

Penetriamo nuovamente in epoche che non aspettano dal filosofo né una spiegazione né una trasformazione del mondo, ma la costruzione di rifugi contro l'inclemenza del tempo. *Nicolás Gómez Dávila*

Questo numero.

Ci sono più cose in cielo e in terra, Orazio, di quante ne sogni la tua Economia.

Questo numero ha un obiettivo ambizioso: far comprendere anche ai meno pratici l'importanza di idee e fatti che, in questi ultimi anni, stanno rivoluzionando il mondo dell'informatica.

Prendiamo il cosiddetto *browser*, il programma che utilizziamo per navigare in Internet: nel luglio 2003 l'84,9% degli utenti utilizzava una delle varie versioni di *Explorer*, il "software proprietario" (= protetto da brevetto e, per quanto possibile, segreto) di *Microsoft*. Esattamente sette anni dopo *Explorer* è crollato al 30,4%, mentre il 69,6% degli accessi (e la tendenza è all'ulteriore crescita) avviene con software "open source" (= non segreto) e libero come *Firefox*.

Dopo il *browser* il programma forse più utilizzato è quello di scrittura: in questo campo *Microsoft Word* è ancora dominante² ma la caduta pare inarrestabile.

Cosa c'è dietro questa crescita del software di pubblico dominio, gratuito, libero, e dunque a disposizione (non è retorica) dell'umanità compresi i paesi poveri? Due fattori: una grande comunità che con passione e spirito pragmatico ha permesso di esistere a qualcosa che, per le leggi di quella pseudoscienza che è l'Economia³, doveva risultare impossibile, e una singola persona che ha fatto, rischiando

in proprio, da primo motore: **Richard Matthew Stallman**. Il testo di una sua conferenza del 1998, che pubblichiamo in forma ridotta, dovrebbe essere sufficiente per farne comprendere motivazioni iniziali e linea di pensiero. Ci sarà poi tempo per una riflessione sul significato economico e sociale di quello che sta succedendo anche grazie a lui. 🦅

Agli attenti lettori non sfuggirà che Stallman è anche un attivista politico della sinistra radicale e che molte altre sue idee e iniziative sono in contrasto con il nostro sentire, ma ciò non toglie nulla al valore della sua impresa in materia di *copyright*.



Parla Richard Stallman.

Università di Parigi 8, 10 novembre 1998⁴.

Posso descrivere l'idea del software libero in tre parole: libertà, uguaglianza e fraternità.

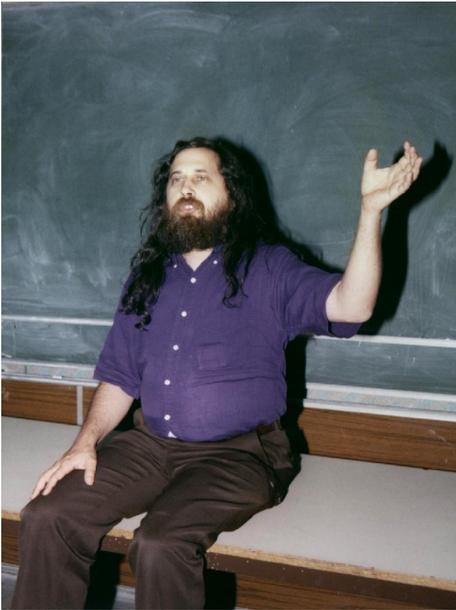
- **Libertà:** La libertà di fare copie, di diffonderle e di darle agli altri, sia agli amici, alle persone che lavorano con voi, che agli sconosciuti. La libertà di operare dei cambiamenti che adattino il

¹ Secondo i dati dell'autorevole sito www.w3schools.com.

² In Italia le stime attuali parlano di circa l'80% del mercato contro il 18% di *OpenOffice*, prodotto gratuito ormai maturo.

³ La stessa che si è dimostrata così cieca e ottusa di fronte alla grande crisi che stava arrivando.

⁴ La conferenza fu organizzata dall'associazione *Avril*. La traduzione (ridotta) dalla trascrizione di Frédéric Couchet e Sébastien Blondeel, è di Gabriella Rouf e Francesco Borselli, con la supervisione scientifica di Davide Dell'Aquila.



Richard Stallman durante la conferenza.

software ai vostri bisogni. La libertà di pubblicare versioni migliorate in modo che la società intera ne riceva i benefici.

- *Fraternità*: Col software libero, tutti sono incoraggiati a cooperare, ad aiutarsi gli uni con gli altri.
- *Uguaglianza*: Ognuno possiede le stesse libertà: non c'è un padrone che detiene tutto il potere mentre il resto del mondo è completamente impotente, completamente limitato nell'utilizzo del software.

♣ GLI INIZI: UNA COMUNITÀ DI CONDIVISIONE DEL SOFTWARE.

Ma come sono arrivato all'idea del software libero? Quando sono diventato informatico, negli anni '70, ho cominciato a lavorare nel laboratorio di intelligenza artificiale del *MIT*, *Massachusetts Institute of Technology*. Ero impiegato come programmatore di sistema e il mio compito era migliorare il sistema in qualsiasi modo [...]. Operavamo in una comunità di condivisione del software che comprendeva varie università e che talvolta collaborava direttamente con i produttori di

computer, come *Digital*.

