

## SOFTWARE OPEN E FREE SOFTWARE

### Obiettivi formativi

L'obiettivo di un docente, oltre la trasmissione di contenuti e significati, deve basarsi e concentrarsi, principalmente, sullo sviluppo delle potenzialità e delle capacità dello studente stimolando quel senso critico necessario ad affrontare la vita nei suoi tortuosi percorsi che lo portano a dover decidere su quale sia la strada giusta, di volta in volta, da percorrere. Non credo, di conseguenza, possa considerarsi del tutto corretto il principio secondo cui a casa, o a scuola, è presente questo o quel software e, pertanto, questo o quello “deve” essere studiato. Credo sia più corretto presentare il panorama più vasto possibile, compatibilmente alle risorse a disposizione e alle possibilità, anche economiche, dei genitori che desiderano vedere realizzati i sogni e i progetti dei loro figli. Con questo intento, seguendo i principi democratici a cui ciascun paese, secondo me, “deve” tendere, tali da offrire a tutti i cittadini la possibilità di crescere e di imparare, ritengo necessario affrontare ad inizio corso un tema-stimolo apparentemente banale come la scelta del software da utilizzare, affinché possano riflettere e, successivamente, decidere su/con quali software studiare, poiché non si è veramente liberi se non si hanno gli strumenti che ci permettono realmente di pensare e di agire secondo coscienza. Anche la scelta di un software, in un certo senso, può aiutarci a raggiungere questo obiettivo.

### Obiettivi specifici per la didattica (e per l'amministrazione)

L'utilizzo del free software e dell'open source offre vantaggi sia per la didattica che per l'amministrazione pubblica in quanto:

1. L'uso del software libero permette di risparmiare il costo delle licenze con un utilizzo più intelligente delle risorse finanziarie a disposizione.
2. Chi usa questo software non necessita di piratare il software, educando naturalmente alla legalità.
3. Il software libero permette di padroneggiare la tecnologia, mettendo a disposizione il codice sorgente, realizzando così il sogno di ogni insegnante: avere il controllo di ciò che insegna.
4. La scuola può modificare il software e adattarlo alle proprie esigenze senza dover sottostare a dei presunti "standard" monopolistici: ovvero non è la didattica in funzione del software ma il software in funzione delle esigenze didattiche.
5. Il software libero è notoriamente più stabile e sicuro del software proprietario, permettendo anche qui delle grosse economie.

### Contenuti

1. Che cosa si intende per Open Source
2. Radici storiche
3. La cultura hacker
4. Stallman e il Free Software
5. L'arrivo di Linux
6. Vantaggi e svantaggi del software libero
7. Il software libero nella scuola e nella pubblica amministrazione
8. APPENDICE:
  - *Codici ribelli, la libertà corre sulla tastiera*  
Intervista a Richard Stallman, fondatore della «Free Software Foundation» e ispiratore del sistema operativo per computer Gnu-Linux, di Arturo di Corinto.
  - *Perché scegliere il software libero nella didattica?*  
Intervista ad Antonio Bernardi - responsabile del progetto “Software libero nella scuola”.

### Metodologie e strumenti

Lo sviluppo dell'unità didattica a partire dall'esperienza degli studenti è realizzata mediante l'utilizzo di metodi attivi di insegnamento e l'utilizzo di strumenti informatici e visione di filmati selezionati opportunamente dal docente inerenti all'argomento trattato.

*"Il progetto Open Source si concentra sui vantaggi pratici che si possono ottenere da una comunità di utenti che collaborano scambiandosi e migliorando il software. Il movimento Free Software si fonda sulla libertà di collaborare con gli altri. Credo che la libertà di avere una comunità sia importante per la qualità della nostra vita, per avere una buona società in cui vivere e questo, per me, è molto più importante di avere un software potente e affidabile".*

**Richard Stallman (Fondatore, GNU Project)**

## 1. CHE COSA SI INTENDE PER OPENSOURCE

Il termine composto "Open Source" nasce dal mondo della programmazione informatica e sta ad indicare alcune caratteristiche tecniche del **software**.

Un programma non è altro che un insieme di istruzioni per far sì che il computer svolga determinate funzioni. Il computer "ragiona" in un sistema binario in cui ogni informazione deve essere codificata in una serie composta da sole due cifre: "0" e "1". Il programmatore scrive il programma in un determinato linguaggio, dopo di che un altro programma (chiamato **compilatore**) trasformerà automaticamente durante l'installazione le istruzioni in linguaggio binario rendendole così assimilabili da parte del computer. Il codice binario viene denominato "codice oggetto", mentre il codice con cui è stato sviluppato il software rappresenta il "codice sorgente", comprensibile e modificabile da qualunque programmatore che conosca il determinato linguaggio informatico usato.

'Sorgente' in Inglese si traduce 'source' e sottintende in un'unica parola il sostantivo 'codice': è per questo che, per trasposizione, nel gergo corrente italiano è più facile trovare 'il sorgente', unendo un articolo maschile con un sostantivo femminile in quanto si fa riferimento al 'codice sorgente'.

La traduzione letterale di 'open source' diviene dunque 'sorgente aperto' (e non 'sorgente aperta'). L'aggettivo 'open' sta a significare che il codice sorgente rimane accessibile a chiunque voglia intervenire sul programma, correggerne gli errori (detti in gergo '**bug**') sorti durante la sua compilazione, aggiornarlo, perfezionare le sue funzioni, oppure semplicemente studiarne i meccanismi e prendere spunto per altri programmi.

La scelta di lasciare accessibile il sorgente era la prassi nei primi anni di vita della nuova scienza informatica; la tendenza man mano s'invertì quando le imprese produttrici di software iniziarono a irrigidire il sistema inserendo dei **meccanismi di crittazione** per i quali è invece molto difficile o impossibile intervenire sul programma. Tutto ciò ovviamente con lo scopo di massimizzare lo sfruttamento economico e facendo leva sulle tutele proprie del diritto d'autore e sui relativi diritti esclusivi.

La locuzione 'open source' individua un fenomeno, una filosofia, un movimento culturale. Appunto, un movimento basato sui principi (e sull'orgoglio) di coloro che potremmo chiamare i programmatori di prima generazione (gli indipendenti, gli **hacker**<sup>1</sup>) e nato come reazione a quella tendenza ormai costante ad iper-tutelare le creazioni informatiche.

---

<sup>1</sup> Si tratta di un concetto che esula da qualsivoglia connotazione negativa e che nasce in un mondo ben lontano da quello evoluto e interconnesso in cui possono pavoneggiarsi i veri pirati informatici. Un hacker, nell'accezione originaria di questo neologismo, è solo e semplicemente un esperto di informatica a cui piace programmare, che lo fa non con intenti di profitto ma per una sorta di irrefrenabile passione, quasi per vocazione. "[...] la programmazione per gli hacker è in primo luogo gratificazione dell'ego e solo a volte diventa, in aggiunta, leva di reddito. Allo stesso modo del canto o della pittura per gli artisti". Cfr. BASSI, *Open Source - analisi di un movimento*, Apogeo, Milano, 2000; disponibile anche alla pagina web <http://www.apogeonline.com/ebook/90026/scheda.xhtml> oppure alla pagina web:

## 2. RADICI STORICHE

I primi calcolatori (a valvole e - solo poi - a transistor) comparvero già nei primi anni 50; ovviamente, si trattava di congegni mastodontici con limitatissime funzioni, che rimasero operanti a livello di università e centri militari e rappresentavano ancora una fase semi sperimentale della scienza informatica.

