

RICHARD STALLMAN

SOFTWARE LIBERO
PENSIERO LIBERO

VOLUME SECONDO

Libertà di software e di pensiero, grazie!

Prosegue la presentazione italiana dei saggi e degli interventi più significativi di Richard Matthew Stallman, fondatore del movimento del software libero. Come annunciato nel primo volume (maggio 2003), questo secondo tomo include i testi integrali delle varie licenze GNU, a partire dalla più affermata, la GPL, General Public License, nonché le trascrizioni di alcuni importanti interventi dal vivo di Stallman (quali “Copyright e globalizzazione nell’epoca delle reti informatiche” e “Software libero: libertà e cooperazione”). Viene così completata la versione nostrana di un libro originale – Free Software, Free Society: Selected Essays of Richard M. Stallman – dove vengono condensati oltre 20 anni di testi e interventi pubblici che hanno modificato la nostra stessa concezione dell’informatica e della tecnologia. Testi che, a scanso di equivoci, non si limitano a fare la storia del movimento del software libero, ma gettano anzi forti luci sul futuro di dinamiche al crocevia tra etica e legge, business e software, libertà individuale e società trasparente.

Questo secondo volume è inoltre arricchito da un’importante appendice: un documento in cui l’Associazione Software Libero aggiorna lo scenario su alcuni temi di scottante attualità anche per la scena italiana: l’EUCD (European Union Copyright Directive), i brevetti sul software, la pubblica amministrazione e il software libero. Si tratta in pratica di tre cavalli di battaglia – avviati dall’Associazione ma ovviamente aperti a chiunque voglia e vorrà coinvolgersi – attraverso i quali “scardinare, o almeno tentare, la visione imperante del-

l'informatica legata al pagamento delle licenze, alla mera esecuzione dei programmi, alla limitazione delle informazioni per trasformare gli utenti in 'pigiatori' di tasti e icone a cui manca però la conoscenza di fondo, quella che si cela dietro alle interfacce grafiche e che costituisce uno dei presupposti della libertà.”

In ambito più generale, va invece segnalata l'iniziativa legale avviata lo scorso marzo dalla ex-Caldera, ora nota come Santa Cruz Operation (SCO), contro IBM e altre aziende che fanno uso del sistema GNU/Linux. Iniziativa che sembra porsi come una sfida lanciata contro l'intero mondo dell'open source e del software libero, ovvero mirata agli stessi utenti insieme a qualche big dell'industria informatica. Al riguardo è senz'altro il caso di riportare alcuni stralci della posizione assunta a fine giugno dalla Free Software Foundation (FSF), in cui Eben Moglen, esperto legale e Consigliere Generale della stessa FSF, interviene su alcuni dettagli importanti:

“SCO accusa IBM di aver infranto i vincoli contrattuali tra le due aziende e di aver incorporato in ciò che SCO chiama genericamente “Linux” segreti industriali che riguardano la progettazione del sistema operativo UNIX. Quest'ultima affermazione è stata recentemente ampliata in dichiarazioni extragiudiziarie da parte del personale di SCO e di alcuni suoi dirigenti che hanno specificato che “Linux” incorpora materiale copiato da UNIX, violando il Copyright di SCO. Un'affermazione di questo tenore è contenuta anche in una lettera che SCO pare abbia inviato a 1500 tra le più grandi aziende al mondo mettendole in guardia dall'utilizzo del Software Libero sulla base di possibili responsabilità concernenti violazione di Copyright.”

Il testo di Eben Moglen (reperibile interamente in versione italiana: <http://www.it.gnu.org/philosophy/sco-statement.it.html>) prosegue sottolineando la confusione, apparentemente voluta, della posizione

di SCO rispetto all'utilizzo del termine "Linux" per intendere "tutto il Software Libero" o "tutto il Software Libero che costituisce un sistema operativo simile a UNIX". Aggiungendo come pur a fronte delle contestazioni sul segreto industriale, le uniche mosse nella causa contro IBM, va comunque notato il "semplice fatto che SCO ha per anni distribuito copie del kernel, Linux, come parte di sistemi GNU/Linux. Tali sistemi sono stati distribuiti da SCO nel pieno rispetto della GPL, e per ciò, includevano l'intero codice sorgente." Con una conclusione che offre una precisa presa di posizione della FSF e dei sostenitori del software libero:

"Di fronte a questi fatti, le dichiarazioni pubbliche di SCO sono quantomeno fuorvianti ed irresponsabili. SCO ha abilmente approfittato del lavoro di coloro che hanno fornito contributi da tutto il mondo. Le loro attuali dichiarazioni pubbliche costituiscono un volgare abuso dei principi della comunità del Software Libero, da parte di un membro che ha utilizzato tutto il nostro lavoro per il proprio tornaconto economico. La Free Software Foundation invita SCO a ritirare le proprie sconsiderate ed irresponsabili dichiarazioni e di provvedere a separare immediatamente i propri disaccordi commerciali con IBM dai propri doveri e le proprie responsabilità nei confronti della comunità del Software Libero."

Da parte sua, Richard Stallman ha commentato qua e là la faccenda, ma senza eccessive preoccupazioni. Chiarendo, tra l'altro, punti chiave come il seguente: "In una comunità di più di mezzo milione di sviluppatori, non possiamo aspettarci che non avvengano mai casi di plagio. Ma non è un disastro; possiamo scartare tale materiale e proseguire. Se c'è materiale in Linux che è stato aggiunto senza il diritto legale di farlo, gli sviluppatori di Linux lo individueranno e lo sostituiranno. SCO non può usare i propri copyright, o i propri contratti

con altre parti, per sopprimere i legittimi contributi di migliaia di altri soggetti. Lo stesso Linux non è più essenziale: il sistema GNU è diventato popolare in congiunzione con Linux, ma oggi gira su due kernel BSD e con il kernel GNU. La nostra comunità non può essere sconfitta da questa vicenda.”

Va aggiunto che, in piena estate 2003, le posizioni di media e industria, aziende e organizzazioni, addetti ai lavori e avvocati, a partire dalla scena statunitense per ampliarsi al resto del mondo, concordano su un fatto: l'iniziativa a tutto campo di SCO appare assurda, immotivata e condannata alla sconfitta. L'impressione generale è che l'ex-Caldera miri a risultati finanziariamente vantaggiosi, onde recuperare i diversi milioni di dollari persi sul mercato in anni recenti. Ciò include varie possibilità, dall'acquisto da parte della stessa IBM denunciata o altro gigante high-tech all'imposizione di licenze ad aziende Linux e/o ai singoli utenti. Non a caso l'ultima mossa di SCO è la registrazione di una nuova licenza di UnixWare avente come target gli utenti commerciali di Linux, e in cui si impone ai distributori di usarne il kernel soltanto in versione binaria, bloccando in pratica l'accesso al codice sorgente, dal kernel 2.4 in poi. Pur senza quantificare, al momento SCO propone insomma ad aziende e utenti di pagare la licenza per il suo UnixWare 7.1.3, su cui girano sia applicazioni Linux che Unix. Comunque sia, finora tali iniziative non hanno provocato alcuna ripercussione negativa a livello di mercato, mentre le testate specializzate USA confermano che gli utenti non paiono per nulla scossi dalle minacce di SCO, vere o presunte che siano. A ridosso di ferragosto, è infine arrivata l'attesa contro-denuncia di IBM, la quale chiede in sostanza ai giudici dello Utah di archiviare la pratica vista la falsità delle accuse, imponendo anzi a SCO un rimborso danni "compensatori e punitivi", pur senza specificarne l'ammontare.

In attesa di ulteriori sviluppi giudiziari o, forse meglio, del previsto patteggiamento tra le parti in causa, la vicenda ribadisce innanzitutto la forza raggiunta dal movimento free software (e open source) a livello commerciale, tale da replicare con fermezza anche a imprevise sfide legali ed eventuali ricadute a largo raggio. Un ambito complessivo in cui, oltre a difendersi in aula se e quando sarà il caso, restano più che valide le concezioni efficacemente illustrate da Richard Stallman in queste stesse pagine. Per rafforzarsi ulteriormente, ribattere a simili accuse e affrontare adeguatamente ogni tipo di sfida futura, possiamo giurare su un fatto: il movimento continuerà ad evolversi in sintonia con le pratiche di massima apertura e condivisione a livello globale che lo contraddistinguono da sempre. Espressione tanto concreta quanto catalizzante di un esperimento teso, ieri come oggi e domani, all'affermazione della libertà di tutti e di ciascuno.

Bernardo Parrella

berny@cybermesa.com

agosto 2003

Parte Prima

**Libertà, società
e software**

Possiamo fidarci del nostro computer?

Da chi dovrebbe ricevere ordini il nostro computer? La maggior parte della gente ritiene che il computer dovrebbe obbedire all'utente, non a qualcun altro. Con un progetto denominato "informatica fidata" (trusting computing), le grandi corporation dei media, incluse l'industria cinematografica e quella musicale, insieme ad aziende informatiche quali Microsoft e Intel, stanno cercando di fare in modo che il computer obbedisca a loro anziché all'utente. I programmi proprietari presentavano già delle funzioni ambigue, ma tale progetto le renderebbe universali.

Sostanzialmente software proprietario significa che l'utente non può controllarne le funzionalità; né può studiarne il codice sorgente o modificarlo. Non deve sorprendere il fatto che qualche sagace imprenditore trovi il modo di usare il proprio potere per metterci in svantaggio. Microsoft lo ha fatto parecchie volte: una versione di Windows era progettata per segnalare a Microsoft tutto il software presente sull'hard disk dell'utente; un recente upgrade "di sicurezza" per Windows Media Player imponeva l'assenso dell'utente per nuove restrizioni. Ma Microsoft non è certo l'unica: il software di file sharing per la musica KaZaa è progettato in modo che i suoi partner commerciali possano affittare ai propri clienti l'uso del computer dell'utente. Spesso simili funzioni ambigue rimangono segrete, ma perfino quando se ne conosce l'esistenza, è difficile rimuoverle perché l'utente non ne possiede il codice sorgente.

Nel passato questi erano incidenti isolati. "L'informatica fidata" li

renderebbe dilaganti. Una definizione più appropriata sarebbe “informatica ingannevole” (treacherous computing), poiché il piano è progettato per assicurarsi che il computer disubbidisca sistematicamente all’utente. Anzi, è progettato per impedire al computer di operare come un computer per usi generici. Ogni operazione potrebbe richiedere un’autorizzazione esplicita.

L’idea tecnica alla base dell’informatica ingannevole è che il computer include un congegno per la cifratura e la firma digitale, le cui chiavi vengono tenute segrete all’utente. (La versione Microsoft di tale sistema si chiama “Palladium”). I programmi proprietari useranno questo congegno per controllare quali altri programmi l’utente possa far girare, a quali documenti o dati può accedere e in quali programmi possa trasferirli. Tali programmi preleveranno in continuazione nuove autorizzazioni via Internet, imponendole automaticamente all’utente. Se quest’ultimo non consente al proprio computer di ottenere periodicamente nuove regole da Internet, alcune capacità smetteranno automaticamente di funzionare.

Naturalmente, Hollywood e le aziende discografiche prevedono di ricorrere all’informatica ingannevole per il “DRM” (Digital Restrictions Management, gestione delle restrizioni digitali), in modo che i video e la musica scaricata possano essere visti e ascoltati soltanto su un determinato computer. Risulterà del tutto impossibile condividerli, almeno usando i file autorizzati ottenuti da tali aziende. Noi, il pubblico, dovremmo avere sia la libertà sia la capacità di condividere queste cose. (Prevedo che qualcuno troverà il modo di produrre delle versioni cifrate, di diffonderle online e condividerle, in modo che il DRM non potrà avere pieno successo, ma ciò non è una scusante per l’esistenza di tale sistema). Negare la possibilità di condividere è già qualcosa di negativo, ma

c'è di peggio. Si prevede di usare procedure analoghe per email e documenti – provocando la scomparsa dell'email entro due settimane, oppure consentendo la lettura dei documenti unicamente sui computer di una sola azienda.

Immaginiamo di ricevere una email dal nostro datore di lavoro che ci dica di fare qualcosa che consideriamo rischioso; un mese dopo, quando scoppia qualche grana, non potremo usare quell'email per dimostrare che non siamo stati noi a prendere la decisione. “Metterlo per iscritto” non ci tutela quando l'ordine è scritto con inchiostro (simpatico?) che scompare.

Immaginiamo di ricevere un'email in cui il nostro datore di lavoro voglia imporci una procedura illegale o moralmente equivoca, come la distruzione dei documenti aziendali relativi a un'audizione fiscale, o lasciar passare senza verifiche una pericolosa minaccia al nostro paese. Oggi è possibile far arrivare il messaggio a un giornalista e rendere pubblica quell'attività. Ma grazie all'informatica ingannevole, il giornalista potrebbe non leggere il documento, il suo computer rifiuterebbe di obbedirgli. L'informatica ingannevole diventa il paradiso della corruzione.

Gli elaboratori di testi come Microsoft Word potrebbero ricorrere all'informatica ingannevole quando salvano i documenti, per assicurarsi che non possano esser letti da nessun elaboratore di testi rivale. Oggi dobbiamo scoprire i segreti del formato Word tramite laboriosi esperimenti onde poter realizzare elaboratori di testi liberi capaci di leggere i documenti Word. Se quest'ultimo dovesse cifrare i documenti ogni volta che li salva, la comunità del software libero non avrebbe alcuna possibilità di sviluppare software in grado di leggerli – e anche se riuscissimo a farlo, simili programmi potrebbero essere dichiarati illegali sotto il Digital Millennium Copyright Act.

I programmi che usano l'informatica ingannevole scaricheranno in continuazione via Internet nuove regole per le autorizzazioni, onde imporle automaticamente al nostro lavoro. Qualora a Microsoft, o al governo statunitense, non dovesse piacere quanto andiamo scrivendo in un documento, potrebbero diffondere nuove istruzioni dicendo a tutti i computer di impedire a chiunque la lettura di tale documento. Una volta scaricate le nuove istruzioni, ogni computer dovrà obbedire. Il nostro documento potrebbe subire la cancellazione retroattiva, in pieno stile "1984". Lo stesso utente che lo ha redatto potrebbe trovarsi impossibilitato a leggerlo.

È il caso di riflettere sulle spiacevoli conseguenze dell'applicazione dell'informatica ingannevole, studiarne le dolorose possibilità, e decidere se sia il caso di accettarle o meno. Sarebbe stupido e inopportuno accettarle, ma il punto è che l'affare che si crede di fare non potrà rivelarsi tale. Una volta dipendenti da quel programma, non se ne potrà più fare a meno, e loro lo sanno bene; a quel punto, vi apporteranno delle modifiche. Alcune applicazioni faranno automaticamente un upgrade che comporta cambiamenti funzionali – e non è possibile scegliere di rifiutare tale upgrade.

Oggi si possono evitare le restrizioni del software proprietario facendone a meno. Usando GNU/Linux o un altro sistema operativo libero, ed evitando di installarvi sopra delle applicazioni proprietarie, allora è l'utente a controllare cosa fa il computer. Se un programma libero include una funzione dannosa, altri programmatori della comunità la toglieranno e se ne potrà usare la versione corretta. Sarà inoltre possibile far girare applicazioni e strumenti liberi su sistemi operativi non-liberi; ciò non offre piena libertà, ma molti utenti lo fanno.

L'informatica ingannevole pone a rischio l'esistenza stessa dei siste-

mi operativi liberi e delle applicazioni libere, perché potrebbe essere del tutto impossibile farle girare. Qualche versione dell'informatica ingannevole potrebbe richiedere che il sistema operativo sia specificamente autorizzato da un'azienda particolare. Potrebbe essere impossibile installare dei sistemi operativi liberi. Altre versioni dell'informatica ingannevole potrebbero richiedere che ciascun programma sia specificamente autorizzato da chi ha sviluppato il sistema operativo. Sarebbe impossibile per l'utente far girare dei programmi liberi su tale sistema. Se trovate il modo di farlo, e lo raccontate in giro, potrebbe essere un reato.

Negli Stati Uniti esistono già delle proposte legislative che vorrebbero imporre a tutti i computer di supportare l'informatica ingannevole, con il divieto di collegare a Internet i vecchi computer. Una di queste è il CBDTPA (da noi definito Consume But Don't Try Programming Act, Legge per consumare ma senza cercare di programmare). Ma pur se non potranno costringerci legalmente a passare all'informatica ingannevole, ci sarà un'enorme pressione perché venga accettata. Spesso oggi si usa il formato Word per comunicare, nonostante ciò provochi un gran numero di problemi (si veda, in inglese, <http://www.gnu.org/no-word-attachments.html>; in italiano: <http://www.gnu.org/philosophy/no-word-attachments.it.html>). Se soltanto una macchina basata sull'informatica ingannevole fosse in grado di leggere i documenti Word più recenti, molta gente finirà per adeguarvisi, qualora considerino la questione puramente in termini individuali (prendere o lasciare). Onde opporsi all'informatica ingannevole dobbiamo unire le forze ed affrontare la situazione come una scelta collettiva.

Per ulteriori dettagli sull'informatica ingannevole, si veda (in inglese) <http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html>.

Per bloccare l'informatica ingannevole occorre la mobilitazione di un vasto numero di cittadini. C'è bisogno del vostro aiuto! La Electronic Frontier Foundation (www.eff.org) e l'organizzazione Public Knowledge (www.publicknowledge.org) hanno avviato una campagna di opposizione all'informatica ingannevole, e lo stesso sta facendo il Digital Speech Project sponsorizzato dalla Free Software Foundation (www.digitalspeech.org). Visitate questi siti Web e cercate di sostenerne l'attività. Si può aiutare anche scrivendo agli uffici per i rapporti con il pubblico di Intel, IBM, HP/Compaq, o qualsiasi altro produttore da cui abbiate acquistato un computer, spiegando loro che non volete subire pressioni per l'adozione di sistemi informatici "fidati" (trusted) e che quindi non volete ne producano affatto. Ciò servirà a dare potere ai consumatori. Se contate di scrivere lettere simili, inviatene copia alle organizzazioni nominate sopra.

Post Scriptum:

Il progetto GNU distribuisce GNU Privacy Guard, programma per l'implementazione di firme digitali e cifratura a chiave pubblica, che può essere usato per inviare email sicure e private. È utile esplorare il modo in cui GPG differisce dall'informatica fidata, e vedere cosa rende vantaggioso uno e così pericolosa l'altra.

Quando si usa GPG per l'invio di un documento cifrato, e si ricorre a GPG per decodificarlo, il risultato è un documento non cifrato che è possibile leggere, inoltrare, copiare e perfino ri-cifrare onde essere inviato con sicurezza a qualcun altro. Un'applicazione di informatica ingannevole ci consentirebbe di leggere le parole sul monitor, ma non di produrre un documento non cifrato da utilizzare in altri modi. GPG, un pac-

chetto di software libero, mette le funzioni di sicurezza a disposizione degli utenti; sono questi ultimi a usare il programma. L'informatica ingannevole è progettata per imporre le restrizioni sugli utenti; è tale informatica a usare gli utenti.

Questa è la prima versione mai pubblicata di questo saggio e fa parte del libro *Free Software, Free Society: The Selected Essays of Richard M. Stallman*, GNU Press, 2002.

La copia letterale e la distribuzione di questo testo nella sua integrità sono permesse con qualsiasi mezzo, a condizione che sia mantenuta questa nota.

Perché il software dovrebbe essere libero

Inevitabilmente l'esistenza del software solleva la domanda su come dovrebbero essere prese le decisioni per il suo utilizzo. Supponiamo ad esempio che una persona in possesso di una copia di un programma incontra qualcun altro che ne vorrebbe anch'egli una copia. C'è la possibilità di copiare quel programma; a chi spetta la decisione se farlo o meno? Alle persone coinvolte? Oppure ad un altro soggetto, definito il "proprietario"?

Gli sviluppatori di software tipicamente affrontano simili questioni sulla base dell'assunto che il criterio per la risposta sia quello di massimizzare i profitti degli stessi sviluppatori. Il potere politico dell'imprenditoria ha portato all'adozione da parte del governo sia di questo criterio sia della risposta degli sviluppatori. Ovvero, che esiste il proprietario del programma, tipicamente una corporation associata con il suo sviluppo.

Vorrei affrontare la medesima domanda sulla base di un diverso criterio: la prosperità e la libertà del pubblico in generale.

Questa risposta non può essere decisa dall'attuale legislazione – dovrebbe essere la legge a conformarsi all'etica, non viceversa. Non spetta neppure alle pratiche correnti decidere su tale questione, pur potendo queste suggerire possibili risposte. L'unico modo di giudicare è considerare chi viene aiutato e chi danneggiato dal riconoscimento dei proprietari del software. In altri termini, si dovrebbe condurre un'analisi costi-benefici per conto della società nel suo

complesso, prendendo in considerazione la libertà individuale come pure la produzione di beni materiali.

Nel presente saggio illustrerò gli effetti dovuti all'esistenza dei proprietari, dimostrandone i risultati negativi. La mia conclusione è che i programmatori hanno il dovere di incoraggiare gli altri a condividere, ridistribuire, studiare e migliorare il software che scriviamo: in altri termini, a scrivere software libero ("free software").¹

In che modo i proprietari giustificano il proprio potere

Quanti traggono benefici dall'attuale sistema in cui i programmi sono considerati una proprietà, offrono due argomenti a sostegno delle proprie tesi a favore di tale proprietà: l'argomento emotivo e quello economico.

L'argomento emotivo funziona così: "Ci metto il sudore e l'anima in questo programma. Viene da me, è mio!"

Quest'argomento non richiede una seria refutazione. La sensazione di attaccamento è qualcosa che i programmatori possono coltivare quando meglio si adatta loro; non è inevitabile. Consideriamo, ad esempio, con quanta decisione gli stessi programmatori siano soliti concedere con una firma tutti i diritti ad una grande corporation in cambio di uno stipendio; l'attaccamento emotivo scompare misteriosamente. All'opposto, consideriamo invece i grandi artisti e artigiani del Medioevo, che non si curavano neppure di firmare le proprie opere. Per loro, il nome dell'artista non era importante. Quello che contava era portare a compimento quell'opera – e lo scopo cui era destinata. Questa la visione prevalente per centinaia di anni.

¹ Il termine "free" in "free software" si riferisce alla libertà, non al prezzo ('free' in inglese significa sia 'libero' sia 'gratuito'); il prezzo pagato per la copia di un programma libero può essere zero, basso oppure (raramente) piuttosto elevato.

L'argomento economico funziona così: "Voglio diventare ricco (generalmente descritto in maniera poco accurata come 'guadagnarsi da vivere'), e se non mi consentite di diventare ricco con la programmazione, allora smetterò di scrivere programmi. Tutti gli altri sviluppatori la pensano come me, per cui nessuno vorrà mai programmare. E allora rimarrete senza alcun programma!" In genere questa minaccia viene velata come un amichevole avviso da una saggia fonte.

Più avanti spiegherò perché tale minaccia non è altro che bluff. Prima vorrei affrontare un assunto implicito che acquista maggior visibilità in un'altra formulazione dell'argomento.

Questa formulazione parte mettendo a confronto l'utilità sociale di un programma proprietario con l'assenza di programmi, per poi concludere che lo sviluppo di software proprietario è, nel suo insieme, qualcosa di benefico e andrebbe incoraggiato. L'errore qui consiste nel contrapporre soltanto due scenari – software proprietario contro niente software – assumendo l'inesistenza di ulteriori possibilità.

Considerando un sistema di copyright sul software, lo sviluppo del software viene solitamente connesso all'esistenza di un proprietario che ne controlli l'utilizzo. Fintantoché esisterà questa connessione, siamo spesso costretti a confrontarci con la scelta tra software proprietario o niente software. Tuttavia, tale connessione non è qualcosa di inerente o inevitabile; è una conseguenza della specifica decisione socio-legale che mettiamo in discussione: la decisione di avere dei proprietari. Presentare la scelta come tra software proprietario o niente software significa travisare la questione.

L'argomento contro l'esistenza di proprietari

La domanda in ballo è: "Lo sviluppo del software dovrebbe esse-

re connesso con l'esistenza di proprietari cui spetti limitarne l'utilizzo?"

Onde poter sciogliere il quesito, dobbiamo giudicare gli effetti sulla società di ciascuna delle due attività seguenti in maniera indipendente tra loro: l'effetto dello sviluppo del software (prescindendo dai relativi termini di distribuzione) e l'effetto derivante dalla limitazione del suo utilizzo (assumendo che il software sia stato sviluppato). Se una di queste attività dovesse risultare giovevole e l'altra dannosa, sarebbe bene lasciar andare tale connessione e optare soltanto per quella giovevole.

In altri termini, se limitare la distribuzione di un programma già sviluppato risulta dannoso per la società in generale, allora lo sviluppatore etico rifiuterà una simile opzione.

Onde stabilire l'effetto di limitazioni nella condivisione, occorre paragonare il valore sociale di un programma ristretto (ovvero, proprietario) con quello dello stesso programma disponibile a chiunque. Ciò significa mettere a confronto due mondi possibili. L'analisi affronta anche il semplice argomento talvolta contrapposto a questo, secondo cui "il beneficio al vicino nel fornirgli o fornirle la copia di un programma viene cancellato dal danno procurato al proprietario". Questo contro-argomento assume che il danno e il beneficio siano di proporzioni identiche. L'analisi include un raffronto fra tali proporzioni, per concludere che il beneficio risulta di gran lunga maggiore.

Per meglio illustrare l'argomento, applichiamo ad un altro ambito: la costruzione di una rete stradale.

Tramite i pedaggi sarebbe possibile finanziare la costruzione di tutte le strade. Ciò comporterebbe la presenza di appositi caselli a tutti gli angoli. Un simile sistema risulterebbe di grande incentivo al miglioramento della rete stradale. Offrirebbe inoltre il vantaggio di costrin-

gere quanti usano quella specifica strada al pagamento del relativo pedaggio. Eppure un casello per il pedaggio rappresenterebbe un'ostruzione artificiale alla fluidità di guida – artificiale perché non è una conseguenza di come funzionano le strade o le autovetture.

Paragonando le strade libere e quelle a pedaggio sulla base della loro utilità, scopriamo che (tutto il resto essendo identico) le strade senza caselli richiedono meno denaro per essere costruite e gestite, sono più sicure e più efficienti nell'utilizzo.²

In un paese povero molti cittadini sarebbero impossibilitati a usare le strade a pedaggio. Perciò le strade senza caselli offrono alla società maggiori benefici a costi minori; sono da preferire per la società. Di conseguenza, la società dovrebbe scegliere di reperire i finanziamenti in un altro modo, non tramite i caselli a pagamento. L'uso delle strade, una volta costruite, dovrebbe essere libero. Quando i sostenitori dei caselli li propongono unicamente come un modo per raccogliere denaro, distorcono la scelta a disposizione. È vero che i caselli a pedaggio portano soldi, ma provocano anche altre conseguenze: in realtà degradano la strada. Una strada a pedaggio non è così positiva come una libera; pur offrendo strade in numero maggiore o tecnicamente superiori, ciò non sarebbe un miglioramento qualora dovesse comportare la sostituzione delle strade libere con quelle a pagamento.

Ovviamente la costruzione di una strada libera necessita di fondi, che in qualche modo il pubblico è chiamato a sborsare. Tuttavia

² *Le questioni relative a inquinamento e congestioni di traffico non alterano questa conclusione. Nel caso si volesse rendere più esosa la guida onde scoraggiarla in generale, è svantaggioso farlo tramite dei caselli a pedaggio, i quali contribuiscono sia all'inquinamento che al traffico. Molto meglio ricorrere a una tassa sulla benzina. Allo stesso modo, la volontà di ottenere maggior sicurezza limitando la massima velocità consentita non è un fattore rilevante; una strada ad accesso libero estende la velocità media evitando fermate e ritardi, all'interno dei limiti di velocità designati.*

ciò non implica l'inevitabilità del pedaggio. Noi che dobbiamo comunque sostenerne le spese, ricaviamo maggior valore dal nostro denaro acquistando una strada libera.

Non sto dicendo che una strada a pagamento sia peggiore di non avere alcuna strada. Ciò sarebbe vero nel caso il pedaggio fosse talmente elevato da impedirne l'utilizzo quasi a tutti – procedura questa improbabile per chi dovesse riscuotere il pedaggio. Tuttavia, finché i caselli provocano spreco e inconvenienti significativi, è meglio raccogliere i fondi necessari in maniera meno limitante. Applicando la medesima logica allo sviluppo del software, passerò ora a dimostrare come l'esistenza di “caselli a pedaggio” per utili programmi risulti assai esosa per la società: rende i programmi più costosi da realizzare, più costosi da distribuire, meno soddisfacenti ed efficaci da usare. Ne consegue che la costruzione dei programmi va incoraggiata in qualche altro modo. Proseguirò poi spiegando altri metodi per incoraggiare e (finché sia realmente necessario) per reperire fondi per lo sviluppo del software.

Il danno provocato dall'ostruzione del software

Consideriamo per un momento lo scenario in cui un programma sia stato sviluppato, e ogni pagamento necessario al suo sviluppo sia stato coperto; ora la società deve scegliere se renderlo proprietario oppure se garantirne condivisione e utilizzo liberi. Come assunto, l'esistenza e la disponibilità del programma sono qualcosa di desiderabile.³

³ *Un particolare programma informatico potrebbe essere considerato dannoso fino a negarne la disponibilità, come è accaduto al database per dati personali Lotus Marketplace, ritirato dal mercato a seguito della disapprovazione del pubblico. Gran parte di quanto sostengo non si applica a casi simili, ma ha poco senso affermare la necessità di un proprietario sulla base del fatto che quest'ultimo potrebbe limitare la disponibilità del programma. Il proprietario non lo metterà mai completamente fuori commercio, come si vorrebbe nel caso di un programma il cui impiego è considerato distruttivo.*

Le restrizioni sulla distribuzione e sulla modifica del programma non possono facilitarne l'utilizzo. Possono soltanto interferire. Così l'effetto sarà soltanto negativo. Ma fino a che punto? E di che tipo?

Questa ostruzione produce tre diversi livelli di danno materiale:

1. Un minor numero di persone usa il programma.
2. Nessun utente può adattare o aggiustare il programma.
3. Gli altri sviluppatori non possono imparare dal programma, né usarlo come base per nuovi lavori.

Ciascun livello di danno materiale riflette una forma concomitante di danno psico-sociale. Ciò si riferisce all'effetto prodotto dalle decisioni della gente sui successivi sentimenti, attitudini e predisposizioni. Questi cambiamenti nel modo di pensare avranno poi un ulteriore effetto sulle relazioni con i concittadini, e potranno causare conseguenze materiali.

I tre livelli di danno materiale fanno sprecare parte del valore che il programma potrebbe offrire, ma non possono ridurlo a zero. Se provocano lo spreco quasi totale del valore, allora scrivere il programma danneggia la società in misura pari allo sforzo necessario per scriverlo. A ragione, un programma che produce dei guadagni nelle vendite deve fornire qualche beneficio materiale diretto. Tuttavia, considerando il concomitante danno psico-sociale, non esistono limiti al danno causato dallo sviluppo del software proprietario.

Impedire l'uso dei programmi

Il primo livello di danno impedisce il semplice utilizzo del programma. La copia di un programma ha un costo marginale quasi vicino a zero (e si può pagare tale costo facendola da soli), così che in un mercato libero avrebbe un prezzo vicino a zero. Una licen-

za a pagamento è un disincentivo significativo nell'uso del programma. Se un programma molto utile è software proprietario, verrà usato da un numero alquanto ridotto di persone.

È facile dimostrare che il contributo complessivo di un programma rispetto alla società risulta ridotto assegnandogli un proprietario. Ogni utente potenziale del programma, di fronte alla necessità di dover pagare per usarlo, potrebbe scegliere di pagare o meno, o anche di rinunciare a utilizzarlo. Quando un utente sceglie di pagare, questo è un trasferimento di ricchezza a quota zero tra due entità. Ma ogni volta che qualcuno decide di rinunciare all'uso del programma, ciò danneggia quell'individuo senza giovare a nessuno. La somma di cifre negative e zeri risulta negativa. Ma ciò non riduce l'ammontare di lavoro necessario per sviluppare il programma. Come risultato, viene ridotta l'efficacia dell'intero processo, ovvero la soddisfazione dell'utente ottenuta per ogni ora di lavoro.

Ciò riflette una differenza cruciale tra le copie di un programma e oggetti quali automobili, sedie o panini. Non esiste una macchina per copiare gli oggetti materiali al di fuori della fantascienza. Ma i programmi sono facili da copiare; chiunque è in grado di produrne quante copie ne vuole con uno sforzo minimo. Ciò non si applica agli oggetti materiali perché la materia si conserva: ogni nuova copia dev'essere costruita dal materiale grezzo nella stessa maniera con cui è stata costruita la prima copia.

Con gli oggetti materiali ha senso creare un disincentivo al loro impiego, perché l'acquisto di un minor numero di oggetti significa minor materiale grezzo e minor lavoro per costruirli. È vero che generalmente esiste anche un costo iniziale d'ammortamento, un costo di sviluppo, che viene frazionato lungo la fase di produzione. Ma finché il costo marginale della produzione è signifi-

cativo, l'aggiunta di una quota del costo di sviluppo non provoca una differenza qualitativa. E non richiede restrizioni sulla libertà dei comuni utenti.

Tuttavia l'imposizione di un prezzo su qualcosa che altrimenti sarebbe libera rappresenta un cambiamento qualitativo. Una tariffa sulla distribuzione del software, imposta a livello centralizzato, diventa un potente disincentivo.

Inoltre, la produzione centrale come viene praticata oggi è inefficace perfino in quanto mezzo per diffondere copie del software. Questo sistema prevede la chiusura di dischi o nastri materiali all'interno di pacchetti inutili, la loro spedizione in gran quantità intorno al mondo, e il loro immagazzinamento per la vendita. Questo costo viene presentato come una spesa di tale attività commerciale; in realtà, è parte dello spreco causato dall'esistenza dei proprietari.

Danneggiare la coesione sociale

Supponiamo che una persona e il proprio vicino considerino utile l'impiego di un certo programma. Con attenzione etica per il vicino, ci si dovrebbe rendere conto che un'adeguata gestione della situazione consentirà a entrambi di utilizzarlo. La proposta di consentire l'uso del programma soltanto a uno dei due, impedendolo all'altro, è divisoria; entrambi la considererebbero inaccettabile.

Firmare un tipico contratto per la licenza del software significa tradire il vicino: "Prometto di privare il mio vicino di questo programma in modo che possa averne una copia tutta per me". Quanti optano per simili scelte subiscono una pressione psicologica interna per giustificarle, diminuendo l'importanza di aiutare il vicino – in tal modo, lo spirito pubblico ne soffre. Questo è il danno psicologico associato con quello materiale di scoraggiare l'uso del programma.

Molti utenti riconoscono inconsciamente l'errore nel rifiuto a condividere, così decidono di ignorare licenze e leggi, e condividere comunque il programma. Ma spesso si sentono in colpa per averlo fatto. Sanno che devono infrangere la legge onde poter essere dei buoni vicini, ma rispettano ancora l'autorità giuridica, e concludono che il fatto di essere dei buoni vicini (come sono in effetti) è qualcosa di male o di cui vergognarsi. Anche questo è un tipo di danno psicologico, che però si può evitare decidendo che tali licenze e leggi non posseggono alcuna forza morale.

Anche i programmatori subiscono il danno psicologico di sapere che a numerosi utenti non verrà consentito di utilizzare il proprio lavoro. Ciò porta a un'attitudine di cinismo o diniego. Uno sviluppatore potrà descrivere in maniera entusiasta un lavoro che considera tecnicamente eccitante; poi quando gli si chiede, "Mi sarà permesso di usarlo?", impallidisce e ammette che la risposta è no. Per evitare di sentirsi scoraggiato, la maggior parte delle volte finisce per ignorare questo fatto oppure adotta un atteggiamento di cinica distanza mirato a minimizzarne l'importanza. Fin dal periodo di Reagan⁴, la maggiore scarsità degli Stati Uniti non riguarda l'innovazione tecnica, ma piuttosto la volontà di lavorare insieme per il bene pubblico. Non ha alcun senso incoraggiare la prima a spese del secondo.

Impedire gli adattamenti personalizzati di programmi

Il secondo livello di danno materiale è l'impossibilità di adattare i programmi. La facilità di modificare il software è uno dei suoi grandi vantaggi rispetto alle vecchie tecnologie. Ma la gran parte

⁴ Ronald Reagan, il 40° Presidente degli Stati Uniti, è famoso per aver tagliato numerosi programmi sociali. A lui si deve inoltre la creazione di una politica economica, definita "trickle down economics" (economia che sgocciola), considerata da molti un fallimento.

del software disponibile in ambito commerciale non può essere modificato, neppure dopo averlo acquistato. È disponibile a scatola chiusa, prendere o lasciare, tutto qui.

Un programma che è possibile far girare è composto da una serie di numeri dall'oscuro significato. Nessuno, neanche un buon programmatore, è in grado di modificare facilmente tali numeri onde rendere il programma diverso in qualche modo.

Normalmente gli sviluppatori lavorano con il “codice sorgente” di un programma, che è scritto in un linguaggio di programmazione tipo Fortran o C. Ricorre a dei nomi per indicare i dati usati e le parti di un programma, e rappresenta le operazioni con simboli quali + per le addizioni e - per le sottrazioni. È progettato per aiutare gli sviluppatori a leggere e modificare il programma. Ecco un esempio; un programma per calcolare la distanza tra due punti su un piano:⁵

```
float
distance (p0, p1)
struct point p0, p1;
{
float xdist = p1.x - p0.x;
float ydist = p1.y - p0.y;
return sqrt (xdist * xdist + ydist * ydist);
}
```

Ecco lo stesso programma in formato eseguibile⁶, sul computer che uso normalmente:

```
1314258944 -232267772 -231844864 1634862
```

⁵ Non è importante capire in che modo operi il codice sorgente; quel che è importante è notare che tale codice sorgente viene scritto ad un livello di astrazione piuttosto comprensibile.

⁶ Si noti l'incomprensibilità del codice eseguibile; è chiaramente più difficile capirne qualcosa rispetto al codice sorgente di cui sopra.

1411907592 -231844736 2159150 1420296208
-234880989 -234879837 -234879966 -232295424
1644167167 -3214848 1090581031 1962942495
572518958 -803143692 1314803317

Il codice sorgente è utile (almeno potenzialmente) per chiunque usi un programma. Ma alla maggioranza degli utenti non è concesso avere copie di tale codice. Normalmente il codice sorgente di un programma proprietario viene tenuto segreto dal proprietario, prevenendo chiunque altro dall'impararne qualcosa. Gli utenti ricevono unicamente i file di numeri incomprensibili che il computer eseguirà. Ciò significa che soltanto il proprietario di un programma può modificarlo.

Una volta un'amica mi raccontò di aver lavorato come programmatore in una banca per circa sei mesi, scrivendo un programma simile a qualcosa di commercialmente disponibile. Se avesse potuto avere il codice sorgente di quel programma commerciale, avrebbe potuto facilmente adattarlo alle necessità del caso. La banca era disposta a pagare per questo, ma non le venne concesso – il codice sorgente era un segreto. Così fu costretta a lavorare in tal modo per sei mesi, lavoro che fa parte del prodotto nazionale lordo ma che in realtà fu uno spreco.

Intorno al 1977 il laboratorio di intelligenza artificiale del MIT ricevette in regalo una stampante grafica dalla Xerox. Era gestita da un software libero a cui aggiungemmo parecchie funzioni utili. Ad esempio, il software avrebbe notificato l'utente non appena finito il lavoro di stampa. In caso di problemi, come l'inceppamento dei fogli o la mancanza di carta, il software ne avrebbe informato tutti gli utenti che avevano in corso una stampa. Queste funzioni facilitavano il corso delle operazioni.

Più tardi la Xerox diede al laboratorio una stampante più nuova e

veloce, una delle prime stampanti laser. Veniva gestita da un software proprietario che girava su un apposito computer dedicato, in modo che non potemmo aggiungere alcuna delle nostre opzioni favorite. Riuscimmo a sistemare l'invio di una notifica quando la stampa veniva inviata al computer dedicato, ma non quando avveniva effettivamente la stampa (e generalmente il ritardo era considerevole). Non c'era alcun modo di sapere quando il lavoro veniva realmente stampato, si poteva solo indovinare. E nessuno veniva informato nel caso di fogli incastrati, così spesso la stampante finiva fuori uso per un'ora.

Il sistema di programmatori del laboratorio di intelligenza artificiale era in grado di sistemare simili problemi, probabilmente tanto quanto gli autori originari del programma. Ma la Xerox non aveva alcun interesse a risolverli, e scelse di impedircelo, di modo da costringerci ad accettare tali problemi. Non vennero mai risolti. Molti buoni programmatori hanno sperimentato una simile frustrazione. La banca poteva permettersi di risolvere il problema scrivendo un nuovo programma da zero, ma un comune utente, a prescindere dalle proprie capacità, può soltanto rinunciare.

La rinuncia provoca un danno psico-sociale – allo spirito della fiducia in se stessi. È demoralizzante vivere in una casa che non si può riarrangiare secondo i propri bisogni. Porta a rassegnazione e scoraggiamento, che finiscono per colpire altri aspetti della vita di una persona. Chi si trova in simili circostanze diventa infelice e non fa un buon lavoro.

Immaginiamo come sarebbe qualora le ricette venissero trattate alla stregua del software. Qualcuno potrebbe dire, “Come faccio a modificare questa ricetta per toglierci il sale?” e il grande cuoco risponderebbe, “Come osi insultare la mia ricetta, il prodotto del mio cervello e del mio palato, cercando di interferire? Non pos-

siedi il giudizio necessario per poterla modificare e farla funzionare bene!”

“Ma il dottore mi ha detto di non mangiare cibi salati!”

“Sarò contento di farlo. La mia tariffa è di appena 50.000 dollari”. (Dato che il proprietario ha il monopolio sulle modifiche, la tariffa tende ad essere elevata). “Però adesso non ho tempo. Sono occupato con una commissione per il progetto di una nuova ricetta di biscotti per le navi con il Ministero della Marina. Potrò occuparmi di te fra un paio d’anni.”

Impedire lo sviluppo del software

Il terzo livello di danno materiale colpisce lo sviluppo del software. Solitamente questo era un processo evolutivo, in cui una persona ne riscriveva delle parti per una nuova funzione, e poi qualcun altro ne avrebbe riscritto altre parti per aggiungere un’altra funzione; in alcuni casi, ciò andava avanti per un periodo di vent’anni. Nel frattempo, parti del programma sarebbero state “cannibalizzate” per dar forma alla nascita di ulteriori programmi. L’esistenza di proprietari impedisce un’evoluzione di questo tipo, rendendo necessario partire da zero nello sviluppo di un programma. Impedisce altresì a nuovi praticanti di studiare i programmi esistenti onde imparare tecniche utili o anche la struttura di programmi di ampie dimensioni.

I proprietari impediscono anche l’educazione. Ho incontrato brillanti studenti d’informatica che non avevano mai visto il codice sorgente di un programma di ampie dimensioni. Possono essere bravi a scrivere piccoli programmi, ma non potranno acquisire le diverse capacità necessarie per scriverne di grandi se non possono vedere come hanno fatto gli altri.

In ogni ambito intellettuale, è possibile raggiungere altezze mag-

giori stando sulle spalle di chi ci ha preceduti. Ma in genere ciò non è più consentito nel campo del software – si può stare soltanto sulle spalle dei colleghi *della propria azienda*.

Il danno psico-sociale associato colpisce lo spirito della cooperazione scientifica, solitamente così solido tra i ricercatori al punto che questi collaboravano anche quando i rispettivi paesi erano in guerra tra loro. Sulla base di questo spirito, gli oceanografi giapponesi abbandonati in un laboratorio su un'isola del Pacifico conservarono con attenzione il lavoro svolto per i Marine statunitensi in arrivo, lasciando una nota per chiedere loro di prendersene cura.

I conflitti per denaro hanno distrutto quello che si era salvato nei conflitti internazionali. Oggigiorno i ricercatori di molte discipline non pubblicano dati sufficienti nelle proprie ricerche onde consentire agli altri di replicare quegli esperimenti. Pubblicano soltanto quanto basta per fare in modo che i lettori possano meravigliarsi di quanto siano riusciti a ottenere. Ciò è sicuramente vero per l'informatica, dove il codice sorgente dei programmi su cui si scrive è generalmente segreto.

Non importa come si limiti la condivisione

Ho discusso finora gli effetti di prevenire la gente dall'attività di copia, modifica e costruzione su un programma. Non ho specificato il modo in cui questa ostruzione viene portata avanti, perché ciò non ne influenza la conclusione. Sia che ciò venga imposto tramite il divieto di copia, o il diritto d'autore, o le licenze, o la citazione, o le schede ROM, o i numeri seriali sull'hardware, se riesce a impedirne l'utilizzo, allora procura dei danni. Gli utenti considerano comunque alcuni di questi metodi più sgradevoli di altri. Io suggerirei che i metodi più odiosi sono quelli che raggiungono lo scopo prefissato.

Il software dovrebbe essere libero

Fin qui ho illustrato il modo in cui la proprietà di un programma – il potere di limitarne la modifica o la copia – sia d'intralcio. I suoi effetti negativi sono diffusi e importanti. Ne consegue che la società non dovrebbe avere proprietari per i programmi.

Un altro modo di comprenderlo è che ciò di cui abbisogna la società è il software libero, e il software proprietario ne è un sostituto insoddisfacente. Incoraggiare i sostituti non è un modo razionale di ottenere ciò di cui abbiamo bisogno.

Vaclav Havel ci ha messo sull'avviso dicendo, "Lavorare per qualcosa perché è bene, non soltanto perché si ha possibilità di riuscire". L'attività di realizzare software proprietario contiene possibilità di riuscita in termini ristretti, ma non è un bene per la società.

Perché si sviluppa il software

Se eliminiamo il copyright come mezzo per incoraggiare la gente a sviluppare software, all'inizio se ne svilupperà di meno, ma tale software risulterà maggiormente utile. Non è chiaro se la soddisfazione generale degli utenti sarà minore; ma se così fosse, oppure se volessimo incrementarla comunque, esistono altri modi per incoraggiare lo sviluppo, proprio come esistono altri modi oltre i caselli a pedaggio per raccogliere denaro per la costruzione di strade. Prima di illustrare le modalità con cui ciò può esser fatto, vorrei affrontare la questione di quanto incoraggiamento artificiale sia davvero necessario.

