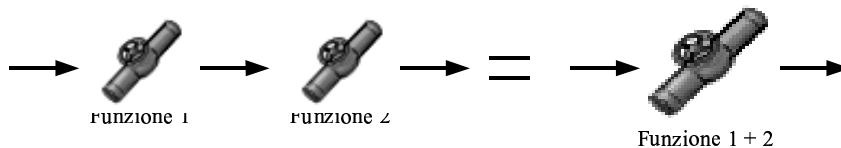


Floppy

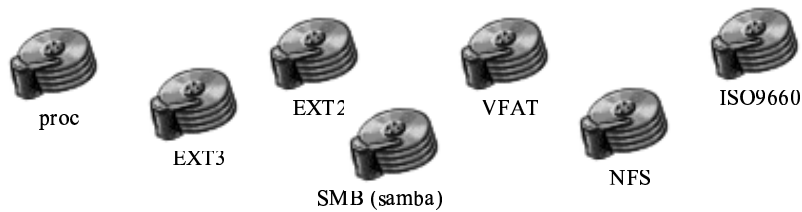


Scheda audio

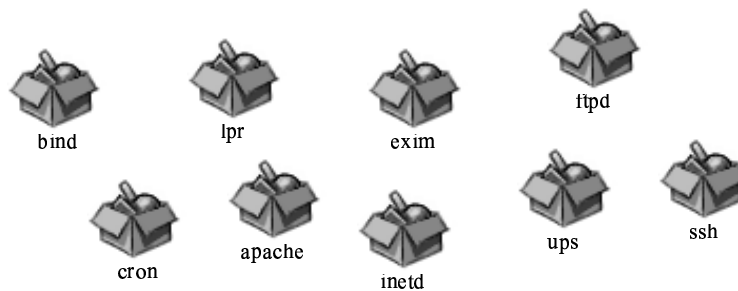
- Enorme disponibilità di *filtri*. Sono programmi che fanno solo una cosa e la fanno bene!!! Collegandoli in cascata si ottengono funzionalità molto complesse a partire da funzionalità semplici.



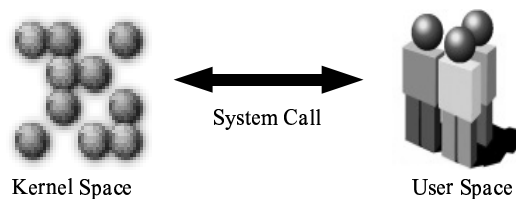
- Supporto per diversi tipi di *filesystem*. Il *filesystem* contiene tutti i file del sistema (anche i riferimenti alle periferiche).



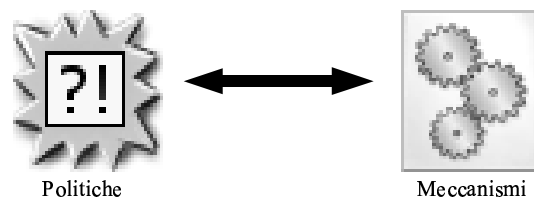
- Una gestione dei processi efficiente. Nei sistemi *UNIX* sopravvivono molte entità che evolvono nel tempo e che eseguono compiti (o programmi) specifici.



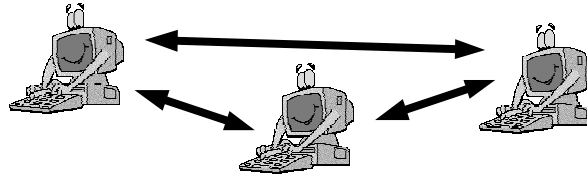
- Suddivisione tra *kernel space* e *user space*. Questa suddivisione è alla base della teoria dei sistemi operativi e serve per implementare una serie di politiche di «supervisione» dell'attività dei processi: permessi di accesso ai file/periferiche, protezione di memoria, ecc..