Nel 1969, la prima ristretta comunità hacker venne costretta (dalla rivoluzione culturale in atto in quel periodo) ad uscire dal suo originario isolamento nelle università e nei centri di ricerca e ad affacciarsi al mondo reale. In quell'anno infatti vide la luce il sistema operativo **Unix**, grazie al lavoro di uno sviluppatore dei laboratori Bell: Ken Thompson, personaggio appartenente appunto a questa prima generazione di hacker. Unix era il primo sistema operativo sviluppato in linguaggio C (un particolare linguaggio di programmazione) e non in linguaggio macchina (binario) ed era il primo a ricercare l'idea di portabilità e compatibilità. Prima di Unix, cioè, ogni computer necessitava un apposito sistema di software (sistema operativo + programmi vari); ogni volta che la macchina veniva aggiornata o sostituita era necessario riprogettare gran parte del sistema software. Grazie a Thompson invece il ruolo del software si fece più dinamico e più facilmente gestibile, indipendentemente dal supporto hardware su cui era installato; fu dunque possibile affacciarsi su un mercato dell'informatica decisamente più ampio ed elastico.

Il 1969 è inoltre l'anno in cui vennero collegati per via telematica i nodi dei centri di ricerca informatici di quattro grandi università statunitensi (Los Angeles, Santa Barbara, Stanford, Utah): nacque così ARPAnet, riconosciuta da tutti come l'effettivo embrione dell'Internet dei nostri tempi.

Si passa così, con l'inizio degli anni 70, ad una seconda generazione di hacker fedele ai principi etici originari, ma interessata più che altro alla diffusione del mezzo su cui amavano operare. Il loro obiettivo era quello di fare uscire lo strumento 'computer' dai grandi centri di ricerca, per renderlo più familiare alla massa degli utenti; si impegnavano affinché le apparecchiature fossero più piccole, maneggevoli ed economiche. In questo periodo apparvero i primi computer in kit di montaggio: apparecchi piuttosto spartani venduti ad un prezzo base di 397 dollari e contenenti i primi processori Intel. È sempre in questo periodo che si cominciò a sentir parlare di Bill Gates (il magnate di Microsoft) il quale ebbe il merito assieme a Paul Allen di aver utilizzato efficacemente il linguaggio Basic per rendere più semplice il funzionamento dei computer **Altair**.

Nacque dunque nei primi anni 80 il concetto di personal computer, sicuramente grazie all'impegno degli hacker nel "liberare l'hardware", ma anche per ben più venali interessi economici da parte delle imprese che iniziarono a sentire odore di affari. La International Business Machine infatti mise sul mercato il suo primo computer da tavolo chiamato appunto IBM-PC; e in contemporanea la stessa scelta di marketing venne compiuta dalla Apple e dalla Atari. IBM adotta inizialmente una politica aziendale piuttosto "illuminata", cercando di incoraggiare la diffusione e lo sviluppo del software e stimolando la collaborazione di altre importanti imprese, come la Microsoft che realizzò il sistema operativo per i nuovi computer: il sistema MS-DOS, tuttora fondamentale per il funzionamento dei nostri PC.

In tal modo, quello strano aggeggio ibrido fra una macchina da scrivere e un televisore cominciava a fare capolino negli arredi delle case e degli uffici di tutto il mondo e in molti casi dovette "svilire" la sua funzione, essendo sfruttato come gioco e passatempo invece che come strumento di calcolo. In questo modo una massa di persone inesperte si trovò ad utilizzare giochi e software senza essere in grado di capire (o senza nemmeno voler capire) di cosa effettivamente si trattasse e di come

---

<http://www.dvara.net/HK/open.asp> ; p. 36 (par. 2.2.).

fossero stati sviluppati; scegliendo i prodotti in base alla pubblicità o semplicemente affidandosi a pacchetti standard.

Una conseguenza logica di questa espansione a macchia d'olio: più gli utenti divenivano numerosi e più questa terza generazione di hacker risultava frazionata e composita. Non più solo lo zoccolo duro degli studiosi di informatica e di tecnologia, ma anche una sempre più numerosa schiera di curiosi, ai quali era però difficile trasmettere in modo completo e autentico certi principi etici nati in una sorta di ristretta casta.

Si arrivò così ad uno scenario abbastanza simile ai giorni nostri, in cui gli utenti si dividevano in varie macro-comunità rese compatte, più che dai principi, dagli usi che fanno del PC e collegate dalla prima vera e propria Internet (come la si intende oggi): i vecchi fedelissimi dediti alla libera ricerca, coloro che usavano il PC negli uffici quindi per lavoro, gli appassionati dei giochi con la loro smania di scambiarsi trucchi e versioni aggiornate.

### 3. LA CULTURA HACKER

Soffermiamoci ora sugli aspetti più rappresentativi dell'etica hacker. È, in effetti, fondamentale capire come siano stati proprio questi principi di “filosofia sotterranea” ad influire maggiormente sulle nuove istanze in fatto di copyright del software.

I punti cardine di questa filosofia sono principalmente i seguenti:

- **libertà** di accesso alle risorse, siano esse intese come accesso alle informazioni, ai dati, oppure come accesso alle macchine e ai relativi componenti tecnologici necessari al loro miglior funzionamento;

- **condivisione** delle conoscenze e degli strumenti;

- **cooperazione** ed **unità** nella realizzazione dei progetti utili alla comunità;

- **semplificazione** sia a livello tecnico sia a livello burocratico;

- **creatività**: la progettazione, conoscenza (e in certi casi manomissione) dei sistemi informatici è considerata un'arte e quindi ogni operazione deve essere compiuta con stile e originalità;

- **onore** e **credibilità**: tutti i cardini etici fin qui citati sono poi amalgamati da un grande senso dell'onore, della reputazione, della rispettabilità che pervadono la comunità hacker; le varie sotto-comunità e i singoli progetti infatti hanno un loro leader il quale si è guadagnato la credibilità con i meriti e l'anzianità; e non mancano gli opinion-leader ovvero gli ideologi dell'hacking i quali si distinguono per carisma e capacità comunicativa e si fanno perciò portavoce della comunità e catalizzatori di attenzione.

- Ovviamente una simile organizzazione non può che dotarsi di un proprio **linguaggio** originale formatosi con anni di strambe etimologie e distorsioni linguistiche (tratte magari da termini ultra-tecnici) e dal quale in questi primi paragrafi ho mutuato molte espressioni: uno slang caratteristico di matrice quasi totalmente “American-English” che si distingue per la sua insostituibile efficacia.

- Non bisogna inoltre dimenticare lo spirito che da sempre contraddistingue le azioni di hacking: ovvero quello **spirito ironico** e **dissacratorio** che fa apparire spesso come goliardiche anche le incursioni più fastidiose.

### 4. STALLMAN E IL FREE SOFTWARE

Richard M. Stallman, abile ed esperto programmatore indipendente, appartiene a quella prima generazione di hacker e ne rappresenta lo stereotipo integerrimo e incontaminato. Questo suo essere un uomo ‘tutto d'un pezzo’ è la caratteristica che da un lato gli permette di potersi innalzare a massimo portavoce della comunità hacker di quel periodo, ma dall'altro gli causa grandi problemi di convivenza nella gestione dei progetti concreti.

