

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

W obronie wolności

Autor: Sam Williams

Tłumaczenie: Krzysztof Maślowski

ISBN: 83-7361-247-5

Tytuł oryginału: [Free as in Freedom](#)

Format: B5, stron: 290



Dlaczego rządzący Microsoftem budzą się w nocy na myśl o dokonaniach długowłosego hakera Richarda Stallmana? Dlaczego kilku najstojniejszych programistów świata nazywa go „świętym Ignacym”? W jaki sposób ten uparty, nad wiek rozwinięty chłopak wyrósł na Dawida zagrażającego Goliatowi przemysłu programistycznego ?

Dla Stallmana istnienie wolnego oprogramowania, wolnego swobodą wprowadzania zmian, ale nie zwolnionego od konieczności zapłaty – to sprawa priorytetowa. Swoje życie poświęcił uwalnianiu świata od oprogramowania o zastrzeżonym prawie własności i modyfikacji.

W roku 1983 wydał swój Manifest GNU, rozpoczynając realizację zuchwałego projektu stworzenie wolnego systemu operacyjnego mogącego zastąpić Uniksa. Zrezygnował z czoła przyjętemu systemowi intelektualnej własności, wstawiając do stworzonej przez siebie „Powszechnej Licencji Publicznej” (General Public Licence – GPL) notę „copyleft”. Był to ruch, który naczelny dyrektor techniczny Microsoftu Craig Mundie nazwał miną podłożoną pod niezależny sektor oprogramowania komercyjnego.

„W obronie wolności” jest śledzeniem losów ekscentrycznego geniusza. Przeprowadzając wiele wyczerpujących wywiadów z „Robin Hoodem nowych technologii”, jego rodziną, kolegami hakerami i osobistościami przemysłu nowych technologii, autor Sam Williams kreśli portret bojownika o wolność, któremu udało się zmienić świat.



Spis treści

Wstęp	7
1. Gdy potrzebna drukarka	11
2. 2001: Odyseja hakerska	25
3. Portret hakera w wieku młodzieńczym	41
4. Impeach God	57
5. Niewielkie bajorko wolności	83
6. Wspólnota Emacs	105
7. Trudny wybór moralny	123
8. Św. IGNUcy	149
9. Powszechna Licencja Publiczna GNU	167
10. GNU/Linux	193
11. Oprogramowanie Open Source	209
12. Krótka podróż przez hakerskie piekło	229
13. Kontynuacja walki	235
Epilog: Miażdżąca samotność	251
A. Terminologia	269
B. Hak, hakerzy i hakowanie	271
C. Licencja GNU Wolnej Dokumentacji (GNU Free Documentation License — GFDL)	279

1

Gdy potrzebna drukarka

Obawiam się Greków, nawet gdy przynoszą dary
— Wergiliusz
Eneida

Nowa drukarka znowu się zacięła.

Richard M. Stallman, programista w Laboratorium Sztucznej Inteligencji¹ w Massachusetts Institute of Technology² (zwanym w skrócie AI Lab), dotkliwie odczuł odkrycie tego uszkodzenia. W godzinę po posłaniu do druku 50-stronicowego pliku 27-letni Stallman oderwał się na chwilę od zajmującej go pracy i poszedł po dokumenty. Po dotarciu do drukarki znalazł tylko cztery wydrukowane strony, co gorsza — nie były to strony z jego dokumentu, co oznaczało, że jego plik w całości nadal tkwił w przepastnej płataninie sieci AI Lab.

Ryzyko oczekiwania na wykonanie zadania przez maszynę jest częścią codziennej pracy programisty, więc Stallman dość spokojnie przyjął rozczarowanie, które go spotkało. Jest jednak różnica między oczekiwaniem, aż maszyna zakończy działanie, a czekaniem na to, czy w ogóle raczy je rozpocząć. Wielokrotnie był zmuszony czekać na kartki dość wolno „wypluwane” przez drukarkę. Jako kogoś, kto spędził znaczną część życia na poprawianiu efektywności urządzeń technicznych i oprogramowania nimi sterującego, zawsze korciło go, by

otworzyć pokrywę, zajrzeć do wnętrza działającej maszyneryi i poszukać przyczyn problemu.