Programmare è divertente

Ci sono alcuni tipi di lavori che pochi accetterebbero se non per denaro; la costruzione di strade, ad esempio. Esistono altri campi di studio e arte in cui esistono scarse possibilità di diventare ric-

chi, ma che la gente sceglie per il loro fascino o per il presunto valore agli occhi della società. Gli esempi includono la logica matematica, la musica classica, l'archeologia e l'attivismo politico organizzato tra i lavoratori. La gente compete, più tristemente che amaramente, per le poche posizioni pagate disponibili, nessuna delle quali è remunerata granché. Qualcuno è perfino disposto a pagare di tasca propria pur di lavorare in quel campo, se può permetterselo.

Un certo settore può trasformarsi nel giro di una notte se inizia ad offrire la possibilità di diventare ricchi. Quando un lavoratore diventa ricco, altri chiedono la medesima opportunità. Presto tutti potrebbero volere ingenti somme di denaro per fare quel che erano soliti fare per il piacere. Trascorsi un altro paio d'anni, chiunque coinvolto in quel campo deriderà l'idea che si debba lavorare in quel settore senza grossi ricavi economici. Spiegheranno ai pianificatori sociali che è possibile assicurare tali ricavi, assegnando privilegi sociali, poteri e monopoli man mano che si renderà necessario.

Questo è il cambiamento avvenuto nel campo della programmazione informatica durante lo scorso decennio. Quindici anni fa⁷ giravano articoli sulla "computer-dipendenza": gli utenti erano "on-line" tutto il tempo e avevano vizi da cento dollari a settimana. In genere si accettava il fatto che la gente amava a tal punto la programmazione da rompere non di rado il proprio matrimonio. Oggi, in genere si accetta che nessuno farebbe il programmatore se non in cambio di un lauto stipendio. La gente ha dimenticato come stavano le cose quindici anni fa.

Anche se fosse vero che ad un certo punto la maggior parte della gente lavorerà in un certo campo soltanto per un lauto stipendio,

⁷ *Quindici anni prima della stesura di quest'articolo correva l'anno 1977.*

lo scenario non deve necessariamente rimanere tale. La dinamica del cambiamento può girare all'inverso, qualora la società fornisca l'input adatto. Se eliminiamo la possibilità di grandi ricchezze, allora dopo qualche tempo, una volta risistemate le attitudini personali, la gente sarà nuovamente ansiosa di lavorare in quel campo per la gioia di riuscire.

La domanda "Come fare a pagare i programmatori?" trova facile risposta una volta compreso che non si tratta di pagarli una fortuna. Il semplice guadagnarsi da vivere è più facile da mettere insieme.

Finanziare il software libero

Le istituzioni che pagano i programmatori non devono essere i produttori di software. Esistono già parecchie altre istituzioni in grado di farlo.

I produttori di hardware considerano essenziale sostenere lo sviluppo del software pur non potendone controllare l'utilizzo. Nel 1970 gran parte del loro software era libero perché non si curavano di porre limitazioni. Oggi la crescente volontà di aderire a dei consorzi dimostra che hanno compreso come possedere il software non è una cosa veramente importante per loro.

Le università svolgono numerosi progetti di programmazione. Oggi spesso ne rivendono i risultati, ma non fu così negli anni '70. Esiste forse alcun dubbio che le università svilupperebbero software libero qualora non fosse loro consentito di vendere il software? Questi progetti potrebbero essere finanziati dagli stessi contratti e borse di studio governativi che attualmente finanziano lo sviluppo di software proprietario.

Oggi è comune per i ricercatori universitari ottenere borse di studio per lo sviluppo di un sistema, realizzarlo fino quasi al punto

finale e definirlo “completato”, per poi fondare delle aziende in cui portano davvero a compimento il progetto e lo rendono utilizzabile. Talvolta dichiarano “libera” la versione non finita; se sono corrotti fino in fondo, ottengono invece una licenza esclusiva dall’università. Ciò non è certo un segreto, viene ammesso apertamente da ogni soggetto coinvolto. Eppure se i ricercatori non fossero esposti alla tentazione di comportarsi in questo modo, proseguirebbero tuttora le proprie ricerche.

Gli sviluppatori che scrivono software libero possono guadagnarsi da vivere vendendo servizi connessi al software. Io sono stato assunto per portare il compiler GNU C su un nuovo hardware, e per realizzare le estensioni dell’interfaccia-utente per GNU Emacs. (Offrirò queste migliorie al pubblico una volta completate). Tengo anche dei corsi per cui vengo pagato.

Non sono il solo a lavorare in tal modo; oggi esiste una corporation di successo e in crescita che fa soltanto questi tipi di lavori. Anche diverse altre aziende offrono supporto commerciale per il software libero del sistema GNU. Questo è l’inizio dell’industria a sostegno del software indipendente – un’industria che potrebbe raggiungere dimensioni piuttosto ampie se il software libero diventasse prevalente. Ciò offre agli utenti un’opzione generalmente non disponibile per il software proprietario, eccetto a chi è molto ricco.

Nuove⁸ istituzioni quali la Free Software Foundation possono altresì sostenere i programmatori.

Gran parte dei finanziamenti della Foundation arrivano dagli acquirenti di dischi e nastri tramite posta. Il software sui nastri è libero, il che significa che ogni utente ha libertà di copiarlo e modificarlo, ma in ogni caso molti pagano per averne delle copie.

⁸ *Quest’articolo è stato scritto il 24 aprile 1992.*

(Ricordiamoci che “free software” si riferisce alla libertà, non al prezzo). Alcuni utenti già in possesso di una copia, ordinano i nastri come modo per offrire l’obolo che secondo loro noi meritiamo. La Foundation riceve inoltre donazioni di una certa ampiezza dai produttori di computer.

La Free Software Foundation è un ente senza fini di lucro, e i ricavi vengono spesi per ingaggiare quanti più programmatori possibile. Se fosse stata impostata come attività commerciale, distribuendo al pubblico lo stesso software libero per la medesima cifra odierna, fornirebbe un’ottima fonte di sostentamento al suo fondatore.

Essendo la Foundation un ente senza fini di lucro, spesso i programmatori vi lavorano per metà della cifra che potrebbero chiedere altrove. Lo fanno perché siamo liberi dalla burocrazia, e perché sono soddisfatti nel sapere che il loro lavoro non subirà ostruzioni nell’utilizzo. Ma soprattutto lo fanno perché programmare è divertente. In aggiunta, dei volontari non pagati hanno scritto per noi molti programmi utili. (Abbiamo perfino scrittori tecnici volontari).

Ciò conferma come la programmazione sia tra i campi più affascinanti di tutti, insieme alla musica e all’arte. Non dobbiamo temere che non ci sarà gente che vorrà programmare.

Cosa devono gli utenti agli sviluppatori?

C’è una buona ragione per chi usa il software di sentirsi moralmente obbligati a contribuire al suo sostegno. Gli sviluppatori di software libero contribuiscono alle attività degli utenti, e oltre che giusto è nell’interesse a lungo termine degli stessi utenti dare loro i finanziamenti per continuare.

Ciò tuttavia non si applica a chi sviluppa software proprietario, per-

ché l'ostruzionismo merita una punizione anziché una ricompensa. Eccoci così di fronte a un paradosso: chi sviluppa software utile ha diritto al sostegno degli utenti, ma qualsiasi tentativo di trasformare quest'obbligo morale in un requisito distrugge le basi stesse di tale obbligo. Uno sviluppatore può meritare o richiedere una ricompensa, ma non entrambe le cose.

Credo che di fronte a tale paradosso uno sviluppatore dotato di senso etico deve fare in modo di meritare la ricompensa, ma dovrebbe anche stimolare gli utenti alle donazioni volontarie. Alla fin fine gli utenti impareranno a sostenere gli sviluppatori senza costrizione, così come hanno imparato a sostenere le stazioni radio e televisive pubbliche.

Cos'è la produttività del software?

Se il software fosse libero, esisterebbero ancora i programmatori, ma forse in numero minore. Ciò sarebbe un male per la società? Non necessariamente. Oggi le nazioni avanzate hanno meno agricoltori che nel 1900, ma non lo consideriamo un male per la società, visto che un numero minore offre ai consumatori una quantità maggiore di cibo. Ciò viene definito miglioramento produttivo. Il software libero richiederebbe un numero assai minore di programmatori per soddisfare la domanda, per via dell'accresciuta produttività del software a tutti i livelli:

- Utilizzo più ampio di ciascun programma sviluppato.
- Capacità di adattare i programmi esistenti per la personalizzazione, invece di partire da zero.
- Migliore educazione dei programmatori.
- L'eliminazione dei doppioni nei progetti di sviluppo.

Quanti si oppongono alla cooperazione sostenendo che provocherebbe l'assunzione di un numero minore di programmatori van-

no in realtà opponendosi alla maggiore produttività. Eppure costoro generalmente accettano la credenza comune secondo cui l'industria del software necessita di un incremento produttivo. Come mai?⁹ La “produttività del software” può avere due significati diversi: la produzione complessiva dell'intero settore di sviluppo del software oppure la produttività di progetti individuali. La produttività complessiva è quel che la società vorrebbe migliorare, e la maniera più diretta per farlo è eliminare gli ostacoli artificiali alla cooperazione che riducono tale produttività. Ma i ricercatori che studiano il campo della “produttività del software” si concentrano unicamente sul secondo, limitato, significato del termine, dove il miglioramento richiede difficili avanzamenti tecnologici.

La competizione è inevitabile?

È inevitabile che si cerchi di competere, di sorpassare i propri rivali nella società? Forse lo è. Ma la competizione in se stessa non è dannosa; la cosa dannosa è il combattimento.

Esistono molti modi di competere. La competizione può consistere nel cercare di raggiungere sempre di più, di superare quel che hanno fatto gli altri. Ad esempio, in passato c'era competizione tra i maghi della programmazione – competizione per chi riusciva a far fare al computer le cose più incredibili, o per chi riusciva a creare il programma più breve o più veloce per un particolare compi-

⁹ Secondo Eric Raymond, il 95% dei posti di lavoro dell'industria del software riguarda la produzione di software personalizzato, nient'affatto previsto per la pubblicazione. Ne consegue che pur assumendo lo scenario teorico peggiore, ovvero che non esisteranno posti di lavoro per lo sviluppo di software libero (e già sappiamo che ne esistono alcuni), il passaggio al software libero potrà avere uno scarso effetto sul numero totale di posti di lavoro per il software. Esiste una gran quantità di spazio per chi voglia scrivere software personalizzato e sviluppare software libero quando avanza tempo. Non esiste alcun modo per sapere se la piena conversione al software libero porterebbe all'aumento o alla diminuzione del numero di posti di lavoro nel campo del software.

to. Questo tipo di competizione può giovare a chiunque, fintantoché si mantiene lo spirito della buona lealtà sportiva.

La competizione costruttiva è sufficientemente competitiva da spingerci a fare grandi sforzi. Un certo numero di persone stanno gareggiando per essere i primi ad aver visitato tutti i paesi sulla terra; qualcuno spende anche una fortuna nel tentativo di riuscirci. Ma non cercano di corrompere i capitani delle navi perché abbandonino i rivali su un'isola deserta. Sono contenti di lasciar vincere la persona più in gamba.

La competizione diventa lotta quando coloro che gareggiano iniziano a bloccarsi a vicenda invece di pensare al proprio avanzamento – quando “lasciar vincere la persona più in gamba” si trasforma in “lasciar vincere me stesso, più in gamba o meno che sia”. Il software proprietario è dannoso, non perché sia una forma di competizione, ma perché è una forma di combattimento tra i cittadini della società.

Nell'imprenditoria competizione non significa necessariamente lotta. Ad esempio, quando due negozi di alimentari sono in competizione, i loro sforzi si concentrano sul miglioramento delle proprie operazioni, non sul sabotaggio del rivale. Ma ciò non dimostra un particolare attaccamento all'etica commerciale; piuttosto, ha poco senso combattere in questo tipo di attività senza ricorrere alla violenza fisica. Non tutti i settori imprenditoriali condividono questa caratteristica. Tenere segrete informazioni che potrebbero aiutare tutti ad avanzare è una forma di combattimento. L'ideologia commerciale non prepara la gente a resistere alla tentazione di lottare come forma di competizione. Alcuni tipi di combattimento sono stati vietati con legislazioni anti-monopolio, leggi sulla veridicità della pubblicità, e così via, ma anziché generalizzare ciò in un principio di rifiuto della lotta in generale, i diri-

genti hanno inventato altre forme di combattimento che non sono specificamente proibite. Le risorse della società vengono sperperate nell'equivalente economico di una guerra civile tra fazioni.

“Perché non te ne vai in Russia?”

Negli Stati Uniti chiunque sostenga qualsiasi posizione diversa dalla forma più estrema di *laissez-faire* egoistico ha sentito spesso quest'accusa. Viene ad esempio usata contro i sostenitori di un sistema nazionale d'assistenza sanitaria, come ne esistono in tutte le altre nazioni industrializzate del mondo libero. Viene usata contro i sostenitori del sostegno pubblico alle arti, anch'esso universale nei paesi avanzati. In America l'idea che i cittadini abbiano qualche obbligo nei confronti del bene pubblico viene identificata con il comunismo. Ma si tratta davvero di idee similari?

Il comunismo per come fu praticato nell'Unione Sovietica era un sistema di controllo centralizzato dove tutta l'attività era irreggimentata, apparentemente per il bene comune, ma in realtà a vantaggio dei membri del partito comunista. E dove i dispositivi per la copia erano sorvegliati da vicino onde evitare la copia illegale. Il sistema americano del diritto d'autore sul software impone il controllo centralizzato sulla distribuzione di un programma, e i dispositivi per la copia sono sorvegliati tramite sistemi anti-copia automatici onde evitare la copia illegale.

All'opposto, il mio lavoro punta alla costruzione di un sistema in cui la gente sia libera di decidere sulle proprie azioni; in particolare, libera di aiutare i vicini, e libera di alterare e migliorare gli strumenti che usano nella vita quotidiana. Un sistema basato sulla cooperazione volontaria e sulla decentralizzazione. Perciò, se dovessimo giudicare le posizioni sulla somiglianza al comunismo russo, sarebbero i proprietari di software a essere comunisti.

La questione delle premesse

In questo saggio parto dalla premessa che l'utente di software non sia meno importante di un autore, o anche del datore di lavoro di un autore. In altri termini, gli interessi e le necessità di tutti costoro hanno un uguale peso quando si tratta di decidere il miglior corso d'azione. Questa premessa non è accettata a livello universale. Molti sostengono come il datore di lavoro di un autore sia essenzialmente più importante di chiunque altro. Si dice, ad esempio, che lo scopo nell'aver proprietari di software è quello di dare al datore di lavoro di un autore il vantaggio che merita – prescindendo dal modo in cui ciò possa influenzare il pubblico.

È inutile cercare di convalidare o confutare tali premesse. Ogni prova si basa su premesse condivise. Perciò gran parte di quanto vado sostenendo è indirizzato soltanto a quanti condividono le premesse che uso, o almeno a quanti sono interessati a vederne le conseguenze. Per quanti ritengono che i proprietari siano più importanti di chiunque altro, questo saggio è semplicemente irrilevante.

Ma perché un gran numero di americani dovrebbe accettare una premessa che eleva l'importanza di alcuni individui su chiunque altro? In parte perché ci si basa sulla credenza che tale premessa faccia parte della tradizione legale della società americana. Per qualcuno, dubitare della premessa significa mettere in discussione le basi stesse della società.

È importante informare costoro che tale premessa non è parte della nostra tradizione legale. Né lo è mai stata.

Ovvero, la Costituzione dice che lo scopo del copyright è quello di “promuovere il progresso della scienza e delle arti utili”. La Corte Suprema ha elaborato su quest'idea, affermando nella causa *Fox Film vs. Doyal* che “l'unico interesse degli Stati Uniti e l'oggetto primario nel conferire il monopolio [del copyright] risiede nei

benefici generali derivanti al pubblico dal lavoro degli autori”. Non siamo obbligati a essere d’accordo con la Costituzione o con la Corte Suprema. (A un certo punto, entrambi perdonarono la schiavitù). Le loro posizioni non condannano la premessa sulla supremazia del proprietario. Spero però che la consapevolezza per cui ciò sia un assunto della destra radicale, piuttosto che un fatto tradizionalmente riconosciuto, perderà il proprio fascino.

Conclusione

Ci piace pensare che la società incoraggi l’aiuto al vicino; ma ogni volta che ricompensiamo qualcuno perché fa ostruzione, o lo ammiriamo per la ricchezza accumulata in tal modo, stiamo inviando il messaggio opposto.

L’accumulazione del software è una forma della nostra volontà generale di non considerare il benessere della società a favore del profitto personale. Possiamo notare questa mancanza di considerazione da Ronald Reagan a Jim Bakker¹⁰, da Ivan Boesky¹¹ a Exxon¹², dalle banche fallite alle scuole fallite. Possiamo misurarla con la quantità di gente senza casa e di popolazione carceraria. Lo spirito antisociale si nutre da solo, perché più ci rendiamo conto che gli altri non ci aiuteranno, più sembra futile aiutarli. Così la società si trasforma in una giungla.

¹⁰ Negli anni ‘80 Jim Bakker raccolse milioni di dollari in televisione per i suoi gruppi religiosi Heritage USA, PTL e Inspirational Network. Venne condannato a 45 anni di carcere per frode via posta e banca per le campagne di raccolta fondi a favore di PTL.

¹¹ Ivan Boesky fu mandato in prigione e multato per 100 milioni di dollari per trading scorretto negli anni ‘80. Divenne famoso per aver detto una volta, “L’avarizia è un bene. Voglio farvi sapere che ritengo salutare l’avarizia. Potete essere avari e sentirvi comunque in pace con voi stessi”.

¹² Negli anni ‘80 la Exxon Valdez provocò la più vasta fuoriuscita di petrolio al mondo al largo delle coste dell’Alaska, causando danni immensi. Finora le multe e le operazioni di pulizia gli sono costate oltre un miliardo di dollari.

Se non vogliamo vivere in una giungla, dobbiamo modificare il nostro atteggiamento. Dobbiamo iniziare a veicolare il messaggio che un buon cittadino è quello che coopera quando appropriato, non quello che è bravo a prendere dagli altri. Spero che il movimento del software libero possa offrire dei contributi in tal senso: almeno in un campo, sostituiremo la giungla con un sistema più efficace che incoraggi e giri sulla cooperazione volontaria.

Originalmente scritto nel 1992, questa versione fa parte del libro *Free Software, Free Society: The Selected Essays of Richard M. Stallman*, GNU Press, 2002.

La copia letterale e la distribuzione di questo testo nella sua integrità sono permesse con qualsiasi mezzo, a condizione che sia mantenuta questa nota.

Diritto d'autore e globalizzazione nell'era delle reti informatiche

Introduzione

David Thorburn, moderatore: Il relatore di oggi, Richard Stallman, è una figura leggendaria del mondo informatico, e la mia esperienza nel tentativo di trovare un correlatore per condividere il podio con lui è stata istruttiva. Un distinto professore del MIT mi ha detto che Stallman va considerato al pari della figura carismatica di una parabola biblica – una sorta di aneddoto-lezione del Vecchio Testamento. “Immagina”, mi ha detto, “un Mosè o un Geremia – meglio, un Geremia”. Gli ho replicato: “Bene, davvero ammirevole. Ciò conferma la mia sensazione sul tipo di contributo che ha dato al mondo. Allora perché sei riluttante a dividere il podio con lui?” La sua risposta: “Come Geremia o Mosè, ne sarei semplicemente sopraffatto. Non posso tenere un intervento insieme a lui, ma se mi avessi chiesto di nominare cinque persone viventi nel mondo che hanno veramente aiutato tutti noi, Richard Stallman sarebbe uno di questi.”

L'intervento di Stallman

Dovrei iniziare spiegando perché ho rifiutato il permesso alla trasmissione sul web di questo forum, nel caso la questione non fos-

se sufficientemente chiara: il programma impiegato per la trasmissione sul web richiede all'utente di prelevare determinato software per poter ricevere il segnale. Questo software non è software libero. È disponibile a costo zero ma soltanto come eseguibile, cioè una misteriosa sequenza di numeri.

Ciò che fa è segreto. Non lo si può studiare, non lo si può modificare, e certamente non se ne può pubblicare una versione modificata. E queste rientrano fra le libertà essenziali incluse nella definizione di "software libero".

Così, se devo essere un onesto sostenitore del software libero, non posso andare in giro a fare discorsi e poi spingere le persone ad usare software non libero. Nuocerei alla mia stessa causa. E se non dimostro di prendere i miei principi sul serio, non posso aspettarmi che altri li prendano sul serio.

Tuttavia il mio intervento non riguarda il software libero. Dopo aver lavorato per anni nel movimento per il software libero e aver visto la gente usare alcune parti del sistema operativo GNU, ho iniziato a ricevere inviti per tenere interventi nei quali il pubblico prese a chiedermi: "Bene, come è possibile ampliare le idee sulla libertà per gli utenti del software ad altri tipi di cose?"

E naturalmente facevano domande stupide tipo, "Dovrebbe forse essere libero anche l'hardware?", "Dovrebbe essere libero questo microfono?"

Cosa significa tutto ciò? Dovremmo essere liberi di copiarlo e modificarlo? Per quanto riguarda le modifiche, una volta comprato il microfono, nessuno ci impedirà di modificarlo. E per copiarlo, nessuno possiede una copiatrice di microfoni. Al di fuori di *Star Trek*, queste cose non esistono. Forse un giorno ci saranno analizzatori ed assemblatori nanotecnologici, e sarà realmente possibile copiare un oggetto fisico, e allora il problema del se si è

liberi di farlo o meno comincerà ad essere davvero importante. Vedremo aziende agricole che vorranno impedire alla gente di copiare il cibo, e questo diventerà un problema politico importante, se mai esisterà una simile capacità tecnologica. Non so se succederà, a questo punto si tratta soltanto di pura speculazione. Ma per altri tipi di informazioni, il problema può essere sollevato perché ogni tipo di informazione che può essere memorizzata su un computer, presumibilmente, può essere copiata e modificata. Così i problemi etici del software libero, i problemi del diritto di un utente di copiare e modificare il software, sono identici a quelli relativi ad altri tipi di informazioni pubblicate. Non mi riferisco a informazioni private, diciamo, i dati personali, che non devono essere mai rese disponibili al pubblico. Parlo dei diritti di un utente quando ottiene copie di cose pubblicate senza alcun tentativo di tenerle segrete.

La storia del copyright

Per meglio illustrare le mie idee in materia, vorrei rivedere la storia della distribuzione delle informazioni e del diritto d'autore. Nel mondo antico, i libri erano scritti a mano con una penna, e chiunque sapesse leggere e scrivere poteva copiare un libro in maniera efficace come chiunque altro. Probabilmente qualcuno che vi si dedicava tutto il giorno, aveva imparato a farlo meglio, ma non esisteva alcuna differenza sostanziale. E poiché le copie erano fatte una per volta, non c'era un grande valore economico. Fare dieci copie richiedeva dieci volte il tempo necessario per fare una copia. Non esisteva alcuna costrizione verso la centralizzazione – un libro poteva essere copiato ovunque.

A causa di questa tecnologia, poiché non imponeva che le copie fossero identiche, nel mondo antico non esisteva una differenza

sostanziale tra copiare e scrivere un libro. Ci sono cose nel mezzo che avevano senso. Comprendevano l'idea di un autore. Sapevano, diciamo, che una certa commedia era stata scritta da Sofocle, ma tra scrivere un libro e copiare un libro, c'erano altre cose utili che si potevano fare. Per esempio, si poteva copiare parte di un libro, poi scrivere alcune parole nuove, copiare ancora e scrivere alcune nuove parole e così via. Questo lo si definiva "scrivere un commentario". Si trattava di un'attività piuttosto comune, e questi commentari erano apprezzati.

Si poteva anche copiare un passaggio da un libro, poi scrivere altre parole, e copiare un passaggio da un altro libro e scrivere ancora e così via, e così si creava un compendio. Anche i compendi erano molto utili. Ci sono opere che sono andate perdute, ma parti di esse sono sopravvissute se riprese in altri libri che hanno avuto più popolarità dell'originale. Forse si sono copiate le parti più interessanti, e così se ne sono fatte molte copie, ma non ci si è preoccupati di copiare l'originale, perché non era abbastanza interessante.

Ora, per quanto posso saperne, nel mondo antico non esistevano cose come il diritto d'autore. Chiunque volesse copiare un libro poteva farlo. Più avanti, è stata sviluppata la tecnica della stampa e i libri hanno iniziato ad essere stampati. La stampa non è stata solo un miglioramento quantitativo per copiare in maniera più agevole. Ha influenzato i diversi tipi di copia in modo diverso perché ha introdotto una certa economia di scala. Occorreva lavorare parecchio per impostare la macchina tipografica, e assai meno lavoro per fare molte copie identiche della pagina. Come risultato, la copia dei libri iniziò a diventare un'attività centralizzata di produzione di massa. Probabilmente le copie di un certo libro saranno state fatte soltanto in determinati luoghi.

Ciò significava inoltre che i comuni lettori non potevano copiare i libri in maniera efficace. Ci si riusciva solo se si aveva una macchina per la stampa: era un'attività industriale.

Ora, nei primi secoli di vita della stampa, i libri stampati non hanno sostituito totalmente le copie fatte a mano. Se ne facevano ancora, a volte dai ricchi e a volte dai poveri. I ricchi lo facevano per avere una copia particolarmente bella che avrebbe dimostrato quanto erano ricchi, e i poveri lo facevano perché forse non avevano abbastanza soldi per comprare una copia stampata, ma avevano tempo per copiare un libro a mano. Come dice la canzone, "Il tempo non è denaro quando tutto ciò che hai è il tempo". Così, si facevano ancora copie a mano. Fu nel 1800, credo, che la stampa divenne in realtà abbastanza economica da consentire anche ai poveri di acquistare libri stampati, se sapevano leggere. Ora, il diritto d'autore si è sviluppato insieme all'uso della stampa e, considerando questa tecnologia, ha avuto l'effetto di una regolamentazione industriale. Non regolava quel che fosse concesso fare ai lettori; limitava ciò che potevano fare gli editori e gli autori. Inizialmente in Inghilterra il diritto d'autore è stato una forma di censura. Per pubblicare un libro bisognava avere il permesso del governo. Ma poi il concetto è cambiato. Con la Costituzione degli Stati Uniti, si giunse ad un'idea diversa sugli scopi del diritto d'autore, e credo che tale idea venisse accettata anche in Inghilterra.

Per la Costituzione statunitense venne proposto di assegnare il copyright agli autori, un monopolio sulla copia dei propri libri. Questa proposta fu respinta. Se ne adottò invece una fondamentalmente diversa e cioè che, per promuovere il progresso, il Congresso poteva stabilire eventualmente un sistema di diritti d'autore che avrebbe poi creato questi monopoli. Così i monopoli,

secondo la Costituzione statunitense, non esistono per il bene di chi li possiede; esistono per promuovere il progresso della scienza. I monopoli sono concessi agli autori come modalità per modificare il proprio comportamento onde fare qualcosa di utile per il pubblico.

Così l'obiettivo è avere un maggior numero di libri scritti e pubblicati che gli altri possano poi leggere. E si ritiene che questo [il copyright] contribuisca ad una maggiore attività letteraria, ad ampliare la produzione di scritti scientifici e in altri campi, e che la società possa imparare grazie a ciò. Questo è il fine da perseguire. La creazione di monopoli privati è stata solo un mezzo per raggiungere un certo fine, e questo è un fine pubblico.

Nell'epoca della stampa il diritto d'autore era praticamente indolore perché si trattava di una regolamentazione industriale. Limitava soltanto le attività di editori ed autori. In senso stretto, anche i poveri che copiavano i libri a mano potevano infrangere il diritto d'autore. Ma nessuno cercò mai di imporre il copyright nei loro confronti perché venne considerato una regolamentazione a livello industriale.¹

Inoltre, nell'era della stampa era semplice imporre diritto d'autore, perché andava applicato soltanto laddove esistesse un editore, e gli editori, per la natura stessa della loro attività, si facevano conoscere. Se si cerca di vendere libri, bisogna dire al pubblico dove andare a comprarli. Non si va nelle case di tutti ad imporre il diritto d'autore.

Infine, in quel contesto il diritto d'autore può essere stato un sistema benefico. Negli Stati Uniti il copyright viene considerato dagli studiosi di legge un patto tra il pubblico e gli autori. Il pubblico

¹ *Gli statuti originali parlavano soltanto di editoria e stampa. Non esisteva alcuna regolamentazione per la copia a mano – molto probabilmente perché la regolamentazione riguardava l'industria.*

scambia alcuni dei propri diritti naturali a fare copie, e in cambio ottiene il beneficio di avere un maggior numero di libri scritti e pubblicati.

Ora, si tratta di un patto vantaggioso? Be', quando il pubblico non può fare delle copie perché queste vengono fatte in maniera efficace solo con macchine per la stampa – e la maggior parte della gente non possiede tali macchine – il risultato è che il pubblico sta scambiando una libertà che non può esercitare, una libertà che non è di nessun valore pratico. Perciò se si possiede qualcosa che non è di primaria importanza o è inutile, e si ha la possibilità di scambiarla per qualcos'altro di un qualche valore, ci si guadagna. Ecco perché a quel tempo può darsi che il diritto d'autore sia stato un patto vantaggioso per il pubblico.

Ma il contesto va mutando, e ciò deve cambiare la nostra valutazione etica del diritto d'autore. Ora, i principi alla base dell'etica non sono modificati dai progressi nella tecnologia; sono troppo fondamentali per essere toccati da simili contingenze. Ma le nostre decisioni su una determinata questione dipendono dalle conseguenze delle alternative disponibili, e le conseguenze di una certa scelta possono cambiare quando cambia il contesto. Questo è ciò che succede nell'area della legge sul copyright, perché l'epoca della stampa sta per chiudersi, lasciando gradualmente spazio all'era delle reti informatiche.

Le reti informatiche e le tecnologie dell'informazione digitale ci riportano ad un ambito più simile al mondo antico, dove chiunque sapesse leggere ed usare le informazioni poteva anche copiarle e poteva fare copie facilmente al pari di chiunque altro. Si tratta di copie perfette e valide quanto le copie che potrebbe fare chiunque altro. Così la centralizzazione e l'economia introdotta dalla stampa e da simili tecnologie va scomparendo.

Questo mutamento nel contesto generale cambia il modo in cui funziona la legislazione in tema di diritto d'autore. Il copyright non svolge più una funzione di regolamentazione dell'industria: è diventata una restrizione draconiana imposta al pubblico. Originariamente tale legislazione voleva essere una restrizione imposta agli editori a favore degli autori; oggi, all'atto pratico, è una restrizione imposta al pubblico a favore degli editori. Una volta il diritto d'autore era una pratica relativamente priva di effetti negativi, che non suscitava discussioni, e non costituiva una limitazione per il pubblico. Oggi ciò non è più vero. Se possedete un computer, l'interesse primario degli editori è quello di imporvi delle restrizioni. Il diritto d'autore una volta era facile da far rispettare perché era una restrizione solo per gli editori, ed era facile trovarli per esaminare quanto pubblicavano. Ora il diritto d'autore è una restrizione su tutti e ciascuno di voi. Per imporne il rispetto occorre sorveglianza, intrusioni e pene severe, e stiamo osservando l'introduzione di queste misure nelle leggi degli Stati Uniti e di altri paesi.

Il diritto d'autore era effettivamente uno scambio vantaggioso per il pubblico, perché quest'ultimo dava in cambio delle libertà che di fatto non poteva esercitare. Ma oggi il pubblico può esercitare tali libertà. Cosa fate se avete un sottoprodotto che una volta non vi serviva, eravate abituati a scambiarlo e improvvisamente ne scoprite un uso? Potete consumarlo o utilizzarlo. Cosa farete in pratica? Non lo scambiate affatto, ne tenete almeno una parte. E naturalmente questo è quanto vorrebbe fare la gente. È ciò che il pubblico fa ogniqualvolta gli viene data voce per esprimere la propria preferenza, conserva una parte della propria libertà e la esercita. Napster, in cui il pubblico decide di esercitare la libertà di copia invece di rinunciarvi, è un grande esempio di questo principio. La

cosa naturale da farsi per rendere la normativa sul diritto d'autore adatta alla situazione odierna è ridurre l'ammontare di potere nelle mani dei detentori del copyright, ridurre la quantità di restrizioni che essi impongono al pubblico, e aumentare la libertà conservata dalla gente.

Ma ciò non piace agli editori, i quali vogliono esattamente l'opposto. Gli editori intendono aumentare i poteri del diritto d'autore fino al punto in cui possano controllare rigidamente ogni utilizzo delle informazioni. Ciò ha portato a legislazioni che concedono un aumento di potere senza precedenti per i detentori del copyright. Vengono così sottratte quelle libertà che il pubblico era solito mantenere nell'era della carta stampata.

Consideriamo ad esempio gli e-book, i libri elettronici. Oggi vanno tremendamente di moda, è difficile evitarli. Mentre ero in Brasile ho preso un aereo e nella rivista a bordo c'era un articolo in cui si prevedeva che entro 10 o 20 anni saremmo tutti passati agli e-book. Chiaramente, questo tipo di campagna pubblicitaria è pagata da qualcuno. Perché? Credo di saperlo. La ragione è che gli e-book costituiscono l'opportunità per togliere alcune delle libertà che i lettori della carta stampata hanno sempre avuto e continuano ad avere – la libertà, per esempio, di prestare un libro ad un amico, o di prenderlo a prestito da una biblioteca pubblica, o di venderne una copia ad un negozio di libri usati, o di comprarne una copia in modo anonimo e senza dover inserire i dati dell'acquirente in un apposito database. E forse persino il diritto di leggerlo due volte.

Sono queste le libertà che gli editori vorrebbero eliminare, ma non possono farlo per i libri stampati perché sarebbe una presa di potere troppo ovvia e provocherebbe una reazione generalizzata. E così hanno trovato una strategia indiretta. Prima, ottengono una legi-

slazione che elimini questi diritti per gli e-book quando non ci sono e-book; così non si crea alcuna controversia. Non esistono lettori di libri elettronici a difendere le libertà a cui erano abituati. Obiettivo ottenuto con il Digital Millennium Copyright Act del 1998. Successivamente hanno introdotto gli e-book, convincendo tutti a passare gradualmente dai libri stampati a quelli elettronici, e il risultato finale è che i lettori hanno perso quelle libertà senza che ci sia stato un preciso momento in cui quelle libertà siano state sottratte e ci si potesse battere per conservarle.

Allo stesso tempo possiamo osservare analoghi tentativi per privare la gente delle proprie libertà nell'uso di altri tipi di opere pubblicate. Ad esempio, i film su DVD vengono cifrati in un formato considerato segreto – era stato progettato per essere segreto – e l'unico modo per farsi dire dalle società cinematografiche il metodo di cifratura, in modo da poter costruire dei lettori DVD, era firmare un contratto che obbligava a implementare determinate restrizioni negli apparecchi, con il risultato di impedire al pubblico perfino l'esercizio dei propri diritti legali. A un certo punto alcuni programmatori europei scoprirono il formato dei DVD e scrissero un programma libero per leggerli.² Ciò rese possibile utilizzare software libero e GNU/Linux per guardare un DVD regolarmente acquistato, il che è una cosa assolutamente legittima. È giusto poterlo fare con software libero.

Ma le società cinematografiche non erano d'accordo, e portarono la questione in tribunale. Sapete, l'industria cinematografica una volta produceva un sacco di film con scienziati pazzi e qualcuno diceva, "Ma, dottore, ci sono alcune cose che l'Uomo non dovrebbe conoscere". Probabilmente queste società hanno visto troppi dei loro film, perché sono giunte alla conclusione che il formato

² Oggi esistono diversi pacchetti analoghi; il primo si chiamava "DeCSS".

dei DVD fosse qualcosa che l'Uomo non doveva conoscere, e sono riuscite ad ottenere una sentenza di censura totale sul software usato per leggere i DVD. È stato proibito persino l'inserimento di un link ad un sito fuori dagli Stati Uniti dove è legale diffondere queste informazioni. La sentenza è già stata portata in appello. In tale occasione ho presentato un documento di sostegno, e ne sono orgoglioso, anche se in effetti sto giocando un ruolo minimo in questa specifica battaglia.

Il governo degli Stati Uniti è intervenuto direttamente a sostegno della parte avversa. Ciò non deve sorprendere se consideriamo in primo luogo il motivo per cui è stato approvato il Digital Millennium Copyright Act. La ragione risiede nel sistema di finanziamento delle campagne elettorali che abbiamo negli Stati Uniti, cioè essenzialmente una corruzione legalizzata in cui i candidati vengono comprati dalle varie aziende prima ancora di essere eletti. E, ovviamente, conoscono bene i loro padroni – sanno per chi stanno lavorando, e approvano quelle leggi che danno maggior potere a tali aziende.

Non sappiamo come andrà a finire questa battaglia. Nel frattempo l'Australia ha approvato una legge simile e anche l'Europa si appresta a farlo; il piano è di non lasciare alcun luogo al mondo dove siano disponibili queste informazioni. Gli Stati Uniti rimangono comunque i primi nel tentativo di impedire al pubblico la distribuzione di informazioni già pubblicate.

Gli Stati Uniti non sono tuttavia il primo paese per cui ciò rappresenta una priorità: era molto importante anche per l'Unione Sovietica. Dove l'attività di fare copie non autorizzate e ridistribuirle era nota come Samizdat, e per debellare il fenomeno misero a punto una serie di metodi. Primo, guardie vicino ad ogni dispositivo di copia, per controllare cosa venisse copiato e preve-

nire copie vietate. Secondo, dure punizioni per chiunque fosse sorpreso in attività di copia illecita: si poteva essere spediti in Siberia. Terzo, incoraggiare la delazione, chiedendo a tutti di spiare vicini e colleghi e riferire alla polizia dell'informazione. Quarto, responsabilità collettiva: "Tu! Tu sei responsabile per quel gruppo di persone! Se becco uno qualsiasi di loro a fare copie illegali, in prigione ci vai tu. Quindi, è meglio se li controlli per bene". E quinto, la propaganda, a partire dall'infanzia, per convincere tutti che solo un acerrimo nemico del popolo rischierebbe di fare copie illecite.

Oggi gli Stati Uniti stanno utilizzando tutti questi metodi. Primo, agenti a guardia dei dispositivi di copia. Nelle copisterie ci sono guardie umane per controllare cosa si copia. Ma costerebbe troppo ingaggiare degli esseri umani per fare lo stesso con gli utenti di computer, il lavoro umano è troppo caro. Così hanno messo a guardia dei robot. Questo è lo scopo del Digital Millennium Copyright Act. L'unico modo per accedere a certi dati è inserire nel computer un particolare software, che vi impedisce di copiare quegli stessi dati.

Oggi esiste un progetto per l'introduzione di tale software in ogni hard disk, e in questo modo potreste trovarvi dei file a cui non potete accedere, a meno di ottenere il permesso da qualche server di rete. E il tentativo di aggirare quel software, o persino di dire ad altri come aggirarlo, costituisce un crimine.

Secondo, punizioni dure. Alcuni anni fa, fare copie di qualcosa e passarle a un amico giusto per aiutarlo, non costituiva reato; non lo era mai stato negli Stati Uniti. Poi venne trasformato in un crimine, e si può essere incarcerati per anni solo per aver condiviso qualcosa con il vicino.

Terzo, informatori. Avrete visto quelle pubblicità in TV e nella

metropolitana di Boston che incitavano a denunciare i colleghi alla polizia dell'informazione, ufficialmente nota come Software Publishers Association.

Quarto, responsabilità collettiva. Negli Stati Uniti ciò è stato fatto cooptando i provider d'accesso a internet, rendendoli legalmente responsabili per tutto ciò che distribuiscono i loro utenti. L'unico modo che hanno per non essere considerati comunque responsabili è l'applicazione di una procedura non modificabile per disconnettere l'utente o rimuovere l'informazione inviata entro due settimane da un reclamo. Giusto qualche giorno fa ho sentito che un sito di protesta contro alcune brutte pratiche di Citybank è stato cancellato in questo modo. Oggigiorno, non si fa nemmeno in tempo ad arrivare in tribunale: il vostro sito viene semplicemente fatto sparire.

E infine, la propaganda a cominciare dall'infanzia. È per questo che si ricorre al termine "pirata". Se ci pensate, qualche anno fa tale termine veniva utilizzato per definire quegli editori che non pagavano gli autori. Ma ora il significato è stato completamente stravolto. Oggi viene applicato a quei membri del pubblico che sfuggono al controllo degli editori. Viene usato per convincere la gente che solo un nemico del popolo farebbe delle copie illegali. Il termine trasmette il messaggio che "condividere qualcosa con il vicino è moralmente equivalente ad attaccare una nave". Spero che non siate d'accordo con questa caratterizzazione, e se non lo siete, spero che vi rifiuterete di utilizzare il termine in tal senso.

Gli editori stanno comprando delle leggi onde dotarsi di maggiori poteri. Stanno inoltre estendendo la durata del diritto d'autore. La Costituzione degli Stati Uniti dice che il copyright è valido per un periodo di tempo limitato, ma gli editori vogliono farlo durare per sempre. Ma siccome ottenere una modifica costituzionale

risulterebbe piuttosto difficile, hanno trovato un modo più semplice per raggiungere lo stesso risultato. Ogni 20 anni si estende in maniera retroattiva il diritto d'autore di 20 anni. Così il risultato è che in ogni momento il diritto d'autore dura nominalmente per un periodo determinato e un giorno qualsiasi copyright è destinato a terminare. Ma quel termine non verrà mai raggiunto perché ogni 20 anni quel copyright verrà esteso di 20 anni; così nessuna opera potrà mai tornare di pubblico dominio. Questa pratica è stata chiamata "perpetual copyright on the installment plan", copyright perpetuo nel progetto rateale.

La legge del 1998 che estende il diritto d'autore per ulteriori 20 anni è nota come il "Mickey Mouse Copyright Extension Act"³ perché uno dei suoi principali sponsor fu la Disney. Questa si rese conto che il diritto d'autore su Topolino stava per scadere, qualcosa che andava assolutamente evitato perché da quel copyright guadagnava molto denaro.

Globalizzazione

In effetti il titolo di questo intervento doveva essere "Diritto d'autore e Globalizzazione". Per quanto riguarda la globalizzazione, questa viene portata avanti tramite una serie di politiche implementate in nome dell'efficienza economica ovvero i cosiddetti accordi per il libero commercio, che in realtà sono pensati per dare potere alle imprese a scapito delle leggi e della politica. Non hanno nulla a che fare con il libero scambio, ma piuttosto con un trasferimento di potere: togliere il potere decisionale e legislativo ai cittadini di qualsiasi paese possa plausibilmente tenere in considerazione i propri interessi, e assegnarlo alle imprese, che non ter-

³ Il titolo ufficiale è "The Sonny Bono Copyright Term Extension Act".

ranno nella minima considerazione l'interesse di quei cittadini. Dal loro punto di vista, è la democrazia ad essere un problema e questi accordi vengono stipulati per porvi fine. Ad esempio, il NAFTA⁴ contiene dei provvedimenti che, mi sembra, consentono alle aziende di querelare un altro governo per sbarazzarsi di qualche legge che ritengano stia interferendo con i loro profitti in quel paese. Così le società straniere possono avere maggior potere dei cittadini di una nazione.

Sono in atto tentativi per ampliare queste posizioni oltre il NAFTA. È ad esempio questo uno degli obiettivi della cosiddetta "area del libero scambio delle Americhe", che mira ad estendere questo principio a tutti i paesi sudamericani e caraibici, mentre l'accordo multilaterale sugli investimenti doveva estenderlo al mondo intero.

Una cosa evidenziatasi negli anni '90 è che tali accordi hanno iniziato ad imporre il copyright in tutto il mondo, in maniera sempre più forte e restrittiva. Questi trattati non sono accordi per il libero scambio. Sono in realtà accordi commerciali utilizzati per fornire alle aziende il controllo sul commercio in tutto il mondo, in modo da eliminare il libero scambio.

Quando nel 1800 gli Stati Uniti erano un paese in via di sviluppo, lo stato non riconosceva i diritti d'autore esteri. Questa fu una decisione presa con attenzione, e si rivelò intelligente. Si convenne che, per gli Stati Uniti, il riconoscimento di quei diritti sarebbe stato semplicemente svantaggioso e avrebbe succhiato denaro senza dimostrarsi particolarmente utile.

Oggi si dovrebbe applicare la stessa logica ai paesi in via di sviluppo, ma gli Stati Uniti hanno sufficiente potere per costringerli ad andare contro i propri interessi. In realtà è un errore parlare

⁴ *North American Free Trade Agreement.*

degli interessi delle nazioni in questo contesto. Infatti, sono sicuro che la maggior parte di voi sappia quanto sia sbagliato tentare di giudicare l'interesse pubblico facendo la somma della ricchezza individuale. Se i lavoratori americani perdessero un miliardo di dollari e Bill Gates ne guadagnasse due, sarebbero forse gli americani in generale più ricchi? Potrebbe dirsi un bene per l'America? Se si considera solo la cifra totale, sembra lo sia. Tuttavia, l'esempio mostra in concreto che guardare al totale è il modo sbagliato di giudicare, poiché Bill Gates non ha alcun bisogno di altri due miliardi di dollari, ma la perdita di un miliardo di dollari da parte di chi non ha altrettanto denaro da cui partire può essere dolorosa. Dunque, discutendo su uno di questi trattati commerciali, quando si sente qualcuno parlare degli interessi di questa o di quella nazione, non si fa altro che sommare le entrate di tutti i cittadini. I ricchi vengono sommati ai poveri. È solo una scusa per mettere in atto lo stesso inganno per farci ignorare l'effetto sulla distribuzione delle ricchezze all'interno del paese e il fatto che quegli accordi la renderanno ancora più disomogenea, come è accaduto negli Stati Uniti.