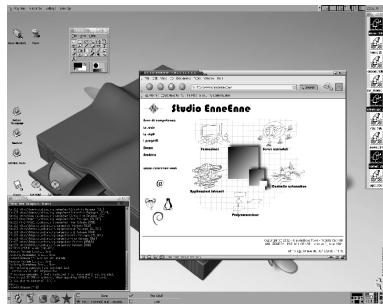
- Suddivisione tra *politiche* e *meccanismi*. Distinguere tra *meccanismi di accesso* alle risorse e le loro *politiche di utilizzo* permette di risolvere molti problemi in maniera elegante ed efficace.



- Possibilità di utilizzare i *socket*. Sono essenzialmente dei canali di comunicazione che possono essere utilizzati anche per la comunicazione tra macchine remote (stack *TCP/IP*).

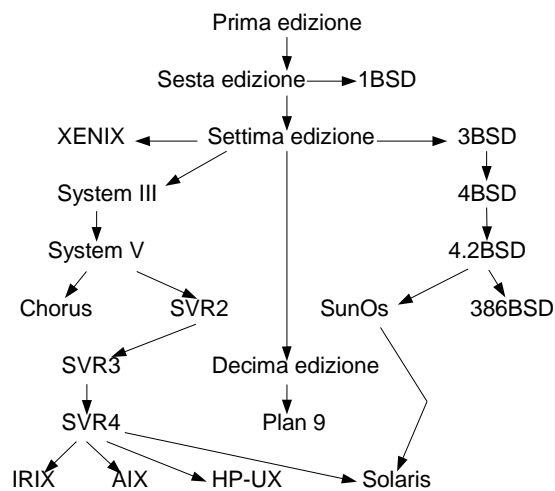


- Possibilità di usare l'ambiente grafico *X-Window*. È un supporto di tipo grafico molto potente anche per la sua possibilità di essere «remotizzabile».



Le evoluzioni di *UNIX*

La distribuzione in *forma sorgente* ha permesso la ramificazione e la successiva *proprietarizzazione* ne ha poi impedito la riunificazione.



Ritornare alle origini: progetto GNU

L'acronimo GNU sta per *Gnu is Not Unix* ed è stato fondato da R. M. Stallman nel 1984.

Non chiamavamo il nostro software «software libero», poiché questa espressione ancora non esisteva, ma si trattava proprio di questo.

La situazione cambiò drasticamente all'inizio degli anni '80 [...] I moderni elaboratori di quell'epoca, [...] avevano il proprio sistema operativo, ma nessuno di questi era libero: si doveva firmare un accordo di non-diffusione persino per ottenerne una copia eseguibile.

Questo significava che il primo passo per usare un computer era promettere di negare aiuto al proprio vicino. Una comunità cooperante era vietata. La regola creata dai produttori di software proprietario era: «se condividi il software col tuo vicino sei un pirata. Se vuoi modifiche, pregaci di farle».

L'idea che la concezione sociale di software proprietario, cioè il sistema che impone che il software non possa essere condiviso o modificato, sia antisociale, contraria all'etica, semplicemente sbagliata, può apparire sorprendente a qualche lettore. Ma che altro possiamo dire di un sistema che si basa sul dividere utenti e lasciarli senza aiuto?

Allora cercai un modo in cui un programmatore potesse fare qualcosa di buono. Mi chiesi dunque: c'erano un programma o dei programmi che io potessi scrivere, per rendere nuovamente possibile l'esistenza di una comunità? [...]

[R.M. Stallman]

Hacker

Fondamentalmente Stallman era un *hacker* e quindi aveva le competenze per poter portare a termine il suo intento.

È bene evidenziare subito che un *hacker* è:

***hacker* : 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary.**

[Jargon File (4.3.3, 20 Set. 2002)]

L'uso del termine *hacker* nel senso di *pirata* è una confusione di termini creata dai mezzi di informazione.

Gli *hacker* si rifiutano di riconoscere questo dualismo, e continuano ad utilizzare la parola nel senso di «uno che ama programmare, e a cui piaccia essere bravo a farlo».

La comunità *hacker* ha coniato il termine di *cracker* per definire:

cracker n. One who breaks security on a system. Coined ca. 1985 by hackers in defense against journalistic misuse of {hacker} [...]