Niestety, jego sprawność i zdolności programisty nie sięgały zagadnień techniczno-inżynierskich. Gdy wreszcie wyjął z drukarki świeżo wydrukowane strony, zaczął się zastanawiać, w jaki sposób mógłby na przyszłość uniknąć tego problemu.

Niedawno zespół AI Lab z otwartymi ramionami powitał nową drukarkę. Urządzenie było darem Xerox Corporation. Prototyp najnowszego modelu był zmodyfikowaną wersją popularnej fotokopiarki. Zamiast dokumentami przeznaczonymi do kopiowania, urządzenie było „karmione” danymi produkowanymi przez oprogramowanie i dostarczanych przez kabel sieciowy. Dane te były następnie przekształcane w profesjonalnie wyglądające wydrukowane dokumenty. Maszyna, zaprojektowana przez znane w świecie biuro badawczo-projektowe Xerox Palo Alto Research Facility, była pierwszym zwiastunem urządzeń *desktop publishing*³, które do końca dekady miały zająć znaczące miejsce na rynku komputerowym.

Nagleni instynktowną potrzebą skorzystania z najnowszego urządzenia, programiści z AI Lab szybko zintegrowali je ze skomplikowaną infrastrukturą swej sieci komputerowej. Natychmiast osiągnięto zadawalające rezultaty. Nowe urządzenie Xeroxa było znacznie szybsze od poprzednio używanej starej drukarki laserowej. Zadrutowane kartki opuszczały nową drukarkę w tempie jedna na sekundę, co np. dawne 20-minutowe drukowanie skracało do 2 minut. Nowa drukarka działała również o wiele precyzyjniej: koła było okrągłe, a nie owalne, linie proste były naprawdę liniami prostymi, a nie sinusoidami o niewielkiej amplitudzie.

Był to ze pod każdym względem dar, który należało docenić.

Jednakże po kilku tygodniach zaczęły dawać o sobie znać słabe punkty urządzenia. Głównym była skłonność do zawieszania się wskutek zaklinowania papieru. Programiści o zdolnościach inżynierskich wkrótce zrozumieli przyczynę. Urzą-

dzenie, będące niegdyś fotokopiarką, było przeznaczone do działania pod jakimś nadzorem. Osoba ta zawsze była „pod ręką” i jeżeli zdarzało się zaklinowanie papieru, mogła od razu usunąć przyczynę zacięcia. Przyjmując to za rzecz oczywistą, inżynierowie Xeroxa skupiali uwagę na rozwiązywaniu innych problemów, ten pomijając. W ich założeniach czujna obecność użytkownika było częścią systemu.

Zamieniając urządzenie w drukarkę, w subtelny sposób zmienili również relację między nim i użytkownikiem. Poprzednio służyło jednej osobie, teraz przez sieć komputerową miało służyć wielu. Użytkownik nie stał już nad drukarką i nie nadzorował jej działania, lecz gdzieś z daleka z dowolnego miejsca sieci wysyłał polecenie drukowania. Przechodząc przez wiele urządzeń, docierało ono do celu, gdzie miało się przekształcić w drukowane dokumenty. Użytkownik nie miał żadnego kontaktu z drukarką aż do czasu, gdy po dotarciu do niej w celu odebrania wydruku przekonywał się, że jedynie niewielka jego część została wykonana.

Stallman był jednym z pierwszych, którzy ustalili przyczynę kłopotów, i pierwszym, który zaproponował, jak się ich pozbyć. Rozwiązał już podobny problem kilka lat temu, gdy laboratorium używało jeszcze starej drukarki. Nie mógł usunąć mechanicznych przyczyn zacinania się papieru, ale dogrzebał się do wnętrza programu, który na PDP-11 sterował działaniem drukarki i umieścił tam polecenie okresowego sprawdzania jej stanu oraz wysyłania raportu do głównego komputera sieci PDP-10. Aby być pewnym, że niefrasobliwość użytkownika, którego wydruk zawiesił działanie drukarki, nie spowoduje ogólnego zatoru wstrzymującego drukowanie całej kolejki, dopisał polecenia wysyłające zawiadomienie o awarii do wszystkich, których pliki czekały na wydrukowanie. Komunikat był prosty, coś w rodzaju: „Drukarka się zacięła. Zajmij się tym”, ale ponieważ był wysyłany do osób najbardziej zainteresowanych problemem, awarię zwykle szybko usuwano.