Non è quindi l'interesse degli Stati Uniti a giovare dell'inasprimento delle norme sul diritto d'autore in tutto il mondo. È solo quello di alcuni imprenditori, molti dei quali vivono in quel paese, mentre altri risiedono altrove. Ciò non favorisce in alcun modo l'interesse pubblico.

Ripensare il copyright

Ma cosa ha senso fare? Se crediamo nello scopo del diritto d'autore incluso, ad esempio, nella Costituzione degli Stati Uniti, cioè quello di promuovere il progresso, quali linee politiche intelligenti andrebbero seguite nell'era delle reti informatiche? Chiaramente,

anziché aumentare i poteri del copyright dovremmo ridimensionarli in modo da offrire al pubblico un certo grado di libertà, con cui trarre vantaggio dai benefici della tecnologia digitale e delle reti informatiche. Ma fino a che punto si deve arrivare? È una domanda interessante, poiché non credo sia necessario abolire totalmente il diritto d'autore. L'idea di rinunciare ad alcune libertà in cambio di un maggior progresso risulterebbe vantaggiosa ad un certo livello, anche se il copyright tradizionale rinuncia a troppa libertà. Ma per ragionare in maniera intelligente sulla questione, la prima cosa da riconoscere è che non esiste alcun motivo per rendere tutto uniforme. Non c'è alcuna ragione per insistere nell'avere un unico accordo per ogni tipo di opere.

Anzi, questo è un caso già superato poiché esistono numerose eccezioni per la musica. La legislazione sul diritto d'autore considera le opere musicali in maniera molto diversa tra loro. Ma i produttori ricorrono astutamente a un'arbitraria insistenza sull'uniformità. Prendono in esame alcuni casi particolari e sostengono degli argomenti secondo cui, in quei casi particolari, sarebbe vantaggioso avere tutto questo diritto d'autore. Successivamente affermano che, per uniformità, le restrizioni applicate nei casi particolari devono essere estese al tutto. Così, chiaramente, scelgono un caso particolare in cui possono produrre gli argomenti più forti, anche se tale caso è piuttosto raro e non così importante nel contesto generale.

Ma forse, per quel caso specifico, è giusto che esista tutto quel copyright. Non dobbiamo pagare lo stesso prezzo per ogni cosa che compriamo. Spendere un migliaio di dollari per una macchina nuova sarebbe un ottimo affare, ma spenderne altrettanti per una confezione di latte sarebbe, al contrario, un pessimo affare. Di certo, in altre situazioni della vita, non si pagherebbe un prez-

zo speciale per tutto ciò che si compra. Perché farlo in questo caso? Dobbiamo considerare le varie opere in maniera diversa tra loro, e vorrei proporvi un possibile modo per farlo.

La prima serie di opere riguarda quelle funzionali – ovvero, opere il cui utilizzo consente di portare a termine un qualche compito.

Questa categoria include ricette, programmi informatici, manuali e libri di testo, opere di consultazione come dizionari ed enciclopedie. Credo che per tutte queste opere funzionali la questione sia essenzialmente identica a quella del software e possano essere applicate le stesse conclusioni. Si dovrebbe avere la libertà di pubblicarne anche una versione modificata, poiché è molto utile modificare una di tali opere. Le necessità della gente non sono le stesse per tutti. Se io scrivo un'opera affinché faccia qualcosa che credo debba essere fatto, qualcun altro potrebbe avere un'idea diversa al riguardo. E costui potrebbe voler modificare quell'opera per fargli fare ciò che è meglio per lui. A questo punto, altre persone potrebbero avere le stesse esigenze di chi ha modificato l'originale, e la versione modificata potrebbe andar bene anche per loro. Chiunque cucini lo sa, e lo sa da centinaia di anni. È del tutto normale fare copie di ricette e distribuirle ad altri, ed è altrettanto normale modificare una ricetta. Se si cambia una ricetta e la si prepara per gli amici e a loro piace, potrebbero chiedere "Posso avere la ricetta?". Allora si scriverà la propria versione della ricetta e se ne daranno copie agli amici. Questa è esattamente la stessa cosa che, molto tempo dopo, abbiamo iniziato a fare nella comunità del software libero.

Ecco dunque una prima categoria di opere.

La seconda, tratta delle opere il cui obiettivo è diffondere il pensiero di determinate persone. Il loro intento è parlare di tali persone. Ciò include, ad esempio, memorie, saggi d'opinione, arti-

coli scientifici, offerte di compravendita, cataloghi di prodotti in vendita. Il punto centrale di queste opere è che esprimono ciò che qualcuno pensa, ha visto o crede. Modificarle significa mistificare quel che intende dire l'autore; modificare queste opere non è attività socialmente utile. Perciò la copia letterale è l'unica cosa che si può veramente essere autorizzati a fare.

La domanda successiva è: si dovrebbe avere il diritto di commerciare con tali copie letterali? Oppure è sufficiente la copia letterale senza fini di lucro? Come vediamo, qui si tratta di due attività diverse tra loro, per cui affronteremo le due questioni in maniera separata – il diritto di fare copie letterali a scopo non commerciale e quello di farle a scopo di lucro. Dunque, un buon compromesso potrebbe essere quello di avere copie commerciali coperte dal diritto d'autore accordando però a tutti il diritto di farne copie letterali non commerciali. In tal modo, il copyright sulle copie commerciali, così come sulle versioni modificate – solo l'autore può approvare una versione modificata – continuerà a produrre lo stesso flusso di entrate fornito ora per sovvenzionare la scrittura di tali opere, in qualsiasi misura lo faccia.

Consentire la copia letterale non commerciale vuol dire che il diritto d'autore non deve invadere più la casa di nessuno. Diventa nuovamente una regolamentazione industriale, facile da rinforzare e indolore, non richiedendo più punizioni draconiane e informatori per imporla. Così possiamo ottenere il massimo del beneficio del sistema attuale, evitandone buona parte dell'orrore.

La terza categoria riguarda le opere artistiche o di intrattenimento, in cui la cosa più importante è la sensazione che si prova nel guardarle. In questo caso il problema della modifica è molto complesso poiché, se da un lato c'è l'idea che queste opere riflettono la visione di un artista, modificandole se ne mistifica tale visione.

Dall'altro lato, si ha il fatto che spesso esiste un processo di rielaborazione popolare, in base al quale attraverso le modifiche apportate da una serie di persone, si producono a volte dei risultati di alto livello. Spesso anche per gli stessi artisti è utile attingere da opere precedenti. Alcuni dei lavori di Shakespeare si basano su trame prese da opere altrui. Se fossero state in vigore le attuali leggi sul diritto d'autore, queste opere sarebbero state illegali. La questione del cosa fare circa la pubblicazione di versioni modificate di un'opera artistica o estetica è complessa, e per risolvere il problema forse dovremmo introdurre ulteriori suddivisioni della categoria. Ad esempio, il settore dei computer game potrebbe essere trattato in maniera particolare, dove ognuno sarebbe libero di pubblicarne versioni modificate. Ma forse un romanzo dovrebbe essere trattato diversamente; in questo caso, forse la pubblicazione a scopo commerciale dovrebbe richiedere un accordo con l'autore originale.

Ora, se la pubblicazione commerciale di queste opere artistiche è coperta da copyright, ciò fornirà la maggior parte delle entrate oggi disponibili per sostenere gli autori e i musicisti, nella misura limitata in cui il sistema attuale li sostiene, visto che [tale sistema] svolge una pessima funzione. Potrebbe perciò trattarsi di un compromesso ragionevole, come nel caso precedente delle opere che rappresentano determinate persone.

Se si guarda al futuro, al tempo in cui l'era delle reti informatiche sarà davvero iniziata, quando avremo superato questo stadio transitorio, si può immaginare per gli autori un altro metodo per far soldi dalle proprie opere. Si pensi ad un sistema di pagamento digitale che consenta di ottenere denaro per il proprio lavoro. Possiamo immaginare un sistema di pagamento digitale che permetta di inviare a qualcuno denaro attraverso Internet; ciò può essere rea-

lizzato in vari modi, utilizzando la crittografia, ad esempio. Si immagini che sia consentita la copia letterale di queste opere artistiche, ma scritte in modo che, quando le si ascolti o le si legga o le si guardi, in un angolino dello schermo appaia una finestrella che dice “Premere questo pulsante per spedire un dollaro all’autore”, o al musicista o a chiunque sia. Questa finestra compare e basta, non è invadente, sta nel suo angolino. Non dà fastidio, ma è lì, a ricordarvi che è una buona cosa sostenere scrittori e musicisti.

Perciò, se l’opera che si sta leggendo o ascoltando piace, si può pensare “Perché non dare un dollaro a questa gente? È solo un dollaro. Cosa vuoi che sia? Non mi mancherà di certo.” E così le persone cominceranno a spedire un dollaro. La cosa più bella di tutto ciò è che il meccanismo renderebbe la copia un alleato di autori e musicisti. Quando qualcuno ne invia a un amico una copia per email, costui potrebbe spedire un dollaro. Se l’opera piace veramente, si potrebbe mandare un dollaro più di una volta e questo dollaro sarebbe più di quanto gli autori guadagnano oggi quando si acquista un loro libro o CD, poiché ora ricevono solo una piccola parte del guadagno. Gli stessi editori che chiedono il controllo totale sul pubblico in nome di autori e musicisti, li stanno fregando da sempre.

Vi raccomando la lettura dell’articolo di Courtney Love sulla rivista online Salon, un articolo sui pirati che usano il lavoro dei musicisti senza pagarli. Questi pirati sono le case discografiche che pagano ai musicisti il 4% dell’ammontare delle vendite, in media. Ovviamente, i musicisti di successo hanno più potere: prendono più del 4% delle loro ampie vendite, il che vuol dire che la maggior parte dei musicisti con un contratto di produzione riceve meno del 4% delle loro vendite limitate.

Ecco come funziona: la società discografica investe in pubblicità e considera questa spesa un anticipo ai musicisti, anche se questi ultimi non vedranno mai quei soldi. Perciò, in teoria, quando si acquista un CD, parte di quel denaro dovrebbe andare ai musicisti, ma in pratica non sarà così. In realtà andrà a rimborsare le spese pubblicitarie e i musicisti vedranno parte di quel denaro solo se otterranno molto successo.

I musicisti, naturalmente, firmano i contratti perché sperano di essere tra i pochi fortunati ad avere un grande successo. Sostanzialmente vengono lusingati con l'offerta di una lotteria. Anche se sono bravi, possono non essere così bravi e così sottili nel ragionamento da accorgersi della trappola. Perciò firmano e probabilmente quello che ottengono è solo la pubblicità. Allora, perché non facciamo loro pubblicità in modo diverso, non attraverso un sistema da complesso industriale che limita il pubblico e che ci riempie di brutta musica facile da vendere? Invece, perché non far sì che il naturale impulso dell'ascoltatore a condividere la musica preferita diventi alleato dei musicisti? Se utilizziamo il riquadro che appare sul monitor per inviare un dollaro ai musicisti, allora le reti informatiche potrebbero essere il sistema per far loro pubblicità, la stessa pubblicità ora ottenuta tramite i contratti.

Dobbiamo riconoscere che l'attuale sistema del diritto d'autore rende un cattivo servizio nel sostenere i musicisti, così come il commercio mondiale produce un cattivo servizio nell'elevare gli standard di vita nelle Filippine e in Cina. Esistono "aree imprenditoriali" estere in cui la gente lavora per aziende che li sfrutta e tutti i prodotti sono fabbricati in aziende che sfruttano i dipendenti. La globalizzazione è un sistema assai poco efficace per elevare gli standard di vita all'estero. Diciamo che per produrre qualcosa uno statunitense viene pagato 20 dollari l'ora; per lo stesso

lavoro, un messicano riceve forse 6 dollari al giorno. In quest'ultimo caso succede che a un lavoratore americano viene tolta una notevole quantità di denaro, e una piccola frazione di questa, ovvero una minima percentuale, viene data a un messicano mentre il resto va all'azienda. Perciò, se l'obiettivo è elevare gli standard di vita dei lavoratori messicani, ecco un modo del tutto inefficace di farlo.

È interessante notare come un identico fenomeno si riscontri nell'industria del copyright, la stessa idea generale. Nel nome di quei lavoratori che certamente meritano qualcosa, si propongono misure che riservano loro una minima parte sostenendo invece prioritariamente il potere delle grandi società che controllano le nostre vite.

Per sostituire un sistema valido, bisogna lavorare molto seriamente onde proporre un'alternativa migliore. Sapendo che il sistema attuale è inefficace, non è così difficile trovare un'alternativa migliore: lo standard con cui ci si confronta è oggi molto basso. Dovremmo sempre ricordare tutto ciò, quando prendiamo in considerazione le problematiche sulla politica del diritto d'autore. Penso di aver detto la maggior parte di quanto volevo dire. Vorrei ricordarvi che domani in Canada è il Phone-In Sick Day.⁵

Domani inizierà un summit mirato alla firma dell'accordo sull'area di libero scambio per le Americhe, con l'intento di estendere il potere delle grandi società in altri paesi. In Quebec sono previste grosse iniziative di protesta. Per bloccarle sono stati impiegati metodi estremi. A molti americani viene impedito l'ingresso in Canada attraverso quel medesimo confine che è loro permesso oltrepassare in ogni altro momento. Con la più debole delle scu-

⁵ 20 aprile 2001; iniziativa nazionale di protesta, in cui tutti i dipendenti prendono un giorno di malattia dal lavoro.

se, intorno al centro di Quebec è stata costruita una fortezza onde impedire l'accesso ai dimostranti. Abbiamo visto utilizzare molti sporchi trucchi contro la manifestazione pubblica di protesta rispetto a questi trattati. Qualsiasi democrazia ci sia rimasta, dopo aver tolto i poteri di governo ai governatori democraticamente eletti per darli ad aziende e organismi internazionali non eletti, qualsiasi cosa rimanga dopo tutto questo, potrebbe non sopravvivere alla repressione delle proteste pubbliche.

Ho dedicato 17 anni della mia vita a lavorare per il software libero e le annesse questioni. Non l'ho fatto perché penso che questo sia il problema politico più importante al mondo. L'ho fatto perché ho ritenuto che fosse il settore in cui dovevo usare la mia competenza per fare del bene. Ma è successo che gli aspetti generali della politica sono mutati e il maggior problema politico attuale è contrastare la tendenza a dare potere all'imprenditoria nei confronti del pubblico e dei governi. Io considero il software libero e i problemi correlati agli altri aspetti dell'informazione che ho discusso oggi, come parte del problema principale. Perciò mi sono trovato a lavorare indirettamente su tale questione. Spero di poter dare il mio contributo a questa causa.

Sessione di domande e risposte

David Thorburn: Tra poco il pubblico potrà fare domande e commenti. Ma permettetemi di replicare brevemente. Mi sembra che il consiglio pratico più incisivo ed importante che Stallman ci dà abbia due elementi chiave. Uno è riconoscere che i vecchi presupposti ed usi del diritto d'autore sono inappropriati: vengono messi in discussione e delegittimati dall'avvento del computer e delle reti informatiche. Può sembrare ovvio, ma è fondamentale. L'altro è riconoscere che l'era digitale esige che si riconsideri il

modo in cui distinguiamo e valutiamo le diverse tipologie di lavoro intellettuale e creativo. Stallman ha sicuramente ragione sul fatto che certi tipi di imprese intellettuali abbiano bisogno più di altre di essere protette dal copyright. Cercare di individuare sistematicamente questi diversi tipi o livelli di protezione del diritto d'autore mi sembra un modo efficace di affrontare i problemi relativi al lavoro intellettuale posti dall'avvento del computer.

Ma penso di intravedere un altro tema che sta dietro a quanto ha affermato Stallman e che non riguarda direttamente i computer, ma più ampiamente le istituzioni democratiche e il potere che il governo e le aziende esercitano in misura sempre maggiore sulla nostra vita. Questo aspetto populista e anti-monopolista del discorso di Stallman è interessante ma anche riduttivo e potenzialmente semplicistico. Ed è forse anche eccessivamente idealista. Per esempio, come potrebbe sopravvivere un romanziere o un poeta o un autore di canzoni o un musicista o un autore di testi accademici in un mondo meraviglioso in cui le persone siano incoraggiate a pagare gli autori, ma non siano obbligate a farlo? In altre parole, mi sembra che la differenza tra la pratica attuale e le possibilità visionarie su cui specula Stallman sia ancora enorme.

Concludo chiedendo a Stallman di approfondire maggiormente alcuni aspetti del suo intervento, e in particolare di ampliare i concetti sul modo in cui potrebbero essere tutelati dal suo sistema di copyright quelli che noi chiamiamo “creatori tradizionali”.

Richard M. Stallman: Prima di tutto, devo far notare che non dovremmo utilizzare il termine “protezione” per descrivere le incombenze del diritto d'autore. Il diritto d'autore limita le persone. “Protezione” è un termine di propaganda per le aziende che detengono il copyright. Tale termine significa impedire la distru-

zione di qualcosa. Be', non credo che una canzone venga distrutta se ne esistono più copie suonate più volte. Non penso nemmeno che un romanzo si distrugga se più persone ne leggono una copia. Perciò non userei quel termine. Penso che porti ad identificarsi con la parte sbagliata.

Inoltre, pensare in termini di proprietà intellettuale è una cattiva idea per due motivi: in primo luogo perché pregiudica la domanda cruciale in questo campo, che è: in che modo dovrebbero essere trattate queste cose? Dovrebbero essere trattate o meno come qualche tipo di proprietà? Utilizzare il termine "proprietà intellettuale" per descrivere quest'ambito significa presupporre che la risposta alla seconda domanda sia "sì", che è questo, e non un altro, il modo in cui la questione va considerata.

In secondo luogo, incoraggia una iper-generalizzazione. La proprietà intellettuale è un concetto onnicomprensivo per parecchi sistemi legali diversi tra loro con origini indipendenti, come per esempio diritti d'autore, brevetti, marchi registrati, segreti industriali e altro. Sono quasi completamente diversi tra loro, non hanno nulla in comune. Ma l'uso del termine "proprietà intellettuale" porta la gente a pensare erroneamente che esista un principio generale di proprietà intellettuale da applicare a settori specifici, presupponendo così che questi diversi campi di applicabilità della legge siano simili. Ciò porta non solo a pensare in modo confuso su quel che sia giusto fare, ma conduce anche alla mancata comprensione di cosa dica realmente la legge, perché si suppone che la legge sul copyright e la legge sui brevetti e la legge sui marchi registrati siano simili, mentre invece, di fatto, sono completamente diverse.

Se si vuole incoraggiare un'attenta riflessione e una corretta comprensione di quanto dice la legge, evitiamo il termine "proprietà

intellettuale”. Si parli di diritto d’autore, o di brevetti, o di marchi registrati o di qualsiasi altro argomento si voglia. Ma non si parli di proprietà intellettuale. Sarebbe assurdo avere un’opinione sulla proprietà intellettuale. Non ho un’opinione sulla proprietà intellettuale, ma ho opinioni sul diritto d’autore, sui brevetti e sui marchi registrati, e sono diverse. Ci sono arrivato attraverso processi di pensiero diversi perché questi sistemi legali sono completamente diversi tra loro.

Ho fatto una digressione, ma era estremamente importante.

Adesso arrivo al dunque. Naturalmente ora non possiamo sapere se funzionerà bene, se funzionerà chiedere agli utenti di pagare volontariamente autori e musicisti preferiti. Una cosa ovvia è che tale sistema funzionerà proporzionalmente al numero di persone che utilizzeranno la rete e quel numero, si sa, aumenterà di vari ordini di grandezza nei prossimi anni. Se lo provassimo oggi, potrebbe fallire, ma questo non proverebbe nulla, perché potrebbe funzionare se il numero delle persone paganti fosse dieci volte maggiore.

L’altra è che non abbiamo ancora a disposizione un tale sistema digitale di pagamento in contanti, quindi oggi non siamo in grado di metterlo alla prova. Possiamo provare a fare qualcosa di simile. Esistono servizi tramite i quali è possibile pagare qualcuno, cose come Pay Pal. Ma prima di poterlo fare, si devono affrontare un mucchio di formalità e fornire i propri dati personali. E vengono effettuate registrazioni sui destinatari dei pagamenti. Ci si può fidare che non se ne abusi?

Non è il dollaro da pagare che potrebbe scoraggiare, ma i problemi connessi alle modalità di pagamento. L’idea generale è che quando si vuole pagare qualcuno, dovrebbe essere facilissimo e non dovrebbe esserci nulla che lo sconsigli se non la somma stes-

sa di denaro. E se la somma è abbastanza piccola, perché dovrebbe scoraggiare? Sappiamo comunque che i fan amano davvero i musicisti e sappiamo che alcuni gruppi musicali che avevano ed hanno un certo successo, come i Grateful Dead, hanno incoraggiato i propri fan a copiare e ridistribuirne la musica. Non hanno avuto problemi a guadagnarsi da vivere con la loro musica per aver incoraggiato i fan a registrarla e a copiare le cassette. Ciò non ha neppure provocato riduzioni nella vendita di dischi.

Stiamo gradualmente passando dall'epoca della stampa all'era delle reti informatiche, ma ciò non può accadere in un giorno. La gente continua ad acquistare molti dischi, e probabilmente continuerà a farlo per molti anni ancora, forse per sempre. Finché si andrà avanti in questo modo, continuare semplicemente ad applicare i diritti d'autore alla vendita commerciale di dischi dovrebbe sostenere i musicisti quasi altrettanto bene di oggi. Naturalmente, il sistema non è del tutto soddisfacente, ma almeno non sarà peggiore.

Domanda: [Un commento e una domanda riguardo la libertà di download e il tentativo di Stephen King di vendere uno dei suoi racconti a puntate sul web.]

Stallman: Sì, è interessante sapere quello che ha fatto e cosa è accaduto. Quando ne sentii parlare la prima volta ero euforico. Pensavo, forse sta per fare un passo verso un mondo non basato sulla volontà di tenere il pubblico in pugno. Poi ho visto che in realtà scriveva per chiedere al pubblico di pagare. Per spiegare cosa ha fatto, stava pubblicando un racconto a puntate, a rate, e diceva: "se otterrò abbastanza denaro, ne scriverò ancora". Una richiesta che ben difficilmente poteva considerarsi tale. Era una minaccia

contro il lettore. Diceva: “se non pagate siete cattivi, e se ci sono troppi fra voi che si comportano male, semplicemente smetterò di scrivere”.

Be', chiaramente questo non è il modo di far sentire il pubblico invogliato a mandarti dei soldi. Devi far sì che ti amino, non che ti temano.

Stessa persona del pubblico: I dettagli sono che chiese ad una certa percentuale di persone – non so esattamente, probabilmente intorno al 90% – d’inviare una certa quantità di denaro, che, credo, fosse di un dollaro o due o qualcosa di quest’ordine di grandezza. Per scaricare il racconto bisognava fornire il proprio nome, l’indirizzo e-mail e alcune altre informazioni, e se la percentuale non fosse stata raggiunta dopo il primo capitolo, King disse che non avrebbe diffuso il capitolo successivo. Era molto antagonistico nei confronti di quanti scaricavano il testo.

Domanda: Lo schema in cui non esiste diritto d’autore ma alle persone è richiesto di fare donazioni volontarie, non è aperto all’abuso da parte dei plagiaristi?

Stallman: No. Non è quello che ho proposto. Ricordate, sto proponendo l’esistenza di un diritto d’autore che copra la distribuzione commerciale e permetta solo la redistribuzione letterale non commerciale. Così chiunque abbia modificato un’opera per mettere un puntatore al proprio sito invece che a quello dell’autore reale, continuerebbe a violare il diritto d’autore e potrebbe essere citato in giudizio esattamente come avviene oggi.

Domanda: Capisco. Quindi immagini comunque un mondo in cui esista sempre il copyright?

Stallman: Sì. Come ho detto, per questo tipo di opere. Non sosten-

go che dovrebbe essere permesso tutto. Sto proponendo di diminuire i poteri del diritto d'autore, non di abolirli.

Thorburn: Una domanda che mi è venuta in mente mentre stavi parlando, Richard, e di nuovo adesso che stavi rispondendo a questa domanda, è perché non consideri i modi con cui il computer può eliminare completamente l'intermediario – e che Stephen King si è rifiutato di usare – per stabilire così una relazione personale.

Stallman: Certo, possono farlo, e infatti la donazione volontaria è uno di questi modi.

Thorburn: Pensi davvero che questa modalità non debba passare affatto tramite un editore?

Stallman: Assolutamente no. Spero di no, perché gli editori sfruttano gli autori in maniera terribile. Quando chiedi qualcosa ai rappresentanti degli editori su questo, loro dicono: “Be', sì, se un autore o un gruppo non vogliono passare attraverso di noi, non gli viene legalmente richiesto di farlo”. Ma in realtà operano al meglio per fare in modo che ciò non sia fattibile. Ad esempio, stanno proponendo dei formati multimediali con restrizioni sulla copia, e per poter pubblicare in tali formati devi passare per i grandi editori perché non spiegano a nessuno come farlo. Sperano in un mondo in cui tutti i riproduttori utilizzino questi formati e per poter ottenere qualcosa da riprodurre sarà necessario passare dagli editori. Così, anche senza nessuna legge che impedisca ad un autore o un musicista di pubblicare direttamente un'opera, la cosa non sarebbe fattibile. C'è poi anche il richiamo di una possibile ricchezza. Dicono: “Ti faremo pubblicità e forse diventerai ricco come i Beatles” (o qualsiasi altro gruppo a scelta). Naturalmente

solo a un numero minimo di musicisti potrà accadere quello che è successo a loro. Ma ciò potrebbe indurli a firmare un contratto che li imprigionerebbe per sempre.

Gli editori tendono ad essere molto scorretti nel rispetto dei contratti con gli autori. Per esempio, di solito i contratti dei libri stabiliscono che se un libro è esaurito i diritti tornano all'autore, e gli editori non riescono a convivere bene con questa clausola. Spesso bisogna costringerli a farlo. Be', adesso stanno cominciando ad usare le pubblicazioni elettroniche per dire che non si può esaurire un'edizione; così non dovranno mai restituire i diritti. La loro idea è: quando l'autore non ha voce in capitolo, spingerlo a firmare, e da allora non avrà più potere; il potere rimane solo all'editore.

Domanda: Sarebbe bene avere delle licenze libere per diverse tipi di opere che salvaguardino la libertà di tutti gli utenti di copiarle nella maniera più appropriata per quel tipo di opera?

Stallman: Be', qualcuno sta lavorando. Ma per opere non funzionali, una cosa non sostituisce l'altra. Prendiamo un'opera di tipo funzionale, diciamo un elaboratore di testi. Bene, se qualcuno realizza un elaboratore di testi libero, lo si può usare, non serve l'elaboratore di testi non libero. Ma non direi che una canzone libera possa sostituire tutte le canzoni non libere, o che un racconto libero possa sostituire tutti i racconti non liberi. Per questo tipo di opere le cose sono diverse. Penso perciò che dobbiamo semplicemente riconoscere come queste leggi non meritino di essere rispettate. Non è sbagliato condividere qualcosa con il vicino, e se qualcuno dice che non puoi farlo, non bisogna dargli retta.

Domanda: A proposito delle opere funzionali, come si bilancia l'e-

sigenza di abolire il copyright con l'esigenza di incentivi economici per favorire lo sviluppo di queste opere funzionali?

Stallman: Possiamo notare che, prima di tutto, questi incentivi economici sono molto meno necessari di quanto si fosse supposto. Basta guardare al movimento del software libero in cui abbiamo più di 100.000 volontari a tempo ridotto che sviluppano software libero. Dunque esistono altri modi per raccogliere fondi, non sono basati sull'impedire al pubblico di copiare e modificare queste opere. Questa è l'interessante lezione data dal movimento del software libero. A parte il fatto che offre un modo per utilizzare il computer mantenendo la libertà di condividere e cooperare con altri, ci mostra anche che è semplicemente sbagliato presupporre che la gente non farebbe mai cose simili senza dare loro poteri speciali per costringere gli altri a pagarli. Molta gente è disposta a fare queste cose. Inoltre, considerando ad esempio la stesura di monografie che servono come libri di testo in molti campi scientifici, tranne per quelle piuttosto basilari, ci si accorge che in questo modo gli autori non guadagnano nulla. Abbiamo un progetto di enciclopedia libera che è, di fatto, un progetto commerciale di enciclopedia libera, e sta facendo progressi. Avevamo un progetto per una enciclopedia GNU, ma ci siamo uniti a quello commerciale quando hanno adottato la nostra licenza. In gennaio sono passati alla Licenza per Documentazione Libera GNU per tutti gli articoli di quell'enciclopedia. Così abbiamo detto, "Bene, uniamo le nostre forze e invitiamo la gente a contribuire". Si chiama NUPEDIA, e ne trovate il link all'indirizzo <http://www.gnu.org/encyclopedia>. Così abbiamo esteso lo sviluppo comunitario di una base libera di conoscenze utili dal software all'enciclopedia. Sono piuttosto fiducioso che in tutte queste aree del lavoro funzio-

nale non serva un incentivo economico fino al punto di dover rivedere l'uso di queste opere.

Thorburn: E a proposito delle altre due categorie [le opinioni di un autore e l'intrattenimento]?

Stallman: Per le altre due categorie di opere, non saprei come fare. Non so se un giorno si scriveranno romanzi senza preoccuparsi se ci si faranno dei soldi o meno. In una società del dopo-scarità, credo si preoccuperanno. Forse quello che dobbiamo fare per poter raggiungere una società del dopo-scarità è liberarci dal controllo delle corporation sull'economia e sulle leggi. Così in effetti si tratta del problema dell'uovo e della gallina. Cosa facciamo prima? Come possiamo ottenere un mondo dove le persone non debbano disperatamente rincorrere il denaro se non eliminando il controllo delle corporation? E come possiamo rimuovere tale controllo? Non lo so, ma ecco perché sto tentando di proporre prima un sistema di copyright di compromesso e, successivamente, il pagamento volontario sulla base di un tale sistema di compromesso come modo per procurare un reddito a chi scrive queste opere.

Domanda: Come ti aspetti in pratica di realizzare questo sistema di diritto d'autore di compromesso sotto la stretta soffocante degli interessi delle corporation sui politici americani, dovuti al sistema di finanziamento delle campagne elettorali?

Stallman: Non saprei. Vorrei saperlo. È un problema terribilmente difficile. Se sapessi come risolvere questo problema, lo risolverei e niente al mondo mi renderebbe più fiero.

Domanda: Come si può lottare contro il controllo delle corporation? Perché, se consideriamo le somme di denaro attivato dalle lobby aziendali nelle cause processuali, è enorme. Credo che il caso

del DeCSS (Decryption of Contents Scrambling System) di cui stai parlando, stia costando qualcosa come un milione e mezzo di dollari alla difesa. Dio sa quanto stia costando alle corporation. Hai una qualche idea di come avere a che fare con queste enormi somme di denaro?

Stallman: Ho una proposta. Se suggerissi di boicottare totalmente i film, credo che la gente lo ignorerebbe. Lo considererebbero troppo radicale. Perciò vorrei offrire un suggerimento leggermente diverso, ma che, alla fine, arriva quasi allo stesso risultato, e cioè: non andate a vedere un film a meno che non abbiate un valido motivo per pensare che è bello.

Questo condurrà in pratica allo stesso risultato di boicottare totalmente i film di Hollywood. Per estensione è quasi identico, ma nelle intenzioni è molto diverso. Mi sono reso conto che molta gente va al cinema per ragioni che non hanno nulla a che fare con il fatto di ritenere valido quel film. Se cambiamo le nostre abitudini, se si va a vedere un film solo quando si ha qualche sostanziale ragione per pensare che sia valido, si toglierà loro un sacco di soldi.

Thorburn: Una maniera per capire tutto questo discorso, penso, è riconoscere che quando una qualunque tecnologia radicale e potenzialmente rivoluzionaria fa la sua comparsa nella società, si crea uno scontro su chi la controlla. Oggi stiamo ripetendo quello che è avvenuto in passato. Perciò da questo punto di vista può darsi che non ci sia motivo di disperare, o anche solo di essere pessimisti, su quanto avverrà nel lungo periodo. Ma a breve termine la lotta per il controllo dei testi e delle immagini, su tutte le altre forme di informazione, sarà probabilmente dolorosa e pervasiva. Per esempio, come insegnante di

tecnologie della comunicazione il mio accesso alle immagini è stato di recente limitato in una maniera mai vista prima. Se scrivo un saggio in cui voglio utilizzare immagini, tratte anche da film, è diventato molto più difficile ottenere il permesso di utilizzarle, e i prezzi richiesti per usarle sono molto più elevati – anche quando sostengo argomenti quali la ricerca intellettuale e la categoria legale dell’uso legittimo (“fair use”). Per questo ritengo che, in un momento di diffuse trasformazioni, le prospettive di lungo periodo possano non essere così sconvolgenti come quanto accade a breve termine. Ma in ogni caso dobbiamo comprendere che l’insieme della nostra esperienza contemporanea è una versione rinnovata dello scontro per il controllo delle risorse tecnologiche che è un principio ricorrente della società occidentale.

È anche essenziale capire che la storia delle tecnologie più antiche è di per sé una materia complessa. L’impatto della stampa in Spagna, per esempio, è radicalmente diverso dall’impatto avuto in Inghilterra o in Francia.

Domanda: Una delle cose che mi sconcerta nelle discussioni sul diritto d’autore è che spesso si comincia con: “Vogliamo un cambiamento totale. Vogliamo sbarazzarci di ogni tipo di controllo”. Mi pare che a monte della suddivisione nelle tre categorie suggerite ci sia il riconoscimento che il copyright abbia qualche senso. Alcuni critici dell’attuale sistema del diritto d’autore credono che in effetti bisognerebbe sostenerlo e farlo funzionare in modo molto più simile a brevetti e marchi registrati per quanto riguarda la sua durata. Vorrei che il nostro ospite commentasse questa strategia.

Stallman: Concordo sul fatto che abbreviare la durata del dirit-

to d'autore sia una buona idea. Non c'è assolutamente bisogno, per quanto riguarda l'incoraggiamento alla pubblicazione, della possibilità che i diritti d'autore durino fino a 150 anni, cosa possibile in alcuni casi con le attuali legislazioni. Ora, le aziende sostenevano che un diritto d'autore di 75 anni su un'opera da loro pagata non fosse abbastanza lungo per renderne possibile la produzione. Vorrei sfidare queste aziende a presentare proiezioni di bilancio per i prossimi 75 anni a partire da ora, onde validare una simile affermazione. Quel che volevano davvero era semplicemente poter estendere il copyright sulle vecchie opere, in modo da poter continuare a restringerne l'utilizzo. Sinceramente mi sfugge come si possa incoraggiare una maggiore produzione di opere prodotte negli anni Venti estendendo oggi il diritto d'autore, a meno che tali aziende non abbiano una macchina del tempo da qualche parte. Certamente in uno dei loro film avevano una macchina del tempo. Quindi forse è questo che li ha influenzati.

Domanda: Hai mai pensato di estendere il concetto di “uso legittimo”, e potresti chiarircene qualche sfumatura?

Stallman: L'idea di dare a tutti il permesso di fare copie integrali per usi non commerciali, per due dei tre tipi di opere, certamente potrebbe essere intesa come un'estensione dell'uso legittimo (“fair use”). È un concetto più ampio dell'attuale. Se l'idea è che il pubblico rinunci a certe libertà per avere più progresso, allora si può segnare il confine in vari punti diversi: quali libertà il pubblico abbandona e quali libertà mantiene?

Domanda: Per ampliare un attimo la discussione, in certi campi dello spettacolo esiste il concetto di rappresentazione pubblica. Così, ad esempio, il diritto d'autore non ci impedisce di cantare i

canti natalizi al momento opportuno, ma impedisce la loro esecuzione in pubblico. E mi chiedo se non sarebbe utile espandere l'uso legittimo, anziché alla copia letterale e non commerciale senza limiti, a qualcosa di più restrittivo ma comunque più ampio del concetto attuale di "fair use".

Stallman: Prima pensavo che ciò sarebbe stato sufficiente, ma poi Napster mi ha convinto del contrario, perché gli utenti lo usano per una redistribuzione letterale non commerciale. Il server di Napster, in sé, è un'attività commerciale, ma chi mette a disposizione il materiale lo fa senza scopo di lucro, ed avrebbe potuto altrettanto facilmente metterlo a disposizione sui propri siti web. L'incredibile eccitazione, interesse e utilizzo di Napster ne dimostra la grande utilità. Perciò ora sono convinto che si debba avere il diritto a copie ridistribuite, non commerciali e letterali di qualsiasi cosa.

Domanda: Un'analogia suggeritami recentemente dall'intera vicenda di Napster è quella di una biblioteca pubblica. Credo che quanti abbiano seguito le discussioni su Napster devono averla già sentita. Vorrei che la commentassi. A volte, quanti difendono la posizione secondo cui Napster dovrebbe continuare senza restrizioni, sostengono qualcosa del tipo: "Quando si va in una biblioteca pubblica e si prende in prestito un libro, non lo si paga, e lo si può prendere in prestito decine, centinaia di volte, senza alcun pagamento aggiuntivo. Perché Napster sarebbe diverso?"

Stallman: Non è esattamente la stessa cosa. Ma bisogna sottolineare che gli editori vogliono trasformare le biblioteche pubbliche in negozi "pay-per-use". Quindi sono contro le biblioteche pubbliche.

Domanda: Queste idee sul copyright potrebbero suggerire nuove soluzioni per certe questioni relative alle leggi sui brevetti, come la produzione di farmaci generici a basso costo da usare in Africa?

Stallman: No, sono due cose completamente diverse. Le questioni sui brevetti sono completamente diverse da quelle sul diritto d'autore. L'idea che abbiano qualcosa in comune è una delle spiacevoli conseguenze dell'uso del termine "proprietà intellettuale", e della pressione a tentare di trattare alla stessa stregua queste questioni, perché, come avete sentito, finora ho parlato di questioni in cui il prezzo della copia non è l'elemento centrale. Ma qual è la questione cruciale a proposito della produzione di farmaci per l'AIDS da usare in Africa? È il prezzo, null'altro che il prezzo.

Qui spunta fuori la questione di cui parlavo, perché la tecnologia dell'informazione digitale offre a ogni utente la possibilità di fare copie. Insomma, nulla può dare a tutti la capacità di copiare dei medicinali. Io non sono capace di copiare un medicinale che ho. E nessuno è capace; i medicinali non si fanno così. Quei medicinali si possono fare solo in grandi industrie e in effetti sono tutti prodotti in costose società centralizzate, sia i farmaci generici sia quelli importati dagli Stati Uniti. In ogni caso, sono destinati ad essere prodotti in un piccolo numero di aziende, e la questione è semplicemente quanto costino e se siano disponibili ad un prezzo che gli africani possano permettersi.

Si tratta perciò di una questione terribilmente importante, ma completamente diversa. Esiste un'unica area in cui emerge un problema con i brevetti che in effetti è simile alle questioni concernenti la libertà di copia, cioè il settore dell'agricoltura. Poiché in effetti ci sono delle cose brevettate che possono essere

copie, più o meno, e sono gli esseri viventi. Si copiano quando si riproducono. Non è necessariamente una copia esatta, c'è un rimescolamento di geni. Ma il fatto è che da millenni i contadini sfruttano la capacità di autocopiarci degli organismi viventi che coltivano. L'agricoltura consiste, essenzialmente, nel copiare le cose che si sono coltivate e continuare a copiarle ogni anno. Quando varietà vegetali e animali vengono brevettate, quando dei geni vengono brevettati e utilizzati tra loro, il risultato è che i contadini non possono più comportarsi come hanno sempre fatto.

Un contadino canadese aveva una varietà brevettata che cresceva nel suo campo, e spiegò: "Non l'ho fatto apposta. Il polline è stato trascinato dal vento, e i geni di quel polline sono entrati nel mio assortimento di piante". Gli è stato risposto che ciò non importava: doveva distruggerle in ogni caso. È un caso limite di quanto il governo possa appoggiare un monopolista.

Quindi credo che, seguendo gli stessi principi che applico alla copia di cose sul computer, gli agricoltori dovrebbero avere l'indiscutibile diritto di mettere da parte i semi e di fare incroci con il bestiame. Ci potrebbero forse essere brevetti che proteggano i produttori di semi, ma non dovrebbero comunque influenzare l'operato dei contadini.

Domanda: Perché un modello abbia successo serve ben più di una licenza. Puoi parlarcene?

Stallman: Certo. Ecco, non è che possa trovare tutte le risposte giuste. Ma credo che l'idealismo abbia una parte essenziale nello sviluppo di un'informazione libera e funzionale. Occorre capire l'importanza di mantenere libera l'informazione, solo quando è libera se ne può fare pieno uso: quando è limitata, è impossibile

farlo. Bisogna riconoscere che l'informazione non libera è un tentativo di dividerci, tenerci impotenti e sottometterci. Allora si potrà afferrare il concetto: "Lavoriamo insieme per produrre l'informazione che vogliamo usare, in modo che non sia sotto il controllo di qualche persona potente che ci ordini cosa possiamo farne". Ciò offre una tremenda spinta [allo sviluppo della comunità del software libero]. Non so quanto potrebbe funzionare in aree diverse, ma credo sia possibile farlo concretamente nel campo dell'istruzione, quando si cercano libri di testo. Ci sono tantissimi insegnanti al mondo, docenti che non lavorano in università prestigiose (magari insegnano alle superiori, o al college) dove non scrivono né pubblicano un sacco di cose e non sono molto ricercati.

Ma molti di loro sono brillanti. Molti conoscono bene le loro materie e potrebbero scrivere libri di testo per molte discipline, condividerli con il mondo intero e ricevere molti apprezzamenti da chi avrà imparato da loro.

Domanda: È quanto avevo proposto. Ma, per combinazione, conosco la storia dell'istruzione. È il mio lavoro: progetti educativi, elettronici, multimediali. E non sono riuscito a trovare neppure un esempio di questo tipo. Tu ne conosci qualcuno?

Stallman: No. Ho cominciato a proporre questa enciclopedia e a seguire le risorse per l'apprendimento libero un paio di anni fa, e credevo che ci sarebbero voluti una decina d'anni perché la cosa cominciasse a funzionare. Invece già adesso abbiamo un'enciclopedia che funziona.

Quindi le cose stanno andando più velocemente di quanto sperassi. Credo che ci serva qualcuno che cominci a scrivere dei libri di testo liberi.

Scrivetene uno sulla vostra materia preferita, o scrivetene una parte. Scrivetene alcuni capitoli e sollecitate altri a finirlo.

Domanda: Veramente cercavo qualcosa di più. Quello che conta in quest'ambiente è qualcuno che crei un'infrastruttura a cui chiunque altro possa contribuire. Non c'è da nessuna parte un'infrastruttura a livello di scuola di base a cui contribuire per questo tipo di materiali. Le informazioni si possono ottenere da molti posti, ma non sono fornite sotto licenze libere, quindi non si possono usare per fare un libro di testo libero.

Stallman: Veramente non esiste un copyright sulle informazioni in sé. Il diritto d'autore copre il modo in cui un'informazione è scritta. Quindi si può imparare una materia da qualsiasi parte e poi scrivere un libro di testo, e poi rendere quel libro di testo libero, se si vuole.

Domanda: Ma non posso scrivere da solo tutti i libri di testo di cui uno studente ha bisogno nella sua carriera scolastica.

Stallman: Certo, è vero. Ma anch'io non ho scritto un intero sistema operativo libero. Ho scritto qualche pezzo e invitato altri ad unirsi e scrivere altri pezzi. Quindi, ho solo dato l'esempio. Ho detto: "Vado in questa direzione. Unitevi a me e raggiungeremo l'obiettivo". E si è unita abbastanza gente finché siamo riusciti a raggiungerlo. Può essere scoraggiante pensare in termini di "Come farò da solo a finire quest'immenso lavoro?". Quindi il punto è: non considerarlo in questo modo, ma pensa a fare il primo passo e ti accorgerai che dopo averlo fatto, altri faranno nuovi passi e, insieme, alla fine il progetto sarà portato a termine.

Supponendo che l'umanità non si spazzi via da sola, il lavoro

che facciamo oggi per produrre l'infrastruttura educativa libera, la risorsa libera di apprendimento per il mondo, sarà utile finché esisterà l'umanità.

Anche se ci volessero vent'anni per farla, cosa importa? Non bisogna pensare alle dimensioni dell'intero lavoro, ma alle dimensioni della parte che si vuol fare. Ciò dimostrerà a tutti che è possibile crearla, e così altri faranno altre parti.

Questa è la trascrizione riveduta di un intervento tenuto durante il Communications Forum svoltosi al MIT il 19 aprile 2001, e fa parte del libro *Free Software, Free Society: The Selected Essays of Richard M. Stallman*, GNU Press, 2002.

La copia letterale e la distribuzione di questo testo nella sua integrità sono permesse con qualsiasi mezzo, a condizione che sia mantenuta questa nota.

Software libero: libertà e cooperazione

Introduzione

Mike Uretsky: Sono Mike Uretsky della Stern School of Business. Sono anche uno dei condirettori del Center for Advanced Technology. E a nome di tutti noi del Dipartimento d'informatica, sono qui a darvi il benvenuto. Vorrei fare alcuni commenti, prima di passare il microfono a Ed, che introdurrà il nostro relatore.

Il ruolo di una università è quello di stimolare il dibattito e proporre discussioni interessanti. E il ruolo di una grande università è quello di proporre discussioni particolarmente interessanti. E questa specifica presentazione, questo seminario rientra in questa categoria. Trovo particolarmente interessanti le discussioni sull'open source. In un certo senso.... [il pubblico ride]

Richard M. Stallman: Io faccio software libero. L'open source è un movimento diverso. [il pubblico ride] [applausi]

Mike Uretsky: Quando negli anni '60 sono entrato per la prima volta in questo campo, praticamente il software era libero. E si andava avanti seguendo dei cicli. Diventò libero, e poi i produttori di software, per la necessità di espandere il mercato, lo spinsero in altre direzioni. Gran parte dello sviluppo che ebbe

luogo con l'arrivo del PC seguì esattamente il medesimo tipo di ciclo.