[Jargon File (4.3.3, 20 Set. 2002)]

Lecture consigliate:

- *Hackers* di Steven Levy
- *L'etica hacker* di Pekka Himanen

La licenza *GPL*

Questa licenza è quella propria del progetto *GNU* e l'acronimo *GPL* sta per *General Public License* e si basa su quattro libertà:

- LIBERTÀ 0, o libertà fondamentale.

La libertà di eseguire il programma per qualsiasi scopo.

- LIBERTÀ 1.

La libertà di studiare il funzionamento del programma, e adattarlo alle proprie esigenze.

- LIBERTÀ 2.

La libertà di redistribuire copie del programma.

- LIBERTÀ 3.

La libertà di migliorare il programma, e distribuirne i miglioramenti.

La *GPL*, non si limita a concedere all'utente del programma le 4 libertà suddette, ma si occupa anche di proteggerle; infatti **chi modifichi un programma protetto da *GPL* e lo distribuisca con le modifiche, deve distribuirlo sotto licenza *GPL*.**

È grazie a questa protezione che la *GPL* è ad oggi la licenza più usata per il software libero.

Quando si utilizza la *GPL* è bene tenere presente la differenza che c'è tra *uso* del programma e la sua *modifica*.



La libertà 0 mi garantisce che posso **usare** un programma per fare quello che mi pare (anche «cose» non libere!)



Se **modifico/adatto** un programma coperto da *GPL* allora non ho scelta: il risultato **DEVE** essere *GPL*!!!

Le «altre» licenze

C'è da notare che queste libertà nelle «normali» licenze non vengono in alcun modo riconosciute:

- ILLIBERTÀ 0.

Non si può usare il programma per qualsiasi scopo.

Se il PRODOTTO SOFTWARE è Windows 2000, non potrà essere utilizzato da più di due (2) processori sul COMPUTER [...]

[licenza Windows]

Se il PRODOTTO SOFTWARE è Windows Millennium Edition o Windows 98, è possibile connettere un massimo di cinque (5) COMPUTER al singolo COMPUTER che esegue il PRODOTTO SOFTWARE esclusivamente per accedere a Internet [...]

[licenza Windows]

- ILLIBERTÀ 1.

Non si può studiare il funzionamento del programma.

L'utente non può decodificare, decompilare o disassemblare il PRODOTTO SOFTWARE [...]

[licenza Windows ME]

Decodificazione. Se il SOFTWARE è stato acquistato nell'ambito della Comunità Europea, l'utente non può convertire, decodificare, decompilare o disassemblare il SOFTWARE, se non nei limiti e per gli specifici fini autorizzati dalle singole leggi applicabili.

[licenza EULA di Office]

- ILLIBERTÀ 2.

Non si possono redistribuire copie del programma.

[...] l'utente è autorizzato a utilizzare una (1) copia del PRODOTTO SOFTWARE [...]

[licenza Windows ME]

L'utente non può concedere in locazione, in leasing o in prestito il PRODOTTO SOFTWARE a un altro utente. [...]

[licenza Windows ME]

Locazione. L'utente non può concedere in locazione, in leasing o in prestito il PRODOTTO SOFTWARE a un altro utente.

[licenza EULA di Office]

- ILLIBERTÀ 3.

Non si può migliorare (o correggere) il programma ne e distribuirne i miglioramenti.

Deriva direttamente dalle illibertà 1 e 2 che impediscono rispettivamente lo «studio e il funzionamento del programma» e «la possibilità di redistribuirne delle copie».

Si noti come questa «illibertà» ingigantisca il concetto di «pirateria», se si utilizzassero prodotti basati sulla *GPL* (o comunque sul software libero) molti dei comportamenti che qualificano un utente come «pirata» sarebbero permessi dalla licenza d'uso, e quindi legali.



Quindi la comunità del software libero dice **STOP alla pirateria: usiamo tutti il software libero!!!** {:^)

Molti accusano la *GPL* quando dice «[...] chiunque modifichi un programma protetto da *GPL* e lo distribuisca con le modifiche, deve distribuirlo sotto licenza *GPL*».