Było to obejście problemu, a nie jego rozwiązanie, ale obejście eleganckie i skuteczne. Nie potrafił rozwiązać problemu mechanicznego, ale zrobił to, co mógł zrobić najlepszego, zamykając w pętli drukarkę i zainteresowanych użytkowników. Dzięki kilku dodatkowym wierszom programu pracownicy AI Lab mogli zaoszczędzić tygodniowo średnio od 10 do 15 minut traconych poprzednio na zbędne wędrowanie do drukarki i z powrotem. W języku programistów można powiedzieć, że rozwiązanie Stallmana było wykorzystaniem skumulowanej inteligencji sieci.

„Jeżeli otrzymasz komunikat, nie możesz być pewny, że ktoś inny usunie zacięcie” — logicznie wnioskował Stallman. „Na pewno pójdziesz do drukarki. W ten sposób w ciągu minuty zjawia się tam dwie lub trzy zainteresowane osoby i zajmą się problemem. Z tych dwóch lub trzech osób przynajmniej jedna będzie wiedziała, co należy zrobić”.

Takie inteligentne rozwiązywanie problemów było znakiem firmowym AI Lab i grupy pracujących tam programistów. Najlepsi z nich pogardzali terminem „programista” i woleli nazywać się hakerami. Ta nazwa obejmowała bezmiar różnych działań, wszystko — od radosnej twórczej zabawy, po poważne prace ulepszające istniejące systemy komputerowe. W sposób niejawny zawierała w sobie także pamiętną z czasów pionierskich jankeską pomysłowość i zaradność. Bycie hakerem oznaczało akceptację poglądu, że napisanie programu to dopiero początek. Prawdziwym testem hakerskich umiejętności jest ulepszenie kodu⁴.

Taka filozofia bardzo odpowiadała firmom w rodzaju Xeroxa, które chętnie darowały swoje urządzenia i oprogramowanie instytucjom będącym znanymi skupiskami hakerów. Z poprawek przez nich wprowadzanych korzystano potem, wprowadzając na rynek nowe wersje i uaktualnienia. Zgodnie z terminologią firm, społeczności hakerskie były zwiększającym wartością społecznym kapitałem oraz dodatkowym działem badawczo-rozwojowym dostępnym za śmiesznie małe pieniądze.

Z powodu tej filozofii dawania i brania Stallman, po wykryciu zacinania się papieru w nowej drukarce Xeroxa, ze spokojem zaczął się zastanawiać, jak stary sposób omijania problemu przenieść do nowego systemu. Jednakże przyglądając się oprogramowaniu nowej drukarki laserowej, zrobił niepokojące odkrycie. Do drukarki nie dołączono oprogramowania, a przynajmniej nie było go w postaci czytelnej dla niego i jego kolegów. Dotychczas większość firm grzecznościowo publikowała kody źródłowe pozwalające poznać poszczególne polecenia sterujące urządzeniem. W tym przypadku Xerox dostarczył oprogramowanie w wersji prekompilowanej lub binarnej. Programiści mogli bez kłopotu otworzyć pliki, ale nie będąc ekspertami od dekodowania długich ciągów zer i jedynek, patrzyli na nie jak na nieznaną szyfr.

Choć Stallman wiele wiedział o komputerach, nie znał się na odczytywaniu plików binarnych. Jednakże jako haker miał do dyspozycji inne źródła informacji. Wiedział, że to tylko kwestia czasu, kiedy gdzieś w świecie jakiś haker na uniwersytecie lub w firmie rozszyfruje oprogramowanie nowej drukarki laserowej Xeroxa wraz z plikami pełnego kodu źródłowego. Wiedział także, że udostępni te pliki innym, gdyż pogląd o konieczności dzielenia się z innymi zdobytymi informacjami jest centralnym elementem filozofii hakerskiej.

Po kilkakrotnym zacięciu się drukarki Stallman przypomniał sobie sytuację sprzed lat. W laboratorium potrzebny był program obsługi sieci zwiększający wydajność współpracy PDP-11 z PDP-10. Hakerzy z AI Lab byli gotowi zająć się tym problemem, ale Stallman, absolwent Harvardu, przypomniał sobie, że podobny program został tam napisany przez programistów z działu badań komputerowych. W Harvardzie używano takich samych maszyn PDP-10 i PDP-11, ale z innym systemem operacyjnym. W Harvardzie przestrzegano również zasady, że instalowaniu jakichkolwiek programów na PDP-10 musi towarzyszyć ujawnianie ich kodów źródłowych.

