

## Per volontà di una stampante

Temo i Greci, anche se portano doni.

– Virgilio, *L'Eneide*

La nuova stampante si era bloccata un'altra volta.

Richard M. Stallman, programmatore dello staff del laboratorio di intelligenza artificiale (AI Lab) presso il Massachusetts Institute of Technology, aveva fatto le spese di quel malfunzionamento. Un'ora dopo aver inviato un file di 50 pagine alla stampante laser, Stallman, 27 anni, decise d'interrompere una produttiva sessione di lavoro per andare a recuperare il documento. Arrivato in loco, trovò appena quattro pagine stampate. Al colmo della frustrazione, si accorse che quelle pagine erano di un altro utente, confermando così che il suo documento per intero e parte delle stampe di qualcun altro erano ancora intrappolati nel circuito elettrico della rete interna.

Per un programmatore il fatto di dover aspettare i comodi di una macchina è un rischio del mestiere, perciò Stallman decise di affrontare la cosa con un po' di buonsenso.

Tuttavia, la differenza tra l'attesa del funzionamento *di* una macchina e quella *su* una macchina è alquanto considerevole. Non era la prima volta che era costretto a chinarsi sopra la stampante per seguire l'uscita delle pagine una alla volta. Abituato a trascorrere gran parte del giorno e della notte a migliorare l'efficienza degli apparecchi e dei relativi programmi di controllo, Stallman sentì il bisogno naturale di aprire quella macchina, dare un'attenta occhiata all'interno e isolare le radici del problema.

Purtroppo le sue capacità di programmatore informatico non si estendevano all'ambito dell'ingegneria meccanica. Mentre la stampante continuava a produrre documenti freschi d'inchiostro, Stallman ebbe l'opportunità di riflettere su altre modalità per eludere il problema dell'intasamento dei fogli. Quanto tempo era passato da quando i membri del laboratorio avevano accolto a braccia aperte la nuova stampante? Stallman se lo andava chiedendo. La macchina era stata donata dalla Xerox Corporation. Si trattava di un prototipo d'avanguardia, versione modificata della diffusa fotocopiatrice Xerox. Soltanto che, anziché effettuare copie, seguiva le istruzioni di un apposito software diffuse tramite la rete locale interna per trasformare le informazioni in documenti di qualità professionale. Creata dagli ingegneri dello Xerox Palo Alto Research Center, un centro di fama mondiale, rappresentava, per dirla semplicemente, un assaggio della rivoluzione della stampa desktop che avrebbe poi raggiunto il resto dell'industria informatica entro la fine del decennio.

Guidati da una passione innata a giocare con le migliori apparecchiature esistenti, i programmatori del laboratorio di intelligenza artificiale si erano dati immediatamente da fare per integrare la nuova macchina all'interno della sofisticata infrastruttura informatica. I risultati apparvero subito positivi. Al contrario della vecchia stampante laser, la nuova Xerox era assai veloce. Le pagine volavano fuori al ritmo di una al secondo, trasformando un lavoro di 20 minuti in uno da due minuti. Il lavoro risultava anche più preciso. I cerchi venivano stampati correttamente, non come ovali. Le linee rette rimanevano tali e non diventavano sinusoidali. Sotto tutti gli aspetti e le intenzioni, si trattava di un regalo troppo bello da rifiutare.

Soltanto qualche settimana dopo il suo arrivo iniziarono a evidenziarsi i primi difetti. Tra i problemi maggiori vi era l'insita predisposizione all'inceppamento della carta. Le menti ingegneristiche di quei programmatori compresero al volo la questione: in quanto fotocopiatrice, generalmente la macchina funzionava con la supervisione diretta di un operatore. Ci sarebbe sempre stato qualcuno, insomma, pronto a risolvere problemi di carta inceppata – così dovevano aver ragionato gli ingegneri della Xerox, dedicandosi perciò alla risoluzione di altre noie. In termini tecnici, il sistema di funzionamento faceva affidamento sulla diligenza dell'utente.