C'è un filosofo francese molto interessante, Pierre Levy, che parla del movimento in questa direzione e del passaggio al cyberspazio non soltanto in quanto connesso alla tecnologia ma anche alla ristrutturazione sociale, alla ristrutturazione politica, tramite un cambiamento nel tipo di relazioni che miglioreranno il benessere dell'umanità. Speriamo che questo dibattito rappresenti un movimento in tale direzione, che questo dibattito sia qualcosa che tagli attraverso le molte discipline che normalmente agiscono in solitudine all'interno dell'Università. Confidiamo in una discussione molto interessante. Ed?

Ed Schonberg: Sono Ed Schonberg del Dipartimento d'informatica presso il Courant Institute. Dò a tutti voi il benvenuto a quest'evento. In genere, e particolarmente qui, chi fa l'introduzione è un aspetto inutile delle presentazioni pubbliche, ma in questo caso in realtà presenta una qualche utilità, come Mike ha facilmente dimostrato, perché un commento inaccurato di chi fa l'introduzione, ad esempio, può consentire [al relatore] di raddrizzare e correggere, [il pubblico ride] per rifinire in maniera considerevole i parametri del dibattito.

Consentitemi perciò di fare la più breve introduzione possibile a una persona che non ne ha bisogno. Richard è il perfetto esempio di qualcuno che, agendo a livello locale, ha iniziato a pensare a livello globale – a partire, parecchi anni fa, dai problemi concernenti l'indisponibilità del codice sorgente dei driver della stampante al laboratorio di intelligenza artificiale del MIT. Egli ha messo a punto una filosofia coerente che ha costretto tutti noi a riesaminare le nostre idee su come viene prodotto il software, su cosa significa proprietà intellettuale, e su cosa rap-

presenta in concreto la comunità del software. Diamo il benvenuto a Richard Stallman. [applausi]

Software libero: libertà e cooperazione

Richard M. Stallman: Qualcuno può prestarmi un orologio? [il pubblico ride] Grazie. Vorrei ringraziare Microsoft per avermi fornito l'opportunità [il pubblico ride] di essere su questo podio. Nelle ultime settimane mi sono sentito come un autore il cui libro è stato fortuitamente vietato da qualche parte.¹ [il pubblico ride] Eccetto per il fatto che tutti gli articoli sulla vicenda riportano in maniera errata il nome dell'autore, perché Microsoft descrive la GNU GPL come una licenza open source, e lo stesso ha fatto gran parte della stampa. La maggior parte della gente, ovviamente altrettanto in buona fede, non si rende conto che il nostro lavoro non ha nulla a che fare con l'open source, che anzi abbiamo svolto la maggior parte di tale lavoro prima che il termine "open source" venisse perfino coniato.

Noi facciamo parte del movimento del software libero, e mi accingo a parlare su cosa sia il movimento del software libero, cosa significhi, quanto abbiamo fatto e, poichè quest'evento è in parte sponsorizzato da una School of Business, dirò qualcosa, più di quanto sia solito fare, sulla relazione tra software libero e imprenditoria, e su altri ambiti della vita sociale.

Ora, alcuni di voi non scriveranno mai programmi informatici, ma forse cucinate. E se cucinate, a meno che non siate davvero bravi, probabilmente userete delle ricette. E se usate delle ricette, è probabile abbiate sperimentato di ricevere la copia di una ricetta da un amico che voglia dividerla. E probabilmente avrete

¹ *Meno di un mese prima, il vicepresidente di Microsoft, Craig Mundie, aveva tenuto un intervento in cui attaccava il software libero (definendolo "open source").*

anche sperimentato, a meno che non siate dei completi neofiti, l'atto di modificare una ricetta. Questa dice certe cose, ma non dovete seguirla in maniera esatta. Se ne può omettere qualche ingrediente. Aggiungervi dei funghi perché vi piacciono. Metterci meno sale perché il medico vi ha consigliato di ridurre il sale – o quant'altro. Si possono anche apportare cambiamenti sostanziali, a seconda delle capacità individuali. E se avete modificato qualcosa in una ricetta, e la preparate per degli amici, e a loro piace, un amico potrebbe dire, "Puoi darmi la ricetta?" E allora cosa fate? Scrivete la vostra versione modificata della ricetta e ne date una copia all'amico. Questa è la cosa naturale da fare con ricette utili e funzionali di ogni tipo.

Una ricetta assomiglia molto a un programma informatico. E un programma informatico è assai simile a una ricetta: una serie di passaggi per arrivare al risultato che ci si è prefissi. Perciò è altrettanto naturale fare la stessa cosa con i programmi informatici – passarne una copia agli amici. E apportarvi delle modifiche, perché il lavoro per cui era stato scritto non è esattamente quanto vogliamo. Ha fatto un buon lavoro per qualcun altro, ma il nostro lavoro è diverso. E, dopo averlo modificato, è probabile possa tornare utile ad altri. Forse costoro devono fare un lavoro simile al nostro. Così ci chiederanno, "Posso averne una copia?" Naturalmente, se vogliamo essere gentili, gliene diamo una copia. È così che si comporta una persona deccente. Immaginiamo allora cosa accadrebbe se le ricette venissero impacchettate dentro scatole nere. Non se ne potrebbero vedere gli ingredienti usati, per non parlare neppure di modificarli, e immaginando di averne fatto una copia per un amico, vi chiamerebbero pirata e cercherebbero di sbattervi in galera per anni. Un mondo simile creerebbe proteste tremende da parte di tutti coloro che sono soliti scambiare ricetta-

te. Ma questo è esattamente il mondo del software proprietario. Un mondo in cui la comune decenza verso gli altri è proibita o impedita.

Ora, come mi sono accorto di tutto ciò? Me ne sono reso conto perché negli anni '70 ho avuto la buona fortuna di far parte di una comunità di programmatori che condividevano il software. Le radici di questa comunità possono essere rintracciate sostanzialmente fino agli albori dell'informatica. Tuttavia negli anni '70 era un po' raro trovare una comunità in cui si condivideva il software. Si trattava anzi di un caso estremo, perché nel laboratorio dove lavoravo l'intero sistema operativo era software sviluppato dai membri della comunità, e lo condividevamo con tutti. Chiunque era il benvenuto nel venire a dare un'occhiata e prenderne una copia, e farci quel che voleva. Su questi programmi non c'era alcuna nota di copyright. La cooperazione era il nostro modo di vivere. E ci sentivamo sicuri di vivere a quel modo. Non lottammo per ottenerlo. Non dovemmo combattere per averlo. Vivevamo semplicemente in quel modo. E, per quanto ci riguardasse, avremmo continuato a vivere così. C'era il software libero, ma non il movimento del software libero.

Ma poi la comunità fu distrutta da una serie di calamità che la colpirono. Alla fine fu spazzata via. Alla fine il computer PDP-10,² che usavamo per ogni lavoro, venne messo fuori uso.

Il nostro sistema – denominato Incompatible Timesharing System – fu scritto a partire dagli anni '60, perciò era scritto in linguaggio assembler. Negli anni '60 si usava tale linguaggio per scrivere un sistema operativo. Ovviamente l'assembler vale per una particolare architettura informatica; se non viene più usato, tutto il

² *Programming Data Processor modello 10, un computer mainframe usato negli anni '70 da molti importanti enti di ricerca e governativi.*

lavoro svolto diventa polvere – è inutile. E questo è quanto avvenne nel nostro caso. I circa 20 anni di lavoro della nostra comunità divennero polvere.

Ma prima che ciò accadesse, ebbi un'esperienza che mi preparò, mi aiutò a capire cosa fare, perché ad un certo punto la Xerox diede al laboratorio di intelligenza artificiale, dove lavoravo, una stampante laser, e fu un regalo stupendo, perché era la prima volta che qualcuno al di fuori della Xerox aveva una stampante laser. Era molto veloce, stampava una pagina al secondo, assai precisa sotto molti punti di vista, ma era inaffidabile, perché in realtà era una fotocopiatrice per ufficio ad alta velocità modificata in stampante. E le fotocopiatrici s'incestrano, ma c'è qualcuno pronto a sistemarle. La stampante s'incestrava e nessuno se ne accorgeva. Così rimaneva bloccata per parecchio tempo.

Be', ci venne un'idea per risolvere il problema. Modificarla in modo che ogni volta che la stampante s'inceppava, il computer che la gestiva poteva informarne la nostra macchina timesharing, e far sapere agli utenti in attesa della stampa di andare a sistemare la stampante – perché se soltanto avessero saputo che era incestrata... ovviamente, se sei in attesa di una stampa e sai che la stampante è inceppata, non vuoi startene seduto ad aspettare per sempre, ti alzi e vai a sistemarla.

Ma a quel punto eravamo completamente bloccati, perché il software che gestiva la stampante non era software libero. Era arrivato incluso nella stampante, era soltanto un file binario. Non potevamo averne il codice sorgente; la Xerox non ci avrebbe fatto avere il codice sorgente.

Così, nonostante le nostre capacità di sviluppatori – dopotutto avevamo scritto il nostro sistema timesharing – eravamo del tutto inadeguati ad aggiungere questa funzione al software della stampante.

Non ci restava che soffrire rimanendo in attesa. Per stampare qualcosa ci voleva un'ora o due, perché la maggior parte delle volte la stampante s'inceppava. Aspettavi un'ora pensando, "So che finirà per incastrarsi. Aspetterò un'ora e poi andrò a prendere la mia stampa", e allora ti rendevi conto che era rimasta incastrata per tutto il tempo, nessun altro l'aveva sistemata. Così la rimettevi a posto e aspettavi un'altra mezz'ora. Poi tornavi a controllare, e vedevi che si era nuovamente inceppata, prima di eseguire il tuo lavoro. Stampava per tre minuti e poi s'inceppava per trenta minuti. La frustrazione saliva alle stelle. Ma la cosa peggiore era sapere che avremmo potuto risolvere la cosa, eppure qualcun altro, per egoismo personale, ci bloccava, ci impediva di migliorare il software. Così, naturalmente, ce ne risentimmo.

Allora venni a sapere che qualcuno alla Carnegie Mellon University aveva una copia di quel software. Qualche tempo dopo mi ci recai in visita, andai nel suo ufficio e gli feci, "Salve, vengo dal MIT. Potrei avere una copia del codice sorgente della stampante?" E lui replicò, "No, ho promesso che non ve l'avrei data." [il pubblico ride] Rimasi di stucco. Ero talmente – talmente arrabbiato, e non avevo alcuna idea su come ottenere giustizia. Tutto ciò che riuscii a pensare fu di girarmi sui tacchi e uscire da quella stanza. Forse ho sbattuto la porta. [il pubblico ride] E ripensandoci più tardi, mi resi conto che non stavo osservando un tipaccio isolato, ma un fenomeno sociale che era importante e colpiva parecchie persone.

Fui fortunato, ne ebbi appena un assaggio. Altri dovevano farci i conti tutto il tempo. Ci riflettei sopra a lungo. Vedete, quel tizio aveva promesso di rifiutare ogni collaborazione con noi, i colleghi del MIT. Ci aveva traditi. Ma non lo fece soltanto con noi. È probabile che abbia tradito anche te [indicando qualcuno tra il pub-

blico]. E credo che molto probabilmente abbia fatto lo stesso a te [indicando qualcun altro tra il pubblico] [il pubblico ride]. Ed è probabile lo abbia fatto anche a te [indicando una terza persona tra il pubblico]. Probabilmente ha tradito la maggioranza dei presenti in questa sala – eccetto, forse, i pochi che non erano ancora nati nel 1980. Perché aveva promesso di rifiutare ogni cooperazione praticamente con l'intera popolazione del pianeta terra. Aveva firmato un accordo di non divulgazione (“non-disclosure agreement”).

Ora, questo era il mio primo incontro diretto con un accordo di non divulgazione, e m'insegnò una lezione importante – una lezione che è importante perché la maggioranza dei programmatori non l'impara mai. Quello fu il mio primo incontro con un accordo di non divulgazione, e io ne ero la vittima. Io e l'intero laboratorio ne fummo le vittime. E la lezione che m'insegnò fu che gli accordi di non divulgazione provocano delle vittime. Non sono qualcosa d'innocente. Non sono innocui. La maggior parte dei programmatori s'imbattono per la prima nell'accordo di non divulgazione quando vengono invitati a firmarne uno. E c'è sempre qualche tentazione – qualche vantaggio che finiscono per ottenere se firmano. Così inventano qualche scusa. Dicono, “Be', quel tizio non riuscirà mai a ottenerne una copia in ogni caso, perché quindi non dovrei unirmi alla cospirazione per impedirglielo?” Dicono, “Si è sempre fatto così. Chi sono io per oppormici?” Dicono, “Se non lo firmo io, lo farà qualcun altro.” Scuse varie per mettere il bavaglio alla propria coscienza.

Ma quando qualcuno mi invitò a firmare un accordo di non divulgazione la mia coscienza era già sensibilizzata. Si ricordò di quanto fossi arrabbiato quando qualcuno promise di non aiutare me e l'intero laboratorio a risolvere il problema. Non potevo voltarmi

dall'altra parte e fare la stessa identica cosa a qualcun altro che non mi aveva mai arrecato alcun danno. Se qualcuno mi avesse chiesto di promettere di non condividere qualche informazione utile con un odioso nemico, avrei detto di sì. Se qualcuno ha fatto qualcosa di male, lo merita. Ma gli estranei – non mi hanno fatto alcun danno. Come possono meritare quel tipo di trattamento negativo? Non puoi consentire a te stesso di trattare male praticamente tutti e chiunque. A quel punto diventi un predatore sulla società. Così risposi, “Grazie mille per avermi offerto questo bel pacchetto software. Ma in tutta coscienza non posso accettarlo sulla base delle condizioni richieste, perciò ne farò a meno. Grazie molte.” E così, non ho mai firmato volontariamente un accordo di non divulgazione su informazioni tecniche d'utilità generica come il software.

Esistono altri tipi d'informazioni che sollevano questioni etiche diverse. Ad esempio, le informazioni personali. Se un'amica vuole raccontarmi quel che va accadendo tra lei e il suo ragazzo, e mi chiede di non rivelarlo a nessuno, posso dirmi d'accordo nel mantenere il segreto, perché non si tratta di informazioni tecniche d'utilità generica.

Almeno, probabilmente non si tratta di informazioni generalmente utili [il pubblico ride]. Esiste una scarsa probabilità – e comunque potrebbe essere possibile – che l'amica possa rivelarmi qualche nuova strabiliante tecnica sessuale [il pubblico ride], e allora sentirei il dovere morale [il pubblico ride] di informarne il resto dell'umanità, in modo che tutti possano trarne beneficio. Forse dovrei porre una condizione a quella promessa.

Qualora si trattasse soltanto di dettagli su chi voglia questa cosa, e chi s'arrabbia contro chi, e robe da telenovela... tutto ciò posso tenerlo in privato, ma non così per qualcosa di cui l'umanità possa beneficiare terribilmente, qualora ne fosse informata. Obiettivi

vo della scienza e della tecnologia è quello di sviluppare informazioni utili per l'umanità, onde aiutare la gente a vivere una vita migliore. Se promettiamo di non rivelare tali informazioni, se le teniamo segrete, allora stiamo tradendo la missione della nostra disciplina. E ciò, decisi, non dovrei farlo.

Ma nel frattempo la mia comunità si era frantumata, e mi ritrovai in una brutta situazione. Vedete, l'intero Incompatible Time-sharing System divenne obsoleto, perché lo era il PDP-10, e così non esisteva alcun modo per cui potessi continuare a lavorare in quanto sviluppatore di un sistema operativo come avevo fatto fino ad allora. Ciò dipendeva dal far parte di una comunità, dall'usare il software della comunità e dal migliorarlo. Questa possibilità non esisteva più, e ciò mi pose un dilemma morale. Cosa avrei fatto? Perché la possibilità più ovvia consisteva nell'andare contro quella decisione che avevo preso. Accettare che le cose fossero diverse, che dovevo semplicemente abbandonare quei principi e iniziare a firmare accordi di non divulgazione per sistemi operativi proprietari, e molto probabilmente anche scrivere software proprietario. Ma compresi che in tal modo avrei potuto divertirmi con il codice e guadagnare bene – soprattutto se l'avessi fatto al di fuori del MIT – ma alla fine, osservando la mia carriera all'indietro avrei detto, “Ho speso la vita a costruire muri che dividono la gente”, e mi sarei vergognato di quella vita.

Così mi son messo alla ricerca di un'alternativa, e una era ovvia. Avrei potuto lasciare il campo del software e mettermi a fare qualcosa d'altro. Non ero dotato di altre capacità particolari, ma sono sicuro che avrei potuto fare il cameriere [il pubblico ride]. Non in un ristorante di lusso, non mi avrebbero assunto, [il pubblico ride] ma da qualche parte avrei fatto il cameriere. Molti sviluppatori mi dicono, “Quelli che assumono i programmatori richiedono que-

sto, questo e questo. Se non lo faccio, morirò di fame”. È la terminologia che usano letteralmente. Be’, come cameriere non si può morire di fame. [il pubblico ride] Così, in realtà, non si trovano affatto in pericolo. Ma – e questo è importante – talvolta tendiamo a giustificare qualcosa che danneggia gli altri sostenendo che altrimenti a noi accadrà qualcosa di peggio. Se costoro stessero veramente morendo di fame, allora sarebbero giustificati a scrivere software proprietario [il pubblico ride]. Se qualcuno ti punta contro una pistola, allora direi che è perdonabile [il pubblico ride]. Ma trovai il modo di sopravvivere senza dover fare qualcosa di poco etico, perciò quella scusa non si può usare. Però mi sono reso conto che fare il cameriere non sarebbe stato divertente, e avrei sprecato le mie capacità in quanto sviluppatore di sistemi operativi. Almeno avrebbe evitato di usare male tali capacità. Sviluppare software proprietario avrebbe significato usare male le mie capacità. Meglio perciò sprecarle che usarle male, ma non è davvero una bella cosa.

Per queste ragioni, decisi di cercare altre alternative. Cosa può fare qualcuno che sviluppa sistemi operativi per migliorare veramente la situazione, per rendere migliore il mondo? E mi resi conto che ciò di cui c’era bisogno era esattamente qualcuno capace di sviluppare sistemi operativi. Il problema, il dilemma, esisteva per me e per chiunque altro, poiché tutti i sistemi operativi disponibili per i computer moderni erano proprietari. I sistemi operativi liberi erano per computer vecchi e obsoleti, giusto? Così per i computer moderni – chi voleva avere un computer moderno e usarlo era costretto a ricorrere a un sistema operativo proprietario. Perciò se uno sviluppatore avesse scritto un altro sistema operativo per poi dire, “Venite tutti qui e condividete questo sistema, siete i benvenuti” – ciò avrebbe offerto a chiunque un via d’uscita al

dilemma, un'alternativa. Compresi così che c'era qualcosa che avrei potuto fare per risolvere il problema. Avevo proprio le capacità adatte per riuscirci. Ed era la cosa più utile che avessi potuto immaginare di fare con la mia vita. Si trattava di un problema che nessun altro stava cercando di risolvere. Se ne stava lì, a peggiorare, e non c'era nessuno tranne il sottoscritto. Così mi dissi qualcosa come, "Sono un eletto. Devo lavorarci sopra. Se non io, chi?" Decisi perciò che avrei sviluppato un sistema operativo libero, oppure sarei morto provandoci... di vecchiaia, naturalmente [il pubblico ride].

Ovviamente dovevo decidere che tipo di sistema operativo sarebbe stato. Bisognava prendere delle decisioni di progettazione tecnica. Per una serie di motivi, optai per renderlo compatibile con Unix. Prima di tutto, avevo appena visto diventare obsoleto un sistema operativo che amavo davvero, perché scritto per un tipo di computer specifico. Non volevo che accadesse di nuovo. Dovevamo avere un sistema portatile. Be', Unix lo era. Perciò se ne avessi seguito il progetto, avrei avuto buone probabilità di realizzare un sistema che sarebbe stato anch'esso portatile e funzionale. Inoltre, perché non renderlo compatibile nei dettagli? Il motivo è: gli utenti odiano le modifiche incompatibili. Se avessi progettato il sistema secondo le mie preferenze – cosa che mi sarebbe piaciuto molto fare, ne sono certo – avrei prodotto qualcosa di non compatibile. I dettagli sarebbero stati diversi. Se avessi scritto un tale sistema, la gente mi avrebbe detto, "È molto bello, ma incompatibile. Richiede troppo lavoro passare a questo. Non possiamo permetterci tanti problemi giusto per usare il tuo sistema invece di Unix, per cui ce ne restiamo con Unix".

Se avessi voluto creare concretamente una comunità di persone che usavano questo sistema libero e traevano vantaggio dalla

libertà e dalla cooperazione, dovevo realizzare un sistema che la gente avrebbe utilizzato, un sistema al quale sarebbe stato semplice passare, che non doveva cadere su un ostacolo simile appena all'inizio. Rendere il sistema compatibile con Unix fece decidere tutta la successiva progettazione, perché Unix è composto da molti pezzi che comunicano tra loro grazie a interfacce più o meno documentate. Se si vuole essere compatibili con Unix occorre quindi sostituire ciascun pezzo, uno ad uno, con un altro compatibile. Le rimanenti decisioni di progettazione rimangono all'interno di ciascun pezzo, e possono essere prese in seguito da chiunque decida di scrivere quel pezzo. Non devono essere decise all'inizio.

Tutto quel che dovemmo fare per iniziare a lavorare fu trovare un nome al sistema. Noi hacker cerchiamo sempre qualche nome divertente o strambo per un programma, perché riteniamo che pensare alla gente che se la ride per il nome costituisca metà del divertimento di scrivere il programma [il pubblico ride]. E abbiamo la tradizione degli acronimi ricorsivi, per dire che il programma che si sta scrivendo è in qualche modo analogo a un altro già esistente. È possibile chiamarlo con un acronimo ricorsivo per dire: questo non è quell'altro. Così, ad esempio, negli anni '60 e '70 esistevano troppi text editor Tico, e generalmente si chiamavano qualcosa-o-qualcos'altro TECO. Poi un hacker arguto chiamò il proprio Tint, per Tint Is Not TECO (Tint non è TECO) – il primo acronimo ricorsivo. Nel 1975 sviluppai il primo text editor Emacs, e c'erano parecchie imitazioni di Emacs, ma una si chiamava Fine, per Fine Is Not Emacs, e c'era Sine, per Sine Is Not Emacs, e Eine, per Eine Is Not Emacs, e MINCE per Mince Is Not Complete Emacs (questa era un'imitazione ridotta all'osso) [il pubblico ride]. Poi Eine venne riscritto quasi completamente,

e la nuova versione fu chiamata Zwei, per Zwei Was Eine Initially (Zwei era Eine all'inizio).³

Mi misi così alla ricerca di un acronimo ricorsivo per Something Is Not Unix (Qualcosa non è Unix). Provai tutte le 26 lettere dell'alfabeto (inglese) per scoprire che nessuna poteva formare una parola. [il pubblico ride] Hmm, prova qualcos'altro. Usai una contrazione. In tal modo avrei avuto un acronimo a tre lettere, per Something's Not Unix. Provai le varie lettere, e venne fuori il termine "GNU" – la parola "GNU" è la più divertente della lingua inglese. [il pubblico ride] Proprio così. Ovviamente il motivo per cui è divertente sta nel fatto che secondo il dizionario si pronuncia "new" (nuovo). Ecco perché la si usa in parecchi giochi di parole. Ma devo informarvi che si tratta del nome di un animale che vive in Africa. E la pronuncia africana aveva un suono come di un clic. [il pubblico ride] Forse ce l'ha ancora. I colonizzatori europei, quando arrivarono lì, non si preoccuparono di imparare a pronunciare quel suono di clic. Lo lasciarono fuori, e scrissero una 'g' che stava a significare "qui dovrebbe esserci un altro suono che non pronunciamo" [il pubblico ride]. Stanotte partirò per il Sud Africa, e li ho implorati, spero che riescano a trovarmi qualcuno che possa insegnarmi a pronunciare GNU nel modo corretto, quando indica l'animale.

Ma quando si tratta del nostro sistema, la pronuncia corretta è "guh-NEW" (guh-niu), con la 'g' dura. Se si parla del "new" (niu) sistema operativo, la gente finirà col confondersi perché ci stiamo lavorando ormai da 17 anni, per cui non è più così "new", nuovo. [il pubblico ride] Ma è ancora, e sarà sempre, GNU, "guh-NEW" – non importa quante persone lo chiameranno Linux per errore [il pubblico ride].

³ *Eine e Zwei significano uno e due in tedesco, rispettivamente.*

Così nel gennaio 1984, lascio il mio posto al MIT per iniziare a scrivere le varie parti di GNU.⁴

Al MIT furono così bravi da consentirmi di usare le strutture interne. Allora pensai che avremmo scritto tutti i pezzi per costruire l'intero sistema GNU, e poi avremmo annunciato, "Venite a prenderlo," e la gente avrebbe iniziato a usarlo. Non è andata così. Le prime parti che scrissi non erano altro che buone sostituzioni, con un numero minore di bug, di alcuni pezzi di Unix, ma nulla di particolarmente eccitante. Nessuno pareva interessato a volerli e a installarli. Ma poi, nel settembre 1984, iniziai a scrivere GNU Emacs, che era la mia seconda implementazione di Emacs, e prese a funzionare all'inizio del 1985. Potei usarlo per tutto il mio lavoro di editing, il che fu un grande sollievo perché non avevo alcuna intenzione di imparare a usare vi, l'editor di Unix. [il pubblico ride] Fino a quel momento, feci l'editing su qualche altra macchina, e salvavo i file via rete, in modo che potessi fare dei test. Ma quando GNU Emacs prese a girare abbastanza bene da consentirmi di usarlo, spuntò fuori altra gente che voleva usarlo.

Così dovetti occuparmi dei dettagli relativi alla distribuzione. Ovviamente ne misi una copia nella directory dell'anonymous FTP, e ciò andava bene per quanti erano in rete – bastava che prelevassero un file tar⁵ – ma c'erano anche parecchi programmatori che nel 1985 non erano ancora in rete.

Mi mandavano email chiedendo, "Come posso averne un copia?" Dovevo decidere come avrei replicato. Be', avrei potuto dire: "Voglio spendere il mio tempo scrivendo altro software GNU, non a copiare nastri, perciò trova un amico che è su Internet, che abbia

⁴ È possibile leggere l'annuncio originario del progetto GNU nel testo "Il Manifesto GNU".

⁵ Programma Unix per l'archiviazione. Integrato con gzip, rappresenta l'alternativa GNU al formato di compressione non libero ZIP.

voglia di fare il download e metterlo su nastro per te,” e sono sicuro che prima o poi avrebbero trovato qualche amico disposto a farlo. Avrebbero ottenuto quelle copie.

Ma ero senza lavoro. Anzi, fin da quando avevo lasciato il MIT nel gennaio 1984 non avevo avuto alcun impiego. Stavo cercando il modo di fare dei soldi grazie a quanto andavo facendo con il software libero, e perciò avviai un'attività commerciale di software libero. Diedi l'annuncio, “Mandami 150 dollari e ti spedisco un nastro con Emacs.” E gli ordini presero ad arrivare. A metà anno divennero regolari.

Ricevevo 8-10 ordini al mese. Se necessario, avrei potuto vivere soltanto con questi, perché ho sempre vissuto con poco. Praticamente vivo come uno studente. E mi piace, perché significa che non è il denaro a impormi cosa fare. Posso fare quel che ritengo sia importante per me. Ciò mi ha consentito di fare quel che mi sembrava valido. Sforzatevi seriamente di evitare di cadere in tutte le abitudini dello stile di vita dei tipici americani. Perché se lo fate, allora sarà la gente con i soldi a imporvi cosa fare con la vostra vita. Non sarete in grado di fare quello che è veramente importante per voi.

Le cose andavano bene, ma la gente mi chiedeva, “Cosa intendi con software libero se costa 150 dollari? [il pubblico ride] Il motivo di queste domande stava nella confusione generata dai significati multipli del termine “free” in inglese. Un significato indica il prezzo, e un altro la libertà. Quando parlo di software libero mi riferisco alla libertà, non al prezzo. Pensiamo alla libertà d'espressione (“free speech”), non alla birra gratis (“free beer”). [il pubblico ride] Non avrei certo dedicato così tanti anni della mia vita per esser certo che i programmatori guadagnassero di meno. Non è questo il mio obiettivo. Sono un programmatore anch'io e non mi

dispiacerebbe guadagnare bene. Non dedicherei la mia vita a far soldi, ma non mi dispiace averne. Quindi, dato che l'etica è la stessa per chiunque, non sono neppure contrario al fatto che altri programmatori guadagnino bene. Non voglio che i prezzi siano bassi. Non è affatto questo il punto. La questione è la libertà. Libertà per chiunque usi il software, che si tratti o meno di un programmatore.

A questo punto dovrei darvi la definizione di software libero. Meglio passare ai dettagli concreti, perché è troppo vago dire soltanto "credo nella libertà". Esistono così tante libertà in cui si può credere, e sono in conflitto tra loro, perciò la vera domanda politica è: quali sono le libertà importanti, le libertà di cui occorre assicurare l'esistenza a tutti?

Vi darò la mia risposta a questa domanda relativamente all'area dell'utilizzo del software. Un programma è "software libero" per uno specifico utente quando quest'ultimo ha le seguenti libertà:

- Primo, *libertà zero* è la libertà di far girare il programma per qualsiasi scopo, in ogni modo che si vuole.
- *Libertà uno* è la libertà di aiutare se stessi a modificare il programma secondo le proprie necessità.
- *Libertà due* è la libertà di aiutare il vicino a distribuire copie del programma.
- E *libertà tre* è la libertà di aiutare a costruire una comunità pubblicando una versione migliorata in modo che gli altri possano trarre vantaggi dal proprio lavoro.

Se avete tutte queste libertà, il programma è software libero, per l'utente – e ciò è cruciale. Ecco perché uso questa terminologia. Lo spiegherò meglio più avanti, quando parlerò della Licenza Pubblica Generica GNU (GNU GPL), ma ora illustrerò cosa significa software libero, che è una questione più fondamentale.

La libertà zero è piuttosto ovvia. Qualora all'utente non venga neppure concesso di far girare il programma come meglio preferisce, si tratta di un programma decisamente restrittivo. Ma succede che la maggior parte dei programmi concedono almeno la libertà zero. E legalmente la libertà zero deriva come conseguenza della libertà uno, due e tre – questo è il modo con cui funziona la legislazione sul copyright. Così le libertà che distinguono il software libero dal software comune sono le libertà uno, due e tre, per cui le illustrerò a fondo e ne spiegherò l'importanza.

La libertà uno è la libertà di aiutare se stessi a modificare il software secondo le proprie necessità. Ciò potrebbe significare sistemare i bug presenti. Potrebbe significare aggiungere nuove funzioni. Potrebbe significare crearne la versione per un computer diverso. Potrebbe significare tradurne i messaggi d'errore in Navajo. Qualsiasi cambiamento l'utente voglia apportare, deve essere libero di farlo.

È ovvio come gli sviluppatori professionisti possano fare uso di tale libertà in maniera assai efficace, ma non sono i soli. Chiunque dotato di una comune dose d'intelligenza può imparare un po' di programmazione. Ci sono lavori difficili e altri facili, e la maggioranza non imparerà abbastanza da fare i lavori difficili. Ma parecchia gente può imparare a sufficienza per occuparsi dei lavori facili, proprio come 50 anni fa molti uomini americani hanno imparato a riparare le automobili, situazione che consentì agli Stati Uniti di avere un esercito motorizzato nella seconda guerra mondiale e vincerla. È assai importante avere parecchie persone in grado di riparare qualcosa.

E se si tratta di qualche popolano, che non vuole saper nulla della tecnologia, ciò significa che probabilmente avrà un sacco di amici e potrà chiedere loro dei favori [il pubblico ride]. È probabile

che alcuni di questi siano dei programmatori. Così potrà chiedere loro: “Potresti modificarlo per me? Puoi aggiungerci questa funzione?” Ecco allora che potranno essere in parecchi a beneficiare di questa libertà.

Ora, se non avete tale libertà, ciò provoca danno materiale, concreto alla società. Ci rende prigionieri del nostro stesso software. Ho spiegato cosa accadde con la stampante laser. Funzionava male, e non potemmo ripararla, perché eravamo prigionieri di quel software.

Ma ciò colpisce anche il morale della gente. Se l'uso del computer è continuamente frustrante, e la gente lo usa, la loro vita finirà per essere frustrante, e se lo usano al lavoro, sarà quest'ultimo ad essere frustrante, finiranno per odiarlo. E per autoprotettersi dalla frustrazione, si decide di fregarsene. Così ci troviamo davanti persone la cui attitudine è, “Be', oggi sono andato a lavorare. Non devo fare altro. Se non compio progressi, il problema non è mio, ma del datore di lavoro”. E quando ciò accade, è negativo per costoro e negativo per la società come insieme. Questa è la libertà uno, la libertà di aiutare se stessi.

La libertà due è la libertà di aiutare il vicino di casa distribuendo copie di un programma. Per degli esseri in grado di pensare e imparare, la condivisione di conoscenze utili è un atto fondamentale di amicizia. Quando si usa il computer, quest'atto di amicizia prende la forma di condivisione del software. Gli amici condividono tra loro. Gli amici si aiutano a vicenda. È la natura stessa dell'amicizia. Anzi, questo spirito di buona volontà – lo spirito di aiutare il proprio vicino, volontariamente – è la risorsa più importante della società. È l'elemento di differenza tra una società vivibile e una giungla tipo cane-mangia-cane. Per migliaia d'anni quest'importanza è stata riconosciuta dalle maggiori religioni del

mondo, le quali cercano esplicitamente d'incoraggiare un simile atteggiamento.

Quando andavo all'asilo, i maestri cercavano d'insegnarci quest'attitudine – lo spirito della condivisione – con delle applicazioni pratiche. Credevano che in tal modo l'avremmo imparato. Così dicevano, “Se porti delle caramelle a scuola, non puoi tenerle per te; devi dividerle con gli altri bambini”. La società è stata organizzata in modo da insegnare questo spirito di cooperazione. E perché dovremmo farlo? Perché la gente non coopera del tutto. Questo è un aspetto della natura umana, ma ce ne sono altri. Esistono molti altri aspetti della natura umana. Perciò, se si vuole una società migliore bisogna incoraggiare lo spirito della condivisione. Non si arriverà mai al 100%. Ciò è comprensibile. Occorre anche prendersi cura di se stessi. Ma se in qualche modo ne ampliamo la portata, sarà un bene per tutti.

Oggi giorno, secondo il governo statunitense, gli insegnanti dovrebbero fare esattamente l'opposto. “Oh, Johnny, hai portato del software a scuola. Non puoi dividerlo. No, condividere è sbagliato. Condividere significa essere un pirata”. Cosa intendono quando dicono ‘pirata’? Sostengono che aiutare il vicino sia l'equivalente morale di attaccare una nave. [il pubblico ride]

Cosa avrebbero detto al riguardo Gesù o Budda? Scegliete pure il vostro leader religioso preferito. Non so, forse Manson avrebbe detto qualche altra cosa, [il pubblico ride] Chissà cosa direbbe Ron Hubbard? Ma...

Domanda: [inascoltabile]

RMS: Certo, è morto. Ma non l'ammettono. Cosa?

Domanda: Lo stesso vale per gli altri, tutti morti. [il pubblico ride]
Anche Charles Manson è morto. Sono tutti morti, Gesù, Budda...

RMS: Sì, è vero [il pubblico ride]. Credo allora che in tal senso Ron Hubbard non sarebbe peggiore degli altri. Comunque...

Domanda: Ron Hubbard usava sempre il software libero – lo ha liberato da Zanu [il pubblico ride].

RMS: Comunque, ritengo questo sia il motivo più importante perché il software debba essere libero: non possiamo permetterci di inquinare la risorsa più importante della società. È vero che non si tratta di una risorsa materiale come l'aria pulita e l'acqua pulita. È una risorsa psico-sociale, ma è altrettanto concreta, e produce una differenza tremenda sulle nostre vite. Le azioni che compiamo influenzano il pensiero degli altri. Quando ce ne andiamo in giro dicendo alla gente, "Non condividete tra voi", se dovessero ascoltarci, avremo un effetto sulla società, e non sarebbe certo positivo. Questa è la libertà due, la libertà di aiutare il proprio vicino.

Ah, inoltre, se tale libertà viene a mancare, ciò non provoca danno soltanto a questa risorsa psico-sociale, ma causa uno spreco – un danno pratico, materiale. Se il programma ha un proprietario, e costui organizza le cose in modo che ciascun utente debba pagare onde poterlo usare, qualcuno finirà per dire, "Vorrà dire che ne farò a meno". E ciò è uno spreco, spreco inflitto in maniera deliberata. La cosa interessante riguardo il software, naturalmente, è che quantità minore non significa doverne fare di meno. Se c'è meno gente che compra autovetture, se ne faranno di meno. C'è un risparmio. Nel costruire una macchina, esistono delle risorse da allocare o meno. Si può dire cioè che mettere un prezzo su un'autovettura sia qualcosa di positivo. Previene la gente dall'usare molte risorse sprecate per costruire macchine che non sono veramente necessarie. Ma se ogni ulteriore macchina non richiedesse alcuna risorsa, non ci sarebbe nulla di positivo nel risparmiarne la

costruzione. Per gli oggetti materiali, come le autovetture, bisognerà sempre usare delle risorse per costruirne un altro, per ogni esemplare aggiuntivo.

Ma ciò non vale per il software. Chiunque può farne una copia. Ed è talmente semplice farlo. Non occorre alcuna risorsa, eccetto un minimo di elettricità. Non c'è nulla che possiamo risparmiare, nessuna risorsa da allocare in maniera migliore imponendo un disincentivo economico sull'uso del software. Spesso si trova gente che prende le conseguenze di un ragionamento economico, basato su premesse inapplicabili al software, cercando di trapiantarle da altri ambiti della vita in cui si applicano quelle premesse, e le conclusioni possono essere valide. Non fanno altro che prendere tali conclusioni e ne assumono la validità anche per il software, quando invece l'argomento è basato sul nulla, nel caso del software. In tal caso le premesse non funzionano. È molto importante esaminare il modo in cui si raggiungono le conclusioni, e su quali premesse siano basate, per vedere se possano essere considerate valide o meno. Così, questa è la libertà due, la libertà di aiutare il vicino.

La libertà tre è la libertà di aiutare a costruire la propria comunità tramite la pubblicazione di una versione migliorata del software. La gente mi diceva di solito, "Se il software è libero, allora nessuno verrà pagato per lavorarci sopra, perché mai qualcuno dovrebbe farlo?" Ovviamente facevano confusione tra i due diversi significati di 'free', tale ragionamento era basato su un'incomprensione di partenza. Ma in ogni caso questa era la loro teoria. Oggi possiamo confrontare tale teoria con dei fatti empirici, per constatare che centinaia di persone sono pagate per scrivere software libero, e oltre 100.000 lo fanno come volontari. Abbiamo un sacco di gente che lavora sul software libero, per motivi diversi.

Quando diffusi per la prima volta GNU Emacs – il primo pezzo

del sistema GNU che la gente prese davvero ad usare – e quando iniziò ad avere degli utenti, dopo qualche tempo, ricevetti un messaggio che diceva, “Credo di aver visto un bug nel codice sorgente, ed ecco qui come sistemarlo.” E poi arrivò un altro messaggio, “Questo è il codice per aggiungere una nuova funzione.” E poi un altro bug, e un’altra funzione nuova. E così via finché ne fui sommerso in maniera talmente rapida che stava diventando un lavoro enorme soltanto riuscire a usare tutto quest’aiuto. Microsoft non ha di questi problemi. [il pubblico ride]

Alla fine, il fenomeno attirò l’attenzione generale. Negli anni ‘80 parecchi di noi ritenevano che forse il software libero non sarebbe stato così efficace come il software non libero, perché non avremmo avuto denaro sufficiente per pagare la gente coinvolta. E naturalmente persone come il sottoscritto, a cui stanno a cuore la libertà e la comunità, dicevano: “Be’, useremo comunque il software libero.” Vale la pena di fare qualche sacrificio riguardo la mera convenienza tecnica in cambio della libertà. Ma quel che la gente iniziò a notare, verso il 1990, fu che il nostro software in realtà era migliore. Era più potente e più affidabile delle alternative proprietarie.

All’inizio degli anni ‘90 qualcuno trovò il modo di misurare scientificamente l’affidabilità del software. Ecco cosa fece. Prese alcune serie diverse di programmi comparabili che eseguivano lavori analoghi – gli stessi identici lavori – su sistemi differenti. Ciò grazie all’esistenza di certe utilità di base simili a Unix. E i lavori eseguiti erano più o meno gli stessi – ovvero, seguivano le specifiche POSIX – in modo da essere identici in termini di risultati ottenuti; ma erano mantenuti da persone diverse, e scritti separatamente tra loro. Il codice era diverso. Così dissero, bene, prenderemo questi programmi e li faremo girare con dati a caso, e misu-

reremo quanto spesso si bloccano o funzionano male. Lo misurarono, e la serie di programmi più affidabile fu quella GNU. Tutte le alternative commerciali, che erano software proprietario, risultarono meno affidabili. Così l'autore della ricerca la pubblicò e ne informò tutti gli sviluppatori. Qualche anno dopo ripeté lo stesso esperimento con versioni più recenti, e ottenne il medesimo risultato. Le versioni GNU si dimostrarono più affidabili. Sapete, ci sono cliniche per il cancro e i servizi del 911⁶ che usano il sistema GNU, perché è così affidabile, e per loro l'affidabilità è decisamente importante.

Comunque, c'è perfino un gruppo di persone che usa questo particolare vantaggio come motivazione principale nel consentire agli utenti di fare queste varie cose, di avere queste libertà. Se mi avete ascoltato, avrete notato come, parlando per il movimento del software libero, parli di questioni etiche, e di quale tipo di società vogliamo avere, su cos'è che rende positiva una società, come pure sui vantaggi pratici, materiali. Sono entrambi questioni importanti. Questo è il movimento del software libero.

L'altro gruppo di persone – definito il *movimento open source* – cita soltanto tali vantaggi pratici. Negano che si tratti di una questione di principio. Negano il riconoscimento alla libertà di condividere con il proprio vicino e di vedere come funziona il programma e di modificarlo se non ci piace. Sostengono tuttavia l'utilità di consentire alla gente di fare tutto ciò. Così vanno dalle varie aziende e dicono loro, "Potete fare più soldi se consentite alla gente di fare ciò." Quel che possiamo vedere, in un certo senso, è che spingono verso una direzione simile, ma per ragioni filosofiche fondamentalmente, totalmente diverse.

⁶ In molte aree degli Stati Uniti il 911 è il numero telefonico per le chiamate d'emergenza.

Sulla questione più profonda di tutte, quella etica, i due movimenti sono in disaccordo. Nel movimento del software libero diciamo, “L’utente ha diritto a queste libertà. Non dovrebbero impedirgli di fare queste cose.” Nel movimento open source dico, “Sì, possono impedirglielo se vogliono, ma cercheremo di convincerli a permettere all’utente di fare queste cose.” Be’, hanno dato un contributo – hanno convinto un certo numero di aziende a diffondere parti sostanziali di software come software libero nella nostra comunità. Il movimento open source ha contribuito in maniera sostanziale alla comunità, e lavoriamo insieme [a loro] su progetti pratici. Ma a livello filosofico esiste un disaccordo terribile.

Purtroppo il movimento open source è quello che ottiene il sostegno della maggioranza dell’imprenditoria, e quindi la gran parte degli articoli sul nostro lavoro ci descrivono come open source, e parecchia gente ritiene parimenti in buona fede che facciamo tutti parte del movimento open source. Ecco perché insisto con questa distinzione. Voglio farvi notare che movimento del software libero, che ha creato questa comunità e ha sviluppato il sistema operativo libero, è ancora qui – e che continueremo a batterci per questa filosofia etica. Voglio farvelo sapere, in modo che non possiate informare erroneamente qualcun altro.

Ma anche perché così potete riflettere sulle vostre stesse posizioni. Spetta a ciascuno di voi decidere quale movimento sostenere. Potete dichiararvi d’accordo con il movimento del software libero e con le mie osservazioni. Potete dichiararvi d’accordo con il movimento open source. Potete dichiararvi in disaccordo con entrambi i movimenti. Tocca a voi decidere da che parte stare su queste faccende politiche.

Ma nel caso siate d’accordo con il movimento del software libero

– se credete esista la questione della gente la cui vita è controllata e diretta da simili decisioni, e volete dire la vostra al riguardo – spero allora che vorrete dichiararvi d'accordo con il movimento del software libero, e un modo per dimostrarlo è usare il termine software libero (“free software”) e aiutare a informare gli altri della nostra esistenza.

Allora, la libertà tre è molto importante a livello sia pratico sia psico-sociale. La mancanza di tale libertà provoca danno pratico e materiale, perché questa comunità di sviluppo non esisterebbe, e non potremmo avere software potente e affidabile. Ma ciò causa anche danno psico-sociale perché colpisce lo spirito della cooperazione scientifica – l'idea che stiamo lavorando insieme per l'avanzamento della conoscenza umana. Il progresso scientifico dipende in modo cruciale dalla capacità di lavorare insieme. Oggi-giorno, tuttavia, ci si imbatte spesso in piccoli gruppi di ricercatori che operano come fossero in guerra con ogni altra banda di ricercatori e ingegneri. Ma se non condividono i dati a vicenda, rimangono tutti bloccati.

Queste sono dunque le tre libertà che distinguono il software libero dal software comune. La libertà uno è la libertà di aiutare se stessi, operando i cambiamenti necessari alle proprie esigenze. La libertà due è la libertà di aiutare il proprio vicino distribuendo delle copie. E la libertà tre è la libertà di aiutare a costruire la propria comunità apportando delle modifiche e pubblicandole per l'uso altrui. Se esistono tutte queste libertà, il programma è software libero per chi lo usa. Ora, perché lo definisco in tal modo rispetto a un utente specifico? Si tratta forse di software libero per te? [indicando qualcuno tra il pubblico] È software libero per te? [indicando qualcun altro tra il pubblico] Oppure per te? [indicando una terza persona tra il pubblico] Sì?

Domanda: Puoi illustrare meglio la differenza tra la libertà due e la tre?

RMS: Be', sono certamente correlate, perché non avendo affatto la libertà di redistribuzione, sicuramente non si potrà avere quella di ridistribuirne una versione modificata, ma si tratta di attività diverse.