In quegli anni, se vi capitava di vedere sullo schermo di un altro qualcosa di interessante, potevate chiedergli: "Che programma è?" ed egli diceva "Ah, è il nuovo programma *Tal de' Tali*, il codice sorgente è in questa cartella...". E voi potevate lanciare il programma, utilizzarlo, cambiarlo, aggiungere altre funzioni, come pure tagliare dei pezzi di vecchi programmi per metterli nei nuovi che scrivevate [...]. Insomma, si poteva sempre supporre di ricevere quantomeno una cooperazione passiva. Magari non la cooperazione attiva — gli altri erano sempre molto occupati — ma almeno nessuno faceva niente per ostacolarvi.

♣ COOPERAZIONE.

Per esempio, una volta capitò che il laboratorio avesse bisogno di un assembler per il codice di *PDP11* che funzionasse sul *PD-110*. Invece di scrivere tale programma, andai ad Harvard, dove utilizzavano il sistema operativo di *Digital* ed avevano un assembler che funzionava sul sistema di *Digital*. Dunque, feci una copia del codice sorgente che non aveva segnalazione di diritti d'autore, lo portai al *MIT* ed apportai delle modifiche affinché funzionasse sul nostro Sistema Operativo [...]. Qualche anno dopo, qualcuno che lavorava alla società *BBN* ebbe bisogno di questo assembler per il loro sistema operativo, *Tenex*. Dunque, prese il codice sorgente della mia versione, fece dei cambiamenti per farlo girare su *Tenex* ed aggiungere altre funzioni.

Dato che queste altre funzioni mi piacevano, decidemmo di fare una versione combinata che funzionava sui due sistemi con degli adattamenti [...]. Facemmo poi collettivamente l'aggiornamento di questa versione per i due sistemi per parecchi anni.

Il procedimento di aggiornamento del software funzionava così: tutti i programmi erano disponibili nella nostra comunità.

♣ DIFFICOLTÀ COL SOFTWARE PROPRIETARIO.

Compresi di colpo cosa significasse utilizzare un programma di cui non si conosce il codice sorgente. Ciò accadde quando *Xerox* diede al *MIT* una stampante laser. Era la prima che vedevamo: si trattava di una fotocopiatrice adattata per essere una stampante, la prima generazione delle stampanti laser. Funzionava molto rapidamente, con punti molto piccoli, ma non era completamente affidabile. Talvolta, parecchie volte in un'ora, si bloccava. Insomma, era un po' difficile da utilizzare.

La nostra vecchia stampante aveva lo stesso problema, ma era controllata da un software libero, di cui possedevamo il codice sorgente, e a cui quindi avevamo potuto aggiungere funzionalità speciali per affrontare simili problemi.

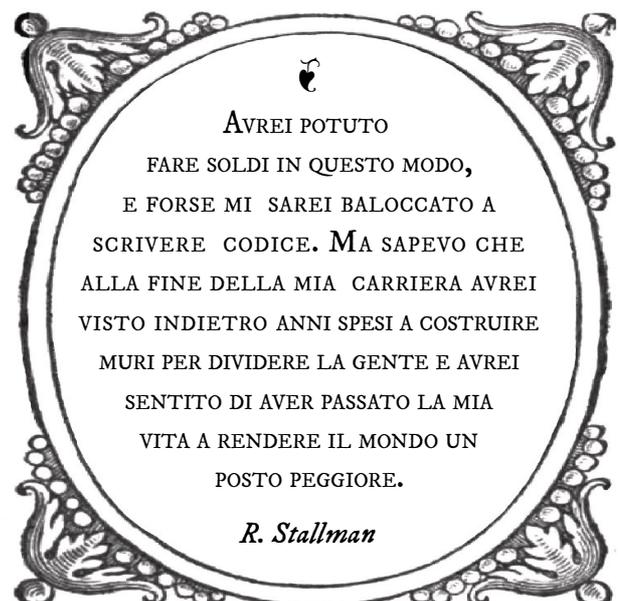
Con la vecchia stampante bisognava aspettare molto tempo per la fine della stampa dei documenti. Così, una volta terminato il processo, il programma di controllo della stampante mandava un messaggio al *PDP10* per indicare che la stampa era finita, e sullo schermo dell'utente appariva la scritta "il fascicolo è completato". In questo modo potevamo recuperare i documenti senza attesa supplementare, senza aspettare, sapendo che il processo era terminato. Fui io stesso ad aggiungere un'altra funzione: quando la stampante si piantava ed era bloccata, il programma di controllo mandava il messaggio "la stampante è bloccata", ed il *PDP10* trasmetteva questa scritta agli schermi degli utenti [...]. Se avevate ricevuto questo messaggio, sapevate che dovevate rimediare al problema

alla svelta, perché non volevate aspettare che qualcun altro lo correggesse. Dunque 30 secondi dopo che il problema si era manifestato arrivavano due o tre colleghi a risolverlo, e la stampante funzionava quasi sempre a dispetto dei suoi difetti. Aggiungemmo insomma il *feedback*, includendo l'intervento dell'utente nel sistema, arrivando così ad un funzionamento affidabile.