Nell'apportare le modifiche necessarie per l'utilizzo come stampante, gli ingegneri della Xerox avevano modificato anche il rapporto utente-macchina in maniera sottile ma profonda. Invece di prevedere l'intervento di un singolo operatore, la macchina sottostava a un'intera popolazione di operatori collegati in rete. Anziché starvi letteralmente sopra, un utente di tale rete inviava il comando di stampa attraverso una lunga catena di macchine, contando sul fatto che il contenuto desiderato giungesse a destinazione in maniera corretta e appropriata. Solamente alla fine, quando si recava a ritirare l'output finale, poteva rendersi conto di quanto poco del contenuto fosse stato effettivamente stampato.

Stallman era stato tra i primi a identificare il problema e il primo a suggerirne la soluzione. Qualche anno addietro, quando il laboratorio utilizzava ancora la vecchia stampante, Stallman aveva risolto un problema analogo "aprendo" il software di controllo della stampante collegata alla macchina PDP-11 del laboratorio. Pur non riuscendo a impedire l'inceppamento della carta, egli inserì nel programma un comando che ordinava al PDP-11 di verificare periodicamente i vari meccanismi e inviarne relazione al PDP-10, il computer centrale del laboratorio. Per esser certo che la negligenza di qualcuno non danneggiasse tutta una serie di stampe in corso, Stallman aggiunse inoltre un comando che istruiva il PDP-10 a notificare l'eventuale blocco della stampante a ogni utente con una stampa in attesa. Si trattava di una semplice nota, qualcosa del tipo "La stampante è bloccata, si prega di sistemarla"; e poiché tale nota raggiungeva proprio coloro che avevano maggior necessità di risolvere il problema, esistevano buone probabilità che ciò avvenisse in tempi brevi.

Come per ogni soluzione, quella di Stallman era indiretta ma elegante. Non risolveva la parte meccanica del problema, però ci andava vicino chiudendo il cerchio informativo tra utente e macchina. Grazie all'aggiunta di poche righe di codice, lo staff del laboratorio di intelligenza artificiale poteva così eliminare una perdita di tempo pari a 10-15 minuti ogni settimana per correre avanti e indietro a controllare la stampante. In termini di programmazione, la soluzione di Stallman traeva vantaggio dall'intelligenza amplificata che sosteneva la rete locale nel suo complesso.

"Se ricevevi quel messaggio, non potevi dare per scontato che qualcun altro avrebbe risolto il problema", spiega Stallman, sottolineando

la logica applicata in quell'occasione. "Dovevi controllare di persona. Un minuto o due dopo il mancato funzionamento, i due o tre utenti che avevano ricevuto l'avviso erano andati a risolvere il problema. Di questi, generalmente almeno uno sapeva come fare per risistemare i fogli."

Questo tipo di soluzioni indirette rappresentavano una sorta di marchio del laboratorio di intelligenza artificiale e dei programmatori originari che lo popolavano. Anzi, i migliori tra costoro disdegnavano lo stesso termine programmatore, preferendo invece l'appellativo gergale di hacker. Una qualifica professionale che indicava uno spettro di attività alquanto ampio – qualsiasi cosa compresa tra il caos creativo e il miglioramento di software e sistemi esistenti. Nel termine era tuttavia implicita la vecchia idea di una certa ingenuità *yankee*. Per essere un hacker, bisognava accettare la filosofia secondo cui la scrittura di un programma era soltanto l'inizio. Stava poi nella capacità di migliorarlo la vera prova di abilità per un hacker.<sup>1</sup>

Questa filosofia costituiva il motivo principale che spingeva aziende come la Xerox a seguire la politica di donare macchine e programmi a quelle entità che costituivano il terreno abituale degli hacker. Se questi ultimi riuscivano a migliorarne il software, tali migliorie sarebbero state riutilizzate dalle stesse aziende, incorporandole nelle nuove versioni dei programmi destinate al mercato commerciale. Nella terminologia imprenditoriale, gli hacker rappresentavano una sorta di capitale comunitario, un'unità aggiuntiva per la ricerca e lo sviluppo disponibile a costi minimi.