La libertà due è: l'utente ne fa una copia esatta, e la passa agli amici, in modo che anche costoro possano farne uso. O forse ne fa delle copie identiche e le vende a un po' di persone, e questi possono farne uso.

La libertà tre riguarda i miglioramenti apportati dall'utente – o almeno, quelli che quest'ultimo ritiene tali e su cui anche altri si dichiarano d'accordo. Questa è la differenza. Ah, c'è anche un altro punto cruciale. Le libertà uno e tre dipendono dall'accesso al codice sorgente. Perché modificare un programma solo binario è estremamente difficile [il pubblico ride] – perfino cambiamenti stupidi come le quattro cifre per la data⁷ – se non si hanno i sorgenti. Così per motivi impellenti, pratici, l'accesso al codice sorgente è una pre-condizione, un requisito, per il software libero.

Perché allora definisco tale il software libero rispetto a un utente specifico? Il motivo è che talvolta uno stesso programma può essere software libero per qualcuno, e non libero per altri. Siccome questa potrebbe sembrare una situazione limite, vi farò un esempio così da chiarirne meglio le circostanze. Un grande esempio – forse il più grande possibile – di questo problema fu il sistema X Windows, sviluppato al MIT e diffuso sotto una licenza che lo

⁷ Ci si riferisce qui al problema "Y2K", dove molti vecchi programmi riportavano l'anno in due cifre; non era quindi chiaro se la data "00" indicasse il 2000 o il 1900, o qualsiasi altro anno che finiva in 00. Svaretiati milioni di dollari vennero spesi per riparare il problema in migliaia di sistemi informatici prima dell'anno 2000.

rese software libero. La versione con la licenza del MIT offriva all'utente le libertà uno, due e tre. Per chi lo usava era software libero. Ma tra quanti ne ottennero delle copie c'erano alcuni produttori che distribuivano sistemi Unix, e vi apportarono le modifiche necessarie per farlo girare sui quei sistemi. Probabilmente tali modifiche interessavano appena qualche migliaio di righe su un totale di centinaia di migliaia di righe. Così lo compilarono, misero i file binari nel sistema Unix e lo distribuirono sotto gli stessi accordi di non divulgazione che coprono il resto del sistema Unix. E allora milioni di persone ottennero queste copie. Avevano il sistema X Windows, ma nessuna di queste libertà. Per quegli utenti non era software libero.

Così, il paradosso fu che X era software libero a seconda da dove lo si misurava. Misurandolo tra il gruppo di sviluppatori, si diceva, "Io rispetto tutte queste libertà. È software libero." Misurandolo tra gli utenti, si diceva "Hmmm, la maggioranza degli utenti non ha queste libertà. Non è software libero." Per gli sviluppatori di X ciò non rappresentava un problema, poiché il loro obiettivo era semplicemente la popolarità – ego, sostanzialmente. Miravano a un grande successo professionale. Volevano sentirsi tipo, "Ah, c'è un sacco di gente che usa il nostro software." Ed era vero. Molte persone usavano quel software ma non avevano libertà. Be', nel progetto GNU, se qualcosa di simile fosse accaduto al software GNU, sarebbe stato un fallimento, perché il nostro obiettivo non era soltanto diventare popolari; lo scopo era dare libertà alla gente, e incoraggiare la cooperazione, consentire alla gente di cooperare. Ricordiamolo, non bisogna costringere mai nessuno a cooperare con qualcun altro, ma occorre assicurarsi che a chiunque sia consentito cooperare, che tutti abbiano la libertà di farlo, se così decidono. Se milioni di persone avessero usato le versioni

non libere di GNU, ciò non sarebbe stato affatto un successo. L'intera situazione sarebbe scaduta in qualcosa di ben lontano dalla meta.

Così mi misi alla ricerca di un modo per impedire ciò. Il metodo che misi a punto è definito "copyleft". Viene chiamato copyleft (permesso d'autore) perché è qualcosa di simile a prendere il copyright (diritto d'autore) e rigirarlo sottosopra. [il pubblico ride] A livello legale, il copyleft funziona sulla base del copyright. Usiamo le attuali leggi sul copyright ma in modo tale da raggiungere un obiettivo molto diverso. Ecco cosa facciamo. Diciamo, "Questo programma è sotto copyright." E naturalmente ciò significa per definizione che ne è vietata la copia, la distribuzione o la modifica. Ma poi diciamo, "L'utente è autorizzato a distribuirne delle copie. L'utente è autorizzato a modificarlo. L'utente è autorizzato a distribuirne versioni modificate e versioni ampliate. Può modificarlo in qualsiasi modo voglia".

Esiste però una condizione. E la condizione, ovviamente, è il motivo per cui ci siamo dati la pena di fare tutto ciò, in modo da poter aggiungere tale condizione. La condizione recita: Ogni volta che l'utente distribuisce qualunque cosa che contenga una parte qualsiasi di questo programma, quell'intero programma deve essere distribuito sotto gli stessi termini, niente di più e niente di meno. L'utente può cambiare il programma e distribuirne una versione modificata, ma quando lo fa, a quanti lo ricevono dall'utente spetta la medesima libertà che abbiamo dato a tale utente. E non soltanto per le parti che quest'ultimo ha copiato dal nostro programma, ma anche per le altre parti di tale programma che gli altri hanno ricevuto dallo stesso utente. Per coloro che lo ricevono, il programma nella sua interezza deve essere libero.

Le libertà di cambiare e ridistribuire questo programma diventano

diritti inalienabili – un concetto derivato dalla Dichiarazione d'Indipendenza statunitense. Diritti che vogliamo esser certi non vengano portati via all'utente. La licenza specifica che dà corpo all'idea di copyleft è la Licenza Pubblica Generica GNU (GNU GPL), una licenza controversa perché in realtà possiede la forza di dire no a coloro che vorrebbero fare i parassiti della nostra comunità.

Ci sono molte persone che non apprezzano gli ideali di libertà. E sarebbero molto contente di prendere il lavoro che abbiamo fatto per usarlo come spinta d'avvio nella distribuzione di un programma non libero e nell'allettare la gente a rinunciare alle proprie libertà. Il risultato sarebbe – se consentiamo alla gente di farlo – che dopo aver sviluppato questi programmi liberi, dovremmo costantemente competere con le versioni migliorate dei nostri stessi programmi. Nient'affatto divertente.

Molta gente chiede anche, "Vorrei offrire il mio tempo per contribuire senza compenso alla comunità, ma perché dovrei offrire volontariamente il mio tempo per contribuire a migliorare il programma proprietario di quell'azienda?" Qualcuno potrebbe perfino non ritenere che ciò sia un male, ma vogliono essere pagati per un simile lavoro. Personalmente, preferisco piuttosto non farlo affatto.

Tuttavia entrambi questi gruppi di persone – quelli come me che dicono, "Non voglio aiutare i programmi non liberi a mettere un piede nella nostra comunità", e quelli che sostengono, "Certo, sono disposto a lavorare per loro, ma voglio essere pagato" – hanno una buona ragione per usare la Licenza Pubblica Generica GNU. Perché questa dice a ogni azienda, "Non puoi soltanto appropriarti del mio lavoro, e distribuirlo senza la libertà." Mentre ciò viene consentito dalle licenze non copyleft, come quella per X Windows. Questa dunque è la grande divisione tra le due categorie di software libero, a livello di licenza. Ci sono programmi sotto copyleft in

modo che la licenza tuteli la libertà del software per ciascun utente. E ci sono i programmi non coperti da copyleft per i quali sono permesse versioni non libere. Qualcuno può prendere questi ultimi programmi e strapparne via la libertà. L'utente può ricevere quel programma in una versione non libera.

Questo problema esiste ancor oggi. Ci sono ancora versioni non libere di X Windows usate sui nostri sistemi operativi liberi. C'è perfino dell'hardware che non è veramente supportato eccetto che da una versione non libera di X Windows. E questo è un problema importante per la nostra comunità. Ciò nonostante non definirei negativamente X Windows. Direi che gli sviluppatori non operarono al meglio. Ma diffusero comunque parecchio software che tutti noi possiamo usare.

C'è molta differenza tra meno che perfetto e negativo. Esistono molte gradazioni tra bene e male. Dobbiamo resistere alla tentazione di dire, se qualcuno non ha fatto la cosa assolutamente migliore possibile, che è tutto negativo. Le persone che hanno sviluppato X Windows hanno fornito un grande contributo alla comunità. Ma avrebbero potuto fare qualcosa di meglio. Avrebbero potuto mettere sotto copyleft alcune parti del programma onde prevenire l'ulteriore distribuzione di quelle versioni che negavano la libertà.

Ora, il fatto che la Licenza Pubblica Generica GNU difenda la libertà dell'utente, ricorrendo alle leggi sul copyright per tutelare la libertà dell'utente, è naturalmente il motivo per cui oggi Microsoft ci attacca. Microsoft vorrebbe davvero riuscire a prendere tutto il codice che scriviamo e metterlo in programmi proprietari, qualcuno poi gli apporgerà qualche miglioria... o forse tutto ciò di cui hanno bisogno è soltanto qualche modifica incompatibile [il pubblico ride].

Con la potenza del marketing, Microsoft non deve migliorarlo per far sì che la propria versione soppianti la nostra. Gli basta renderlo diverso e incompatibile. E poi metterlo sulla scrivania di tutti gli utenti. Perciò non gradiscono affatto la GNU GPL. Perché questa non consente loro di fare così. La GNU GPL non permette di “abbracciare ed estendere.” Dice, se l’utente vuole condividere il codice dei suoi programmi, può farlo. Ma deve condividere, e condividere allo stesso modo. Ogni cambiamento fatto dall’utente dev’essere parimenti condiviso da tutti noi. Si tratta cioè di una cooperazione a due canali, che è poi la vera cooperazione.

Molte aziende – anche quelle grandi tipo IBM e Hewlett-Packard – hanno deciso di usare il nostro software su queste basi. IBM e Hewlett-Packard hanno offerto miglioramenti sostanziali al software GNU. E sviluppano altro software libero. Ma Microsoft non vuole farlo, rinunciando così a ogni attività commerciale che abbia a che fare con la GPL. Be’, se tale attività non includesse IBM, Hewlett-Packard e Sun, allora forse avrebbero ragione [il pubblico ride]. Ne parleremo meglio più avanti.

Vorrei concludere l’aspetto storico. Siamo partiti nel 1984 non soltanto per scrivere software libero ma anche per fare qualcosa di molto più coerente: sviluppare un sistema operativo che fosse interamente software libero. Ciò significava che avremmo dovuto scriverne un pezzo dopo l’altro. Naturalmente eravamo sempre alla ricerca di scorciatoie. Un compito talmente vasto che la gente sosteneva che non saremmo mai stati capaci di portarlo a termine. Io ritenevo che avevamo almeno una possibilità di completarlo, ma ovviamente valeva la pena di cercare delle scorciatoie. Così continuavamo a guardarci intorno. Esistevano programmi scritti da altri che avremmo potuto maneggiare in modo da adattarli, da inserirli qui dentro, e in tal modo evitare di doverli scrivere partendo da

zero? Ad esempio, il sistema X Windows. È vero che non era sotto copyleft, ma era software libero, quindi potevamo usarlo.

Ora, fin dal primo giorno avrei voluto mettere un sistema a finestre all'interno di GNU. Avevo scritto un paio di sistemi a finestra al MIT, prima di iniziare GNU. E così pur se nel 1984 Unix non aveva un sistema a finestre, decisi che GNU l'avrebbe avuto. Ma non arrivammo mai a scrivere un sistema a finestre GNU perché arrivò X. E allora feci: "Ottimo! Un grosso lavoro che non dobbiamo fare. Useremo X." Dissi, prendiamo X e inseriamolo nel sistema GNU. E faremo in modo che le altre parti di GNU operino bene con X, quando necessario. Poi trovammo nuovi pezzi di software scritti da altri, come il formattatore di testi TeX, e qualche codice di libreria da Berkeley. A quei tempi esisteva Berkeley Unix, ma non era software libero. Inizialmente questo codice di libreria veniva da un gruppo diverso a Berkeley, che faceva ricerche sul punto di fluttuazione. Così mettemmo insieme questi pezzi. Nell'ottobre 1985 fondammo la Free Software Foundation. È il caso di notare che fu il progetto GNU a venire prima. La Free Software Foundation arrivò almeno due anni dopo l'annuncio del progetto GNU. La Free Software Foundation è un ente senza fini di lucro che raccoglie fondi per promuovere la libertà di condividere e modificare il software. Negli anni '80 una delle cose più importanti che facemmo con i fondi raccolti fu assumere gente per scrivere parti di GNU. E programmi essenziali, quali la shell e la libreria C, vennero scritti in questo modo, come pure parti di altri programmi. Il programma tar, che è assolutamente essenziale, pur se nient'affatto eccitante [il pubblico ride], è stato scritto in questo modo. Credo lo stesso valga anche per il grep GNU. E così ci avvicinavamo alla meta prefissata.

Nel 1991 mancava soltanto un pezzo importante, si trattava del

kernel. Ora, perché lasciai da parte il kernel? Probabilmente perché non importa granché in quale ordine si integrano i vari pezzi, almeno a livello tecnico non importa. Devi comunque farli tutti. E in parte perché speravo che saremmo partiti da un kernel trovato da qualche parte. E così fu. Trovammo Mach, che era stato sviluppato alla Carnegie Mellon. Non era il kernel intero, solo la metà inferiore. Dovevamo perciò scrivere quella superiore; cose tipo il file system, il codice di rete, e così via. Ma quelli che girano su Mach sono essenzialmente programmi per utenti, i cui bug devono risultare facili da sistemare. Il debug si può fare contemporaneamente con un vero debugger a livello di codice. Credevo che in tal modo avremmo potuto fare in tempi brevi le parti di alto livello del kernel. Non andò così. Questi processi asincroni e plurilivelli, l'invio di messaggi tra di noi, si dimostrarono molto difficili per il debug. E il sistema basato sul Mach che usavamo aveva un terribile ambiente per il debug, ed era inaffidabile. Impiegammo anni e anni per far funzionare il kernel GNU.

Ma fortunatamente la comunità non doveva aspettare il kernel GNU. Perché nel 1991 Linus Torvalds sviluppò un altro kernel libero, chiamato Linux. Usò il buon vecchio progetto monolitico e accadde che il suo funzionò in tempi molto più rapidi del nostro. Forse questo fu uno degli errori che commisi: la decisione progettuale. Comunque, all'inizio non sapevamo nulla di Linux, perché Torvalds non ci contattò mai per parlarne, pur sapendo dell'esistenza del progetto GNU.

Ma l'annunciò ad altre persone e altrove in rete. E poi qualcun altro si occupò di integrare Linux con il resto del sistema GNU, arrivando a completare il sistema operativo libero. Sostanzialmente, realizzando la combinazione GNU più Linux.

Ma non avevano capito che era questo che stavano facendo. Dice-

vano, “Abbiamo un kernel – guardiamoci intorno per vedere quali altri pezzi possiamo trovare da mettere insieme al kernel.” Così si guardarono intorno, ed ecco che tutto ciò di cui avevano bisogno era già disponibile. “Che bella fortuna,” andavano dicendo. [il pubblico ride] “È tutto bell’e pronto. Possiamo trovare tutto quanto ci occorre. Basta prendere tutte queste cose diverse e metterle insieme per avere un sistema.”

Non sapevano che la maggior parte di quanto trovarono erano pezzi del sistema GNU. Così non si accorsero che stavano integrando Linux nel varco del sistema GNU. Credevano di star costruendo un sistema derivante direttamente da Linux. Perciò lo chiamarono Linux. [Qualcuno tra il pubblico chiede] “Ma si tratta forse di maggior fortuna che trovare il sistema X Windows e Mach?” [Stallman replica e poi prosegue] Giusto. La differenza sta nel fatto che gli sviluppatori di X e Mach non avevano l’obiettivo di fare un sistema operativo libero completo. Eravamo soltanto noi a volerlo fare. E il sistema esisteva grazie al nostro tremendo lavoro. In realtà noi facemmo la parte più ampia del sistema rispetto a qualsiasi altro progetto. Non si trattò di una coincidenza, perché quella gente scrisse utili parti del sistema. Ma non lo fecero perché volevano completare il sistema. Avevano altri motivi.

Quanti svilupparono X ritenevano che realizzare un sistema a finestre tramite la rete fosse un buon progetto, e lo era. E accadde che ci fu d’aiuto nel realizzare un buon sistema operativo libero. Ma non è quanto speravano quegli sviluppatori. Non ci pensavano neppure. Fu un incidente. Un incidente benefico. Non sto sostenendo che quanto fecero fosse mal fatto. Portarono a termine un ampio progetto di software libero. Un’ottima cosa da fare. Ma non avevano una visione complessiva. Fu il progetto GNU a incarnare tale visione. E così noi riuscimmo a fare ogni piccolo pezzo che non aveva fat-

to nessun altro. Perché sapevamo che senza questi pezzi non avremmo avuto un sistema completo. E pur essendo totalmente noioso e poco romantico, come tar o mv⁸ [il pubblico ride], lo facemmo.

Oppure ld – sapete, non c'è nulla di eccitante in ld, ma io ne ho scritto uno [il pubblico ride]. E mi sono sforzato per far sì che occupasse una quantità minima di disco I/O in modo da poter essere più veloce e gestire programmi più ampi. Mi piace fare un buon lavoro; mi piace migliorare varie cose nel programma mentre ci lavoro su. Ma la ragione per cui lo feci non era che avessi delle idee per un ld migliore. Il motivo per cui lo feci era che avevamo bisogno di una versione che fosse libera. E non potevamo aspettarci che lo facesse qualcun altro. Dovevamo farlo noi, o trovare qualcuno che lo facesse. Così, nonostante a questo punto sono migliaia le persone e i progetti che hanno contribuito a questo sistema, c'è un progetto alla base dell'esistenza stessa di tale sistema, ed è il progetto GNU. Sostanzialmente si tratta del sistema GNU con altre cose aggiunte in seguito.

La pratica di chiamare il sistema Linux è stato un grave colpo per il progetto GNU, perché normalmente non si riconosce a qualcuno quel che non ha fatto. Credo che Linux, il kernel, sia un pezzo di software libero assai utile, e ho soltanto cose buone da dire al riguardo. In realtà, avrei un paio di cattiverie da dire su Linux. [il pubblico ride] Ma in sostanza ho apprezzamenti positivi. Tuttavia, la pratica di definire il sistema GNU con “Linux” è proprio un errore. Vorrei pregarvi di compiere il piccolo sforzo necessario per chiamare il sistema GNU/Linux, aiutandoci così ad ottenere parte del riconoscimento.

[Qualcuno tra il pubblico urla] “Vi serve una mascotte! Trova un

⁸ *Un semplice programma per spostare o rinominare i file.*

animale di peluche!” [Stallman risponde] Ne abbiamo uno. [La persona replica] “E qual’è?” [Stallman risponde, provocando grandi risate] Abbiamo un animale, lo gnu. Giusto, quando disegnatte un pinguino metteteci anche uno gnu di fianco. Ma conserviamo le domande per la fine. Devo parlare di altre cose.

Perché dunque mi preoccupo di questa faccenda? Perché credo valga la pena di disturbarvi e forse farvi una cattiva impressione [il pubblico ride] sollevando la questione del riconoscimento? Quando lo faccio, qualcuno ritiene che è per soddisfare il mio ego, vero? Naturalmente non vi sto chiedendo di chiamarlo “Stallmanix”, giusto? [il pubblico ride] [applausi]

Vi chiedo di chiamarlo GNU perché voglio che il progetto GNU ottenga il giusto riconoscimento. E c’è un motivo specifico per questo, che è molto più importante del riconoscimento per chicchessia, in e per se stesso. Oggigiorno guardandosi intorno nella nostra comunità la maggior parte della gente ne parla e ne scrive senza mai menzionare GNU, e non citano mai neppure questi obiettivi di libertà – questi concetti politici e sociali. Perché il luogo da cui arrivano tali concetti è GNU. Le idee associate con Linux, la loro filosofia è molto diversa. Praticamente è la filosofia apolitica di Linus Torvalds. Così, quando si ritiene che l’intero sistema sia Linux, si tende a pensare: “Oh, tutto dev’essere stato iniziato da Linus Torvalds. La sua filosofia è quella che dovremmo considerare con attenzione.” E quando sentono parlare della filosofia GNU, dicono: “Ma ciò è così idealista, dev’essere terribilmente impraticabile. Io sono un utente Linux, non un utente GNU.” [il pubblico ride]

Com’è ironico! Se soltanto sapessero! Se sapessero che il sistema che piace loro – o, in alcuni casi, che amano e di cui vanno pazzi – è la nostra filosofia idealista, politica resa in concreto.

Non sarebbero comunque d'accordo con noi. Ma almeno vedrebbero un motivo per prenderla sul serio, per rifletterci con attenzione, per darle una possibilità. Vedrebbero come ciò sia in relazione con la propria vita. Se si rendessero conto, "Uso il sistema GNU. Ecco la filosofia GNU. Questa filosofia è la ragione per l'esistenza stessa del sistema che mi piace così tanto," quantomeno lo considererebbero con mente più aperta. Ciò non significa che sarebbero tutti d'accordo. La gente pensa in modo diverso. Va bene così – ciascuno dovrebbe ragionare con la propria testa. Ma voglio che questa filosofia ottenga il beneficio del riconoscimento per i risultati raggiunti.

Guardandosi intorno nella comunità, si noterà che quasi ovunque le istituzioni definiscono il sistema Linux. Per lo più i giornalisti lo chiamano Linux. Non è corretto, ma lo fanno. Lo stesso dicasi per le varie aziende che impacchettano il sistema. E gran parte di quei giornalisti, quando scrivono degli articoli, generalmente non la considerano una questione politica o sociale. Normalmente parlano di una faccenda puramente commerciale o di quali aziende avranno più o meno successo, che in realtà è una questione decisamente minore per la società. E se consideriamo le aziende che impacchettano il sistema GNU/Linux per l'utenza, be', la maggioranza lo chiama Linux. E tutte vi aggiungono software non libero.

La GNU GPL dice che se si prende del codice, e si tratta di codice da un programma coperto dalla GPL, e vi si aggiunge altro codice per fare un programma più ampio, quest'ultimo va diffuso nella sua interezza sotto la GPL. Ma si possono mettere altri programmi separati su uno stesso disco (hard disk o CD) e questi possono avere altre licenze. Ciò viene considerata una mera aggregazione e, sostanzialmente, la semplice distribuzione contemporanea di due programmi è qualcosa su cui non abbiamo nulla da

dire. Perciò non è vero – talvolta vorrei lo fosse – che se un’azienda usa un programma coperto dalla GPL all’interno di un prodotto questo debba essere completamente software libero. Non è così, non raggiunge quest’ampiezza, questo scopo. Si tratta dell’intero programma. Se ci sono due programmi separati che comunicano tra loro a distanza ravvicinata, tipo inviandosi messaggi a vicenda, allora sono legalmente separati, in generale. Queste aziende, quando aggiungono al sistema software non libero, danno agli utenti un’idea molto sbagliata, a livello filosofico e politico. Dicono loro, “È bene usare software non libero. Lo mettiamo perfino qui in aggiunta.”

Se diamo un’occhiata alle riviste sull’uso del sistema GNU/Linux, in maggioranza hanno titoli quali “Linux qualcosa-o-qualcos’altro”. La maggior parte delle volte definiscono il sistema Linux. E sono zeppe di annunci di software non libero che si può far girare sul sistema GNU/Linux. Questi annunci suggeriscono un messaggio comune. Dicono: “Il software non libero è un bene per l’utente. A tal punto che puoi persino pagare per averlo.” [il pubblico ride]

E li chiamano “pacchetti a valore aggiunto”, il che ne rivela i valori. Questa gente dice: Date valore alla convenienza pratica, non alla libertà. Io non sono d’accordo con questi valori, per cui li chiamo “pacchetti a libertà sottratta.” [il pubblico ride] Perché, avendo installato un sistema operativo libero, allora si vive nel mondo libero. Si apprezzano i benefici della libertà su cui abbiamo lavorato per anni onde garantirli all’utente. Quei pacchetti offrono invece la possibilità di legarsi a una catena.

Se poi consideriamo le mostre specializzate dedicate all’uso del sistema GNU/Linux, le chiamano tutte “Linux”. E sono piene di stand che espongono software non libero, sostanzialmente ponendo

do il timbro d'approvazione sul software non libero. Così, quasi ovunque si guardi nella comunità, le istituzioni appoggiano il software non libero, negando completamente quell'idea di libertà per cui fu sviluppato GNU. L'unico ambito in cui ci si imbatte con l'idea di libertà è in connessione con GNU, e in connessione con il software libero, il termine software libero. Ecco perché vi chiedo: Per favore chiamate il sistema GNU/Linux. Per favore fate sapere alla gente da dove arriva e perché esiste quel sistema.

Naturalmente, il solo uso di quel nome non significa offrirne una spiegazione storica. Si possono inserire altri quattro caratteri e scrivere GNU/Linux; si possono dire due sillabe aggiuntive. Ma GNU/Linux contiene meno sillabe di Windows 2000 [il pubblico ride]. Non si spiega granché agli altri, ma li si prepara a saperne di più su cos'è GNU, e così vedranno come sia collegato a loro e alla propria vita. E ciò, indirettamente, può fare una differenza tremenda. Per favore dateci una mano.

Avrete notato come Microsoft abbia definito la GPL una "licenza open source." Non vogliono che la gente consideri la questione in termini di libertà. Si scopre che invitano la gente a pensare in maniera ristretta, in quanto consumatori, e naturalmente a pensare neppure molto razionalmente in quanto tali, se si apprestano a scegliere prodotti Microsoft. Ma non vogliono che la gente pensi in quanto cittadini o uomini di stato. Ciò risulterebbe avverso per loro. Almeno è avverso al loro attuale modello commerciale. Ora, come fa il software libero... be', potrei illustrarvi la relazione tra software libero e società. Un argomento secondario che potrebbe interessare alcuni di voi è la relazione tra software libero e imprenditoria. In effetti, il software è tremendamente utile per l'imprenditoria. Dopo tutto, nei paesi avanzati la maggioranza delle aziende fa uso di software. Soltanto una minima parte lo sviluppa.

E il software libero è *tremendamente* vantaggioso per qualsiasi azienda che fa uso di software, perché significa esserne in controllo. Praticamente software libero significa che gli utenti sono in controllo di quello che fa il programma. Sia individualmente, se si vuole esserlo, sia collettivamente, quando si vuole esserlo. Chiunque abbia motivo sufficiente per farlo, può esercitare qualche influenza. Se non t'importa granché, non lo compri. Allora usi quel che preferiscono gli altri. Ma se la cosa ti preme, allora puoi dire la tua. Con il software proprietario, sostanzialmente non hai voce in capitolo.

Con il software libero l'utente può modificare quel che vuole. E non importa se in azienda non ci sono programmatori, va bene così. Se si vogliono spostare le pareti di casa, non occorre essere una ditta di costruzioni. Basta trovarne una e chiedere, "Quanto volete per fare questo lavoro?" Volendo cambiare qualcosa nel software che si usa, non occorre essere un'azienda di programmazione. Basta trovarne una e chiedere "Quanto volete per implementare queste funzioni? E quando potreste occuparvene?" E se non possono farlo, si cerca qualcun altro.

Per l'assistenza c'è il mercato libero. Perciò ogni azienda interessata al supporto troverà un vantaggio tremendo nel software libero. Con il software proprietario il supporto è un monopolio, perché è una sola azienda ad avere il codice sorgente – o forse un piccolo numero di aziende che pagano un'esorbitante somma di denaro per avere il codice sorgente, se si tratta dei sorgenti condivisi di un programma Microsoft – ma sono ben poche. Non esistono molte possibili fonti di assistenza per l'utente. E ciò significa che, a meno che non si tratti un vero gigante industriale, non si curano di quell'utente. L'azienda di costui non è abbastanza importante perché possano preoccuparsi delle sue perdite com-

merciali. Una volta che l'utente usa il programma, lo ritengono incastrato nel dover chiedere loro il supporto, perché passare a un programma diverso è un lavoro gigantesco. Così si finisce con cose tipo pagare per il privilegio di segnalare un bug. E dopo aver pagato, dicono all'utente, "Bene, abbiamo preso nota della segnalazione. Nel giro di qualche mese potrai comprare l'aggiornamento e vedrai se l'abbiamo sistemato o meno." [il pubblico ride]

I fornitori di assistenza per il software libero non possono cavarcela così facilmente. Devono mostrarsi gentili con il cliente. Ovviamente si può anche ottenere un sacco di supporto gratis. Basta spiegare il problema in un messaggio su Internet. Può darsi che il giorno seguente si ottenga una risposta. Ma naturalmente non è affatto garantito. Per esserne certi, meglio mettersi d'accordo con un'azienda e pagarla. E questo è, naturalmente, uno dei modi in cui funziona l'attività commerciale legata al software libero.

Un altro vantaggio del software libero per l'imprenditoria che fa uso di software riguarda la sicurezza e la privacy. Vedete, quando un programma è proprietario, è impossibile perfino dire come opera in concreto.

Potrebbe avere delle funzioni inserite deliberatamente che non piacciono all'utente, qualora ne fosse informato. Potrebbe avere ad esempio una back door onde consentire allo sviluppatore di accedere alla macchina dell'utente. Potrebbe spiare quanto si sta facendo e ridistribuire queste informazioni. Non è insolito. Qualche software Microsoft lo faceva. Ma non è soltanto Microsoft. Esistono altri programmi proprietari che spiano l'utente. E quest'ultimo non può neppure dire se ciò avvenga o meno. Naturalmente anche assumendo la completa onestà dello sviluppatore, ogni programmatore compie degli errori. Potrebbero esserci dei bug relativi alla sicurezza di cui nessuno ha colpa. Ma il pun-

to è: se non è software libero, l'utente non può trovarli. E non può sistemarli.

Nessuno ha il tempo di verificare i sorgenti di ogni programma che usa. Non lo fa nessuno. Ma con il software libero esiste un'ampia comunità, e al suo interno c'è gente che controlla queste cose. Si può trarre vantaggio da quest'attività, perché se c'è un bug accidentale, ce ne sono di sicuro, di tanto in tanto, in ogni programma, qualcuno può trovarlo e sistemarlo. Ed è meno probabile che la gente ci metta dentro apposta un Trojan horse, o una funzione per spiare, se pensano di poter essere beccati. Gli sviluppatori di software proprietario fanno di non poter essere colti in flagrante. Possono farla franca senza che nessuno lo noti. Ma uno sviluppatore di software libero deve tener conto che la gente verificherà il codice e vedrà cosa c'è. Nella nostra comunità, sentiamo che è impossibile farla franca nel forzare una funzione nella gola degli utenti se questi non la gradiscono. Sappiamo che se agli utenti non piace, ne faranno una versione modificata priva di tale funzione. E poi prenderanno tutti ad usare questa versione.

Anzi, possiamo tutti ragionarci su, possiamo tutti prevedere ciò abbastanza in anticipo che probabilmente non inseriremo quella funzione. Dopo tutto, si tratta di scrivere un programma libero; voglio che alla gente piaccia la mia versione; non voglio metterci qualcosa che molte persone finiranno per odiare, e vedere un'altra versione modificata prendere il posto della mia. Così ci si rende conto che nel mondo del software libero l'utente è il re. Nel mondo del software proprietario, il cliente non è il re. Perché non è altro che un cliente. Non ha alcuna voce in capitolo nel software che usa.

In tal senso, il software libero è un nuovo meccanismo per far ope-

rare la democrazia. Il professor Lessig,⁹ ora a Stanford, ha fatto notare che il codice opera come una specie di legislazione.

Chiunque riesca a scrivere quel codice che viene usato praticamente da tutti per ogni scopo e intento, sta scrivendo le leggi che gestiscono la vita della gente. Con il software libero, tali leggi vengono scritte in maniera democratica. Non la forma classica di democrazia – non abbiamo grandi elezioni e cose tipo, “Tutti devono votare su come fare questa funzione” [il pubblico ride]. Invece diciamo, praticamente, quelli tra voi che vogliono lavorare all’implementazione della funzione in questo modo, procedano pure. E se qualcuno vuole lavorare all’implementazione della funzione in quell’altro modo, lo faccia. In un modo o nell’altro la cosa viene realizzata. E se parecchia gente la vuole in questo o quel modo, è così che viene fatta. Tutti contribuiscono alla decisione sociale compiendo semplicemente dei passi nella direzione che ciascuno preferisce.

E ognuno è libero di fare tanti passi quanti se ne vuole, a livello personale. Un’azienda è libera di commissionare a qualcuno i passi che ritiene utile compiere. E dopo aver aggiunto tutte queste cose, ciò indica la direzione in cui sta andando il software.

Spesso torna molto utile poter prendere dei pezzi da un programma già esistente – presumibilmente pezzi alquanto ampi, in genere – e poi scrivere in proprio una certa quantità di codice, onde avere un programma che fa esattamente quanto ci occorre, che costerebbe una gamba e un braccio da sviluppare nel caso bisognasse scriverlo da zero, se non fosse possibile cannibalizzare ampie porzioni da qualche pacchetto di software libero preesistente.

Un’altra cosa derivante dal fatto che l’utente è il re, è che si tende ad essere molto precisi rispetto a commutabilità e standardizza-

⁹ *Lawrence Lessig ha scritto l’introduzione al volume originale inglese.*

zione. Perché? Perché è quel che piace agli utenti. È probabile che gli utenti rifiutino un programma che presenti incompatibilità inutili. Talvolta c'è un determinato gruppo di utenti che ha davvero bisogno di un certo tipo di incompatibilità, e allora l'avranno. A posto così. Ma quando gli utenti vogliono aderire a uno standard, noi sviluppatori dobbiamo seguirlo, e lo sappiamo. Al contrario, se osserviamo gli sviluppatori di software proprietario, spesso trovano vantaggioso il fatto di non seguire deliberatamente uno standard, e non perché ritengano in tal modo di offrire un vantaggio all'utente, ma piuttosto perché così s'impongono all'utente, lo incastrano. E si scopre perfino che di tanto in tanto apportano modifiche al formato dei file, soltanto per costringere la gente a dotarsi della versione più recente.

Gli archivisti¹⁰ si stanno confrontando con un problema, che spesso non si può accedere ai file scritti su un computer di dieci anni fa; vennero scritti con software proprietario che sostanzialmente oggi è andato perduto. Se fossero stati scritti con software libero, li si potrebbe aggiornare e far girare. E quel materiale non sarebbe perduto, non sarebbe inaccessibile. Recentemente ci si lamentava di questa situazione anche sulla National Public Radio,¹¹ citando il software libero come soluzione. In effetti, usare un programma non libero per archiviare i propri dati è come nascondere la testa sottoterra.

Ho illustrato il modo in cui il software libero interessi la maggior parte dell'imprenditoria. Ma come influisce su quella particolare area ristretta che è l'attività commerciale del software stesso? Bene,

¹⁰ Numerosi archivisti conservano e condividono migliaia di file via Internet.

¹¹ La National Public Radio è un ente privato e non-profit che, all'epoca di questo intervento, conta 620 stazioni radio pubbliche che trasmettono quotidianamente notizie e musica.

la risposta è che per lo più non ha alcuna influenza. E la ragione è che il 90% dell'industria del software, da quanto mi si dice, riguarda lo sviluppo di software personalizzato, software che non è affatto destinato ad essere distribuito. Per il software personalizzato, questa questione, la questione etica di essere libero o proprietario, non sussiste. La faccenda è che, vedete, è consentito agli utenti modificare e ridistribuire il software? Qualora esista un unico utente, il quale ne detiene i diritti, non c'è problema. Quell'utente è libero di fare tutte queste cose. In effetti, qualsiasi programma personalizzato sviluppato da un'azienda per usi interni è software libero, fintantoché si ha il senso di insistere ad avere il codice sorgente e tutti i diritti.

La questione praticamente non esiste per il software che opera in un orologio da polso o in un forno a microonde o nel sistema d'accensione di un'automobile, perché si tratta di luoghi in cui non si preleva alcun software da installare. Non è un vero computer, per quanto riguarda l'utente, così non solleva simili questioni al punto tale da essere eticamente importanti. Per la maggior parte, l'industria del software andrà avanti proprio come ha sempre fatto. E la cosa interessante è che una frazione così ampia di posti di lavoro riguarda tale industria, pur in mancanza di possibilità per attività commerciali di software libero, gli sviluppatori di software libero potrebbero trovare impiego scrivendo software personalizzato. [il pubblico ride] Ce ne sono così tanti; la percentuale è talmente ampia.

Succede però che esiste un'imprenditoria del software libero. Esistono aziende di software libero, e alla conferenza stampa che seguirà parteciperanno anche rappresentanti di un paio di tali aziende. Naturalmente ci sono anche aziende che non si occupano di software libero ma che sviluppano utili parti di swa-

re libero per la distribuzione, e il software libero che producono è sostanziale.

Ora, come operano le aziende di software libero? Alcune di loro vendono copie. L'utente è libero di copiare, ma tali aziende possono anche vendere migliaia di copie al mese. Altre vendono supporto e altri tipi di servizi. Personalmente, nella seconda metà degli anni '80 ho venduto servizi di assistenza al software libero. Praticamente, dicevo, per 200 dollari l'ora posso modificare qualsiasi cosa volete nel software GNU che ho scritto. Sì, era una tariffa esosa, ma si trattava di un programma di cui ero l'autore, la gente avrebbe capito che avrei concluso il lavoro in molte meno ore. [il pubblico ride] Mi guadagnai da vivere in tal modo. Anzi, guadagnai più di quanto avessi mai fatto prima. Ho anche tenuto dei corsi. Continuai così fino al 1990, quando ricevetti un grosso premio¹² e non dovetti più farlo.

Ma il 1990 fu quando si formò la prima corporation nell'imprenditoria del software libero, che fu Cygnus Support. Sostanzialmente la loro attività era dello stesso tipo di quanto avevo fatto io. Sicuramente avrei potuto lavorare per loro, qualora ne avessi avuto bisogno. Non avendone bisogno, ritenni positivo per il movimento rimanere indipendente da qualsiasi azienda. In tal modo avrei potuto dire cose buone e cattive sulle varie aziende di software libero e di software non libero, senza conflitti d'interesse. Così avrei potuto servire meglio il movimento. Ma se avessi avuto bisogno di guadagnarci da vivere, avrei lavorato per loro. È un'attività commerciale etica. Per nessuna ragione mi sarei vergognato di lavo-

¹² Il riferimento è al MacArthur Fellowship, definito da altri anche come "il sussidio del genio". Si tratta di un sussidio di sostentamento della durata di cinque anni, dato a individui che mostrano meriti eccezionali e promettono di proseguire e migliorare il proprio lavoro creativo.

rare con loro. E quell'azienda raggiunse dei guadagni netti nel primo anno di vita. Venne fondata con un capitale minimo, soltanto il denaro dei tre fondatori. Continuò a crescere ogni anno e ad avere profitti ogni anno, finché divennero ingordi di denaro e cercarono investitori esterni, e rovinarono tutto. Ma ci furono svariati anni di successo, prima di diventare ingordi.

Ciò illustra uno degli aspetti eccitanti del software libero. Il software libero dimostra che non si ha bisogno di tirare su dei capitali per sviluppare software libero. Se si hanno dei capitali, si può assumere qualcuno per scrivere vario software. Ma si può fare parecchio con un piccolo numero di persone. Anzi, la tremenda efficacia del processo di sviluppo del software libero è una delle ragioni per cui è importante che il mondo passi al software libero. Ciò inoltre smentisce quanto sostengono in Microsoft, quando dicono che la GNU GPL è negativa perché rende loro più difficile mettere insieme i capitali per lo sviluppo di software non libero e prendere il nostro software libero e mettere il nostro codice nei loro programmi che non vogliono condividere. Praticamente non abbiamo bisogno che raccolgano dei capitali in tal modo. Possiamo farcela comunque. Stiamo per farcela.

La gente era solita dire che non avremmo mai potuto avere un sistema operativo libero completo. Ora l'abbiamo fatto e anche molto di più. Direi che abbiamo raggiunto un ordine di grandezza superiore allo sviluppo di software ad uso generico sufficiente a coprire le necessità del mondo intero. E ciò in un mondo in cui oltre il 90% degli utenti ancora non usa software libero. Ciò in un mondo dove oltre la metà di tutti i server Web del mondo girano su GNU/Linux con Apache come server Web.

Domanda: Cos'hai appena detto, Linux?

RMS: Ho detto GNU/Linux.

Domanda: Davvero?

RMS: Sì, se parlo del kernel lo definisco Linux. È così che si chiama. Il kernel è stato scritto da Linus Torvalds, e dovremmo chiamarlo con il nome scelto da lui, per rispetto all'autore.

In generale, nell'imprenditoria la maggioranza degli utenti non usa GNU/Linux. Gran parte dei comuni utenti ancora non usa il nostro sistema. Quando lo faranno, dovremmo avere automaticamente 10 volte il numero di volontari e 10 volte il numero di clienti per le aziende di software libero degli attuali. Ciò ci porterà a quell'ordine di grandezza. A questo punto, nutro molta fiducia che riusciremo a farcela.

Ciò è importante perché Microsoft sembra voglia sentirci disperati. Dicono, "L'unico modo con cui poter avere software, l'unico modo con cui si può avere innovazione, è dandoci potere. Consentiteci di dominarvi. Fateci controllare cosa potete fare con il software che usate, in modo che possiamo spremervi un sacco di soldi, e usarne una parte per sviluppare software, tenendo il resto come guadagno".

Be', non dovremmo mai sentirci talmente disperati. Non dovremmo mai sentirci così disperati da rinunciare alla libertà. Sarebbe molto pericoloso.

Un'altra cosa riguardo Microsoft, be', non soltanto Microsoft, la gente che non sostiene il software libero generalmente adotta un sistema di valori dove l'unica cosa che conta sono i benefici pratici a breve termine: Quanti soldi riuscirò a fare quest'anno? Quale lavoro potrò concludere oggi? Pensieri a breve termine e pensieri ristretti. L'assunto è che è ridicolo immaginare che qualcuno possa mai fare un sacrificio in nome della libertà.

Ieri¹³ molta gente ha tenuto discorsi sugli americani che hanno fatto dei sacrifici per la libertà dei compatrioti. Alcuni di loro hanno fatto grandi sacrifici. Hanno sacrificato perfino la propria vita per quei tipi di libertà di cui ognuno nel nostro paese ha sentito parlare. (Almeno in alcuni casi; credo che occorra ignorare la guerra in Vietnam).

Ma fortunatamente conservare la libertà nell'uso del software non richiede grandi sacrifici. Bastano sacrifici minimi, piccoli, come imparare l'interfaccia a linee di comando, se ancora non abbiamo un programma con interfaccia grafica. Come fare un lavoro in questo modo, perché non abbiamo ancora un pacchetto di software libero per farlo in quell'altro modo. Come dare dei soldi a un'azienda che sta per sviluppare un certo pacchetto di software libero, in modo che lo si possa avere nel giro di qualche anno. Vari piccoli sacrifici che tutti noi possiamo fare. E in tempi lunghi ne avremo beneficiato anche noi. In realtà è più un investimento che un sacrificio. Dobbiamo soltanto mantenere una prospettiva a lungo termine per comprendere che è positivo per noi investire nel miglioramento della società, senza fare i conti in tasca a chi guadagna qualcosa a seguito di tale investimento.

A questo punto, direi di aver praticamente concluso.

Vorrei menzionare che un nuovo approccio all'attività commerciale del software libero viene proposto da Tony Stanco, da lui definito "sviluppatori liberi" ("Free Developers"), che include una certa struttura imprenditoriale che spera alla fine di pagare una parte degli utili a tutti gli autori di software libero aderenti a tale organizzazione. Ora stanno considerando la prospettiva di ottenere dei grossi contratti governativi per lo sviluppo di software in India,

¹³ Il giorno prima era Memorial Day, la festività statunitense in cui si commemorano gli eroi di guerra.

perché qui si apprestano a usare software libero come base di partenza, visti i tremendi risparmi di spesa.

E ora credo che dovrei sollecitare le vostre domande.

Sessione di domande e risposte

Domanda: In che modo un'azienda come Microsoft potrebbe includere un contratto di software libero?

RMS: Veramente, Microsoft prevede di trasformare buona parte della propria attività in servizi. E quel che pensano di fare è qualcosa di sporco e pericoloso, ovvero legare i servizi ai programmi, uno all'altro, in una specie di zigzag. In modo che per usare questo servizio bisogna prima usare questo programma Microsoft, il che significa che si avrà bisogno di quel servizio, quel programma e così via... è tutto cucito insieme. Questo è il loro piano.

Ora, la cosa interessante è che la vendita di tali servizi non solleva la questione etica del software libero e del software non libero. Potrebbe essere perfettamente corretto per loro avere un'attività per le aziende che vendono quei servizi in rete. Tuttavia, quel che Microsoft prevede di fare è usarle per imporre un blocco ancor più grande, un monopolio ancora più ampio, sul software e sui servizi, e ciò veniva descritto in un recente articolo. Qualcun altro sostiene che sta trasformando la rete nella Microsoft Company Town.

E ciò ha una certa rilevanza perché i giudici del processo antitrust raccomandavano di dividere l'azienda – ma sotto un certo punto di vista ciò non avrebbe alcun senso, non porterebbe nulla di positivo – in una parte per il sistema operativo e un'altra per le applicazioni.

Ma dopo aver letto quell'articolo, credo possa tornare utile e efficace dividere Microsoft in due parti, un'azienda per i servizi e

un'altra per il software, imponendo a entrambe di operare a una certa distanza tra loro. All'azienda dedicata ai servizi andrebbe altresì imposto di pubblicare il codice delle interfacce impiegate, in modo che chiunque possa scrivere un programma client per comunicare con tali interfacce, e direi che simili servizi debbano essere a pagamento. Be', ciò sarebbe positivo. Questa è una faccenda completamente diversa.