Con la nuova stampante non fummo in grado di farlo perché *Xerox*, non fornendoci il codice sorgente, ci impedì di apportare cambiamenti al programma. Eravamo ben capaci di creare quelle funzioni, ma *Xerox* ci aveva bloccati volutamente: eravamo prigionieri di un software che funzionava male (e che continuò a farlo per parecchi anni). Bisognava aspettare anche un'ora con una stampante in grado di stampare in pochi minuti. Essendoci abituati al fatto che era improbabile che la stampa si completasse in un più breve periodo di tempo, pazientavamo tutti per un'ora. Era disgustoso.

♣ UN PROBLEMA MORALE.

Un giorno sentii dire che un ricercatore della *Carnegie Mellon University* possedeva una copia del codice sorgente del programma per la



stampante laser. Quindi andai a Pittsburg, nel suo ufficio, a chiedergli se poteva darmene una copia. Mi rispose: “No, ho promesso di non darlo.” Mi mancarono le parole, uscii senza dire nulla, perché non sapevo come reagire ad un’azione così vergognosa.

Fu un danno per il laboratorio, perché non avendo ricevuto il codice sorgente di questo programma non avemmo modo di correggere i problemi, e la stampante funzionò male per tutti gli anni in cui l’utilizzammo, ma per me si rivelò, in modo paradossale, una buona cosa, perché mi permise di comprendere quanto fosse perverso questo meccanismo.

Capii che questo ricercatore non l’aveva fatto soltanto a me, non aveva rifiutato la sua collaborazione solo a me, ma l’aveva fatto anche a lei [indica uno nel pubblico], aveva negato di dare una copia anche a lei [indica un altro nel pubblico], probabilmente a voi tutti in questa sala. Aveva tradito quasi tutti per ricevere qualche cosa per se stesso.

Aveva fatto la promessa di rifiutare agli altri ciò che voleva per sé, e questa è l’immoralità fondamentale: rifiutare agli altri qualche cosa di utile che si desidera per sé. Ha tradito tutti firmando un contratto di non diffusione. Fu la prima occasione in cui incontrai un tale contratto, ed è stato positivo per me averlo sperimentato in quanto vittima perché così mi resi conto che esistevano delle vittime.

La maggior parte delle persone scopre questi contratti quando è invitata a firmarli, e così può dirsi che non ci sono vittime: “Non faccio del torto a nessuno. Se non lo faccio io, qualcun altro lo farebbe al posto mio; e non l’avranno mai comunque, dunque non perdono niente”. Tutte cose che uno può dire a se stesso per ignorare di fare qualche cosa di immorale.

Tuttavia io non potevo farlo, perché sapevo

vo cosa vuol dire essere vittima, sapevo bene che può costare caro agli altri firmare un tale contratto, e dunque, quando qualcuno mi ha invitato a firmare un contratto simile, ho detto: “No, non posso farlo, la mia coscienza è contro, perché se firmo questo contratto, farò qualche cosa di male agli altri, ed io non voglio essere colpevole. Dunque, questo software, grazie mille, ma dico di no. Lo desidero ma non posso accettarlo in questi termini.”

♣ DI FRONTE AD UN BIVIO.

Qualcos’altro accadde quasi nello stesso periodo, all’inizio degli anni ’80: *Digital* interruppe la fabbricazione dei *PDP10* ed il nostro sistema operativo divenne superato, perché scritto in assembler per quel tipo di macchine. Era completamente inutile per qualunque altro computer, e non c’era altro sistema libero. Dunque, per comprare una macchina nuova e cominciare ad utilizzarla, bisognava acquisire un sistema proprietario attraverso un contratto di non-diffusione: non c’era un procedimento etico per utilizzare un computer senza tradire il resto del mondo.

Che cosa potevo fare? Era una scelta difficile. Avrei potuto adattarmi, continuare la mia carriera di informatico con un sistema proprietario che firmava dei contratti di non-diffusione e, così, avrei potuto guadagnare del denaro, avrei potuto divertirmi a scrivere codice. Tuttavia, alla fine, sapevo che avrei dovuto dirmi: “Ho passato la mia vita a costruire dei muri per la divisione tra le persone, per separare le persone, per impedire la cooperazione tra persone”, e sarebbe stata una vita sprecata. Era la scelta facile.

C’era anche una scelta un po’ faticosa: smettere di essere informatico, smettere di utilizzare i computer. Triste, ma morale. Perché ci sono attualmente parecchi milioni

di cittadini che non fabbricano software, che non utilizzano software o computer. È possibile vivere senza utilizzare i computer. Ci sono dei lavori, non lavori per cui ho uno speciale talento, ma lavori che potevo fare certamente. Ma questo sarebbe stato uno spreco dei miei talenti. Ed anche così non avrei potuto più divertirmi.

Ho cercato dunque una terza via. Ero convinto che per creare una comunità di condivisione del software occorresse un sistema operativo libero, perché il primo insieme di software che occorre per utilizzare un computer è un sistema operativo [...]. Il sistema operativo fa la differenza tra poter utilizzare e non poter utilizzare un computer. La mia specialità erano i sistemi operativi, dunque se qualcuno al mondo poteva scrivere un tale sistema, quello ero io.

♣ NASCE UN'IDEA.