Nel 1983 egli è il primo ad avvertire repulsione per quel nuovo modello d'informatica che andava a configurarsi; ed è il primo a muoversi concretamente e visibilmente nella direzione del risveglio delle coscienze di coloro che l'informatica l'avevano simbolicamente partorita. All'epoca egli poteva già vantare (oltre ad una laurea in fisica *cum laude*) una decina d'anni d'esperienza come programmatore, dato che già nel 1971 era stato assunto nel laboratorio di Artificial Intelligence (AI lab) del MIT. In quegli anni aveva avviato la prassi di distribuire gratuitamente e liberamente i suoi programmi, incoraggiando chiunque a modificarli e migliorarli: fu questa la sorte dell'Emacs, il diffusissimo programma di videoscrittura da lui stesso creato, che portò alla genesi della "Comune di Emacs", una specie di catena di utenti che avevano come unico obbligo quello di diffondere a loro volta liberamente le modifiche apportate.

È ben intuibile ora come mai un personaggio del genere abbia messo in atto la più ferrea ribellione contro i nuovi criteri di diffusione del software: di fronte allo scorrere inevitabile di quegli eventi, Stallman si sentiva l'ultimo vero hacker sopravvissuto al cambiamento del suo habitat naturale. Sempre nel 1983 decide dunque di abbandonare il MIT, per dedicarsi a lungimiranti progetti personali che tenessero in vita lo spirito hacker a cui tanto era affezionato: primo fra tutti, la realizzazione di un sistema operativo di tipo Unix che fosse però dall'impostazione esclusiva del copyright e distribuibile liberamente. Nasce appunto il **Progetto GNU** (leggasi /'gnu/ con la "g" dura di "gatto", e non /'jnu/ come in "gnocchi"), acronimo ricorsivo che sta per Gnu's Not Unix (Gnu non è Unix) con un'espressa vena di antagonismo e di sfida, il quale appunto avrebbe dovuto coinvolgere migliaia di irriducibili e condensare le conoscenze e gli sforzi condivisi nel nuovo sistema operativo.

Ovviamente tutti i potenziali "adepti" avrebbero dovuto poter avere un punto di riferimento unico che si facesse portavoce e mecenate dell'impresa; nasce così, sempre per opera di Stallman e sotto la sua stretta supervisione, la **Free Software Foundation**: un'organizzazione *no-profit* mirata alla raccolta di fondi, al coordinamento dei progetti e alla sensibilizzazione del popolo dell'informatica. Nei primi anni di vita entrambi i progetti (che si potrebbero considerare come due manifestazioni di un unico progetto) faticarono ad ingranare, a causa della già accennata situazione di frammentazione della comunità degli informatici.

Le cose cambiarono invece con la raggiunta maturazione della rete telematica: quando infatti Internet verso la fine degli anni 80 iniziò a connettere stabilmente un numero cospicuo di utenti, il messaggio della Free Software Foundation (FSF) poté raggiungere gli hacker di vecchio stampo sparsi per il mondo, che erano rimasti anch'essi disorientati e isolati dal cambiamento. Il seme della rivoluzione 'scongelato' – per così dire – da Stallman dopo un periodo di ibernazione, poteva infatti trovare solo nelle coscienze dei singoli appassionati (e non certo nelle *lobbies* di potere dell'imprenditoria tecnologica) il terreno fertile di cui aveva bisogno per svilupparsi.

Richard Stallman, fu il primo a formalizzare il concetto di software libero. Tale software deve offrire alcune libertà fondamentali per essere considerato tale:

- **Libertà 0**, libertà di eseguire il programma per qualsiasi scopo, senza vincoli sul suo utilizzo;
- **Libertà 1**, libertà di studiare il funzionamento del programma, e di adattarlo alle proprie esigenze;
- **Libertà 2**, libertà di ridistribuire copie del programma;
- **Libertà 3**, libertà di migliorare il programma e di distribuirne i miglioramenti. Il software libero è accompagnato da una licenza che definisce in termini legali le libertà garantite.

Come accennato, fu lo stesso Stallman, nel 1984, a creare il progetto GNU; il progetto aveva lo scopo di tradurre in realtà le teorie sul software libero, realizzando un nuovo sistema operativo che gli utenti avrebbero potuto scambiarsi, ed ovviamente modificare liberamente. Il progetto GNU fu sostenuto dalla fondazione creata da Stallman, la Free Software Foundation (Fondazione per il Software Libero), a livello legale ed economico. Creare un nuovo sistema operativo, avrebbe significato gettare la basi per una grande comunità che avrebbe potuto finalmente scrivere programmi



adatti al sistema operativo per scambiarli liberamente. Dunque, l'abbozzo di sistema operativo creato di Stallman e da una piccola comunità di programmatori fu battezzato, come abbiamo avuto modo di accennare, GNU. Un requisito fondamentale di GNU fu l'essere "open source". Infatti, mentre il software proprietario è fornito già "compilato" (cioè già tradotto da linguaggio di programmazione - comprensibile agli utenti - a linguaggio "macchina" - comprensibile al computer) in modo da impedire ogni tentativo di interpretazione da parte dell'utilizzatore, il software "open source" rende disponibili i "sorgenti" (source) cioè tutte le righe di codice non tradotte in linguaggio macchina. Il software libero deve perciò essere "open source" in modo che, conformemente alle libertà fondamentali garantite, possa essere "letto", analizzato e modificato.

Il software si divide in diversi tipi. Potremmo costruire una piramide immaginaria, al cui gradino più basso vi è il software proprietario, subito sopra il cosiddetto "freeware" (che è ugualmente software proprietario, ma che fornisce garanzia di libero utilizzo all'utente). Tra i vari tipi di software proprietario e i massimi gradi di libertà esistono infiniti gradini: ognuno è rappresentato da diverse licenze di utilizzo del software che vanno da quelle più restrittive a quelle meno restrittive. Il grado massimo di libertà sarebbe rappresentato dal software detto "di pubblico dominio". Materiale di pubblico dominio potrebbe ovviamente essere utilizzato per creare software proprietario.

Perciò Stallman decise che GNU non sarebbe stato di pubblico dominio. Si trattava di una limitazione necessaria affinché il progetto non si estinguesse nel giro di poco tempo. In effetti, se distribuire GNU con licenze "di pubblico dominio", avrebbe significato senz'altro una enorme diffusione in tempi rapidi del progetto, al tempo stesso avrebbe concretamente alimentato le idee dei creatori di software proprietario. Fu dunque scelto di proteggere il prodotto con un nuovo tipo di licenza, chiamata da Stallman e dai suoi collaboratori GPL (**General Public Licence**), che chiarisce in termini legali cosa gli utenti possono fare e non fare con il software.

La licenza fu scherzosamente detta "**copyleft**", un gioco di parole fra il diritto d'autore ("copyright") e la contrapposizione dell'idea di Stallman ("right" significa anche "destra", mentre "left" che significa sia "sinistra", sia "concessione"). Il "copyleft" fu chiamato da Stallman "permesso d'autore", in modo da distinguerlo dal "diritto d'autore", concetto strettamente correlato con il copyright. La GPL e il concetto di "copyleft" hanno ispirato tra l'altro l'innovativo sistema di licenze "Creative Commons", che vengono largamente usate su Internet. "Il software", spiega Stallman riferendosi al "copyleft", "è coperto da copyright e gli autori danno il permesso agli utenti di studiarlo, modificarlo, apportare dei cambiamenti. Ma quando lo ridistribuite dovete farlo a certe condizioni, né più né meno. In questo modo le persone a cui lo darete saranno in grado, se vorranno, di cooperare con gli altri utenti. In questo modo, dove va il software, va anche la libertà".