Korzystając ze swych kontaktów z wydziałem komputerowym Harvardu, Stallman sporządził kopię pliku źródłowego interesującego go programu i zabrał do AI Lab. Potem przekształcił go do postaci wymaganej przez stosowany tam system operacyjny. Bez zbędnego krzyku i zamieszania AI Lab załatało istotną dziurę w swojej strukturze informatycznej. Stallman przy okazji zrobił kilka poprawek i dodał kilka nowych przydatnych funkcji, których brakowało w wersji opracowanej w Harvardzie. Zwiększyło to użyteczność programu, z którego — jak powiedział Stallman — „mogliśmy korzystać przez kilka lat”.

W latach siedemdziesiątych takie postępowanie można było porównać do pożyczania od sąsiada jakiegoś narzędzia lub torby cukru. Jedyną różnicą było to, że sporządzając kopię programu, Stallman nie pozbawił Harvardu możliwości korzystania z jego wersji oryginalnej. Hakerzy z Harvardu odnieśli przy tym dodatkową korzyść, gdyż w zamian za udostępnienie kodu źródłowego mogli swobodnie korzystać z poprawek wprowadzonych przez Stallmana. Nikt z Harvardu nie zainteresował się poprawionym programem, ale Stallman przypomina sobie programistę z prywatnej firmy Bolt & Beranek, który skorzystał z ulepszanego kodu, dodając do niego kilka zmian, które z kolei Stallman mógł ewentualnie swobodnie zintegrować z wersją działającą w AI Lab.

„Program mógłby się rozwijać jak miasto” — mówi Stallman, wspominając infrastrukturę informatyczną AI Lab. — „Części mogłyby być wymieniane i rekonstruowane, a nowe elementy dodawane. Można by, spoglądając na jakiś element, mówić «Hmm... sądząc po stylu, musiało to być napisane w połowie lat siedemdziesiątych»”.

Dzięki takiemu nakładaniu się warstw intelektualnych hakerzy z AI Lab i innych ośrodków mogli tworzyć rzeczy działające doskonale. Na zachodnim wybrzeżu naukowcy komputerowi z UC Berkeley współpracujący z kilkoma inżynierami systemowymi z AT&T zbudowali w ten sposób pełny system

operacyjny. Nazwany Uniksem, oparty na starszym, bardziej akademickim i poważanym systemie Multics, był dostępny dla każdego, kto zechciał zapłacić za taśmę do wykonania kopii i przesyłkę. Nie wszyscy programiści uczestniczący w tym przedsięwzięciu uważali się za hakerów, ale większość podzielała poglądy Richarda Stallmana. Jeżeli jakieś rozwiązanie problemu było dla jednego użytkownika, mogło przydać się również innemu. Dlaczego więc nie podzielić się nim z innymi, choćby dla zyskania dobrego samopoczucia?

Fakt, że Xerox nie ma ochoty na udzielenie kodów źródłowych, na początku niezbyt zdenerwował Stallmana. Starając się dotrzeć do kodów, zrazu nawet nie niepokoił Xeroxa. „Podarowali nam drukarkę laserową” — myślał — „czy wypada ich niepokoić dodatkowymi żądaniem?”

Jednakże po kilku nieudanych próbach dotarcia do plików źródłowych stawał się coraz bardziej podejrzliwy. Rok wcześniej stał się z doktorantem z Carnegie Mellon University⁵. Student Brian Reid był autorem użytecznego programu do formatowania tekstu o nazwie Scribe. Był to jeden z pierwszych programów pozwalających użytkownikowi na dołączanie do dokumentu przesyłanego przez sieć własnych definicji kształtu czcionek i stylów. Był to zwiastun HTML-a, lingua franca Word Wide Web. W roku 1979 Reid zdecydował się sprzedać Scribe’a działającej w rejonie Pittsburga firmie programistycznej Unilogic. Studencki rozdział w życiu Reida kończył się i, jak twierdził, starał się przekazać program komuś, kto będzie go rozwijał, strzegąc jednocześnie przed przedostaniem się do domeny publicznej. Jakby tego było mało, Reid zgodził się na umieszczenie w kodzie kilku zależnych od czasu funkcji — bomb z opóźnionym zapłonem, które powodowały deaktywację swobodnie rozprowadzanych kopii, jeżeli użytkownicy w ciągu 90 dni nie wnieśli stosownej opłaty. Jeżeli taka opłata została wniesiona, firma wysyłała kod wyłączający „podłożone ładunki”.