Dunque perché non fare un tentativo per cambiare la situazione e scrivere un sistema operativo libero? Sapevo che se ci fossi riuscito avrei potuto creare di nuovo una comunità, e anche se avessi fallito almeno avrei compiuto uno sforzo che meritava di essere fatto. Così decisi di scrivere un sistema operativo libero che rendesse possibile la creazione di una comunità nella quale la cooperazione fosse permessa. Il passo successivo fu la concezione tecnica del sistema. Scelsi di sviluppare un sistema compatibile con *Unix* e di seguire l'organizzazione di *Unix* per varie ragioni. Avevo appena visto un intero sistema operativo diventare inutile, inutilizzabile in quanto scritto per l'architettura di una macchina ben precisa, quindi sapevo che il nuovo sistema doveva essere portabile. A quell'epoca (1983) non c'era che un sistema portabile utilizzato veramente su architetture diverse, ed era *Unix* [...].

♣ STORIA DI NOMI.

Occorreva un nome e, siccome sono un *hacker*, avevo bisogno di un nome divertente ed un po' cattivo. C'era una tradizione degli *hacker* per cui quando si scrive un programma che è compatibile con un altro si può dare al nuovo programma un nome che è un acronimo ricorsivo che significa "questo programma non è l'altro, somiglia all'altro ma non è l'altro." Così, negli anni 60 e 70, c'erano molti editor *TECO* e, per la maggior parte, erano chiamati "qualche cosa-*TECO*", ma un *hacker* molto dotato ebbe l'idea di chiamare il suo programma *TINT*, *Tint Is Not TECO*. Quando in seguito scrissi il primo editor *Emacs*, c'erano molti editor *Emacs* tutti compatibili, la maggior parte chiamati "qualche cosa-*Emacs*", ce n'era uno chiamato *FINE*, *Fine Is Not Emacs*, un altro *SINE*, *Sine Is Not Emacs*, e poi *EINE*, *Eine Is Not Emacs*. Più tardi *EINE* fu riscritto quasi interamente, e la versione nuova era chiamata *ZWEI*, *ZWEI Was Eine Initially*.

Dunque cercavo un nome di questo tipo, ma mi resi conto velocemente che non c'era



un acronimo di quattro lettere che componesse una parola. Cercai ancora e trovai l'acronimo di tre lettere *GNU*, *GNU's Not Unix*. In inglese la parola *gnu* è molto divertente, perché il 'g' muto in inizio di parola è bizzarro, anche per l'inglese. Ci sono delle canzoni, delle battute su questa parola: era evidentemente la parola che dovevo scegliere.

♣ GNU.

Avevo un nome e un progetto, ma per iniziare avevo bisogno di denaro per pagare chi l'avrebbe scritto. Feci una descrizione del progetto e la mostrai ai dirigenti delle fabbriche di computer; risposero talvolta con iniziale interesse, ma senza dare seguito. Dopo alcuni mesi mi dissi: "Inutile aspettare, devo cominciare senza denaro."

Così nel gennaio 1984 mi licenziai dal *MIT* e cominciai a scrivere il sistema *GNU*. Fortunatamente il capo del laboratorio mi permise di continuare ad utilizzarne le macchine. Dovetti licenziarmi perché talvolta le università prendono i software scritti dai loro impiegati per venderli come prodotti proprietari [...].



GNU. Logo.

Dunque, cominciai a scrivere; il primo programma che scrissi e che era utile in sé era l'*Emacs* di *GNU*. Lo cominciai nel settembre 1984 e agli inizi del 1985 avevo già una prima versione rudimentale ma funzionante. C'erano già delle richieste ed avevo il problema della diffusione del programma. [...] [A quel punto avrei potuto mettere il programma in rete, ma allora pochissimi vi po-

tevano accedere, io] ero disoccupato ed avevo bisogno di trovare altri mezzi di guadagnare del denaro, dunque decisi di vendere i dischetti. Scrissi un annuncio sulla rete, "Vendo i dischetti per 150 dollari."

Sulle prime gli ordini cominciarono ad arrivare poco a poco, poi più frequentemente e, in luglio, ricevetti forse 10 ordini, e potevo vivere quasi solamente con questi ordini. Poi, a questo punto, mi resi conto che era evidente a tutti che il progetto *GNU* era in grado di scrivere del software, che non erano solo chiacchiere. Perciò valeva la pena di cercare delle donazioni. Creammo dunque la *Free Software Foundation*, nell'ottobre 1985, per ricevere donazioni detassate ed anche per vendere i dischetti al posto mio.



GNU GPLv3. Logo.

♣ QUALCOSA CHE IN TEORIA NON SAREBBE POTUTO ACCADERE.

La maggior parte del denaro veniva dalla vendita, il che era sorprendente. Infatti la vendita di cose di cui tutti possono fare delle copie ad alcuni sembrava, e sembra tuttora, impossibile. Mi dicevano: "Se il software è libero, non puoi venderne che un esemplare, perché quello che ha acquistato il primo esemplare andrà a venderne delle copie a un prezzo più basso, così tutti acquisteranno da lui piuttosto che da te. Oppure dovrai chiedere un prezzo equivalente al costo di sviluppo per la prima copia, cosa che è evidentemente impossibile. Dunque, l'idea è stupida." Eppure, da allora, la Fondazione ha venduto molte copie. Ma perché questo processo non funziona secondo la teoria economica "Si può solo speculare"? Evidentemente, la teo-

ria non è completamente corretta. Mi piace contraddire le leggi dell'economia. Infatti è vero che c'è chi vende copie dello stesso software a un prezzo meno elevato, perché cerchiamo sempre di avere il prezzo più alto, tuttavia ci sono tanti che continuano a comprare da noi.