La licenza GPL comportò un grande passo avanti nella storia del software libero perché si trattò della prima licenza scritta dalla parte della comunità (o se vogliamo degli utenti), e non da una azienda con il solo scopo di tutelarsi o, come nel caso di un'altra licenza (BSD), di ottenere finanziamenti governativi. Un evento fondamentale per il progetto fu la fondazione di aziende ispirate al movimento del software libero. Fu lo stesso Stallman ad intuire la possibilità di guadagno insita nel progetto del software libero. "Se per lavoro usi del software libero", spiega Stallman, "hai la possibilità di scegliere tra un vasto numero di persone, che operano in diverse aziende del settore. In linea generale devono offrire un buon servizio di assistenza, altrimenti bisogna rivolgersi presso altre aziende. Con il software proprietario l'assistenza è un monopolio: c'è una sola azienda che possiede il codice sorgente, e solo lei può garantire assistenza, perciò non c'è da stupirsi che l'assistenza sia così scadente".

## 5. L'ARRIVO DI LINUX

Nel 1990 il sistema GNU era quasi ultimato: mancava ancora il "**kernel**", cioè l'insieme di programmi base che consentono la gestione delle risorse fondamentali come l'unità di calcolo e la memoria centrale. Il kernel era certamente la parte più importante, ma la sua realizzazione fu rinviata in attesa della liberalizzazione di un kernel sviluppato dall'università dello Utah. "Avevamo appena iniziato a lavorare sul kernel, quando arrivò Linus Torvalds", racconta Stallman. Linus Torvalds, uno studente ventenne dell'università di Helsinki, aveva sviluppato un kernel che qualcuno riuscì a far funzionare molto velocemente con GNU. Linus aveva deciso infatti di creare un sistema simile ad UNIX, che aveva conosciuto presso l'università, per farlo "girare" sul suo nuovo computer. La sua "creatura" fu chiamata Linux, e dal 1996 ebbe come mascotte il simpatico pinguino chiamato "Tux".

Linus ebbe la magnifica idea di diffondere la sua opera su Internet mettendola a disposizione di chi fosse stato interessato ad utilizzarlo o migliorarlo. Insomma, quando qualcuno riuscì a far funzionare insieme Linux e GNU fu chiaro che il progetto era arrivato ad un importante punto di svolta. La combinazione tra GNU e Linux diede vita al primo sistema operativo completo, battezzato semplicemente GNU/Linux. Questo sistema operativo è attualmente utilizzato da oltre dieci milioni di persone attraverso i server di Internet, come vedremo, ma ben poche di esse ne sono al corrente.

Affinché GNU/Linux potesse uscire dal mondo dei programmatori era necessario un ulteriore passo, che fu attuato con la creazione di un programma che rese possibile l'erogazione di siti web complessi: **Apache HTTP Server**.

È stato osservato che la curva che descrive la diffusione di Linux e la diffusione di Internet sono direttamente proporzionali: dal 1993, anno di nascita di Apache, è iniziato il boom degli Internet provider e quindi la diffusione della rete Internet come un bene di massa. In sostanza fu proprio Apache a convincere i provider e le aziende di e-commerce ad adottare Linux e non Microsoft Windows. Inoltre praticamente tutte le "web-application" che permettono agli utenti di creare un "blog" sono open source.

Da allora un numero sempre maggiore di utenti venne a conoscenza di GNU/Linux, e molti di loro decisero di adottarlo come sistema operativo. Con il passare del tempo gli utenti si sono organizzati in gruppi, ognuno dei quali rilasciava "distribuzioni" (in gergo dette "distro") del sistema operativo sempre nuove e diverse. Il tutto per pura passione: spesso gli utenti che sono in prima linea per il software libero, sono quelli che scrivono programmi perché semplicemente ne hanno bisogno. Le implicazioni sociali del software libero sono davvero notevoli. Oltre a quelle che derivano dalla convinzioni e dalle teorie prettamente filosofiche, esistono quelle decisamente più pragmatiche: la condivisione del sapere non premette a piccoli gruppi di persone di usarla per acquisire una posizione di potere politico o economico.

## 6. VANTAGGI E SVANTAGGI DEL SOFTWARE LIBERO

Tra i punti di forza dell'utilizzo del software libero si annoverano sicuramente: l'affidabilità, l'efficienza, la sicurezza e la gratuità.

- **Affidabilità:** il software libero è in primo luogo affidabile. In che senso? Nel senso che è molto stabile, ed è difficile che presenti errori o bug per il semplice fatto che, essendo i codici sorgente disponibili per tutti, migliaia di programmatori possono leggere il codice sorgente, aumentando esponenzialmente le possibilità di scoprire eventuali problemi. Almeno un programmatore su un migliaio individuerà il problema, e lo correggerà, rilasciando per tutta la comunità un buon software funzionante: mentre l'utente non vuole problemi con un programma, il programmatore vuole una buona reputazione nella comunità. Al contrario, le aziende che vendono software senza rendere disponibile il codice sorgente, rilasciano correzioni dopo lunghi lassi di tempo, che spesso sono incluse nel nuovo prodotto che l'utente può ottenere solo dietro pagamento.

Nel mondo del software libero, tutti coloro che sono in grado di leggere e migliorare il codice sono incentivati a correggere gli errori, dato che saranno proprio loro ad utilizzare il programma; e poi, dato che a loro volta il software lo hanno ricevuto dalla comunità, si sentiranno in dovere di "ripagarla" attraverso la correzione dei bug.

- **Efficienza:** il software libero è in generale molto efficiente, cioè è in grado di "girare" anche su computer abbastanza vecchi. Ciò porta con sé due vantaggi: in primo luogo permette di risparmiare tanti soldi sull'acquisto di una nuova macchina ed evitare di contribuire a far aumentare i tecno-rifiuti, che sono tra i più dannosi per l'ambiente (chi l'avrebbe mai detto: il software libero è anche amico dell'ambiente!); in secondo luogo permette a chi possiede nuove macchine di lavorare in maniera molto più rapida. Il vantaggio sarebbe anche dei produttori di computer, a pensarci bene: potrebbero infatti produrre macchine di "basso profilo" che, equipaggiate con software libero, possono fare più o meno quello che facevano i computer vecchi.
- **Sicurezza:** il software libero è sicuro infatti, solo per fare un esempio, un utente di GNU/Linux non deve preoccuparsi di contrarre virus mentre naviga in rete o scarica la posta elettronica, semplicemente perché non esistono (o comunque sono ben pochi) virus destinati ad attaccare certi sistemi.

Il vantaggio della sicurezza deriva ancora una volta dal lavoro della comunità di programmatori, che sono sempre alla ricerca di vulnerabilità nel software da correggere. Insomma oltre a portare avanti un progetto di civiltà il software libero ha innumerevoli vantaggi pratici tra cui quelli che abbiamo descritto.

- **Gratuità, senza pirateria.** Sul piano pratico è importante considerare un altro importante vantaggio del software libero: nella maggior parte dei casi è gratuito.

Non tutto il software di cui sono disponibili i codici sorgente è gratuito. Alcune aziende infatti vendono insieme al programma anche i codici sorgente, lasciando in una certa misura l'utente libero di studiare il codice. Ma il sistema operativo GNU/Linux e quasi tutte le sue distribuzioni sono, oltre che "open source", anche "free", cioè gratuite. Le distribuzioni GNU/Linux si possono scaricare liberamente da Internet, come anche un grande quantità di software "open" e "free". Adottare il software libero è quindi conveniente anche sotto l'aspetto economico. Purtroppo i computer che sono assemblati dalla casa produttrice hanno sistemi operativi Microsoft Windows pre-installati (i computer Apple sono equipaggiati esclusivamente con sistemi operativi della casa). Ciò lascia gli utenti poco esperti privi di alternative.



Chi è in grado, invece, può assemblare il computer acquistando i vari pezzi e scegliere il sistema operativo da installare, senza essere soggetto alle costrizioni esercitate dalla posizione di monopolio di Microsoft.