Dla Reida była to procedura doskonała. Scribe nie stał się publicznie dostępny, a Unilogic odzyskał zainwestowane pieniądze. Natomiast dla Stallmana była to oczywista zdrada etosu programisty. Zamiast honorować ideę dzielenia się wiedzą Reid pokazał firmom, w jaki sposób zmusić programistów do płacenia za dostęp do informacji.

Mijały tygodnie i próby Stallmana uzyskania dostępu do kodów źródłowych oprogramowania drukarki Xeroxa odbijały się od muru umowy milczenia. Wreszcie pojął, że teraz gra odbywa się według nowego scenariusza: „chcesz wiedzieć, to płać”. Zanim jednak zdążył się na ten temat wypowiedzieć lub zacząć działać, pocztą pantoflową dotarły do niego dobre wieści. Dowiedział się, że jeden z pracowników wydziału nauk komputerowych Carnegie Mellon University przed miesiącem odszedł z Xerox Palo Alto Research Center. Jak wieść niosła, nie dość, że przedtem pracował nad oprogramowaniem do interesującej Stallmana drukarki, to po przeniesieniu się na uniwersytet ponoć kontynuował tę pracę, traktując ją jako część swoich naukowych zadań.

Odkładając na bok swoje uprzednie podejrzenia, Stallman postanowił zatem odszukać tego osobnika podczas kolejnej wizyty w kampusie Carnegie Mellon.

Nie musiał długo czekać. W Carnegie Mellon też istniało laboratorium zajmujące się badaniami nad sztuczną inteligencją, więc w ciągu kilku miesięcy nadarzyła się okazja do złożenia tam roboczej wizyty. Odwiedził wówczas wydział komputerowy, gdzie w biurze skierowano go do człowieka prowadzącego projekt Xeroxa. Dotarłszy do wskazanego pokoju, zastał profesora przy pracy.

Rozmowa inżyniera z inżynierem była uprzejma lecz daremna. Przedstawiwszy się krótko jako gość z MIT, poprosił o udostępnienie kopii kodów źródłowych drukarki laserowej, aby mógł poprawić jej współdziałanie z PDP-11. Ku jego zaskoczeniu profesor odmówił spełnienia prośby.

„Powiedział, że zgodnie z daną obietnicą nie może mi dać kopii” — wspomina Stallman.

Pamięć jest ciekawą rzeczą. Dwadzieścia lat po tym fakcie taśma pamięci Stallmana pełna jest pustych miejsc. Nie pamięta, po co pojechał wtedy do Carnegie Mellon, ani pory roku, kiedy wizyta miała miejsce, nie pamięta także profesora, ani doktoranta, z którymi wtedy rozmawiał. Według Reida rozmówcą Stallmana mógł być Robert Sproull, dawniejszy pracownik badawczy Xerox PARC, a obecny dyrektor Sun Laboratories, oddziału badawczego konglomeratu komputerowo-technologicznego Sun Microsystems. W latach siedemdziesiątych Sproull kierował w Xerox PARC tworzeniem programów obsługi interesującego nas typu drukarek laserowych. W latach osiemdziesiątych zajmował się badaniami naukowymi na Carnegie Mellon, gdzie wśród innych zadań kontynuował prace nad oprogramowaniem drukarek.

„Kod, o który prosił Stallman, był najnowszym osiągnięciem sztuki programowania, napisanym przez Sproulla w ostatnim roku przed zatrudnieniem w Carnegie Mellon” — twierdzi Reid. — „Gdy dotarła do niego prośba Stallmana, był w Carnegie Mellon zapewne niecały miesiąc”.

Jednakże zapytany wprost o tamtą prośbę Sproull nie potrafi niczego wyjaśnić. W e-mailu napisał — „Nie mogę tego faktu skomentować, gdyż zupełnie go nie pamiętam”.