Questa è una clausola che, da un lato, serve per proteggere il lavoro dell'autore che ha deciso che il suo lavoro deve essere distribuito in forma libera (impedendone la proprietarizzazione) e, dall'altro, permette a più persone di avere una applicazione complessa anche contribuendo solo ad una parte.

A chi non piace questa clausola può benissimo non usare il prodotto coperto da *GPL* nella stessa identica maniera consigliata da altri produttori di software di tipo proprietario:

Qualora l'utente non accetti i termini e le condizioni del presente Contratto di Licenza, il Produttore e Microsoft Licensing Inc. ("Microsoft") non sono disposti a concedere all'utente il PRODOTTO SOFTWARE in licenza. [...]

[licenza Windows ME]

Altro aspetto da non sottovalutare è il perché con il software libero non si vede mai la richiesta sulla licenza:

Accetto

Rifiuto

Come evidenziato da Alessandro Rubini <rubini@gnu.org>:

*«Una licenza di tipo EULA, che viene accettata e firmata (se no non te la danno) è un contratto in cui l'utente **cede** diritti che gli sarebbero propri. Per esempio potresti impegnarti a non installare su un'altra macchina quando butti via quella attuale, o impegnarti a non far usare il programma a tua moglie, o impegnarti a non usare il programma sotto VNC, o a non usare il compilatore per compilare un compilatore [...]*»

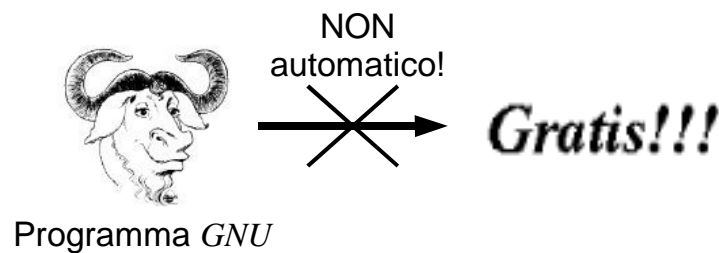
«La GPL e le altre licenze [del software libero] dicono "invece di impedirti di redistribuire, te lo permetto solo in certe condizioni". Ripeto, non chiede le condizioni, cede l'esclusiva solo fino a un certo punto.».

Libertà non prezzo

Free as free speech and not as free beer

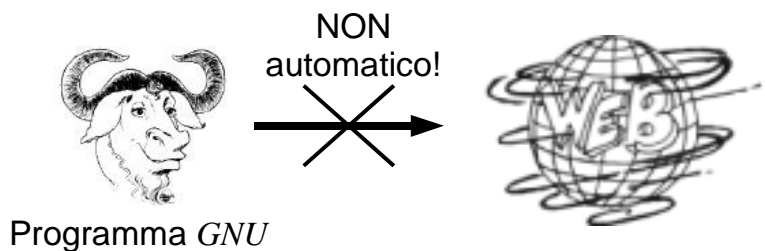
[R. M. Stallman]

Le licenze *GPL* e *LGPL* sono progettate per assicurare la libertà di utilizzo, modifica o copia del software e di farsi anche pagare per questo (se si vuole), quindi **un programma coperto da licenza *GPL* non è automaticamente un programma gratis** (e viceversa)!!!.



La *GPL* poi dice anche che solo chi possiede già il programma può vantare «diritti» su di esso.

Quindi **rilasciare un programma sotto *GPL* non significa che questo debba essere pubblicato necessariamente sul web!!!**



Per ulteriori informazioni si veda:

- <http://www.gnu.org>,
- <http://www.fsf.org>, <http://www.fsfeurope.org>
- <http://www.linux.it/GNU>, <http://www.linuxdidattica.org>
- <http://www.softwarelibero.it>

Software libero: una necessità

[...] Ad oggi, il software libero è ampiamente diffuso in ambito accademico, industriale e fra gli appassionati di calcolatori, soprattutto grazie ai sistemi GNU/Linux. Questi sistemi liberi sono disponibili a costi molto bassi, ben inferiori a quelli di analoghi sistemi proprietari. Tuttavia, a causa delle loro caratteristiche, il loro uso richiede una buona cultura di base nel campo del software.