Il boom di Internet ha aperto gli occhi a molti utenti, che si sono finalmente resi conto di spendere per ogni nuova macchina centinaia di dollari a causa di una licenza software che non avevano richiesto. Ultimamente grandi aziende tra cui Dell hanno deciso di offrire agli utenti la possibilità di acquistare i computer con sistemi operativi liberi pre-installati, facendo calare notevolmente i prezzi. In Europa solo in Francia gli utenti sono liberi di acquistare i computer senza sistema operativo pre-installato.

A quando questa iniziativa anche in Italia? Poiché molti privati non desiderano spendere centinaia di euro acquistando licenze per "suite" da ufficio (come *Microsoft Office*), spesso e volentieri si ricorre alla pirateria informatica. Migliaia di utenti "scaricano" copie con numeri di licenza contraffatti di software proprietario attraverso le reti di condivisione file (che tuttavia in sé non hanno nulla di illegale).

La diffusione del software libero arginerebbe anche questo fenomeno: Condividere software libero (a differenza di quello proprietario) non comporta alcun reato.

Il software libero non è esente da **svantaggi**, ovviamente.

L'utente finale deve avere una solida (o almeno basilare) competenza informatica e deve conoscere a sufficienza del sistema operativo con cui interagisce per poter proficuamente operare nell'ambito di software libero. Le grandi aziende invece vogliono esattamente il contrario: la loro politica è rivolta a creare ed allargare un tipo di utenza "ignorante", che si abbandoni alla guida rappresentata dal produttore.

I sistemi operativi e i programmi proprietari che eseguono operazioni in totale autonomia non vogliono il bene dell'utente per motivi economici: più un software sarà "facile", più l'utenza sarà invogliata ad acquistarlo.

Come si è detto in precedenza, la tecnologia non è (e non deve diventare) esclusivamente un mezzo di aiuto per svolgere molto rapidamente operazioni difficili o noiose: è uno stimolo per la mente umana, un mezzo per mettere alla prova la creatività, l'inventiva, la capacità di ragionamento e, ovviamente, una valida assistente nella vita lavorativa e quotidiana. Possiamo definire i computer "macchine stupide", ma di certo non possiamo estendere la definizione agli inventori e programmatori.

Il software libero, a differenza di quello proprietario, oltre a creare una benefica cultura del dono e della condivisione, stimola il ragionamento e la creatività.

Naturalmente non bisogna essere spaventati da queste considerazioni se si è deciso di intraprendere un viaggio nel mondo del software libero: tutte le distribuzioni GNU/Linux attuali hanno una interfaccia grafica molto semplice, funzionale e gradevole che rende, specie per l'ex utente Windows, meno "traumatico" il passaggio (è il caso della famosa distribuzione "Ubuntu", nota per la sua grafica e per la sua semplicità di utilizzo attraverso interfaccia grafica).

## 7. IL SOFTWARE LIBERO NELLA SCUOLA E NELLA PUBBLICA AMMINISTRAZIONE

Prima di discutere intorno a questo punto, credo sia necessario fare una precisazione. "Scuola" è diverso da "didattica". Infatti, essere interessati ad introdurre il software libero nella "didattica" risulta differente dal voler introdurre il software libero nella "scuola" in generale (quindi in amministrazione, segreteria...).

Analizzeremo perciò la questione sotto due aspetti differenti, a causa delle implicazioni notevolmente differenti. Sarebbe inoltre utile chiarire qual è la situazione attuale nella scuola italiana. In questi anni gli insegnanti si sono visti scippare l'esercizio della libertà di insegnamento ad opera del software proprietario monopolistico. Infatti lo scopo di creare una cultura scientifica e tecnologica nei ragazzi era ben lontano dalle intenzioni delle "major": le casse dell'azienda di Redmond sono state notevolmente rimpinguate dai generosi ed ingenui acquisti del settore scuola. È stata creata una "sottocultura tecnologica perpetrata dall'uso di questo software", scrive Antonio Bernardi, responsabile del progetto *Software libero nella scuola*; negli anni si è insomma creata una "visone magica della tecnologia". Gli studenti sarebbero sempre più assimilabili a "passivi consumatori di tecnologia", piuttosto che ad esseri umani critici e creativi. "Potremmo dire", scrive ancora Bernardi, "che crea maggior consapevolezza di libertà *OpenOffice* (la suite da ufficio sorella di Microsoft Office, ma in versione open source) su Windows, nel ruolo di alternativa al monopolio, che con un router Linux in una rete dove tutti usano Windows e Microsoft Office. Al limite si perpetuerebbe la sottocultura informatica: Linux come stampella di Windows". Ciò significa che se non si desidera abbandonare completamente il software proprietario, si potrebbero quanto meno inserire elementi di "concorrenza", che servano da monito per gli studenti, per far capire che un'alternativa esiste. Il fatto è che il software proprietario è venduto come se fosse il migliore in assoluto. Il che non è sempre vero: esiste un gran numero di software liberi che svolgono i loro compiti come, se non meglio, dei software proprietari. Osserva Bernardi: "si riesce ad insegnare ad andare in bicicletta anche su una vecchia bici".

Il software libero ha perciò tutti gli strumenti per essere introdotto nella didattica. Bisogna innanzitutto partire dal lato "client", cioè da quello dell'utente: il primo passo è installare su tutti i computer programmi come *OpenOffice* o *LibreOffice*, in modo da restituire agli utenti due importati libertà, cioè quella dell'uso del software, e quella della copia dello stesso. Il passaggio ad *OpenOffice* o *LibreOffice* fornisce a docenti e studenti l'importante messaggio che esiste una alternativa al monopolio di Microsoft Office. Ma se il software libero, oltre a non essere soggetto a costi, ha anche innumerevoli vantaggi pratici, perché non viene adottato in ambito scolastico o di pubblica amministrazione in generale?

Alcune indagini promosse dalla Comunità Europea rivelano che l'utilizzo di software open source nella pubblica amministrazione è concentrato nella fascia "server": ciò significa che a livello di server sono molto diffuse le combinazioni Linux più Apache Web Server. A livello europeo, hanno sorpreso le cifre riguardanti l'uso di suite da ufficio open source: anche se con dimensioni più contenute rispetto ad Apache, questi software risultano in espansione. Ben differente è, come vedremo la realtà italiana. Alcuni ricercatori dell'Osservatorio "Software open source nella Pubblica Amministrazione" dell'Università di Bologna hanno effettuato una rilevazione automatica circa il tipo di web server utilizzato dalle pubbliche amministrazioni locali italiane. Risulta che Appena il 40% dei server utilizza Apache (contro il 60% europeo). Secondo l'Osservatorio dell'Università di Bologna, poiché Apache rappresenta una delle applicazioni di più ovvio e diffuso utilizzo, una organizzazione che non possiede neppure un server open source, "mostra in media di possedere una scarsa propensione all'uso del software open source all'interno di tali amministrazioni.

Perciò i dati della rilevazione, pur non riferendosi esplicitamente all'uso di software libero in generale, valutano la propensione dell'amministrazione pubblica all'utilizzo del software libero.