Mając dwóch uczestników rozmowy niemogących sobie przypomnieć jej istotnych szczegółów, łącznie z faktem, czy ta rozmowa w ogóle miała miejsce, trudno ocenić zapamiętaną przez Stallmana bezceremonialność odpowiedzi Sproulla. W wystąpieniach publicznych Stallman powtarza, że przyczyną odmowy Sproulla był kontrakt zawarty z Xerox Corporation dający jemu i innym dostęp do kodu źródłowego ale pod warunkiem nieudostępniania go osobom postronnym. Teraz takie zobowiązanie dochowania tajemnicy firmowej zwane w skrócie NDA od „non disclosure agreement” jest standardem

w firmach produkujących oprogramowanie, ale wówczas było nowością. Stanowiło ono odbicie wielkiej wartości handlowej, jaką dla Xeroxa przedstawiały drukarki, oraz potrzeby zdobycia informacji koniecznej do ich uruchomienia. „Xerox właśnie starał się zrobić z drukarki laserowej produkt handlowy” — wspomina Reid. — „Wariowali wówczas ze strachu, że kod może zostać ujawniony”.

Ale dla Stallmana NDA, czy jak tam się ta umowa nazywała, było czymś więcej. Było odcięciem go przez spółkę Xeroxa ze Sproullem, czy jak tam się ten osobnik nazywał, od potrzebnego mu kodu źródłowego i zniszczeniem systemu, który do tychczas działał sprawnie — traktowania programów jak wartości społecznej. Był jak wieśniak, któremu nagle wysechł kanał irygacyjny od stuleci dostarczający wodę na pola. Idąc wzdłuż tego kanału, trafił na zamykającą koryto tamę hydroelektrowni ze znakiem firmowym Xeroxa.

Dopiero po pewnym czasie przyjął do wiadomości, że Xerox zmusił programistę do przyjęcia wyznaczonych warunków i działania w ramach nowo wymyślonego systemu nakazanej dyskrekcji. Na początku uznał odmowę za skierowaną przeciwko sobie. Jako osoba czująca się niepewnie w kontaktach twarzą w twarz, sądził że najlepiej wpaść do kolegi programisty bez zapowiedzi, po prostu jak do sąsiada. Teraz, gdy spotkała go odmowa, sądził, że popełnił gafę. „Byłem tak wściekły, że nie mogłem zebrać myśli, więc się po prostu odwróciłem i wyszedłem bez słowa. Być może nawet trzasnąłem drzwiami. Kto wie?” — wspomina. „Pamiętam jedynie, że chciałem wyjść jak najszybciej”.

Choć upłynęło już 20 lat, czas nie stłumił złości, do tego stopnia, że Stallman uczynił z tego zdarzenia punkt zwrotny swego postępowania. W ciągu kilku następnych miesięcy na Stallmana i społeczność hakerów z AI Lab spadła seria zdarzeń, w porównaniu z którymi 30-sekundowy epizod w Carnegie Mellon wyglądał trywialnie. Niezależnie od sposobu posortowania

tych zdarzeń według ich ważności, przemieniły one Stallmana z hakera instynktownie nieufnego wobec wszelkich centralnych autorytetów w rycerza krucjaty walczącej o utrzymanie w świecie twórców oprogramowania tradycyjnych wartości: wolności, równości i braterstwa. Jednakże on sam uważa zdarzenie w Carnegie Mellon za najważniejsze.

„Dodało mi to odwagi do sformułowania idei, o której myślałem od dawna” — powiada — „że oprogramowanie powinno być wspólną, ogólnie dostępną własnością. Myślałem o tym od dawna, ale nigdy nie byłem pewien, jak należy to sformułować. Myśli nie były uporządkowane, co zaciemniało obraz, nie pozwalając na przedstawienie go światu w zrozumiałej postaci.”