Se Microsoft venisse divisa in tal senso... servizi e software, non potrebbero usare il proprio software per far fuori la competizione con i servizi Microsoft. E non sarebbero in grado di usare i propri servizi per bloccare la competizione con il software Microsoft. Noi potremmo creare il software libero, e forse gli utenti lo userebbero per parlare con i servizi Microsoft, e non ce ne preoccuperemo. Perché, dopo tutto, nonostante Microsoft sia l'azienda di software proprietario che ha sottomesso la maggior parte di persone – gli altri ne hanno sottomessi di meno, non che non ci abbiano provato, è soltanto che non sono riusciti a conquistarne così tanti. Il problema non è Microsoft e soltanto Microsoft. Microsoft è l'esempio più eclatante del problema che stiamo cercando di risolvere, ovvero il software proprietario che ruba agli utenti la libertà di cooperare e dare vita a una società etica. Per cui non dovremmo concentrarci troppo su Microsoft, anche se mi hanno offerto l'opportunità di essere su questo podio. Ciò non li rende così importanti. Non sono tutto per tutti.

Domanda: Prima parlavi delle differenze filosofiche tra open source e software libero. Come vedi la recente tendenza delle distribuzioni GNU/Linux verso l'esclusivo supporto di piattaforme Intel? E il fatto che sembra che un numero sempre minore di sviluppatori scriva programmi in modo corretto, software che si compili ovunque? E fa software che funziona unicamente su sistemi Intel?

RMS: Non vedo alcuna questione etica qui. Pur se, in effetti, talvolta le aziende produttrici portano il sistema GNU/Linux sulle proprie macchine. Sembra che recentemente lo abbia fatto Hewlett-Packard. E non si sono curati di pagare per portarvi Windows, perché sarebbe costato troppo. Ma per il supporto di GNU/Linux, credo siano serviti cinque programmatori per qualche mese. Era facilmente fattibile.

Naturalmente io incoraggio la gente a usare autoconf, un pacchetto GNU che facilita la portabilità dei programmi. Li incoraggio a fare così. Oppure quando qualcun altro sistema quel bug che non compilava su una certa versione del sistema, e ti spedisce la riparazione, dovresti inserirla. Ma non considero ciò come una questione etica.

Domanda: Due commenti. Uno è: recentemente sei intervenuto al MIT, ne ho letto la trascrizione. Qualcuno ha fatto una domanda sui brevetti, e tu hai risposto che “i brevetti sono una faccenda completamente diversa. Non ho nulla da commentare al riguardo.”

RMS: Esatto. In realtà avrei molto da dire sui brevetti, ma ci vuole un'ora. [il pubblico ride]

Domanda: Volevo dire questo: Mi sembra che ci sia una questione da affrontare. Esiste un motivo per cui le aziende definiscono sia i brevetti che i copyright come delle proprietà importanti nel tentativo di far passare questo concetto, ovvero che vogliono usare il potere dello Stato per creare un monopolio per se stesse. Ciò che accomuna queste cose non è che ruotino intorno alle medesime questioni, che la motivazione non è veramente una faccenda di servizio pubblico, quanto quella dell'imprenditoria di arrivare a un monopolio per i propri interessi privati.

RMS: Hai ragione sul fatto che vogliono questo. Ma esiste un ulteriore motivo perché insistono a usare il termine proprietà intellettuale. È che non vogliono incoraggiare la gente a pensare con attenzione alle questioni del copyright o dei brevetti. Perché le leggi sul copyright e quelle sui brevetti sono totalmente differenti, e gli effetti del copyright sul software e gli effetti dei brevetti sul software sono completamente diversi tra loro.

I brevetti sul software sono una restrizione sui programmatori, vietando loro di scrivere certi tipi di programmi, diversamente da quanto fa il copyright. Con il copyright, almeno se si è l'autore del programma, se ne consente la distribuzione in proprio. Perciò è assai importante separare le due questioni.

Hanno appena qualcosa in comune, a un livello minimo, e tutto il resto è diverso. Vi invito quindi a incoraggiare una riflessione chiara, o si discute di copyright oppure di brevetti. Ma non parliamo di proprietà intellettuale. Non ho opinioni su quest'ultima. Ma ho opinioni sul copyright, sui brevetti e sul software.

Domanda: All'inizio hai detto che un programma informatico è un linguaggio funzionale, come le ricette. Ma c'è un ampio divario per passare dalle ricette di cucina ai programmi informatici, e dalla lingua inglese ai programmi informatici – la definizione di “linguaggio funzionale” è molto ampia. Ciò sta provocando dei problemi nel caso del DeCSS per i DVD.

RMS: Le questioni sono in parte simili ma in parte diverse, per le cose che non sono funzionali in natura. Parte della questione può essere trasferita ma non per intero. Purtroppo ci vorrebbe un'altra ora per chiarirlo, non ho il tempo di farlo. Ma direi che tutte le opere funzionali devono essere libere nello stesso senso del software. Intendo, libri di testo, manuali, dizionari, ricette, e via di seguito.

Domanda: Stavo pensando alla musica online. Esistono differenze e analogie in tutti questi casi.

RMS: Esatto. Direi che la libertà minima che dovremmo avere per qualsiasi tipo di informazioni pubblicate sia la libertà di ridistribuirle a livello non commerciale, in integrale. Per opere funzionali, dobbiamo avere la libertà di pubblicare a livello commerciale una versione modificata, perché ciò è incredibilmente utile alla società. Per opere non funzionali, lavori d'intrattenimento, o estetici, che esprimono il punto di vista di una persona, forse non dovrebbero essere modificate. E forse ciò significa che è bene avere il copyright a coprirne la distribuzione commerciale.

Teniamo a mente che secondo la costituzione statunitense lo scopo del copyright è a beneficio del pubblico. È quello di cambiare il comportamento di certe entità private, in modo che pubblichino più libri. E il beneficio è che la società può discutere sulle varie questioni e trarne giovamento. E poi abbiamo la letteratura. Abbiamo le opere scientifiche. Lo scopo è incoraggiare tutto ciò. Il copyright non esiste a favore degli autori, per non parlare del vantaggio degli editori. Costoro esistono a servizio dei lettori e di tutti coloro che trarranno beneficio dalla comunicazione dell'informazione che avviene quando qualcuno scrive e altri leggono. E quest'obiettivo mi trova d'accordo.

Ma nell'epoca delle reti informatiche il metodo non è più sostenibile, perché oggi richiede leggi draconiane che invadono la privacy di chiunque e terrorizzano tutti. Anni di carcere per aver condiviso con il vicino. Non era così all'epoca della stampa. Allora il copyright era una regolamentazione industriale. Poneva limitazioni agli editori. Oggi sono questi ultimi a imporre restrizioni al pubblico. Il rapporto di potere è stato ribaltato di 180 gradi, pur trattandosi della medesima legislazione.

Domanda: Cioè, potremmo avere la stessa cosa, tipo fare musica da altra musica?

RMS: Giusto. Questa è un'interessante...

Domanda: E opere uniche, nuove, e ciò significa ancora parecchia cooperazione.

RMS: Lo è. Credo che probabilmente richieda un qualche concetto di uso legittimo ("fair use"). Sicuramente prendere qualche secondo di musica per usarlo in altri lavori, dovrebbe essere chiaramente un uso legittimo. Non saprei dire se i giudici siano d'accordo, ma dovrebbero. Ciò non comporterebbe alcun cambiamento sostanziale nel sistema in vigore finora.

Domanda: Cosa pensi della pubblicazione di informazioni pubbliche in formati proprietari?

RMS: Oh, non dovrebbe succedere. Il governo non dovrebbe mai imporre ai cittadini di usare programmi non liberi per l'accesso, per comunicare con il governo in qualsiasi modo, in qualunque contesto.

Domanda: Ho usato, ora lo dico, GNU/Linux...

RMS: Grazie. [il pubblico ride]

Domanda: ...per gli ultimi quattro anni. Una cosa che mi ha creato problemi e che è piuttosto essenziale, credo, per tutti noi, è poter navigare sul Web.

RMS: Sì.

Domanda: Una cosa che è stata decisamente una debolezza nell'uso del sistema GNU/Linux riguarda la navigazione sul Web, perché lo strumento più diffuso per farlo, Netscape...

RMS:... non è software libero.

Consentitemi di rispondere su questo. Voglio arrivare al punto, per meglio chiarire la questione. Sì, c'è stata una forte tendenza a usare Netscape Navigator sui sistemi GNU/Linux. Anzi, viene incluso in tutti i sistemi commerciali. È una situazione ironica: abbiamo lavorato così duramente per fare un sistema operativo libero e ora, quando si va in un negozio, vi si trovano versioni di GNU/Linux, in maggioranza viene chiamato Linux, e non sono libere. Be', per una parte lo sono. Ma poi c'è Netscape Navigator, e forse anche altri programmi non liberi. Così diventa difficile trovare un sistema veramente libero, a meno di non sapere bene cosa si stia facendo. O, naturalmente si può non installare Netscape Navigator.

In realtà per molti anni sono esistiti dei browser Web liberi. C'è un browser libero che ero solito usare, chiamato Lynx, è un browser non grafico, solo testuale. Ha un grosso vantaggio, nel senso che non vedi la pubblicità. [il pubblico ride] [applausi]

In ogni caso, esiste un progetto grafico libero chiamato Mozilla, che ora sta raggiungendo il punto in cui è possibile usarlo. E occasionalmente lo faccio.

Domanda: Konqueror 2.01 funziona molto bene.

RMS: Ecco un altro browser grafico libero. Finalmente stiamo risolvendo il problema, credo.

Domanda: Puoi parlare delle divisioni filosofiche/etiche tra software libero e open source? Ritieni che siano inconciliabili [cambio di nastro, manca la fine delle domanda e l'inizio della risposta]

RMS:... per la libertà e l'etica. O se basta dire, Spero che voi imprenditori decidiate che è economicamente vantaggioso per voi consentirci di fare queste cose.

Ma, come ho già detto, in molti lavori pratici non importa veramente quali siano le idee politiche di una persona. Quando costui si offre di aiutare il progetto GNU, non gli diciamo: “Devi essere d’accordo con le nostre idee politiche.” Gli diciamo che in un pacchetto GNU devi definire il sistema GNU/Linux e chiamarlo software libero. Quel che dici quando non hai a che fare con il progetto GNU è affar tuo.

Domanda: L’IBM ha lanciato una campagna diretta alle agenzie governative, per vendere le loro grandi macchine nuove, usando Linux come punto forte di vendita, loro dicono Linux.

RMS: Sì, naturalmente si tratta di GNU/Linux. [il pubblico ride]

Domanda: Infatti! Ma vallo a dire al manager delle vendite. Non sa nulla di GNU.

RMS: Il problema è che hanno già deciso con cura quel che vogliono dire onde trarne vantaggio. E la questione di cosa sia più accurato, o legittimo, o il modo corretto di descriverlo non è di primaria importanza per un’azienda come quella. Per qualche azienda minore, c’è un responsabile. E se il responsabile si mostra incline a considerare piccole questioni come questa, potrebbe decidere in tal senso. Ma non una grande corporation. È una vergogna.

C’è un’altra questione più importante e più sostanziale su quanto va facendo IBM. Dicono che stanno mettendo un miliardo di dollari in “Linux.” Ma forse dovrei usare le virgolette anche per “in”, perché parte di quel denaro servirà a pagare persone che sviluppano software libero. Ciò è un contributo concreto alla comunità. Ma con altre parti si pagheranno persone per scrivere software proprietario, o versioni di software proprietario che possano girare su GNU/Linux, e ciò non è affatto un contributo alla nostra comunità. Ma IBM sta ammucciando tutto insieme. Parte potrebbe essere

pubblicità, che è un qualche tipo di contributo, pur se parzialmente errato. È una situazione complicata. Una parte di quanto vanno facendo è un contributo, un'altra non lo è, e un'altra parte ancora lo è in qualche modo ma non precisamente. E non si può soltanto ammucciare tutto insieme e dire, "Bene! Un miliardo di dollari dall'IBM." [il pubblico ride] È una semplificazione eccessiva.

Domanda: Puoi parlare ancora un po' delle idee che portarono alla Licenza Pubblica Generica?

RMS: Le idee che portarono alla Licenza Pubblica Generica? In parte volevo tutelare la libertà della comunità contro i fenomeni che ho descritto con X Windows, che si sono verificati anche con altri programmi liberi. Anzi, quando riflettevo su questa faccenda X Windows non era stato ancora diffuso.

Ma avevo visto sorgere il problema in altri programmi liberi. TeX, ad esempio. Volevo esser certo che tutti gli utenti avessero avuto la libertà. Altrimenti, mi resi conto che avrei scritto un programma e forse un sacco di gente l'avrebbe usato, ma senza avere la libertà. E qual'è il punto di fare così?

Ma l'altra questione su cui andavo riflettendo era che volevo dare alla comunità la sensazione che non fosse uno zerbino, la sensazione che non potesse essere preda di un parassita qualsiasi che se ne andava in giro. Se non si usa il copyleft, sostanzialmente si dice: [con voce sommessa] "Prendi il mio codice, fanne quel che vuoi, non dico no." Così può avvicinarsi chiunque e dire: [con voce molto decisa] "Ah, ne farò una versione non libera, me lo prendo." Allora probabilmente lo miglioreranno un po', e quelle versioni non libere piaceranno agli utenti, finendo per sostituire le versioni libere. E allora, cosa avremmo ottenuto? Avremmo soltanto fatto una donazione a qualche progetto di software proprietario.

E quando la gente si accorge di cosa succede, quando vede altri prendere quel che ho fatto io, senza dare nulla in cambio, può essere demoralizzante. Non si tratta di pure congetture. L'ho visto accadere. Fu parte di quanto avvenne con la scomparsa della comunità a cui appartenevo negli anni '70. Qualcuno prese a diventare poco cooperativo. E ne assumemmo che stavano guadagnandoci sopra in qualche modo. Certamente si comportavano come se stessero guadagnandoci sopra. Ci rendemmo conto come fosse possibile appropriarsi semplicemente della nostra cooperazione e non dar nulla in cambio. E non potevamo far nulla per impedirlo. Fu molto scoraggiante. Ne discutemmo tra quelli che non gradivano la tendenza, ma non ci venne in mente alcuna idea per bloccarla.

La GPL è progettata per bloccare ciò. Dice: Sì, sei benvenuto nella nostra comunità e puoi usare questo codice. Puoi usarlo per ogni tipo di lavoro. Ma se ne diffondi una versione modificata, devi farlo nella comunità, come parte di questa, come parte del mondo libero. In effetti esistono ancora parecchi modi in cui è possibile trarre vantaggio dal nostro lavoro senza contribuirvi, come il fatto di non dover scrivere alcun software. Molta gente usa GNU/Linux e non scrive alcun software. Non esiste alcun requisito perché l'utente debba fare qualcosa per noi. Ma se si fa un certo tipo di cose, bisogna dare il proprio contributo. È questo che s'intende dicendo che la comunità non è un zerbino. E credo che ciò abbia aiutato la gente a rafforzarsi per dire, Non vogliamo essere calpestati dal primo che passa, ci difenderemo.

Domanda: Considerando il software libero ma non sotto copyleft, visto che chiunque può prenderlo e renderlo proprietario, non è possibile anche prenderlo, modificarlo e diffondere l'intero programma sotto la GPL?

RMS: Sì, è possibile.

Domanda: Allora ciò imporrebbe la GPL a tutte le copie future.

RMS: Derivanti da quel programma. Ecco perché in genere non lo facciamo. Fatemi spiegare. Volendo, potevamo prendere X Windows e farne una copia coperta dalla GPL con delle modifiche. Ma c'è un gruppo più ampio di gente che lavora a migliorare X Windows senza metterlo sotto GPL. Così, se lo avessimo fatto, ci saremmo divisi da loro ("forking"). E ciò non è un comportamento positivo nei loro confronti. Fanno parte della comunità, producono dei contributi.

Secondo, si sarebbe rivelato un boomerang, perché stanno lavorando su X assai più di quanto avremmo fatto noi. La nostra versione sarebbe stata inferiore alla loro, la gente non l'avrebbe usata, il che vuol dire, perché mai darsi la pena di farlo?

Se qualcuno apporta dei miglioramenti a X, gli direi di collaborare con il gruppo di sviluppo di X. Faglielo avere e lascia che lo usino a modo loro. Perché stanno sviluppando un pezzo di software libero molto importante. È positivo per noi collaborare con loro.

Domanda: Eccetto che, considerando X in particolare, circa due anni fa, X Consortium era ben addentro nell'open source non libero....

RMS: Be', in realtà non era open source. Possono aver detto lo fosse, non ricordo se lo dissero o meno. Ma non era open source. Era ristretto. Non lo si poteva distribuire a livello commerciale, credo. O non se ne poteva distribuire commercialmente una versione modificata, o qualcosa del genere. C'era una restrizione considerata inaccettabile sia dal movimento del software libero sia da quello open source.

E sì, ciò è quanto si rischia quando si usa una licenza non copyleft. In effetti, X Consortium aveva una posizione molto rigida.

Dicevano: Se una parte anche minima del programma è sotto copyleft, non lo distribuiremo affatto. Non lo inseriremo nella nostra distribuzione.

In tal modo molte persone vennero pressate a non usare il copyleft. E il risultato fu che più tardi tutto il loro software rimase aperto e disponibile. Quando le stesse persone fecero pressione su uno sviluppatore perché troppo permissivo, allora il gruppo di X disse: “Va bene, ora possiamo porre delle restrizioni,” il che non era molto etico da parte loro. Ma considerata la situazione, vogliamo veramente danneggiare le nostre risorse per mantenere una versione alternativa di X coperta dalla GPL? Non avrebbe alcun senso farlo. Ci sono molte altre cose che dobbiamo fare. Occupiamoci di queste invece. Meglio collaborare con gli sviluppatori di X.

Domanda: Puoi commentare, GNU è un marchio registrato? Ed è pratico includerlo come parte della Licenza Generica Pubblica che consente i marchi registrati?

RMS: In realtà stiamo per richiedere la registrazione del marchio per GNU. Ma non ha nulla a che fare con ciò. Sarebbe una storia lunga spiegarne il perché.

Domanda: Si potrebbe imporre di esporre il marchio nei programmi coperti dalla GPL.

RMS: No, non credo sia possibile. Le licenze coprono programmi singoli. E quando un programma fa parte del progetto GNU, nessuno mente al riguardo. Il nome del sistema è una questione completamente diversa. È una faccenda laterale. Non vale la pena discuterne ulteriormente.

Domanda: Se potessi premere un pulsante per costringere tutte le aziende a rendere libero il proprio software, quale pulsante sarebbe?

RMS: Be', lo userei soltanto per il software pubblicato. Credo che la gente abbia il diritto a scrivere programmi privati e a farne uso. E ciò include le aziende. Questa è una questione di privacy. È vero, possono esserci delle volte in cui ciò sia sbagliato, come quando qualcosa è terribilmente utile all'umanità e invece la si tiene segreta. È sbagliato, ma è un errore di tipo diverso. Si tratta di una faccenda diversa, pur riguardando la medesima area.

Sì, direi che tutto il software pubblicato debba essere libero. E ricordiamolo, quando non è software libero, lo si deve all'intervento governativo. Il governo interviene per renderlo non libero. Il governo crea poteri giuridici speciali da trasferire ai proprietari dei programmi, in modo che questi possano chiamare la polizia per impedirci di usare quei programmi in un certo modo. E con questo, direi di terminare.

Ed Schonberg: L'intervento di Richard ha creato una quantità enorme di energia intellettuale. Vorrei suggerire che parte di questa dovrebbe essere diretta per usare, e possibilmente per scrivere, software libero. Dovremmo chiudere rapidamente. Vorrei aggiungere che Richard ha iniettato in una professione nota al pubblico generale per un'apatia politica terminale, un livello di discussione politica e morale che, credo, sia senza precedenti nella nostra professione. E per questo gli dobbiamo davvero molto. [il pubblico applaude]

Questa è la trascrizione di un intervento tenuto alla New York University il 29 maggio 2001. Questa versione fa parte del libro *Free Software, Free Society: The Selected Essays of Richard M. Stallman*, GNU Press, 2002.

La copia letterale e la distribuzione di questo testo nella sua integrità sono permesse con qualsiasi mezzo, a condizione che sia mantenuta questa nota.

Termini da evitare

Ci sono vari termini ed espressioni che raccomandiamo di evitare perché risultano ambigui oppure implicano un'opinione che speriamo non condiviate per intero.

Licenze tipo-BSD

L'espressione "licenza di tipo BSD" crea confusione perché fa un solo fascio di licenze che presentano differenze importanti. Ad esempio, la licenza BSD originale con la sua clausola pubblicitaria è incompatibile con la Licenza Pubblica Generica (GPL), mentre invece la nuova licenza BSD è compatibile con la GPL.

Per evitare confusioni, è meglio indicare la specifica licenza cui ci si riferisce, evitando la vaga locuzione "di tipo BSD".

Commerciale

È bene non usare "commerciale" come sinonimo di "non-libero". Ciò confonde due questioni del tutto diverse tra loro.

Un programma è commerciale se viene sviluppato come attività imprenditoriale. Un programma commerciale può essere libero o non-libero, a seconda della relativa licenza. Analogamente, un programma sviluppato da una scuola o da un individuo può essere libero o non-libero sulla base della relativa licenza. Le due questioni, quale tipo di entità ha sviluppato il programma e quale la libertà concessa agli utenti, sono indipendenti tra loro.

Nel primo decennio di attività del Movimento del Software Libero, i pacchetti di software libero erano quasi sempre non-commerciali; le componenti del sistema operativo GNU/Linux furono sviluppate da individui, da organizzazioni senza scopo di lucro

come la Free Software Foundation o da università. Ma negli anni '90 ha preso a circolare il software libero commerciale. Il software libero commerciale è un contributo per la nostra comunità, perciò dovremmo incoraggiarlo.

Ma quanti credono che “commerciale” significhi “non-libero”, tenderanno a considerare contraddittoria la combinazione “libero commerciale”, scartando questa possibilità. Bisogna stare attenti a non utilizzare il termine “commerciale” in tal senso.

Contenuto

L'utilizzo del sostantivo “contenuto” (content) per descrivere opere scritte e di altro tipo create da un autore rivela un atteggiamento specifico nei confronti di tali opere: le considera come beni di consumo intercambiabili il cui scopo è quello di riempire una scatola e far soldi. In realtà ciò significa considerare in maniera irrispettosa le opere stesse.

Coloro che usano questo termine sono spesso editori tesi ad ottenere un maggior potere nel copyright a nome degli autori (o “creatori”, come vanno definendoli) delle opere. Il termine “contenuto” ne rivela i sentimenti concreti.

Finché altri continuano ad usare l'espressione “fornitori di contenuto”, i dissidenti politici potranno sicuramente autodefinirsi “fornitori di scontento”.

(In inglese “content” significa sia “contento” che “contenuto”).

Creatore

Applicare il termine “creatore” agli autori significa paragonarli implicitamente a una deità (“il creatore”). Questo termine viene usato dagli editori per elevare la statura morale degli autori al di sopra della gente comune, onde giustificare un maggior potere del copyright che gli editori possono esercitare a nome degli stessi autori.

Digital Rights Management (Gestione dei diritti digitali)

Il software per il “Digital Rights Management” in realtà è progettato per imporre restrizioni agli utenti di computer. Il ricorso al termine “diritti” in questo contesto è pura propaganda, mirata a farci considerare inconsciamente la questione dal punto di vista dei pochi che impongono tali restrizioni, ignorando al contempo quella dei molti a cui le restrizioni vengono imposte. Buone alternative sono espressioni quali “Digital Restrictions Management” (Gestione delle restrizioni digitali) e “handcuffware” (software-manette).

For free (gratuito)

Quando ci si riferisce a un programma di software libero (free software), meglio non dire che è disponibile “for free”, gratuitamente. Questo termine (in inglese) significa specificamente “a costo zero”. Il software libero è una questione di libertà, non di prezzo. Spesso le copie di programmi di software libero sono disponibili “for free”, a costo zero – ad esempio, tramite download via FTP. Ma copie di programmi di software libero sono disponibili anche a pagamento su CD-ROM; invece, le copie di software proprietario talvolta sono disponibili gratuitamente a fini promozionali e alcuni pacchetti proprietari sono normalmente disponibili a costo zero per determinati utenti.

Onde evitare confusioni, si può dire che il programma è disponibile “come software libero”.

Freeware

Evitiamo per favore il termine “freeware” come sinonimo di “software libero” (free software). Il termine “freeware” veniva spesso usato negli anni ‘80 per indicare programmi rilasciati per la sola esecuzione, senza renderne disponibili i codici sorgenti. Oggi tale termine non indica alcuna specifica definizione generale.

Inoltre, per lingue diverse dall'inglese, è bene evitare di prendere in prestito termini inglesi come “free software” o “freeware”. Cercate di utilizzare le espressioni spesso meno ambigue offerte dalla vostra lingua. Questo un elenco di traduzioni raccomandate e non ambigue per il termine “free software” in altre lingue:

Ceco: svobodny software
Coreano: ja-yu software
Danese: fri software OPPURE frit programmel
Esperanto: libera softwaro
Finnico: vapaa ohjelmisto
Francese: logiciels libres
Giapponese: jiyuu-na software
Indonesiano: perangkat lunak bebas
Islandese: frjls hugbnaur
Italiano: software libero
Norvegese: fri programvare
Olandese: vrije software
Polacco: wolne oprogramowanie
Portoghese: software livre
Slovacco: slobodny softver
Sloveno: prosto programje
Spagnolo: software libre
Svedese: fri programvara
Tedesco: freie Software
Turco: ozgur yazilim
Ungherese: szabad szoftver

Usando un termine nella vostra lingua, dimostrate che vi riferite effettivamente alla libertà e non state semplicemente cercando di scimmiettare qualche misterioso concetto straniero di marketing.

All'inizio, il riferimento alla libertà potrà sembrare strano o fastidioso ai vostri concittadini, ma quando ne considereranno il significato preciso, capiranno veramente di cosa si tratta.

Furto

I sostenitori del diritto d'autore spesso usano termini quali “rubato” e “furto” per descrivere le infrazioni al copyright. Allo stesso tempo costoro ci chiedono di considerare il sistema giudiziario come un'autorità in campo etico: se copiare è vietato, allora dev'essere qualcosa di male.

Perciò è pertinente ricordare che il sistema giuridico – almeno negli USA – nega il concetto secondo cui l'infrazione al diritto d'autore sia un “furto”. I sostenitori del copyright si appellano a un'autorità... e presentano in maniera sbagliata quanto sostiene tale autorità. L'idea secondo cui siano le leggi a stabilire ciò che è giusto o sbagliato in generale è errata. Nel migliore dei casi, queste norme rappresentano il tentativo di ottenere giustizia: sostenere che siano le leggi a definire la giustizia o il comportamento etico equivale a ribaltare completamente le cose.

Pirateria

Spesso gli editori descrivono l'attività proibita della copia come “pirateria” (piracy). In questo modo, sottintendono che effettuare una copia illegale equivale eticamente all'assalto di navi in alto mare, al rapimento e all'assassinio di quanti si trovano a bordo. Se non ritenete che effettuare copie illegali sia analogo al rapimento e all'assassinio, forse preferirete evitare il ricorso al termine “pirateria” per descrivere tale pratica. In sostituzione, si possono usare espressioni neutre quali “copia proibita” o “copia non autorizzata”. Alcuni potrebbero addirittura preferire un'espressione positiva come “condividere informazioni con il vicino”.

Proprietà intellettuale

Editori e avvocati amano descrivere il diritto d'autore come "proprietà intellettuale". Questo termine contiene un presupposto nascosto – che il modo più naturale di considerare la questione della copia sia basato su un'analogia con gli oggetti fisici, e sull'idea di considerarli una proprietà.

Ma quest'analogia ignora la differenza cruciale esistente tra gli oggetti materiali e l'informazione: l'informazione può essere copiata e condivisa quasi senza sforzo, mentre ciò non è vero degli oggetti materiali. Basare la riflessione su una tale analogia equivale a ignorare questa differenza.

Neppure il sistema giuridico statunitense accetta per intero l'analogia, poiché non tratta il copyright alla pari del diritto di proprietà relativo agli oggetti fisici.

Se non volete limitarvi a questo modo di pensare, è bene evitare l'uso del termine "proprietà intellettuale" nelle vostre parole e riflessioni.

Esiste un ulteriore problema con "proprietà intellettuale": è un contenitore generico in cui vengono messi insieme svariati sistemi giuridici differenti, incluso il copyright, i brevetti, i marchi registrati e altre cose che hanno pochissimo in comune tra loro. Questi sistemi giuridici hanno origini separate, coprono attività diverse, operano in maniera differente, e suscitano questioni diverse di politica pubblica. Se, ad esempio, imparate qualcosa riguardo le norme sul copyright, fareste bene a presumere che ciò non possa applicarsi alla legislazione sui brevetti, perché è quasi sempre così.

Trattandosi di legislazioni talmente diverse tra loro, il termine "proprietà intellettuale" è un invito a fare una super-generalizzazione semplicistica. Qualsiasi opinione sulla "proprietà intellettuale" risulterà quasi sicuramente avventata. Ad un livello così

generico, è impossibile perfino prendere in considerazione le specificità di politica pubblica suscitate dalle norme sul copyright, o le diverse questioni sollevate dalla legislazione sui brevetti o su uno qualsiasi degli altri settori. Il termine “proprietà intellettuale” porta la gente a concentrarsi sul minimo aspetto comune di queste legislazioni differenti tra loro, vale a dire sul fatto che istituiscono una serie di concetti astratti che possono essere acquistati e venduti, per ignorarne l’aspetto centrale, ovvero le restrizioni che tali norme impongono al pubblico e le conseguenze positive o negative che ne risultano.

Onde riflettere con chiarezza sulle problematiche sollevate dai brevetti, dal diritto d’autore e dai marchi registrati, o anche soltanto per conoscere il contenuto di queste norme, il primo passo è dimenticare di aver mai sentito il termine “proprietà intellettuale”. Meglio invece presentare il tema come copyright, brevetti, o qualsiasi altra legislazione specifica di cui si stia discutendo.

Secondo il professor Mark Lemley della University of Texas Law School, l’uso generalizzato del termine “proprietà intellettuale” è una moda recente, diffusasi a partire dalla fondazione dell’Organizzazione Mondiale della Proprietà Intellettuale (World Intellectual Property Organization) nel 1967.¹ Quest’organizzazione rappresenta gli interessi dei detentori di copyright, di brevetti e di marchi registrati, ed esercita pressione sui governi per incrementarne il potere. Uno dei trattati dell’organizzazione segue le direttive del Digital Millennium Copyright Act, che negli Stati Uniti è stato usato per censurare l’impiego di utili pacchetti di software libero.²

¹ Si veda la nota n. 123 alla sua recensione del libro di James Boyle, *Romantic Authorship and the Rhetoric of Property*, pubblicata nel marzo 1997 nella *Texas Law Review*.

² Si veda <http://www.wipout.net/> per la campagna contro la World Intellectual Property Organization.

Protezione, tutela

Gli avvocati degli editori adorano ricorrere al termine “protezione” o “tutela” in riferimento al diritto d’autore. Questi termini implicano l’idea di voler bloccare qualche distruzione o sofferenza; di conseguenza, incoraggiano la gente a identificarsi con il proprietario e con l’editore che traggono dei benefici dal copyright, anziché con gli utenti che ne subiscono le restrizioni.

È facile evitare “protezione” o “tutela” per sostituirli invece con altri termini. Ad esempio, anziché: “La tutela del copyright dura molto a lungo”, si può dire: “Il copyright dura molto a lungo”. Per criticare il diritto d’autore piuttosto che sostenerlo, basta ricorrere all’espressione “le restrizioni del copyright”.

RAND (reasonable and non-discriminatory)

Le entità incaricate di stabilire gli standard limitati dai brevetti che vietano il software libero, in genere seguono la prassi di ottenere licenze su tali brevetti dietro il pagamento di una somma fissa per ogni copia di programma conforme. Spesso queste licenze vengono indicate con il termine “RAND”, acronimo che sta per “ragionevoli e non discriminatorie” (reasonable and non-discriminatory).

Il termine conferisce una rispettabilità apparente a una serie di licenze sui brevetti che normalmente non sono né ragionevoli né non-discriminatorie. È vero che tali licenze non discriminano contro nessun particolare individuo, e tuttavia discriminano a sfavore della comunità del software libero, e ciò le rende irragionevoli. Perciò, una metà del significato di “RAND” è fuorviante mentre l’altra metà esprime un pregiudizio. Le entità responsabili degli standard dovrebbero riconoscere che queste licenze sono discriminatorie e abbandonare l’uso dell’espressione “ragionevoli e non discriminatorie” per descriverle. Finché non lo faranno, altri scrittori che non vogliono essere associati a quella rispettabilità fasulla, bene farebbero a riget-

tare tale espressione. Accettarla e usarla soltanto perché le aziende che detengono i brevetti l'hanno ampiamente diffusa significa consentire a tali aziende di imporre agli altri quelle opinioni. In sostituzione, suggerisco l'espressione "uniform fee only", soltanto dietro pagamento di una tariffa uniforme, o l'acronimo "UFO" (gioco di parole: UFO comunemente sta per unidentified flying objects, i dischi volanti). È una descrizione accurata perché la sola condizione per queste licenze è il pagamento di una tariffa uniforme per le *royalties*.

Regalare software

È fuorviante usare il termine "regalare" (give away) quando si vuole intendere "distribuire un programma come software libero". È lo stesso problema già visto in "for free" (in inglese): implica che il punto in questione sia il prezzo, non la libertà. Un modo per evitare questa confusione consiste nel dire: "rilasciare come software libero".

Vendere software

L'espressione "vendere software" è ambigua. In senso stretto, scambiare la copia di un programma libero con una somma di denaro significa "vendere", ma in genere si associa il termine "vendere" alle restrizioni proprietarie nel successivo utilizzo del software. Per essere più precisi, ed evitare confusioni, si può dire: "distribuire copie di un programma dietro pagamento di una quota" oppure "imporre restrizioni proprietarie sull'uso di un programma", a seconda di ciò cui ci si riferisce.

Originariamente scritto nel 1996, questa versione fa parte del libro *Free Software, Free Society: The Selected Essays of Richard M. Stallman*, GNU Press, 2002.

La copia letterale e la distribuzione di questo testo nella sua integrità sono permesse con qualsiasi mezzo, a condizione che sia mantenuta questa nota.

Parte Seconda

Le licenze

Licenza Pubblica Generica (GPL) del Progetto GNU

Versione 2, Giugno 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Chiunque può copiare e distribuire copie letterali di questo documento di licenza, ma non ne è permessa la modifica.

(NdT: Questa è una traduzione italiana non ufficiale della Licenza Pubblica Generica, GPL. Non è pubblicata dalla Free Software Foundation e non ha valore legale nell'esprimere i termini di distribuzione del software che usa la licenza GPL. Solo la versione originale inglese della licenza ha valore legale. Speriamo ad ogni modo che questa traduzione aiuti le persone di lingua italiana a comprendere meglio il significato della GPL.)

Preambolo

Le licenze della maggior parte dei programmi hanno lo scopo di togliere all'utente la libertà di condividere e modificare il programma stesso. Viceversa, la Licenza Pubblica Generica GNU è intesa a garantire la libertà di condividere e modificare il software libero, al fine di assicurare che i programmi siano liberi per tutti i loro utenti. Questa Licenza si applica alla maggioranza dei programmi della Free Software Foundation e ad ogni altro program-

ma i cui autori hanno deciso di usare questa Licenza. Alcuni altri programmi della Free Software Foundation sono invece coperti dalla Licenza Pubblica Generica Minore. Chiunque può usare questa Licenza per i propri programmi.

Quando si parla di software libero (free software), ci si riferisce alla libertà, non al prezzo. Le nostre Licenze (la GPL e la LGPL) sono progettate per assicurarsi che ciascuno abbia la libertà di distribuire copie del software libero (e farsi pagare per questo, se vuole), che ciascuno riceva il codice sorgente o che lo possa ottenere se lo desidera, che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi liberi e che ciascuno sappia di potere fare queste cose.

Per proteggere i diritti dell'utente, abbiamo bisogno di creare delle restrizioni che vietino a chiunque di negare questi diritti o di chiedere di rinunciarvi. Queste restrizioni si traducono in certe responsabilità per chi distribuisce copie del software e per chi lo modifica.

Per esempio, chi distribuisce copie di un programma coperto da GPL, sia gratis sia in cambio di un compenso, deve concedere ai destinatari tutti i diritti che ha ricevuto. Deve anche assicurarsi che i destinatari ricevano o possano ottenere il codice sorgente. E deve mostrar loro queste condizioni di licenza, in modo che essi conoscano i propri diritti.

Proteggiamo i diritti dell'utente in due modi: (1) proteggendo il software con un copyright, e (2) offrendo una licenza che dia il permesso legale di copiare, distribuire e modificare il Programma.

Inoltre, per proteggere ogni autore e noi stessi, vogliamo assicurarci che ognuno capisca che non ci sono garanzie per i programmi

coperti da GPL. Se il programma viene modificato da qualcun altro e ridistribuito, vogliamo che gli acquirenti sappiano che ciò che hanno non è l'originale, in modo che ogni problema introdotto da altri non si rifletta sulla reputazione degli autori originari.

Infine, ogni programma libero è costantemente minacciato dai brevetti sui programmi. Vogliamo evitare il pericolo che chi ridistribuisce un programma libero ottenga la proprietà di brevetti, rendendo in pratica il programma cosa di sua proprietà. Per prevenire questa evenienza, abbiamo chiarito che ogni brevetto debba essere concesso in licenza d'uso a chiunque, o non avere alcuna restrizione di licenza d'uso.

Seguono i termini e le condizioni precisi per la copia, la distribuzione e la modifica.

Termini e condizioni per la copia, la distribuzione e la modifica

0. Questa Licenza si applica a ogni programma o altra opera che contenga una nota da parte del detentore del copyright che dica che tale opera può essere distribuita sotto i termini di questa Licenza Pubblica Generica. Il termine "Programma" nel seguito si riferisce ad ogni programma o opera così definita, e l'espressione "opera basata sul Programma" indica sia il Programma sia ogni opera considerata "derivata" in base alla legge sul copyright; in altre parole, un'opera contenente il Programma o una porzione di esso, sia letteralmente sia modificato o tradotto in un'altra lingua. Da qui in avanti, la traduzione è in ogni caso considerata una "modifica". Vengono ora elencati i diritti dei beneficiari della licenza.

Attività diverse dalla copiatura, distribuzione e modifica non sono coperte da questa Licenza e sono al di fuori della sua influenza.

L'atto di eseguire il Programma non viene limitato, e l'output del programma è coperto da questa Licenza solo se il suo contenuto costituisce un'opera basata sul Programma (indipendentemente dal fatto che sia stato creato eseguendo il Programma). In base alla natura del Programma il suo output può essere o meno coperto da questa Licenza.

1. È lecito copiare e distribuire copie letterali del codice sorgente del Programma così come viene ricevuto, con qualsiasi mezzo, a condizione che venga riprodotta chiaramente su ogni copia una appropriata nota di copyright e di assenza di garanzia; che si mantengano intatti tutti i riferimenti a questa Licenza e all'assenza di ogni garanzia; che si dia a ogni altro destinatario del Programma una copia di questa Licenza insieme al Programma.

È possibile richiedere un pagamento per il trasferimento fisico di una copia del Programma, è anche possibile a propria discrezione richiedere un pagamento in cambio di una copertura assicurativa.

2. È lecito modificare la propria copia o copie del Programma, o parte di esso, creando perciò un'opera basata sul Programma, e copiare o distribuire tali modifiche o tale opera secondo i termini del precedente comma 1, a patto che siano soddisfatte tutte le condizioni che seguono:

a) Bisogna indicare chiaramente nei file che si tratta di copie modificate e la data di ogni modifica.

b) Bisogna fare in modo che ogni opera distribuita o pubblicata, che in parte o nella sua totalità derivi dal Programma o da parti di esso, sia concessa nella sua interezza in licenza gratuita ad ogni terza parte, secondo i termini di questa Licenza.

c) Se normalmente il programma modificato legge comandi interattivamente quando viene eseguito, bisogna fare in modo che all'inizio dell'esecuzione interattiva usuale, esso stampi un messaggio contenente una appropriata nota di copyright e di assenza di garanzia (oppure che specifichi il tipo di garanzia che si offre). Il messaggio deve inoltre specificare che chiunque può ridistribuire il programma alle condizioni qui descritte e deve indicare come reperire questa Licenza. Se però il programma di partenza è interattivo ma normalmente non stampa tale messaggio, non occorre che un'opera basata sul Programma lo stampi.

Questi requisiti si applicano all'opera modificata nel suo complesso.

Se sussistono parti identificabili dell'opera modificata che non siano derivate dal Programma e che possono essere ragionevolmente considerate lavori indipendenti, allora questa Licenza e i suoi termini non si applicano a queste parti quando queste vengono distribuite separatamente. Se però queste parti vengono distribuite all'interno di un prodotto che è un'opera basata sul Programma, la distribuzione di quest'opera nella sua interezza deve avvenire nei termini di questa Licenza, le cui norme nei confronti di altri utenti si estendono all'opera nella sua interezza, e quindi ad ogni sua parte, chiunque ne sia l'autore.

Quindi, non è nelle intenzioni di questa sezione accampare diritti, né contestare diritti su opere scritte interamente da altri; l'intento è piuttosto quello di esercitare il diritto di controllare la distribuzione di opere derivati dal Programma o che lo contengano.

Inoltre, la semplice aggregazione di un'opera non derivata dal Programma col Programma o con un'opera da esso derivata su di un mezzo di memorizzazione o di distribuzione, non è sufficien-

te a includere l'opera non derivata nell'ambito di questa Licenza.
3. È lecito copiare e distribuire il Programma (o un'opera basata su di esso, come espresso al comma 2) sotto forma di codice oggetto o eseguibile secondo i termini dei precedenti commi 1 e 2, a patto che si applichi una delle seguenti condizioni:

a) Il Programma sia corredato del codice sorgente completo, in una forma leggibile da calcolatore, e tale sorgente sia fornito secondo le regole dei precedenti commi 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.

b) Il Programma sia accompagnato da un'offerta scritta, valida per almeno tre anni, di fornire a chiunque ne faccia richiesta una copia completa del codice sorgente, in una forma leggibile da calcolatore, in cambio di un compenso non superiore al costo del trasferimento fisico di tale copia, che deve essere fornita secondo le regole dei precedenti commi 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.

c) Il Programma sia accompagnato dalle informazioni che sono state ricevute riguardo alla possibilità di ottenere il codice sorgente. Questa alternativa è permessa solo in caso di distribuzioni non commerciali e solo se il programma è stato ottenuto sotto forma di codice oggetto o eseguibile in accordo al precedente comma B.

Per “codice sorgente completo” di un'opera si intende la forma preferenziale usata per modificare un'opera. Per un programma eseguibile, “codice sorgente completo” significa tutto il codice sorgente di tutti i moduli in esso contenuti, più ogni file associato che definisca le interfacce esterne del programma, più gli script usati per controllare la compilazione e l'installazione dell'eseguibile. In ogni caso non è necessario che il codice sorgente fornito

includa nulla che sia normalmente distribuito (in forma sorgente o in formato binario) con i principali componenti del sistema operativo sotto cui viene eseguito il Programma (compilatore, kernel, e così via), a meno che tali componenti accompagnino l'eseguibile.

Se la distribuzione dell'eseguibile o del codice oggetto è effettuata indicando un luogo dal quale sia possibile copiarlo, permettere la copia del codice sorgente dallo stesso luogo è considerata una valida forma di distribuzione del codice sorgente, anche se copiare il sorgente è facoltativo per l'acquirente.

4. Non è lecito copiare, modificare, sublicenziare, o distribuire il Programma in modi diversi da quelli espressamente previsti da questa Licenza. Ogni tentativo di copiare, modificare, sublicenziare o distribuire il Programma non è autorizzato, e farà terminare automaticamente i diritti garantiti da questa Licenza. D'altra parte ogni acquirente che abbia ricevuto copie, o diritti, coperti da questa Licenza da parte di persone che violano la Licenza come qui indicato non vedranno invalidata la loro Licenza, purché si comportino conformemente ad essa.

5. L'acquirente non è tenuto ad accettare questa Licenza, poiché non l'ha firmata. D'altra parte nessun altro documento garantisce il permesso di modificare o distribuire il Programma o i lavori derivati da esso. Queste azioni sono proibite dalla legge per chi non accetta questa Licenza; perciò, modificando o distribuendo il Programma o un'opera basata sul programma, si indica nel fare ciò l'accettazione di questa Licenza e quindi di tutti i suoi termini e le condizioni poste sulla copia, la distribuzione e la modifica del Programma o di lavori basati su di esso.

6. Ogni volta che il Programma o un'opera basata su di esso vengono distribuiti, l'acquirente riceve automaticamente una licenza d'uso da parte del licenziatario originale. Tale licenza regola la copia, la distribuzione e la modifica del Programma secondo questi termini e queste condizioni. Non è lecito imporre restrizioni ulteriori all'acquirente nel suo esercizio dei diritti qui garantiti. Chi distribuisce programmi coperti da questa Licenza non è comunque tenuto a imporre il rispetto di questa Licenza a terzi.

7. Se, come conseguenza del giudizio di un tribunale, o di una imputazione per la violazione di un brevetto o per ogni altra ragione (non limitatamente a questioni di brevetti), vengono imposte condizioni che contraddicono le condizioni di questa licenza, che queste condizioni siano dettate dalla corte, da accordi tra le parti o altro, queste condizioni non esimono nessuno dall'osservazione di questa Licenza. Se non è possibile distribuire un prodotto in un modo che soddisfi simultaneamente gli obblighi dettati da questa Licenza e altri obblighi pertinenti, il prodotto non può essere affatto distribuito. Per esempio, se un brevetto non permettesse a tutti quelli che lo ricevono di ridistribuire il Programma senza obbligare al pagamento di diritti, allora l'unico modo per soddisfare contemporaneamente il brevetto e questa Licenza è di non distribuire affatto il Programma.

Se una qualunque parte di questo comma è ritenuta non valida o non applicabile in una qualunque circostanza, deve comunque essere applicata l'idea espressa da questo comma; in ogni altra circostanza invece deve essere applicato questo comma nel suo complesso.