Fu grazie a questa filosofia del dare e del ricevere che quando Stallman isolò il difetto di carta inceppata nella stampante laser Xerox, non si fece certo prendere dal panico. Semplicemente, si mise alla ricerca di un modo per aggiornare le vecchie procedure e adattare al nuovo sistema. Tuttavia, mentre si apprestava a dare un'occhiata al software della stampante Stallman scoprì qualcosa di preoccupante. La macchina non sembrava essere dotata di alcun software, almeno nulla che lui stesso o gli altri programmatori fossero in grado di leggere. Fino ad allora, la maggioranza delle aziende riteneva una forma di cortesia la pubblicazione in formato testuale dei codici sorgenti che documentavano i singoli comandi su cui era basato il comportamento della macchina. In questo caso, la Xerox aveva fornito i file in formato precompilato,

---

1 Per maggiori dettagli sul termine "hacker", si veda l'Appendice B.

ovvero binario. I programmatori erano liberi di aprirli, ma a meno che non fossero degli esperti nel decifrare sequenze infinite di zero e di uno, il testo finale sarebbe rimasto una sfilza di caratteri incomprensibili.

Pur sapendone parecchio d'informatica, Stallman non poteva considerarsi certo un esperto nella decifrazione dei file binari. L'esperienza acquisita come hacker gli consentiva, però, di fare affidamento su altre risorse. Il concetto della condivisione dell'informazione occupava un ruolo talmente centrale nella cultura hacker che sarebbe stata solo questione di tempo prima che un collega di qualche laboratorio universitario o di altra struttura aziendale riuscisse a fornirgli la versione testuale dei sorgenti relativi a quel software.

Dopo i primi blocchi della stampante, Stallman si consolò ripensando a una situazione analoga sperimentata qualche anno addietro. Il laboratorio aveva avuto bisogno di un programma sulla rete per aiutare il PDP-11 ad operare in maniera più efficiente con il PDP-10. Gli hacker del laboratorio potevano facilmente cavarsela da soli, ma Stallman, avendo studiato ad Harvard, si ricordò di un programma analogo scritto dai programmatori del dipartimento d'informatica di quell'università. Tale dipartimento usava un computer del medesimo modello, il PDP-10, pur se dotato di un diverso sistema operativo. Le procedure in vigore nel laboratorio richiedevano altresì che tutti i programmi installati sul PDP-10 fossero accompagnati dalla pubblicazione dei relativi codici sorgenti.

Potendo contare sul pieno accesso al laboratorio d'informatica di Harvard, Stallman vi si recò, fece una copia di quei codici e li portò con sé al laboratorio di intelligenza artificiale. Qui prese a riscriverne alcune parti onde renderle compatibili con il sistema operativo locale. Senza troppo clamore, il laboratorio di intelligenza artificiale richiese una falla importante all'interno della propria infrastruttura informatica. Stallman aggiunse persino alcune funzioni assenti nel programma originale di Harvard, rendendolo così ancora più utile. "Ne facemmo uso per parecchi anni senza problemi", sostiene Stallman.

Nella concezione di un programmatore degli anni '70, questa operazione era l'equivalente software di un vicino di casa che veniva a chiederci in prestito un utensile o una tazza di zucchero. Nel caso del prestito di una copia del software per il laboratorio di intelligenza artificiale, l'unica differenza stava nel fatto che Stallman non aveva privato gli hacker di Harvard dell'utilizzo del programma originale. Anzi, erano stati

proprio costoro a trarne vantaggio, considerate le nuove funzionalità aggiuntevi dallo stesso Stallman, funzionalità che rimanevano a loro completa disposizione. Anche se nessuno ad Harvard richiese mai quel programma, Stallman ricorda che lo fece invece uno sviluppatore della società privata Bolt, Beranek & Newman, il quale vi apportò ulteriori aggiunte poi reintegrate da Stallman nell'archivio dei codici sorgenti organizzato presso lo stesso laboratorio di intelligenza artificiale.

“Normalmente ogni programma seguiva uno sviluppo analogo a quello della pianificazione cittadina”, dice Stallman in riferimento all'infrastruttura informatica in vigore in quel laboratorio. “Alcune aree venivano interamente sostituite e ricostruite. Vi si aggiungevano nuovi elementi. Ma era sempre possibile isolarne qualche sezione e dire, ‘A giudicare dallo stile, queste stringhe sono state scritte all'inizio degli anni '60, mentre queste altre risalgono a metà anni '70.’”