Io propongo che acquistiate da noi per sostenere la scrittura del software libero o che almeno, quando comprate CDROM di software liberi, cerciate di verificare che il venditore dia delle sovvenzioni alla scrittura di questo tipo di software. Se non lo fa, potete scegliere un altro venditore per sostenere la scrittura di software liberi. [...]

Durante gli anni abbiamo scritto parecchi programmi, perché in un sistema operativo di tipo *Unix* occorrono molte componenti e svariati programmi per ottenere un intero sistema. Agli inizi potevamo scrivere qualunque componente perché avevamo bisogno di quasi tutto, più tardi abbiamo dovuto redigere un elenco di ciò che ancora mancava, la *tasklist* di *GNU*. A questo punto abbiamo cercato delle persone interessate a scrivere questi programmi [...]. Poco a poco, ci avvicinavamo ad un sistema completo⁵.

♣ LINUX.

Combinando [il kernel] *Linux* col sistema *GNU*, siamo arrivati ad un sistema completo

⁵ [Dettagli tecnici spostati in nota.] “Verso il 1990 tutto il sistema esisteva quasi, mancava solamente il *kernel*. Non bisognava cominciare da un componente particolare, perché quando si ha bisogno di tutti i componenti, perché cominciare da quello? Bisognava scrivere tutto, dunque l'ordine non aveva importanza. Abbiamo cominciato a lavorare su un *kernel*, ma ho fatto un errore di concezione. Ho supposto che utilizzando un *microkernel*, evitavamo il compito più difficile: il *debugging* a basso livello del software di sistema, in assenza di strumenti appositi. Ma la concezione tecnica che avevo scelto era in effetti molto difficile da completare, con parecchi processi asincroni in comunicazione, e dunque problemi di sincronizzazione, pertanto lo sforzo è durato parecchi anni. Contemporaneamente, Linus Torvalds ha scritto un *kernel* monolitico che si chiama *Linux*.”

funzionante totalmente libero, leggermente diverso da *GNU*, ma che mantiene in toto lo scopo e lo spirito del progetto. [...]

Al momento gli utenti di *GNU/Linux* sono circa 10 milioni⁶, e questo ci permette di ritenere il progetto *GNU* un successo, soprattutto perché queste persone sono libere: possiedono infatti i tre livelli di libertà.

♣ IL PRIMO LIVELLO DI LIBERTÀ.

Libertà di fare dei cambiamenti, dei miglioramenti atti a soddisfare i propri bisogni. Questo è esattamente quel che non potevamo fare al *MIT* con la stampante di *Xerox*.

Ho già parlato dei problemi pratici che conseguono dall'essere privi di questa libertà, ma vi sono anche dei problemi psicosociali [...]. Quando si ha davvero intenzione di svolgere adeguatamente un certo compito, perché lo si ritiene importante, ma qualcuno dice: “Non puoi far niente perché Tizio ha rinchiuso gli utensili”, la mentalità di partenza cambia radicalmente.

Si arriva a dirsi: “Bene, tanto sono pagato comunque, e ho un libro da leggere. Se un giorno mi si permetterà di svolgere il mio compito, vorrà dire che ricomincerò, ma per oggi sono pagato per leggere.” Non amo questa mentalità.

Trovo che vi sia un problema sociale se la situazione è tale da costringere gli individui ad assumere questo atteggiamento. Questo è esattamente quel che accade se [...] vi si impedisce di effettuare dei cambiamenti neces-



Linux. Kernel Sistema Operativo. Licenza: GNU General Public License (GPL)

sari nei software che utilizzate. Così si viola il primo livello di libertà.

♣ IL SECONDO LIVELLO DI LIBERTÀ.

Libertà di aiutare gli altri a diffondere delle copie. Si tratta della libertà di cooperare con i propri amici. Perché l'atto fondamentale di cooperazione tra chi usa i computer è dare le copie del software che si utilizza. Se possedete un qualcosa che può aiutare la comunità, e incontrate chi ne ha bisogno, gliela date. Che cosa c'è di più naturale? Ma in campo informatico vi chiamano "pirati" se trattate i vostri amici da amici. Trovo che la mancanza di questa libertà ponga dei problemi pratici [e psicologici]. [...].

♣ SPRECO DI LAVORO.

Per arrivare ad un livello specifico di qualità bisogna realizzare lo stesso lavoro, che gli utenti siano mille o un milione. Esattamente lo stesso lavoro. Se un sistema proprietario impone uno scoraggiamento finanziario, delle restrizioni finanziarie, per l'utilizzazione di un certo programma, ci saranno meno

utenti, anche se il lavoro richiesto ai programmatori avrebbe potuto soddisfare anche chi non ha voluto pagare. Il lavoro è andato parzialmente sprecato.