Choć już wcześniejsze zdarzenia budziły jego gniew, twierdzi, że dopiero po scenie w Carnegie Mellon uświadomił sobie, że stanowiły one wstęp do ataku na świat zasad, które przez lata uważał za świętość. Należąc do elitarnej grupy programistów pracujących w najlepszych środkach światowych, był skłonny do całkowitego lekceważenia kompromisów i zobowiązań przyjmowanych przez jego kolegów dopóty, dopóki nie miało to żadnego związku z jego pracą. Do czasu zdarzenia z drukarką Xeroxa spoglądał z góry na urządzenia i programy, które inni użytkownicy zmuszeni byli tolerować w ponurym milczeniu. W rzadkich przypadkach, gdy jakiś program czynił wyłom w murach otaczających AI Lab, na przykład gdy dokonano zamiany starego szanowanego Z Niczym Niezgodnego Systemu Operacyjnego na komercyjną wersję TOPS 20⁶, Stallman wraz z kolegami miał swobodę pisania od nowa, zmieniania postaci i nazw poszczególnych elementów zgodnie z własnymi upodobaniami.

Teraz nowa drukarka stała się oznaką zmian, które nastąpiły. Działała świetnie, poza klinowaniem się papieru od czasu do czasu, ale zniknęła możliwość dokonywania zmian zgodnie z własnymi upodobaniami. Drukarka ta była oznaką przebu-

dzenia się całego przemysłu programistycznego. Oprogramowanie stało się dla firm rzeczą tak ważną, że uznały publikowanie kodów źródłowych za zbędne i niebezpieczne, gdyż mogło stać się dla konkurencji tanim źródłem pożądanых informacji. Z punktu widzenia Stallmana drukarka była koniem trojańskim. W ciągu następnej dekady oprogramowanie z zastrzeżonymi prawami własności wśliznęło się jak wąż do wnętrza AI Lab. Pojawiło się w postaci daru.

Stallman musiał przełknąć dwie gorzkie pigułki. Pierwszą było to, że Xerox dał kilku programistom AI Lab pełny dostęp do kolejnych darów w zamian za zobowiązanie do zachowania tajemnicy, a drugą — uświadomienie sobie, że być może on sam uległby takiej propozycji, gdyby spotkała go ona w młodszym wieku. Jednakże irytujące zdarzenie w Carnegie Mellon utrwaliło go w dotychczasowych przekonaniach. Po pierwsze, wywołało nieufność, z jaką od tej pory podchodził do wszystkich darów, a po drugie, postawiło przed nim niewygodne pytanie, jak on by się zachował, gdyby ktoś z kolegów wpadł do jego pokoju i poprosił o kod, którego nie powinien ujawnić.

„Było to moje pierwsze spotkanie z NDA i doprowadziło do wniosku, że umowy tego typu muszą mieć swoje ofiary” — twierdzi z przekonaniem Stallman — „W tym przypadku ofiarą byłem ja i AI Lab”.

Była to lekcja, która pomogła mu przetrwać burzliwe lata osiemdziesiąte — dziesięciolecie, w którym wielu jego kolegów z MIT odeszło z AI Lab i zaczęło działać na własny rachunek, podpisując umowy NDA. Fakt, że zwykle były to umowy na czas określony, był dla nich usprawiedliwieniem. Uważali, że prędzej, czy później oprogramowanie stanie się własnością publiczną. Obiecywali dotrzymanie tajemnicy w czasie najwcześniejszego wykluwania się oprogramowania, na jego najwcześniejszych etapach rozwoju i traktowali to jako konieczny kompromis, jako cenę, którą trzeba zapłacić za dopuszczenie hakerów do prac nad najlepszymi projektami. Dla Stallmana był to początek staczania się po równi pochyłej.

„Gdy ktoś zaprosił mnie do zdradzenia kolegów przez wzięcie udziału w takim przedsięwzięciu, przypomniałem sobie, jaką złość wywołało u mnie doświadczenie na sobie takiego traktowania.” — wspomina — „Odpowiedziałem zatem: «Dziękuję bardzo za chęć podarowania mi tak pięknego pakietu oprogramowania, ale nie mogę go przyjąć na proponowanych warunkach, więc obejdę się bez niego»”. Jak się szybko przekonał, rezygnowanie z takich ofert to coś więcej niż osobiste poświęcenie. Było to budowanie muru oddzielającego go od grupy innych hakerów, którzy choć dzielali jego pogardę dla okrywania pracy tajemnicą, skłonni byli stosować o wiele bardziej elastyczne kryteria moralne. Z upływem czasu stawał się coraz bardziej wyobcowany, nawet wewnątrz AI Lab i zaczęto mu przyczepiać etykietkę „ostatniego prawdziwego haker”. Powiększało to jego izolację od rynku coraz bardziej zdominowanego przez oprogramowanie z zastrzeżonymi prawami własności. Uznał, że odmawianie innym prawa do poznania kodów źródłowych jest nie tylko zdradą zasad działalności naukowej obowiązujących od czasów II wojny światowej, lecz także pogwałceniem „złotej zasady”, podstawowego prawa moralnego nakazującego traktować innych tak, jakbyśmy sami chcieli być przez nich traktowani.