Grazie a questo semplice sistema di crescita intellettuale, gli hacker operanti nel laboratorio di intelligenza artificiale e in altri ambiti riuscirono a mettere in piedi prodotti particolarmente affidabili. Fu ricorrendo a tale strategia che, sulla costa occidentale degli Stati Uniti, i ricercatori della University of California di Berkeley, in collaborazione con alcuni ingegneri di primo livello presso la AT&T, realizzarono un intero sistema operativo. Chiamato Unix, con un gioco di parole riferito ad un sistema operativo più vecchio e più rispettabile in ambito accademico noto come Multics, il software venne messo a disposizione di qualunque programmatore disposto ad accollarsi le spese per copiarlo su un nuovo nastro magnetico e per la successiva spedizione.

Non tutti gli sviluppatori coinvolti in questa cultura amavano autodefinirsi hacker, ma la vasta maggioranza condivideva i sentimenti di Richard M. Stallman. Se un programma o una correzione software si dimostravano sufficientemente validi da risolvere i problemi dell'autore, potevano risolvere anche quelli di qualcun altro. Perché allora non condividerli anche soltanto per guadagnarsi un buon karma?

Inizialmente il fatto che la Xerox avesse deciso di non condividere i propri sorgenti non provocò alcun fastidio. Mentre andava alla ricerca di quei file, Stallman sostiene di non essersi minimamente preoccupato di contattare la Xerox: “Ci avevano già regalato la stampante, perché mai importunarli ancora?”

Quando però si accorse che i file non comparivano da nessuna parte, Stallman iniziò a farsi sospettoso. L'anno precedente aveva già

subito un brutto colpo da parte di un laureando presso la Carnegie Mellon University. Lo studente, Brian Reid, aveva realizzato un utile programma di formattazione testi chiamato Scribe. Si trattava di uno dei primi software grazie al quale l'utente poteva scegliere font e stili di un documento da inviare attraverso una rete, un antesignano del linguaggio HTML, la lingua franca del World Wide Web. Nel 1979, Reid decise di vendere Scribe ad un'azienda produttrice di software nell'area di Pittsburgh, la Unilogic. Ormai vicino alla laurea, Reid stava semplicemente cercando una soluzione per mollare il programma a qualche sviluppatore disposto ad accollarsi l'onere di impedirne la caduta nel pubblico dominio. Per addolcire la trattativa, Reid si dichiarò inoltre d'accordo all'inserimento di una serie di funzioni con limiti temporali – “bombe a tempo”, nel gergo dei programmatori – grazie alle quali le versioni gratuite del programma smettevano di funzionare dopo 90 giorni. Onde evitare la disattivazione, gli utenti dovevano così pagare una tariffa al produttore, il quale a sua volta avrebbe fornito loro il codice necessario a disattivare il blocco.

Secondo Reid, era un buon affare per entrambe le parti. Scribe non si sarebbe eclissato tra i programmi di pubblico dominio, mentre la Unilogic avrebbe recuperato il denaro investito. Per Stallman si trattava invece del puro e semplice tradimento dell'etica del programmatore. Anziché onorare il concetto della condivisione, Reid collaborava con l'industria costringendo gli sviluppatori a pagare per avere accesso all'informazione.

Con il trascorrere delle settimane, mentre i tentativi di rintracciare il codice sorgente della stampante apparivano senza via d'uscita, Stallman sentì incombere un analogo scenario di scambio denaro-codice. Tuttavia, prima di poter dire o fare qualunque cosa al riguardo, ecco la rete dei programmatori produrre una buona notizia. Girava voce che un ricercatore presso il dipartimento d'informatica della Carnegie Mellon University si era appena dimesso dallo Xerox Palo Alto Research Center. E non soltanto aveva lavorato sulla stampante in questione, ma, stando alle stesse voci, se ne stava tuttora occupando come parte dei progetti assegnatigli alla Carnegie Mellon.