GIMP. Trattamento immagini. Licenza: GNU General Public License (GPL)

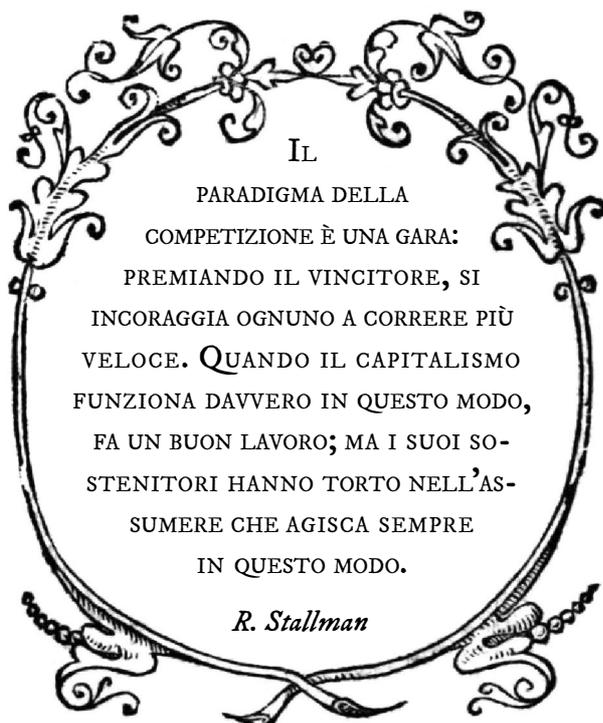
♣ BUONA VOLONTÀ DISINCENTIVATA.

Questo per quanto concerne il problema pratico; invece il problema psicologico sopraggiunge quando le persone vengono insultate e la cooperazione qualificata come pirateria. Perché ogni comunità è basata sulla buona volontà, la cooperazione con gli altri ma, dato che la natura umana è complessa, occorre sempre fare degli sforzi per incoraggiare questo spirito nella società.

Infatti molti sentimenti coabitano nella natura umana: ci sono buone e cattive volontà. Per avere una buona società, bisogna prendere delle misure per incoraggiare la buona volontà. Quando andavo a scuola, ed ero molto piccolo, la maestra diceva sempre: "Bisogna condividere le proprie caramelle!" [risa dal pubblico]. Adesso il governo degli Stati Uniti propone che le maestre dicano: "Oh no! Non condividete! Condividere è pirateria! Non si deve condividere mai con gli altri bambini!" [...]

♣ DUE TEORIE ECONOMICHE (FALLITE).

Ci sono certamente *altre* teorie. Un tempo era in voga la teoria secondo cui si poteva fare a meno della buona volontà grazie alla "pianificazione centrale": basta che ci sia chi ordina ad ognuno di fare il proprio lavoro e



tutto andrà bene. Ma ad oggi possiamo vedere come, nella maggior parte dei casi, questa teoria sia da considerarsi fallita.

C'è una teoria concorrente secondo cui si può fare a meno della buona volontà perché il dio, la mano invisibile, può operare in modo che anche i cittadini totalmente avidi debbano cooperare in modo efficace, pur senza metterci la buona volontà. Ma basta aprire gli occhi per rendersi conto che questo genere di società non solo non funziona, ma è anche totalmente detestabile.



Firefox. Browser. Licenze: Mozilla Public License (MPL), GNU General Public License (GPL) e GNU Lesser General Public License (LGPL)

♣ DANNI SOCIALI.

Dunque, spero che siate d'accordo che ci vuole della buona volontà. E quando sono il governo ed i potenti, le forze potenti della società, a dire che la cooperazione è una forma di pirateria, cosa ne deriva?

C'è chi rifiuta la condivisione delle copie dicendo: "Ah, è contro la legge! Si deve sempre ubbidire alle leggi!" [boccaccia di disgusto di Stallmann e risa del pubblico]. Ecco un danno psicosociale.

Ci sono altri che dicono: "Vorrei darvi una copia, ma ho paura." Una società basata sulla paura persuasiva, controllata dalla paura, è ciò che vorrebbero i proprietari di software. Perché occorre che la paura sia persuasiva per impedire alla gente di dare delle copie, e questo è un altro danno psicosociale.

Altri ancora invece dicono: [aria complice] "Ah sì... ora ti faccio una copia pirata... hi hi hi!". [risa] Essi cooperano, ma credono che la cooperazione sia una loro debolezza, credono di dover rifiutare ma di non essere abbastanza forti per farlo. È triste; fanno la cosa giusta ma ne traggono comunque un danno psicosociale: convincersi che la cooperazione sia una cosa sbagliata.

♣ IL TERZO LIVELLO DI LIBERTÀ.

Libertà di aiutare a edificare la società e la comunità aggiungendo funzioni ai software e pubblicandone versioni migliorate, affinché tutti possano utilizzarli. Questo processo è molto importante per il software libero, perché è attraverso di esso che il software si sviluppa più rapidamente.

♣ LA VERITÀ È EMPIRICA.

C'è chi ancora oggi mi dice: "Ma se il software fosse davvero libero verrebbe di sicuro a mancare la manodopera: nessuno lo scriverebbe!"

Tuttavia la verità è empirica, e i fatti empirici dimostrano che sono in molti a lavorare sul software libero.

In effetti, una volta che un programma libero viene accettato, molta gente comincia a contribuire agli ulteriori miglioramenti: spesso accade addirittura che l'afflusso di contributi sia così grande da impedire all'autore di utilizzarlo interamente... è un problema che *Microsoft* non ha mai. [risa] [...]