L'adozione del software libero nella pubblica amministrazione comporta, come abbiamo ripetuto più volte, un risparmio iniziale in termini di licenze; il risparmio deriva anche dagli eventuali servizi di assistenza, installazione, gestione, manutenzione, aggiornamento. Disporre dei codici sorgente dei propri programmi è inoltre un notevole vantaggio per la sicurezza. La ricerca di vulnerabilità ad attacchi di *hacking* o di *backdoor* nei sistemi proprietari è affidata esclusivamente ai produttori; con il software libero sono più semplici e sicuri i controlli. Indiscutibilmente il software libero è meno vulnerabile ad attacchi dall'esterno da parte di malintenzionati. L'esempio fornito da Cuba è emblematico: la pubblica amministrazione dell'isola ha deciso di adottare il software libero. Ma è importante sottolineare che anche l'amministrazione Obama si sta muovendo verso l'adozione del software libero, sollecitato da aziende produttrici di software open source come *Sun Microsystems*. Il software libero renderebbe la PA indipendente da un produttore di software; inoltre i documenti digitali diverrebbero interoperabili, poiché sarebbero basati su formati aperti e standard (in questo modo anche sistemi diversi sarebbero in grado di condividere gli stessi dati). I "formati testo aperti standard" hanno il beneficio della persistenza, Un problema al contrario presente nel software proprietario. Mentre, infatti, tra una versione e l'altra del software, la casa produttrice potrebbe (ma succede molto spesso) cambiare lo standard del formato (che di conseguenza diverrebbe inutilizzabile su sistemi dotati di software più vecchio), i documenti creati con software libero sono persistenti perché non sono soggetti ad "invecchiamento" (tra l'altro è importante sottolineare che il software libero è molto spesso in grado di gestire i formati "proprietary" - cioè "chiusi").

## APPENDICE

### *Codici ribelli, la libertà corre sulla tastiera – il Manifesto – 20 Marzo 2003*

*di Arturo Di Corinto*

Intervista a Richard Stallman, fondatore della «Free Software Foundation» e ispiratore del sistema operativo per computer Gnu-Linux.

**Bill Gates è stato invitato a Roma dal presidente del senato Marcello Pera per parlare di globalizzazione e Berlusconi ha chiesto al fondatore della Microsoft lumi su come favorire l'informatizzazione della Pubblica Amministrazione italiana. Che ne pensi?**

Berlusconi e Gates sono spiriti affini: entrambi cercano di arricchirsi controllando la distribuzione dell'informazione. Credono di farlo in maniera diversa e di non competere l'uno contro l'altro, ma si sbagliano. Quelli che non usano software libero sono dominati dai proprietari del software, perciò l'Italia sarà più debole nel lungo termine se usa il sistema operativo Windows sviluppato dalla Microsoft.

**È per questo motivo che sei considerato la nemesis di Bill Gates?**

È sbagliato concentrarsi troppo sul singolo produttore di software. Nessuna particolare azienda è infatti il «Grande Satana». Noi stiamo cercando di risolvere un problema che è assai più grande di Microsoft: il problema del software non libero.

**Il parlamento italiano ha avviato una discussione sull'introduzione del software open source nella Pubblica amministrazione – senza una chiara menzione al Free Software – e il Ministro Italiano dell'Innovazione, Lucio Stanza, ex amministratore delegato di IBM Italia, ha istituito una commissione di studio su questo tema. Secondo te è un bene adottare il software libero nella Pubblica amministrazione italiana?**

Col software libero gli utenti controllano ciò che fanno col computer. Le agenzie governative possono beneficiare della stessa libertà esattamente come i singoli utenti. Ma c'è un'altra importante ragione per cui i governi dovrebbero adottare il software libero. Governare un paese nella giusta direzione è il compito primario dei governi e ogni agenzia di governo dovrebbe considerare questo fatto.

**La Nato, l'Inghilterra, la Russia e la Cina hanno deciso di aderire al «Microsoft Government Security Programme». Che ne pensi?**

Questi governi cominciano a riconoscere l'importanza della libertà del software ma Microsoft gli sta solo offrendo un pezzetto della libertà che il free software garantisce. In base a quanto stabilisce la *Microsoft Government Security*, i governi possono solo leggere il codice sorgente dei programmi informatici e non possono usarlo liberamente. Spero che essi capiscano che è insufficiente. La licenza Microsoft per il codice è basata sul *nondisclosure agreement*, cioè sulla promessa di non condividere con nessun altro ciò che impari da quel software. È un accordo non etico che una persona onesta dovrebbe rigettare da un punto di vista morale. Col software libero invece sei libero di condividere e scambiare la tua conoscenza. In ogni caso *Microsoft* si offre di mostrare il codice sorgente solo a pochi governi e poche aziende selezionate. Perciò questa strategia non ha alcun effetto su me e te.

**Cos'è il free software?**

Free (libero) software significa che tu come utente sei libero. Hai la libertà di usare il programma che vuoi; la libertà di studiare il codice sorgente dei programmi informatici, di cambiarlo e adattarlo ai tuoi desideri; hai inoltre la libertà di redistribuirne copie ad altri, la libertà di pubblicarne una versione nuova o modificata. Se non sei un programmatore e non sai come farti le cose da solo puoi pagare qualcuno che lo faccia per te.

Il free software consente agli utilizzatori la libertà di formare una comunità. Se non sai programmare ma cucini, dovresti trovare queste libertà familiari. Le libertà che penso debbano appartenere alla produzione e all'uso del software libero sono le stesse libertà che hanno i cuochi nell'usare una ricetta. Immagina se il governo cercasse di impedirti di cambiare ricetta o dicesse che se scambi e condividi le ricette sei un «pirata». Come minimo ti offenderesti.

Venti anni fa, come programmatore di computer, è esattamente ciò che ho visto succedere: il software proprietario (non free) cominciava a sostituire il free software. A me e agli altri utenti veniva sottratta una libertà. Pensando al futuro mi

dissi che avrei rifiutato di vivere in quel modo. Ma qual era l'alternativa? Dovevamo crearne una. Dovevamo creare un nuovo continente nel cyberspace, un posto dove vivere in libertà. E questo è esattamente ciò che abbiamo fatto.

## **Qual è la differenza fra il software «open source» e il «free software»?**

Il movimento del software libero è una campagna per la libertà. Vogliamo essere liberi di controllare i nostri computer e liberi di aiutarci l'un l'altro. Insistiamo sul fatto che il software dovrebbe essere libero e se non c'è il software libero per fare una certa cosa, lo scriviamo. Il nostro idealismo e la nostra determinazione sono il motivo per cui oggi esiste GNU/Linux. Negli anni 90 molti adottarono GNU/Linux per i suoi vantaggi pratici senza apprezzare il suo valore di libertà. Nel 1998 alcuni lanciarono il «movimento dell'open source». Un movimento simile al nostro ma la cui filosofia è molto diversa dalla nostra. Quelli dell'«open source» non affrontano il problema da un punto di vista etico; non dicono mai che gli utilizzatori hanno il diritto morale di ridistribuire e modificare il software. Parlano solo dei vantaggi pratici. Quando il movimento dell'open source convince la gente a sviluppare programmi liberi contribuisce alla crescita della comunità. Li apprezziamo e non li criticiamo. Tuttavia si tratta di idee che non forniscono un fondamento sufficientemente forte per la nostra comunità. Per difendere la libertà bisogna prima imparare ad apprezzarla.

## **Allora ti consideri un hacker?**

Ero un hacker ante litteram nel 1960, ma ho appreso questo termine solo quando cominciai a lavorare al *Massachusetts Institute of Technology* (MIT) nel 1971. Un hacker è uno che fa le cose con uno spirito di gioiosa bravura. Quello che spesso facciamo è programmare. Ma non solo. Ad esempio Guillaume de Machaut, famoso compositore, ha scritto un'aria musicale chiamata *Ma Fin Est Mon Commencement*, che è una sorta di palindromo ma anche un pezzo di ottima musica. Era un ottimo hack! Il sistema operativo Gnu/Linux è anch'esso un ottimo *hack*! Quanto sia buono un *hack* lo giudichi da solo.