Mettendo da parte ogni sospetto iniziale, Stallman decise di andarlo a trovare durante la sua visita successiva al campus di quell'università. Non dovette attendere a lungo. Anche la Carnegie Mellon aveva un laboratorio specializzato in ricerche sull'intelligenza artificiale, e nel

giro di qualche mese egli dovette recarvisi per questioni di lavoro. Durante la visita, fece in modo di fermarsi al dipartimento d'informatica. Qualcuno dei ricercatori gli indicò l'ufficio del docente incaricato del progetto Xerox. Entrando in quell'ufficio, lo trovò al lavoro.

Nel tipico confronto tra ingegneri, la conversazione che ne seguì fu cordiale ma schietta. Dopo essersi rapidamente presentato come un collega del MIT, Stallman gli chiese una copia dei sorgenti della stampante laser in modo da poterli adattare al PDP-11. Con sua sorpresa, il professore gli oppose un netto rifiuto.

“Mi disse di aver promesso che non me ne avrebbe fornito una copia”, sostiene Stallman.

A volte la memoria gioca brutti scherzi. Vent'anni dopo quel fatto, il ricordo mentale di Stallman è notoriamente pieno di lacune. Non soltanto non rammenta la motivazione alla base del viaggio di lavoro né l'anno in cui questo avvenne, ma neppure riesce a descrivere in qualche modo la controparte con cui tenne quello scambio di battute. A sentire Reid, probabilmente la persona che avrebbe potuto soddisfare la richiesta di Stallman era Robert Sproull, ex ricercatore presso lo Xerox PARC e attuale direttore dei Sun Laboratories, divisione del conglomerato informatico Sun Microsystems. Negli anni '70, mentre lavorava allo Xerox PARC, Sproull aveva diretto il gruppo di sviluppo del software per la stampante laser in questione. Verso il 1980, assunse il ruolo di ricercatore informatico presso la Carnegie Mellon, dove tra i vari progetti continuò ad occuparsi ancora di quella stampante.

“Il codice oggetto della richiesta di Stallman rappresentava la parte migliore del software che Sproull aveva scritto nell'anno precedente al suo arrivo alla Carnegie Mellon”, ricorda Reid. “Credo che Sproull si trovasse alla Carnegie Mellon da meno di un mese quando arrivò quella richiesta.”

Interrogato direttamente sulla questione, tuttavia, Sproull cade dalle nuvole. “Non posso offrire alcun commento”, mi scrive via e-mail. “Non ho assolutamente il minimo ricordo dell'incidente”.

Vista l'amnesia di entrambe le persone coinvolte in quella breve conversazione, inclusa perfino la veridicità sull'effettivo svolgimento, è difficile valutare la rudezza del rifiuto opposto da Sproull, almeno nella ricostruzione di Stallman. Nei suoi interventi pubblici, quest'ultimo ha più volte fatto esplicito riferimento all'incidente, sottolineando come il rifiuto di Sproull fosse dovuto al cosiddetto accordo

di non divulgazione (non-disclosure agreement), norma contrattuale stipulata tra la Xerox Corporation e il proprio dipendente, sulla base della quale a quest'ultimo veniva garantito accesso ai codici sorgenti del software in cambio della promessa di mantenere la loro segretezza. Oggi clausola standard nei contratti dell'industria informatica, a quel tempo l'accordo di non divulgazione costituiva una novità, un segnale del valore commerciale acquisito per la Xerox sia dalla stampante sia dalle informazioni necessarie al relativo funzionamento. "In quel periodo la Xerox stava cercando di lanciare la stampante laser come prodotto commerciale", spiega Reid. "Per loro sarebbe stato folle regalarne in giro i codici sorgenti".

Invece Stallman considerava la faccenda in maniera diametralmente opposta. Si trattava del rifiuto da parte della Xerox e di Sproull, o chiunque fosse la persona che quel giorno si trovò di fronte, di far parte di un sistema che fino ad allora aveva spinto gli sviluppatori a considerare i programmi al pari di risorse comuni. Come un contadino i cui canali d'irrigazione vecchi di secoli si fossero improvvisamente prosciugati, Stallman aveva seguito quei canali fino alla sorgente soltanto per scoprire una nuova diga idroelettrica in cui spiccava il logo della Xerox.