Seven Zip. Utilità di compressione. Licenze: GNU General Public License (GPL) e GNU Lesser General Public License (LGPL).

♣ SCIENZIATI GIAPPONESI.

Riguardo alla cooperazione scientifica [...] ho letto che una volta, durante la seconda guerra mondiale, alcuni soldati americani sbarcati su un'isola dell'Oceano Pacifico rinvennero un edificio che recava la seguente scritta:

“Ai soldati americani. Questo edificio è un laboratorio di biologia marina. Abbiamo messo in ordine tutti i nostri esemplari e rapporti affinché voi possiate proseguire i nostri lavori.”

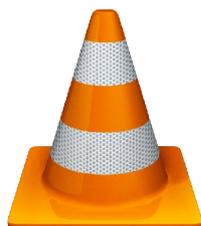
I biologi che avevano lasciato questo messaggio sentivano di lavorare per l'Uomo: per la scienza e per l'umanità intera, non solamente per il Giappone. Volevano che il proprio lavoro fosse utile all'umanità, qualunque fosse il risultato della guerra.

Oggi, in tutto il mondo, viviamo in uno stato di guerra civile tra piccoli gruppi, e ognuno agisce per bloccare gli altri, ostacolare gli altri, danneggiare gli altri. Tutto questo è molto triste.

♣ DIFFICOLTÀ TECNICO-LEGALI.

Questi dunque sono i tre livelli di libertà che costituiscono la definizione di software libero. Un programma è libero per voi, l'utente, se possedete queste tre libertà.

La definizione dev'essere formulata nei termini di un utente specifico perché un programma potrebbe essere libero per alcuni e



VLC. Media player. Licenza: GNU General Public License (GPL).

non per altri. Posso spiegarlo con un esempio, quello del sistema [...] *XWindow*.

♣ SI PUÒ BREVETTARE QUALCOSA DI COMUNITARIO ED IMPOSSESSARSENE.

Il sistema X era stato scritto al MIT e diffuso dal MIT come software libero: se avevate ricevuto una copia dal MIT possedevate tutte le libertà di cui sopra. Ma c'erano anche delle società produttrici di computer che detenevano una propria versione del sistema *Unix*, e che avevano l'abitudine di non concedere alcun livello di libertà ai propri utilizzatori.

Queste società, insomma, presero X, lo adattarono al proprio sistema e diffusero delle copie come prodotto proprietario [...].

Se eravate un utente, se sapevate che esisteva il problema delle libertà ed avevate voglia di avere queste libertà, potevate cercare la versione del MIT ed installarlo sul computer, e se questa versione funzionava potevate avere il sistema X godendo dei tre livelli di libertà.

Ma poteva anche darsi che questa versione non funzionasse sul vostro sistema. In tal caso, il sistema X non faceva per voi: non vi restava che il software proprietario.

Agli autori del sistema X questo meccanismo andava bene, perché il loro scopo era di avere successo. Ma il mio scopo era di creare una comunità che godeva dei tre livelli di libertà e dunque, per me, questo risultato sarebbe stato un insuccesso, un insuccesso totale! Avrei considerato il mio lavoro sprecato se avessi ottenuto questo risultato. Così ho cercato un mezzo di vietarlo, ed il mezzo che ho trovato è il «*copyleft*».

♣ COPYLEFT.

L'idea è che, diversamente da quel che accade con il sistema X, tutti possono fare delle copie, tutti possono fare dei cambia-

menti, tutti possono diffondere delle copie anche con i cambiamenti, ma ogni copia diffusa deve mantenere tutte le libertà dell'originale⁷, le stesse libertà che vi garantisco io. [...] Così non c'è copia senza libertà, da nessuna parte.



OpenOffice.org. Word Processor, foglio elettronico, presentazioni, ecc.
Licenza: GNU Lesser General Public License (LGPL).

♣ UNA COMUNITÀ CHE DEVE CRESCERE.

Per un software libero occorre una documentazione libera e dei manuali liberi: è molto importante, anche se attualmente nella comunità i più non lo comprendono. Voglio insistere su questo punto, perché la documentazione è parte essenziale del software.

Ogni programma deve arrivare con la sua documentazione completa. Sul computer, ma anche, se lo volete, su carta. Certo, normalmente si deve pagare per la copia cartacea perché costa stamparla, ma, d'altro canto, si chiama "software libero", non "gratuito"! [...]

Comunque il punto su cui voglio insistere è che il software deve essere fornito della do-

⁷ È questa l'invenzione di Stallman: una particolare licenza, chiamata GNU General Public License, che permette agli autori di un *software* di renderlo di dominio pubblico, ma non lasciandolo senza *copyright* in modo che qualcuno se ne possa impossessare brevettandolo! Inoltre la licenza consente di incorporare tale software in altri programmi solo se anch'essi sono sotto la stessa licenza, generando così una specie di catena di S. Antonio. Nel tempo la licenza si è evoluta e ne sono comparse molte varianti. Il termine *copyleft* è un gioco di parole su *copyright* nel quale "right" (destra) è scambiata con "left" (sinistra).