In ambito accademico viene molto apprezzata la possibilità di personalizzare ogni parte del sistema, visto che i programmi liberi sono liberamente modificabili (libertà numero uno). In ambito industriale, si apprezza l'affidabilità dei sistemi liberi, dovuta al fatto che quando un utente corregge un errore in un programma solitamente rende disponibile la correzione agli altri utenti (libertà numero tre). Gli appassionati di calcolatori apprezzano lo spirito di condivisione esistente fra gli utenti di software libero.

Ma le implicazioni dell'uso del software libero non sono soltanto tecniche ed economiche, perché il software da tempo ormai è avviato ad occupare un ruolo di primo piano nella nostra vita quotidiana, ed è destinato a cambiare in maniera profonda la società.

È per queste ragioni che la nostra libertà futura dipenderà anche dalla capacità di ognuno di noi di controllare il software. È per queste ragioni che ai tradizionali principi di libertà sessuale, di culto, di movimento, di espressione deve essere affiancata la libertà del software. È per queste ragioni che la nostra libertà futura dipenderà anche dall'uso di software libero.

[Francesco Potorti - Associazione Software libero]

Lavorare con il *Software Libero*

Dal punto di vista del produttore di software

- Si riesce a vendere solo poche copie (tranne casi eccezionali) e la commercializzazione è solitamente per contatto diretto.
- Il ritorno economico avviene solitamente tramite i contratti di assistenza e/o consulenza.
- Si possono utilizzare le soluzioni software già adottate da altri diminuendo enormemente i tempi di sviluppo.

- Si hanno a disposizione moltissimi *tool* di sviluppo a costi ridotti favorendo così anche l'iniziativa privata.
- Le conoscenze di base sono tutte disponibili in rete e di accesso immediato.
- Per poter lavorare occorre una conoscenza degli aspetti tecnici più elevata del normale per poter essere sempre aggiornati sulle continue innovazioni e fornire ai clienti consulenze di qualità.

Lavorare con il *Software Libero*

Dal punto di vista dell'utilizzatore del software

- È possibile utilizzare il programma senza i vincoli (alle volte assurdi) delle licenze proprietarie.
- I dati generati possono essere convertiti in formati diversi rendendoli quindi accessibili (e recuperabili) anche dopo molti anni.
- Il prodotto software è svincolato dall'assistenza tecnica, che può essere richiesta anche a fornitori diversi, e questo non muore se il produttore lo abbandona (volontariamente o involontariamente).

- L'eventuale costo del prodotto software (che è sempre commisurato al lavoro fatto e non predeterminato a priori) o della sua assistenza può essere ripartito sul numero di sistemi che lo utilizzano.

- Si possono utilizzare anche macchine ormai considerate obsolete per implementare semplici sistemi di varia utilità (server di stampa, fax server, gateway per basso/medio traffico, ecc.).

Considerazioni finali

Il mercato attuale dell'informatica:

- È teatro di inutile consumismo.
- Necessita di grossi investimenti anche per compiti semplici.
- È dominato dai colossi del settore e tende ad oscurare i singoli programmatori e le piccole imprese.

Quello che può fare il *Software Libero*:

- Limitare lo spreco di macchinari, e di conseguenza l'inquinamento, permettendo l'uso di macchine obsolete per i compiti poco onerosi.
- Aprire il mercato all'iniziativa individuale facendo prevalere i più bravi grazie alla possibilità di poter scegliere e giudicare in base alle caratteristiche del prodotto (o delle consulenze fornite).
- Permettere una maggiore diffusione della cultura informatica anche permettendo il dirottamento di grossi capitali, ora usati prevalentemente per acquistare licenze, verso la ricerca e lo sviluppo.