Non fu semplice per Stallman mandar giù il fatto che la Xerox avesse coinvolto alcuni programmatori in questo nuovo e stravagante sistema di segretezza forzata. All'inizio riuscì a comprendere soltanto la natura personale di quel rifiuto. Dato che in genere gli incontri faccia a faccia lo mettevano a disagio e in imbarazzo, la visita non annunciata di Stallman ad un collega poteva essere intesa come una dimostrazione di socialità tra buoni vicini. Ma ora il rifiuto bruciava come un grave atto di scortesia. "Ero talmente arrabbiato che non riuscii neppure ad esprimerlo. Così mi sono girato e sono uscito, senz'aggiungere neanche una parola", rammenta Stallman. "Può anche darsi che abbia sbattuto la porta, chissà? Tutto quel che ricordo è che volevo andarmene subito via".

Vent'anni dopo quel fatto, la rabbia brucia ancora tanto da spingere Stallman a ritenere l'evento un punto di svolta sostanziale. Nei pochi mesi successivi, Stallman e gli hacker del laboratorio di intelligenza artificiale sarebbero stati colpiti da un'ulteriore serie di eventi talmente negativi da far impallidire la tensione di quei 30 secondi in un remoto ufficio della Carnegie Mellon. E tuttavia, quando si tratta di individuare quei singoli episodi che avrebbero trasformato Stallman da hacker

solitario, istintivamente sospettoso verso ogni autorità centralizzata, ad attivista animatore di una crociata che applica le nozioni tradizionali di libertà, uguaglianza e fraternità al mondo dello sviluppo software, lo stesso Stallman non esita a isolare l'incontro alla Carnegie Mellon, dedicandogli un'attenzione tutta particolare.

“L'evento mi spinse a riflettere più a fondo su quel che andavo già pensando”, spiega Stallman. “Avevo già l'opinione che il software avrebbe dovuto essere condiviso, ma non sapevo bene come procedere. Non avevo le idee chiare e organizzate al punto da poterle esprimere in maniera concisa al resto del mondo”.

Sebbene alcuni episodi precedenti avessero già suscitato le ire di Stallman, egli sostiene come fino all'incontro della Carnegie Mellon non si fosse reso conto del fatto che quella serie di eventi andasse infiltrandosi in una cultura considerata sacrosanta da molto tempo. Facendo parte di un'élite di programmatori in una delle maggiori istituzioni al mondo, Stallman aveva coscientemente deciso di ignorare ogni patto o compromesso sottoscritto dai suoi colleghi fin tanto che ciò non interferiva con il proprio lavoro. Fino all'arrivo della stampante laser della Xerox, egli si era accontentato di dedicarsi a macchine e programmi a malapena tollerati da altri utenti. Nelle rare occasioni in cui tali programmi uscirono dalle stanze del laboratorio di intelligenza artificiale – quando, per esempio, il laboratorio sostituì il venerabile sistema operativo Incompatible Time Sharing con una variante commerciale chiamata TOPS 20 – Stallman e gli altri hacker erano stati liberi di riscrivere, ricostruire e rinominare il software sulla base del gusto personale.

Ora che la stampante laser si era insinuata nella rete del laboratorio, tuttavia, qualcosa era cambiato. La macchina funzionava bene, al di là di occasionali problemi di carta inceppata, ma era scomparsa la possibilità di modificarla secondo il gusto personale. Dal punto di vista dell'industria del software nel suo complesso, la stampante rappresentava un campanello d'allarme. Il software era divenuto un bene talmente prezioso che le aziende non sentivano la necessità di diffonderne i sorgenti, soprattutto quando la loro pubblicazione significava offrire ai potenziali concorrenti la possibilità di duplicare qualche programma a costi irrisori.

Dal punto di vista di Stallman, la stampante era una sorta di cavallo di Troia. Dopo un decennio di tentativi falliti, il software sotto

proprietà privata – in seguito gli hacker introdurranno il termine “proprietario” – aveva messo piede nel laboratorio di intelligenza artificiale grazie al più subdolo dei metodi. Vi era entrato sotto le mentite spoglie di un regalo.