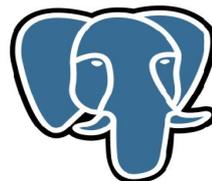
cumentazione completa. Questa documentazione dev'essere anch'essa libera perché, quando fate delle copie di programmi e le diffondete, insieme dovete diffondere il manuale, e quando fate dei cambiamenti nel programma [...] dovete cambiare anche il manuale, affinché corrisponda alla nuova versione [...]. Se il manuale non dispone della libertà di fare dei cambiamenti, insomma, non è utile alla nostra comunità. [...]

Quindi vi propongo questo: se in una libreria trovate in vendita un manuale libero, acquistatelo; se invece trovate un manuale proprietario, non lo fate. Perché è bene dare denaro agli editori che pubblicano manuali liberi: possiamo augurar loro il successo ed il profitto. Ma agli editori che pubblicano i manuali proprietari, bisogna augurare l'insuccesso. [risa]

La fondazione del software libero vende manuali liberi, ed io spero che gli editori commerciali facciano la stessa cosa. Proprio come spero che, se fate l'informatico, scriviate del software libero. Perché se avete goduto dei benefici della nostra comunità, spero che sentiate un debito morale, il dovere di contribuire alla comunità stessa.

Se non siete informatici, se non sapete scrivere del software ma sapete scrivere dei testi, potreste forse occuparvi dei manuali [...]; se siete capaci di gestire la burocrazia, abbiamo bisogno anche di questo. [...]

PostgreSQL



PostgreSQL. Database relazionale.
Licenza; PostgreSQL License, simile a quella BSD o MIT.

Ma, in ogni caso, dovete capire che il sistema *GNU/Linux* appartiene alla comunità: è il *vostro* sistema e spero che anche voi cercherete di contribuire a costruire questa comunità.

♣ UNA SPECIE DI GUERRA.

Recentemente, in una conferenza a cui ho partecipato, qualcuno mi ha chiesto: “Che cosa pensa *Microsoft* del software libero?”. Ho risposto: “Che io sappia, niente! Semplicemente non se ne cura”. [risa]

Sembra che non sia più così. Ci sono dei documenti dai quali risulta che *Microsoft* sta organizzando per combattere il software libero. In essi vengono utilizzate due diverse strategie:

- utilizzare protocolli e formati di dati segreti
- utilizzare i brevetti contro noi

Suppongo che in dei casi queste strategie funzioneranno, perché non sempre possiamo difenderci: è troppo difficile. Ma che possiamo fare? ⁸ [...]

Per quanto riguarda i brevetti, funziona come con le mine: si possono prendere delle precauzioni, ma talvolta una mina esplode. È certo che delle mine esploderanno su noi e ci saranno delle cose che non potremo scrivere, delle funzionalità che non potremo offrire agli utenti; forse per un periodo di quattro anni, o dieci, o anche venti. Il periodo varia in funzione dei casi particolari.

Che cosa possiamo fare? Se valorizziamo la libertà, possiamo scegliere un software li-

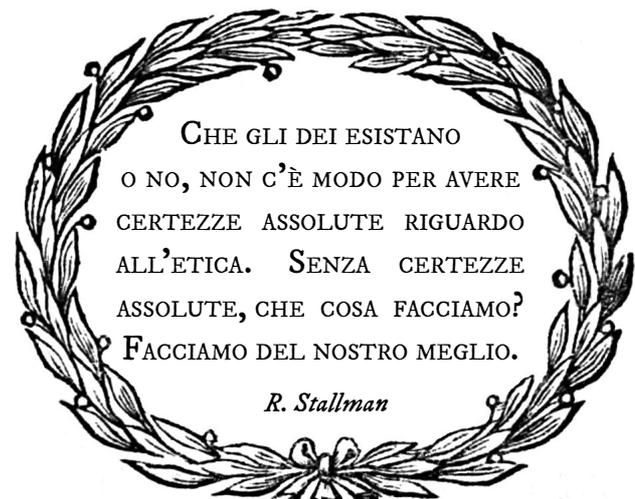
bero che non offre una certa funzionalità, piuttosto che un programma proprietario che l'offre, perché la libertà è più importante di una funzionalità. [...]

♣ GUARDANDO AL FUTURO.

Dobbiamo dunque diffondere soprattutto il valore della libertà di cooperare, che è la più importante perché è la libertà di fare qualcosa di buono per gli altri. Più importante addirittura della libertà di fare qualcosa per sé stessi.

Solo con questa forza morale potremo cambiare le leggi sui brevetti. Un giorno, quando avremo qualcosa come cento milioni di utenti, potremo forse cambiare le leggi sui brevetti e dimenticare per sempre questo problema. Se invece le persone preferiranno scegliere programmi proprietari che offrono funzionalità brevettate, allora perderemo.

RICHARD STALLMAN



⁸ [Dettagli tecnici spostati in nota.] “Contro protocolli e formati segreti occorre l'ingegneria a rovescio, il *reverse engineering*. Ci sono due mezzi di farlo. Si può fare per tentativi, oppure attraverso la decompilazione. Negli Stati Uniti la decompilazione può essere sia illegale che legale, a seconda della corte che legifera [...], ma in Europa c'è una legge esplicita che ne garantisce la legalità. Di conseguenza, è molto importante far sì che questo lavoro venga svolto in Europa.”