Na tym polega znaczenie drukarki laserowej i starcia, które spowodowała. Bez tego, jak twierdzi Stallman, jego życie pobięłoby bardziej utartym trybem, balansując między zadowoleniem płynącym z dostatku pracującego dla rynku programisty, a frustracją spowodowaną pisaniem niewidocznego dla innych kodu programów. Nie byłoby potrzeby zachowania przejrzystości kodu, ani zajmowania się problemami pomijanymi przez innych. Co ważniejsze, nie byłoby też uzasadnionej złości i emocji, które wkrótce stały się siłą napędową jego działalności, na podobieństwo ideologii politycznej lub wiary w zasady etyczne.

„Tego dnia pojąłem i zdecydowałem, że istnieje coś, w czym nigdy nie będę mógł uczestniczyć” — twierdzi, mając przede wszystkim na myśli praktyki handlowania dla wygody osobistą wolnością, jak określa umowy NDA, oraz wspieranie takich etycznie podejrzanych działań przez powszechnie przyjęte wzorce kulturowe. „Postanowiłem, że nikt nigdy nie stanie się z mojej winy ofiarą takich poczynań, jakie mnie dotknęły”.

PRZYPISY

¹ Artificial Intelligence Laboratory — *przyp. tłum.*

² Instytut Techniczny Massachusetts — jedna z najbardziej prestiżowych uczelni technicznych Ameryki. MIT powstała w 1895 roku z inicjatywy kilkudziesięciu przedsiębiorców z okolic Nowego Jorku i Bostonu. Jej celem jest jednocześnie kształcenie studentów i prowadzenie badań podstawowych. MIT jest uczelnią całkowicie prywatną (patrz <http://pl.wikipedia.org/wiki/MIT>). Ponieważ w publikacjach polskich nazwa uczelni jest zwykle podawana w formie oryginalnej lub w postaci skrótu MIT, ja również nie będę jej zastępować polskim tłumaczeniem — *przyp. tłum.*

³ DTP (Desktop Publishing) to określenie oprogramowania oraz urządzeń komputerowo-technicznych do edycji tekstu i grafiki, a często także całego procesu przygotowania papierowej publikacji do druku, czyli wstawiania i łamania tekstów oraz rozmieszczania ewentualnych ilustracji. Zastosowanie komputerów pozwoliło na przeniesienie tego procesu z hal drukarskich na biurko. Stąd określenie „desktop”, czyli „nabiurkowe”, mieszczące się na biurku lub stole — *przyp. tłum.*

⁴ Pojęcie „haker” zostało szerzej wyjaśnione w dodatku B.

⁵ Uczelnie powstała w Pittsburgu w roku 1900 z fundacji przemysłowca i filantropa Andrew Carnegiego. W ciągu ponad 100 lat istnienia trzykrotnie zmieniała nazwę, od Carnegie Technical Schools przez pierwszych 12 lat istnienia, przez Carnegie Institute of Technology od roku 1912 do 1967, po obecną nazwę Carnegie Mellon University. Dzięki hojnej dotacji Richarda K. Mellona w 1965 został utworzony wydział zajmujący się naukami komputerowych — Computer Science Department, którego działalność stała się podstawą doskonalej światowej reputacji tej uczelni w dziedzinie nauk najnowszej technologii — *przyp. tłum.*

⁶ TOPS 20 pojawił się na rynku w roku 1976 jako system operacyjny opracowany przez firmę DEC dla komputerów PDP-10. Był rozbudowaną wersją starszego systemu TENEX zaprojektowanego w roku 1969 przez firmę Bolt Beranek & Newman Inc. TENEX został zakupiony przez DEC i w latach 1973-76 przekształcony w TOPS 20. Historię tego systemu można przeczytać po angielsku np. pod adresem: <http://www.linique.com/dlm/tenex/hbook.html> — *przyp. tłum.*