## **Cosa è Gnu/Linux e perché è sbagliato chiamarlo Linux?**

Nel 1984 cominciammo a sviluppare un sistema operativo libero di nome GNU. Nel 1991 il sistema era quasi completo, gli mancava solo il *kernel* (una delle parti più importanti dei programmi per computer). All'epoca Linus Torvalds, studente finlandese, scrisse il *kernel* e nel 1992 lo distribuì sotto la GNU General Public License. Era software libero. GNU e Linux insieme formano un sistema operativo completo e totalmente libero. Linux è una parte importante del sistema nel suo insieme ma il contributo maggiore (lo GNU) era precedente al suo sviluppo. Perciò non diamo molto credito a quelli che usano la parola Linux per indicare l'intero sistema operativo di cui Linux è una parte. Chiamandolo GNU/Linux invece si rende giustizia a entrambi. Non ti sembra più giusto?

## **Chi è invece *Saint GNUcius*?**

*Saint GNUcius*, della Chiesa di *Emacs*, è il modo in cui ironizzo su me stesso e sulla mia immagine di guru (*Emacs* è il mio editore di testi, che per molti è diventato un modo di vivere, quasi una religione). La sua aureola è un vecchio hard disk dei tempi in cui erano grandi come lavatrici. *St GNUcius* si appella ai fedeli per esorcizzare dai computer diabolici sistemi operativi proprietari e sostituirli con sistemi operativi che siano liberi. *Saint GNUcius* è un *hack*. A giudicare dalle risate della gente quando lo introduco è un *hack* piuttosto ben riuscito.

## **Nei tuoi interventi, sostieni che il free software più affidabile di quello non free. Perché?**

L'esperienza ci dimostra che i sistemi GNU/Linux sono altamente affidabili, molto di più di Windows. Sui motivi di ciò possiamo solo speculare. Alcuni sostengono che dipende dalle migliaia di programmatori che cercano e risolvono i bachi (gli errori, n.d.r.) di funzionamento del software. Ma anche perché i programmatori mettono molta cura nello scrivere un codice sorgente chiaro e pulito quando si aspettano che il loro lavoro sarà letto da altre migliaia di persone. Ma questa cosa non è per me la più importante. Io uso software libero per essere libero e sceglierei di usare software libero anche se fosse meno affidabile di quello che è in effetti.

## **Molti computer, soprattutto i server Internet, usano «free software». Invece i computer a casa per la maggior parte usano software non libero (soprattutto Microsoft). Pensi che il free software cambierà il mercato domestico?**

Il business del «Free software» ha un mercato ma il movimento del software libero è un movimento sociale, non un business. Noi pensiamo in termini di cittadinanza, non in termini di mercato. Quando abbiamo cominciato a sviluppare il sistema GNU volevamo qualcosa di simile allo Unix (altro sistema operativo), e nel 90' abbiamo avuto quello che volevamo: Gnu/Linux. Potente come lo Unix e difficile da imparare come Unix. Non aveva l'interfaccia grafica che molti vogliono. Questo è il motivo per cui GNU/Linux si è diffuso per i server, perché solo i più tecnici lo usavano. Adesso abbiamo interfacce utente a icone e applicativi grafici per l'ufficio. Non sono in grado di predire il futuro ma queste nuove opportunità danno a GNU/Linux la possibilità di diventare assai popolare nell'home *computing*.



**Il free software può essere venduto, ma molti sostengono che non riesce a garantire il reddito necessario per vivere dignitosamente. Qual è la tua opinione?**

Ci sono centinaia di persone, forse migliaia, che vivono sviluppando software libero e adattandolo alle esigenze dei loro clienti. La libertà che è incorporata nel software libero ti dà la possibilità di assumere qualcuno in grado di modificare il programma in base alle tue necessità. Non sappiamo se passando al free software si crea o meno occupazione, di certo non la diminuisce. Il free software si oppone a un modello commerciale che forza l'utente a pagare l'uso del programma. Il free software favorisce infatti un modello commerciale basato sull'adattamento e l'estensione del software a più clienti. In definitiva potrebbe favorire l'occupazione molto di più del software non libero. Ma non lo si può sapere prima di farlo. Quello che è evidente è che il software non libero favorisce le compagnie multinazionali e arricchisce poche persone mentre il software libero incoraggia l'industria locale del software e favorisce gli investimenti sul territorio.

**Dopo gli attentati dell'11 settembre molti paesi adottano misure restrittive delle libertà in rete e le cose potrebbero peggiorare a causa del conflitto iracheno. Nel tuo sito personale ([www.stallman.org](http://www.stallman.org)) sostieni che patriottismo significa difendere le libertà civili....**

Gli attacchi dell'11 settembre sono la scusa più comoda per molti governi, incluso il regime di Bush, per attaccare la libertà della loro stessa gente. Il governo americano si arroga il potere di imprigionare arbitrariamente chiunque senza un processo, qualcosa che non può succedere in un paese libero. Il terrorismo rappresenta un pericolo reale, ma il pericolo è maggiore quando è il nostro stesso governo ad attaccare la nostra libertà. Osama bin Laden può volere distruggere gli Stati Uniti, ma non può riuscirci. Bush invece può veramente distruggere gli Stati Uniti come terra di libertà. Per questo io cerco di richiamare l'attenzione degli americani sul pericolo che Bush rappresenta. Se ami il tuo paese, aiutalo a rimanere libero.

## Perché scegliere il software libero nella didattica?

di Antonio Bernardi - responsabile del progetto "Software libero nella scuola"

Da più parti vi è la consapevolezza che il software, utilizzato nell'insegnamento con le nuove tecnologie o delle nuove tecnologie, corrisponda ad un libro di testo, inteso in senso lato (materiali didattici, ecc.);

Nel mercato se ne sono affermati due:

1. Quello riconducibile al **software proprietario**
2. Quello riconducibile al **software libero**.

La diversità sta nel tipo di "valori" di cui sono portatori:

- Il primo è portatore dei valori tipici della cultura del monopolio privato che per sua natura non ama il confronto (la concorrenza) e si impegna il più possibile a soffocarla al fine di impedire all'utente (consumatore) qualsiasi "scelta". Per questa cultura l'utente non deve essere messo in condizioni di scegliere.

- Il secondo libro di testo, il software libero, è viceversa portatore di valori di libertà compresa la libertà di scelta. Per questa cultura l'utente (consumatore) non solo deve essere messo in condizioni di poter scegliere, ma le sue stesse licenze d'uso sono state progettate per stabilire un rapporto di uguaglianza tra "produttore del software" e "utente del software".

Ma per scegliere il libro da usare, questa premessa potrebbe non essere convincente. Vediamo allora perché un docente dovrebbe scegliere il software libero per il suo lavoro.

### 1. Non discrimina economicamente.

Libertà economica

- **Software a costo zero** o comunque a un prezzo contenuto (costo connessione Internet, costo cd-rom, ecc.); fine della necessità di sottostare a licenze proprietarie. Il software libero non discrimina i soggetti su base economica ma seleziona su base meritocratica; la gratuità delle licenze d'uso è conforme ai criteri stabiliti dalle varie circolari sull'adozione dei libri di testo che stabiliscono un tetto di spesa, a seconda del tipo di scuola.
- **Hardware a costo zero** (quando si trova negli scantinati) o comunque a un prezzo contenuto; possibilità di riutilizzo di hardware "obsoleto". Il lato server e l'utilizzo didattico con l'interfaccia a caratteri può utilizzare "vecchio" materiale (486, primi *pentium*, ecc.).