Non è nelle finalità di questo comma indurre gli utenti ad infran-

gere alcun brevetto né ogni altra rivendicazione di diritti di proprietà, né di contestare la validità di alcuna di queste rivendicazioni; lo scopo di questo comma è unicamente quello di proteggere l'integrità del sistema di distribuzione dei programmi liberi, che viene realizzato tramite l'uso di licenze pubbliche. Molte persone hanno contribuito generosamente alla vasta gamma di programmi distribuiti attraverso questo sistema, basandosi sull'applicazione fedele di tale sistema. L'autore/donatore può decidere di sua volontà se preferisce distribuire il software avvalendosi di altri sistemi, e l'acquirente non può imporre la scelta del sistema di distribuzione.

Questo comma serve a rendere il più chiaro possibile ciò che crediamo sia una conseguenza del resto di questa Licenza.

8. Se in alcuni paesi la distribuzione o l'uso del Programma sono limitati da brevetto o dall'uso di interfacce coperte da copyright, il detentore del copyright originale che pone il Programma sotto questa Licenza può aggiungere limiti geografici espliciti alla distribuzione, per escludere questi paesi dalla distribuzione stessa, in modo che il programma possa essere distribuito solo nei paesi non esclusi da questa regola. In questo caso i limiti geografici sono inclusi in questa Licenza e ne fanno parte a tutti gli effetti.

9. All'occorrenza la Free Software Foundation può pubblicare revisioni o nuove versioni di questa Licenza Pubblica Generica. Tali nuove versioni saranno simili a questa nello spirito, ma potranno differire nei dettagli al fine di coprire nuovi problemi e nuove situazioni.

Ad ogni versione viene dato un numero identificativo. Se il Pro-

gramma asserisce di essere coperto da una particolare versione di questa Licenza e “da ogni versione successiva”, l’acquirente può scegliere se seguire le condizioni della versione specificata o di una successiva. Se il Programma non specifica quale versione di questa Licenza deve applicarsi, l’acquirente può scegliere una qualsiasi versione tra quelle pubblicate dalla Free Software Foundation.

10. Se si desidera incorporare parti del Programma in altri programmi liberi le cui condizioni di distribuzione differiscano da queste, è possibile scrivere all’autore del Programma per chiederne l’autorizzazione. Per il software il cui copyright è detenuto dalla Free Software Foundation, si scriva alla Free Software Foundation; talvolta facciamo eccezioni alle regole di questa Licenza. La nostra decisione sarà guidata da due finalità: preservare la libertà di tutti i prodotti derivati dal nostro software libero e promuovere la condivisione e il riutilizzo del software in generale.

Nessuna garanzia

11. POICHÉ IL PROGRAMMA È CONCESSO IN USO GRATUITAMENTE, NON C’È GARANZIA PER IL PROGRAMMA, NEI LIMITI PERMESSI DALLE VIGENTI LEGGI. SE NON INDICATO DIVERSAMENTE PER ISCRITTO, IL DETENTORE DEL COPYRIGHT E LE ALTRE PARTI FORNISCONO IL PROGRAMMA “COSÌ COM’È”, SENZA ALCUN TIPO DI GARANZIA, NÉ ESPlicita NÉ IMPLICITa; CIÒ COMPRENDE, SENZA LIMITARSI A QUESTO, LA GARANZIA IMPLICITa DI COMMERCIALIZZABILITÀ E UTILIZZABILITÀ PER UN PARTICOLARE SCOPO. L’INTERO RISCHIO CONCERNENTE LA QUALITÀ E LE PRESTAZIONI DEL PROGRAMMA È DELL’ACQUIRENTE. SE IL PROGRAMMA DOVESSE RIVELARSI DIFETTOSO, L’ACQUIRENTE SI ASSUME IL COSTO DI OGNI MANUTENZIONE, RIPARAZIONE O CORREZIONE NECESSARIA.

12. NÉ IL DETENTORE DEL COPYRIGHT NÉ ALTRE PARTI CHE POSSONO MODIFICARE O RIDISTRIBUIRE IL PROGRAMMA COME PERMESSO IN QUESTA LICENZA SONO RESPONSABILI PER DANNI NEI CONFRONTI DELL'ACQUIRENTE, A MENO CHE QUESTO NON SIA RICHiesto DALLE LEGGI VIGENTI O APPAIA IN UN ACCORDO SCRITTO. SONO INCLUSI DANNI GENERICI, SPECIALI O INCIDENTALI, COME PURE I DANNI CHE CONSEGUONO DALL'USO O DALL'IMPOSSIBILITÀ DI USARE IL PROGRAMMA; CIÒ COMPRENDE, SENZA LIMITARSI A QUESTO, LA PERDITA DI DATI, LA CORRUZIONE DEI DATI, LE PERDITE SOSTENUTE DALL'ACQUIRENTE O DA TERZI E L'INCAPACITÀ DEL PROGRAMMA A INTERAGIRE CON ALTRI PROGRAMMI, ANCHE SE IL DETENTORE O ALTRE PARTI SONO STATE AVVISATE DELLA POSSIBILITÀ DI QUESTI DANNI.

Fine dei termini e delle condizioni

Appendice: come applicare questi termini a nuovi programmi

Se si sviluppa un nuovo programma e lo si vuole rendere della maggiore utilità possibile per il pubblico, la cosa migliore da fare è rendere tale programma libero, cosicché ciascuno possa ridistribuirlo e modificarlo sotto questi termini.

Per fare questo, si inserisca nel programma la seguente nota. La cosa migliore da fare è mettere la nota all'inizio di ogni file sorgente, per chiarire nel modo più efficiente possibile l'assenza di garanzia; ogni file dovrebbe contenere almeno la nota di copyright e l'indicazione di dove trovare l'intera nota.

Una riga per dire in breve il nome del programma e cosa fa Copyright (C) anno nome dell'autore

Questo programma è software libero; è lecito ridistribuirlo o modi-

ficarlo secondo i termini della Licenza Pubblica Generica GNU come è pubblicata dalla Free Software Foundation; o la versione 2 della licenza o (a propria scelta) una versione successiva.

Questo programma è distribuito nella speranza che sia utile, ma SENZA ALCUNA GARANZIA; senza neppure la garanzia implicita di NEGOZIABILITÀ o di APPLICABILITÀ PER UN PARTICOLARE SCOPO. Si veda la Licenza Pubblica Generica GNU per avere maggiori dettagli.

Questo programma deve essere distribuito assieme ad una copia della Licenza Pubblica Generica GNU; in caso contrario, se ne può ottenere una scrivendo alla Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Si aggiungano anche informazioni su come si può essere contattati tramite posta elettronica e cartacea.

Se il programma è interattivo, si faccia in modo che stampi una breve nota simile a questa quando viene usato interattivamente:

Orcaloca versione 69, Copyright (C) anno nome dell'autore Orcaloca non ha ALCUNA GARANZIA; per dettagli usare il comando 'show g'. Questo è software libero, e ognuno è libero di ridistribuirlo secondo certe condizioni; usare il comando 'show c' per maggiori dettagli.

Gli ipotetici comandi “show g” e “show c” mostreranno le parti appropriate della Licenza Pubblica Generica. Chiaramente, i comandi usati possono essere chiamati diversamente da “show g” e “show c” e possono anche essere selezionati con il mouse o attraverso un menù, o comunque sia pertinente al programma.

Se necessario, si deve anche far firmare al proprio datore di lavoro

ro (per chi lavora come programmatore) o alla propria scuola, per chi è studente, una “rinuncia al copyright” per il programma. Ecco un esempio con nomi fittizi:

Yoyodinamica SPA rinuncia con questo documento ad ogni diritto sul copyright del programma ‘Orcaloca’ (che s’inchina davanti ai compilatori) scritto da Giovanni Smanettone.

firma di Pinco Pallino, 1 Aprile 1989

Pinco Pallino, Presidente

I programmi coperti da questa Licenza Pubblica Generica non possono essere incorporati all’interno di programmi proprietari. Se il proprio programma è una libreria di funzioni, può essere più utile permettere di collegare applicazioni proprietarie alla libreria. Se si ha questa intenzione consigliamo di usare invece la GNU Library General Public License.

Licenza Pubblica Generica Attenuata (LGPL) del progetto GNU

Versione 2.1, Febbraio 1999

Copyright © 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Chiunque può copiare e distribuire copie letterali di questo documento di licenza, ma non ne è permessa la modifica.

[Questa è la prima versione rilasciata della Licenza Pubblica Generica Attenuata, e conta come successore della Licenza Pubblica Generica per Librerie del Progetto GNU, versione 2, da cui la versione numero 2.1]

(NdT: Questa è una traduzione italiana non ufficiale della Licenza Pubblica Generica Attenuata, LGPL. Non è pubblicata dalla Free Software Foundation e non ha valore legale nell'esprimere i termini di distribuzione del software che usa la licenza LGPL. Solo la versione originale inglese della licenza ha valore legale. Speriamo ad ogni modo che questa traduzione aiuti le persone di lingua italiana a comprendere meglio il significato della LGPL.)

Preambolo

Le licenze della maggior parte dei programmi hanno lo scopo di togliere all'utente la libertà di condividere e modificare il pro-

gramma stesso. Viceversa, le Licenze Pubbliche Generiche GNU sono intese a garantire la libertà di condividere e modificare il software libero, al fine di assicurare che i programmi siano liberi per tutti i loro utenti.

Questa Licenza, la Licenza Pubblica Generica Attenuata (LGPL), si applica a specifici pacchetti software, tipicamente librerie, della Free Software Foundation e di altri autori che decidono di usare questa Licenza. Chiunque può usare questa licenza, ma suggeriamo prima di valutare attentamente se questa licenza, piuttosto che la normale Licenza Pubblica Generica, sia la migliore strategia da usare per ogni specifico caso, sulla base delle seguenti spiegazioni.

Quando si parla di software libero (free software), ci si riferisce alla libertà, non al prezzo. Le nostre Licenze Pubbliche Generiche sono progettate per assicurarsi che ciascuno abbia la libertà di distribuire copie del software libero (e farsi pagare per questo, se lo si vuole); che ciascuno riceva il codice sorgente o che, se vuole, possa ottenerlo; che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi liberi; e che ciascuno sappia di poter fare queste cose.

Per proteggere i diritti dell'utente, abbiamo bisogno di imporre restrizioni che vietino ai distributori di negare tali diritti o di chiedere agli utenti di rinunciarvi. Queste restrizioni si traducono in determinate responsabilità a carico di chi distribuisce copie del software o di chi lo modifica.

Ad esempio, chi distribuisce copie di una libreria LGPL, sia gratis sia in cambio di un compenso, deve concedere ai destinatari tutti i diritti che ha ricevuto. Deve anche assicurarsi che i destinatari ricevano o possano ottenere il codice sorgente. Se è stato collegato altro codice alla libreria, deve fornire tutti questi codi-

ci ai destinatari, in modo che essi possano ricollegarli alla libreria dopo averla modificata e ricompilata. E deve mostrar loro queste condizioni della licenza, in modo che essi conoscano i propri diritti.

Tuteliamo i diritti dell'utente in due modi: (1) proteggendo la libreria attraverso il copyright, e (2) offrendo una licenza che dia il permesso legale di copiare, distribuire e modificare la libreria. Per proteggere ogni distributore, vogliamo rendere assolutamente chiaro che non esistono garanzie per la licenza libera. Inoltre, se la licenza viene modificata da qualcun altro e ridistribuita, gli acquirenti dovrebbero essere informati che quanto in loro possesso non è la versione originale, in modo che ogni problema eventualmente introdotto da altri non danneggi la reputazione dell'autore originario.

Infine, l'esistenza di ogni programma libero è costantemente sotto la minaccia dei brevetti sul software. Vogliamo esser certi che una azienda non possa effettivamente porre restrizioni sugli utenti di un programma libero tramite l'uso di licenze restrittive di qualche proprietario di brevetto. Perciò insistiamo sul fatto che qualsiasi licenza di brevetto ottenuta per una versione della libreria debba risultare coerente con la piena libertà d'uso specificata in questa licenza.

La maggior parte del software GNU, incluse alcune librerie, è coperto dalla normale Licenza Pubblica Generica (GPL) del Progetto GNU. Questa licenza, la Licenza Pubblica Generica Attenuata (LGPL), si applica a certe librerie specifiche ed è assai diversa dalla Licenza Pubblica Generica normale. Questa licenza viene usata per determinate librerie in modo da permettere il collegamento di tali librerie a programmi non liberi.

Quando un programma è collegato con una libreria, sia statica-

mente che usando una libreria condivisa, legalmente parlando la combinazione dei due elementi è un lavoro combinato, un derivato della libreria originale. Perciò la normale Licenza Pubblica Generica permette tale collegamento solo se l'intera combinazione risulta conforme ai propri criteri di libertà. La Licenza Pubblica Generica Attenuata consente criteri più rilassati per collegare altro codice alla libreria.

Questa licenza viene definita la Licenza Pubblica Generica «Attenuata» perché fa meno per proteggere la libertà dell'utente rispetto alla normale Licenza Pubblica Generica. Essa fornisce inoltre minori vantaggi agli sviluppatori di software libero nella competizione con programmi non liberi. Questi svantaggi sono la ragione per cui usiamo la Licenza Pubblica Generica per molte librerie. Tuttavia, la Licenza Pubblica Generica Attenuata fornisce dei vantaggi per certe circostanze speciali.

Ad esempio, in rare occasioni, può presentarsi la necessità particolare di incoraggiare l'uso più ampio possibile di una determinata libreria, in modo che divenga uno standard de facto. Onde raggiungere quest'obiettivo, i programmi non liberi devono essere in grado di utilizzare la libreria. Un caso più frequente è che la libreria libera svolga lo stesso compito di librerie non libere molto usate.

In questa situazione, ha poco senso limitare la libreria libera al solo software libero, quindi utilizziamo la Licenza Pubblica Generica Attenuata.

In altri casi, il permesso di usare una specifica libreria in programmi non liberi consente a un maggior numero di persone l'uso di un'ampia quantità di programmi liberi. Per esempio, il permesso di utilizzare la libreria C del Progetto GNU in programmi non liberi consente a molte più persone di usare l'intero sistema

operativo GNU, come pure della sua variante più comune, il sistema operativo GNU/Linux.

Sebbene la Licenza Pubblica Generica Attenuata tuteli la libertà degli utenti in misura minore, garantisce all'utente di un programma collegato alla Libreria la libertà e i mezzi per eseguire tale programma usando una versione modificata della Libreria.

Seguono i termini e le condizioni precise per la copia, la distribuzione e la modifica. Si faccia molta attenzione alla differenza tra "opera basata sulla libreria" e "opera che usa la libreria". La prima contiene codice derivato dalla libreria, mentre la seconda deve essere combinata con la libreria per poter funzionare.

Termini e condizioni per la copia, la distribuzione e la modifica

Questa Licenza si applica a ogni libreria software o altro programma che contenga una nota posta dal detentore del copyright o da altro soggetto autorizzato in cui si specifichi che tale libreria o programma vada distribuito secondo i termini della Licenza Pubblica Generica Attenuata (definita anche "questa Licenza"). Per "libreria" s'intende una raccolta di funzioni software e/o dati preparati in modo da poter essere facilmente collegati con programmi applicativi (che utilizzano alcune di queste funzioni e dati) così da formare degli eseguibili.

Il termine "Libreria" usato da qui in poi si riferisce a ogni tipo di libreria software o opera che sia stata distribuita in questi termini. L'espressione "un'opera basata sulla Libreria" indica sia la Libreria sia ogni opera derivativa come definito dalla legge sul diritto d'autore: ovvero, un'opera contenente la Libreria o una sua parte, sia inalterata sia con modifiche e/o tradotta direttamente in un altro linguaggio. (Da qui in avanti, la traduzione viene inclusa senza limitazioni nel termine "modifica".)

Per “codice sorgente” di un’opera s’intende la forma di codice usato di preferenza per apportare modifiche. Per una libreria, il codice sorgente completo è il codice sorgente di tutti i moduli contenuti, più ogni file associato per la definizione delle interfacce, più gli script utilizzati per controllare la compilazione e l’installazione della libreria.

Attività diverse dalla copia, distribuzione e modifica non sono coperte da questa Licenza e sono al di fuori della sua influenza. L’atto di eseguire un programma che usa la Libreria non viene limitato, e l’output di tale programma è coperto da questa Licenza solo nel caso in cui il contenuto costituisce un’opera basata sulla Libreria (indipendentemente dal fatto che sia stato creato utilizzando la Libreria). Se ciò corrisponda o meno al vero, dipende da cosa fa la Libreria e da cosa fa il programma che usa la Libreria.

1. È lecito copiare e distribuire copie letterali del codice sorgente completo della Libreria così come viene ricevuto, con qualsiasi mezzo, a condizione che venga riprodotta chiaramente su ogni copia un’appropriata nota per il copyright e per la mancanza di garanzie; che si mantengano intatti tutti i riferimenti a questa Licenza e all’assenza di ogni garanzia; e che si distribuisca una copia di questa Licenza insieme alla Libreria. Si può richiedere un pagamento per il trasferimento fisico di una copia, ed è anche possibile, a propria discrezione, offrire a pagamento una garanzia aggiuntiva.

2. È consentito modificare la propria copia o le copie della Libreria o qualsiasi sua parte, creando in questo modo un’opera basata sulla Libreria, e copiare o distribuire tali modifiche o tale opera secondo i termini del precedente comma 1, purché vengano soddisfatte tutte le seguenti condizioni:

- a) L'opera modificata deve essere a sua volta una libreria software.
- b) Bisogna inserire nei file modificati una chiara nota in cui si spieghi che avete cambiato il file e riporti la data di ogni modifica.
- c) Occorre fare in modo che l'opera venga concessa nella sua interezza in licenza gratuita ad ogni terza parte sotto i termini di questa Licenza.

d) Se una funzionalità della Libreria modificata implica che una funzione o una tabella dati vengano forniti da un programma applicativo che usa tale funzionalità, in casi diversi dal passaggio di argomenti quando la funzionalità viene invocata, allora bisogna accertarsi al meglio delle proprie possibilità che, nel caso l'applicazione non fornisca tale funzione o tabella, la funzionalità possa operare comunque ed esegua qualsiasi parte della propria funzione abbia ancora senso.

(Ad esempio, la funzione di una libreria per il calcolo delle radici quadrate ha un fine ben determinato indipendente dall'applicazione. Di conseguenza, il sotto-comma 2d richiede che ogni funzione fornita dall'applicazione o dalla tabella usata da tale funzione debbano essere opzionali: Qualora l'applicazione non le fornisca, la funzione radice quadrata deve comunque poter calcolare le radici quadrate.)

Questi requisiti si applicano all'opera modificata nella sua interezza. Se sezioni identificabili di questa opera non sono derivate dalla Libreria e possono essere ragionevolmente considerate indipendenti e opere separate in quanto tali, allora questa Licenza, e i suoi termini, non si applicano a quelle sezioni che vengano distribuite come opere separate. Ma quando tali sezioni sono distribuite in blocco come parte di un'opera basata sulla Libreria, la distri-

buzione dell'opera completa deve essere rilasciata sotto i termini di questa Licenza, i cui permessi per successivi licenziatari si estendono all'opera completa, e quindi ad ogni sua parte, indipendentemente da chi l'abbia scritta.

Così l'intento di questa sezione non è quello di accampare o contestare alcun diritto su opere scritte interamente da altri; piuttosto, l'intento è quello di esercitare il diritto al controllo della distribuzione di lavori derivati o collettivi basati sulla Libreria in questione.

In aggiunta, la semplice aggregazione con la Libreria di un'altra opera non basata sulla Libreria (o anche con un'opera basata sulla Libreria) su un mezzo di memorizzazione o distribuzione, non implica che l'altra opera ricada sotto l'influenza di questa Licenza.

3. È lecito decidere di applicare a una copia della Libreria i termini della normale Licenza Pubblica Generica GNU (GNU GPL) al posto di questa Licenza. Per farlo, è necessario cambiare tutti i riferimenti a questa Licenza, in modo che rimandino alla normale Licenza Pubblica Generica GNU versione 2, anziché a questa Licenza. (Se dovesse essere pubblicata una versione della Licenza Pubblica Generica GNU successiva alla 2, volendo si può specificare questa nuova versione). Non va cambiato nessun altro riferimento o nota. Una volta operato questo cambiamento su una determinata copia, esso diviene irreversibile e la Licenza Pubblica Generica GNU si applica a tutte le successive copie e opere derivate create a partire da tale copia. Questa opzione torna utile qualora si voglia copiare parte del codice della Libreria in un programma che non è una libreria.

4. È consentito copiare e distribuire la Libreria (o parti o derivati di essa, come espresso dal comma 2) sotto forma di codice ogget-

to o eseguibile secondo i termini dei precedenti commi 1 e 2, a condizione che venga allegato il corrispondente codice sorgente completo, in formato leggibile dal calcolatore, distribuito secondo quanto stabilito dai commi 1 e 2 su un mezzo comunemente utilizzato per lo scambio di software.

Nel caso la distribuzione di codice oggetto dovesse avvenire tramite accesso alla copia da un determinato luogo, allora l'offerta di analogo accesso per copiare il codice sorgente dal medesimo luogo soddisfa il requisito di distribuzione del codice sorgente, anche se terze parti non sono obbligate a copiare il sorgente insieme al codice oggetto.

5. Un programma che non contenga alcun derivato di nessuna porzione della Libreria, ma è progettato per lavorare con la Libreria attraverso compilazione o collegamento con questa, viene definito "un'opera che usa la Libreria". Tale opera, isolata, non è derivata dalla Libreria, e pertanto ricade al di fuori dell'influenza di questa Licenza.

Tuttavia, collegando "un'opera che usa la Libreria" con quest'ultima si crea un eseguibile che è derivato dalla Libreria stessa (poiché ne contiene delle parti), piuttosto che "un'opera che usa la Libreria". Di conseguenza, il codice eseguibile è coperto da questa Licenza. Il comma 6 illustra i termini per la distribuzione di questo tipo di eseguibili.

Quando "un'opera che usa la Libreria" utilizza materiale da un file di header che fa parte della Libreria, il codice oggetto dell'opera può essere un'opera derivata dalla Libreria anche se il codice sorgente non lo è. Per determinare questa condizione risulta particolarmente significativo il fatto che l'opera possa essere compilata

senza la Libreria, oppure nel caso l'opera sia una libreria essa stessa. La soglia per determinare questa distinzione non viene stabilita in modo preciso dalla legge.

Se tale file oggetto utilizza solo parametri numerici, schemi di strutture dati e accessori, e piccole macro-funzioni o piccole funzioni in linea (lunghe al massimo 10 righe), allora l'uso del file oggetto non è sottoposto a restrizioni, indipendentemente dal fatto che sia o meno un'opera derivata a livello legale. (Eseguibili che contengano tale codice oggetto in aggiunta a porzioni della Libreria sono comunque regolati dal comma 6).

Altrimenti, nel caso l'opera sia derivata dalla Libreria, si può distribuire il codice oggetto dell'opera in base ai termini del comma 6. Ogni eseguibile contenente quell'opera ricade comunque sotto i termini del comma 6, prescindendo dal fatto che siano direttamente collegati o meno alla Libreria stessa.

6. Come eccezione al comma precedente, si può combinare o collegare "un'opera che usa la Libreria" con quest'ultima onde creare un'opera che contenga porzioni della Libreria, e distribuire tale opera secondo termini di propria scelta, purché questi termini consentano la modifica dell'opera ad uso privato e il reverse engineering per il debugging delle modifiche.

Occorre includere in ogni copia dell'opera una chiara nota in cui si specifichi l'utilizzo della Libreria e il fatto che la Libreria e il suo impiego vengono regolati da questa Licenza. È obbligatorio fornire una copia di questa Licenza. Se durante l'esecuzione l'opera visualizza le note di copyright, insieme a queste bisogna mostrare le note di copyright della Libreria, oltre al riferimento diretto ad una copia di questa Licenza. È inoltre necessario fare una delle seguenti cose:

a) Fornire insieme all'opera il codice sorgente completo della Libreria in un formato leggibile dal calcolatore, comprese tutte le modifiche apportate (che devono essere distribuite secondo i termini previsti dai commi 1 e 2); e, nel caso l'opera sia un eseguibile collegato con la Libreria, fornire "l'opera che usa la Libreria" con il codice oggetto e/o sorgente completo, in modo che l'utente possa modificare la Libreria e poi ricollegare il tutto onde produrre un eseguibile modificato contenente la Libreria modificata. (È assodato che l'utente che dovesse cambiare il contenuto dei file di definizione della Libreria non sarà necessariamente in grado di ricompilare l'applicazione per usare tali definizioni modificate).

b) Usare un appropriato meccanismo di condivisione delle librerie per collegare la Libreria. Un meccanismo appropriato è quello che (1) durante l'esecuzione utilizza una copia della libreria già presente nel computer dell'utente, anziché copiare le funzioni della libreria nell'eseguibile, e (2) funzionerà correttamente con una versione modificata della libreria, se l'utente ne installa una, fintanto che la versione modificata non sia compatibile a livello di interfaccia con la versione con la quale è stata creata l'opera.

c) Allegare all'opera un'offerta scritta, valida per almeno 3 anni, per la fornitura allo stesso utente dei materiali specificati nel precedente sotto-comma 6a, ad un costo non superiore a quello di distribuzione.

d) Se la distribuzione dell'opera viene effettuata tramite accesso alla copia da un luogo specifico, va offerto analogo accesso alla copia dei materiali sopra specificati dallo stesso luogo.

e) Verificare che l'utente abbia già ricevuto una copia di questi materiali o che gliene sia già stata trasferita una copia.

Per un eseguibile, bisogna fornire ogni dato o programma di utilità necessario per ricreare l'eseguibile che forma "l'opera che usa la Libreria". Tuttavia, come eccezione particolare, tra i materiali da distribuire non vanno necessariamente inclusi tutti quelli normalmente distribuiti (in forma sorgente o binaria) con i principali componenti (compilatore, kernel e così via) del sistema operativo sul quale funziona l'eseguibile, a meno che tali componenti non siano distribuiti insieme all'eseguibile.

Può accadere che questo requisito contraddica le restrizioni dettate da licenze di altre librerie proprietarie normalmente non fornite con il sistema operativo.

Queste incongruenze comportano l'impossibilità di utilizzare insieme tali librerie e la Libreria in un eseguibile da distribuire.

7. È possibile inserire in un'unica libreria delle funzionalità che sono un'opera basata sulla Libreria, di fianco ad altre funzionalità non regolate da questa Licenza, e distribuire questa libreria combinata, purché venga comunque consentita la distribuzione separata dell'opera basata sulla Libreria e delle altre funzionalità di libreria, e posto che vengano rispettate le seguenti due condizioni:

a) Insieme alla libreria combinata, occorre fornire una copia della stessa opera basata sulla Libreria, non combinata con nessun'altra funzionalità di libreria. Questa deve essere distribuita rispettando i termini enunciati sopra.

b) Affiancare alla libreria combinata una chiara nota in cui viene specificato che parte di essa è un'opera basata sulla Libreria, spiegando altresì dove trovare la versione non combinata della stessa opera.

8. Non è consentito copiare, modificare, rilicenziare, collegare con o distribuire la Libreria se non nei termini espressamente enunciati in questa Licenza. Qualsiasi tentativo di copiare, modificare, rilicenziare, collegare con o distribuire la Libreria sotto altri termini non è valido e terminerà automaticamente i diritti ricevuti con questa Licenza. Tuttavia, ai quei soggetti che avessero ricevuto copie, o diritti, sotto i termini di questa Licenza non verrà terminata la licenza fintanto che tali soggetti ne rimangano in piena conformità.

9. L'utente non è tenuto ad accettare questa Licenza, poichè non l'ha firmata. In ogni caso, nessun altro documento garantisce il permesso di modificare o distribuire la Libreria o le opere da essa derivate. Queste azioni sono proibite dalla legge per chi non accetta questa Licenza. Di conseguenza, modificando o distribuendo la Libreria (o qualsiasi opera basata sulla Libreria), si indica l'accettazione di questa Licenza in tal senso, e quindi di tutti i suoi termini e condizioni relativamente a copia, distribuzione e modifica della Libreria o di opere basate su questa.

10. Ogni volta che la Libreria (o un'opera basata sulla Libreria) viene distribuita, il ricevente ottiene automaticamente una licenza d'uso da parte del licenziatario originario che regola la copia, la distribuzione, la modifica e il collegamento con la Libreria secondo i termini e le condizioni ivi specificate. Non è consentito imporre ulteriori restrizioni ai riceventi nell'esercizio dei propri diritti qui garantiti. Chi distribuisce programmi coperti da questa Licenza non è comunque tenuto a imporne il rispetto nei confronti di terze parti.

11. Se, a seguito di una sentenza di tribunale o di una imputazione per violazione di brevetto o per qualsiasi altro motivo (non limitatamente a questioni di brevetti), vengano imposte all'utente, sia dal tribunale sia da accordi tra le parti o altro, delle condizioni in contrasto con quanto stabilito da questa Licenza, tali condizioni non esimono alcun soggetto dal rispetto di questa Licenza. Nel caso non sia possibile distribuire un programma in un modo da soddisfare simultaneamente gli obblighi dettati da questa Licenza e altri obblighi ad essa pertinenti, non si potrà procedere ad alcuna distribuzione. Se, ad esempio, un brevetto vietasse a tutti quelli che ricevono direttamente o indirettamente la Libreria, la sua redistribuzione senza pagamento di diritti, allora l'unico modo per rispettare contemporaneamente tale brevetto e questa Licenza è quello di non distribuire affatto la Libreria.

Se una parte qualsiasi di questo comma venga ritenuta non valida o inapplicabile in una qualunque circostanza specifica, deve comunque essere applicato quanto espresso in questo comma, e in ogni altra circostanza va applicato questo comma nel suo complesso.

Non rientra nelle finalità di questo comma indurre l'utente ad infrangere alcun brevetto né altre rivendicazioni sul diritto di proprietà, né di contestare la validità di tali rivendicazioni. L'obiettivo di questo comma è unicamente quello di proteggere l'integrità del sistema di distribuzione dei programmi liberi implementato tramite l'utilizzo di licenze pubbliche. Molte persone hanno generosamente contribuito alla vasta gamma di programmi distribuiti attraverso questo sistema, basandosi sulla fedele applicazione di tale sistema. Spetta soltanto all'autore/donatore decidere se preferisca o meno distribuire il software tramite altri sistemi, e l'utente non può imporre tale scelta.

Questo comma punta a chiarire fino in fondo ciò che crediamo sia una conseguenza del resto di questa Licenza.

12. Se in alcuni paesi la distribuzione o l'impiego della Libreria sono limitati da brevetti o da interfacce coperte da copyright, il detentore del copyright originario che pone la Libreria sotto questa Licenza può aggiungere esplicite limitazioni geografiche alla distribuzione onde escluderne tali paesi, in modo da consentire la distribuzione soltanto in quei paesi non inclusi in queste restrizioni. In tal caso, le limitazioni geografiche vengono incorporate a tutti gli effetti nel testo di questa Licenza.

13. Di quando in quando Free Software Foundation potrebbe pubblicare versioni nuove o riviste della Licenza Pubblica Generica Attenuata (LGPL). Tali versioni saranno simili a questa nello spirito, ma potranno differire nei dettagli al fine di coprire problemi e situazioni nuove.

A ciascuna versione viene assegnato un numero identificativo. Se la Libreria specifica di essere coperta da una particolare versione di questa Licenza e "da qualsiasi versione successiva, l'utente può scegliere di aderire alle condizioni della versione specificata o a quelle di una successiva. Se la Libreria non specifica il numero della versione, l'utente può optare per una versione qualsiasi tra quelle pubblicate dalla Free Software Foundation.

14. Nel caso si voglia incorporare parti della Libreria in altri programmi liberi le cui condizioni di distribuzione siano incompatibili con queste, si può scrivere all'autore per chiederne l'autorizzazione. Per il software sotto il copyright della Free Software Foundation, occorre contattare quest'ultima; talvolta facciamo delle

eccezioni a queste regole. La nostra decisione sarà guidata da due finalità: preservare la libertà di tutti i prodotti derivati dal nostro software libero e promuovere la condivisione e il riutilizzo del software in generale.

Nessuna garanzia

15. POICHÈ LA LIBRERIA VIENE CONCESSA CON LICENZA GRATUITA, NON ESISTE ALCUNA GARANZIA PER LA LIBRERIA, NEI LIMITI CONSENTITI DALLE VIGENTI LEGGI. SE NON INDICATO DIVERSAMENTE PER ISCRITTO, IL DETENTORE DEL COPYRIGHT E LE ALTRE PARTI FORNISCONO IL PROGRAMMA “COSÌ COM’È”, SENZA ALCUN TIPO DI GARANZIA, NÉ ESPlicitA NÉ IMPLICITA; CIÒ INCLUDE, SENZA LIMITARSI A QUESTO, LA GARANZIA IMPLICITA DI COMMERCIALIZZABILITÀ E UTILIZZABILITÀ PER UNO SCOPO PARTICOLARE. TUTTI I RISCHI SU QUALITÀ E PRESTAZIONI DELLA LIBRERIA SONO A CARICO DELL’UTENTE. SE LA LIBRERIA DOVESSE RIVELARSI DIFETTOSA, L’UTENTE SI ASSUME L’ONERE DI OGNI MANUTENZIONE, RIPARAZIONE O CORREZIONE NECESSARIA.

16. NÉ IL DETENTORE DEL COPYRIGHT, NÉ ALTRE PARTI AUTORIZZATE A MODIFICARE E/O RIDISTRIBUIRE LA LIBRERIA SECONDO QUANTO STABILITO IN QUESTA LICENZA, SONO RESPONSABILI IN ALCUN MODO PER EVENTUALI DANNI NEI CONFRONTI DELL’UTENTE, A MENO CHE CIÒ NON SIA RICHIESTO DALLE LEGGI VIGENTI O SIA SPECIFICATO IN UN ACCORDO SCRITTO. SONO INCLUSI DANNI GENERICI, SPECIALI O INCIDENTALI, COME PURE I DANNI CONSEGUENTI DALL’USO O DALL’IMPOSSIBILITÀ DI USARE LA LIBRERIA (INCLUSO, MA SENZA LIMITARSI A QUESTO, LA PERDITA E LA CORRUZIONE DEI DATI, LE PERDITE SOSTENUTE DALL’UTENTE O DA TERZE PARTI E L’INCAPACITÀ DA PARTE DELLA LIBRERIA DI INTERAGIRE

CON ALTRO SOFTWARE), ANCHE NEL CASO IL DETENTORE O LE ALTRE PARTI SIANO STATE AVVISATE DELL'EVENTUALITÀ DI TALI DANNI.
Fine dei termini e delle condizioni

Come applicare questi termini a nuove librerie

Se si sviluppa una nuova libreria, e la si vuole rendere della maggiore utilità possibile per il pubblico, la cosa migliore è renderla libera, in modo che chiunque possa ridistribuirla e modificarla sotto questi termini (o, alternativamente, sotto i termini della normale Licenza Pubblica Generica).

Per applicare questi termini, basta inserire nella libreria le seguenti note. La procedura migliore è inserirle all'inizio di ogni file sorgente, per chiarire nel modo più efficace possibile l'assenza di garanzie; e ciascun file dovrebbe contenere almeno la nota di copyright e l'indicazione di dove poter reperire la nota per esteso.

Una riga per indicare il nome della libreria e dare un'idea di cosa faccia.

Copyright (C) anno nome dell'autore

Questa libreria è software libero; ne è concessa la ridistribuzione o la modifica secondo i termini della Licenza Pubblica Generica Attenuata GNU come pubblicata dalla Free Software Foundation; si può scegliere a piacimento la versione 2.1 della Licenza oppure una qualsiasi versione successiva.

Questa libreria è distribuita nella speranza possa mostrarsi utile, ma SENZA ALCUNA GARANZIA; senza neppure la garanzia implicita di COMMERCIALIZZABILITÀ o APPLICABILITÀ PER UN PAR-

TICOLARE SCOPO. Per maggiori dettagli si veda la Licenza Pubblica Generica Attenuata GNU.

Insieme a questa libreria, l'utente dovrebbe aver ricevuto copia della Licenza Pubblica Generica Attenuata GNU; in caso contrario, si può contattare la Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

È inoltre il caso di aggiungere informazioni per poter essere contattati tramite posta elettronica e cartacea.

Se necessario, occorre far firmare al proprio datore di lavoro (per chi lavora come programmatore) o al proprio istituto, per gli studenti, una “rinuncia al copyright” per la Libreria. Ecco un esempio contenente nomi fittizi:

Yoyodinamica SPA rinuncia con questo documento ad ogni diritto sul copyright della libreria 'Orcaloca' (una libreria per girarsi i pollici) scritto da Giovanni Smanettone.

firma di Pinco Pallino, 1 Aprile 1990

Pinco Pallino, Presidente

Questo è tutto!

Licenza per Documentazione Libera (FDL) del Progetto GNU

Versione 1.1, Marzo 2000

Copyright (C) 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Chiunque può copiare e distribuire copie letterali di questo documento di licenza, ma non ne è permessa la modifica.

(Ndt: Questa è una traduzione italiana non ufficiale della Licenza per Documentazione Libera, FDL. Non è pubblicata dalla Free Software Foundation e non ha valore legale nell'esprimere i termini di distribuzione del software che usa la licenza FDL. Solo la versione originale inglese della licenza ha valore legale. Speriamo ad ogni modo che questa traduzione aiuti le persone di lingua italiana a comprendere meglio il significato della FDL.)

0. Preambolo

Lo scopo di questa licenza è di rendere un manuale, un testo o altri documenti scritti “liberi” nel senso di assicurare a tutti la libertà effettiva di copiarli e redistribuirli, con o senza modifiche, a fini di lucro o no. In secondo luogo questa licenza prevede per autori ed editori il modo per ottenere il giusto riconoscimento del

proprio lavoro, preservandoli dall'essere considerati responsabili per modifiche apportate da altri.

Questa licenza è un “copyleft”: ciò vuol dire che i lavori che derivano dal documento originale devono essere ugualmente liberi. È il complemento alla Licenza Pubblica Generale GNU, che è una licenza di tipo “copyleft” pensata per il software libero.

Abbiamo progettato questa licenza al fine di applicarla alla documentazione del software libero, perché il software libero ha bisogno di documentazione libera: un programma libero dovrebbe accompagnarsi a manuali che forniscano la stessa libertà del software. Ma questa licenza non è limitata alla documentazione del software; può essere utilizzata per ogni testo che tratti un qualsiasi argomento e al di là dell'avvenuta pubblicazione cartacea. Raccomandiamo principalmente questa licenza per opere che abbiano fini didattici o per manuali di consultazione.

1. Applicabilità e definizioni

Questa licenza si applica a qualsiasi manuale o altra opera che contenga una nota messa dal detentore del copyright che dica che si può distribuire nei termini di questa licenza. Con “Documento”, in seguito ci si riferisce a qualsiasi manuale o opera. Ogni fruitore è un destinatario della licenza e viene indicato con “voi”.

Una “versione modificata” di un documento è ogni opera contenente il documento stesso o parte di esso, sia riprodotto alla lettera che con modifiche, oppure traduzioni in un'altra lingua.

Una “sezione secondaria” è un'appendice cui si fa riferimento o una premessa del documento e riguarda esclusivamente il rapporto dell'editore o dell'autore del documento con l'argomento gene-

rale del documento stesso (o argomenti affini) e non contiene nulla che possa essere compreso nell'argomento principale. (Per esempio, se il documento è in parte un manuale di matematica, una sezione secondaria non può contenere spiegazioni di matematica). Il rapporto con l'argomento può essere un tema collegato storicamente con il soggetto principale o con soggetti affini, o essere costituito da argomentazioni legali, commerciali, filosofiche, etiche o politiche pertinenti.

Le “sezioni non modificabili” sono alcune sezioni secondarie i cui titoli sono esplicitamente dichiarati essere sezioni non modificabili, nella nota che indica che il documento è realizzato sotto questa licenza.

I “testi copertina” sono dei brevi brani di testo che sono elencati nella nota che indica che il documento è realizzato sotto questa licenza.

Una copia “trasparente” del documento indica una copia leggibile da un calcolatore, codificata in un formato le cui specifiche sono disponibili pubblicamente, i cui contenuti possono essere visti e modificati direttamente, ora e in futuro, con generici editor di testi o (per immagini composte da pixel) con generici editor di immagini o (per i disegni) con qualche editor di disegni ampiamente diffuso, e la copia deve essere adatta al trattamento per la formattazione o per la conversione in una varietà di formati atti alla successiva formattazione. Una copia fatta in un altro formato di file trasparente il cui markup è stato progettato per intralciare o scoraggiare modifiche future da parte dei lettori non è trasparente. Una copia che non è trasparente è “opaca”.

Esempi di formati adatti per copie trasparenti sono l'ASCII puro

senza markup, il formato di input per Texinfo, il formato di input per LaTeX, SGML o XML accoppiati ad una DTD pubblica e disponibile, e semplice HTML conforme agli standard e progettato per essere modificato manualmente. Formati opachi sono PostScript, PDF, formati proprietari che possono essere letti e modificati solo con word processor proprietari, SGML o XML per cui non è in genere disponibile la DTD o gli strumenti per il trattamento, e HTML generato automaticamente da qualche word processor per il solo output.

La “pagina del titolo” di un libro stampato indica la pagina del titolo stessa, più qualche pagina seguente per quanto necessario a contenere in modo leggibile, il materiale che la licenza prevede che compaia nella pagina del titolo. Per opere in formati in cui non sia contemplata esplicitamente la pagina del titolo, con “pagina del titolo” si intende il testo prossimo al titolo dell’opera, precedente l’inizio del corpo del testo.

2. Copie letterali

Si può copiare e distribuire il documento con l’ausilio di qualsiasi mezzo, per fini di lucro e non, fornendo per tutte le copie questa licenza, le note sul copyright e l’avviso che questa licenza si applica al documento, e che non si aggiungono altre condizioni al di fuori di quelle della licenza stessa. Non si possono usare misure tecniche per impedire o controllare la lettura o la produzione di copie successive alle copie che si producono o distribuiscono. Però si possono ricavare compensi per le copie fornite. Se si distribuiscono un numero sufficiente di copie si devono seguire anche le condizioni della sezione 3.

Si possono anche prestare copie e con le stesse condizioni sopra menzionate possono essere utilizzate in pubblico.

3. Copiare in notevoli quantità

Se si pubblicano a mezzo stampa più di 100 copie del documento, e la nota della licenza indica che esistono uno o più testi copertina, si devono includere nelle copie, in modo chiaro e leggibile, tutti i testi copertina indicati: il testo della prima di copertina in prima di copertina e il testo di quarta di copertina in quarta di copertina. Ambedue devono identificare l'editore che pubblica il documento. La prima di copertina deve presentare il titolo completo con tutte le parole che lo compongono egualmente visibili ed evidenti. Si può aggiungere altro materiale alle copertine. Il copiare con modifiche limitate alle sole copertine, purché si preservino il titolo e le altre condizioni viste in precedenza, è considerato alla stregua di copiare alla lettera.

Se il testo richiesto per le copertine è troppo voluminoso per essere riprodotto in modo leggibile, se ne può mettere una prima parte per quanto ragionevolmente può stare in copertina, e continuare nelle pagine immediatamente seguenti.

Se si pubblicano o distribuiscono copie opache del documento in numero superiore a 100, si deve anche includere una copia trasparente leggibile da un calcolatore per ogni copia o menzionare per ogni copia opaca un indirizzo di una rete di calcolatori pubblicamente accessibile in cui vi sia una copia trasparente completa del documento, spogliato di materiale aggiuntivo, e a cui si possa accedere anonimamente e gratuitamente per scaricare il documento usando i protocolli standard e pubblici generalmente usati. Se si adotta l'ultima opzione, si deve prestare la giusta attenzione, nel momento in cui si inizia la distribuzione in quantità elevata di copie opache, ad assicurarsi che la copia trasparente rimanga accessibile all'indirizzo stabilito fino ad almeno un anno di

distanza dall'ultima distribuzione (direttamente o attraverso rivenditori) di quell'edizione al pubblico.

È caldamente consigliato, benché non obbligatorio, contattare l'autore del documento prima di distribuirne un numero considerevole di copie, per metterlo in grado di fornire una versione aggiornata del documento.

4. Modifiche

Si possono copiare e distribuire versioni modificate del documento rispettando le condizioni delle precedenti sezioni 2 e 3, purché la versione modificata sia realizzata seguendo scrupolosamente questa stessa licenza, con la versione modificata che svolge il ruolo del “documento”, così da estendere la licenza sulla distribuzione e la modifica a chiunque ne possieda una copia. Inoltre nelle versioni modificate si deve:

* A. Usare nella pagina del titolo (e nelle copertine se ce ne sono) un titolo diverso da quello del documento, e da quelli di versioni precedenti (che devono essere elencati nella sezione storia del documento ove presenti). Si può usare lo stesso titolo di una versione precedente se l'editore di quella versione originale ne ha dato il permesso.

* B. Elencare nella pagina del titolo, come autori, una o più persone o gruppi responsabili in qualità di autori delle modifiche nella versione modificata, insieme ad almeno cinque fra i principali autori del documento (tutti gli autori principali se sono meno di cinque).

* C. Dichiarare nella pagina del titolo il nome dell'editore della versione modificata in qualità di editore.

* D. Conservare tutte le note sul copyright del documento originale.

- * E. Aggiungere un'appropriata licenza per le modifiche di seguito alle altre licenze sui copyright.
- * F. Includere immediatamente dopo la nota di copyright, un avviso di licenza che dia pubblicamente il permesso di usare la versione modificata nei termini di questa licenza, nella forma mostrata nell'addendum alla fine di questo testo.
- * G. Preservare in questo avviso di licenza l'intera lista di sezioni non modificabili e testi copertina richieste come previsto dalla licenza del documento.
- * H. Includere una copia non modificata di questa licenza.
- * I. Conservare la sezione intitolata "Storia", e il suo titolo, e aggiungere a questa un elemento che riporti al minimo il titolo, l'anno, i nuovi autori, e gli editori della versione modificata come figurano nella pagina del titolo. Se non ci sono sezioni intitolate "Storia" nel documento, createne una che riporti il titolo, gli autori, gli editori del documento come figurano nella pagina del titolo, quindi aggiungete un elemento che descriva la versione modificata come detto in precedenza.
- * J. Conservare l'indirizzo in rete riportato nel documento, se c'è, al fine del pubblico accesso ad una copia trasparente, e possibilmente l'indirizzo in rete per le precedenti versioni su cui ci si è basati. Questi possono essere collocati nella sezione "Storia". Si può omettere un indirizzo di rete per un'opera pubblicata almeno quattro anni prima del documento stesso, o se l'originario editore della versione cui ci si riferisce ne dà il permesso.
- * K. In ogni sezione di "Ringraziamenti" o "Dediche", si conservino il titolo, il senso, il tono della sezione stessa.
- * L. Si conservino inalterate le sezioni non modificabili del documento, nei propri testi e nei propri titoli. I numeri della sezione o equivalenti non sono considerati parte del titolo della sezione.