Il fatto che Xerox offrisse ad alcuni programmatori l’accesso a doni ulteriori in cambio del loro riserbo era un’amara scoperta, tuttavia Stallman non può fare a meno di notare come egli stesso, nel caso gli fosse stato proposto un simile scambio nei suoi anni giovanili, probabilmente avrebbe accettato l’offerta della Xerox Corporation. Però l’imbarazzante incontro alla Carnegie Mellon provocò un salutare effetto sulla sua apatia morale. Non soltanto l’evento gli fornì la rabbia necessaria per considerare ogni futuro regalo con una buona dose di sospetto, ma lo costrinse a porsi una domanda inquietante: cosa accadrebbe se un giorno un collega hacker entrasse nell’ufficio di Stallman e improvvisamente egli dovesse trovarsi a rifiutare un’analogha richiesta del codice sorgente?

“Era la prima volta che m’imbattevo in una clausola di non divulgazione, e mi resi immediatamente conto come questi accordi producano delle vittime”, sostiene con convinzione Stallman. “In questo caso la vittima ero io. Io e il mio laboratorio”.

Fu questa una lezione che avrebbe accompagnato Stallman lungo i tumultuosi anni ’80, un decennio nel corso del quale parecchi colleghi al MIT avrebbero abbandonato il laboratorio di intelligenza artificiale per firmare analoghi accordi di non-divulgazione a livello personale. Poiché gran parte di questi accordi prevedono un limite di durata, gli hacker che decisero di sottoscriverli non ci pensarono su due volte. Prima o poi, questo il loro ragionamento, il software diverrà di pubblico dominio. D’altra parte, la promessa di mantenere segreto il programma nel primo periodo di sviluppo faceva parte di quel compromesso che permetteva agli hacker di lavorare sui progetti migliori. Anche se per Stallman ciò non rappresentò altro che il primo passo verso un terreno molto sdruciolevole.

“Quando ricevevo un invito a tradire in tal modo tutti i miei colleghi, sentivo riaffiorare la rabbia provata quando qualcun altro aveva fatto lo stesso con me e l’intero laboratorio”, insiste Stallman. “Allora rispondevo, ‘Grazie mille per l’offerta di questo bel pacchetto software, ma non posso accettarlo alle condizioni che mi avete richiesto, per cui ne farò a meno’.”

Come avrebbe imparato presto, il rifiuto implicava molto di più di qualche sacrificio personale. Significava piuttosto l'isolamento dagli altri hacker che, pur condividendo un'analogha avversione per i segreti, tendevano a esprimerla in maniera più elastica a livello etico. Non trascorse molto tempo prima che Stallman, sempre più isolato all'interno del suo stesso laboratorio, prese ad autodefinirsi "l'ultimo vero hacker", allontanandosi sempre più da un mercato caduto sotto il dominio del software proprietario. Il rifiuto opposto a una qualsiasi richiesta per il codice sorgente, decise Stallman, rappresentava non soltanto il tradimento della missione scientifica che aveva alimentato lo sviluppo del software a partire dalla fine della Seconda Guerra Mondiale, ma anche una violazione della regola aurea, il fondamento morale che imponeva di non fare altri quello che non vorresti fosse fatto a te.

Da qui l'importanza della stampante laser e del successivo incontro alla Carnegie Mellon. In mancanza di questa circostanza, afferma Stallman, la sua vita avrebbe potuto seguire un percorso più ordinario, alla ricerca di un equilibrio tra la ricchezza di un programmatore commerciale e la frustrazione finale di un'esistenza spesa a scrivere codice invisibile. Non si sarebbe evidenziato alcun senso di chiarezza, nessuna urgenza di affrontare un problema di cui nessun altro pareva curarsi. Fatto ancor più importante, non avrebbe preso corpo quella giusta ira, un sentimento che, lo vedremo più avanti, rimarrà il motore propulsore della carriera di Stallman tanto quanto l'ideologia politica o le considerazioni etiche.

"Da quel giorno in poi decisi che non avrei mai potuto partecipare a quel sistema", dice Stallman riferendosi alla pratica di barattare la libertà personale per pura convenienza – così descrive gli accordi di non divulgazione – ma anche e soprattutto alla cultura globale che incoraggiava simili accordi moralmente dubbi. "Presi la decisione di non causare mai altre vittime, come era invece successo a me."