Conseguenze della libertà economica:

- Le scelte didattiche non sono più condizionate dall'aspetto finanziario; i dirigenti scolastici e i docenti hanno la libertà di scelte indipendenti dai centri del potere; i dirigenti e docenti sono liberati dal peso delle procedure burocratiche relative agli acquisti; gli studenti possono fornirsi del software senza gravami economici per le famiglie.

### 2. Incentiva la copia educando alla legalità

Libertà di copia=libertà di (ri)produzione

- Con la disponibilità dei sorgenti la **libertà di copia** (riproduzione) è un fatto naturale e incoraggiato; questo permette, sia all'insegnante che all'allievo, l'assoluta libertà di copia, fondamento dello scambio della conoscenza; il docente non deve ricorrere alla pirateria informatica per svolgere la normale attività educativa.
- Il diritto di copiare il software libero corrisponde alla **libertà di stampa e di parola**. Si realizza il principio della condivisione della conoscenza, dove il sapere libero è un diritto di tutti; a scuola può essere distribuito agli studenti liberamente, realizzando un momento fondamentale della didattica: la trasmissione di cultura/conoscenza.
- Il software libero **educa alla cultura della legalità** e rende consapevole l'utente, insegna a distinguere il software che si può copiare, da quello la cui copia è vietata e combatte il malcostume della copia illegale (pirateria del software).

### 3. Rispetta la libertà di scelta

- Nel mercato vi sono **parecchie aziende e distribuzioni** basate sul software libero: possiamo dire che è un mercato in "libera concorrenza". In GNU/Linux, oltre alle varie distribuzioni, vi è una pluralità di produttori di software: es. KDE, GNOME, server web, server di database, linguaggi, ecc.; un insegnante è libero di scegliere quella più idonea ai suoi scopi didattici. Ma c'è di più: un docente e la scuola più in generale possono crearsi la propria distribuzione, come ad esempio: eduKnoppix, nanoLinux, DemoLinux, ecc.
- Non è il mercato (monopolistico) che impone il prodotto secondo i suoi interessi (più o meno legittimi) ma è la **scuola che può scegliere** in base a criteri didattici (che spesso non coincidono con quelli del monopolista).

In definitiva vi è ampia scelta di prodotti e soluzioni per un insegnante permettendogli l'esercizio della libertà d'insegnamento.

### 4. Non soffoca la libertà di parola

Libertà di parola = Libertà di modificare e di migliorare il software

- Le licenze libere (GNU General Public License, ecc.) permettono la modifica e il miglioramento del software; si crea una comunità scientifica in cui ci si aiuta reciprocamente, senza discriminazioni, per lo sviluppo e il miglioramento; questo sviluppo non è condizionato esclusivamente da esigenze di mercato (profitto).
- Da questo punto di vista il software libero genera un nuovo umanesimo, una rivoluzione nella produzione del software. Il sapere informatico passa dalle mani di pochi alla comunità scientifica, è una questione di civiltà.

### 5. Educa alla cultura galileiana

Libertà di studio e di verifica (principio di trasparenza e di affidabilità-garanzia)

- La disponibilità dei sorgenti permette lo **studio del software**, di "aprire il cofano", permette di imparare, di crescere. I sorgenti permettono la verifica, che è un aspetto fondamentale della scienza moderna e del metodo scientifico; senza verifica c'è il dogma; il principio di autorità (monopolio); siamo garantiti dalla comunità scientifica: il software libero funziona meglio (**affidabilità**) perché gli errori vengono trovati prima e più efficacemente rispetto al software proprietario. Con i sorgenti abbiamo la possibilità di "**sporcarci le mani**" in laboratorio e risolvere i problemi che di volta in volta si pongono.
- Il software libero educa alla **cultura sperimentale di laboratorio** e non alimenta negli utenti una mentalità oscurantista (mistica). Di conseguenza il software libero ha un alto valore formativo perché permette di appropriarsi dei meccanismi conoscitivi completi del software, rende consapevole lo studente delle procedure che svolge, non lo appiattisce su interfacce grafiche, non alimenta luoghi comuni, stereotipi, pregiudizi, atteggiamenti fideistici, ecc.

### 6. Incentiva l'economia locale

Libertà di redistribuzione e di fare affari

- La licenza GPL permette di **vendere il software**: il codice sorgente e la licenza sono gratuiti ma si pagano i servizi attorno a questo prodotto.
- L'uso di software libero permette una **libera concorrenza** e lo **sviluppo di economie locali**: essendo il sorgente a disposizione di chiunque esiste una vera concorrenza in quanto non vi sono barriere di accesso a livello software. Nel mercato vi sono molte società che producono e commercializzano distribuzioni diverse di GNU/Linux e di prodotti su GNU/Linux.
- Le scuole stesse utilizzando GNU/Linux contribuirebbero ad alimentare questa economia locale.

## Conclusioni

Abbiamo avuto modo di scoprire la storia del software libero, e i suoi pro e contro. Abbiamo visto perché adottare il software libero e possibilmente il "pinguino" a livello didattico o di pubblica amministrazione ha i suoi vantaggi. Indiscutibilmente molti quesiti sono dovuti al fatto che molti utenti sono in grado di risolvere eventuali problemi e rilasciare aggiornamenti frequentemente.

La libera circolazione della conoscenza, nell'ambito informatico ha permesso a grandi comunità di sviluppatori di scambiarsi in rete idee e file, senza un controllo o una autorità centrale, per creare software complessi.

Il passaggio al software libero nel mondo della didattica e della pubblica amministrazione potrebbe essere realtà se solo da parte degli enti competenti vi fosse maggiore sensibilità ed interessamento, e se tra i fruitori di esso vi sia una sufficiente competenza.

## Fonti:

- Articolo “GNU/Linux: la rivoluzione culturale”, in <http://www.megalab.it/4394/gnu-linux-la-rivoluzione-culturale>.
- Tesi di Simone Aliprandi “Open source e opere non software”, Tesi di laurea in giurisprudenza dell'Università di Pavia discussa nell'anno accademico 2002-2003.
- Sito: [www.tecnologieinformatiche.tk](http://www.tecnologieinformatiche.tk) di Salvatore Cianciabella.
- *Revolution OS*: documentario statunitense del 2001, diretto da J.T.S. Moore. Nel film si ripercorrono venti anni di storia di GNU, Linux, del software libero e dell'open source.
- *Revolution OS II*: questo libro e questo film sono idealmente il seguito di *Revolution OS*. Quel film era stato realizzato negli Stati Uniti, questo è stato girato in Italia sotto la direzione di Arturo Di Corinto. Vi si possono vedere i protagonisti del mondo del software libero e dell'open source ripercorrere le tappe della "rivoluzione" negli ultimi anni, raccontare le esperienze internazionali, in Sudamerica e in Europa, le pressioni del mercato e le iniziative di valore sociale, il ruolo del software libero nella pubblica amministrazione e nell'istruzione. Accanto al film, il libro raccoglie un gruppo di saggi che approfondiscono gli stessi temi, del curatore Arturo Di Corinto, di Raffaele Meo, Richard Stallman, Lawrence Lessig, Eric Kluytens.
- Intervista a Richard Stallman, fondatore della «Free Software Foundation» e ispiratore del sistema operativo per computer Gnu-Linux in <http://dicorinto.it/testate/il-manifesto/codici-ribelli-la-liberta-corre-sulla-tastiera-il-manifesto-20-marzo-2003/>
- "Perché scegliere il software libero nella didattica" Intervista ad Antonio Bernardi, responsabile del progetto “Software libero nella scuola”, in <http://linuxdidattica.org>.