- * M. Si cancelli ogni sezione intitolata “Riconoscimenti”. Solo questa sezione può non essere inclusa nella versione modificata.
- * N. Non si modifichi il titolo di sezioni esistenti come “miglioria” o per creare confusione con i titoli di sezioni non modificabili.

Se la versione modificata comprende nuove sezioni di primaria importanza o appendici che ricadono in “sezioni secondarie”, e non contengono materiale copiato dal documento, si ha facoltà di rendere non modificabili quante sezioni si voglia. Per fare ciò si aggiunga il loro titolo alla lista delle sezioni immutabili nella nota di copyright della versione modificata. Questi titoli devono essere diversi dai titoli di ogni altra sezione.

Si può aggiungere una sezione intitolata “Riconoscimenti”, a patto che non contenga altro che le approvazioni alla versione modificata prodotte da vari soggetti – per esempio, affermazioni di revisione o che il testo è stato approvato da una organizzazione come la definizione normativa di uno standard.

Si può aggiungere un brano fino a cinque parole come Testo Copertina, e un brano fino a 25 parole come Testo di Retro Copertina, alla fine dell’elenco dei Testi Copertina nella versione modificata. Solamente un brano del Testo Copertina e uno del Testo di Retro Copertina possono essere aggiunti (anche con adattamenti) da ciascuna persona o organizzazione. Se il documento include già un testo copertina per la stessa copertina, precedentemente aggiunto o adattato da voi o dalla stessa organizzazione nel nome della quale si agisce, non se ne può aggiungere un altro, ma si può rimpiazzare il vecchio ottenendo l’esplicita autorizzazione dall’editore precedente che aveva aggiunto il testo copertina.

L’autore/i e l’editore/i del “documento” non ottengono da questa licenza il permesso di usare i propri nomi per pubblicizzare la ver-

sione modificata o rivendicare l'approvazione di ogni versione modificata.

5. Unione di documenti

Si può unire il documento con altri realizzati sotto questa licenza, seguendo i termini definiti nella precedente sezione 4 per le versioni modificate, a patto che si includa l'insieme di tutte le Sezioni Invarianti di tutti i documenti originali, senza modifiche, e si elenchino tutte come Sezioni Invarianti della sintesi di documenti nella licenza della stessa.

Nella sintesi è necessaria una sola copia di questa licenza, e multiple sezioni invarianti possono essere rimpiazzate da una singola copia se identiche. Se ci sono multiple Sezioni Invarianti con lo stesso nome ma contenuti differenti, si renda unico il titolo di ciascuna sezione aggiungendovi alla fine e fra parentesi, il nome dell'autore o editore della sezione, se noti, o altrimenti un numero distintivo. Si facciano gli stessi aggiustamenti ai titoli delle sezioni nell'elenco delle Sezioni Invarianti nella nota di copyright della sintesi.

Nella sintesi si devono unire le varie sezioni intitolate "storia" nei vari documenti originali di partenza per formare una unica sezione intitolata "storia"; allo stesso modo si unisca ogni sezione intitolata "Ringraziamenti", e ogni sezione intitolata "Dediche". Si devono eliminare tutte le sezioni intitolate "Riconoscimenti".

6. Raccolte di documenti

Si può produrre una raccolta che consista del documento e di altri realizzati sotto questa licenza; e rimpiazzare le singole copie di questa licenza nei vari documenti con una sola inclusa nella raccolta,

solamente se si seguono le regole fissate da questa licenza per le copie alla lettera come se si applicassero a ciascun documento.

Si può estrarre un singolo documento da una raccolta e distribuirlo individualmente sotto questa licenza, solo se si inserisce una copia di questa licenza nel documento estratto e se si seguono tutte le altre regole fissate da questa licenza per le copie alla lettera del documento.

7. Raccogliere insieme a lavori indipendenti

Una raccolta del documento o sue derivazioni con altri documenti o lavori separati o indipendenti, all'interno di o a formare un archivio o un supporto per la distribuzione, non è una “versione modificata” del documento nella sua interezza, se non ci sono copyright per l'intera raccolta. Ciascuna raccolta si chiama allora “aggregato” e questa licenza non si applica agli altri lavori contenuti in essa che ne sono parte, per il solo fatto di essere raccolti insieme, qualora non siano però loro stessi lavori derivati dal documento.

Se le esigenze del Testo Copertina della sezione 3 sono applicabili a queste copie del documento allora, se il documento è inferiore ad un quarto dell'intero aggregato i Testi Copertina del documento possono essere piazzati in copertine che delimitano solo il documento all'interno dell'aggregato. Altrimenti devono apparire nella copertina dell'intero aggregato.

8. Traduzioni

La traduzione è considerata un tipo di modifica, e di conseguenza si possono distribuire traduzioni del documento seguendo i termini della sezione 4. Rimpiazzare sezioni non modificabili con traduzioni richiede un particolare permesso da parte dei detento-

ri del diritto d'autore, ma si possono includere traduzioni di una o più sezioni non modificabili in aggiunta alle versioni originali di queste sezioni immutabili. Si può fornire una traduzione della presente licenza a patto che si includa anche l'originale versione inglese di questa licenza. In caso di discordanza fra la traduzione e l'originale inglese di questa licenza la versione originale inglese prevale sempre.

9. Termini

Non si può applicare un'altra licenza al documento, copiarlo, modificarlo, o distribuirlo al di fuori dei termini espressamente previsti da questa licenza. Ogni altro tentativo di applicare un'altra licenza al documento, copiarlo, modificarlo, o distribuirlo è deprecato e pone fine automaticamente ai diritti previsti da questa licenza. Comunque, per quanti abbiano ricevuto copie o abbiano diritti coperti da questa licenza, essi non ne cessano se si rimane perfettamente coerenti con quanto previsto dalla stessa.

10. Revisioni future di questa licenza

La Free Software Foundation può pubblicare nuove, rivedute versioni della Licenza per Documentazione Libera GNU volta per volta. Qualche nuova versione potrebbe essere simile nello spirito alla versione attuale ma differire in dettagli per affrontare nuovi problemi e concetti. Si veda <http://www.gnu.org/copyleft>.

Ad ogni versione della licenza viene dato un numero che distingue la versione stessa. Se il documento specifica che si riferisce ad una versione particolare della licenza contraddistinta dal numero o "ogni versione successiva", si ha la possibilità di seguire termini e condizioni sia della versione specificata che di ogni versione successiva pubblicata (non come bozza) dalla Free Software Founda-

tion. Se il documento non specifica un numero di versione particolare di questa licenza, si può scegliere ogni versione pubblicata (non come bozza) dalla Free Software Foundation.

ADDENDUM: Come usare questa licenza per i vostri documenti

Per applicare questa licenza ad un documento che si è scritto, si includa una copia della licenza nel documento e si inserisca il seguente avviso subito dopo la pagina del titolo:

Copyright (c) anno nome.

È garantito il permesso di copiare, distribuire e/o modificare questo documento seguendo i termini della Licenza per Documentazione Libera GNU, Versione 1.1 o ogni versione successiva pubblicata dalla Free Software Foundation; con le Sezioni Non Modificabili ELEN-CARNE I TITOLI, con i Testi Copertina ELENCO, e con i Testi di Retro Copertina ELENCO. Una copia della licenza è acclusa nella sezione intitolata “Licenza per Documentazione Libera GNU”.

Se non ci sono Sezioni non Modificabili, si scriva “senza Sezioni non Modificabili” invece di dire quali sono non modificabili. Se non c'è Testo Copertina, si scriva “nessun Testo Copertina” invece di “il testo Copertina è ELENCO”; e allo stesso modo si operi per il Testo di Retro Copertina.

Se il vostro documento contiene esempi non banali di programma in codice sorgente si raccomanda di realizzare gli esempi contemporaneamente applicandovi anche una licenza di software libero di vostra scelta, come ad esempio la Licenza Pubblica Generale GNU, al fine di permetterne l'uso come software libero.

Parte Terza

Risorse utili

ASSOLI: Progetti operativi in Italia e in Europa

Introduzione

L'Associazione Software Libero (<http://www.softwarelibero.it/>) nasce nel novembre 2000 ponendosi come scopi di base la corretta informazione e la diffusione della conoscenza secondo quanto definito nel 1984 da Richard Stallman all'atto della creazione della Free Software Foundation: libertà di utilizzo del software, libertà di studio, libertà di modifica, libertà di redistribuzione. Il tutto poggia su un presupposto: l'accesso al codice sorgente. Sono stati sufficienti tre anni di attività perché fosse chiaro come le principali minacce a questo genere di libertà non derivassero esclusivamente dalle holding dell'informatica, promulgatrici dagli Anni Ottanta in avanti di una politica di chiusura delle informazioni. Le minacce, ad oggi, arrivano anche dalle istituzioni che, in ritardo e a volte con scarsa cognizione di causa, legiferano sotto la spinta di lobby interessate alla preservazione dei propri privilegi di mercato. Ed ecco che non si va più solo a ledere la libertà di sviluppo del software, elemento che già di per sé dovrebbe indurre a una riflessione, ma anche le libertà fondamentali dei cittadini, identificabili con quella di espressione e parola. Perché provvedimenti come la brevettazione delle invenzioni immateriali (che coincide con il software) e la direttiva europea sull'armonizzazione del diritto d'autore insinuano proprio questo genere di rischi. Danneggiando sviluppatori e utenti, tecnici e profani.

Ma a volte capita anche che un movimento nato spontaneamente agli albori dell'era digitale e poi ufficializzatosi per difendersi dalle aggressioni delle regole del business, qual è il software libero, penetri le mura di quelle stesse istituzioni portando, seppur lentamente e con difficoltà di comprensione, i propri valori al vaglio di amministrazioni centrali e periferiche. Contribuendo così non solo alla veicolazione del proprio messaggio intrinseco, ma anche a una maggiore democratizzazione degli enti pubblici attraverso l'accesso ai dati e alle informazioni. E, non ultimo, a volte anche al superamento del gap tecnologico tra il primo mondo, l'Occidente industrializzato e potente, e gli altri mondi, quelli dai capitali limitati e dall'assenza di tecnologia capillare o, quanto meno, diffusa.

Sono questi i motivi per cui di seguito vengono presentati tre progetti dell'Associazione Software Libero, tre cavalli di battaglia attraverso i quali scardinare, o almeno tentare, la visione imperante dell'informatica legata al pagamento delle licenze, alla mera esecuzione dei programmi, alla limitazione delle informazioni per trasformare gli utenti in 'pigiatori' di tasti e icone a cui manca però la conoscenza di fondo, quella che si cela dietro alle interfacce grafiche e che costituisce uno dei presupposti della libertà.

Parte dei testi riportati sono presenti sul sito dell'Associazione, sono liberamente consultabili e altrettanto liberamente sono veicolabili secondo quanto riportato in calce alle pagine web: "la copia letterale e la distribuzione del materiale qui raccolto nella sua integrità sono permesse con qualsiasi mezzo, a condizione che questa nota sia riprodotta (se non diversamente indicato)".

EUCD (European Union Copyright Directive)

L'EUCD viene contrassegnata come legge comunitaria 29/2002,

e le sue intenzioni sono l'armonizzazione della disciplina relativa al diritto d'autore. Peccato che tale provvedimento, recepito in Italia a fine marzo 2003 per decreto legislativo ed entrato in vigore il 29 aprile successivo, non miri tanto a tutelare chi, teoricamente, dovrebbe beneficiare dalla sua introduzione, cioè gli autori. Il suo obiettivo è invece la preservazione di benefici già esistenti (quelli imposti di fatto dalle multinazionali dell'informatica e dell'entertainment, tanto per citarne due) e semmai radicalizzarli attraverso recrudescenze legislative.

L'EUCD è la direttiva della Comunità europea nata per uniformare la legislazione sul diritto d'autore in vigore nei Paesi membri. Si tratta di un argomento estremamente importante per la società, perché ogni persona ha a che fare con il diritto d'autore ogni volta che accede ad una qualunque opera, documento o informazione.

Il diritto d'autore è un insieme di leggi che forniscono agli autori alcune prerogative sulle proprie creazioni (come il monopolio sulla riproduzione). L'esistenza di tali leggi è giustificata dal fatto che esse incoraggiano la produzione di nuove opere, favorendo la diffusione del sapere e il progresso sociale:

(<http://www.biblio.liuc.it:8080/biblio/liucpap/pdf/44.pdf>).

L'EUCD introduce nuove norme che ampliano il diritto d'autore, ma di fatto ne contraddicono le finalità positive. Concede nuovi privilegi legali ai colossi del settore, senza però offrire alcuna nuova garanzia agli utenti. Questa impostazione sposta il bilanciamento dei diritti e dei doveri, favorendo i grossi produttori a spese di tutti coloro che, in vario modo, utilizzano le moderne tecnologie. Contiene quindi molte norme pericolose, tutte riconducibili a un problema fondamentale: la tutela legale per le "misure tecnologiche di protezione", ovvero per i sistemi che regolano l'accesso e la copia di

materiali coperti da diritto d'autore. La "tutela legale" implica che ogni tentativo di aggirare queste misure diventa reato.

Sancisce quindi un nuovo potere per gli editori: quello di ricorrere a sistemi digitali che stabiliscono in che modo gli utenti possano utilizzare le opere possedute (come e-book, CD contenenti musica o dati, DVD).

Questo significa che domani assisteremo alla diffusione di e-book a tempo, che diventano inutilizzabili dopo un certo periodo, e non possono essere stampati o ceduti a parenti o amici; CD musicali che non si possono copiare, o memorizzare sul computer o sul lettore MP3 portatile; film in DVD che si possono guardare solo in certi Paesi e con certi sistemi operativi; programmi che automaticamente cancellano dal proprio PC i file ritenuti "illegali"; computer, periferiche e sistemi operativi che si rifiutano di leggere dati ritenuti "non autorizzati". L'elenco potrebbe continuare ed è potenzialmente molto ampio. Una applicazione estensiva di questi sistemi potrà togliere agli utenti ogni controllo sul funzionamento delle macchine in loro possesso.

Non si tratta di semplici ipotesi: tutto questo avviene già oggi. Ma l'EUCD richiede che gli Stati europei difendano queste misure tecnologiche, creando leggi apposite: domani, quindi, ogni tentativo di aggirare le vessazioni di questi sistemi di protezione potrebbe essere punito con il carcere; chi crea programmi che leggono certi tipi di file potrebbe commettere un reato; anche chi solamente discute su come evitare una limitazione tecnologica potrebbe rischiare la galera. Le persone accusate potrebbero essere punite anche se non avessero mai commesso atti oggi illeciti perché ritenuti violazioni del diritto d'autore.

Con l'applicazione dell'EUCD, alcuni casi recentemente balzati all'onore delle cronache avrebbero avuto conseguenze diverse: il

creatore del DeCSS sarebbe stato condannato; chi ha scoperto come superare le limitazioni dei CD anti-copia usando un pennarello sarebbe un criminale; anche chi utilizza o semplicemente rende note queste invenzioni correrebbe il rischio di ritorsioni legali.

Negli Stati Uniti questo scenario è già realtà, a causa del Digital Millennium Copyright Act (DMCA): una legge che ha permesso a varie aziende di ottenere arresti, intimidazioni e censure che hanno colpito utenti, programmatori, ricercatori. E le norme del DMCA sono le stesse previste dall'EUCD.

Con le norme introdotte dalla direttiva, diventa illegale l'aggiramento di tutte le "misure tecnologiche" (anche se facilmente superabili) che regolano l'accesso e la copia delle opere digitali, e diventa illegale l'offerta di informazioni e servizi, o la creazione di programmi che possano facilitare tale aggiramento. Gli autori/editori possono proibire agli utenti di cedere o rivendere le opere digitali, come software o e-book, regolarmente acquistate attraverso Internet, e possono controllarne qualunque diffusione.

Agli utenti non viene riconosciuta alcuna garanzia di poter utilizzare in modo ragionevole le opere in formato digitale. Il riconoscimento legale delle "misure tecnologiche" di protezione sancisce, di fatto, l'introduzione di un nuovo privilegio per i detentori dei diritti sulle opere: la possibilità di poter influire sull'utilizzo delle opere stesse. Infatti le "misure tecnologiche" dichiarate intoccabili dall'EUCD potrebbero imporre restrizioni estremamente severe per gli utenti, ed esse non potrebbero essere aggirate per alcun motivo. Si pensi ai film in DVD che possono essere guardati solamente in certi Paesi, agli e-book che non possono essere stampati, o ai cosiddetti "CD anti-copia" che non possono essere ascoltati su computer: queste vere e proprie truffe ai danni degli utenti verrebbero protette dalla legge, e rese inaggirabili.

A rischio anche la possibilità di scegliere quale software utilizzare per gestire i propri dati. La creazione di software interoperante richiede il superamento delle misure tecnologiche che proteggono i formati dati. Questa procedura è indispensabile per creare, per esempio, programmi che leggano i DVD, o altri documenti (anche di propria creazione) criptati, protetti da password o comunque memorizzati con alterazioni (anche molto semplici) che ne impediscano una lettura diretta. L'aggiramento delle misure tecnologiche, tuttavia, è vietato dall'EUCD – e quindi la direttiva di fatto riserva ad una sola azienda la possibilità di creare applicazioni che gestiscano un formato dati da essa creato. Gli utenti potrebbero perdere qualunque possibilità di scelta.

Verrebbe altresì negata la possibilità di cedere o rivendere i materiali digitali ottenuti attraverso Internet. Questo rende impossibile la nascita di un mercato degli e-book o del software “usati”, che porti a una riduzione dei prezzi come avvenuto nel mercato del libro tradizionale. Inoltre, ogni documento diffuso via Internet potrebbe essere censurato in qualunque momento dalla sua fonte, l'autore o l'editore, e nessuna delle persone che ne abbia ottenuto legalmente una copia avrebbe il diritto di renderla nota in alcun modo. Questo pone dei gravi rischi alla futura possibilità di accesso a materiale di rilevanza storica e documentaristica.

Sarà impossibile sapere se i programmi utilizzati siano sicuri o meno. Le informazioni sulle falle e difetti (bug) del software potrebbero agevolare l'aggiramento di “misure tecnologiche” difettose. Tali informazioni potrebbero essere quindi censurate, e gli utenti potrebbero essere tenuti all'oscuro dei problemi dei programmi utilizzati – con grande vantaggio di varie aziende produttrici di software proprietario, non più costrette a correggere i problemi dei propri prodotti.

La libertà di espressione su Internet sarà in grave pericolo. Qualunque informazione in grado di agevolare l'elusione di misure tecnologiche potrebbe essere dichiarata illegale, con gravi conseguenze sulla libertà di espressione e di stampa. Inoltre, le informazioni e i programmi resi illegali dall'EUCD potrebbero essere rimossi da Internet in modo estremamente rapido, con atti di censura che non richiedono l'intervento di un tribunale. Infatti, a causa della già citata direttiva sul commercio elettronico, ogni ISP (Internet Service Provider) che ospita le pagine Web degli utenti verrebbe di fatto costretto a soddisfare le richieste di oscuramento (più o meno motivate) provenienti dalle grosse aziende. Agli utenti verrebbero lasciate ben poche garanzie e possibilità di sfuggire alla censura.

Agli sviluppatori viene proibita la creazione di software in grado di interoperare con altri programmi e sistemi operativi proprietari: per produrre software interoperante, infatti, è necessario studiare il comportamento del software originario, aggirando le misure tecnologiche che rendono difficoltosa la lettura dei formati di scambio dei dati. Ma l'EUCD rende illegale questa pratica di aggiramento – e quindi un'azienda che subisca la concorrenza di un nuovo programma in grado di gestire gli stessi dati potrebbe denunciarne il creatore per il reato di “elusione di misure tecnologiche”. Uno sviluppatore che superi una misura tecnologica per garantire l'interoperabilità potrebbe essere incarcerato, anche se egli non avesse mai compiuto alcuna violazione del diritto d'autore. Inoltre, i programmi (liberi e non) ritenuti scomodi da qualche azienda potrebbero essere rimossi da Internet con una semplice telefonata intimidatoria all'ISP che ne ospita il sito, magari con l'accusa di essere degli strumenti in grado di agevolare l'elusione di misure tecnologiche. Tutto questo porta direttamente al monopolio legale sui formati dei dati.

Da notare, infine, che gli studi su crittografia e sicurezza sono basati essenzialmente sull'analisi della robustezza del software e degli algoritmi; questa analisi viene effettuata eseguendo dei tentativi di aggiramento delle misure tecnologiche di protezione. Purtroppo, tale pratica è vietata dall'EUCD – ed anche la comunicazione dei dati su questi studi è dichiarata illegale, e censurabile con estrema facilità. Chiunque aggiri delle misure tecnologiche, o diffonda informazioni su questo argomento, potrebbe essere arrestato, pur senza aver mai compiuto violazioni del diritto d'autore.

Tali limitazioni rappresentano un evidente ostacolo alla libertà di ricerca, e frenano inevitabilmente i progressi nel campo della crittografia e della sicurezza informatica: fino ad oggi, queste attività sono state svolte alla luce del sole, ed hanno portato enormi benefici per il miglioramento del software disponibile; ma, a causa dei divieti previsti dall'EUCD, le ricerche su crittografia e sicurezza diventerebbero sostanzialmente illegali e per questo verrebbero trattate solo “sotterraneamente”, per scopi tutt'altro che leciti (gli stessi che l'EUCD cerca di contrastare).

EUCD in Italia

Per l'Italia il pericolo appare assai concreto e vicino: infatti è diventato esecutivo il decreto legislativo per il recepimento dell'EUCD. Essendo nato da una delega ottenuta dal Governo, il decreto finale potrebbe essere approvato senza alcuna discussione in Parlamento. Da quando ha iniziato a circolare, la bozza del decreto ha destato notevole interesse per un suo aspetto in particolare: gli aumenti di prezzo previsti per i supporti di memorizzazione come CD-R e CD-RW, causati da un incremento delle tasse destinate alla SIAE. La rivista AFDigitale ha addirittura indetto una petizione on-line per l'annullamento dei rincari (<http://www.edisport.it/edi>

sport/afdigitale/petizione.nsf/Editoriale?Openpage). Tuttavia, per quanto estremamente discutibile, la questione rincari appare paradossalmente il problema meno grave della bozza di decreto legislativo. Tale decreto, infatti, comprende soprattutto le pericolose innovazioni previste dall'EUCD:

- rende estremamente problematica e complessa la tutela dei diritti degli utenti, specie contro gli abusi legati a “misure tecnologiche” troppo restrittive;
- rende pericolosa e potenzialmente illegale la produzione di software interoperante, specie se libero: i creatori di applicazioni poco gradite a qualche grosso produttore di software proprietario rischiano fino a tre anni di carcere;
- rende illegale la ricerca su crittografia e sicurezza informatica: lo schema di decreto legislativo vieta l'aggiramento di “misure tecnologiche” e la diffusione di informazioni sull'argomento, senza alcuna tutela per la ricerca scientifica;
- vieta di cedere o rivendere il materiale digitale acquistato via Internet, con i rischi già illustrati per la futura possibilità di accesso al sapere.

Grazie ad una delega ottenuta dal Governo, tale decreto è giunto in tempi brevi ad una forma definitiva ed esecutiva, senza dibattito parlamentare. Purtroppo, il clamore attorno ai (discutibilissimi) aumenti di prezzo ha posto in secondo piano altri aspetti decisamente più importanti e preoccupanti dello schema di decreto legislativo: esso, infatti, rappresenta il recepimento dell'EUCD nella legislatura italiana, ed introduce tutte le pericolose innovazioni previste dalla direttiva.

Per ulteriori dettagli, incluse le modifiche al decreto proposte da Assoli, oltre che per seguire i futuri sviluppi della questione: <http://www.softwarelibero.it/progetti/eucd/index.shtml>

Brevetti sul software

“Proteggiamo l’innovazione in Europa: no ai brevetti software” è il titolo dell’appello ai parlamentari europei diffuso in appoggio all’iniziativa intrapresa da FFII (Free Information Infrastructure, <http://ffii.org/>), associazione non-profit di Monaco di Baviera che si dedica alla maggior diffusione possibile dell’informazione sull’elaborazione dei dati. Un’azione, quella partita dalla Germania, sostenuta in Italia da Assoli, anche se nel caso specifico si appoggia al sito <http://swpat.xsec.it/> e all’ufficio dell’europarlamentare Marco Cappato. Già dall’intestazione dell’appello si evince il bersaglio: la brevettazione delle opere immateriali e, più nello specifico, del software.

Un pericolo che in Europa è alla sua terza apparizione e che va a insidiare gli articoli 52.2 e 52.3 della Convenzione di Monaco (Convenzione sulla concessione di brevetti europei, Cbe), approvata il 5 ottobre 1973. Il primo tentativo coincide con i lavori della Conferenza di Parigi (14-25 giugno 1999). Il secondo si ripresenta meno di un anno e mezzo più tardi con la Conferenza di Monaco (20-29 novembre 2000). Entrambi non hanno sortito l’effetto desiderato.

In risposta al terzo tentativo, lanciato a settembre, Assoli ha raccolto il testimone europeo diffondendo la “Richiesta di azione” proposta negli altri paesi e sollecitando piccole e medie imprese e professionisti attivi nel campo del software libero a firmare l’appello ai parlamentari europei contro i brevetti.

La proposta della Commissione Europea sulla brevettabilità delle innovazioni nel software richiede che il Parlamento Europeo, i governi degli stati membri ed altre figure politiche diano una chiara risposta. Le maggiori preoccupazioni riguardano diversi fatti: – l’Ufficio Europeo per i Brevetti (UEB), in contraddizione con

il testo e lo spirito della legge, abbia garantito decine di migliaia di brevetti sulle idee riguardanti la programmazione e gli affari, che chiameremo “brevetti sul software”;

– la Commissione Europea (CEC) sta esercitando pressioni affinché questi brevetti siano legalizzati e resi applicabili in tutta Europa. Nel fare ciò, la CEC sta ignorando il chiaro e ben argomentato appello della grande maggioranza di professionisti del software, compagnie, scienziati ed economi;

– la CEC basa la sua proposta su una bozza di documento scritta apparentemente dalla Business Software Alliance (BSA), una organizzazione statunitense guidata da poche grandi compagnie, come Microsoft, che ha un considerevole interesse su tale argomento, dato che attualmente il 60% dei brevetti per il software accordati dall’UEB sono detenuti da compagnie statunitensi;

– i brevetti sul software interferiscono con il diritto d’autore su questo e per i creatori di software tendono a portare all’espropriazione piuttosto che alla protezione della loro proprietà. Dei numerosi studi economici esistenti, nessuno conclude che i brevetti sul software portino ad una maggiore produttività, innovazione, diffusione del sapere o siano, in qualche altro modo, macro-economicamente vantaggiosi. La brevettabilità del software proposta da CEC/BSA, inoltre, porta a diverse inconsistenze all’interno del sistema dei brevetti e annulla le centrali assunzioni su cui si basa. Come risultato, ogni cosa diventa brevettabile e non ci può più essere alcuna sicurezza legale;

– le istituzioni del sistema europeo dei brevetti non sono in alcun modo soggette significativamente ad un controllo democratico. La divisione tra potere legislativo e giudiziario non è sufficiente ed in particolare l’UEB sembra essere terreno fertile per gli abusi e la pratica dell’illegalità.

Per queste ragioni, gli appelli delle varie Associazioni avanzano le seguenti richieste:

- sollecitiamo il Parlamento ed il Consiglio Europeo a rifiutare la proposta direttiva COM(2002)92 2002/0047;
- sollecitiamo il Parlamento Europeo a trovare un modo per obbligare l'UEB a rifondarsi, per come è intesa la brevettabilità, sulle sue linee guida d'indagine del 1978 o un equivalente, in modo da reinstaurare la corretta interpretazione della CBE;
- suggeriamo che un tribunale europeo indipendente sia obbligato a riesaminare su richiesta di un qualunque cittadino un qualsiasi brevetto che a prima vista possa sembrare accordato sulla base di una scorretta interpretazione delle direttive sulla brevettabilità dell'EPC, e che l'UEB, in tali casi, sia obbligata a rimborsare ai precedenti detentori del brevetto tutte le tasse da loro pagate;
- sollecitiamo i legislatori, sia a livello europeo che nazionale, affinché approvino il corrente testo dell'EPC e considerino la sua riapplicazione in accordo alla proposta (<http://swpat.ffii.org/stidi/epc52/index.de.html>), ciò fino a quando sarà ritenuto necessario, in modo da evitare interpretazioni scorrette da parte dei tribunali;
- proponiamo che il Parlamento ed il Consiglio Europeo considerino di rendere palesi i limiti della brevettabilità nel caso del software e delle creazioni dell'ingegno emanando una direttiva europea secondo le linee delle contro-proposte disponibili su <http://swpat.ffii.org/stidi/javni/index.de.html> e <http://swpat.ffii.org/papri/eubsaswpat0202/index.en.html#prop>
- chiediamo che ogni proposta di legge (incluse le proposte della direttiva CEC e le regole create dai precedenti giuridici) riguardante la brevettabilità sia verificata attraverso un sistema di prove costituito da esempi di applicazione del brevetto, in modo da vede-

re al di là di ogni dubbio se ciò porterà effettivamente i risultati desiderati e non lascerà spazio ad alcuna interpretazione sbagliata;

– proponiamo che il Parlamento Europeo crei un Comitato Permanente sulla Brevettabilità, con lo scopo di assicurare che i brevetti siano accordati solo nelle condizioni in cui questi vadano nella direzione del pubblico interesse. Questo comitato dovrebbe essere composto da persone del MEP ed indipendenti, esperti in vari campi dell'ingegno quali matematica, informatica, scienze naturali, ingegneria, economia, epistemologia, etica e giurisprudenza. Il numero dei detentori di brevetti, funzionari dell'ambiente o altre persone le cui entrate e carriere dipendano dalla comunità dei brevetti, deve essere mantenuto esiguo (ad esempio il 10-20%). Il comitato dovrà controllare ogni legge sui brevetti così come le interpretazioni che gli uffici brevetti e i tribunali ne faranno. Inoltre dovrà istituire incontri, proporre studi specifici sugli effetti del sistema dei brevetti e stimolare una ricerca correlata nel modo più aperto e inclusivo possibile. Il comitato dovrebbe segnalare al Parlamento Europeo in che misura la realtà dei brevetti è conforme alla teoria ed agli obiettivi di politica pubblica della Comunità Europea e dei relativi membri. Il lavoro di questo comitato dovrà rivolgersi verso le preoccupazioni sollevate dal Comitato del Parlamento Europeo per gli Affari Legali ed il Mercato Interno per il Controllo di Qualità nell'UEB, come espresso nella discussione sulla regolamentazione comunitaria sui brevetti COM(2000)0412;

– proponiamo che il Parlamento Europeo crei un Comitato d'inchiesta per investigare sulle varie accuse di comportamenti irregolari tenuti da coloro che propongono le direttive sulla brevettabilità del software e delle opere d'ingegno all'UEB ed al CEC, come la loro stretta collaborazione con una limitata cerchia di potenze, il loro ragionare incoerente ed il loro apparente disprezzo dei princi-

pi democratici e legali, e di proporre misure per una riforma in modo da prevenire il ritorno di questi fenomeni nel futuro;

– riteniamo che, almeno fino a quando i problemi nell'UEB non saranno risolti, ogni nuova regolamentazione, come Brevetto comunitario, sia implementata attraverso istituzioni differenti dall'UEB.

Per ulteriori dettagli, nonchè per seguire i futuri sviluppi della questione: <http://swpat.ffii.org/> e <http://swpat.xsec.it/>

Pubblica Amministrazione

Il software libero sta diventando un argomento sempre più spesso collegato alla pubblica amministrazione. La conferma più istituzionale a questa affermazione deriva dalle indicazioni contenute nell'“Indagine conoscitiva sul software a codice sorgente aperto nella Pubblica Amministrazione”:

(http://www.innovazione.gov.it/ita/comunicati/open_source/indagine_commissione_os.pdf). Un documento frutto del lavoro della Commissione ministeriale di indagine sull'open source, presieduta da Angelo Raffaele Meo, docente al Politecnico di Torino, che dà un rilievo ufficiale all'argomento.

L'argomento, tuttavia, non è nuovo. E se anche l'Aipa (Autorità per l'informatica nella pubblica amministrazione), nel maggio 2002, aveva rilasciato uno studio dal titolo “Il software Open Source (OSS), scenario e prospettive” curato da Francesco Grasso (<http://www.aipa.it/servizi%5B3/notizie%5B2/scenariooss.pdf>), la comunità del software libero si interroga da tempo sui rapporti e sui vantaggi per la cosa pubblica dalla sua adozione. Da questo presupposto, scaturisce anche il progetto di Assoli “Il Software Libero per la Pubblica Amministrazione” il cui contenuto è illustrato nella sua introduzione.

“L’Associazione Software Libero constata con soddisfazione che la pubblica amministrazione italiana, a vari livelli, sta favorendo l’adozione del software libero tramite appositi provvedimenti volti a rimuovere ostacoli alla sua diffusione, o lo promuove attivamente riconoscendone l’importanza strategica. Gli enti pubblici locali e nazionali che vorrebbero impegnarsi su questa strada, però, rischiano di essere ostacolati dalla carenza di informazione sul tema. Vista la delicatezza del dibattito che si è ormai esteso al di fuori della comunità, per coinvolgere media ed importanti organi politici, l’Associazione costituisce un gruppo di lavoro, che si propone di raccogliere sistematicamente le fonti di conoscenza sull’argomento, e di diffonderle comunicando direttamente con stampa e organi politici. Come primo passo per facilitare il lavoro è stata creata una mailing list di discussione e coordinamento all’indirizzo: pa@softwarelibero.it”.

Il documento prosegue indicando quelli che, nell’opinione di Assoli, sono i cardini – riportati integralmente – su cui confrontarsi e discutere. Prima di procedere nella loro presentazione, va ricordato che l’Associazione, per mantenere aggiornata l’evoluzione della questione, ha avviato anche due sezioni che si occupano di monitorare rispettivamente il dibattito e quanto viene pubblicato.

Un numero sempre maggiore di amministrazioni locali e nazionali privilegiano, con appositi atti di legge, il software libero nell’utilizzo all’interno delle proprie infrastrutture informatiche. Ricordiamo in breve i benefici principali attribuiti all’adozione del software libero da parte degli enti pubblici:

– La promozione dello sviluppo economico locale: il modello di sviluppo e vendita del software libero è, infatti, un modello collaborativo e diffuso, che consente a una pluralità di attori economici di beneficiarne (sviluppatori, imprese di servizi, utenti). I costi di acquisto del software, infatti, si spostano dalle licenze alla personalizza-

zione, consentendo lo sviluppo di imprese locali di servizi. Questo può rilanciare un'economia nazionale del software, mentre finora molti paesi erano costretti ad importare costosi pacchetti dall'estero.

– La trasparenza e la sicurezza: il fatto di poter disporre del codice sorgente dei programmi con cui vengono memorizzati e trattati i dati dei cittadini offre la possibilità di verificare la sicurezza delle applicazioni software. La disponibilità del sorgente a una vasta comunità di programmatori interessati a testarlo comporta notevoli vantaggi nella risoluzione di problemi, senza dover attendere le soluzioni ufficiali emesse dai produttori.

– La condivisibilità del software: il fatto di poter mettere a disposizione di altri enti pubblici il software sviluppato da un ente, nell'ottica di promozione del bene comune che è propria di questo settore; questa possibilità è garantita dalle licenze software "libere", in conformità alla legge n. 340 del 24 novembre 2000 (art. 25, comma I) che richiedeva proprio questo in ogni contratto di acquisto software stipulato da un ente pubblico, legge ampiamente disattesa anche perché poco nota.

– L'ereditarietà del software: il fatto di poter cambiare fornitore senza dover perdere l'investimento fatto; grazie agli standard aperti utilizzati nel software libero infatti tutti i professionisti IT, utilizzando linguaggi comuni, possono lavorare sulle applicazioni sviluppate in precedenza da altri. Diventa così più semplice mettere in concorrenza i propri fornitori (non è un caso che il modello di sviluppo proprietario o chiuso abbia generato un impressionante monopolista), quando si hanno le specifiche e la documentazione relative al codice software.

Per ulteriori dettagli e aggiornamenti:

<http://www.softwarelibero.it/pa/>

Associazione Software Libero, luglio 2003

Indice

Libertà di software e di pensiero, grazie! 3

PARTE PRIMA:

Libertà, società e software. 9

Possiamo fidarci del nostro computer?. 11

Perché il software dovrebbe essere libero 18

Diritto d'autore e globalizzazione nell'era delle reti informatiche 45

Software libero: libertà e cooperazione 87

Termini da evitare 150

PARTE SECONDA:

Le licenze 159

Licenza Pubblica Generica (GPL) del progetto GNU 160

Licenza Pubblica Generica Attenuata (LGPL) del Progetto GNU. 173

Licenza per Documentazione Libera (FDL) del Progetto GNU. 191

PARTE TERZA:

Risorse utili 203

ASSOLI: Progetti operativi in Italia e in Europa 204

Contro il comune senso del pudore, contro la morale codificata, controcorrente. Questa collana vuole abbattere i muri editoriali che ancora separano e nascondono coloro che non hanno voce. Siano i muri di un carcere o quelli, ancora più invalicabili e resistenti, della vergogna e del conformismo.

Saggi scelti di **Richard Stallman**
Software libero
Pensiero libero

A cura di Bernardo Parrella e Associazione Software Libero

Traduzioni di Bernardo Parrella e Gruppo traduttori italiani del progetto GNU

Titolo originale: *Free Software, Free Society: The Selected Essays of Richard M. Stallman*
Copyright © 2002 Free Software Foundation, Inc.

Free Software Foundation
59 Temple Place, Suite 330,
Boston, MA 02111-1307, USA
Email: gnu@gnu.org Web: <http://www.gnu.org>

Si consente la copia letterale e la distribuzione di uno o di tutti gli articoli di questo libro, nella loro integrità, a condizione che su ogni copia sia mantenuta la citazione del copyright e questa nota.

progetto grafico **Anyone!**

impaginazione **Littlered**

© 2004 **Nuovi Equilibri**

su concessione della Free Software Foundation
Casella postale 97 - 01100 Viterbo fax 0761.352751

Attenzione! *I manoscritti inviati all'editore non si restituiscono. Non vengono forniti pareri e schede di lettura. Non si considerano testi inviati per e-mail.*

finito di stampare nel mese di gennaio 2004

presso la tipografia **Graffiti**
via Catania 8 - 00040 Pavona (Roma)

ERETICA

■ **REBIBBIA RHAPSODY**

di Echaurren e Fioravanti

■ **SNATCH COMICS**

di JD Jachini

■ **UOMINI SU UOMINI**

di M.B. Bianchi

■ **STORIE DI SOGNI
E MALATTIE**

di I. Majore

■ **SI VIVE SOLO DUE VOLTE**

di C. Castaneda

■ **TAXI BROUSSE**

di M. Aime

■ **CUORE DI PULP**

di Giovannini & Tentori

■ **PER RAGAZZE DI COLORE...**

di N. Shange

■ **IL MANIFESTO DI
UNABOMBER**

■ **CHI HA VERAMENTE
COSTRUITO LE PIRAMIDI
E LA SFINGE**

di Giacobbo & Luna

■ **ERESIE PSICHEDELICHE**

di AA.VV.

■ **TUTTO VERO! MEMBRI DI
PARTITO**

di A. Selvaggi

■ **PICCOLI ERGASTOLI**

di Echaurren e Fioravanti

■ **RUBA QUESTO LIBRO**

di A. Hoffman

■ **LUCI ROSSE**

di D. Soffiati

■ **NON PROVATE A DEFINIRCI**

di AA.VV.

■ **L'OBEDIENZA NON È PIÙ
UNA VIRTÙ**

di L. Milani

■ **L'ARTE DELLA GIOIA**

di G. Sapienza

■ **PAPALAGI**

di Tuivavii di Tiavea

■ **CASTANEDA E LE
STREGHE DEL NAGUAL**

■ **SCIAMANI DELLE DUE
AMERICHE**

■ **APOCALISSE GIOIOSA**

di T. McKenna

■ **MONDO HACKER 1.0**

di A. Forni

■ **HOTEL CALIFORNIA**

di A. Azzaroni

■ **KATANGA CHE SORPRESA!**

■ **LINGUE**

di AA.VV.

■ **BANCA BASSOTTI**

di G. Cloza

■ **COSÌ PARLÒ BALAUSTRÀ**

di Vercillo & Zecchino

■ **SOMMI PECCATORI**

di A. Cavoli

■ **CREDERE OBBEDIRE
COMBATTERE**

di C. Galeotti

■ **CANNABIS, NON SOLO
FUMO**

di B. Parrella

■ **LA NOTTE DI STALIN**

di P. Pieri

■ **NEOPAGANESIMO**

di AA.VV.

■ **CORSARI VERDI**

di S. Apuzzo

■ **UN LETTO DI RISO**

di AA.VV.

■ **COME UCCISI MIA MADRE**

di L. Puliti

■ **EXTRATERRESTRI**

di T.C. Lethbridge

■ **I SIGNORI
DELLA TRANSIZIONE**
di A. Segre

■ **SESSO ANNUNCIATO**
di B.J. Loz

■ **CONTROARREDATURA**
di S. Ricciardi

■ **GIORDANO BRUNO
IL PROCESSO
E LA CONDANNA**
di A. Castronovo

■ **PIOGGIAFANGOMERDA
SOLEBLUES**
di M. Rossi

■ **DEPUTATI A FAR RIDERE**
di C.A. Colombo

■ **SOGNI AMERICANI**
di Sapphire

■ **IL LIBRO È NUDO**
di F. Del Moro

■ **QUESTA È L'AFRICA**
di G.A. Rolla

■ **BAMBINI ASSASSINI**
a cura di Giovannini e Tentori

■ **PINO ZAC**
di V. Vecellio

■ **QUATTRO SBERLE
IN PADELLA**
di S. Carnazzi e S. Apuzzo

■ **LA VENDETTA
DEL RISPARMIATORE**
di G. Cloza

■ **DONNE COL PISELLO**
di K. Valli Bentivoglio

■ **L'ANTICRISTO**
di F. Nietzsche

■ **WACO – UNA STRAGE
DI STATO AMERICANA**
di C. Stagnaro

■ **PERCHÉ GLI INGLESI
NON USANO IL BIDE?**
di P. Guagliumi

■ **ERESIA PURA**
di A. Petta

■ **IN AMORE VINCE IL CANE**
di S. Cecchi

■ **SIGNORA EROINA**
di A. Bongusto

■ **IO, ULTRAS**
di A. Arena

■ **CORPI ESTRANEI**
di P. Echaurren

■ **LA NOTTE DEGLI
STRAMURTI VIVENTI**
di E. Verrengia

■ **EDITORI A PERDERE**
di M. Bendia e A. Barocci

■ **ASSASSINATI**
di S. Carnazzi

■ **BLOC BOOK**
a cura di F. Giovannini

■ **ROMA DIVINA**
di P. Ravasenga

■ **CAMERATA TOPOLINO**
di A. Barbera

■ **IL PAROLIFERO**
di I. Capizzi

■ **MANUALE PRATICO
DELLA DONNA PADANA**
di N. Bresciani

■ **FIDO NON SI FIDA**
di S. Apuzzo e E. Meyer

■ **PALESTINESI**
di J. Genet

■ **ROGHI FATUI**
di A. Petta

■ **VADO, L'AFFONDO E
TORNO**
di M. La Ferla

■ **SESSO DENARO POTERE**
di Osho

■ **OMOCIDI**
di A. Pini

■ **IL SENSO DELLA VITA È
NON ROMPERE I COGLIONI**

di G. Nardella

■ **PELLE DI TERRA**

di V. Bottaro

■ **MANUALE PER
DIFENDERSI DAI
GIORNALISTI**

di C. Draghi

■ **FUMA PURE
SCIENZA SENZA SENSO**

di S. Milloy

■ **CEFALONIA
DOPPIA STRAGE**

di L. Caroppo

■ **MACHI
DI CARTA**

di A. Torreguitart Ruiz

■ **ERBA MEDICA**

Associazione Canapa Terapeutica

■ **ECCHIME**

di V. Cavallo

■ **VITE MINIME**

di D. Boccardi

■ **IO SONO GESÙ CRISTO**

di A. Artaud

■ **DA FIUME A ROMA**

di G. Ferrero

■ **PSICOFUNGHI ITALIANI**

di G. Camilla

■ **L'UOMO DI ATLANTIDE**

di M. La Ferla

■ **SOFTWARE LIBERO
PENSIERO LIBERO**

di R. Stullman

■ **L'ARTE DELLA GIOIA**

di G. Sapienza

■ **PARTO DI TESTA**

di A. Barocci

■ **LE PAROLE DELLA TERRA**

di L. Veronelli - P. Euchaurren

■ **I FIGLI DI BABELLE**

di V. Ruotolo

■ **L'ULTIMO COLPO DI**

HORST FANTAZZINI

di P. Diamante

■ **SELVATICO E COLTIVATO**

Rete Bioregionale Italiana

■ **LA MARIJUANA FA BENE
FINI FA MALE**

di G. Blumir

■ **PORNO ITALIA**

di F. Giovannini

■ **ADDIO, MAREMMA BELLA**

di A. Cavoli

■ **RACCONTI CONTRO TUTTI**

di M. Twain

■ **SOFTWARE LIBERO
PENSIERO LIBERO - Vol. 2**

di R. Stullman

Redazione:

Stampa Alternativa

C.P. 741

00100 Roma Centro

e mail: nuovi.equilibri@agora.it

http: [//www.stampalternativa.it](http://www.stampalternativa.it)

